
Insurance Developer Guide

Version 66.0, Spring '26



CONTENTS

Chapter 1: Get Started with Insurance Developer Resources	1
Chapter 2: Insurance Brokerage	3
Insurance Brokerage Business APIs	4
Resources	5
Requests	16
Responses	25
Insurance Brokerage Standard Invocable Actions	49
Compute Producer Splits Action	49
Create Producer Commissions Action	51
Find Insurance Policy Action	53
Insurance Brokerage Metadata API Types	55
Flow for Insurance Brokerage	55
Insurance Brokerage Apex Reference	56
ConnectApi Namespace	56
Chapter 3: Insurance Claims Management	69
Insurance Claims Business APIs	70
Resources	70
Requests	94
Responses	108
Insurance Claims Standard Invocable Actions	127
Calculate Adjustments Action	128
Cancel Claim Coverage Payment Detail Action	130
Create Claim Action	131
Create Claim Coverage Payment Detail Action	134
Create Policy Limits Action	136
Delete Claim Coverage Payment Detail Action	138
Edit Claim Coverage Payment Detail Action	139
Get Claim Action	141
Get Insurance Policy Current Standings Action	144
Open Claim Coverage Action	146
Pay Ex Gratia Claim Coverage Payment Detail Action	148
Pay Claim Coverage Payment Detail Action	150
Recalculate Adjustments Action	152
Update Claim Action	153
Update Claim Coverage Action	156
Verify Policy Coverage Action	158
Insurance Claims Apex Reference	159

ConnectApi Namespace	160
Insurance Claims Metadata API Types	242
Flow for Insurance Claims	242
Insurance Claims Developer Journey	243
Claims Processing Journey	244
Chapter 4: Insurance Group Benefits	248
Insurance Group Benefits Business APIs	249
Resources	249
Requests	272
Responses	293
Insurance Group Benefits Apex Reference	319
ConnectApi Namespace	319
Insurance Group Benefits Standard Invocable Actions	361
Create Contacts and Users from Group Census Members Action	362
Create Contract From Quote Action	366
Create Person Accounts and Users from Group Census Members Action	368
Get Group Census Members Action	371
Get Group Census Members and Plans Action	374
Get Member Ratings and Contributions Action	378
Process Member Enrollment Action	381
Save Group Census Members Data Action	385
Save Group Census Member Plans Data Action	388
Validate Group Census Members Action	392
Validate Group Census Members Plans Action	394
Insurance Group Benefits Metadata API Types	398
Flow for Insurance Group Benefits	398
Chapter 5: Insurance Policy Administration	400
Insurance Policy Business APIs	401
Resources	402
Requests	419
Responses	428
Insurance Policy Standard Invocable Actions	437
Cancel Insurance Policy Action	438
Create Out-of-sequence Policy Endorsement Action	440
Endorse Insurance Policy Action	444
Endorse Multi-Root Insurance Policy Action	446
Get Insurance Policy Action	453
Issue Insurance Policy Action	454
Issue Multi-Root Insurance Policy Action	457
Reinstate Insurance Policy Action	463
Renew Insurance Policy Action	467
Renew Insurance Policies Action	470

Renew Multi-Root Insurance Policy Action	471
Insurance Policy Administration Metadata API Types	477
Flow for Insurance Policy Administration	477
Insurance Policy Standard Objects	478
InsProductClauseVariableMap	479
InsuranceClause	481
InsurancePolicyProductClause	484
InsuranceProductClause	488
QuotelineItemInsProductClause	491
Insurance Policy Administration Apex Reference	493
ConnectApi Namespace	493
Insurance Policy Developer Journey	533
Create and Manage Insurance Policies by Using Salesforce Quoting	533
Create and Manage Insurance Policies Without Using Salesforce Quoting	542
Cancel Policy	549
Chapter 6: Insurance Product Administration	551
Insurance Product Administration Business APIs	552
Resources	552
Requests	559
Responses	569
Insurance Product Administration Invocable Actions	571
Invoke Underwriting Rules Action	571
Insurance Product Administration Metadata API Types	575
Flow for Insurance Product Administration	575
Chapter 7: Insurance Quoting	576
Insurance Quoting Business APIs	577
Resources	578
Requests	601
Responses	629
Insurance Quoting Standard Invocable Actions	655
Add Eligible Insurance Clauses Action	655
Create Insurance Quote Action	660
Create Insurance Rating Action	665
Generate Insurance Clauses Action	676
Get Eligible Insurance Clauses Action	681
Get Insurance Quote Details Action	684
Get Insurance Rating Input Action	688
Reprice Insurance Product Action	694
Update Insurance Quote Action	707
Insurance Quoting Metadata API Types	712
Flow for Insurance Quoting	713
Insurance Quoting Developer Journey	714

Contents

Build a Self-service Quote and Policy Creation Functionality for Community Portal	
Users	714
Calculate Quote Price Without Creating a Formal Quote	715
Implement Quote Creation Functionality for Insurance Representatives	715
Implement Quote Update Functionality for Insurance Representatives	715
Chapter 8: Insurance Apex Reference	716
runtime_industries_insurance Namespace	717
AddEligibleInsuranceClausesOptions Class	717
CreateInsuranceQuoteOptions Class	719
CreateInsuranceRatingOptions Class	723
GenerateInsuranceClausesOptions Class	727
UpdateInsuranceQuoteOptions Class	730
Chapter 9: Error Codes and Responses	735
Common Error Codes	736
Insurance Policy Administration Error Codes	736
Insurance Quoting Errors	737

CHAPTER 1 Get Started with Insurance Developer Resources

Use the developer resources of Insurance Cloud to manage end-to-end insurance processes. Create and configure quotes, manage policy transactions, and automate processes related to brokerage policies and commissions.

Learn more about the developer resources that are available for these components.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise, Unlimited, and Developer** Editions

Insurance Brokerage

Manage insurance policy lifecycles. Automate processes related to Policy and Commissions Management. See [Insurance Brokerage](#) on page 3.

Insurance Claims Management

Manage insurance claims by automating, streamlining, and orchestrating the full lifecycle of a claim, from coverage validation to final settlement. See [Insurance Claims Management](#) on page 69.

Insurance Group Benefits

Manage insurance group benefit plans that are offered to and covering a group of people, such as employees and their dependents. See [Insurance Group Benefits](#) on page 248.

Insurance Policy Administration

Manage policy lifecycles across quoting, issuance, endorsements, billing, and renewals. See [Insurance Policy Administration](#) on page 400.

Insurance Product Administration

Define the products you sell to customers, including their classifications, categories, and attributes. Create customized price adjustment methods and pricing procedures. Define rules that determine how your products behave when they're quoted, sold, and serviced. See [Insurance Product Administration](#) on page 551.

Insurance Quoting

Automate quoting experiences as part of a cohesive customer-centric experience. Create, update, configure, and delete quotes and quote line items. See [Insurance Quoting](#) on page 576.

CHAPTER 2 Insurance Brokerage

In this chapter ...

- [Insurance Brokerage Business APIs](#)
- [Insurance Brokerage Standard Invocable Actions](#)
- [Insurance Brokerage Metadata API Types](#)
- [Insurance Brokerage Apex Reference](#)

Use the Insurance Brokerages developer resources to automate processes related to Policy and Commissions Management. Invoke policy lifecycle services such as Renew, Repurpose, and Cancel using context service. Manage plan benefits using a flexible service layer and Product Catalog.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Insurance Brokerage Business APIs

Use the Insurance Brokerage Business APIs to manage policy lifecycle, brokerage commissions, and plan benefits.

This table lists the available Insurance Brokerage resources.

Resource	Description
/connect/insurance/brokerage/policies/bulk-cancel (POST)	Bulk cancel policies with the same for specified insurance policy IDs.
/connect/insurance/brokerage/policies/bulk-renew (POST)	Bulk renew policies for specified insurance policy IDs.
/connect/insurance/brokerage/commission/process (POST)	Process commission statements in an asynchronous manner using a preconfigured batch job.
/connect/insurance/brokerage/policies/policyId/expected-rates (PATCH)	Calculate the premium, fee, or premium-equivalent amount at InsuranceRatePlanLineItem based on the underlying rate plan.
/connect/insurance/brokerage/policies/policyId/expected-revenue (PATCH)	Calculate the expected revenue, which is derived both from an Insurance Rate Plan Commission object, such as a flat fee and percentage premium, and the premium, fee, or premium equivalent amount on Insurance Rate Plan.
/connect/insurance/brokerage/coverages/coverageId/benefits (POST)	Get, create, update and delete Insurance Policy Coverage Networks and Insurance Policy Coverage Benefits of an Insurance Policy Coverage.
/connect/insurance/brokerage/coverages/coverageId/benefits/product (GET)	Get the product model details along with child components and their attributes for a given insurance policy coverage.
/connect/insurance/brokerage/policies/policyId/renew (POST)	Renew policy for the specified insurance policy ID.
/connect/insurance/billing/billing-schedules (POST)	Create billing schedules from insurance policy transactions to generate financial records for policy lifecycle events.

[Resources](#)

Learn about the available Insurance Brokerage API resources.

[Requests](#)

Learn about the available Insurance Brokerage API requests.

[Responses](#)

Learn about the response bodies of Insurance Brokerage APIs.

Resources

Learn about the available Insurance Brokerage API resources.

[Bulk Cancel Policy \(POST\)](#)

Bulk cancel policies for the specified insurance policy IDs.

[Bulk Renew Policy \(POST\)](#)

Bulk renew policies for the specified insurance policy IDs.

[Commission Processing \(POST\)](#)

Process commission statements in an asynchronous manner using a preconfigured batch job.

[Cost Calculation \(PATCH\)](#)

Calculate the premium, fee, or premium-equivalent amount at InsuranceRatePlanLineItem based on the underlying rate plan. The InsuranceRatePlanLineItem premium, fees, or premium equivalent amount is rolled up to the Insurance Rate Plan, Insurance Coverage, and Insurance Policy Premium Amount.

[Expected Revenue \(PATCH\)](#)

Calculate the expected revenue, derived both from an Insurance Rate Plan Commission object, such as a flat fee and percentage premium, and the premium, fee, or premium equivalent amount on Insurance Rate Plan. The expected revenue amount is rolled up from Insurance Rate Plan to Insurance Coverage and Insurance Policy expected revenue amount.

[Insurance Billing Schedules \(POST\)](#)

Create billing schedules from insurance policy transactions to generate financial records for policy lifecycle events.

[Plan Benefits \(GET, POST, PATCH, DELETE\)](#)

Get, create, update and delete Insurance Policy Coverage Networks and Insurance Policy Coverage Benefits of an Insurance Policy Coverage.

[Plan Benefits Product Model \(GET\)](#)

Get the product model details along with child components and their attributes for a given insurance policy coverage.

[Renew Policy \(POST\)](#)

Renew policy for the specified insurance policy ID.

Bulk Cancel Policy (POST)

Bulk cancel policies for the specified insurance policy IDs.

Resource

```
/connect/insurance/brokerage/policies/bulk-cancel
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/insurance/brokerage/policies/bulk-cancel
```

Available version

64.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "insurancePolicyIds" : ["0YTx0000000001GAA", "0YTx0000000001GAB"],
  "fieldValueMap": { "Name": "test" }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs to store details of insurance policy. The fieldValueMap expect the keys to be API name and is case sensitive.	Required	64.0
insurancePolicyIds	Object	List of insurance policy IDs that must be renewed from the same carrier.	Required	64.0

Response body for POST[Insurance Policy Bulk](#)

Bulk Renew Policy (POST)

Bulk renew policies for the specified insurance policy IDs.

Resource

```
/connect/insurance/brokerage/policies/bulk-renew
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/brokerage/policies/bulk-renew
```

Available version

64.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "insurancePolicyIds" : ["0YTxx0000000001GAA", "0YTxx0000000001GAB"],
  "fieldValueMap": { "Name": "test" }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs to store details of insurance policy. The fieldValueMap expect the keys to be API name and is case sensitive.	Required	64.0
insurancePolicyIds	Object	List of insurance policy IDs that must be renewed from the same carrier.	Required	64.0

Response body for POST[Insurance Policy Bulk](#)

Commission Processing (POST)

Process commission statements in an asynchronous manner using a preconfigured batch job.

Resource

```
/connect/insurance/brokerage/commission/process
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/brokerage/commission/process
```

Available version

63.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "commissionStatementIds": [
    "199TC0000003kxZYAQ"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
commissionStatementIds	Object	List of commission statement IDs to be processed.	Required	63.0

Response body for POST

[Insurance Commission Processing](#)

Cost Calculation (PATCH)

Calculate the premium, fee, or premium-equivalent amount at InsuranceRatePlanLineItem based on the underlying rate plan. The InsuranceRatePlanLineItem premium, fees, or premium equivalent amount is rolled up to the Insurance Rate Plan, Insurance Coverage, and Insurance Policy Premium Amount.

Resource

```
/connect/insurance/brokerage/policies/policyId/expected-rates
```

Resource example

```
https://yourname.salesforce.com/services/data/connect/insurance/brokerage/policies/00000000944/expected-rates?businessObjectType=InsuranceRatePlanLineItem&CSID=000000060A
```

Available version

63.0

HTTP methods

PATCH

Path parameter for PATCH

Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of parent policy for which the expected cost is calculated.	Required	63.0

Request parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
businessObjectType	String	Name of business SObject to calculate premium. Possible values are: - InsurancePolicy - InsurancePolicyCoverage - InsuranceRatePlan - InsuranceRatePlanLineItem	Optional	63.0

Parameter Name	Type	Description	Required or Optional	Available Version
recordId	String	ID of business SObject for which premium calculation is done.	Optional	63.0

Request body for PATCH

JSON example

```
{
  "action": "rollup",
  "expressionSetName": "expressionSetName",
  "ratePlanLineItems": [
    {
      "id": "xxxxxx",
      "rate": 100,
      "enrollmentCount": 1000,
      "custom_field_1": "xxxxxx"
    },
    {
      "id": "yyyyy",
      "rate": 50,
      "enrollmentCount": 200,
      "custom_field_1": "xxxxxx"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
action	String	Course of action for processing RatePlanLineItems. Valid values: - rollup: Calculates premium and roll ups it to Insurance Policy level. - calculate: Calculates premium and returns it in the output.	Optional	63.0
expressionSetName	String	Name of the expression set for cost calculation. If not specified, the value from the InsPolicyManagementConfig setup object is taken.	Optional	63.0
ratePlanLineItems	List<Map<String, Object>>	List of insurance rate plan line items.	Optional	63.0

Request body for PATCH**JSON example**

```
{
  "expressionSetName": "expressionSetName",
  "ratePlanCommissions": [
    {
      "Name": "RPC-1",
      "Type": "GradedPercent",
      "amountStartRange": 0,
      "amountEndRange": 1000
    },
    {
      "Name": "RPC-1",
      "Type": "GradedPercent",
      "amountStartRange": 1001,
      "amountEndRange": 100000
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>expressionSetName</code>	String	Name of expression set to be used for expected revenue calculation. If unspecified, a value from the <code>InsPolicyManagementConfig</code> setup object is taken.	Optional	63.0
<code>ratePlanCommissions</code>	List	List of rate plan commissions to be used for expected revenue calculation. This is a valid input only when <code>businessObjectType</code> is <code>InsuranceRatePlan</code> .	Optional	63.0

Response body for PATCH[Expected Revenue](#)

Insurance Billing Schedules (POST)

Create billing schedules from insurance policy transactions to generate financial records for policy lifecycle events.

This API automates the creation of billing schedules and invoices for key policy events, including Issuance, Endorsement, Cancellation, and Renewal. The service supports complex scenarios such as out-of-sequence endorsements and cancellations by calculating transient transactions to ensure accurate net billing.

Resource

```
/connect/insurance/billing/billing-schedules
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/billing/billing-schedules
```

Available version

66.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "insPolicyTransactionId": "0k9xx0000000001AAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
insPolicyTransactionId	String	ID of the insurance policy transaction record to get related transaction details and billing configuration.	Required	66.0

Response body for POST
[Insurance Billing Schedules Response](#)
Plan Benefits (GET, POST, PATCH, DELETE)

Get, create, update and delete Insurance Policy Coverage Networks and Insurance Policy Coverage Benefits of an Insurance Policy Coverage.

Resource

```
/connect/insurance/brokerage/coverages/coverageId/benefits
```

Available version

63.0

HTTP methods

GET, POST, PATCH, DELETE

Response body for GET
[PlanBenefits](#)
Request body for POST**JSON example**

```
{
  "networks": [
    {
      "network": "First Network Tier",
      "name": "Preferred Network"
    }
  ]
}
```

```

    },
    {
      "network": "Second Network Tier",
      "name": "In Network"
    }
  ],
  "benefits": [
    {
      "productId": "01tSG000001lyBFxxY",
      "productCompGgpId": "0y7SG0000000ZiXxxU",
      "productName": "Chiropractic Care",
      "network": "First Network Tier",
      "attributes": [
        {
          "attributeDefinitionId": "0tjSG0000001embxxA",
          "name": "Copay",
          "value": 311
        },
        {
          "attributeDefinitionId": "0tjSG0000001YmMxxU",
          "name": "Deductible",
          "value": "100"
        }
      ]
    }
  ]
}

```

Properties

Name	Type	Descriptions	Required or Optional	Available Version
attributes	Benefit Attributes on page 20	List of benefit attributes.	Optional	63.0
network	String	The tier type of the network.	Optional	63.0
networks	Networks on page 23	Insurance Policy Coverage Networks for creating plan benefits.	Required	63.0
productCompGgpId	String	ID of product component group.	Optional	63.0
productId	String	ID of the product.	Required	63.0
productName	String	Name of the product.	Optional	63.0

Response body for POST

[PlanBenefitsUpsert](#) on page 45

Request body for PATCH**JSON example**

```
{
  "networks": [
    {
      "networkId": "1HXSG0000002xxxxAM",
      "name": "Preferred Network Updated"
    },
    {
      "networkId": "1HXSG0000002xxxxA2",
      "name": "In Network Updated"
    },
    {
      "network": "Third Network Tier",
      "name": "Out Network"
    }
  ],
  "benefits": [
    {
      "benefitId": "1HSSG0000005Ws7xxE",
      "productId": "01tSG0000011yBFxxY",
      "productCompGgpId": "0y7SG0000000ZiXxxU",
      "productName": "Chiropractic Care",
      "attributes": [
        {
          "covBnftAttrId": "1HYSG00000002GPxxY",
          "attributeDefinitionId": "0tjSG0000001embxxA",
          "name": "Copay",
          "value": 311
        },
        {
          "covBnftAttrId": "1HYSG00000002GQxxY",
          "attributeDefinitionId": "0tjSG0000001YmMxxU",
          "name": "Notes",
          "value": "Preferred network limitations apply."
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
benefits	Benefits Input []	The benefits of the Insurance Plan Coverage.	Optional	63.0
networks	Networks Input []	The list of networks.	Optional	63.0

Response body for PATCH[PlanBenefitsUpsert](#)**Request parameters for DELETE**

Parameter Name	Type	Description	Required or Optional	Available Version
benefitIds	String[]	ID of Insurance Policy Coverage Benefit. If benefitIds are provided, the service will only delete the respective Insurance Policy Coverage Benefit records.	Optional	63.0
networkIds	String[]	ID of the Insurance Policy Coverage Network. If networkIds are provided, the service will only delete the respective Insurance Policy Coverage Network records and their associated Insurance Policy Coverage Benefit records.	Optional	63.0

Plan Benefits Product Model (GET)

Get the product model details along with child components and their attributes for a given insurance policy coverage.



Note: See [Prerequisites](#) to define the product model structure. If this model isn't followed, the API throws an error.

Resource

```
/connect/insurance/brokerage/coverages/coverageId/benefits/product
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/brokerage/coverages/coverageId/benefits/product
```

Available version

63.0

HTTP methods

GET

Path parameter for GET

Name	Type	Description	Required or Optional	Available Version
coverageId	String	ID of insurance policy coverage.	Required	63.0

Response body for GET[Plan Benefits Product Model](#)

Renew Policy (POST)

Renew policy for the specified insurance policy ID.

Resource

```
/connect/insurance/brokerage/policies/policyId/renew
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/brokerage/policies/policyId/renew
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "fieldValueMap": { "Name": "demo" }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs that store details of insurance policy.	Required	63.0

Response body for POST

[Insurance Policy Process](#)

Requests

Learn about the available Insurance Brokerage API requests.

[Cost Calculation Input](#)

Input representation of the request to calculate brokerage cost.

[Expected Revenue Input](#)

Input representation of the request to calculate expected brokerage revenue.

[Insurance Commission Processing Input](#)

Input representation details of insurance commission processing.

[Insurance Billing Schedules Input](#)

Details for creating billing schedules from an insurance transaction.

[Insurance Policy Bulk Input](#)

Input representation details of the request to bulk renew policies. Send the list of insurance policy IDs and field value map of insurance policy fields for bulk policies renewal.

[Insurance Policy Renew Input](#)

Input representation of the details of the request to renew policy.

[Plan Benefit Attribute Input](#)

Represents details of the plan benefits attributes for an Insurance Policy Coverage.

[Plan Benefit Input](#)

Represents details of the plan benefits for an Insurance Policy Coverage.

[Plan Benefit Update Input](#)

Represents details of the plan benefits for an Insurance Policy Coverage.

[Networks Input](#)

Represents Insurance Policy Coverage Networks for creating plan benefits.

[Update Plan Benefit Input](#)

Update Insurance Policy Coverage Networks and Insurance Policy Coverage Benefits by using the ID of an Insurance Policy Coverage.

Cost Calculation Input

Input representation of the request to calculate brokerage cost.

JSON example

```
{
  "action": "rollup",
  "expressionSetName": "expressionSetName",
  "ratePlanLineItems": [
    {
      "id": "xxxxxx",
      "rate": 100,
      "enrollmentCount": 1000,
      "custom_field_1": "xxxxxx"
    },
    {
      "id": "yyyyyy",
      "rate": 50,
      "enrollmentCount": 200,
      "custom_field_1": "xxxxxx"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
action	String	Course of action for processing RatePlanLineItems. Valid values: - rollup: Calculates premium and roll ups it to Insurance Policy level. - calculate: Calculates premium and returns it in the output.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
expressionSetName	String	Name of the expression set for cost calculation. If not specified, the value from the <code>InsPolicyManagementConfig</code> setup object is taken.	Optional	63.0
ratePlanLineItems	List<Map<String, Object>>	List of insurance rate plan line items.	Optional	63.0

Expected Revenue Input

Input representation of the request to calculate expected brokerage revenue.

JSON example

```
{
  "expressionSetName": "expressionSetName",
  "ratePlanCommissions": [
    {
      "Name": "RPC-1",
      "Type": "GradedPercent",
      "amountStartRange": 0,
      "amountEndRange": 1000
    },
    {
      "Name": "RPC-1",
      "Type": "GradedPercent",
      "amountStartRange": 1001,
      "amountEndRange": 100000
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
expressionSetName	String	Name of expression set to be used for expected revenue calculation. If unspecified, a value from the <code>InsPolicyManagementConfig</code> setup object is taken.	Optional	63.0
ratePlanCommissions	List	List of rate plan commissions to be used for expected revenue calculation. This is a valid input only when <code>businessObjectType</code> is <code>InsuranceRatePlan</code> .	Optional	63.0

Insurance Commission Processing Input

Input representation details of insurance commission processing.

JSON example

```
{
  "commissionStatementIds": [
    "199TC0000003kxZYAQ"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
commissionStatementIds	Object	List of commission statement IDs to be processed.	Required	63.0

Insurance Billing Schedules Input

Details for creating billing schedules from an insurance transaction.

JSON example

```
{
  "insPolicyTransactionId": "0k9xx0000000001AAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
insPolicyTransactionId	String	ID of the insurance policy transaction record to get related transaction details and billing configuration.	Required	66.0

Insurance Policy Bulk Input

Input representation details of the request to bulk renew policies. Send the list of insurance policy IDs and field value map of insurance policy fields for bulk policies renewal.

JSON example

```
{
  "insurancePolicyIds" : ["0YTxx0000000001GAA", "0YTxx0000000001GAB"],
  "fieldValueMap": { "Name": "test" }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs to store details of insurance policy. The fieldValueMap expect the keys to be API name and is case sensitive.	Required	64.0
insurancePolicyIds	Object	List of insurance policy IDs that must be renewed from the same carrier.	Required	64.0

Insurance Policy Renew Input

Input representation of the details of the request to renew policy.

JSON example

```
{
  "fieldValueMap": { "Name": "demo" }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs that store details of insurance policy.	Required	63.0

Plan Benefit Attribute Input

Represents details of the plan benefits attributes for an Insurance Policy Coverage.

JSON example

```
{
  "attributes": [
    {
      "attributeDefinitionId": "0tjSG0000001embxxA",
      "name": "Copay",
      "value": 400
    },
    {
      "attributeDefinitionId": "0tjSG00000020yPxxQ",
      "name": "Deductible waiver",
      "value": "Deductible waived"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
~~attributeId~~	String	ID of the Attribute Definition.	Required	63.0
name	String	Name of the attribute.	Optional	63.0
value	String	Value of the attribute.	Required	63.0

Plan Benefit Input

Represents details of the plan benefits for an Insurance Policy Coverage.

JSON example

```
{
  "networks": [
    {
      "network": "First Network Tier",
      "name": "Preferred Network"
    },
    {
      "network": "Second Network Tier",
      "name": "In Network"
    }
  ],
  "benefits": [
    {
      "productId": "01tSG000001lyBFxxY",
      "productCompGgpId": "0y7SG0000000ziXxxU",
      "productName": "Chiropractic Care",
      "network": "First Network Tier",
      "attributes": [
        {
          "attributeDefinitionId": "0tjSG0000001embxxA",
          "name": "Copay",
          "value": 311
        },
        {
          "attributeDefinitionId": "0tjSG0000001YmMxxU",
          "name": "Deductible",
          "value": "100"
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Descriptions	Required or Optional	Available Version
attributes	Benefit Attributes on page 20	List of benefit attributes.	Optional	63.0
network	String	The tier type of the network.	Optional	63.0
networks	Networks on page 23	Insurance Policy Coverage Networks for creating plan benefits.	Required	63.0
productCompGgpId	String	ID of product component group.	Optional	63.0
productId	String	ID of the product.	Required	63.0
productName	String	Name of the product.	Optional	63.0

Plan Benefit Update Input

Represents details of the plan benefits for an Insurance Policy Coverage.

JSON example

```
{
  "benefits": [
    {
      "benefitId": "1HSSG0000005Ws7xxE",
      "productId": "01tSG0000011yBFxxY",
      "productCompGgpId": "0y7SG0000000ZiXxxU",
      "productName": "Chiropractic Care",
      "attributes": [
        {
          "covBnftAttrId": "1HYSG00000002GPxxY",
          "attributeDefinitionId": "0tjSG00000001embxxA",
          "name": "Copay",
          "value": 311
        },
        {
          "covBnftAttrId": "1HYSG00000002GQxxY",
          "attributeDefinitionId": "0tjSG00000001YmMxxU",
          "name": "Notes",
          "value": "Preferred network limitations apply."
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
benefitId	String	ID of the Insurance Policy Coverage Benefit.	Required	63.0
productId	String	ID of the product.	Optional	63.0
productCompGgpId	String	ID of product component group	Optional	63.0
productName	String	Name of the product.	Optional	63.0
network	String	The tier type of the network.	Optional	63.0
attributes	String	List of attributes.	Optional	63.0
covBnftAttrId	String	ID of Insurance Policy Coverage Benefit Attribute.	Required	63.0
name	String	Name of the attribute.	Optional	63.0
value	String	Value of the attribute.	Optional	63.0
attributeDefinitionId	String	ID of the Attribute Definition.	Optional	63.0

Networks Input

Represents Insurance Policy Coverage Networks for creating plan benefits.

JSON example

```
{
  "networks": [
    {
      "network": "First Network Tier",
      "name": "Preferred Network Updated"
    },
    {
      "network": "Second Network Tier",
      "name": "In Network Updated"
    },
    {
      "network": "Third Network Tier",
      "name": "Out Network"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
name	String	The name of the network.	Required	63.0

Name	Type	Description	Required or Optional	Available Version
network	String	Tier type of the network.	Required	63.0

Update Plan Benefit Input

Update Insurance Policy Coverage Networks and Insurance Policy Coverage Benefits by using the ID of an Insurance Policy Coverage.

JSON example

```
{
  "networks": [
    {
      "networkId": "1HXSG0000002xxxxAM",
      "name": "Preferred Network Updated"
    },
    {
      "networkId": "1HXSG0000002xxxxA2",
      "name": "In Network Updated"
    },
    {
      "network": "Third Network Tier",
      "name": "Out Network"
    }
  ],
  "benefits": [
    {
      "benefitId": "1HSSG0000005Ws7xxE",
      "productId": "01tSG0000011yBFxxY",
      "productCompGgpId": "0y7SG0000000ZiXxxU",
      "productName": "Chiropractic Care",
      "attributes": [
        {
          "covBnftAttrId": "1HYSG00000002GPxxY",
          "attributeDefinitionId": "0tjSG0000001embxxA",
          "name": "Copay",
          "value": 311
        },
        {
          "covBnftAttrId": "1HYSG00000002GQxxY",
          "attributeDefinitionId": "0tjSG0000001YmMxxU",
          "name": "Notes",
          "value": "Preferred network limitations apply."
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
benefits	Benefits Input []	The benefits of the Insurance Plan Coverage.	Optional	63.0
networks	Networks Input []	The list of networks.	Optional	63.0

Responses

Learn about the response bodies of Insurance Brokerage APIs.

[Amount with Frequency](#)

Output representation of the amount details with frequency.

[Attribute Detail](#)

Output representation of product attribute details.

[Attribute Picklist Value](#)

Output representation of the picklist value details.

[Benefits](#)

Output representation details of the plan benefits for an insurance policy coverage plan.

[Cost Calculation](#)

Output representation details of the cost calculation.

[Expected Revenue Coverage](#)

Output representation details of the coverage and associated expected revenue.

[Expected Revenue Policy](#)

Output representation of the associated expected revenue details for a policy, such as coverages and rate plans.

[Expected Revenue](#)

Output representation of the details of the expected revenue.

[Commission Processing Details](#)

Output representation details of the commission processing.

[Insurance Billing Schedules Response](#)

Details of the billing schedule creation results.

[Insurance Policy Bulk](#)

Output representation details of bulk renew policy and bulk cancel policy requests.

[Insurance Policy Process](#)

Output representation details of the renewed policy.

[Plan Benefits Product Model Details](#)

Output representation of the product model details and associated product components.

[Rate Plan Commission](#)

Output representation details of the rate plan commission.

[Rate Plan](#)

Output representation of the details of the rate plans associated directly with the policy or through coverages.

[Product](#)

Output representation of product details.

[Plan Benefits Attribute](#)

Represents details of the plan benefits attributes for an Insurance Policy Coverage.

[Plan Benefits Attributes](#)

Output representation details of the benefits attributes for an insurance policy coverage plan.

[Plan Benefit Networks](#)

Output representation details of the insurance policy coverage networks.

[Plan Benefits](#)

Output representation details of the create and update plan benefits request.

[Plan Benefits Details](#)

Output representation details of the insurance policy benefits.

[Product](#)

Output representation of product details.

Amount with Frequency

Output representation of the amount details with frequency.

JSON example

This example shows a sample success response.

```
"amountWithFrequency": {  
  "annually": 1000.3,  
  "monthly": 100.03,  
  "quarterly": 300.09  
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
annually	Double	Expected revenue amount that's calculated annually.	Small, 63.0	63.0
monthly	Double	Expected revenue amount that's calculated monthly.	Small, 63.0	63.0
quarterly	Double	Expected revenue amount that's calculated quarterly.	Small, 63.0	63.0

Attribute Detail

Output representation of product attribute details.

JSON example

This request shows a sample success response.

```
"attributes": [
  {
    "id": "0tjSG0000001embYAA",
    "sequence": 1, #optional
    "name": "Copay",
    "label": "Copay Currency", #optional
    "dataType": "CURRENCY",
    "keyHidden": false,
    "isActive": true
  },
  {
    "id": "0tjSG0000001eoDYAQ",
    "sequence": 1,
    "name": "Coinsurance",
    "label": "Coinsurance Percentage",
    "dataType": "PERCENTAGE",
    "keyHidden": false,
    "isActive": true
  }
]
```

Property Name	Type	Description	Filter Group and Version	Available Version
attributeDefinitionId	String	ID of the attribute definition.	Small, 63.0	63.0
attributePicklistValueId	String	ID of the attribute picklist value.	Small, 63.0	63.0
dataType	String	Data type of attribute, such as Percentage.	Small, 63.0	63.0
id	String	ID of the product.	Small, 63.0	63.0
isActive	Boolean	Indicates whether the attribute is active (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
keyHidden	Boolean	Indicates whether the attribute key is hidden (<code>true</code>) or (<code>not</code>).	Small, 63.0	63.0
label	String	Label of the attribute.	Small, 63.0	63.0
name	String	Name of the attribute.	Small, 63.0	63.0
picklistValues	Attribute Picklist Value[]	List of picklist values.	Small, 63.0	63.0
sequence	Integer	Sequence of attribute.	Small, 63.0	63.0
value	String	Value of attribute.	Small, 63.0	63.0

Attribute Picklist Value

Output representation of the picklist value details.

Property Name	Type	Description	Filter Group and Version	Available Version
attributePicklistId	String	ID of the attribute picklist.	Small, 63.0	63.0
id	String	ID of the product.	Small, 63.0	63.0
name	String	Name of the attribute.	Small, 63.0	63.0
value	String	Value of the attribute.	Small, 63.0	63.0

Benefits

Output representation details of the plan benefits for an insurance policy coverage plan.

JSON example

```
{
  "benefits": [
    {
      "benefitId": "bft1",
      "benefitSequence": 1,
      "productId": "0ltSG000001lyBFxxY",
      "productName": "Preventive Care",
      "productCompGrpId": "0y7SG0000000Q8zxxE",
      "productCompGgpName": "General plan information",
      "productCompGgpSequence": 1,
      "network": "First Network Tier",
      "attributes": [
        {
          "attributeDefinitionId": "0tjSG0000001embxxA",
          "name": "Copay",
          "value": 400,
          "sequence": 1,
          "keyHidden": false,
          "dataType": "Currency"
        },
        {
          "attributeDefinitionId": "0tjSG00000020yPxxQ",
          "name": "Deductible waiver",
          "value": "Deductible waived",
          "sequence": 2,
          "keyHidden": true,
          "dataType": "Text"
        },
        {
          "attributeDefinitionId": "0tjSG0000001YmMxxU",
          "name": "Notes",
          "value": "In network limitations apply.",
          "sequence": 2,
          "keyHidden": true,

```

```

        "dataType": "Text"
      }
    ]
  }
}

```

Properties

Name	Type	Description	Available Version
benefitId	String	ID of the Insurance Policy Coverage Benefit.	63.0
benefitSequence	String	Field sequence of the Insurance Policy Coverage Benefit.	63.0
network	String	Type of tier of the network.	63.0
productId	String	ID of the product.	63.0
productCompGgpId	String	ID of product component group.	63.0
productCompGgpName	String	Name of product component group.	63.0
productCompGgpSequence	String	Field sequence of the product component group.	63.0
productName	String	Name of the product.	63.0
attributes	Attributes on page 44	List of attributes of the plan benefits.	63.0

Cost Calculation

Output representation details of the cost calculation.

JSON example

Here's a sample success response.

```

{
  "isSuccess": true,
  "errors": [],
  "policy": {
    "id": "p111",
    "premiumAmountWithFrequency": {
      "Annually": "xxx.xx",
      "Monthly": "xxx.xxx",
      "Quarterly": "xx.xxx"
    },
    "premiumEqvAmountWithFrequency": {
      "Annually": "xx.xx",
      "Monthly": "xx.xx",
      "Quarterly": "xxx.xxx"
    },
    "feeAmountWithFrequency": {
      "Annually": "xx.xx",
      "Monthly": "xx.xx",
      "Quarterly": "xxx.xxx"
    }
  }
}

```

```
    },
    "coverages": [
      {
        "id": "pcl11",
        "premiumAmountWithFrequency": {
          "Annually": "xxx.xx",
          "Monthly": "xxx.xxx",
          "Quarterly": "xx.xxx"
        },
        "premiumEqvAmountWithFrequency": {
          "Annually": "xx.xx",
          "Monthly": "xx.xx",
          "Quarterly": "xxx.xxx"
        },
        "feeAmountWithFrequency": {
          "Annually": "xx.xx",
          "Monthly": "xx.xx",
          "Quarterly": "xxx.xxx"
        }
      },
      "ratePlans": [
        {
          "id": "rp11",
          "totalPremiumAmount": 350,
          "totalFeeAmount": "xyz",
          "totalPremiumEqvAmount": "xyz",
          "ratePlanLineItems": [
            {
              "id": "rpl1111",
              "premiumAmount": 100
            },
            {
              "id": "rpl1112",
              "premiumAmount": 150
            }
          ]
        },
        {
          "id": "rp12",
          "totalPremiumAmount": 580,
          "ratePlanLineItems": [
            {
              "id": "rpl1121",
              "premiumAmount": 80
            },
            {
              "id": "rpl1122",
              "premiumAmount": 200
            },
            {
              "id": "rpl1123",
              "premiumAmount": 300
            }
          ]
        }
      ]
    }
  ]
}
```

```

    ]
  }
}
}

```

This request shows a sample error response.

```

{
  "isSuccess":true,
  "errors" : [
    {
      "code":INVALID_INPUT,
      "message": "Specify a valid ratePlanId."
    }
  ]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
policy	Map<String, Object>	Map of policy and cost calculation details from <code>policy</code> to <code>ratePlanLineItems</code> .	Small, 63.0	63.0

Expected Revenue Coverage

Output representation details of the coverage and associated expected revenue.

JSON example

This example shows a sample success response.

```

"coverages": [
  {
    "id": "pc111",
    "amountWithFrequency": {
      "annually": 1000.3,
      "monthly": 100.03,
      "quarterly": 300.09
    },
    "ratePlans": [
      {
        "id": "rp11",
        "totalPremiumAmount": 350,
        "totalFeeAmount": 200,
        "totalPremiumEqvAmount": 100,
        "expectedRevenueAmount": 45,

```

```

    "ratePlanCommissions": [
      {
        "id": "rpc111",
        "premiumPercent": 100,
        "type": "FlatPercentOfPremium"
      }
    ]
  },
  {
    "id": "rp12",
    "totalPremiumAmount": 350,
    "totalFeeAmount": 200,
    "totalPremiumEqvAmount": 100,
    "expectedRevenueAmount": 45,
    "ratePlanCommissions": [
      {
        "id": "rpc111",
        "premiumPercent": null,
        "type": "GradedPercent",
        "amountStartRange": 0,
        "amountEndRange": 1000
      },
      {
        "id": "rpc112",
        "premiumPercent": null,
        "type": "GradedPercent",
        "amountStartRange": 1001,
        "amountEndRange": 10000
      }
    ]
  }
]
}
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
amountWithFrequency	Amount With Frequency	Expected revenue amount with frequency.	Small, 63.0	63.0
id	String	ID of the coverage associated with the policy for which the expected revenue is calculated.	Small, 63.0	63.0
ratePlans	Rate Plan	Details of the rate plans associated with the policy or through coverages.	Small, 63.0	63.0

Expected Revenue Policy

Output representation of the associated expected revenue details for a policy, such as coverages and rate plans.

JSON example

This example shows a sample success response.

```
{
  "policy": {
    "id": "p111",
    "amountWithFrequency": {
      "annually": 1000.3,
      "monthly": 100.03,
      "quarterly": 300.09
    },
    "coverages": [
      {
        "id": "pc111",
        "amountWithFrequency": {
          "annually": 1000.3,
          "monthly": 100.03,
          "quarterly": 300.09
        },
        "ratePlans": [
          {
            "id": "rp11",
            "totalPremiumAmount": 350,
            "totalFeeAmount": 200,
            "totalPremiumEqvAmount": 100,
            "expectedRevenueAmount": 45,
            "ratePlanCommissions": [
              {
                "id": "rpc111",
                "premiumPercent": 100,
                "type": "FlatPercentOfPremium"
              }
            ]
          }
        ],
      },
      {
        "id": "rp12",
        "totalPremiumAmount": 350,
        "totalFeeAmount": 200,
        "totalPremiumEqvAmount": 100,
        "expectedRevenueAmount": 45,
        "ratePlanCommissions": [
          {
            "id": "rpc111",
            "premiumPercent": null,
            "type": "GradedPercent",
            "amountStartRange": 0,
            "amountEndRange": 1000
          },
          {
            "id": "rpc112",
            "premiumPercent": null,
            "type": "GradedPercent",
            "amountStartRange": 1001,
            "amountEndRange": 10000
          }
        ]
      }
    ]
  }
}
```

```

    },
  ],
},
],
},
],
},
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
amountWithFrequency	Amount With Frequency[]	Expected revenue amount with frequency.	Small, 63.0	63.0
coverages	Expected Rev Coverage[]	Details of policy coverages, such as associated rate plans.	Small, 63.0	63.0
id	String	ID of the policy for which expected revenue is calculated.	Small, 63.0	63.0
ratePlans	Rate Plan[]	Details of rate plans.	Small, 63.0	63.0

Expected Revenue

Output representation of the details of the expected revenue.

JSON example

This example shows a sample success response.

```

{
  "policy": {
    "id": "p111",
    "amountWithFrequency": {
      "annually": 1000.3,
      "monthly": 100.03,
      "quarterly": 300.09
    },
    "coverages": [
      {
        "id": "pc111",
        "amountWithFrequency": {
          "annually": 1000.3,
          "monthly": 100.03,
          "quarterly": 300.09
        },
        "ratePlans": [
          {
            "id": "rp11",
            "totalPremiumAmount": 350,
            "totalFeeAmount": 200,
            "totalPremiumEqvAmount": 100,
            "expectedRevenueAmount": 45,
            "ratePlanCommissions": [

```



```

    {
      "id": "rpc111",
      "premiumPercent": 100,
      "type": "FlatPercentOfPremium"
    }
  ],
},
{
  "id": "rp12",
  "totalPremiumAmount": 350,
  "totalFeeAmount": 200,
  "totalPremiumEqvAmount": 100,
  "expectedRevenueAmount": 45,
  "ratePlanCommissions": [
    {
      "id": "rpc111",
      "premiumPercent": null,
      "type": "GradedPercent",
      "amountStartRange": 0,
      "amountEndRange": 1000
    },
    {
      "id": "rpc112",
      "premiumPercent": null,
      "type": "GradedPercent",
      "amountStartRange": 1001,
      "amountEndRange": 10000
    }
  ],
}
]
}
]
}
}
}
```

This example shows a sample error response.

```

{
  "isSuccess":false,
  "errors" : [
    {
      "code":INVALID_INPUT,
      "message": "Specify a valid ratePlanId."
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response[]	List of errors encountered during the processing of the API request.	Small, 63.0	63.0

Property Name	Type	Description	Filter Group and Version	Available Version
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
policy	Expected Revenue Policy[]	Map of the policy and expected revenue calculation details.	Small, 63.0	63.0

Commission Processing Details

Output representation details of the commission processing.

JSON example

This example shows a sample success response.

```
{
  "status": "Success",
  "message": "The processing has started for the commission statement CS-00000001."
}
```

This example shows a sample error response.

```
{
  "status": "Fail",
  "message": "Processing failed for the commission statement The processing has started for the commission statement CS-00000001 ."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Status message of the API execution.	Small, 63.0	63.0
status	String	Status of the API execution. Valid values are: • Success • Fail	Small, 63.0	63.0

Insurance Billing Schedules Response

Details of the billing schedule creation results.

JSON example

```
{
  "errors": [],
  "message": "Billing schedules created successfully."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response	List of errors that are encountered during the billing process.	Small, 66.0	66.0
message	String	Text message of the API request.	Small, 66.0	66.0

Insurance Policy Bulk

Output representation details of bulk renew policy and bulk cancel policy requests.

JSON example

This example shows a sample success response.

```
{
  "errors": [{ "code": "", "message": "" }],
  "status": "Success",
  "message": "policy renewed successfully."
}
```

This example shows a sample error response.

```
{
  "errors": [{ "code": "", "message": "" }],
  "status": "Fail",
  "message": "error message."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
message	String	Status message of the API execution.	Small, 63.0	63.0
status	String	Status of the API execution. Valid values are: • Success • Fail	Small, 63.0	63.0

Insurance Policy Process

Output representation details of the renewed policy.

JSON example

This example shows a sample success response.

```
{
  "errors": [{ "code": "", "message": "" }],
  "status": "Success",
  "policyId": "testId",
}
```

```
{
  "message": "policy renewed successfully."
}
```

This example shows a sample error response.

```
{
  "errors": [{ "code": "", "message": "" }],
  "status": "Fail",
  "policyId": "",
  "message": "error message."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
policyId	String	ID of the renewed policy.	Small, 63.0	63.0

Plan Benefits Product Model Details

Output representation of the product model details and associated product components.

JSON example

This example shows a sample success response.

```
{
  "isSuccess": true,
  "errors": [],
  "productId": "01tSG0000013itZYAQ",
  "benefits": [
    {
      "id": "01tSG0000011y9dYAA",
      "name": "Annual deductible/Individual",
      "componentGroupId": "0y7SG0000000Q8zYAE",
      "componentGroupName": "General plan information",
      "componentGroupSequence": 1, #optional
      "sequence": 1, #optional
      "attributes": [
        {
          "id": "0tjSG0000001embYAA",
          "sequence": 1, #optional
          "name": "Copay",
          "label": "Copay Currency", #optional
          "dataType": "CURRENCY",
          "keyHidden": false,
          "isActive": true
        },
        {
          "id": "0tjSG0000001eoDYAQ",
          "sequence": 1,
          "name": "Coinsurance",
          "label": "Coinsurance Percentage",
          "dataType": "PERCENTAGE",
          "keyHidden": false,
          "isActive": true
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

This request shows a sample error response.

```

{
  "isSuccess":false,
  "errors" : [
    {
      "code":INVALID_INPUT,
      "message": "Specify a valid coverageId."
    }
  ]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
benefits	Product[]	List of the product component groups and associated attributes.	Small, 63.0	63.0
errors	Error Response[]	List of errors during the processing of the API request.	Small, 63.0	63.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
productId	String	ID of the product for which the benefits are fetched.	Small, 63.0	63.0

Rate Plan Commission

Output representation details of the rate plan commission.

JSON example

This example shows a sample success response.

```

[
  {
    "id": "rpc111",
    "premiumPercent": null,
    "type": "GradedPercent",
    "amountStartRange": 0,
    "amountEndRange": 1000
  },
  {
    "id": "rpc112",
    "premiumPercent": null,
    "type": "GradedPercent",
    "amountStartRange": 1001,

```

```

    "amountEndRange": 10000
  },
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
amountEndRange	Double	End range of amount for graded percent of premium.	Small, 63.0	63.0
amountStartRange	Double	Start range of amount for graded percent of premium.	Small, 63.0	63.0
fee	Double	Commission fee when flat fee or capitated fee is selected.	Small, 63.0	63.0
feePercent	Double	Percentage of fee earned as commission when flat percentage of fee is selected.	Small, 63.0	63.0
id	String	ID of the rate plan commission.	Small, 63.0	63.0
numberEndRange	Integer	End range of the grade volume, which represents the employee count.	Small, 63.0	63.0
numberStartRange	Integer	Start range of the grade volume, which represents the employee count.	Small, 63.0	63.0
premiumPercent	Double	Percentage of premium earned as commission when graded percent of premium or flat percent of premium is selected.	Small, 63.0	63.0
type	String	Type of commission structure. Valid values are: • Flat Percent of Premium • Flat Fee • Graded Percent of Premium • Graded Volume • Capitated Fee • Flat percent of Fee	Small, 63.0	63.0

Rate Plan

Output representation of the details of the rate plans associated directly with the policy or through coverages.

JSON example

This example shows a sample success response.

```

"ratePlans": [
  {
    "id": "rp11",

```

```

    "totalPremiumAmount": 350,
    "totalFeeAmount": 200,
    "totalPremiumEqvAmount": 100,
    "expectedRevenueAmount": 45,
    "ratePlanCommissions": [
      {
        "id": "rpc111",
        "premiumPercent": 100,
        "type": "FlatPercentOfPremium"
      }
    ]
  },
  {
    "id": "rp12",
    "totalPremiumAmount": 350,
    "totalFeeAmount": 200,
    "totalPremiumEqvAmount": 100,
    "expectedRevenueAmount": 45,
    "ratePlanCommissions": [
      {
        "id": "rpc111",
        "premiumPercent": null,
        "type": "GradedPercent",
        "amountStartRange": 0,
        "amountEndRange": 1000
      },
      {
        "id": "rpc112",
        "premiumPercent": null,
        "type": "GradedPercent",
        "amountStartRange": 1001,
        "amountEndRange": 10000
      }
    ]
  }
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
expectedRevenueAmount	Double	Expected revenue amount for the specified policy.	Small, 63.0	63.0
id	String	ID of the rate plan.	Small, 63.0	63.0
ratePlanCommissions	Rate Plan Commission[]	Details of the rate plan commissions.	Small, 63.0	63.0
totalFeeAmount	Double	Total fee amount that's calculated for rate plan and then rolled up to policy.	Small, 63.0	63.0
totalPremiumAmount	Double	Total premium amount that's calculated for rate plan and then rolled up to policy.	Small, 63.0	63.0

Property Name	Type	Description	Filter Group and Version	Available Version
totalPremium EqvAmount	Double	Total premium equivalent amount that's calculated for rate plan and then rolled up to policy.	Small, 63.0	63.0

Product

Output representation of product details.

JSON example

This example shows a sample success response.

```

    "benefits": [
      {
        "id": "01tSG000001ly9dYAA",
        "name": "Annual deductible/Individual",
        "componentGroupId": "0y7SG0000000Q8zYAE",
        "componentGroupName": "General plan information",
        "componentGroupSequence": 1, #optional
        "sequence": 1, #optional
        "attributes": [
          {
            "id": "0tjSG0000001embYAA",
            "sequence": 1, #optional
            "name": "Copay",
            "label": "Copay Currency", #optional
            "dataType": "CURRENCY",
            "keyHidden": false,
            "isActive": true
          },
          {
            "id": "0tjSG0000001eoDYAQ",
            "sequence": 1,
            "name": "Coinsurance",
            "label": "Coinsurance Percentage",
            "dataType": "PERCENTAGE",
            "keyHidden": false,
            "isActive": true
          }
        ]
      }
    ]
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
attributes	Attribute Detail[]	List of attributes.	Small, 63.0	63.0
componentGroupId	String	ID of the component group.	Small, 63.0	63.0

Property Name	Type	Description	Filter Group and Version	Available Version
componentGroupName	String	Name of the component group.	Small, 63.0	63.0
componentGroupSequence	Integer	Sequence of the component group.	Small, 63.0	63.0
id	String	ID of the product.	Small, 63.0	63.0
name	String	Name of the product.	Small, 63.0	63.0
sequence	Integer	Sequence of the product.	Small, 63.0	63.0

Plan Benefits Attribute

Represents details of the plan benefits attributes for an Insurance Policy Coverage.

JSON example

```
{
  "attributes": [
    {
      "attributeDefinitionId": "0tjSG0000001embxxA",
      "name": "Copay",
      "value": 400,
      "sequence": 1,
      "keyHidden": false,
      "dataType": "Currency"
    },
    {
      "attributeDefinitionId": "0tjSG00000020yPxxQ",
      "name": "Deductible waiver",
      "value": "Deductible waived",
      "sequence": 2,
      "keyHidden": true,
      "dataType": "Text"
    }
  ]
}
```

Properties

Name	Type	Description	Available Version
attributeDefinitionId	String	ID of the Attribute Definition.	63.0
name	String	Name of the attribute.	63.0
value	String	Value of the attribute.	63.0
sequence	Integer	Sequence field of the product classification attribute or product attribute definition.	63.0
keyHidden	Boolean	IsHidden field of the Product classification attribute or Product attribute definition.	63.0

Name	Type	Description	Available Version
dataType	String	Data type of the Attribute Definition.	63.0

Plan Benefits Attributes

Output representation details of the benefits attributes for an insurance policy coverage plan.

JSON example

```
{
  "attributes": [
    {
      "attributeDefinitionId": "0tjSG0000001embxxA",
      "name": "Copay",
      "value": 400,
      "sequence": 1,
      "keyHidden": false,
      "dataType": "Currency"
    },
    {
      "attributeDefinitionId": "0tjSG00000020yPxxQ",
      "name": "Deductible waiver",
      "value": "Deductible waived",
      "sequence": 2,
      "keyHidden": true,
      "dataType": "Text"
    },
    {
      "attributeDefinitionId": "0tjSG0000001YmMxxU",
      "name": "Notes",
      "value": "In network limitations apply.",
      "sequence": 2,
      "keyHidden": true,
      "dataType": "Text"
    }
  ]
}
```

Properties

Name	Type	Description	Available Version
attributeDefinitionId	String	Indicates whether the request is successful (<code>true</code>) or not (<code>false</code>).	63.0
dataType	String	Datatype of the Attribute Definition.	63.0
keyHidden	String	IsHidden field of the Product classification attribute or Product attribute definition.	63.0
name	String	Type of tier of the network.	63.0
sequence	String	Sequence field of the Product classification attribute or Product attribute definition.	63.0

Name	Type	Description	Available Version
value	String	ID of the product.	63.0

Plan Benefit Networks

Output representation details of the insurance policy coverage networks.

JSON example

```
{
  "networks": [
    {
      "id": "n1",
      "network": "FIRST_NETWORK_TIER",
      "name": "Preferred Network",
      "sequence": 1
    },
    {
      "id": "n2",
      "network": "SECOND_NETWORK_TIER",
      "name": "In Network",
      "sequence": 2
    }
  ]
}
```

Properties

Name	Type	Description	Available Version
id	String	ID of the insurance policy coverage network.	63.0
name	String	Name of the insurance policy coverage network.	63.0
network	String	Insurance policy coverage network tier type.	63.0
sequence	String	Sequence field of the insurance policy coverage network.	63.0

Plan Benefits

Output representation details of the create and update plan benefits request.

JSON example

```
{
  "isSuccess": true,
  "errors": []
}
```

Properties

Name	Type	Description	Available Version
isSuccess	Boolean	Indicates whether the request is successful (true) or not (false).	63.0
errors	String	The details of any errors.	63.0

Plan Benefits Details

Output representation details of the insurance policy benefits.

JSON example

```
{
  "isSuccess": true,
  "errors": [],
  "networks": [
    {
      "id": "n1",
      "network": "FIRST_NETWORK_TIER",
      "name": "Preferred Network",
      "sequence": 1
    },
    {
      "id": "n2",
      "network": "SECOND_NETWORK_TIER",
      "name": "In Network",
      "sequence": 2
    }
  ],
  "benefits": [
    {
      "benefitId": "bft1",
      "benefitSequence": 1,
      "productId": "01tSG000001lyBFYAY",
      "productName": "Preventive Care",
      "productCompGrpId": "0y7SG0000000Q8zYAE",
      "productCompGgpName": "General plan information",
      "productCompGgpSequence": 1,
      "network": "First Network Tier",
      "attributes": [
        {
          "attributeDefinitionId": "0tjSG0000001embYAA",
          "name": "Copay",
          "value": 400,
          "sequence": 1,
          "keyHidden": false,
          "dataType": "Currency"
        },
        {
          "attributeDefinitionId": "0tjSG000000020yPYAQ",
          "name": "Deductible waiver",

```

```

        "value": "Deductible waived",
        "sequence": 2,
        "keyHidden": true,
        "dataType": "Text"
      },
      {
        "attributeDefinitionId": "0tjSG0000001YmMYAU",
        "name": "Notes",
        "value": "In network limitations apply.",
        "sequence": 2,
        "keyHidden": true,
        "dataType": "Text"
      }
    ],
    "combinedValue": "Copay $400 | Deductible waived | In network limitations apply."
  }
]
}

```

Properties

Name	Type	Description	Available Version
attributes	Benefit Attributes on page 43	List of attributes.	63.0
benefits	Benefits on page 28	Details of the plan benefit.	63.0
combinedValue	String	Combined value of the Insurance Policy Coverage Benefit Attribute for a benefit.	63.0
errors	List	List of errors.	63.0
isSuccess	Boolean	Boolean to determine whether it was success or failure.	63.0
network	String	The tier type of the network.	63.0
networks	Networks on page 45	Insurance Policy Coverage Networks for creating plan benefits.	63.0
productCompGgpId	String	ID of product component group	63.0
productId	String	ID of the product.	63.0
productName	String	Name of the product.	63.0

Product

Output representation of product details.

JSON example

This example shows a sample success response.

```

    "benefits": [
      {
        "id": "01tSG0000011y9dYAA",
        "name": "Annual deductible/Individual",
        "componentGroupId": "0y7SG0000000Q8zYAE",
        "componentGroupName": "General plan information",
        "componentGroupSequence": 1, #optional
        "sequence": 1, #optional
        "attributes": [
          {
            "id": "0tjSG0000001embYAA",
            "sequence": 1, #optional
            "name": "Copay",
            "label": "Copay Currency", #optional
            "dataType": "CURRENCY",
            "keyHidden": false,
            "isActive": true
          },
          {
            "id": "0tjSG0000001eoDYAQ",
            "sequence": 1,
            "name": "Coinsurance",
            "label": "Coinsurance Percentage",
            "dataType": "PERCENTAGE",
            "keyHidden": false,
            "isActive": true
          }
        ]
      }
    ]
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
attributes	Attribute Detail[]	List of attributes.	Small, 63.0	63.0
componentGroupId	String	ID of the component group.	Small, 63.0	63.0
componentGroupName	String	Name of the component group.	Small, 63.0	63.0
componentGroupSequence	Integer	Sequence of the component group.	Small, 63.0	63.0
id	String	ID of the product.	Small, 63.0	63.0
name	String	Name of the product.	Small, 63.0	63.0
sequence	Integer	Sequence of the product.	Small, 63.0	63.0

Insurance Brokerage Standard Invocable Actions

Learn more about the standard invocable actions available with Insurance Brokerage.

[Compute Producer Splits Action](#)

Compute the producer splits for the producers associated with an Insurance Policy, for a Commission Statement Line Item.

[Create Producer Commissions Action](#)

Create records for the commissions that producers receive for the insurance policy associated with the specified commission statement line item, and update the commission statement line item record status.

[Find Insurance Policy Action](#)

Get the insurance policy associated with a commission statement line item that matches the specified criteria, and update the status of the commission statement line item record.

Compute Producer Splits Action

Compute the producer splits for the producers associated with an Insurance Policy, for a Commission Statement Line Item.

This action is available in API version 63.0 and later.

Special Access Rules

The Compute Producer Splits action is available in Enterprise, Unlimited, and Developer Editions where where Financial Service Cloud and Insurance Brokerage are enabled. To use this action, you need Insurance Commission Management and Insurance Brokerage User permission sets.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/computeProducerSplits`

Formats

JSON

HTTP Methods

GET, POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
<code>commissionStatementLineItemId</code>	Type String Description ID of the commission statement line item record.

Outputs

Output	Details
producerCommissions	Type sObject Description Collection of producer commission records that are computed.

Example

POST

This sample request is for the Compute Producer Splits action.

```
{
  "inputs": [
    {
      "commissionStatementLineItemId": "1Atxx0000004DeaCAE"
    }
  ]
}
```

This is the sample response for the Compute Producer Splits action.

```
[
  {
    "actionName": "computeProducerSplits",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "producerCommissions": [
        {
          "attributes": {
            "type": "ProducerCommission"
          },
          "CommissionStatementLineItemId": "1Atxx0000004C92CAE",
          "ProducerSplitAssignmentId": "1AIxx0000004C92GAE",
          "CommissionAmount": 4000,
          "InsurancePolicyId": "0YTxx0000000001GAA",
          "ProducerId": "0Yxxx0000000001CAA",
          "ProducerSplitArngLineItemId": "1BHxx0000004C92GAE",
          "Name": "CSLI-0000000001-Amol-Commission"
        }
      ]
    },
    "sortOrder": -1,
    "version": 1
  }
]
```


Create Producer Commissions Action

Create records for the commissions that producers receive for the insurance policy associated with the specified commission statement line item, and update the commission statement line item record status.

This action is available in API version 63.0 and later.

Special Access Rules

The Create Producer Commissions action is available in Enterprise, Unlimited, and Developer Editions where where Financial Service Cloud and Insurance Brokerage are enabled. To use this action, you need Insurance Commission Management and Insurance Brokerage User permission sets.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/createProducerCommissions

Formats

JSON

HTTP Methods

GET, POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
producerCommissions	Type sObject Description A collection of computed producer commission records to be created.
commissionStatementLineItemId	Type String Description ID of the commission statement line item record.
successStatus	Type String Description The status that the associated commission statement line item is set to when the producer commission record is created.

Input	Details
failureStatus	Type String
	Description The status that the associated commission statement line item is set to when the producer commission record isn't created.

Outputs

Output	Details
producerCommissionIds	Type String
	Description A collection of IDs of the producer commission records that are created.

Example

POST

This sample request is for the Create Producer Commissions action.

```
{
  "inputs": [
    {
      "commissionStatementLineItemId": "1Atxx0000004DeaCAE",
      "producerCommissions": [
        {
          "CommissionAmount": 3000,
          "InsurancePolicyId": "0YTxx00000002BVGAY",
          "ProducerId": "0Yxxx00000001IfCAI",
          "ProducerSplitArngLineItemId" : "1BHxx0000004CXEGA2",
          "CommissionStatementLineItemId": "1Atxx0000004DeaCAE",
          "Name": "Sanjeev"
        },
        {
          "InsurancePolicyId": "0YTxx00000002BVGAY",
          "ProducerId": "0Yxxx00000001KHCAI",
          "ProducerSplitArngLineItemId" : "1BHxx0000004CXEGA2",
          "CommissionStatementLineItemId": "1Atxx0000004DeaCAE",
          "Name": "Deepak"
        }
      ]
    }
  ]
}
```

This is the sample response for the Create Producer Commissions action.

```
[
  {
    "actionName": "createProducerCommissions",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "producerCommissionIds": [
        "0kAxx0000000001"
      ]
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Find Insurance Policy Action

Get the insurance policy associated with a commission statement line item that matches the specified criteria, and update the status of the commission statement line item record.

This action is available in API version 63.0 and later.

Special Access Rules

The Find Insurance Policy action is available in Enterprise, Unlimited, and Developer Editions where where Financial Service Cloud and Insurance Brokerage are enabled. To use this action, you need Insurance Commission Management and Insurance Brokerage User permission sets.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/findInsurancePolicy

Formats

JSON

HTTP Methods

GET, POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
commissionStatementLineItemId	Type String

Input	Details
	Description ID of the commission statement line item record.
matchingCriteriaPattern	Type String Description String that identifies the corresponding insurance policy.
failureStatus	Type String Description Indicates the status of the commission statement line item when the item's processing fails.

Outputs

Output	Details
commissionStatementLineItem	Type sObject Description Commission statement line item record populated with the insurance policy ID.

Example

POST

This sample request is for the Find Insurance Policy action.

```
{
  "inputs": [
    {
      "commissionStatementLineItemId": "1Atxx0000004DeaCAE",
      "matchingCriteriaPattern": "InsurancePolicyNumber:PolicyName AND
InsurancePolicyEffectiveDate:EffectiveFromDate AND LineOfCoverage:LineOfCoverage AND
ClientAccountName:NameInsured.Name"
    }
  ]
}
```

This is the sample response for the Find Insurance Policy action.

```
{
  "actionName": "findInsurancePolicy",
  "errors": null,
  "invocationId": null,
}
```

```

    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "commissionStatementLineItem": {
        "attributes": {
          "type": "CommissionStatementLineItem",
          "url":
"/services/data/v64.0/subjects/CommissionStatementLineItem/1Atxx0000004C92CAE"
        },
        "InsurancePolicyEffectiveDate": "2025-02-01",
        "LineOfCoverage": "Medical",
        "ReferenceRecordId": "0YTx0000000001GAA",
        "InsurancePolicyNumber": "test",
        "Id": "1Atxx0000004C92CAE",
        "ClientAccountName": "test"
      }
    },
    "sortOrder": -1,
    "version": 1
  }
}

```

Insurance Brokerage Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Brokerage](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Brokerage

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Brokerage exposes additional `actionType` values for the FlowActionCall Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	InvokableActionType (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Brokerage include: <code>computeProducerSplits</code>—Compute the producer splits for the producers associated with an Insurance Policy, for a Commission Statement Line Item.

Field Name	Field Type	Description
		<code>createProducerCommissions</code>—Create records for the commissions that producers receive for the insurance policy associated with the specified commission statement line item, and update the commission statement line item record status. <code>findInsurancePolicy</code>—Get the insurance policy associated with a commission statement line item that matches the specified criteria, and update the status of the commission statement line item record.

Insurance Brokerage Apex Reference

Apex is a strongly typed, object-oriented programming language that developers can use to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. Insurance Brokerage Apex reference includes built-in Apex classes and interfaces, grouped by namespace.

[ConnectApi Namespace](#)

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Brokerage provides classes for accessing the same data available in the Insurance Brokerage Business APIs.

ConnectApi Namespace

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Brokerage provides classes for accessing the same data available in the Insurance Brokerage Business APIs.

For information about working with the `ConnectApi` classes, see [Connect in Apex](#).

[InsuranceBrokerageFamily Class](#)

Manage policy lifecycle, brokerage commissions, and plan benefits.

[ConnectApi Input Classes](#)

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

[ConnectApi Output Classes](#)

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

InsuranceBrokerageFamily Class

Manage policy lifecycle, brokerage commissions, and plan benefits.

Namespace

`ConnectApi`

[InsuranceBrokerageFamily Methods](#)

These methods are for `InsuranceBrokerageFamily`. All methods are static.

InsuranceBrokerageFamily Methods

These methods are for `InsuranceBrokerageFamily`. All methods are static.

[bulkCancelInsurancePolicy\(insurancePolicyBulkInput\)](#)

Cancel policies in bulk for the specified Insurance Policy IDs.

[bulkRenewInsurancePolicy\(insurancePolicyBulkInput\)](#)

Renew policies in bulk for the specified Insurance Policy IDs.

[createBillingSchedulesFromTransaction\(InsuranceTransactionBillingInput\)](#)

Create billing schedules from an insurance transaction.

[getCommissionProcessing\(commissionProcessingInput\)](#)

Process commission statements in an asynchronous manner by using the preconfigured batch job.

[renewInsurancePolicy\(insurancePolicyProcessInput, policyId\)](#)

Renew policy for the specified insurance policy ID.

bulkCancelInsurancePolicy (insurancePolicyBulkInput)

Cancel policies in bulk for the specified Insurance Policy IDs.

API Version

64.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsurancePolicyBulkOutputRepresentation
bulkCancelInsurancePolicy (ConnectApi.InsurancePolicyBulkInputRepresentation
insurancePolicyBulkInput)
```

```
ConnectApi.InsuranceBrokerageFamily, bulkCancelInsurancePolicy,
[ConnectApi.InsurancePolicyBulkInputRepresentation],
ConnectApi.InsurancePolicyBulkOutputRepresentation
```

Parameters

insurancePolicyBulkInput

Type: [ConnectApi.InsurancePolicyBulkInputRepresentation](#) on page 65

Input representation of the request to cancel policies in bulk.

Return Value

Type: [ConnectApi.InsurancePolicyBulkOutputRepresentation](#) on page 67

Usage

Use this method to cancel insurance policies. The JSON input includes a field value map of insurance policy fields, such as `CancellationEffectiveDate`, `Status`, and `CancellationReason`, and a list of insurance policy IDs. After the request is successfully sent to the message queue, policies are canceled asynchronously, and a success or failure notification is sent to the user after the completion of the request.

bulkRenewInsurancePolicy (insurancePolicyBulkInput)

Renew policies in bulk for the specified Insurance Policy IDs.

API Version

64.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsurancePolicyBulkOutputRepresentation
bulkRenewInsurancePolicy (ConnectApi.InsurancePolicyBulkRenewInputRepresentation
insurancePolicyBulkInput)
```

```
ConnectApi.InsuranceBrokerageFamily, bulkRenewInsurancePolicy,
[ConnectApi.InsurancePolicyBulkRenewInputRepresentation],
ConnectApi.InsurancePolicyBulkOutputRepresentation
```

Parameters

insurancePolicyBulkInput

Type: [ConnectApi.InsurancePolicyBulkRenewInputRepresentation](#) on page 66

Input representation of the request to renew insurance policies in bulk.

Return Value

Type: [ConnectApi.InsurancePolicyBulkOutputRepresentation](#) on page 67

Usage

Insurance Policy Bulk Renewal API takes a JSON input that contains a list of insurance policy IDs that must be renewed along with the `fieldValueMap` details. The API submits the request to a queue, creates policies, along with child records by using the context definition selected in setup. The service also overrides the `fieldValueMap` in the new policy for each policy ID passed in the request. Users receive success and failure notifications on completion.

Keep these considerations in mind when you use this method.

- The `RenewedFromPolicyId` value of the new policy is set to the current policy ID and `IsRenewedPolicy` field of the new policy is set to `true`.
- The name of the new policy is a copy of `{ExistingName}`.
- The `fieldValueMap` expects the keys to be API name and is case-sensitive.

Example

This example shows how to call the bulk renew policy API from Apex code.

```
global class ConnectDriver {

    global static void callBulkRenewConnectAPI() {
        Map<String, Object> input = new Map<String, Object>();

        TestConnectAPI callable = new TestConnectAPI();

        ConnectApi.InsurancePolicyBulkInputRepresentation policyInput = new
ConnectApi.InsurancePolicyBulkInputRepresentation();

        Map<String, Object> fields = new Map<String, Object>();
        fields.put('Name', 'Test Apex Bulk');
        policyInput.fieldValueMap = fields;

        List<String> policyIds = new List<String>();
        policyIds.add('0YTxx000000001dGAA');
        policyIds.add('0YTxx000000001dHAA');
        policyInput.insurancePolicyIds = policyIds;

        input.put('policyInput', policyInput);

        Map<String, Object> output = new Map<String, Object>();
        Map<String, Object> args = new Map<String, Object>();
        args.put('input', input);
        args.put('output', output);

        Object result = callable.call('bulkRenew', args);
        ConnectApi.InsurancePolicyBulkOutputRepresentation bulkData =
(ConnectApi.InsurancePolicyBulkOutputRepresentation) result;
        System.debug('Bulk Renew Data: ' + JSON.serializePretty(bulkData));
        if (output.containsKey('error')) {
            System.debug('Error: ' + output.get('error'));
        }
    }
}
```

createBillingSchedulesFromTransaction(InsuranceTransactionBillingInput)

Create billing schedules from an insurance transaction.

API Version

66.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceTransactionBillingOutputRepresentation
createBillingSchedulesFromTransaction (ConnectApi.InsuranceTransactionBillingInputRepresentation
InsuranceTransactionBillingInput)
```

Parameters

InsuranceTransactionBillingInput

Type: [ConnectApi.InsuranceTransactionBillingInputRepresentation](#)

Details to create billing schedules from an insurance transaction.

Return Value

Type: [ConnectApi.InsuranceTransactionBillingOutputRepresentation](#)

Example

This is a sample to call the Create Billing Schedules connect API from Apex.

Class Definition

```
public with sharing class CreateInsuranceBillingScheduleCallable implements
System.Callable{

    public Object call(String actionType, Map<String, Object> args) {

        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');

        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Retrieve the insurance policy transaction input data from the input map

        ConnectApi.InsuranceTransactionBillingInputRepresentation
insuranceTransactionInput =
        (ConnectApi.InsuranceTransactionBillingInputRepresentation) inputMap.get('insuranceTransactionInput');

        String insPolicyTransactionId = (String) inputMap.get('insPolicyTransactionId');

        System.debug('Start of connect API call to Insurance Transaction Billing');

        // Initialize the insurance transaction billing output representation
        ConnectApi.InsuranceTransactionBillingOutputRepresentation
transactionBillingOutcome;

        // Call the Connect API to send the insurance transaction to billing
        try {
            transactionBillingOutcome =
ConnectApi.InsuranceBillingFamily.createBillingSchedulesFromTransaction(insuranceTransactionInput);
```

```

        System.debug('Policy transaction send to billing successfully: ' +
transactionBillingOutcome);
        // Put the transaction billing outcome in the output map
        outputMap.put('transactionBillingOutcome', transactionBillingOutcome);

    } catch (Exception e) {
        System.debug('Error while sending transaction to billing: ' + e.getMessage());

        outputMap.put('error', e.getMessage());
    }
    // Return the transaction billing outcome
    return transactionBillingOutcome;
}
}

```

Usage Example

```

global class ConnectDriver {

    global static void callInsuranceBillingConnectAPI(String insPolicyTransactionId){

        Map<String, Object> input = new Map<String, Object>();
        CreateInsuranceBillingScheduleCallable callable = new
CreateInsuranceBillingScheduleCallable();

        // Create and populate the InsuranceTransactionBillingInputRepresentation object

        ConnectApi.InsuranceTransactionBillingInputRepresentation
insuranceTransactionInput = new
ConnectApi.InsuranceTransactionBillingInputRepresentation();

        // Populate transactionId for InsuranceTransactionBillingInputRepresentation

insuranceTransactionInput.insPolicyTransactionId = insPolicyTransactionId;

        // Add the insuranceTransactionInput to the input map
input.put('insuranceTransactionInput', insuranceTransactionInput);

        // Prepare the output map
Map<String, Object> output = new Map<String, Object>();

        // Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

        // Call the 'insuranceBillingService' action
Object result = callable.call('InsuranceBilling', args);

        // Retrieve the policy data from the result
ConnectApi.InsuranceTransactionBillingOutputRepresentation
transactionBillingOutcome = (ConnectApi.InsuranceTransactionBillingOutputRepresentation)

```

```

    result;

    // Output the policy data to the debug log
    System.debug('Insurance Transaction Billing API Response: ' +
JSON.serializePretty(transactionBillingOutcome));

    // Check for errors in the output map
    if (output.containsKey('error')) {
        System.debug('Error: ' + output.get('error'));
    }
}
}

```

getCommissionProcessing(commissionProcessingInput)

Process commission statements in an asynchronous manner by using the preconfigured batch job.

API Version

63.0

Requires Chatter

No

Signature

```

public static ConnectApi.InsuranceCommissionProcessingOutputRepresentation
getCommissionProcessing(ConnectApi.InsuranceCommissionProcessingInputRepresentation
commissionProcessingInput)

```

```

ConnectApi.InsuranceBrokerageFamily, getCommissionProcessing,
[ConnectApi.InsuranceCommissionProcessingInputRepresentation],
ConnectApi.InsuranceCommissionProcessingOutputRepresentation

```

Parameters

commissionProcessingInput

Type: [ConnectApi.InsuranceCommissionProcessingInputRepresentation](#) on page 65

Input representation details of the insurance commission processing API.

Return Value

Type: [ConnectApi.InsuranceCommissionProcessingOutputRepresentation](#) on page 67

Usage

Commission Statement Processing API uses a JSON input that contains a list of `commissionStatementIds`. The service processes the commission statement asynchronously by using a batch job. The batch job calls the processing flow to process each commission statement line item. When the processing is complete, the service updates the commission statement and sends a notification to the user.

renewInsurancePolicy (insurancePolicyProcessInput, policyId)

Renew policy for the specified insurance policy ID.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsurancePolicyProcessOutputRepresentation  
renewInsurancePolicy(ConnectApi.InsurancePolicyRenewInputRepresentation  
insurancePolicyProcessInput, String policyId)
```

```
ConnectApi.InsuranceBrokerageFamily, renewInsurancePolicy,  
[ConnectApi.InsurancePolicyRenewInputRepresentation, String],  
ConnectApi.InsurancePolicyProcessOutputRepresentation
```

Parameters

insurancePolicyProcessInput

Type: [ConnectApi.InsurancePolicyRenewInputRepresentation](#) on page 66

Input representation details of the request to renew an insurance policy.

policyId

Type: String

ID of the insurance policy that's to be renewed.

Return Value

Type: [ConnectApi.InsurancePolicyProcessOutputRepresentation](#) on page 67

Usage

The Insurance Policy Renewal API takes a JSON input that contains the insurance policy ID that must be renewed along with the `fieldValueMap` details. The API submits the request to a queue, creates a policy, along with child records by using the context definition selected in setup. The service also overrides the `fieldValueMap` in the new policy for the policy ID passed in the request. Users receive success and failure notifications on request completion.

Keep these considerations in mind when you use this method.

- The `RenewedFromPolicyId` value of the new policy is set to the current policy ID and `IsRenewedPolicy` field of the new policy is set to `true`.
- The `fieldValueMap` expects the keys to be API name and is case-sensitive.

Example

This example shows how to call the renew policy API from Apex code.

```
global class ConnectDriver {

    global static void callRenewConnectAPI() {

        Map<String, Object> input = new Map<String, Object>();
        input.put('policyId', '0YTx00000000ldGAA');

        TestConnectAPI callable = new TestConnectAPI();

        // Create and populate the IssuePolicyInputRepresentation object
        ConnectApi.InsurancePolicyRenewInputRepresentation policyInput = new
ConnectApi.InsurancePolicyRenewInputRepresentation();

        // Populate fields for InsurancePolicyInputRepresentation
        Map<String, Object> fields = new Map<String, Object>();
        fields.put('Name', 'Test Apex');

        policyInput.fieldValueMap = fields;

        // Add the policyInput to the input map
        input.put('policyInput', policyInput);

        // Prepare the output map
        Map<String, Object> output = new Map<String, Object>();

        // Prepare the args map
        Map<String, Object> args = new Map<String, Object>();
        args.put('input', input);
        args.put('output', output);

        // Call the 'issueInsurancePolicy' action
        Object result = callable.call('renew', args);
        // Retrieve the policy data from the result
        ConnectApi.InsurancePolicyProcessOutputRepresentation policyData =
(ConnectApi.InsurancePolicyProcessOutputRepresentation) result;
        // Output the policy data to the debug log
        System.debug('Policy Data: ' + JSON.serializePretty(policyData));
        // Check for errors in the output map
        if (output.containsKey('error')) {
            System.debug('Error: ' + output.get('error'));
        }
    }
}
```

ConnectApi Input Classes

Some ConnectApi methods take arguments that are instances of ConnectApi input classes.

Input classes are concrete unless marked abstract in this documentation. Concrete input classes have public constructors that have no parameters.

Some methods have parameters that are typed with an abstract class. You must pass in an instance of a concrete child class for these parameters.

Most input class properties can be set. Read-only properties are noted in this documentation.

Insurance Brokerage includes these Apex input classes.

[ConnectApi.InsuranceCommissionProcessingInputRepresentation](#)

Input representation details of the insurance commission processing resource. The input includes the `commissionStatementIds` property required to process commission data.

[ConnectApi.InsurancePolicyBulkInputRepresentation](#)

Input representation details of the cancel policies in bulk resource. The input includes the list of insurance policy IDs to be canceled and the details of the associated fields of the insurance policies.

[ConnectApi.InsurancePolicyBulkRenewInputRepresentation](#)

Input representation details of the renew insurance policies in bulk resource. The input includes the list of insurance policy IDs to be renewed and the details of the associated fields of the insurance policies.

[ConnectApi.InsurancePolicyRenewInputRepresentation](#)

Input representation details of the resource to renew an insurance policy.

[ConnectApi.InsuranceTransactionBillingInputRepresentation](#)

Input representation of the details to create billing schedules from an insurance transaction.

ConnectApi.InsuranceCommissionProcessingInputRepresentation

Input representation details of the insurance commission processing resource. The input includes the `commissionStatementIds` property required to process commission data.

Property	Type	Description	Required or Optional	Available Version
<code>commissionStatementIds</code>	Object	List of commission statement IDs to be processed.	Required	63.0

ConnectApi.InsurancePolicyBulkInputRepresentation

Input representation details of the cancel policies in bulk resource. The input includes the list of insurance policy IDs to be canceled and the details of the associated fields of the insurance policies.

Property	Type	Description	Required or Optional	Available Version
<code>fieldValueMap</code>	Object	JSON object that contains the name-value pairs that store the details of the insurance policy.	Required	64.0
<code>insurancePolicyIds</code>	Object	List of Insurance policy IDs.	Required	64.0

ConnectApi.InsurancePolicyBulkRenewInputRepresentation

Input representation details of the renew insurance policies in bulk resource. The input includes the list of insurance policy IDs to be renewed and the details of the associated fields of the insurance policies.

Property	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains the name-value pairs that store the details of the insurance policy.	Required	64.0
insurancePolicyIds	Object	List of insurance policy IDs.	Required	64.0

ConnectApi.InsurancePolicyRenewInputRepresentation

Input representation details of the resource to renew an insurance policy.

Property	Type	Description	Required or Optional	Available Version
fieldValueMap	Object	JSON object that contains name-value pairs that store the details of the insurance policy.	Required	63.0

ConnectApi.InsuranceTransactionBillingInputRepresentation

Input representation of the details to create billing schedules from an insurance transaction.

Property	Type	Description	Required or Optional	Available Version
insuranceTransactionId	String	ID of the insurance policy transaction record.	Required	66.0

ConnectApi Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

All properties are read-only, except for instances of output classes created within the test code.

All output classes are concrete unless marked abstract in this documentation.

All concrete output classes have no-argument constructors that you can invoke only from the test code. See [Testing ConnectApi Code](#).

Insurance Brokerage includes these Apex output classes.

[ConnectApi.InsuranceCommissionProcessingOutputRepresentation](#)

Output representation details of the commission processing resource.

[ConnectApi.InsuranceOutputRepresentation](#)

Output representation details of the bulk renew policy and bulk cancel policy requests.

[ConnectApi.InsurancePolicyBulkOutputRepresentation](#)

Output representation details of the bulk cancel policy resource.

[ConnectApi.InsurancePolicyProcessOutputRepresentation](#)

Output representation details of the insurance policy renewal resource.

[ConnectApi.InsuranceTransactionBillingOutputRepresentation](#)

Output representation of the result of creating billing schedules from an insurance transaction.

ConnectApi.InsuranceCommissionProcessingOutputRepresentation

Output representation details of the commission processing resource.

Property Name	Type	Description	Available Version
message	String	Status message of the API execution.	63.0
status	String	Status of the API execution. Valid values are: • Success • Fail	63.0

ConnectApi.InsuranceOutputRepresentation

Output representation details of the bulk renew policy and bulk cancel policy requests.

This class is abstract.

Superclass of [ConnectApi.InsurancePolicyBulkOutputRepresentation](#) .

Property Name	Type	Description	Available Version
errors	~~ConnectApi.InsurancePolicyBulkOutputRepresentation~~	List of the errors encountered during the processing of the API request.	63.0
message	String	Status message of the API execution.	63.0
status	String	Status of the API execution. Valid values are: • Success • Fail	63.0

ConnectApi.InsurancePolicyBulkOutputRepresentation

Output representation details of the bulk cancel policy resource.

Subclass of [ConnectApi.InsuranceOutputRepresentation](#).

No additional properties.

ConnectApi.InsurancePolicyProcessOutputRepresentation

Output representation details of the insurance policy renewal resource.

Property Name	Type	Description	Available Version
policyId	String	ID of the insurance policy to be returned.	63.0

ConnectApi.InsuranceTransactionBillingOutputRepresentation

Output representation of the result of creating billing schedules from an insurance transaction.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceTransactionBillingOutputRepresentation	List of errors that are encountered during the API request.	65.0
message	String	Text message of the API request.	65.0

CHAPTER 3 Insurance Claims Management

In this chapter ...

- [Insurance Claims Business APIs](#)
- [Insurance Claims Standard Invocable Actions](#)
- [Insurance Claims Apex Reference](#)
- [Insurance Claims Metadata API Types](#)
- [Insurance Claims Developer Journey](#)

Use the Insurance Claims developer resources to set up claim products and create dynamic first notice of loss intake flows for your users. Give claim adjusters a cohesive view of claims and financial data so they can efficiently assess and settle claims. Establish financial controls and automatically route requests for financial approval to the appropriate supervisors and managers.

[View supported editions.](#)

Insurance Claims Business APIs

Use the Insurance Claims APIs to manage claims lifecycle.

Resources

Learn more about the available Insurance Claims API.

Requests

Learn more about the available requests of insurance claim APIs.

Responses

Learn more about the available response bodies of insurance claim APIs.

Resources

Learn more about the available Insurance Claims API.

[Calculate or Recalculate Adjustments \(POST, PATCH\)](#)

Calculate adjustments for a new loss item (POST) or recalculate adjustments for an existing payment detail record (PATCH) under a claim coverage.

[Cancel Claim Coverage Payment Detail \(POST\)](#)

Cancel payment for a claim coverage payment detail record.

[Claim \(POST\)](#)

Create a claim.

[Claim \(GET, PATCH\)](#)

Get or update a claim record and its associated claim items and participants.

[Claim Coverage \(POST\)](#)

Create a claim coverage record for the specified claim.

[Claim Coverage \(PATCH\)](#)

Update a specific claim coverage record.

[Claim Coverage Payment Detail \(POST\)](#)

Create a claim coverage payment detail record. This also generates the related records for the Insurance Policy Limit Tracking and Claim Coverage Payment Adjustment records if applicable.

[Claim Coverage Payment Detail \(PATCH\)](#)

Modify an existing claim coverage payment detail record for both loss and expense types.

[Delete Claim Coverage Payment Detail \(POST\)](#)

Remove unpaid claim coverage payment details by updating the status to Deleted. The API reverses the insurance policy limit tracking data to roll back consumption for the type Loss.

[Field Set Fields \(GET\)](#)

Get the fields of a field set by the field set ID and object name.

[Policy Limits \(POST\)](#)

Create insurance policy limit records for the specified policy based on its defined product attributes and coverages.

[Insurance Policy Limit Current Standings \(POST\)](#)

Get the calculated current standings (consumption status) for all applicable insurance policy limits. This API retrieves standings based on the policy, claim, claim coverage as well as policy coverage, insurance policy participant, and insurance policy asset.

[Pay Claim Coverage Payment Detail \(POST\)](#)

Process the payment for a claim coverage payment detail record.

[Pay Ex Gratia Claim Coverage Payment Detail \(POST\)](#)

Process a discretionary ex gratia payment for a claim coverage payment detail, enabling payment even when coverage or policy limits are exceeded.

[Process Policy Limits \(POST\)](#)

Process the policy limits for a loss item in a claim by using the current standings of the insurance policy limits.

[Verify Policy Coverage \(POST\)](#)

Verify the applicability of a policy coverage for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

Calculate or Recalculate Adjustments (POST, PATCH)

Calculate adjustments for a new loss item (POST) or recalculate adjustments for an existing payment detail record (PATCH) under a claim coverage.

This API uses the default Expression Set Executable Template to perform the calculation and applies the standard policy rules, such as deductibles, co-pays, co-insurance, and limits to an initial claimed amount. You can either extend the executable template to create a custom Expression Set or create an Integration Procedure to include custom calculations.

The API uses the `claimedAmount`, `claimId`, and `claimCoverageId` as input, and returns the final `adjustedAmount` along with the generated data for Claim Coverage Payment Adjustments (CCPAs) and Insurance Policy Limit Tracking (IPLT) records. This API only returns the data; it doesn't save them in database.

Resource

```
/connect/insurance/claims/claimId/coverages/coverageId/calculate-adjustments
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/claims/01c0c000000001c7AA/coverages/01P0c000000002f7AA/calculate-adjustments
```

Available version

65.0

PATCH is available in version 66.0 and later.

HTTP methods

POST, PATCH

Path parameter for POST and PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
<code>claimId</code>	String	ID of the claim for which the payment adjustments must be calculated.	Required	65.0
<code>coverageId</code>	String	ID of the claim coverage record.	Required	65.0

Request body for POST

Calculate adjustments during the creation of a Loss item.

JSON example

```
{
  "benefitName": "Car Rental",
  "claimedAmount": 1500
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
benefitName	String	Name of the specific benefit or sub-coverage that's associated with the coverage, as defined in the product attribute mapping.	Optional	66.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Required	65.0

Response body for POST

[Calculate Adjustments](#)

Request body for PATCH

Recalculate adjustments for an existing payment detail.

JSON example

```
{
  "claimedAmount": 1200.00,
  "paymentDetailId": "012xx0000000001CAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Optional	66.0
paymentDetailId	String	ID of the claim coverage payment detail for recalculating adjustments.	Optional	66.0

Response body for PATCH

[Calculate Adjustments](#)

Cancel Claim Coverage Payment Detail (POST)

Cancel payment for a claim coverage payment detail record.

Resource

```
/connect/insurance/claims/claimId/coverages/coverageId/payment-details/paymentDetailId/cancel
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/claims/01s0000001d194/coverages/012s0000002194/payment-details/012s0000003194/cancel
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim whose coverage payment detail must be canceled.	Required	65.0
coverageId	String	ID of the coverage that applies to the claim.	Required	65.0
paymentDetailId	String	ID of the claim coverage payment detail record.	Required	65.0

Response body for POST
[Cancel Claim Coverage Payment Detail](#)
Claim (POST)

Create a claim.

Resource

```
/connect/claims
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/claims
```

Available version

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "name": "INS-Auto-Claim-27868",
  "lossDate": "2025-10-23T15:41:10.446Z",
  "claimLossDate": "2025-09-15",
```

```
"claimType": "Auto",
"incidentTypeId": "0itxx000000001dEAA",
"claimReason": "Rear-end collision",
"claimReasonType": "Accident",
"lossType": "Accident",
"insurancePolicyId": "0YTx0000000001qEAA",
"productCode": "AutoClaimRoot",
"accountId": "001xx000003DGQyAAO",
"additionalFields": {
  "PoliceReportNumber": "SF2025-00817",
  "IsTotalLoss": false,
  "ReportedDate__c": "2025-09-15"
},
"options": {
  "executeConfigurationRules": "false"
},
"attributes": {
  "list": [
    {
      "apiName": "WeatherConditions",
      "value": "Clear and Sunny"
    },
    {
      "apiName": "NumberOfVehicles",
      "value": "2"
    }
  ]
},
"items": {
  "list": [
    {
      "instanceKey": "item_vehicle_damage_01",
      "name": "Damaged Rear Bumper and Trunk",
      "productCode": "AUTO-SEDAN-2023",
      "category": "Auto Physical Damage",
      "insuredItemId": "02ixx000000415sAAA",
      "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
      "attributes": {
        "list": [
          {
            "apiName": "DamageSeverity",
            "value": "Moderate"
          }
        ]
      },
      "additionalFields": {
        "RepairEstimate": 2750,
        "Date_Damage_First_Noticed__c": "2025-09-15"
      },
      "relatedObjects": {
        "list": [
          {
            "relatedRecordId": "001xx000003DGQzAAP",
            "relatedRecordObjName": "Account"
          }
        ]
      }
    }
  ]
}
```



```

        }
      ]
    },
    "participantInstanceKey": "participant_driver_01"
  }
]
},
"participants": {
  "list": [
    {
      "instanceKey": "participant_driver_01",
      "contactId": "003xx000003DGQyAAO",
      "accountId": "001xx000003DGQyAAO",
      "insurancePolicyParticipantId": "0pPxx000000001qEAA",
      "isInjured": false,
      "roles": [
        "Driver",
        "Claimant"
      ],
      "additionalFields": {
        "StatementTaken": true
      }
    }
  ]
}
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	Object	Additional standard or custom fields that are used to create or update insurance claim.	Optional	65.0
attributes	Claim Attribute Input[]	Attributes of the claim product.	Optional	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't accept time of incident. To specify both the date and time of loss, use the <code>lossDate</code> parameter.	Optional	66.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Category that the claim reason belongs to.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
incidentTypeId	String	ID for the type of incident in the claim.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	Claim Item Input[]	Details of the items that are associated with the claim.	Optional	65.0
lossDate	Date	Date and time when the loss occurred. You must specify both the date and time of the incident; otherwise, the API returns an error. To specify only the date of loss, use the <code>claimLossDate</code> parameter.	Optional	65.0
lossType	String	Type of the loss for which the claim is initiated.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	Claim Options Input	Options for creating or updating a claim.	Optional	65.0
participants	Claim Participant Input[]	Claim participants that are associated with the claim.	Optional	65.0
productCode	String	Product code of the claim product.	Required to create a claim	65.0

Response body for POST[Claim Response](#)**Claim (GET, PATCH)**

Get or update a claim record and its associated claim items and participants.

Resource

```
/connect/claims/claimId
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/claims/0Zkxx000000001dEAA
```

Available version

65.0

HTTP methods

GET, PATCH

Path parameter for GET and PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim to get or update the details for.	Required	65.0

Response body for GET[Claim Details](#)**Request body for PATCH****JSON example**

```
{
  "name": "Claim for Auto Accident",
  "lossDate": "2025-10-23T15:41:10.446Z",
  "claimLossDate": "2025-09-15",
  "claimType": "Auto",
  "incidentTypeId": "0itxx000000001dEAA",
  "claimReason": "Rear-end collision",
  "claimReasonType": "Accident",
  "lossType": "Accident",
  "insurancePolicyId": "0YTxx0000000001qEAA",
  "productCode": "AutoClaimRoot",
  "accountId": "001xx000003DGQyAAO",
  "additionalFields": {
    "PoliceReportNumber": "SF2025-00817",
    "IsTotalLoss": false,
    "ReportedDate__c": "2025-09-15"
  },
  "options": {
    "executeConfigurationRules": "false"
  },
  "attributes": {
    "list": [
      {
        "apiName": "WeatherConditions",
        "value": "Clear and Sunny"
      },
      {
        "apiName": "NumberOfVehicles",
        "value": "2"
      }
    ]
  },
  "items": {
    "list": [
      {
        "instanceKey": "item_vehicle_damage_01",
        "name": "Damaged Rear Bumper and Trunk",
        "productCode": "AUTO-SEDAN-2023",
        "category": "Auto Physical Damage",

```

```

    "insuredItemId": "02ixx000000415sAAA",
    "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
    "attributes": {
      "list": [
        {
          "apiName": "DamageSeverity",
          "value": "Moderate"
        }
      ]
    },
    "additionalFields": {
      "RepairEstimate": 2750,
      "Date_Damage_First_Noticed__c": "2025-09-15"
    },
    "relatedObjects": {
      "list": [
        {
          "relatedRecordId": "001xx000003DGQzAAP",
          "relatedRecordObjName": "Account"
        }
      ]
    },
    "participantInstanceKey": "participant_driver_01",
    "action": "update"
  }
]
},
"participants": {
  "list": [
    {
      "instanceKey": "participant_driver_01",
      "contactId": "003xx000003DGQyAAO",
      "accountId": "001xx000003DGQyAAO",
      "insurancePolicyParticipantId": "0pPxx000000001qEAA",
      "isInjured": false,
      "roles": [
        "Driver",
        "Claimant"
      ],
      "additionalFields": {
        "StatementTaken": true
      },
      "action": "update"
    }
  ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	Object	Additional standard or custom fields that are used to create or update insurance claim.	Optional	65.0
attributes	Claim Attribute Input[]	Attributes of the claim product.	Optional	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't accept time of incident. To specify both the date and time of loss, use the <code>lossDate</code> parameter.	Optional	66.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Category that the claim reason belongs to.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0
incidentTypeId	String	ID for the type of incident in the claim.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	Claim Item Input[]	Details of the items that are associated with the claim.	Optional	65.0
lossDate	Date	Date and time when the loss occurred. You must specify both the date and time of the incident; otherwise, the API returns an error. To specify only the date of loss, use the <code>claimLossDate</code> parameter.	Optional	65.0
lossType	String	Type of the loss for which the claim is initiated.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	Claim Options Input	Options for creating or updating a claim.	Optional	65.0
participants	Claim Participant Input[]	Claim participants that are associated with the claim.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
productCode	String	Product code of the claim product.	Required to create a claim	65.0

Response body for PATCH[Claim Response](#)

Claim Coverage (POST)

Create a claim coverage record for the specified claim.

Resource

```
/connect/claims/claimId/coverages
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/claims/0Zkxx000000001dEAA/coverages
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim to create the coverages for.	Required	65.0

Request body for POST**JSON example**

```
{
  "claimParticipantId": "0kaxx000000001dEAA",
  "claimItemId": "0koxx000000001dEAA",
  "insurancePolicyAssetId": "02ixx000000415sAAA",
  "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
  "reserveProcessingMode": "Manual",
  "currencyIsoCode": "USD",
  "lossReserveAmount": 5000.00,
  "expenseReserveAmount": 750.50,
  "additionalFields": {
    "RequireInspection": true,
    "CoverageSubtype": "Collision-Standard"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields to create or update the record.	Optional	65.0
claimItemId	String	ID of the claim item.	Required	65.0
claimParticipantId	String	ID of the claim participant that's associated with the claim coverage.	Required	65.0
currencyIsoCode	String	Currency ISO code for reserve adjustments.	Optional	65.0
expenseReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
insurancePolicyAssetId	String	ID of the asset covered by the insurance policy that's associated with the claim coverage.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage.	Optional	65.0
lossReserveAmount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0
name	String	Name of the claim coverage.	Optional	65.0
reserveProcessingMode	String	Specifies how financial reserves are managed when a claim coverage is created. The default value is CoverageReserve.	Optional	65.0

Response body for POST

[Claim Coverage](#)

Claim Coverage (PATCH)

Update a specific claim coverage record.

Resource

`/connect/claims/claimId/coverages/coverageId`

Resource example

`https://yourInstance.salesforce.com/services/data/v/connect/claims/0210xx000000001dEAA/coverages/01Px000000002FEAA`

Available version

65.0

HTTP methods

PATCH

Path parameter for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim to update the coverage for.	Required	65.0
coverageId	String	ID of the coverage within the claim that must be updated.	Required	65.0

Request body for PATCH**JSON example**

```
{
  "expenseReserveAmount": 500.00,
  "lossReserveAmount": 2000.00,
  "additionalFields": {
    "Status": "Pending Review"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additional Fields	Object	Additional standard or custom fields to update the record.	Optional	65.0
expense ReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
lossReserve Amount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0

Response body for PATCH[Claim Coverage](#)**Claim Coverage Payment Detail (POST)**

Create a claim coverage payment detail record. This also generates the related records for the Insurance Policy Limit Tracking and Claim Coverage Payment Adjustment records if applicable.

Name	Type	Description	Required or Optional	Available Version
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	Required when coveragePaymentType is Loss	65.0
benefitName	String	Name of the benefit that's associated with the product attribute's mapped scope as a sub-coverage.	Required when coveragePaymentType is Loss	65.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Required	65.0
coveragePaymentType	String	Specifies the category of the payment. The value must be either Loss or Expense. Loss refers to the payment related to the policyholder's covered damages. Expense refers to costs associated with claim processing, such as vendor fees.	Required	65.0
description	String	Description of the claim coverage payment detail record.	Optional	65.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	65.0
recipientId	String	ID of the person or entity who will receive the payment, such as the policyholder, a third-party claimant, or a service vendor.	Optional	65.0

Response body for POST

Claim Coverage Payment Detail

Claim Coverage Payment Detail (PATCH)

Modify an existing claim coverage payment detail record for both loss and expense types.

For loss-type payments, the system automatically re-evaluates the claim by reversing previously applied limits and recalculating adjustments such as deductible and copay before saving the updated payment. For expense-type payments, you can update fields, such as claim amount and recipient. These updates do not trigger any recalculation of adjustments or policy limits.

Resource

```
/connect/insurance/claims/claimId/coverages/coverageId/payment-details/paymentDetailId
```

Resource example

<https://yourinstance.salesforce.com/services/data/v/comet/insurance/claims/01x00000000000000000000000000000000/covages/01x00000000000000000000000000000000/payer-details/012x00000000000000000000000000000000>

Available version

66.0

HTTP methods

PATCH

Path parameters for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim.	Required	66.0
coverageId	String	ID of the coverage that applies to the claim.	Required	66.0
paymentDetailId	String	ID of the claim coverage payment detail record.	Required	66.0

Request body for PATCH**JSON example**

```
{
  "claimedAmount": 5000.00,
  "adjustedAmount": 5000.0,
  "recipientId": "001xx000003DGQyAAO",
  "name": "Rental Car Expense",
  "description": "Payment for 5 days of rental car usage",
  "limitUnitCount": 5,
  "status": "Draft",
  "additionalFields": {
    "State": "CA",
    "policyType": "Auto",
    "LOB__c": "Personal"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields that are used to update the claim coverage payment detail.	Optional	66.0
adjustedAmount	Double	Adjusted amount of the claim.	Required if claimedAmount is specified for a loss item.	66.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Optional	66.0

Name	Type	Description	Required or Optional	Available Version
description	String	Description of the claim line item.	Optional	66.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	66.0
name	String	Name of the claim coverage payment detail record.	Optional	66.0
recipientId	String	ID of the person or entity who will receive the payment, such as the policyholder, a third-party claimant, or a service vendor. For a Loss item, the recipient is the policyholder or a third party. For an Expense item, the recipient ID represents a vendor or internal staff member.	Optional	66.0
status	String	Current status of the claim coverage payment detail, such as Draft.	Optional	66.0

Response body for PATCH

[Edit Claim Coverage Payment Detail Response](#)

Delete Claim Coverage Payment Detail (POST)

Remove unpaid claim coverage payment details by updating the status to Deleted. The API reverses the insurance policy limit tracking data to roll back consumption for the type Loss.

Resource

```
/connect/insurance/claims/claimId/coverages/coverageId/payment-details/paymentDetailId/delete
```

Resource example

```
https://yourinstance.salesforce.com/services/data/v/connect/insurance/claims/01s0000001d3a/coverages/01s00000025a/payment-details/01s00000035a/delete
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim whose coverage payment detail must be deleted.	Required	65.0

Parameter Name	Type	Description	Required or Optional	Available Version
coverageId	String	ID of the claim coverage record.	Required	65.0
paymentDetailId	String	ID of the claim coverage payment detail record.	Required	65.0

Response body for POST

[Delete Claim Coverage Payment Detail](#)

Field Set Fields (GET)

Get the fields of a field set by the field set ID and object name.

Resource

```
/connect/fieldset/objectName/fieldSetId/fields
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/fieldset/ClaimCoveragePaymentDetail/0182819gvaBgtvch/fields
```

Available version

65.0

HTTP methods

GET

Path parameter for GET

Parameter Name	Type	Description	Required or Optional	Available Version
fieldSetId	String	ID of the field set to get the associated fields for.	Required	65.0
objectName	String	API name of the object to get the associated fields for.	Required	65.0

Response body for GET

[Field Set Fields](#)

Policy Limits (POST)

Create insurance policy limit records for the specified policy based on its defined product attributes and coverages.

Resource

```
/connect/insurance/policies/policyId/generate-limits
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx0000000001qEAA/generate-limits
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of the insurance policy for which the insurance policy limit records must be created.	Required	65.0

Request body for POST**JSON example**

```
{
  "attributeScopes": {
    "list": [
      "Policy",
      "PolicyCoverage"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
attributeScopes	String[]	Attribute scopes for which the policy limit records must be created. Valid values are Policy, PolicyCoverage, Claim, ClaimCoverage.	Optional	65.0

Response body for POST[Generate Policy Limits Response](#)

Insurance Policy Limit Current Standings (POST)

Get the calculated current standings (consumption status) for all applicable insurance policy limits. This API retrieves standings based on the policy, claim, claim coverage as well as policy coverage, insurance policy participant, and insurance policy asset.

This API accepts only one ID per request, either `policyId` or `claimId`.

Resource

```
/connect/insurance/policy-limit/calculate-standings
```

Resource example

`https://yourInstance.salesforce.com/services/data/v/connect/insurance/policy-limit/calculate-standings`

Available version

65.0

HTTP methods

POST

Request body for POST

JSON example

This example queries for standings at the entire policy level, using the `viewOption` parameter to get a summary, including pending amounts.

```
{
  "policyId": "01t000000000000001",
  "claimId": null,
  "claimCoverageId": null,
  "policyParticipantId": null,
  "policyAssetId": null,
  "policyCoverageId": null,
  "trackingDate": "2025-10-19",
  "options": {
    "viewOption": "showAllPolicyCoverages",
    "shouldIncludePendingAmount": true
  }
}
```

This example gets the specific limits that are impacted by a single claim, excluding pending amounts.

```
{
  "policyId": null,
  "claimId": "01t000000000000001",
  "claimCoverageId": null,
  "policyParticipantId": null,
  "policyAssetId": null,
  "policyCoverageId": null,
  "trackingDate": "2025-10-19",
  "options": {
    "viewOption": "showOpenClaimPolicyCoverages",
    "shouldIncludePendingAmount": false
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
claimCoverageId	String	ID of the claim coverage record.	Optional	65.0
claimId	String	ID of the claim.	Required to get the standings for a claim	65.0

Name	Type	Description	Required or Optional	Available Version
options	Limit Standing Option Input	Options for the calculation.	Optional	65.0
policyAssetId	String	ID of the policy asset.	Optional	65.0
policyCoverageId	String	ID of the policy coverage record.	Optional	65.0
policyId	String	ID of the insurance policy.	Required to get the standings for a policy	65.0
policyParticipantId	String	ID of the policy participant.	Optional	65.0
trackingDate	Date	Date on which the standing is calculated. The default value is the current system date.	Optional	66.0

Response body for POST[Insurance Policy Limit Current Standings Response](#)**Pay Claim Coverage Payment Detail (POST)**

Process the payment for a claim coverage payment detail record.

Resource

```
/connect/insurance/claims/claimId/coverages/coverageId/payment-details/paymentDetailId/pay
```

Resource example

```
https://yourinstance.salesforce.com/services/data/connect/insurance/claims/01s000000101010101/coverages/01s000000202020202/payment-details/01s000000303030303/pay
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim under which the payment is made.	Required	65.0
coverageId	String	ID of the claim coverage record that the payment detail is associated with.	Required	65.0

Parameter Name	Type	Description	Required or Optional	Available Version
paymentDetailId	String	ID of the claim coverage payment detail record that you want to pay.	Required	65.0

Response body for POST

Pay Claim Coverage Payment Detail

Pay Ex Gratia Claim Coverage Payment Detail (POST)

Process a discretionary ex gratia payment for a claim coverage payment detail, enabling payment even when coverage or policy limits are exceeded.

Use this resource to pay the full adjusted amount for a loss type payment detail, regardless of whether policy limits are reached. The system records limit consumption only up to the remaining available limit and tracks the excess as an ex gratia amount.

Resource

/connect/insurance/claims/**claimId**/coverages/**coverageId**/payment-details/**paymentDetailId**/pay-ex-gratia

Resource example

<https://yourinstance.salesforce.com/services/data/v/comet/insure/dans/02x00000001674/comets/02x00000002574/comet-details/02x00000003574/ave-data>

Available version

66.0

HTTP methods

POST

Path parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim.	Required	66.0
coverageId	String	ID of the coverage that applies to the claim.	Required	66.0
paymentDetailId	String	ID of the claim coverage payment detail record.	Required	66.0

Request body for POST

JSON example

```
{
  "reason": "Customer goodwill gesture despite exceeded limits."
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
reason	String	Reason for the ex gratia payment.	Optional	66.0

Response body for POST

Pay Ex Gratia Claim Coverage Payment Detail

Process Policy Limits (POST)

Process the policy limits for a loss item in a claim by using the current standings of the insurance policy limits.

This API uses the default Expression Set Executable Template to evaluate how a specific transaction affects the monetary and non-monetary policy limits. You can either extend the executable template to create a custom Expression Set or invoke an Integration Procedure for more complex calculations. This API uses coverage and transaction details as input and returns a detailed breakdown of how each applicable limit is consumed by the proposed amount.

Resource

/connect/insurance/claims/**claimId**/coverages/**coverageId**/process-limits

Resource example

<https://yourInstance.salesforce.com/services/data/v/connect/insurance/claims/01xxx00000000010EAA/coverages/01Pxx0000000002EAA/process-limits>

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim to process the policy limits for.	Required	65.0
coverageId	String	ID of the claim coverage within the claim that the loss item belongs to.	Required	65.0

Request body for POST

JSON example

```
{
  "adjustedAmount": 1250.00,
  "benefitName": "Collision Coverage",
  "limitUnitCount": 1
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
adjustedAmount	Double	Amount after adjustments are made based on the current standings of the policy limits.	Optional	65.0
benefitName	String	Name of the specific benefit or sub-coverage that the loss item applies to.	Optional	65.0
limitUnitCount	Integer	Unit count that the payment detail represents. Default value is 1.	Optional	65.0

Response body for POST

[Process Policy Limits Response](#)

Verify Policy Coverage (POST)

Verify the applicability of a policy coverage for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

Resource

```
/connect/insurance/policies/policyId/coverages/coverageId/verify
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0150000000015224/coverages/0150000000026324/verify
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of the insurance policy that the coverage belongs to.	Required	65.0
coverageId	String	ID of the coverage within the insurance policy to verify the details for.	Required	65.0

Request body for POST**JSON example**

```
{
  "lossDate": "2025-09-13",
  "insuredAssetOrParticipantId": "0kaxx000000001dEAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
insuredAsset Or ParticipantId	String	ID of the insured asset or participant that's related to the policy coverage.	Required	65.0
lossDate	String	Date on which the incident occurred.	Required	65.0

Response body for POST[Verify Policy Coverage](#)

Requests

Learn more about the available requests of insurance claim APIs.

[Calculate Adjustments Input](#)

Input representation for calculating adjustments for the claim loss item.

[Claim Attribute Input](#)

Input representation for claim attributes.

[Claim Coverage Input](#)

Input representation for creating a claim coverage.

[Claim Coverage Update Input](#)

Input representation for updating a claim coverage.

[Claim Coverage Payment Detail Input](#)

Input representation for creating a claim coverage payment detail record.

[Claim Input](#)

Input representation for creating or updating a claim.

[Claim Item Input](#)

Input representation for a claim item.

[Claim Options Input](#)

Input representation for claim options.

[Claim Participant Input](#)

Input representation for a claim participant.

[Claim Related Object Input](#)

Input representation for a related object on a claim.

[Edit Claim Coverage Payment Detail Input](#)

Details to edit a claim coverage payment detail record.

[Generate Policy Limits Input](#)

Input representation for creating insurance policy limit records.

[Insurance Policy Limit Current Standings Input](#)

Input representation for the current standings of the insurance policy limits.

[Limit Standing Option Input](#)

Input representation for limit standing options.

[Pay Ex Gratia Claim Coverage Payment Detail Input](#)

Input representation for the ex gratia payment for the claim coverage payment detail.

[Process Policy Limits Input](#)

Input representation for processing policy limits for a claim loss item.

[Recalculate Adjustments Input](#)

Input representation for recalculating the adjustments for existing payment details.

[Verify Policy Coverage Input](#)

Input representation of a policy coverage's applicability for a specified loss date under an insurance policy.

Calculate Adjustments Input

Input representation for calculating adjustments for the claim loss item.

JSON example

```
{
  "benefitName": "Car Rental",
  "claimedAmount": 1500
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
benefitName	String	Name of the specific benefit or sub-coverage that's associated with the coverage, as defined in the product attribute mapping.	Optional	66.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Required	65.0

Claim Attribute Input

Input representation for claim attributes.

Properties

Name	Type	Description	Required or Optional	Available Version
apiName	String	API name of the claim attribute.	Optional	65.0
value	String	Value for the claim attribute.	Optional	65.0

Claim Coverage Input

Input representation for creating a claim coverage.

JSON example

```
{
  "claimParticipantId": "0kaxx000000001dEAA",
  "claimItemId": "0koxx000000001dEAA",
  "insurancePolicyAssetId": "02ixx000000415sAAA",
  "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
  "reserveProcessingMode": "Manual",
  "currencyIsoCode": "USD",
  "lossReserveAmount": 5000.00,
  "expenseReserveAmount": 750.50,
  "additionalFields": {
    "RequireInspection": true,
    "CoverageSubtype": "Collision-Standard"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields to create or update the record.	Optional	65.0
claimItemId	String	ID of the claim item.	Required	65.0
claimParticipantId	String	ID of the claim participant that's associated with the claim coverage.	Required	65.0
currencyIsoCode	String	Currency ISO code for reserve adjustments.	Optional	65.0
expenseReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
insurancePolicyAssetId	String	ID of the asset covered by the insurance policy that's associated with the claim coverage.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
insurancePolicy CoverageId	String	ID of the insurance policy coverage.	Optional	65.0
lossReserveAmount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0
name	String	Name of the claim coverage.	Optional	65.0
reserve ProcessingMode	String	Specifies how financial reserves are managed when a claim coverage is created. The default value is CoverageReserve.	Optional	65.0

Claim Coverage Update Input

Input representation for updating a claim coverage.

JSON example

```
{
  "expenseReserveAmount": 500.00,
  "lossReserveAmount": 2000.00,
  "additionalFields": {
    "Status": "Pending Review"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additional Fields	Object	Additional standard or custom fields to update the record.	Optional	65.0
expense ReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
lossReserve Amount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0

Claim Coverage Payment Detail Input

Input representation for creating a claim coverage payment detail record.

JSON example

```
{
  "claimedAmount": 1200.0,
  "recipientId": "001xx000003GN2rAAG",
  "coveragePaymentType": "Loss",
  "benefitName": "DentalCov",
  "description": "Dental cleaning and checkup.",
  "limitUnitCount": 1,
  "adjustedAmount": 1200.0,
  "additionalFields": {
    "State__c": "CA",
    "LOB__c": "Commercial"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields that are used to create the claim coverage payment detail record.	Optional	65.0
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	Required when ~~coveragePaymentType~~ is Loss	65.0
benefitName	String	Name of the benefit that's associated with the product attribute's mapped scope as a sub-coverage.	Required when ~~coveragePaymentType~~ is Loss	65.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Required	65.0
coveragePaymentType	String	Specifies the category of the payment. The value must be either Loss or Expense. Loss refers to the payment related to the policyholder's covered damages. Expense refers to costs associated with claim processing, such as vendor fees.	Required	65.0
description	String	Description of the claim coverage payment detail record.	Optional	65.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
recipientId	String	ID of the person or entity who will receive the payment, such as the policyholder, a third-party claimant, or a service vendor.	Optional	65.0

Claim Input

Input representation for creating or updating a claim.

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	Object	Additional standard or custom fields that are used to create or update insurance claim.	Optional	65.0
attributes	Claim Attribute Input[]	Attributes of the claim product.	Optional	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't accept time of incident. To specify both the date and time of loss, use the <code>lossDate</code> parameter.	Optional	66.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Category that the claim reason belongs to.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0
incidentTypeId	String	ID for the type of incident in the claim.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	Claim Item Input[]	Details of the items that are associated with the claim.	Optional	65.0
lossDate	Date	Date and time when the loss occurred. You must specify both the date and time of the incident; otherwise, the API returns an error. To specify only the date of loss, use the <code>claimLossDate</code> parameter.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
lossType	String	Type of the loss for which the claim is initiated.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	Claim Options Input	Options for creating or updating a claim.	Optional	65.0
participants	Claim Participant Input[]	Claim participants that are associated with the claim.	Optional	65.0
productCode	String	Product code of the claim product.	Required to create a claim	65.0

Claim Item Input

Input representation for a claim item.

Properties

Name	Type	Description	Required or Optional	Available Version
action	String	Action to perform on the record, such as create, update, and delete. Supported only for Claims (PATCH) requests that add, update, or remove a claim item.	Required to update claim item	65.0
additionalFields	Object	Additional standard or custom fields for the claim item.	Optional	65.0
attributes	Claim Attribute Input[]	Attributes of the claim item.	Optional	65.0
category	String	Category of the claim item.	Required	65.0
instanceKey	String	Unique identifier of the claim item.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage that applies to the claim item.	Optional	65.0
insuredItemId	String	ID of the insured item that's involved in the claim.	Optional	65.0
name	String	Name of the claim item.	Optional	65.0
participantInstanceKey	String	Unique identifier that's used to link a claim item to a specific participant.	Optional	65.0
productCode	String	Product code of the claim item product.	Required	65.0

Name	Type	Description	Required or Optional	Available Version
relatedObjects	Claim Related Object Input[]	Related claim item records.	Optional	65.0

Claim Options Input

Input representation for claim options.

Properties

Name	Type	Description	Required or Optional	Available Version
execute Configuration Rules	String	Specifies whether to execute configuration rules or not.	Optional	65.0

Claim Participant Input

Input representation for a claim participant.

JSON example

```
[
  {
    "instanceKey": "participant_driver_01",
    "contactId": "003xx000003DGQyAAO",
    "accountId": "001xx000003DGQyAAO",
    "insurancePolicyParticipantId": "0pPxx000000001qEAA",
    "isInjured": false,
    "roles": {
      "list": [
        "Driver",
        "Claimant"
      ]
    },
    "additionalFields": {
      "StatementTaken": true
    },
    "action": "update"
  }
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	Account ID of the claim participant.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
action	String	Action to perform on the participant record, such as create, update, or delete. Supported only for Claims (PATCH) requests that add, update, or remove a claim participant.	Required to update claim participant	65.0
additionalFields	Object	Additional standard or custom fields to add details for the claim participant.	Optional	65.0
contactId	String	Contact ID of the claim participant.	Optional	65.0
instanceKey	String	Unique identifier for the participant.	Optional	65.0
insurancePolicyParticipantId	String	ID of the insurance policy participant that's related to the claim.	Optional	65.0
isInjured	String	Specifies if the participant is injured or not.	Optional	65.0
roles	String[]	Roles of the claim participant.	Required	65.0

Claim Related Object Input

Input representation for a related object on a claim.

JSON example

```
{
  "relatedObjects": [
    {
      "relatedRecordId": "500xx000000OpQrStU",
      "relatedRecordObjName": "Police_Report__c"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
recordId	String	ID of the related claim item record.	Optional	65.0
recordObjName	String	Object name of the related claim item record.	Optional	65.0

Edit Claim Coverage Payment Detail Input

Details to edit a claim coverage payment detail record.

JSON example

```
{
  "claimedAmount": 5000.00,
  "adjustedAmount": 5000.0,
  "recipientId": "001xx000003DGQyAAO",
  "name": "Rental Car Expense",
  "description": "Payment for 5 days of rental car usage",
  "limitUnitCount": 5,
  "status": "Draft",
  "additionalFields": {
    "State": "CA",
    "policyType": "Auto",
    "LOB__c": "Personal"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields that are used to update the claim coverage payment detail.	Optional	66.0
adjustedAmount	Double	Adjusted amount of the claim.	Required if <code>claimedAmount</code> is specified for a loss item.	66.0
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Optional	66.0
description	String	Description of the claim line item.	Optional	66.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	66.0
name	String	Name of the claim coverage payment detail record.	Optional	66.0
recipientId	String	ID of the person or entity who will receive the payment, such as the policyholder, a third-party claimant, or a service vendor. For a Loss item, the recipient is the policyholder or a third party. For an Expense item, the recipient ID represents a vendor or internal staff member.	Optional	66.0
status	String	Current status of the claim coverage payment detail, such as Draft.	Optional	66.0

Generate Policy Limits Input

Input representation for creating insurance policy limit records.

JSON example

```
{
  "attributeScopes": {
    "list": [
      "Policy",
      "PolicyCoverage"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
attributeScopes	String[]	Attribute scopes for which the policy limit records must be created. Valid values are Policy, PolicyCoverage, Claim, ClaimCoverage.	Optional	65.0

Insurance Policy Limit Current Standings Input

Input representation for the current standings of the insurance policy limits.

JSON example

This example queries for standings at the entire policy level, using the `viewOption` parameter to get a summary, including pending amounts.

```
{
  "policyId": "01t000000000000001",
  "claimId": null,
  "claimCoverageId": null,
  "policyParticipantId": null,
  "policyAssetId": null,
  "policyCoverageId": null,
  "trackingDate": "2025-10-19",
  "options": {
    "viewOption": "showAllPolicyCoverages",
    "shouldIncludePendingAmount": true
  }
}
```

This example gets the specific limits that are impacted by a single claim, excluding pending amounts.

```
{
  "policyId": null,
  "claimId": "01t000000000000001",
  "claimCoverageId": null,
  "policyParticipantId": null,
  "policyAssetId": null,
}
```

```

"policyCoverageId": null,
"trackingDate": "2025-10-19",
"options": {
  "viewOption": "showOpenClaimPolicyCoverages",
  "shouldIncludePendingAmount": false
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
claimCoverageId	String	ID of the claim coverage record.	Optional	65.0
claimId	String	ID of the claim.	Required to get the standings for a claim	65.0
options	Limit Standing Option Input	Options for the calculation.	Optional	65.0
policyAssetId	String	ID of the policy asset.	Optional	65.0
policyCoverageId	String	ID of the policy coverage record.	Optional	65.0
policyId	String	ID of the insurance policy.	Required to get the standings for a policy	65.0
policyParticipantId	String	ID of the policy participant.	Optional	65.0
trackingDate	Date	Date on which the standing is calculated. The default value is the current system date.	Optional	66.0

Limit Standing Option Input

Input representation for limit standing options.

Properties

Name	Type	Description	Required or Optional	Available Version
shouldIncludePendingAmount	Boolean	Indicates whether to include the pending amount in the calculated output (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
viewOption	String	Filter values for coverages. When a <code>claimId</code> is specified in the request, valid values are: • ~~showAllPolicyCoverages—Retrieves~~ all policy limits for the root insurance policy and all associated policy coverages. This provides a comprehensive view of all coverage terms. • ~~showOpenClaimPolicyCoverages—(Default)~~ Retrieves the policy limits that are associated with the open claim coverages. Additionally, the claim view includes all policy-level and claim-level limits, ensuring visibility into limits that impact the claim, even if no open coverage exists. When a <code>policyId</code> is specified in the request, valid values are: • ~~showAllPolicyCoverages—(Default)~~ Retrieves all policy limit standings, regardless of whether the claim has open coverages. This option provides a full view of the policy's limits without restriction to active claims. • ~~showPolicyLevelLimitsOnly—Retrieves~~ policy limit standings at the policy and policy coverage level, but excludes limits that are associated with policy participants or policy assets. Use this option when only high-level policy limits are needed without participant or asset specific details.	Optional	65.0

Pay Ex Gratia Claim Coverage Payment Detail Input

Input representation for the ex gratia payment for the claim coverage payment detail.

JSON example

```
{
  "reason": "Customer goodwill gesture despite exceeded limits."
}
```


Properties

Name	Type	Description	Required or Optional	Available Version
reason	String	Reason for the ex gratia payment.	Optional	66.0

Process Policy Limits Input

Input representation for processing policy limits for a claim loss item.

JSON example

```
{
  "adjustedAmount": 1250.00,
  "benefitName": "Collision Coverage",
  "limitUnitCount": 1
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
adjusted Amount	Double	Amount after adjustments are made based on the current standings of the policy limits.	Optional	65.0
benefitName	String	Name of the specific benefit or sub-coverage that the loss item applies to.	Optional	65.0
limitUnit Count	Integer	Unit count that the payment detail represents. Default value is 1.	Optional	65.0

Recalculate Adjustments Input

Input representation for recalculating the adjustments for existing payment details.

JSON example

```
{
  "claimedAmount": 1200.00,
  "paymentDetailId": "012xx0000000001CAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Optional	66.0

Name	Type	Description	Required or Optional	Available Version
paymentDetailId	String	ID of the claim coverage payment detail for recalculating adjustments.	Optional	66.0

Verify Policy Coverage Input

Input representation of a policy coverage's applicability for a specified loss date under an insurance policy.

JSON example

```
{
  "lossDate": "2025-09-13",
  "insuredAssetOrParticipantId": "0kaxx000000001dEAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
insuredAsset Or ParticipantId	String	ID of the insured asset or participant that's related to the policy coverage.	Required	65.0
lossDate	String	Date on which the incident occurred.	Required	65.0

Responses

Learn more about the available response bodies of insurance claim APIs.

[Calculate Adjustments](#)

Output representation of the details for calculating the adjustments for the claim loss item.

[Cancel Claim Coverage Payment Detail](#)

Output representation for the canceled claim coverage payment detail.

[Claim Attribute](#)

Output representation for a claim attribute.

[Claim Coverage](#)

Output representation for a claim coverage.

[Claim Coverage Payment Adjustment Details](#)

Output representation for the claim coverage payment adjustment details.

[Claim Coverage Payment Detail](#)

Output representation for creating the claim coverage payment detail record.

[Claim Details](#)

Output representation for the details of a claim record.

[Claim Response](#)

Output representation for a created or updated claim.

[Claim Item](#)

Output representation for the details of a claim item.

[Claim Participant](#)

Output representation for the details of a claim participant.

[Claim Related Object](#)

Output representation for a related object on a claim.

[Delete Claim Coverage Payment Detail](#)

Output representation for the deleted claim coverage payment detail.

[Edit Claim Coverage Payment Detail Response](#)

Output representation for the edited claim coverage payment detail.

[Error Detail](#)

Output representation of the error details.

[Field Set Fields](#)

Output representation of the collection of fields that are associated with a specific field set.

[Field Set Field Details](#)

Output representation of the field that's associated with a specific field set.

[Generate Policy Limits Response](#)

Output representation for the created insurance policy limit records.

[Insurance Policy Limit Calculation Standing Node](#)

Output representation for an insurance policy limit standing calculation node.

[Insurance Policy Limit Current Standings Response](#)

Output representation for the current standings of the insurance policy limits.

[Policy Limit Tracking Details](#)

Output representation for the insurance policy limit tracking details.

[Pay Claim Coverage Payment Detail](#)

Output representation for paying a claim coverage payment detail.

[Pay Ex Gratia Claim Coverage Payment Detail](#)

Output representation for the details of the ex gratia payment that's made for the claim coverage payment detail.

[Process Policy Limits Response](#)

Output representation for processing the policy limits for a claim loss item.

[Verify Policy Coverage](#)

Output representation of a policy coverage's applicability for a specified loss date under an insurance policy.

Calculate Adjustments

Output representation of the details for calculating the adjustments for the claim loss item.

JSON example

```
{  
  "isSuccess": true,
```

```

    "adjustedAmount": 1250.00,
    "adjustmentReason": "Deductible applied.",
    "claimCovPaymentAdjustments": [
      {
        "adjustedAmount": 250.00,
        "adjustmentReason": "Deductible"
      }
    ],
    "insPolicyLimitTrackings": [
      {
        "policyLimitId": "01Yx00000004CvQEAU",
        "category": "Per-Incident Limit",
        "remainingAmount": 4750.00
      }
    ],
    "errors": []
  }

```

Property Name	Type	Description	Filter Group and Version	Available Version
adjustedAmount	Double	Amount after adjustments are made based on the current standings of the policy limits.	Big, 65.0	65.0
adjustmentReason	String	Reason for the claim coverage payment detail adjustment such as deductible, copay, or coinsurance.	Big, 65.0	65.0
claimCovPaymentAdjustments	Claim Coverage Payment Adjustment Details[]	List of claim coverage payment adjustment details.	Big, 65.0	65.0
errors	Error Detail []	List of errors that are associated with the API request.	Big, 65.0	65.0
insPolicyLimitTrackings	Insurance Policy Limit Tracking Details[]	List of insurance policy limit tracking details.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Cancel Claim Coverage Payment Detail

Output representation for the canceled claim coverage payment detail.

JSON example

```

{
  "isSuccess": true,
  "paymentDetailId": "012xx000000003gEAA",
  "errors": []
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail []	Details of the errors that are associated with the API request.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0
paymentDetailId	String	ID of the claim coverage payment detail for which the payment is canceled.	Big, 65.0	65.0

Claim Attribute

Output representation for a claim attribute.

Name	Type	Description	Filter Group and Version	Available Version
apiName	String	API name of the product attribute.	Big, 65.0	65.0
value	String	Value for the product attribute.	Big, 65.0	65.0

Claim Coverage

Output representation for a claim coverage.

JSON example

```
{
  "isSuccess": true,
  "coverageId": "0kPxx000000002fEAA",
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
coverageId	String	ID of the claim coverage record.	Big, 65.0	65.0
errors	Error Detail []	Details of the errors that are associated with the API request.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Claim Coverage Payment Adjustment Details

Output representation for the claim coverage payment adjustment details.

Property Name	Type	Description	Filter Group and Version	Available Version
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	Big, 65.0	65.0
adjustmentReason	String	Reason for the claim coverage payment adjustment such as deductible, copay, or coinsurance.	Big, 65.0	65.0

Claim Coverage Payment Detail

Output representation for creating the claim coverage payment detail record.

JSON example

```
{
  "adjustedAmount": 200.0,
  "adjustmentReason": "Deductible",
  "paymentDetailId": "012xx00000001grAAA",
  "insPolicyLimitTrackingIds": [
    "1TExx0000004DUuGAM",
    "1TExx0000004DUvGAM"
  ],
  "isSuccess": true,
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
adjustedAmount	Double	Amount after adjustments are made based on the current standings of the policy limits.	Small, 65.0	65.0
adjustmentReason	String	Reason for the claim coverage payment detail adjustment, such as deductible, copay, or coinsurance.	Small, 65.0	65.0
errors	Error Detail []	List of errors, including detailed information about each error.	Small, 65.0	65.0
insPolicyLimitTrackingIds	List<String>	IDs of the insurance policy limit tracking records.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
paymentDetailId	String	ID of the claim coverage payment detail record that's created.	Small, 65.0	65.0

Claim Details

Output representation for the details of a claim record.

JSON example

```
{
  "claimId": "0Zkxx000000001dEAA",
  "name": "Claim for Auto Accident",
  "accountId": "001xx000003DGQyAAO",
  "insurancePolicyId": "0YTxx0000000001qEAA",
  "lossDate": "2025-09-13",
  "claimAttributes": [
    {
      "apiName": "PoliceReportNumber",
      "value": "SF-2025-09-13-00123"
    }
  ],
  "participants": [
    {
      "id": "0kaxx000000001dEAA",
      "instanceKey": "part1",
      "contactId": "003xx000003DGQyAAO",
      "roles": [
        "Driver"
      ]
    }
  ],
  "items": [
    {
      "id": "0koxx000000001dEAA",
      "instanceKey": "item_vehicle_bumper_01",
      "name": "Damaged Front Bumper",
      "productCode": "AUTO-BUMPER-SEDAN",
      "category": "Auto Physical Damage",
      "insuredItemId": "02ixx000000415sAAA",
      "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
      "attributes": [
        {
          "apiName": "DamageSeverity",
          "value": "Severe"
        }
      ],
      "additionalFields": {
        "RepairEstimate": 1575.50,
        "IsDrivable": false
      },
      "relatedObjects": [
        {
          "relatedRecordId": "500xx000000OpQrStU",
          "relatedRecordObjName": "Police_Report__c"
        }
      ],
      "participantInstanceKey": "participant_driver_1"
    }
  ],
  "errors": []
}
```

Name	Type	Description	Filter Group and Version	Available Version
accountId	String	ID of the account that's associated with the claim.	Big, 65.0	65.0
attributes	Claim Attribute []	Attributes of the claim product.	Big, 65.0	65.0
claimId	String	ID of the claim.	Big, 65.0	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't show the time of incident.	Big, 66.0	66.0
claimReason	String	Reason for the claim.	Big, 65.0	65.0
claimReasonType	String	Category that the claim reason belongs to.	Big, 65.0	65.0
claimType	String	Type of the claim.	Big, 65.0	65.0
contextId	String	ID of the context.	Big, 65.0	65.0
errors	Error Detail []	List of errors, including detailed information about each error.	Big, 65.0	65.0
incidentTypeId	String	ID for the type of incident in the claim.	Big, 65.0	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Big, 65.0	65.0
items	Claim Item []	Details of the claim items.	Big, 65.0	65.0
lossDate	Date	Date and time when the loss occurred.	Big, 65.0	65.0
lossType	String	Type of loss for which the claim is initiated.	Big, 65.0	65.0
name	String	Name of the claim.	Big, 65.0	65.0
participants	Claim Participant []	Details of the claim participants.	Big, 65.0	65.0
productCode	String	Product code of the claim product.	Big, 65.0	65.0

Claim Response

Output representation for a created or updated claim.

JSON example

```
{
  "claimId": "0Zkxx000000001dEAA",
  "isSuccess": true,
  "errors": []
}
```


Properties

Name	Type	Description	Filter Group and Version	Available Version
claimId	String	ID of the claim that's created or updated.	Big, 65.0	65.0
errors	Error Detail[]	List of errors, including detailed information about each error.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Claim Item

Output representation for the details of a claim item.

JSON example

```
{
  "id": "0koxx000000001dEAA",
  "instanceKey": "item_vehicle_bumper_01",
  "name": "Damaged Front Bumper",
  "productCode": "AUTO-BUMPER-SEDAN",
  "category": "Auto Physical Damage",
  "insuredItemId": "02ixx000000415sAAA",
  "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
  "attributes": [
    {
      "apiName": "DamageSeverity",
      "value": "Severe"
    }
  ],
  "additionalFields": {
    "RepairEstimate": 1575.5,
    "IsDrivable": false
  },
  "relatedObjects": [
    {
      "relatedRecordId": "500xx000000OpQrStU",
      "relatedRecordObjName": "Police_Report__c"
    }
  ],
  "participantInstanceKey": "participant_driver_1"
}
```

Name	Type	Description	Filter Group and Version	Available Version
additionalFields	Object	Additional standard or custom fields that are used to create claim items.	Big, 65.0	65.0
attributes	Claim Attribute []	Attributes of the claim item.	Big, 65.0	65.0

Name	Type	Description	Filter Group and Version	Available Version
category	String	Category of the claim item.	Big, 65.0	65.0
id	String	ID of the claim item.	Big, 65.0	65.0
instanceKey	String	Unique identifier of the claim item.	Big, 65.0	65.0
insurance Policy CoverageId	String	ID of the specific policy coverage that applies to the claim item.	Big, 65.0	65.0
insuredItemId	String	ID of the insured item.	Big, 65.0	65.0
name	String	The name of the claim item.	Big, 65.0	65.0
participant InstanceKey	String	Unique identifier that's used to link a claim item to a specific participant.	Big, 65.0	65.0
productCode	String	Product code of the claim product.	Big, 65.0	65.0
relatedObjects	Claim Related Object []	Related claim item records.	Big, 65.0	65.0

Claim Participant

Output representation for the details of a claim participant.

JSON example

```
"participants": [
  {
    "id": "0kaxx000000001dEAA",
    "instanceKey": "participant_driver_1",
    "accountId": "001xx000003DGQyAAO",
    "contactId": "003xx000003DGQyAAO",
    "insurancePolicyParticipantId": "0pPxx000000001qEAA",
    "roles": [
      "Driver",
      "Claimant"
    ]
  }
]
```

Name	Type	Description	Filter Group and Version	Available Version
accountId	String	Account ID of the claim participant.	Big, 65.0	65.0
contactId	String	Contact ID of the claim participant.	Big, 65.0	65.0
id	String	ID of the claim participant.	Big, 65.0	65.0
instanceKey	String	Instance key of the claim participant.	Big, 65.0	65.0

Name	Type	Description	Filter Group and Version	Available Version
insurance Policy ParticipantId	String	ID of the insurance policy participant that's related to the claim.	Big, 65.0	65.0
roles	String[]	Roles of the claim participant.	Small, 65.0	65.0

Claim Related Object

Output representation for a related object on a claim.

JSON example

```
{
  "relatedObjects": [
    {
      "relatedRecordId": "500xx0000000pQrStU",
      "relatedRecordObjName": "Police_Report__c"
    }
  ]
}
```

Name	Type	Description	Filter Group and Version	Available Version
related RecordId	String	ID of the related claim item record.	Big, 65.0	65.0
relatedRecord ObjName	String	Object name of the related claim item record.	Big, 65.0	65.0

Delete Claim Coverage Payment Detail

Output representation for the deleted claim coverage payment detail.

JSON example

```
{
  "isSuccess": true,
  "paymentDetailId": "012xx000000003gEAA",
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail[]	List of errors that are associated with the API operation.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the operation was successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
payment DetailId	String	ID of the claim coverage payment detail record whose status is marked as Deleted.	Big, 65.0	65.0

Edit Claim Coverage Payment Detail Response

Output representation for the edited claim coverage payment detail.

JSON example

```
{
  "adjustedAmount": 4500.00,
  "adjustedReason": "Deductible applied",
  "insurancePolicyLimitTrackingIds": [
    "0qOxx0000004CvQEAU",
    "0qOxx0000004CvREAU"
  ],
  "isSuccess": true,
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
adjustedAmount	Double	Final amount after all deductions, such as deductibles or co-pays. This field is returned when a Loss item is created and adjustments are applied.	Big, 65.0	65.0
adjustedReason	String	Reason for the claim coverage payment detail adjustment, such as deductible, copay, or coinsurance.	Big, 65.0	65.0
errors	Error Detail[]	Errors details when the operation fails.	Big, 65.0	65.0
insurance PolicyLimitTracking Ids	String[]	Comma-separated IDs of the insurance policy limit tracking records.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Error Detail

Output representation of the error details.

JSON example

```
{
  "isSuccess": false,
```

```
"errors": [  
  {  
    "errorCode": "RESOURCE_NOT_FOUND",  
    "message": "Specify a resource and try again."  
  }  
]
```

Property Name	Type	Description	Filter Group and Version	Available Version
code	String	Code associated with the error.	Small, 65.0	65.0
message	String	Message for the error.	Small, 65.0	65.0

Field Set Fields

Output representation of the collection of fields that are associated with a specific field set.

JSON example

This is a sample success response.

```
{  
  "fields": [  
    {  
      "apiName": "Field1",  
      "label": "Field 1"  
    }  
  ],  
  "fieldSetId": "0182819gvahBghvdh",  
  "objectName": "ClaimCoveragePaymentDetail"  
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Field Set Field Details[]	List of fields within the specified field set ID.	Small, 65.0	65.0
fieldSetId	String	ID of the field set whose associated fields are retrieved.	Small, 65.0	65.0
objectName	String	API name of the object whose associated fields are retrieved.	Small, 65.0	65.0

Field Set Field Details

Output representation of the field that's associated with a specific field set.

Property Name	Type	Description	Filter Group and Version	Available Version
apiName	String	API name of the field in the field set.	Small, 65.0	65.0
label	String	UI label of the field in the field set.	Small, 65.0	65.0

Generate Policy Limits Response

Output representation for the created insurance policy limit records.

JSON example

```
{
  "policyLimitIds": [
    "01Yx00000004CvQEAU",
    "01Yx00000004CvREAU"
  ],
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail[]	List of errors that are encountered during the processing of the API request.	Small, 65.0	65.0
policyLimitIds	String[]	List of IDs for the insurance policy limit records that are created.	Big, 65.0	65.0

Insurance Policy Limit Calculation Standing Node

Output representation for an insurance policy limit standing calculation node.

JSON example

```
{
  "insPolicyLmtCalcStandingNode": [
    {
      "applicableAction": [
        "PaymentAttempted"
      ],
      "category": "LimitCurrency",
      "claimCoverageId": "0cYSG000000UYlp2AG",
      "claimId": "500SG0000009HjJ2AU",
      "durationType": "PolicyTerm",
      "initialAmount": 1000,
      "initialValue": 0,
      "parentMappedScopeId": "0YTxx0000000253GAA",
      "pendingAmount": 500,
      "pendingValue": 0,
      "policyAssetId": "",
      "policyCoverageId": "0cYSG000000UYlq2AG",
    }
  ]
}
```

```

    "policyLimitId": "1TDSG0000007HU84AM",
    "policyLimitStandingKey": "PolicyCoverage LimitCurrency",
    "policyParticipantId": "003SG0000005pIkJAI",
    "predictedRemainingAmount": 500,
    "predictedRemainingValue": 0,
    "scope": "PolicyCoverage",
    "remainingAmount": 1000,
    "remainingValue": 0,
    "totalUsedAmount": 0,
    "totalUsedValue": 0,
    "trackingStartDate": "2025-10-09",
    "trackingEndDate": "2025-12-19"
  }
]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
applicable Actions	String[]	List of actions that initiate attribute processing.	Big, 65.0	65.0
category	String	Category of the limit.	Big, 65.0	65.0
claimCoverage Id	String	ID of the related claim coverage.	Big, 65.0	65.0
claimId	String	ID of the related claim.	Big, 65.0	65.0
durationType	String	Duration type over which attribute consumption is monitored.	Big, 65.0	65.0
initialAmount	Double	Configured amount.	Big, 65.0	65.0
initialValue	Integer	Configured value of the limit.	Big, 65.0	65.0
itemType	String	Type of the item.	Big, 65.0	65.0
parentMapped ScopeId	String	ID of the parent product or coverage that this attribute applies to.	Big, 65.0	65.0
pendingAmount	Double	Total pending amount for open claims or unfinalized transactions, even if the amount exceeds the initial limit.	Big, 65.0	65.0
pendingValue	Integer	Value that's pending.	Big, 65.0	65.0
policyAssetId	String	ID of the related policy asset.	Big, 65.0	65.0
policy CoverageId	String	ID of the related policy coverage.	Big, 65.0	65.0
policyLimitId	String	ID of the insurance policy limit record.	Big, 65.0	65.0
policyLimit StandingKey	String	Unique identifier for the insurance policy limit standing.	Big, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
policyParticipantId	String	ID of the related policy participant.	Big, 65.0	65.0
predictedRemainingAmount	Double	Remaining amount after calculating the pending value.	Big, 65.0	65.0
predictedRemainingValue	Integer	Remaining value after calculating the pending value.	Big, 65.0	65.0
prodAttrMappedScopeId	String	ID of the product or coverage that this attribute applies to.	Big, 65.0	65.0
remainingAmount	Double	Remaining amount to carry forward.	Big, 65.0	65.0
remainingValue	Integer	Remaining value to carry forward.	Big, 65.0	65.0
scope	String	Scope where the limit is applied, such as policy, claim, policy coverage.	Big, 65.0	65.0
subCoverageName	String	Name of the sub-coverage.	Big, 65.0	65.0
totalUsedAmount	Double	Total amount that's used against this limit to date. It shows the actual total amount that's used, even if that amount exceeds the defined maximum limit.	Big, 65.0	65.0
totalUsedValue	Integer	Total value that's used against this limit to date.	Big, 65.0	65.0
trackingEndDate	Date	End date of the duration period that's used to calculate the current standing.	Big, 66.0	66.0
trackingStartDate	Date	Start date of the duration period that's used to calculate the current standing.	Big, 66.0	66.0

Insurance Policy Limit Current Standings Response

Output representation for the current standings of the insurance policy limits.

JSON example

```
{
  "isSuccess": true,
  "errors": [],
  "insPolicyLmtCalcStandingNode": [
    {
      "applicableAction": [
        "PaymentAttempted"
      ]
    }
  ]
}
```



```
    ],
    "category": "LimitCurrency",
    "claimCoverageId": "0cYSG000000UYlp2AG",
    "claimId": "500SG0000009HjJ2AU",
    "durationType": "PolicyTerm",
    "initialAmount": 1000,
    "initialValue": 0,
    "parentMappedScopeId": "0YTxx0000000253GAA",
    "pendingAmount": 500,
    "pendingValue": 0,
    "policyAssetId": "",
    "policyCoverageId": "0cYSG000000UYlq2AG",
    "policyLimitId": "1TDSG0000007HU84AM",
    "policyLimitStandingKey": "PolicyCoverage LimitCurrency",
    "policyParticipantId": "003SG0000005pIkJAI",
    "predictedRemainingAmount": 500,
    "predictedRemainingValue": 0,
    "scope": "PolicyCoverage",
    "remainingAmount": 1000,
    "remainingValue": 0,
    "totalUsedAmount": 0,
    "totalUsedValue": 0,
    "trackingStartDate": "2025-10-09",
    "trackingEndDate": "2025-12-19"
  }
]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail[]	List of errors for calculate standings.	Big, 65.0	65.0
insPolicyLmt CalcStanding Nodes	Insurance Policy Limit Calculation Standing Node[]	List of policy limit current standing nodes.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (true) or not (false).	Big, 65.0	65.0

Policy Limit Tracking Details

Output representation for the insurance policy limit tracking details.

JSON example

```
"insPolicyLimitTrackings": [
  {
    "policyLimitId": "0plx00000000ABCDEF",
    "category": "Deductible",
    "scope": "PerMember",
    "postedAmount": 750.00,
    "postedValue": null,
    "calcInitialAmount": 500.00,
```

```

    "calcInitialValue": null,
    "remainingAmount": 750.00,
    "remainingValue": null,
    "totalUsedAmount": 1250.00,
    "totalUsedValue": null
  }
]

```

Property Name	Type	Description	Filter Group and Version	Available Version
benefitName	String	Name of the benefit that's associated with the policy limit tracking.	Big, 65.0	65.0
calcInitial Amount	Double	Calculated amount that's used as the base value for tracking the policy attribute consumption.	Big, 65.0	65.0
calcInitial Value	Integer	Count that's used as the base for tracking the policy attribute consumption.	Big, 65.0	65.0
category	String	Category of the associated attribute.	Big, 65.0	65.0
policyLimitId	String	ID of the insurance policy limit record.	Big, 65.0	65.0
postedAmount	Double	Amount that's requested by the claimant when submitting the loss.	Big, 65.0	65.0
postedValue	Integer	Count that's populated when submitting the loss.	Big, 65.0	65.0
remaining Amount	Double	Remaining amount to carry forward for the next loss processing.	Big, 65.0	65.0
remaining Value	Integer	Remaining value to carry forward for the next loss processing.	Big, 65.0	65.0
scope	String	Scope for tracking the associated attribute.	Big, 65.0	65.0
totalUsed Amount	Double	Total amount that's used till date.	Big, 65.0	65.0
totalUsed Value	Integer	Total value that's used till date.	Big, 65.0	65.0

Pay Claim Coverage Payment Detail

Output representation for paying a claim coverage payment detail.

JSON example

```

{
  "isSuccess": true,
  "errors": []
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail[]	Details of the errors that are associated with the API operation.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Pay Ex Gratia Claim Coverage Payment Detail

Output representation for the details of the ex gratia payment that's made for the claim coverage payment detail.

JSON example

```
{
  "claimPaymentSummaryId": "0sDxx0000000001CAA",
  "consumedLimitCount": 5,
  "errors": [],
  "exceededAmount": 1500.00,
  "exceededCount": 2,
  "insPolicyLimitTrackingIds": [
    "0qOxx0000004CvQEAU",
    "0qOxx0000004CvREAU"
  ],
  "isSuccess": true,
  "paidAmount": 5000.00
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
claimPaymentSummaryId	String	ID of the claim payment summary record that's added or updated. This ID links individual claim coverage payment details to the corresponding summary of payments that are processed for a claim.	Small, 66.0	66.0
consumedLimitCount	Integer	Total number of coverage units or limit counts that are utilized by this service or transaction.	Small, 66.0	66.0
errors	Error Detail[]	Error details that are shown when the API operation fails.	Small, 66.0	66.0
exceededAmount	Double	Excess amount that's paid above the limits.	Small, 66.0	66.0
exceededCount	Integer	Number of coverage units or counts that are paid above the policy limits. This occurs when payments exceed the allowable limits, such as in discretionary ex gratia payments.	Small, 66.0	66.0

Property Name	Type	Description	Filter Group and Version	Available Version
insPolicyLimitTrackingIds	String[]	IDs of the insurance policy limit tracking records that are created during the ex gratia payment process.	Small, 66.0	66.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 66.0	66.0
paidAmount	Double	Total amount that's paid towards ex gratia.	Small, 66.0	66.0

Process Policy Limits Response

Output representation for processing the policy limits for a claim loss item.

JSON example

```
{
  "isLimitExceeded": false,
  "insPolicyLimitTrackings": [
    {
      "policyLimitId": "01Yx00000004CvQEAU",
      "category": "Per-Incident Limit",
      "remainingAmount": 3750.00
    }
  ],
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail []	Errors that are shown when the API operation fails.	Big, 65.0	65.0
exceededAmount	Double	Amount that exceeds the remaining available policy limit in the specified limit currency.	Big, 66.0	66.0
exceededCount	Integer	Count of coverage units beyond the policy's allowed unit limit.	Big, 66.0	66.0
insPolicyLimitTrackings	Policy Limit Tracking Details []	List of insurance policy limit tracking details.	Big, 65.0	65.0
isLimitExceeded	Boolean	Indicates whether a limit is exceeded (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Verify Policy Coverage

Output representation of a policy coverage's applicability for a specified loss date under an insurance policy.

JSON example

```
{
  "isSuccess": true,
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail[]	List of errors, including detailed information about each one.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the policy coverage is valid for the date on which the incident occurred (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Insurance Claims Standard Invocable Actions

Learn more about the standard invocable actions available with Insurance Claims.

[Calculate Adjustments Action](#)

Calculate the adjusted amount for a loss item and generate data for insurance policy limit tracking and claim coverage payment adjustment records.

[Cancel Claim Coverage Payment Detail Action](#)

Cancel payment for a claim coverage payment detail record.

[Create Claim Action](#)

Create a claim.

[Create Claim Coverage Payment Detail Action](#)

Create a claim coverage payment detail record.

[Create Policy Limits Action](#)

Create insurance policy limit records for policy coverage and root attributes.

[Delete Claim Coverage Payment Detail Action](#)

Remove unpaid claim coverage payment details by updating the status of the claim coverage payment detail record to Deleted.

[Edit Claim Coverage Payment Detail Action](#)

Modify a claim coverage payment detail record.

[Get Claim Action](#)

Get the details of a claim.

[Get Insurance Policy Current Standings Action](#)

Get the current standings of the insurance policy.

[Open Claim Coverage Action](#)

Open a claim coverage.

[Pay Ex Gratia Claim Coverage Payment Detail Action](#)

Process ex gratia payments for the claim coverage payment detail record when the adjusted amount is greater than the policy limit amount or count or both.

[Pay Claim Coverage Payment Detail Action](#)

Initiate the payment for a claim coverage payment detail record, track the policy limit consumption for the payout, and update the status of the claim coverage payment detail record to Paid.

[Recalculate Adjustments Action](#)

Recalculate the adjusted amount for a claim loss item to the maximum amount that's allowed by the insurance policy.

[Update Claim Action](#)

Update a claim.

[Update Claim Coverage Action](#)

Update a claim coverage.

[Verify Policy Coverage Action](#)

Verify whether a policy coverage is applicable for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

Calculate Adjustments Action

Calculate the adjusted amount for a loss item and generate data for insurance policy limit tracking and claim coverage payment adjustment records.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/calculateAdjustments

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
calculateAdjustmentsInputRep	Type Apex-defined Description Required. An Apex ConnectApi.CalculateAdjustmentsInputRep record that contains the details of the claim for which the adjusted amount must be calculated.

Outputs

Output	Details
calculateAdjustmentsOutputRep	Type Apex-defined Description An Apex ConnectApi.CalculateAdjustmentsRepresentation record that contains the details of the adjustments for the claim loss items.

Example

Sample Request

```
{
  "inputs": [
    {
      "calculateAdjustmentsInputRep": {
        "claimId": "0ZkSB00000000tqv0AA",
        "claimCoverageId": "0kPSB0000000G72L2AS",
        "claimedAmount": "1000"
      }
    }
  ]
}
```

Sample Response

```
{
  "actionName": "calculateAdjustments",
  "errors": null,
  "invocationId": null,
  "isSuccess": true,
  "outcome": null,
  "outputValues": {
    "calculateAdjustmentsOutputRep": {
      "adjustedAmount": 1250.00,
      "adjustmentReason": "Deductible applied."
      "claimCovPaymentAdjustments": [
        {
          "adjustedAmount": 250.00,
          "adjustmentReason": "Deductible"
        }
      ],
      "insPolicyLimitTrackings": [
        {
          "policyLimitId": "01Yx00000004CvQEUAU",
          "category": "Per-Incident Limit",
          "remainingAmount": 4750.00
        }
      ]
    }
  }
}
```

```
{,
  "sortOrder": -1,
  "version": 1
}
```

Cancel Claim Coverage Payment Detail Action

Cancel payment for a claim coverage payment detail record.

When the `coveragePaymentType` is `Loss`, this action reverses the insurance policy limit tracking data to roll back consumption. This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/cancelClaimCvrPaymentDetail`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
<code>cancelClaimCvrPymtDtlInputRep</code>	Type Apex-defined Description Required. An Apex <code>ConnectApi.CancelClaimCvrPymtDtlInputRep</code> record that contains the details to cancel the payment for the claim coverage payment detail record.

Outputs

Output	Details
<code>paymentDetailId</code>	Type string Description ID of the payment detail record for which the payment is canceled.

Example

Sample Request

```
{
  "inputs": [
    {
      "cancelClaimCvrPymtDtlInputRep": {
        "claimId": "0clS700000004p9XIAQ",
        "coverageId": "0kPxx0000000002fEAA",
        "paymentDetailId": "012xx0000000003gEAA"
      }
    }
  ]
}
```

Sample Response

```
{
  "actionName": "cancelClaimCvrPaymentDetail",
  "errors": null,
  "invocationId": null,
  "isSuccess": true,
  "outcome": null,
  "outputValues": {
    "paymentDetailId": "5pdS700000004oVfIAI"
  },
  "sortOrder": -1,
  "version": 1
}
```

Create Claim Action

Create a claim.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/createClaim

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
createClaimInputRepresentation	Type Apex-defined Description Required. An Apex ConnectApi.CreateClaimInputRepresentation record that contains the details of the claim to be created.

Outputs

Output	Details
claimId	Type string Description ID of the claim that's created.

Example

Sample Request

```
{
  "inputs": [
    {
      "createClaimInputRepresentation": {
        "name": "Claim for Auto Accident",
        "lossDate": "2025-09-15",
        "claimType": "Auto",
        "incidentTypeId": "0itxx000000001dEAA",
        "claimReason": "Rear-end collision",
        "claimReasonType": "Accident",
        "lossType": "Accident",
        "insurancePolicyId": "0YTxx0000000001qEAA",
        "accountId": "001xx000003DGQyAAO",
        "additionalFields": {
          "key": "RepairShop",
          "value": "NY"
        },
      },
      "attributes": [
        {
          "apiName": "WeatherConditions",
          "value": "Clear and Sunny"
        },
        {
```

```

        "apiName": "NumberOfVehicles",
        "value": "2"
    }
],
"items": [
    {
        "instanceKey": "item_vehicle_damage_01",
        "name": "Damaged Rear Bumper and Trunk",
        "productCode": "AUTO-SEDAN-2023",
        "category": "Auto Physical Damage",
        "insuredItemId": "02ixx000000415sAAA",
        "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
        "attributes": [
            {
                "apiName": "DamageSeverity",
                "value": "Moderate"
            }
        ],
        "additionalFields": {
            "key": "RepairEstimate",
            "value": 2750
        },
        "relatedObjects": [
            {
                "relatedRecordId": "001xx000003DGQzAAP",
                "relatedRecordObjName": "Account"
            }
        ],
        "participantInstanceKey": "participant_driver_01"
    }
],
"participants": [
    {
        "instanceKey": "participant_driver_01",
        "contactId": "003xx000003DGQyAAO",
        "accountId": "001xx000003DGQyAAO",
        "insurancePolicyParticipantId": "0pPxx000000001qEAA",
        "isInjured": false,
        "roles": [
            "Driver",
            "Claimant"
        ],
        "additionalFields": {
            "StatementTaken": true
        }
    }
]
}
]
}
}

```

Sample Response

```
[
  {
    "actionName": "createClaim",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "claimId": "0Z1S70000004oVfIAI"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Create Claim Coverage Payment Detail Action

Create a claim coverage payment detail record.
This action is available in API version 65.0 and later.

Supported REST HTTP Methods

- URI**
/services/data/v/actions/standard/createClaimCvrPaymentDetail
- Formats**
JSON
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer *token*

Inputs

Input	Details
createClaimCvrPymtDtlInputRep	Type Apex-defined Description Required. An Apex ConnectApi.CreateClaimCvrPymtDtlInputRep record that contains the details to create the claim coverage payment detail record.

Outputs

Output	Details
createClaimCvrPymtDtlOutputRep	Type Apex-defined Description An Apex <code>ConnectApi.CreateClaimCvrPymtDtlOutputRep</code> record that contains the details of the claim coverage payment detail record and the generated data for the insurance policy limit tracking and claim coverage payment adjustment records.

Example

Sample Request

```
{
  "inputs": [
    {
      "createClaimCvrPymtDtlInputRep": {
        "claimId": "0Zkxx000000001dEAA",
        "claimCoverageId": "0kPxx000000002fEAA",
        "claimedAmount": 1200,
        "recipientId": "001xx000003GN2rAAG",
        "coveragePaymentType": "Loss",
        "benefitName": "DentalCov",
        "description": "Dental cleaning and checkup.",
        "limitUnitCount": 1,
        "adjustedAmount": 1200,
        "additionalFields": [
          {
            "key": "State",
            "value": "CA"
          },
          {
            "key": "LOB__c",
            "value": "Commercial"
          }
        ]
      }
    }
  ]
}
```

Sample Response

```
{
  "actionName": "createClaimCvrPaymentDetail",
  "errors": null,
  "invocationId": null,
  "isSuccess": true,
  "outcome": null,
}
```

```
"outputValues": {
  "createClaimCvrPymtDtlOutputRep": {
    "adjustedAmount": 200,
    "adjustedReason": "Deductible",
    "paymentDetailId": "01Dxx0000004CUmEAM",
    "insPolicyLimitTrackingIds": [
      "1TExx0000004DUuGAM",
      "1TExx0000004DUvGAM"
    ]
  }
},
"sortOrder": -1,
"version": 1
}
```

Create Policy Limits Action

Create insurance policy limit records for policy coverage and root attributes.
This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI
`/services/data/v/actions/standard/createPolicyLimits`

Formats
JSON

HTTP Methods
POST

Authentication
Authorization: Bearer *token*

Inputs

Input	Details
<code>createPolicyLimitsInputRep</code>	Type Apex-defined Description Required. An Apex ConnectApi.CreatePolicyLimitsInputRep record that contains the details to create the policy limits.

Outputs

Output	Details
createPolicyLimitsOutputRep	Type Apex-defined Description An Apex ConnectApi.CreatePolicyLimitsOutputRep record that contains the IDs for the insurance policy limit records that are created.

Example

Sample Request

```
{
  "inputs": [
    {
      "createPolicyLimitsInputRep": {
        "attributeScopes": [
          "Policy",
          "PolicyCoverage"
        ],
        "policyId": "0YTx0000000001qEAA"
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "createPolicyLimits",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "createPolicyLimitsOutputRep": {
        "policyLimitIds": [
          "1p1S700000004q2kIAA"
        ]
      }
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Delete Claim Coverage Payment Detail Action

Remove unpaid claim coverage payment details by updating the status of the claim coverage payment detail record to Deleted.

When the `coveragePaymentType` is Loss, this action reverses the insurance policy limit tracking data to roll back consumption. This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/deleteClaimCvrPaymentDetail`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
<code>deleteClaimCvrPymtDtlInputRep</code>	Type Apex-defined Description Required. An Apex ConnectApi.DeleteClaimCvrPymtDtlInputRep record that contains the details of the claim coverage payment detail record to be marked for deletion by updating the status.

Outputs

Output	Details
<code>paymentDetailId</code>	Type string Description ID of the payment detail record that's deleted.

Example

Sample Request

```
{
  "inputs": [
    {
      "deleteClaimCvrPymtDtlInputRep": {
        "claimId": "0Zkxx000000001dEAA",
        "coverageId": "0kPxx000000002fEAA",
        "paymentDetailId": "5pdS70000004oVfIAI"
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "deleteClaimCvrPaymentDetail",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "paymentDetailId": "5pdS70000004oVfIAI"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Edit Claim Coverage Payment Detail Action

Modify a claim coverage payment detail record.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/editClaimCvrPaymentDetail

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
editClaimCvrPymtDtlInputRep	Type Apex-defined Description Required. An Apex ConnectApi.EditClaimCvrPymtDtlInputRep record that contains details of the claim coverage payment detail to edit.

Outputs

Output	Details
editClaimCvrPymtDtlOutputRep	Type Apex-defined Description An Apex ConnectApi.EditClaimCvrPymtDtlOutputRep record that contains details of the claim coverage payment detail that's edited.

Example

POST

This sample request is for the Edit Claim Coverage Payment Detail action.

```
{
  "inputs": [
    {
      "editClaimCvrPymtDtlInputRep": {
        "claimId": "0Zkxx0000000001CAA",
        "claimCoverageId": "0kPxx0000000001CAA",
        "claimCoveragePaymentDetailId": "0l2xx0000000001CAA",
        "claimedAmount": 5000,
        "adjustedAmount": 5000,
        "recipientId": "001xx000003DGQyAAO",
        "name": "Rental Car Expense",
        "description": "Payment for 5 days of rental car usage",
        "limitUnitCount": 5,
        "status": "Draft",
        "additionalFields": [
          {
            "key": "State",
            "value": "CA"
          }
        ]
      }
    }
  ]
}
```

```

        "key": "LOB__c",
        "value": "Commercial"
      }
    ]
  }
}
]
}

```

This sample response is for the Edit Claim Coverage Payment Detail action.

```

{
  "actionName": "editClaimCvrPaymentDetail",
  "errors": null,
  "invocationId": null,
  "isSuccess": true,
  "outcome": null,
  "outputValues": {
    "editClaimCvrPyntDtlOutputRep": {
      "isSuccess": true,
      "adjustedAmount": 200,
      "adjustedReason": "Deductible",
      "insPolicyLimitTrackingIds": [
        "1TExx0000004DUuGAM",
        "1TExx0000004DUvGAM"
      ],
      "errors": []
    }
  },
  "sortOrder": -1,
  "version": 1
}

```

Get Claim Action

Get the details of a claim.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getClaim

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
getClaimInputRepresentation	Type Apex-defined Description Required. An Apex ConnectApi.GetClaimInputRepresentation record that contains the details of the claim to be retrieved.

Outputs

Output	Details
getClaimOutputRepresentation	Type Apex-defined Description An Apex ConnectApi.GetClaimOutputRepresentation record that contains the details of the claim that's retrieved.
claimItemOutputs	Type Apex-defined Description A collection of Apex ConnectApi.ClaimItemOutputs records that contain the list of claim items.
claimParticipantOutputs	Type Apex-defined Description A collection of Apex ConnectApi.ClaimParticipantOutputs records that contain the list of claim participants.

Example

Sample Request

```
{
  "inputs": [
    {
      "getClaimInputRepresentation": {
        "claimId": "0Z1S70000004oVfIAI"
      }
    }
  ]
}
```

```
]
}
```

Sample Response

```
[
  {
    "actionName": "getClaim",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "getClaimOutputRepresentation": {
        "claimDetails": {
          "accountId": "001xx000003DGQyAAO",
          "attributes": [
            {
              "apiName": "PoliceReportNumber",
              "value": "SF-2025-09-13-00123"
            }
          ],
          "claimReason": "Auto accident",
          "claimReasonType": "AUTO_ACCIDENT",
          "claimType": "AUTO_ACCIDENT",
          "lossType": "Front bumper damaged"
        },
        "claimId": "0Z1S70000004oVfIAI",
        "contextId": "C-000123",
        "insurancePolicyId": "0e9S70000004oVfIAI",
        "lossDate": "2025-07-22",
        "name": "Claim for Auto Accident",
        "productCode": "AUTO-ACCIDENT"
      },
      "claimItemOutputs": [
        {
          "claimItems": [
            {
              "category": "Auto Physical Damage",
              "claimDetails": {
                "accountId": "001xx000003DGQyAAO",
                "attributes": [
                  {
                    "apiName": "DamageSeverity",
                    "value": "Severe"
                  }
                ],
                "claimReason": "Auto accident",
                "claimReasonType": "AUTO_ACCIDENT",
                "claimType": "AUTO_ACCIDENT",
                "lossType": "Front bumper damaged"
              },
              "id": "0koxx000000001dEAA",
              "instanceKey": "item_vehicle_bumper_01",
              "name": "Damaged Front Bumper",
              "productCode": "AUTO-BUMPER-SEDAN",
              "insuredItemId": "02ixx000000415sAAA"
            }
          ]
        }
      ]
    }
  }
]
```

```

    }
  ]
}
],
"claimParticipantOutputs": [
  {
    "claimParticipants": [
      {
        "id": "0kaxx000000001dEAA",
        "instanceKey": "participant_driver_1",
        "accountId": "001xx000003DGQyAAO",
        "contactId": "003xx000003DGQyAAO",
        "insurancePolicyParticipantId": "0pPxx000000001qEAA",
        "roles": [
          "Driver",
          "Claimant"
        ]
      }
    ]
  }
],
"sortOrder": -1,
"version": 1
}
}
]

```

Get Insurance Policy Current Standings Action

Get the current standings of the insurance policy.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getInsPolicyCurrentStandings

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input

Details

getInsPlcyCurrStndInputRep

Type

Apex-defined

Input**Details****Description**

Required.

An Apex [ConnectApi.GetInsPlcyCurrStndInputRep](#) record that contains the details to retrieve the consumed and available limits for the policy or claim.

Outputs

Output**Details**

getInsPlcyCurrStndOutputRep

Type

Apex-defined

Description

An Apex [ConnectApi.GetInsPlcyCurrStndOutputRep](#) record that contains the details of the consumed and available limits that are retrieved for the policy claim.

Example

Sample Request

```
{
  "inputs": [
    {
      "getInsPlcyCurrStndInputRep": {
        "policyId": "01t000000000000001",
        "claimId": null,
        "policyParticipantId": null,
        "policyAssetId": null,
        "policyCoverageId": null,
        "claimCoverageId": null,
        "options": {
          "viewOption": "showAllPolicyCoverages",
          "shouldIncludePendingAmount": true
        }
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "getInsPolicyCurrentStandings",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
  }
]
```

```

"outputValues": {
  "getInsPlcyCurrStndOutputRep": [
    {
      "iplStandingCalcOutputNodes": [
        {
          "amounts": {
            "initialAmount": 500,
            "pendingAmount": 25,
            "predictedRemainingAmount": 325,
            "remainingAmount": 350,
            "totalUsedAmount": 150
          },
          "coverageInfo": {
            "claimCoverageId": "1TDSG0000007HU84AM",
            "claimId": "0cYSG000000UY1q2AG",
            "policyAssetId": "0cYSG000000UYzt2XY",
            "policyCoverageId": "0cYSG000000UY1q2AG",
            "policyParticipantId": "sdfbdsf976sff",
            "subCoverageName": "auto-parts"
          },
          "limitAttributes": {
            "applicableAction": [
              "PaymentAttempted"
            ],
            "durationType": "PolicyTerm",
            "category": "LimitCurrency",
            "policyLimitStandingKey": "PolicyCover LimitCurrency"
          },
          "mappedScopes": {
            "parentMappedScopeId": "a0MS70000004oVfIAI",
            "prodtAttrMappedScopeId": "a0MS70000004oVgIAI"
          },
          "values": {
            "initialValue": 10000,
            "pendingValue": 500,
            "predictedRemainingValue": 8000,
            "remainingValue": 8500,
            "totalUsedValue": 1500
          }
        }
      ]
    }
  ],
  "sortOrder": -1,
  "version": 1
}
]

```

Open Claim Coverage Action

Open a claim coverage.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/openClaimCoverage

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer ***token***

Inputs

Input	Details
openClaimCoverageInputRep	Type Apex-defined Description Required. An Apex ConnectApi.OpenClaimCoverageInputRep record that contains the details of the claim coverage to be opened.

Outputs

Output	Details
coverageId	Type string Description ID of the claim coverage that's opened.

Example

Sample Request

```
{
  "inputs": [
    {
      "openClaimCoverageInputRep": {
        "claimParticipantId": "0kaxx000000001dEAA",
        "claimItemId": "0koxx000000001dEAA",
        "insurancePolicyAssetId": "02ixx000000415sAAA",
        "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
        "reserveProcessingMode": "Manual",
        "currencyIsoCode": "USD",

```

```
        "lossReserveAmount": 5000,
        "expenseReserveAmount": 750.5,
        "additionalFields": {
            "RequireInspection": true,
            "CoverageSubtype": "Collision-Standard"
        }
    }
}
```

Sample Response

```
[
  {
    "actionName": "openClaimCoverage",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "coverageId": "1ccS70000004oVfIAI"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Pay Ex Gratia Claim Coverage Payment Detail Action

Process ex gratia payments for the claim coverage payment detail record when the adjusted amount is greater than the policy limit amount or count or both.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/payExGratiaClaimCvrPymtDetail

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
<code>payExGratiaClaimCvrPymtDtlInputRep</code>	Type Apex-defined Description Required. An Apex ConnectApi.PayExGratiaClaimCvrPymtDtlInputRep record that contains the details to process the ex gratia payment.

Outputs

Output	Details
<code>payExGratiaClaimCvrPymtDtlRep</code>	Type Apex-defined Description An Apex ConnectApi.PayExGratiaClaimCvrPymtDtlRep record that contains the ex gratia payment details.

Example

POST

This sample request is for the Pay Ex Gratia Claim Coverage Payment Detail action.

```
{
  "inputs": [
    {
      "payExGratiaClaimCvrPymtDtlInputRep": {
        "claimId": "0Zkxx0000000001CAA",
        "coverageId": "0kPxx0000000001CAA",
        "paymentDetailId": "0k5xx0000000001AAA",
        "reason": "Ex gratia payment approved for customer satisfaction."
      }
    }
  ]
}
```

This sample response is for the Pay Ex Gratia Claim Coverage Payment Detail action.

```
[
  {
    "actionName": "payExGratiaClaimCvrPymtDetail",
    "errors": null,
    "invocationId": null,
  }
]
```

```
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "payExGratiaClaimCvrPymtDtlRep": {
        "claimPaymentSummaryId": "0sDxx0000000001CAA",
        "consumedLimitCount": 5,
        "exceededAmount": 1500,
        "exceededCount": 2,
        "insPolicyLimitTrackingIds": [
          "0qOxx0000004CvQEAU",
          "0qOxx0000004CvREAU"
        ],
        "isSuccess": true,
        "paidAmount": 5000
      }
    },
    "sortOrder": -1,
    "version": 1
  }
}
```

Pay Claim Coverage Payment Detail Action

Initiate the payment for a claim coverage payment detail record, track the policy limit consumption for the payout, and update the status of the claim coverage payment detail record to Paid.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/payClaimCvrPaymentDetail

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
payClaimCvrPymtDtlInputRep	Type Apex-defined Description Required.

Input**Details**

An Apex ConnectApi.PayClaimCvrPymtDtlInputRep record that contains the details of the claim coverage payment detail to be paid.

Outputs

Output**Details**

payClaimCvrPymtDtlOutputRep

Type

Apex-defined

Description

An Apex ConnectApi.PayClaimCvrPymtDtlOutputRep record that contains the details of the payment status.

Example

Sample Request

```
{
  "inputs": [
    {
      "payClaimCvrPymtDtlInputRep": {
        "claimId": "0Zkxx000000001dEAA",
        "coverageId": "0kPxx000000002fEAA",
        "paymentDetailId": "5pdS70000004oVfIAI"
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "payClaimCvrPaymentDetail",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "payClaimCvrPymtDtlOutputRep": {
        "isSuccess": true
      }
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Recalculate Adjustments Action

Recalculate the adjusted amount for a claim loss item to the maximum amount that's allowed by the insurance policy.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/recalculateAdjustments

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
recalcAdjustmentsInputRep	Type Apex-defined Description Required. An Apex ConnectApi.RecalcAdjustmentsInputRep record that contains the claim amount and ID of the claim coverage payment detail record to recalculate the adjusted amount.

Outputs

Output	Details
calculateAdjustmentsOutputRep	Type Apex-defined Description An Apex ConnectApi.CalcAdjustmentsOutputRep record containing the calculated adjusted amount and the updated insurance policy limit tracking details.

Example

POST

This sample request is for the Recalculate Adjustments action.

```
{
  "inputs": [
    {
      "recalcAdjustmentsInputRep": {
        "claimId": "0Zkxx0000000001CAA",
        "claimCoverageId": "0kPxx0000000001CAA",
        "paymentDetailId": "0l2xx0000000001CAA",
        "claimedAmount": 5000
      }
    }
  ]
}
```

This sample response is for the Recalculate Adjustments action.

```
[
  {
    "actionName": "recalculateAdjustments",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "calculateAdjustmentsOutputRep": {
        "adjustedAmount": 1250,
        "adjustmentReason": "Deductible applied.",
        "claimCovPaymentAdjustments": [
          {
            "adjustedAmount": 250,
            "adjustmentReason": "Deductible"
          }
        ],
        "insPolicyLimitTrackings": [
          {
            "policyLimitId": "0lYx00000004CvQEAU",
            "category": "Per-Incident Limit",
            "remainingAmount": 4750
          }
        ]
      }
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Update Claim Action

Update a claim.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/updateClaim

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
updateClaimInputRepresentation	Type Apex-defined Description Required. An Apex ConnectApi.UpdateClaimInputRepresentation record that contains the details of the claim to be updated.

Outputs

Output	Details
claimId	Type string Description ID of the claim that's updated.

Example

Sample Request

```
{
  "inputs": [
    {
      "updateClaimInputRepresentation": {
        "calimId": "0Z1S70000004oVfIAI",
        "name": "Claim for Auto Accident",
        "lossDate": "2025-09-15",
```



```

"claimType": "Auto",
"incidentTypeId": "0itxx000000001dEAA",
"claimReason": "Rear-end collision",
"claimReasonType": "Accident",
"lossType": "Accident",
"insurancePolicyId": "0YTx000000001qEAA",
"accountId": "001xx000003DGQyAAO",
"additionalFields": {
  "key": "RepairShop",
  "value": "NY"
},
"attributes": [
  {
    "apiName": "WeatherConditions",
    "value": "Clear and Sunny"
  },
  {
    "apiName": "NumberOfVehicles",
    "value": "2"
  }
],
"items": [
  {
    "instanceKey": "item_vehicle_damage_01",
    "name": "Damaged Rear Bumper and Trunk",
    "productCode": "AUTO-SEDAN-2023",
    "category": "Auto Physical Damage",
    "insuredItemId": "02ixx000000415sAAA",
    "insurancePolicyCoverageId": "0cvxx000000AbCdEfG",
    "attributes": [
      {
        "apiName": "DamageSeverity",
        "value": "Moderate"
      }
    ],
    "additionalFields": {
      "RepairEstimate": 2750
    },
    "relatedObjects": [
      {
        "relatedRecordId": "001xx000003DGQzAAP",
        "relatedRecordObjName": "Account"
      }
    ],
    "participantInstanceKey": "participant_driver_01",
    "action": "update"
  }
],
"participants": [
  {
    "instanceKey": "participant_driver_01",
    "contactId": "003xx000003DGQyAAO",
    "accountId": "001xx000003DGQyAAO",
    "insurancePolicyParticipantId": "0pPxx000000001qEAA",

```

```
        "isInjured": false,
        "roles": [
            "Driver",
            "Claimant"
        ],
        "additionalFields": {
            "key": "StatementTaken",
            "value": true
        },
        "action": "update"
    }
}
]
```

Sample Response

```
[
  {
    "actionName": "createClaim",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "claimId": "0Z1S70000004oVfIAI"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Update Claim Coverage Action

Update a claim coverage.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/updateClaimCoverage

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
updateClaimCoverageInputRep	Type Apex-defined Description Required. An Apex ConnectApi.UpdateClaimCoverageInputRep record that contains the details of the claim coverage to be updated.

Outputs

Output	Details
coverageId	Type string Description ID of the claim coverage that's updated.

Example

Sample Request

```
{
  "inputs": [
    {
      "updateClaimCoverageInputRep": {
        "claimId": "0Z1S70000004oVfIAI",
        "coverageId": "1ccS70000004oVfIAI",
        "expenseReserveAmount": 500,
        "lossReserveAmount": 2000,
        "additionalFields": {
          "Status": "Approved"
        }
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "updateClaimCoverage",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
  }
]
```

```
[{"outcome": null,
  "outputValues": {
    "coverageId": "1ccS70000004oVfIAI"
  },
  "sortOrder": -1,
  "version": 1
}]
```

Verify Policy Coverage Action

Verify whether a policy coverage is applicable for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/verifyPolicyCoverage

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
verifyPolicyCoverageInputRep	Type Apex-defined
	Description Required. An Apex ConnectApi.VerifyPolicyCoverageInputRep record that contains the details of a policy coverage whose applicability on a specified loss date under an insurance policy must be verified.

Outputs

Output	Details
verifyPolicyCoverageOutputRep	Type Apex-defined

Output	Details
	Description An Apex ConnectApi.VerifyPolicyCoverageOutputRep record that contains the details of a policy coverage whose applicability on a specified loss date under an insurance policy is verified.

Example

Sample Request

```
{
  "inputs": [
    {
      "verifyPolicyCoverageInputRep": {
        "coverageId": "1pcS70000004oVfIAI",
        "insuredAssetOrParticipantId": "0kaxx000000001dEAA",
        "lossDate": "2025-08-15",
        "policyId": "0YTx0000000001qEAA"
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "verifyPolicyCoverage",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "verifyPolicyCoverageOutputRep": {
        "errors": [],
        "isSuccess": true
      }
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Insurance Claims Apex Reference

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. Insurance Claims Apex reference includes built-in Apex classes and interfaces, grouped by namespace.

[ConnectApi Namespace](#)

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Claim provides classes for accessing the same data available in the Insurance Policy Administration Business APIs.

ConnectApi Namespace

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Claim provides classes for accessing the same data available in the Insurance Policy Administration Business APIs.

For information about working with the `ConnectApi` classes, see [Connect in Apex](#).

[InsuranceClaimFamily Class](#)

Manage insurance claims.

[ConnectApi Input Classes](#)

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

[ConnectApi Output Classes](#)

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

InsuranceClaimFamily Class

Manage insurance claims.

Namespace

`ConnectApi`

[InsuranceClaimFamily Methods](#)

These methods are for `InsuranceClaimFamily`. All methods are static.

InsuranceClaimFamily Methods

These methods are for `InsuranceClaimFamily`. All methods are static.

[CalculateAdjustments\(CalculateAdjustmentsInput, claimId, coverageId\)](#)

Calculate adjustments for a loss item and determine the final payable amount for a specific claim loss item.

[CancelClaimCoveragePaymentDetail\(claimId, coverageId, paymentDetailId\)](#)

Cancel payment for a claim coverage payment detail record.

[CreateClaim\(CreateClaimInput\)](#)

Create a claim record.

[CreateClaimCoverage\(claimId, CreateClaimCoverageInput\)](#)

Create a claim coverage record.

[CreateClaimCoveragePaymentDetail\(CreateClaimCoveragePaymentDetailInput, claimId, coverageId\)](#)

Create a claim coverage payment detail record.

[CreatePolicyLimits\(policyId, CreatePolicyLimitsInput\)](#)

insurance policy limit records for the specified policy based on the defined attributes and coverages of the insurance policy.

[DeleteClaimCoveragePaymentDetail\(claimId, coverageId, paymentDetailId\)](#)

Delete a claim coverage payment detail record.

[EditClaimCoveragePaymentDetail\(EditClaimCoveragePaymentDetailInput, claimId, coverageId, paymentDetailId\)](#)

Modify a claim coverage payment detail record.

[GetClaim\(claimId\)](#)

Get the details of a specific claim.

[PayClaimCoveragePaymentDetail\(claimId, coverageId, paymentDetailId\)](#)

Process payment for a specific claim coverage payment detail record.

[PolicyLimitStandingCalculation\(InsurancePolicyLimitStandingCalculationInput\)](#)

Calculate the current standing for policy limits.

[postVerifyPolicyCoverage\(policyId, coverageId, verifyPolicyCoverageInput\)](#)

Verify the applicability of a policy coverage for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

[PayExGratiaClaimCvrPaymentDetail\(claimId, coverageId, paymentDetailId, PayExGratiaClaimCvrPaymentDetailInput\)](#)

Process an ex gratia payment for a claim coverage payment detail record.

[ProcessPolicyLimits\(ProcessPolicyLimitsInput, claimId, coverageId\)](#)

Process the policy limits for a loss item in a claim by using the current standings of the insurance policy limits.

[RecalculateAdjustments\(claimId, coverageId, RecalculateAdjustmentsInput\)](#)

Recalculate adjustments for existing payment details.

[UpdateClaim\(claimId, UpdateClaimInput\)](#)

Update the details of a specific claim.

[UpdateClaimCoverage\(claimId, coverageId, UpdateClaimCoverageInput\)](#)

Update the details of a specific coverage in a claim.

CalculateAdjustments(CalculateAdjustmentsInput, claimId, coverageId)

Calculate adjustments for a loss item and determine the final payable amount for a specific claim loss item.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateAdjustmentsRepresentation
CalculateAdjustments(ConnectApi.CalculateAdjustmentsInputRepresentation
CalculateAdjustmentsInput, String claimId, String coverageId)
```

Parameters

*CalculateAdjustmentsInput*Type: [ConnectApi.CalculateAdjustmentsInputRepresentation](#)

Details for calculating adjustments.

claimId

Type: String

ID of the claim for which the adjustments must be calculated.

coverageId

Type: String

ID of the claim coverage.

Return Value

Type: [ConnectApi.CalculateAdjustmentsRepresentation](#)

Example

Here's an example of how to invoke the Calculate Adjustments API by using Apex.

```

public with sharing class CalculateAdjustments implements System.Callable {
    public Object call(String action, Map<String, Object> args) {

        Map<String, Object> bodyMap = new Map<String, Object>();

        // Extract 'input' and 'output'
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        if (inputMap != null) {
            if (inputMap.containsKey('claimId') && inputMap.get('claimId') != null) {
                bodyMap.put('claimId', String.valueOf(inputMap.get('claimId')));
            }
            if (inputMap.containsKey('claimCoverageId') && inputMap.get('claimCoverageId')
!= null) {
                bodyMap.put('claimCoverageId',
String.valueOf(inputMap.get('claimCoverageId')));
            }
            if (inputMap.containsKey('claimedAmount') && inputMap.get('claimedAmount') !=
null) {
                bodyMap.put('claimedAmount', String.valueOf(inputMap.get('claimedAmount')));
            }
        } else {
            System.debug('No input map found in args');
        }

        String jsonBody = JSON.serialize(bodyMap);
        System.debug('Final JSON Body: ' + jsonBody);

        String str = postDataToExternalApi(bodyMap, jsonBody);
        return outputMap.put('API Response: ', str);
    }
}

```



```

    }

    public static String postDataToExternalApi(Map<String, Object> bodyMap, String jsonBody)
    {
        HttpRequest request = new HttpRequest();
        Http http = new Http();

        String claimId = (String) bodyMap.get('claimId');
        String claimCoverageId = (String) bodyMap.get('claimCoverageId');
        String claimedAmount = (String) bodyMap.get('claimedAmount');

        bodyMap.remove('claimId');
        bodyMap.remove('claimCoverageId');

        jsonBody = JSON.serialize(bodyMap);

        if (String.isBlank(claimId) || String.isBlank(claimCoverageId) ||
String.isBlank(claimedAmount)) {
            String errorMsg = 'Missing required fields - claimId: ' + claimId + ',
claimCoverageId: ' + claimCoverageId + ', claimedAmount: ' + claimedAmount;
            System.debug('Validation failed: ' + errorMsg);
            return 'Error: ' + errorMsg;
        }

        // Build endpoint dynamically
        String endpoint = 'callout:invokeUW/services/data/v65.0/connect/insurance/claims/'

                                + claimId + '/coverages/' + claimCoverageId +

'/calculate-adjustments';
        System.debug('>>> Dynamic Endpoint: ' + endpoint);

        request.setEndpoint(endpoint);
        request.setMethod('POST');
        request.setHeader('Content-Type', 'application/json');
        request.setTimeout(120000); // 2 mins
        request.setBody(jsonBody);

        try {
            HttpResponse response = http.send(request);

            System.debug('Status Code: ' + response.getStatusCode());
            System.debug('Response Body: ' + response.getBody());

            return response.getBody();
        } catch (Exception e) {
            System.debug('Exception: ' + e.getMessage());
            return 'Exception: ' + e.getMessage();
        }
    }

    // Sample test method
    public static void testCalculateAdjustments() {
        Map<String, Object> args = new Map<String, Object>();
        Map<String, Object> input = new Map<String, Object>();
    }

```

```

        Map<String, Object> output = new Map<String, Object>();

        input.put('claimId', '0Zkxx000000009hCAA');
        input.put('claimCoverageId', '0kPxx000000006TEAQ');
        input.put('claimedAmount', '1200.0');

        args.put('input', input);
        args.put('output', output);

        CalculateAdjustments test = new CalculateAdjustments();
        test.call('calculateAdjustments', args);
    }
}

```

CancelClaimCoveragePaymentDetail(claimId, coverageId, paymentDetailId)

Cancel payment for a claim coverage payment detail record.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.CancelClaimCoveragePaymentDetailRep
CancelClaimCoveragePaymentDetail(String claimId, String coverageId, String
paymentDetailId)

```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage record.

paymentDetailId

Type: String

ID of the claim coverage payment detail record.

Return Value

Type: [ConnectApi.CancelClaimCoveragePaymentDetailRep](#)

Example

Use this Apex sample to cancel a claim coverage payment detail record using the Cancel Claim Coverage Payment Detail API. This example shows how to invoke the Connect API method

`ConnectApi.InsuranceClaimFamily.CancelClaimCoveragePaymentDetail()` and handle success and error responses in Apex.

Class Definition

This Apex example shows how to cancel a specific claim coverage payment detail for a given claim and coverage.

```
public with sharing class CancelClaimCoveragePaymentDetailCallable implements
System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'cancelClaimCoveragePaymentDetail') {
            // Retrieve the required parameters from the input map
            String claimId = (String) inputMap.get('claimId');
            String claimCoverageId = (String) inputMap.get('claimCoverageId');
            String claimCoveragePaymentDetailId = (String)
inputMap.get('claimCoveragePaymentDetailId');

            System.debug('Start of connect API call to cancel claim coverage payment
detail');
            System.debug('Claim ID: ' + claimId);
            System.debug('Claim Coverage ID: ' + claimCoverageId);
            System.debug('Claim Coverage Payment Detail ID: ' +
claimCoveragePaymentDetailId);

            // Initialize the output representation
            ConnectApi.CancelClaimCoveragePaymentDetailRep paymentDetailData;

            try {
                // Call the Connect API resource using the correct method signature
                // Endpoint:
/connect/insurance/claims/{claimId}/coverages/{coverageId}/payment-details/{paymentDetailId}/cancelPayment

                paymentDetailData =
ConnectApi.InsuranceClaimFamily.cancelClaimCoveragePaymentDetail(
                    claimId,
                    claimCoverageId,
                    claimCoveragePaymentDetailId
                );

                System.debug('Claim coverage payment detail cancelled successfully: '
+ paymentDetailData);

                // Add results to output map
                outputMap.put('paymentDetailData', paymentDetailData);
                outputMap.put('isCancelled', paymentDetailData.isSuccess);
            }
        }
    }
}
```

```

        // Handle errors if present
        if (paymentDetailData.errors != null &&
!paymentDetailData.errors.isEmpty()) {
            outputMap.put('errors', paymentDetailData.errors);
            System.debug('Errors found: ' + paymentDetailData.errors);
        }
    } catch (Exception e) {
        System.debug('Error while canceling claim coverage payment detail: ' +
e.getMessage());
        outputMap.put('error', e.getMessage());
        outputMap.put('isCancelled', false);
    }

    return paymentDetailData;
}

// Return null if the action is unrecognized
return null;
}
}

```

Usage Example

Use the `CancelClaimCoveragePaymentDetailCallable` class to execute the `cancelClaimCoveragePaymentDetail` action and process the result.

```

CancelClaimCoveragePaymentDetailCallable callable = new
CancelClaimCoveragePaymentDetailCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();
input.put('claimId', '0ZkXK00000000010AA'); // Claim ID
input.put('claimCoverageId', '0kPXK00000000012AA'); // Claim Coverage ID
input.put('claimCoveragePaymentDetailId', '012XK0000000006TYAQ'); // Claim Coverage
Payment Detail ID

// Prepare output and argument maps
Map<String, Object> output = new Map<String, Object>();
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'cancelClaimCoveragePaymentDetail' action
Object result = callable.call('cancelClaimCoveragePaymentDetail', args);

// Check the result
if (result != null) {
    System.debug('Claim coverage payment detail processed successfully: ' + result);

    // Access the processed payment detail data
    ConnectApi.CancelClaimCoveragePaymentDetailRep paymentDetailData =
        (ConnectApi.CancelClaimCoveragePaymentDetailRep) result;

    // Process the result
}

```

```

    System.debug('Payment Detail Data: ' + paymentDetailData);

    if (paymentDetailData.isSuccess) {
        System.debug('Cancellation was successful');
    } else {
        System.debug('Cancellation failed');
        if (paymentDetailData.errors != null) {
            System.debug('Error details: ' + paymentDetailData.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

CreateClaim(CreateClaimInput)

Create a claim record.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.CreateClaimOutputRepresentation
CreateClaim(ConnectApi.ClaimInputRepresentation CreateClaimInput)

```

Parameters

CreateClaimInput

Type: [ConnectApi.ClaimInputRepresentation](#)

Details to create a claim.

Return Value

Type: [ConnectApi.CreateClaimOutputRepresentation](#)

Example

Use this Apex example to create a new insurance claim by using the Create Claim API. This example shows how to create a ClaimInputRepresentation with claim details, participants, and items, and then invoke the Connect API method `ConnectApi.ClaimFamily.CreateClaim()` to submit the claim.

Class Definition

This Apex sample shows how to use the Create Claim API from the Connect API namespace to create an insurance claim record. The example includes building nested input structures for claim details, participants, claim items, and related objects.

```
public with sharing class CreateClaimCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'createClaim') {
            // Retrieve the claim input data from the input map
            ConnectApi.ClaimInputRepresentation claimInput =
            (ConnectApi.ClaimInputRepresentation) inputMap.get('claimInput');
            System.debug('Start of connect API call to create claim');

            // Initialize the claim output representation
            ConnectApi.CreateClaimOutputRepresentation claimData;

            // Call the Connect API to create the claim
            try {
                claimData = ConnectApi.ClaimFamily.createClaim(claimInput);
                System.debug('Claim created successfully: ' + claimData);

                // Put the claim data in the output map
                outputMap.put('claimData', claimData);
            }
            catch (Exception e) {
                System.debug('Error while creating claim: ' + e.getMessage());
                outputMap.put('error', e.getMessage());
            }

            // Return the claim data
            return claimData;
        }

        // If the action is not recognized, return null
        return null;
    }
}
```

Usage Example

Use the CreateClaimCallable class to invoke the createClaim action and submit a new claim.

```
CreateClaimCallable callable = new CreateClaimCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Create and populate the ClaimInputRepresentation object
ConnectApi.ClaimInputRepresentation claimInput = new
```

```

ConnectApi.ClaimInputRepresentation();

// Set basic claim fields
claimInput.name = 'Claim-Demo-Test-apex-1';
claimInput.claimType = 'Auto';
claimInput.lossDate = DateTime.now();
claimInput.productCode = 'AutoClaimRoot';
claimInput.claimReasonType = 'Accident';
claimInput.claimReason = 'Damage';
claimInput.lossType = 'Partial Loss';
claimInput.insurancePolicyId = '0YTxx000000000iGAA';
claimInput.accountId = '001xx000003GYiEAAW';

// Set additional fields
Map<String, Object> additionalFields = new Map<String, Object>();
additionalFields.put('Severity', 'High');
claimInput.additionalFields = additionalFields;

// Set claim items
List<ConnectApi.ClaimItemInputRepresentation> itemList = new
List<ConnectApi.ClaimItemInputRepresentation>();

// Create first item
ConnectApi.ClaimItemInputRepresentation item1 = new
ConnectApi.ClaimItemInputRepresentation();
item1.name = 'item1';
item1.category = 'Involved Injury';
item1.productCode = 'InvolvedInjury';
item1.instanceKey = 'item1';
item1.insuredItemId = '0YWxx000000000NbGAI';
item1.insurancePolicyCoverageId = '0cYxx0000000125EAA';
item1.participantInstanceKey = 'par1';

// Set item attributes
List<ConnectApi.ClaimAttributeInputRepresentation> item1AttrList = new
List<ConnectApi.ClaimAttributeInputRepresentation>();

ConnectApi.ClaimAttributeInputRepresentation genderAttr = new
ConnectApi.ClaimAttributeInputRepresentation();
genderAttr.apiName = 'Gender';
genderAttr.value = 'Female';
item1AttrList.add(genderAttr);

item1.attributes = item1AttrList;

// Set item additional fields
Map<String, Object> item1AdditionalFields = new Map<String, Object>();
item1AdditionalFields.put('Description', 'test');
item1.additionalFields = item1AdditionalFields;

// Set item related objects
List<ConnectApi.ClaimRelatedObjectInputRepresentation> relatedObjList = new
List<ConnectApi.ClaimRelatedObjectInputRepresentation>();

```

```

ConnectApi.ClaimRelatedObjectInputRepresentation accountRel = new
ConnectApi.ClaimRelatedObjectInputRepresentation();
accountRel.recordId = '001xx000003GYiBAAW';
accountRel.recordObjName = 'Account';
relatedObjList.add(accountRel);

item1.relatedObjects = relatedObjList;

itemList.add(item1);
claimInput.items = itemList;

// Set claim participants
List<ConnectApi.ClaimParticipantInputRepresentation> participantList = new
List<ConnectApi.ClaimParticipantInputRepresentation>();

// Create first participant
ConnectApi.ClaimParticipantInputRepresentation participant1 = new
ConnectApi.ClaimParticipantInputRepresentation();
participant1.instanceKey = 'par1';
participant1.accountId = '001xx000003GYiEAAW';
participant1.contactId = '003xx000004WhFwAAK';
participant1.insurancePolicyParticipantId = '0aoxx00000000SQAAAY';
participant1.isInjured = 'true';
participant1.roles = new List<String>{'Claimant', 'Involved Driver'};

participantList.add(participant1);

// Create second participant
ConnectApi.ClaimParticipantInputRepresentation participant2 = new
ConnectApi.ClaimParticipantInputRepresentation();
participant2.instanceKey = 'par2';
participant2.contactId = '003xx000004WhFwAAK';
participant2.isInjured = 'true';
participant2.roles = new List<String>{'Claimant', 'Involved Driver'};

participantList.add(participant2);
claimInput.participants = participantList;

// Add the claimInput to the input map
input.put('claimInput', claimInput);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'createClaim' action
Object result = callable.call('createClaim', args);

// Retrieve the claim data from the result

```



```

ConnectApi.CreateClaimOutputRepresentation claimData =
    (ConnectApi.CreateClaimOutputRepresentation) result;

// Output the claim data to the debug log
System.debug('Claim Data: ' + JSON.serializePretty(claimData));

// Check for errors in the output map
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
}

```

CreateClaimCoverage(claimId, CreateClaimCoverageInput)

Create a claim coverage record.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.ClaimCoverageRepresentation CreateClaimCoverage(String claimId,
    ConnectApi.CreateClaimCoverageInputRepresentation CreateClaimCoverageInput)

```

Parameters

claimId

Type: String

ID of the claim to add coverage to.

CreateClaimCoverageInput

Type: [ConnectApi.CreateClaimCoverageInputRepresentation](#)

Object that contains the details to create the claim coverage.

Return Value

Type: [ConnectApi.ClaimCoverageRepresentation](#)

Example

The Apex example shows how to use the `ConnectApi.InsuranceClaimFamily.CreateClaimCoverage()` method to create a claim coverage for an existing claim.

Class Definition

```

public with sharing class CreateClaimCoverageCallable implements System.Callable {

```

```

public Object call(String action, Map<String, Object> args) {
    // Retrieve the input and output maps from the arguments
    Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
    Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

    // Check the action type
    if (action == 'createClaimCoverage') {

        // Retrieve the coverage input data from the input map
        ConnectApi.CreateClaimCoverageInputRepresentation coverageInput =
            (ConnectApi.CreateClaimCoverageInputRepresentation)
            inputMap.get('coverageInput');

        System.debug('Start of Connect API call to create claim coverage');

        // Initialize the coverage output representation
        ConnectApi.ClaimCoverageRepresentation coverageData;

        try {
            // Call the Connect API to create the claim coverage
            // Replace the claim ID with the appropriate value for your org
            coverageData = ConnectApi.ClaimFamily.createClaimCoverage(
                '0Zkxx00000000JNCAY', // Claim ID
                coverageInput
            );

            System.debug('Claim coverage created successfully: ' + coverageData);

            // Put the coverage data in the output map
            outputMap.put('coverageData', coverageData);
        } catch (Exception e) {
            System.debug('Error while creating claim coverage: ' + e.getMessage());

            outputMap.put('error', e.getMessage());
        }

        // Return the coverage data
        return coverageData;
    }

    // If the action is not recognized, return null
    return null;
}

```

Usage Example

This example shows how to call the `CreateClaimCoverageCallable` class and include the required input data.

```

CreateClaimCoverageCallable callable = new CreateClaimCoverageCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Create and populate the CreateClaimCoverageInputRepresentation object

```

```

ConnectApi.CreateClaimCoverageInputRepresentation coverageInput =
    new ConnectApi.CreateClaimCoverageInputRepresentation();

// Set required fields
coverageInput.claimParticipantId = '0aSxx00000000CvEAI';
coverageInput.claimItemId = '0dqxx00000000McAAI';
coverageInput.insurancePolicyAssetId = '0YWxx00000000P5GAI';
coverageInput.insurancePolicyCoverageId = '0cYxx000000012GEAQ';
coverageInput.name = 'Apex Cov Test';

// Set optional fields
coverageInput.lossReserveAmount = 25000.00;
coverageInput.expenseReserveAmount = 2500.00;
coverageInput.reserveProcessingMode = 'CoverageReserve';
coverageInput.currencyIsoCode = 'USD';

// Set additional fields
Map<String, Object> additionalFields = new Map<String, Object>();
additionalFields.put('Description', 'Coverage for vehicle damage claim');
coverageInput.additionalFields = additionalFields;

// Add the coverageInput to the input map
input.put('coverageInput', coverageInput);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'createClaimCoverage' action
Object result = callable.call('createClaimCoverage', args);

// Retrieve the coverage data from the result
ConnectApi.ClaimCoverageRepresentation coverageData =
    (ConnectApi.ClaimCoverageRepresentation) result;

// Log the created coverage data
System.debug('Coverage Data: ' + JSON.serializePretty(coverageData));

// Check for errors in the output map
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
}

```

CreateClaimCoveragePaymentDetail(CreateClaimCoveragePaymentDetailInput, claimId, coverageId)

Create a claim coverage payment detail record.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.CreateClaimCoveragePaymentDetailRep  
CreateClaimCoveragePaymentDetail (ConnectApi.CreateClaimCovPaymentDetailInputRep  
CreateClaimCoveragePaymentDetailInput, String claimId, String coverageId)
```

Parameters

CreateClaimCoveragePaymentDetailInput

Type: [ConnectApi.CreateClaimCovPaymentDetailInputRep](#)

Representation details to create the claim coverage payment detail record.

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage.

Return Value

Type: [ConnectApi.CreateClaimCoveragePaymentDetailRep](#)

Example

Here's an example of how to invoke the Create CCPD API by using Apex.

```
public with sharing class CreateCCPDIPTest3 implements System.Callable {  
    public Object call(String action, Map<String, Object> args) {  
        System.debug('>>> Action: ' + action);  
        System.debug('>>> Args: ' + JSON.serializePretty(args));  
  
        Map<String, Object> bodyMap = new Map<String, Object>();  
  
        // Extract 'input' and 'output'  
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');  
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');  
  
        if (inputMap != null) {  
            if (inputMap.containsKey('claimId') && inputMap.get('claimId') != null) {  
                bodyMap.put('claimId', String.valueOf(inputMap.get('claimId')));  
            }  
            if (inputMap.containsKey('claimCoverageId') && inputMap.get('claimCoverageId')
```

```

    != null) {
        bodyMap.put('claimCoverageId',
String.valueOf(inputMap.get('claimCoverageId')));
    }
    if (inputMap.containsKey('claimedAmount') && inputMap.get('claimedAmount') !=
null) {
        bodyMap.put('claimedAmount', String.valueOf(inputMap.get('claimedAmount')));
    }
    if (inputMap.containsKey('coveragePaymentType') &&
inputMap.get('coveragePaymentType') != null) {
        bodyMap.put('coveragePaymentType',
String.valueOf(inputMap.get('coveragePaymentType')));
    }
    if (inputMap.containsKey('adjustedAmount') && inputMap.get('adjustedAmount')
!= null) {
        bodyMap.put('adjustedAmount',
String.valueOf(inputMap.get('adjustedAmount')));
    }
    } else {
        System.debug('[] No input map found in args');
    }

    String jsonBody = JSON.serialize(bodyMap);
    System.debug('>>> Final JSON Body: ' + jsonBody);

    String str = postDataToExternalApi(bodyMap, jsonBody);
    return outputMap.put('response', str);
}

public static String postDataToExternalApi(Map<String, Object> bodyMap, String jsonBody)
{
    HttpRequest request = new HttpRequest();
    Http http = new Http();

    String claimId = (String) bodyMap.get('claimId');
    String claimCoverageId = (String) bodyMap.get('claimCoverageId');

    bodyMap.remove('claimId');
    bodyMap.remove('claimCoverageId');

    jsonBody = JSON.serialize(bodyMap);
    System.debug('>>> Final JSON Body: ' + jsonBody);

    if (String.isBlank(claimId) || String.isBlank(claimCoverageId)) {
        String errorMsg = 'Missing required fields - claimId: ' + claimId + ',
claimCoverageId: ' + claimCoverageId;
        System.debug('[] ' + errorMsg);
        return 'Error: ' + errorMsg;
    }

    // [] Build endpoint dynamically
    String endpoint = 'callout:invokeUW/services/data/v65.0/connect/insurance/claims/'

```

```

        + claimId + '/coverages/' + claimCoverageId + '/payment-details';

System.debug('>>> Dynamic Endpoint: ' + endpoint);

request.setEndpoint(endpoint);
request.setMethod('POST');
request.setHeader('Content-Type', 'application/json');
request.setTimeout(120000); // 2 mins
request.setBody(jsonBody);

try {
    HttpResponse response = http.send(request);

    System.debug('>>> Status Code: ' + response.getStatusCode());
    System.debug('>>> Response Body: ' + response.getBody());

    return response.getBody();
} catch (Exception e) {
    System.debug('❌ Exception: ' + e.getMessage());
    return 'Exception: ' + e.getMessage();
}

}

// Sample test method
public static void testCreateCCPD() {
    Map<String, Object> args = new Map<String, Object>();
    Map<String, Object> input = new Map<String, Object>();
    Map<String, Object> output = new Map<String, Object>();

    input.put('claimId', '0Zkxx000000009hCAA');
    input.put('claimCoverageId', '0kPxx000000006TEAQ');
    input.put('claimedAmount', '1200.0');
    input.put('coveragePaymentType', 'Loss');
    input.put('adjustedAmount', '1200.0');

    args.put('input', input);
    args.put('output', output);

    CreateCCPDIPTest1 test = new CreateCCPDIPTest1();
    test.call('createCCPD', args);
}
}

```

CreatePolicyLimits(policyId, CreatePolicyLimitsInput)

insurance policy limit records for the specified policy based on the defined attributes and coverages of the insurance policy.

Version

64.0

Requires

No

Signature

```
public static ConnectApi.CreatePolicyLimitsRepresentation CreatePolicyLimits(String
policyId, ConnectApi.CreatePolicyLimitsInputRepresentation CreatePolicyLimitsInput)
```

Parameters

policyId

Type: String

of the policy to generate limits for.

CreatePolicyLimitsInput

Type: [ConnectApi.CreatePolicyLimitsInputRepresentation](#)

that specifies the attribute scopes for which the policy limit records be created.

Return Value

Type: [ConnectApi.CreatePolicyLimitsRepresentation](#)

Example

This Apex example shows how to call the `CreatePolicyLimits` method to automatically generate limit tracking records that are associated with a given policy ID.

Class Definition

```
public with sharing class CreatePolicyLimitsCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'createPolicyLimits') {
            // Retrieve the policy ID from the input map
            String policyId = (String) inputMap.get('policyId');

            System.debug('Start of connect API call to create policy limits');
            System.debug('Policy ID: ' + policyId);

            // Initialize the output representation
            ConnectApi.CreatePolicyLimitsRepresentation policyLimitsData;

            // Call the Connect API to create policy limits
            try {
                // Call the Connect API resource directly using the correct method
signature
                policyLimitsData =
ConnectApi.InsuranceClaimFamily.createPolicyLimits(policyId);

                System.debug('Policy limits created successfully: ' + policyLimitsData);
            } catch (Exception e) {
                System.debug('Error creating policy limits: ' + e.getMessage());
            }
        }
    }
}
```

```

        // Put the policy limits data in the output map
        outputMap.put('policyLimitsData', policyLimitsData);
    }
    catch (Exception e) {
        System.debug('Error while creating policy limits: ' + e.getMessage());
        outputMap.put('error', e.getMessage());
    }

    // Return the policy limits data
    return policyLimitsData;
}

// If the action is not recognized, return null
return null;
}
}

```

Usage Example

```

CreatePolicyLimitsCallable callable = new CreatePolicyLimitsCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Set the policy ID (required)
input.put('policyId', '0YTx000000001LGAQ');

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'createPolicyLimits' action
Object result = callable.call('createPolicyLimits', args);

// Check the result
if (result != null) {
    System.debug('Policy limits created successfully: ' + result);

    // Access the created policy limits data
    ConnectApi.CreatePolicyLimitsRepresentation policyLimitsData =
        (ConnectApi.CreatePolicyLimitsRepresentation) result;

    // Process the result as needed
    System.debug('Policy Limits Data: ' + policyLimitsData);

    // Access the generated policy limit IDs
    if (policyLimitsData.policyLimitIds != null) {
        System.debug('Generated Policy Limit IDs: ' + policyLimitsData.policyLimitIds);
    }
}

```



```

        System.debug('Number of limits generated: ' +
policyLimitsData.policyLimitIds.size());
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

DeleteClaimCoveragePaymentDetail(claimId, coverageId, paymentDetailId)

Delete a claim coverage payment detail record.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.DeleteClaimCoveragePaymentDetailRep
DeleteClaimCoveragePaymentDetail(String claimId, String coverageId, String
paymentDetailId)

```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage record.

paymentDetailId

Type: String

ID of the claim coverage payment detail record.

Return Value

Type: [ConnectApi.DeleteClaimCoveragePaymentDetailRep](#)

Example

Use the `DeleteClaimCoveragePaymentDetailCallable` Apex class to delete an existing claim coverage payment detail record for a specified claim and coverage.

Class Definition

```

public with sharing class DeleteClaimCoveragePaymentDetailCallable implements

```

```

System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'deleteClaimCoveragePaymentDetail') {
            // Retrieve the required parameters from the input map
            String claimId = (String) inputMap.get('claimId');
            String claimCoverageId = (String) inputMap.get('claimCoverageId');
            String claimCoveragePaymentDetailId = (String)
inputMap.get('claimCoveragePaymentDetailId');

            System.debug('Start of connect API call to delete claim coverage payment
detail');

            System.debug('Claim ID: ' + claimId);
            System.debug('Claim Coverage ID: ' + claimCoverageId);
            System.debug('Claim Coverage Payment Detail ID: ' +
claimCoveragePaymentDetailId);

            // Initialize the output representation
            ConnectApi.DeleteClaimCoveragePaymentDetailRep paymentDetailData;

            // Call the Connect API to delete claim coverage payment detail
            try {
                // Call the Connect API resource directly using the correct method
signature
                // The endpoint is:
                /connect/insurance/claims/${claimId}/coverages/${coverageId}/payment-details/${paymentDetailId}/delete

                paymentDetailData =
ConnectApi.InsuranceClaimFamily.deleteClaimCoveragePaymentDetail(
                    claimId,
                    claimCoverageId,
                    claimCoveragePaymentDetailId
                );

                System.debug('Claim coverage payment detail deleted successfully: ' +
paymentDetailData);

                // Put the payment detail data in the output map
                outputMap.put('paymentDetailData', paymentDetailData);
                outputMap.put('isDeleted', paymentDetailData.isSuccess);

                // Check if there are any errors
                if (paymentDetailData.errors != null &&
!paymentDetailData.errors.isEmpty()) {
                    outputMap.put('errors', paymentDetailData.errors);
                    System.debug('Errors found: ' + paymentDetailData.errors);
                }
            }
            catch (Exception e) {

```

```

        System.debug('Error while deleting claim coverage payment detail: ' +
e.getMessage());
        outputMap.put('error', e.getMessage());
        outputMap.put('isDeleted', false);
    }

    // Return the payment detail data
    return paymentDetailData;
}

// If the action is not recognized, return null
return null;
}
}

```

Usage Example

```

DeleteClaimCoveragePaymentDetailCallable callable = new
DeleteClaimCoveragePaymentDetailCallable();
// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();
// Set the required parameters
input.put('claimId', '0ZkXK00000000010AA'); // Claim ID
input.put('claimCoverageId', '0kPXX00000000012AA'); // Claim Coverage ID
input.put('claimCoveragePaymentDetailId', '012XK0000000006TYAQ'); // Claim Coverage
Payment Detail ID
// Prepare the output map
Map<String, Object> output = new Map<String, Object>();
// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);
// Call the 'deleteClaimCoveragePaymentDetail' action
Object result = callable.call('deleteClaimCoveragePaymentDetail', args);
// Check the result
if (result != null) {
    System.debug('Claim coverage payment detail processed successfully: ' + result);

    // Access the processed payment detail data
    ConnectApi.DeleteClaimCoveragePaymentDetailRep paymentDetailData =
        (ConnectApi.DeleteClaimCoveragePaymentDetailRep) result;

    // Process the result as needed
    System.debug('Payment Detail Data: ' + paymentDetailData);

    // Check if the payment was successful
    if (paymentDetailData.isSuccess) {
        System.debug('Deletion was successful');
    } else {
        System.debug('Deletion failed');
        if (paymentDetailData.errors != null) {
            System.debug('Error details: ' + paymentDetailData.errors);
        }
    }
}
}

```

```

    } else {
        System.debug('Error occurred: ' + output.get('error'));
    }
}

```

EditClaimCoveragePaymentDetail(EditClaimCoveragePaymentDetailInput, claimId, coverageId, paymentDetailId)

Modify a claim coverage payment detail record.

API Version

66.0

Requires Chatter

No

Signature

```

public static ConnectApi.EditClaimCoveragePaymentDetailRep
EditClaimCoveragePaymentDetail(ConnectApi.EditClaimCvrPaymentDetailInputRep
EditClaimCoveragePaymentDetailInput, String claimId, String coverageId, String
paymentDetailId)

```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the coverage that applies to the claim.

EditClaimCoveragePaymentDetailInput

Type: [ConnectApi.EditClaimCvrPaymentDetailInputRep](#)

Details to edit the claim coverage payment detail.

paymentDetailId

Type: String

ID of the claim coverage payment detail.

Return Value

Type: [ConnectApi.EditClaimCoveragePaymentDetailRep](#)

Example

Class Definition

This is a sample Apex code to define the `EditClaimCoveragePaymentDetail` class before you can call it from Apex to edit a claim coverage payment detail.

```
public with sharing class EditClaimCoveragePaymentDetailCallable implements
System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'editClaimCoveragePaymentDetail') {
            // Retrieve the required parameters from the input map
            String claimId = (String) inputMap.get('claimId');
            String claimCoverageId = (String) inputMap.get('claimCoverageId');
            String paymentDetailId = (String) inputMap.get('paymentDetailId');

            ConnectApi.EditClaimCvrPaymentDetailInputRep editInputRep =
                (ConnectApi.EditClaimCvrPaymentDetailInputRep)
inputMap.get('EditClaimCoveragePaymentDetailInput');

            System.debug('Start of connect API call to edit claim coverage payment
detail');

            System.debug('Claim ID: ' + claimId);
            System.debug('Claim Coverage ID: ' + claimCoverageId);
            System.debug('Payment Detail ID: ' + paymentDetailId);

            // Initialize the output representation
            ConnectApi.EditClaimCoveragePaymentDetailRep editResponse;

            // Call the Connect API to edit claim coverage payment detail
            try {
                // Call the Connect API resource directly
                // The endpoint is:
/connect/insurance/claims/${claimId}/coverages/${coverageId}/payment-details/${paymentDetailId}

                editResponse =
ConnectApi.InsuranceClaimFamily.EditClaimCoveragePaymentDetail(
                    editInputRep,
                    claimId,
                    claimCoverageId,
                    paymentDetailId
                );

                System.debug('Edit claim coverage payment detail completed successfully:
' + editResponse);

                // Put the editResponse in the output map
                outputMap.put('editResponse', editResponse);
                outputMap.put('isSuccess', editResponse.isSuccess);
            }
        }
    }
}
```

```

        // Check if there are any errors
        if (editResponse.errors != null && !editResponse.errors.isEmpty()) {
            outputMap.put('errors', editResponse.errors);
            System.debug('Errors found: ' + editResponse.errors);
        }
    }
    catch (Exception e) {
        System.debug('Error while editing claim coverage payment detail: ' +
e.getMessage());
        outputMap.put('error', e.getMessage());
        outputMap.put('isSuccess', false);
    }

    return editResponse;
}

// If the action is not recognized, return null
return null;
}
}

```

Usage Example

This example shows how to call the `EditClaimCoveragePaymentDetailCallable` class and include the required input data.

```

PayExGratiaClaimCvrPaymentDetailCallable callable = new
PayExGratiaClaimCvrPaymentDetailCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Set the required parameters
input.put('claimId', '0ZkSB0000001AVH0A2'); // Claim ID
input.put('claimCoverageId', '0kPSB000000GHdu2AG'); // Claim Coverage ID
input.put('paymentDetailId', '012SB000000CLsMYAW'); // Payment Detail ID

// Create the input representation object
// First create a Map with the properties
Map<String, Object> payExGratiaInputMap = new Map<String, Object>();
payExGratiaInputMap.put('reason', 'Customer service goodwill payment');

// Convert the Map to ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep using
JSON.deserialize
String jsonString = JSON.serialize(payExGratiaInputMap);
ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep payExGratiaInputRep =
    (ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep) JSON.deserialize(jsonString,
    ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep.class);

input.put('PayExGratiaClaimCvrPaymentDetailInput', payExGratiaInputRep);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

```

```

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'payExGratiaClaimCvrPaymentDetail' action
Object result = callable.call('payExGratiaClaimCvrPaymentDetail', args);

// Check the result
if (result != null) {
    System.debug('payExGratiaClaimCvrPaymentDetail processed successfully: ' + result);

    // Access the processed payment data
    ConnectApi.PayExGratiaClaimCvrPaymentDetailRep paymentData =
        (ConnectApi.PayExGratiaClaimCvrPaymentDetailRep) result;

    // Process the result as needed
    System.debug('Pay Ex Gratia Payment Data: ' + paymentData);

    // Check if the payment was successful
    if (paymentData.isSuccess) {
        System.debug('Pay Ex Gratia was successful');
        System.debug('Paid Amount: ' + paymentData.paidAmount);
        System.debug('Claim Payment Summary ID: ' + paymentData.claimPaymentSummaryId);

        System.debug('Consumed Limit Count: ' + paymentData.consumedLimitCount);
        System.debug('Exceeded Amount: ' + paymentData.exceededAmount);
        System.debug('Exceeded Count: ' + paymentData.exceededCount);

        if (paymentData.insPolicyLimitTrackingIds != null &&
!paymentData.insPolicyLimitTrackingIds.isEmpty()) {
            System.debug('Insurance Policy Limit Tracking IDs: ' +
paymentData.insPolicyLimitTrackingIds);
        }
    } else {
        System.debug('Pay Ex Gratia failed');
        if (paymentData.errors != null) {
            System.debug('Error details: ' + paymentData.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

GetClaim(claimId)

Get the details of a specific claim.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.ClaimOutputRepresentation GetClaim(String claimId)
```

Parameters

claimId

Type: String

ID of the claim to retrieve the details for.

Return Value

Type: [ConnectApi.ClaimOutputRepresentation](#)

Example

Use this Apex example to retrieve claim details from the Claim Management system by using the Get Claim API. The example shows how to invoke the Connect API method `ConnectApi.ClaimFamily.GetClaim()` and handle responses and errors in Apex.

Class Definition

Here's an example that shows how to invoke the Get Claim API by using Apex.

```
// Output the claim data to the debug log
System.debug('Claim Data: ' + JSON.serializePretty(claimData));
g action, Map<String, Object> args) {
    // Retrieve the input and output maps from the arguments
    Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
    Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

    // Check the action type
    if (action == 'getClaim') {
        // Retrieve the claim ID from the input map
        String claimId = (String) inputMap.get('claimId');
        System.debug('Start of connect API call to get claim with ID: ' + claimId);

        // Initialize the claim output representation
        ConnectApi.ClaimOutputRepresentation claimData;

        // Call the Connect API to get the claim
        try {
            claimData = ConnectApi.ClaimFamily.getClaim(claimId);
            System.debug('Claim retrieved successfully: ' + claimData);

            // Put the claim data in the output map
            outputMap.put('claimData', claimData);
        }
        catch (Exception e) {
            System.debug('Error while retrieving claim: ' + e.getMessage());
        }
    }
}
```



```

        outputMap.put('error', e.getMessage());
    }

    // Return the claim data
    return claimData;
}

// If the action is not recognized, return null
return null;
}
}

```

Usage Example

You can use the `GetClaimCallable` class in Apex to call the `getClaim` action and retrieve claim details by the claim ID.

```

GetClaimCallable callable = new GetClaimCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Get claimId from user input (you can modify this to get from different sources)
// Option 1: From URL parameters (if called from a Visualforce page or Lightning
// component)
String claimId = ApexPages.currentPage().getParameters().get('claimId');

if (String.isBlank(claimId)) {
    System.debug('Please provide a claimId. You can set it in the input map.');
```

PayClaimCoveragePaymentDetail(claimId, coverageId, paymentDetailId)

Process payment for a specific claim coverage payment detail record.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.PayClaimCoveragePaymentDetailRep  
PayClaimCoveragePaymentDetail(String claimId, String coverageId, String paymentDetailId)
```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage.

paymentDetailId

Type: String

ID of the claim coverage payment detail to process.

Return Value

Type: [ConnectApi.PayClaimCoveragePaymentDetailRep](#)

Example

Use the `PayClaimCoveragePaymentDetailCallable` Apex class to trigger payment for a claim coverage payment detail record in an insurance claim. This class serves as an Apex wrapper for the Connect API endpoint:

`/connect/insurance/claims/{claimId}/coverages/{coverageId}/payment-details/{paymentDetailId}/pay`. You can use this method to automate payment operations directly from Apex.

Class Definition

```
public with sharing class PayClaimCoveragePaymentDetailCallable implements System.Callable  
{  
  
    public Object call(String action, Map<String, Object> args) {  
        // Retrieve the input and output maps from the arguments  
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');  
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');  
    }  
}
```

```

// Check the action type
if (action == 'payClaimCoveragePaymentDetail') {
    // Retrieve the required parameters from the input map
    String claimId = (String) inputMap.get('claimId');
    String claimCoverageId = (String) inputMap.get('claimCoverageId');
    String claimCoveragePaymentDetailId = (String)
inputMap.get('claimCoveragePaymentDetailId');

    System.debug('Start of connect API call to pay claim coverage payment
detail');
    System.debug('Claim ID: ' + claimId);
    System.debug('Claim Coverage ID: ' + claimCoverageId);
    System.debug('Claim Coverage Payment Detail ID: ' +
claimCoveragePaymentDetailId);

    // Initialize the output representation
    ConnectApi.PayClaimCoveragePaymentDetailRep paymentDetailData;

    // Call the Connect API to pay claim coverage payment detail
    try {
        // Call the Connect API resource directly using the correct method
signature
        // The endpoint is:
/connect/insurance/claims/${claimId}/coverages/${coverageId}/payment-details/${paymentDetailId}/pay

        paymentDetailData =
ConnectApi.InsuranceClaimFamily.payClaimCoveragePaymentDetail(
            claimId,
            claimCoverageId,
            claimCoveragePaymentDetailId
        );

        System.debug('Claim coverage payment detail paid successfully: ' +
paymentDetailData);

        // Put the payment detail data in the output map
        outputMap.put('paymentDetailData', paymentDetailData);
        outputMap.put('isPaid', paymentDetailData.isSuccess);

        // Check if there are any errors
        if (paymentDetailData.errors != null &&
!paymentDetailData.errors.isEmpty()) {
            outputMap.put('errors', paymentDetailData.errors);
            System.debug('Errors found: ' + paymentDetailData.errors);
        }
    }
    catch (Exception e) {
        System.debug('Error while paying claim coverage payment detail: ' +
e.getMessage());
        outputMap.put('error', e.getMessage());
        outputMap.put('isPaid', false);
    }

    // Return the payment detail data

```

```

        return paymentDetailData;
    }

    // If the action is not recognized, return null
    return null;
}
}

// Usage Example:
PayClaimCoveragePaymentDetailCallable callable = new
PayClaimCoveragePaymentDetailCallable();
// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();
// Set the required parameters
input.put('claimId', '0Zkxx000000002ZCAQ'); // Claim ID
input.put('claimCoverageId', '0kPxx000000000cEAA'); // Claim Coverage ID
input.put('claimCoveragePaymentDetailId', '012xx000000002cAAA'); // Claim Coverage
Payment Detail ID
// Prepare the output map
Map<String, Object> output = new Map<String, Object>();
// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);
// Call the 'payClaimCoveragePaymentDetail' action
Object result = callable.call('payClaimCoveragePaymentDetail', args);
// Check the result
if (result != null) {
    System.debug('Claim coverage payment detail processed successfully: ' + result);

    // Access the processed payment detail data
    ConnectApi.PayClaimCoveragePaymentDetailRep paymentDetailData =
        (ConnectApi.PayClaimCoveragePaymentDetailRep) result;

    // Process the result as needed
    System.debug('Payment Detail Data: ' + paymentDetailData);

    // Check if the payment was successful
    if (paymentDetailData.isSuccess) {
        System.debug('Payment was successful');
    } else {
        System.debug('Payment failed');
        if (paymentDetailData.errors != null) {
            System.debug('Error details: ' + paymentDetailData.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}
}

```

Usage Example

```

PayClaimCoveragePaymentDetailCallable callable = new
PayClaimCoveragePaymentDetailCallable();
// Prepare the input arguments

```

```

Map<String, Object> input = new Map<String, Object>();
// Set the required parameters
input.put('claimId', '0Zkxx000000002ZCAQ'); // Claim ID
input.put('claimCoverageId', '0kPxx000000000cEAA'); // Claim Coverage ID
input.put('claimCoveragePaymentDetailId', '012xx000000002cAAA'); // Claim Coverage
Payment Detail ID
// Prepare the output map
Map<String, Object> output = new Map<String, Object>();
// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);
// Call the 'payClaimCoveragePaymentDetail' action
Object result = callable.call('payClaimCoveragePaymentDetail', args);
// Check the result
if (result != null) {
    System.debug('Claim coverage payment detail processed successfully: ' + result);

    // Access the processed payment detail data
    ConnectApi.PayClaimCoveragePaymentDetailRep paymentDetailData =
        (ConnectApi.PayClaimCoveragePaymentDetailRep) result;

    // Process the result as needed
    System.debug('Payment Detail Data: ' + paymentDetailData);

    // Check if the payment was successful
    if (paymentDetailData.isSuccess) {
        System.debug('Payment was successful');
    } else {
        System.debug('Payment failed');
        if (paymentDetailData.errors != null) {
            System.debug('Error details: ' + paymentDetailData.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

PolicyLimitStandingCalculation(InsurancePolicyLimitStandingCalculationInput)

Calculate the current standing for policy limits.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsPolicyLmtCalcStandingRepresentation  
PolicyLimitStandingCalculation(ConnectApi.InsPolicyLmtCurrentStandingsInputRep  
InsurancePolicyLimitStandingCalculationInput)
```

Parameters

InsurancePolicyLimitStandingCalculationInput
Type: [ConnectApi.InsPolicyLmtCurrentStandingsInputRep](#)
Details to calculate the current standings of the insurance policy limits.

Return Value

Type: [ConnectApi.InsPolicyLmtCalcStandingRepresentation](#)

postVerifyPolicyCoverage(policyId, coverageId, verifyPolicyCoverageInput)

Verify the applicability of a policy coverage for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.VerifyPolicyCoverageRepresentation  
postVerifyPolicyCoverage(String policyId, String coverageId,  
ConnectApi.VerifyPolicyCoverageInputRepresentation verifyPolicyCoverageInput)
```

Parameters

policyId
Type: String
ID of the insurance policy.

coverageId
Type: String
ID of the policy coverage to verify.

verifyPolicyCoverageInput
Type: [ConnectApi.VerifyPolicyCoverageInputRepresentation](#)
JSON object that contains the loss date and other details for verification.

Return Value

Type: `ConnectApi.VerifyPolicyCoverageRepresentation`

Example

Use the `VerifyPolicyCoverageCallable` Apex class to validate whether a given insurance policy provides coverage for a specific insured asset or participant on a given loss date. This class wraps the Connect API endpoint `/connect/insurance/policies/{policyId}/coverages/{coverageId}/verify`. You can call this method from Apex to automate coverage verification during claim workflows.

Class Definition

```
public with sharing class VerifyPolicyCoverageCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        if (action == 'verifyPolicyCoverage') {
            ConnectApi.VerifyPolicyCoverageInputRepresentation verificationInput =
                (ConnectApi.VerifyPolicyCoverageInputRepresentation)
            inputMap.get('verificationInput');

            System.debug('Start of connect API call to verify policy coverage');

            ConnectApi.VerifyPolicyCoverageRepresentation verificationData;

            try {
                verificationData =
                ConnectApi.InsuranceClaimFamily.postVerifyPolicyCoverage(
                    '0YTxx000000000DGAQ', // policyId
                    '0cYxx000000012IEAQ', // coverageId
                    verificationInput
                );

                System.debug('Policy coverage verification completed: ' +
                verificationData);
                outputMap.put('verificationData', verificationData);
            } catch (Exception e) {
                System.debug('Error while verifying policy coverage: ' + e.getMessage());

                outputMap.put('error', e.getMessage());
            }

            return verificationData;
        }

        return null;
    }
}
```

Usage Example

```

VerifyPolicyCoverageCallable callable = new VerifyPolicyCoverageCallable();

// Prepare input
Map<String, Object> input = new Map<String, Object>();

ConnectApi.VerifyPolicyCoverageInputRepresentation verificationInput =
    new ConnectApi.VerifyPolicyCoverageInputRepresentation();

verificationInput.lossDate = '2025-08-15';
verificationInput.insuredAssetOrParticipantId = '0YWxx00000000P5GAI';

input.put('verificationInput', verificationInput);

// Prepare output and args
Map<String, Object> output = new Map<String, Object>();
Map<String, Object> args = new Map<String, Object>{
    'input' => input,
    'output' => output
};

// Call the 'verifyPolicyCoverage' action
Object result = callable.call('verifyPolicyCoverage', args);

ConnectApi.VerifyPolicyCoverageRepresentation verificationData =
    (ConnectApi.VerifyPolicyCoverageRepresentation) result;

// Log verification result
System.debug('Policy Coverage Verification Result: ' +
    JSON.serializePretty(verificationData));

// Check for success
if (verificationData.isSuccess) {
    System.debug('Policy coverage is valid for the specified loss date');
} else {
    System.debug('Policy coverage verification failed. Errors:');
    for (ConnectApi.ErrorDetail error : verificationData.errors) {
        System.debug('Error Code: ' + error.errorCode + ', Message: ' + error.message);
    }
}

// Check for errors in the output map
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
}

```

PayExGratiaClaimCvrPaymentDetail(claimId, coverageId, paymentDetailId, PayExGratiaClaimCvrPaymentDetailInput)

Process an ex gratia payment for a claim coverage payment detail record.

API Version

66.0

Requires Chatter

No

Signature

```
public static ConnectApi.PayExGratiaClaimCvrPaymentDetailRep  
PayExGratiaClaimCvrPaymentDetail(String claimId, String coverageId, String  
paymentDetailId, ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep  
PayExGratiaClaimCvrPaymentDetailInput)
```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the coverage.

paymentDetailId

Type: String

ID of the payment detail.

PayExGratiaClaimCvrPaymentDetailInput

Type: [ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep](#)

Details to process the ex gratia payment.

Return Value

Type: [ConnectApi.PayExGratiaClaimCvrPaymentDetailRep](#)

Example

Class Definition

This is a sample Apex code to define the `PayExGratiaClaimCvrPaymentDetail` class before you can call it from Apex to process ex gratia payment.

```
public with sharing class PayExGratiaClaimCvrPaymentDetailCallable implements  
System.Callable {  
  
    public Object call(String action, Map<String, Object> args) {  
        // Retrieve the input and output maps from the arguments  
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');  
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');
```

```

// Check the action type
if (action == 'payExGratiaClaimCvrPaymentDetail') {
    // Retrieve the required parameters from the input map
    String claimId = (String) inputMap.get('claimId');
    String claimCoverageId = (String) inputMap.get('claimCoverageId');
    String paymentDetailId = (String) inputMap.get('paymentDetailId');

    ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep payExGratiaInputRep =
        (ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep)
inputMap.get('PayExGratiaClaimCvrPaymentDetailInput');

    System.debug('Start of connect API call to pay ex gratia claim coverage
payment detail');
    System.debug('Claim ID: ' + claimId);
    System.debug('Claim Coverage ID: ' + claimCoverageId);
    System.debug('Payment Detail ID: ' + paymentDetailId);

    // Initialize the output representation
    ConnectApi.PayExGratiaClaimCvrPaymentDetailRep payExGratiaResponse;

    // Call the Connect API to pay ex gratia
    try {
        // Call the Connect API resource directly
        // The endpoint is:
/connect/insurance/claims/${claimId}/coverages/${coverageId}/payment-details/${paymentDetailId}/pay-ex-gratia

        payExGratiaResponse =
ConnectApi.InsuranceClaimFamily.PayExGratiaClaimCvrPaymentDetail(
            payExGratiaInputRep,
            claimId,
            claimCoverageId,
            paymentDetailId
        );

        System.debug('Pay ex gratia completed successfully: ' +
payExGratiaResponse);

        // Put the payExGratiaResponse in the output map
        outputMap.put('payExGratiaResponse', payExGratiaResponse);
        outputMap.put('isSuccess', payExGratiaResponse.isSuccess);

        // Check if there are any errors
        if (payExGratiaResponse.errors != null &&
!payExGratiaResponse.errors.isEmpty()) {
            outputMap.put('errors', payExGratiaResponse.errors);
            System.debug('Errors found: ' + payExGratiaResponse.errors);
        }
    }
    catch (Exception e) {
        System.debug('Error while paying ex gratia: ' + e.getMessage());
        outputMap.put('error', e.getMessage());
        outputMap.put('isSuccess', false);
    }
}

```

```

        return payExGratiaResponse;
    }

    // If the action is not recognized, return null
    return null;
}
}

```

Usage Example

This example shows how to call the `EditClaimCoveragePaymentDetailCallable` class from Apex and include the required input data.

```

EditClaimCoveragePaymentDetailCallable callable = new
EditClaimCoveragePaymentDetailCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Set the required parameters
input.put('claimId', '0zkSB0000001AVH0A2'); // Claim ID
input.put('claimCoverageId', '0kPSB000000GHdu2AG'); // Claim Coverage ID
input.put('paymentDetailId', '012SB000000CLsMYAW'); // Payment Detail ID

// Create the input representation object
// First create a Map with the properties you want to update
Map<String, Object> editInputMap = new Map<String, Object>();

// Set the fields you want to edit (all fields are optional)
editInputMap.put('claimedAmount', 5000.00); // Update claimed amount
editInputMap.put('adjustedAmount', 4500.00); // Update adjusted amount
editInputMap.put('name', 'Updated Payment Detail Name'); // Update name
editInputMap.put('description', 'Updated description for payment detail'); // Update
description
editInputMap.put('limitUnitCount', 10); // Update limit unit
count
editInputMap.put('status', 'Paid'); // Update status
// editInputMap.put('recipientId', '0030000000000000AAA'); // Update recipient ID
if needed
// editInputMap.put('additionalFields', new Map<String, Object>()); // Additional fields
if needed

// Convert the Map to ConnectApi.EditClaimCvrPaymentDetailInputRep using JSON.deserialize
String jsonString = JSON.serialize(editInputMap);
ConnectApi.EditClaimCvrPaymentDetailInputRep editInputRep =
    (ConnectApi.EditClaimCvrPaymentDetailInputRep) JSON.deserialize(jsonString,
ConnectApi.EditClaimCvrPaymentDetailInputRep.class);

input.put('EditClaimCoveragePaymentDetailInput', editInputRep);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();

```

```

args.put('input', input);
args.put('output', output);

// Call the 'editClaimCoveragePaymentDetail' action
Object result = callable.call('editClaimCoveragePaymentDetail', args);

// Check the result
if (result != null) {
    System.debug('editClaimCoveragePaymentDetail processed successfully: ' + result);

    // Access the processed edit data
    ConnectApi.EditClaimCoveragePaymentDetailRep editData =
        (ConnectApi.EditClaimCoveragePaymentDetailRep) result;

    // Process the result as needed
    System.debug('Edit Claim Coverage Payment Detail Data: ' + editData);

    // Check if the edit was successful
    if (editData.isSuccess) {
        System.debug('Edit Claim Coverage Payment Detail was successful');
        System.debug('Adjusted Amount: ' + editData.adjustedAmount);
        System.debug('Adjusted Reason: ' + editData.adjustedReason);

        if (editData.insurancePolicyLimitTrackingIds != null &&
!editData.insurancePolicyLimitTrackingIds.isEmpty()) {
            System.debug('Insurance Policy Limit Tracking IDs: ' +
editData.insurancePolicyLimitTrackingIds);
        }
    } else {
        System.debug('Edit Claim Coverage Payment Detail failed');
        if (editData.errors != null) {
            System.debug('Error details: ' + editData.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

ProcessPolicyLimits(ProcessPolicyLimitsInput, claimId, coverageId)

Process the policy limits for a loss item in a claim by using the current standings of the insurance policy limits.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.ProcessPolicyLimitsRepresentation
ProcessPolicyLimits (ConnectApi.ProcessPolicyLimitsInputRepresentation
ProcessPolicyLimitsInput, String claimId, String coverageId)
```

Parameters

ProcessPolicyLimitsInput

Type: [ConnectApi.ProcessPolicyLimitsInputRepresentation](#)

JSON object that contains details for processing the policy limits.

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage.

Return Value

Type: [ConnectApi.ProcessPolicyLimitsRepresentation](#)

Example

Here's an example of how to invoke the Process Policy Limits API by using Apex.

```
public with sharing class PostApiCalloutProcessLimitTest implements System.Callable {
    public Object call(String action, Map<String, Object> args) {
        System.debug('>>> Action: ' + action);
        System.debug('>>> Args: ' + JSON.serializePretty(args));
        Map<String, Object> bodyMap = new Map<String, Object>();
        // Extract 'input' from args
        Map<String, Object> inputMap = (Map<String, Object>)args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>)args.get('output');
        String claimId;
        String claimCoverageId;
        String CLAIM_ID = 'claimId';
        String CLAIM_COVERAGE_ID = 'claimCoverageId';
        String ADJUSTED_AMOUNT = 'adjustedAmount';
        String LIMIT_UNIT_COUNT = 'limitUnitCount';
        String BENEFIT_NAME = 'benefitName';
        if (inputMap != null) {
            if (inputMap.containsKey(ADJUSTED_AMOUNT) &&
                inputMap.get(ADJUSTED_AMOUNT) != null) {
                bodyMap.put(ADJUSTED_AMOUNT,
                    String.valueOf(inputMap.get(ADJUSTED_AMOUNT)));
            } if (inputMap.containsKey(LIMIT_UNIT_COUNT) &&
                inputMap.get(LIMIT_UNIT_COUNT) != null) {
                bodyMap.put(LIMIT_UNIT_COUNT,
                    String.valueOf(inputMap.get(LIMIT_UNIT_COUNT)));
            } if (inputMap.containsKey(BENEFIT_NAME) && inputMap.get(BENEFIT_NAME) != null)
```

```

{
    bodyMap.put(BENEFIT_NAME, String.valueOf(inputMap.get(BENEFIT_NAME)));
    } if (inputMap.containsKey(CLAIM_ID) && inputMap.get(CLAIM_ID) != null) {
    claimId = String.valueOf(inputMap.get(CLAIM_ID));
    } if (inputMap.containsKey(CLAIM_COVERAGE_ID) &&
    inputMap.get(CLAIM_COVERAGE_ID) != null) {

    claimCoverageId = String.valueOf(inputMap.get(CLAIM_COVERAGE_ID));
    }
    }

    String jsonBody = JSON.serialize(bodyMap);
    System.debug('>>> Final JSON Body: ' + jsonBody);
    System.debug('>>> Claim Id : ' + claimId);
    System.debug('>>> Claim Coverage Id : ' + claimCoverageId);
    String str = postDataToExternalApi(jsonBody, claimId, claimCoverageId);
    return outputMap.put('response', str);
    }

    public static String postDataToExternalApi(String jsonBody, String claimId, String
    claimCoverageId) {
    HttpRequest request = new HttpRequest();
    Http http = new Http();

    // Endpoint for Process Limits
    String url = 'callout:invokeUW/services/data/v65.0/connect/insurance/'

    + 'claims/' + claimId + '/coverages/' + claimCoverageId + '/process-limits';

    request.setEndpoint(url);
    request.setMethod('POST');
    request.setHeader('Content-Type', 'application/json');
    request.setTimeout(120000); // 2 mins timeout
    request.setBody(jsonBody);
    System.debug('>>> Calling Process Limits API');
    System.debug('>>> Calling Endpoint : ' + url );
    try {
    HttpResponse response = http.send(request);
    System.debug('>>> Status Code: ' + response.getStatusCode());
    System.debug('>>> Response Body: ' + response.getBody());
    return response.getBody();
    } catch (Exception e) {
    System.debug('☐☐ Exception: ' + e.getMessage());
    return 'Exception: ' + e.getMessage();
    }
    }
    }
    }

```

RecalculateAdjustments(claimId, coverageId, RecalculateAdjustmentsInput)

Recalculate adjustments for existing payment details.

API Version

66.0

Requires Chatter

No

Signature

```
public static ConnectApi.CalculateAdjustmentsRepresentation RecalculateAdjustments(String
claimId, String coverageId, ConnectApi.RecalculateAdjustmentsInputRep
RecalculateAdjustmentsInput)
```

Parameters

claimId

Type: String

ID of the claim for which the adjustments must be recalculated.

coverageId

Type: String

ID of the claim coverage.

*RecalculateAdjustmentsInput*Type: [ConnectApi.RecalculateAdjustmentsInputRep](#)

Details to recalculate the adjustments.

Return Value

Type: [ConnectApi.CalculateAdjustmentsRepresentation](#)

Example

Class Definition

This is a sample Apex code to define the `RecalculateAdjustments` class before you can call it from Apex to recalculate payment adjustments.

```
public with sharing class RecalculateAdjustmentsCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'recalculateAdjustments') {
            // Retrieve the required parameters from the input map
            String claimId = (String) inputMap.get('claimId');
            String claimCoverageId = (String) inputMap.get('claimCoverageId');

            ConnectApi.RecalculateAdjustmentsInputRep recalculateAdjustmentInputRep =
```

```

        (ConnectApi.RecalculateAdjustmentsInputRep)
inputMap.get('RecalculateAdjustmentsInput');

    System.debug('Start of connect API call to recalculate adjustments');
    System.debug('Claim ID: ' + claimId);
    System.debug('Claim Coverage ID: ' + claimCoverageId);

    // Initialize the output representation
    ConnectApi.CalculateAdjustmentsRepresentation reCalcAdjResponse;

    // Call the Connect API to recalculate adjustments
    try {
        // Call the Connect API resource directly
        // The endpoint is:
/connect/insurance/claims/${claimId}/coverages/${coverageId}/recalculate-adjustments
        reCalcAdjResponse =
ConnectApi.InsuranceClaimFamily.RecalculateAdjustments(
            recalculateAdjustmentInputRep,
            claimId,
            claimCoverageId
        );

        System.debug('Recalculate adjustments completed successfully: ' +
reCalcAdjResponse);

        // Put the reCalcAdjResponse in the output map
        outputMap.put('reCalcAdjResponse', reCalcAdjResponse);
        outputMap.put('isSuccess', reCalcAdjResponse.isSuccess);

        // Check if there are any errors
        if (reCalcAdjResponse.errors != null &&
!reCalcAdjResponse.errors.isEmpty()) {
            outputMap.put('errors', reCalcAdjResponse.errors);
            System.debug('Errors found: ' + reCalcAdjResponse.errors);
        }
    }
    catch (Exception e) {
        System.debug('Error while recalculating adjustments: ' + e.getMessage());

        outputMap.put('error', e.getMessage());
        outputMap.put('isSuccess', false);
    }

    return reCalcAdjResponse;
}

// If the action is not recognized, return null
return null;
}
}

```


Usage Example

This example shows how to call the `RecalculateAdjustmentsCallable` class from Apex and include the required input data.

```
RecalculateAdjustmentsCallable callable = new RecalculateAdjustmentsCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Set the required parameters
input.put('claimId', '0ZkSB0000001AVH0A2'); // Claim ID
input.put('claimCoverageId', '0kPSB000000GHdu2AG');

// Create the input representation object
// First create a Map with the properties
Map<String, Object> recalculateAdjustmentInputMap = new Map<String, Object>();
recalculateAdjustmentInputMap.put('paymentDetailId', '012SB000000CLsMYAW');
recalculateAdjustmentInputMap.put('claimedAmount', 4252.35);

// Convert the Map to ConnectApi.RecalculateAdjustmentsInputRep using JSON.deserialize
String jsonString = JSON.serialize(recalculateAdjustmentInputMap);
ConnectApi.RecalculateAdjustmentsInputRep recalculateAdjustmentInputRep =
    (ConnectApi.RecalculateAdjustmentsInputRep) JSON.deserialize(jsonString,
    ConnectApi.RecalculateAdjustmentsInputRep.class);

input.put('RecalculateAdjustmentsInput', recalculateAdjustmentInputRep);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'recalculateAdjustments' action
Object result = callable.call('recalculateAdjustments', args);

// Check the result
if (result != null) {
    System.debug('recalculateAdjustments processed successfully: ' + result);

    // Access the processed adjustment data
    ConnectApi.CalculateAdjustmentsRepresentation reCalcAdjResponse =
        (ConnectApi.CalculateAdjustmentsRepresentation) result;

    // Process the result as needed
    System.debug('recalc Adjustment response: ' + reCalcAdjResponse);

    // Check if the response was successful
    if (reCalcAdjResponse.isSuccess) {
        System.debug('recalculateAdjustment was successful');
    } else {
```

```

        System.debug('recalculateAdjustment failed');
        if (reCalcAdjResponse.errors != null) {
            System.debug('Error details: ' + reCalcAdjResponse.errors);
        }
    }
} else {
    System.debug('Error occurred: ' + output.get('error'));
}

```

UpdateClaim(claimId, UpdateClaimInput)

Update the details of a specific claim.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.CreateClaimOutputRepresentation UpdateClaim(String claimId,
ConnectApi.ClaimInputRepresentation UpdateClaimInput)

```

Parameters

claimId

Type: String

ID of the claim to update.

UpdateClaimInput

Type: [ConnectApi.ClaimInputRepresentation](#)

Details to update the claim.

Return Value

Type: [ConnectApi.CreateClaimOutputRepresentation](#)

Example

The UpdateClaimCallable Apex class provides a way to update insurance claims by using the Connect API.

Class Definition

```

public with sharing class UpdateClaimCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');
    }
}

```

```

        if (action == 'updateClaim') {
            ConnectApi.ClaimInputRepresentation claimInput =
                (ConnectApi.ClaimInputRepresentation) inputMap.get('claimInput');
            System.debug('Starting Connect API call to update claim');

            ConnectApi.CreateClaimOutputRepresentation claimData;
            try {
                claimData = ConnectApi.ClaimFamily.updateClaim('0Zkxx00000000BJCAY',
claimInput);
                System.debug('Claim updated successfully: ' + claimData);
                outputMap.put('claimData', claimData);
            } catch (Exception e) {
                System.debug('Error updating claim: ' + e.getMessage());
                outputMap.put('error', e.getMessage());
            }

            return claimData;
        }

        return null;
    }
}

```

Usage Example

```

UpdateClaimCallable callable = new UpdateClaimCallable();

// Prepare input map
Map<String, Object> input = new Map<String, Object>();

// Create ClaimInputRepresentation object
ConnectApi.ClaimInputRepresentation claimInput = new
ConnectApi.ClaimInputRepresentation();
claimInput.name = 'Claim-Demo-Test-apex-update';
claimInput.claimType = 'Auto';
claimInput.lossDate = DateTime.now();
claimInput.productCode = 'AutoClaimRoot';
claimInput.claimReasonType = 'Accident';
claimInput.claimReason = 'Damage';
claimInput.lossType = 'Repairs';
claimInput.insurancePolicyId = '0YTxx000000000iGAA';
claimInput.accountId = '001xx000003GYnoAAG';

// Additional claim fields
claimInput.additionalFields = new Map<String, Object>{ 'Severity' => 'Medium' };

// Set claim participants
ConnectApi.ClaimParticipantInputRepresentation participant1 = new
ConnectApi.ClaimParticipantInputRepresentation();
participant1.action = 'update';
participant1.instanceKey = 'par1';
participant1.accountId = '001xx000003GYnoAAG';
participant1.contactId = '003xx000004WhFwAAK';

```

```

participant1.insurancePolicyParticipantId = '0aoxx00000000SQAAy';
participant1.isInjured = 'false';
participant1.roles = new List<String>{'Claimant'};

ConnectApi.ClaimParticipantInputRepresentation participant2 = new
ConnectApi.ClaimParticipantInputRepresentation();
participant2.action = 'create';
participant2.instanceKey = 'par3';
participant2.contactId = '003xx000004WhFwAAK';
participant2.isInjured = 'true';
participant2.roles = new List<String>{'Claimant', 'Involved Driver'};

claimInput.participants = new
List<ConnectApi.ClaimParticipantInputRepresentation>{participant1, participant2};

// Set claim items (update or create)
ConnectApi.ClaimItemInputRepresentation item1 = new
ConnectApi.ClaimItemInputRepresentation();
item1.action = 'update';
item1.name = 'item1-updated';
item1.category = 'Involved Injury';
item1.productCode = 'InvolvedInjury';
item1.instanceKey = 'item1';
item1.insuredItemId = '0YWxx00000000NbGAI';
item1.insurancePolicyCoverageId = '0cYxx0000000125EAA';
item1.participantInstanceKey = 'par1';
item1.attributes = new List<ConnectApi.ClaimAttributeInputRepresentation>{
    new ConnectApi.ClaimAttributeInputRepresentation('Gender', 'Male')
};
item1.additionalFields = new Map<String, Object>{ 'Description' => null };
item1.relatedObjects = new List<ConnectApi.ClaimRelatedObjectInputRepresentation>{
    new ConnectApi.ClaimRelatedObjectInputRepresentation('001xx000003GYnoAAG', 'Account')
};

ConnectApi.ClaimItemInputRepresentation item2 = new
ConnectApi.ClaimItemInputRepresentation();
item2.action = 'create';
item2.name = 'item2-created';
item2.category = 'Involved Injury';
item2.productCode = 'InvolvedInjury';
item2.instanceKey = 'item2';
item2.insuredItemId = '0YWxx00000000NbGAI';
item2.insurancePolicyCoverageId = '0cYxx0000000125EAA';
item2.participantInstanceKey = 'par2';
item2.attributes = new List<ConnectApi.ClaimAttributeInputRepresentation>{
    new ConnectApi.ClaimAttributeInputRepresentation('Gender', 'Male')
};
item2.additionalFields = new Map<String, Object>{ 'Description' => 'test added' };
item2.relatedObjects = new List<ConnectApi.ClaimRelatedObjectInputRepresentation>{
    new ConnectApi.ClaimRelatedObjectInputRepresentation('001xx000003GYnoAAG', 'Account')
};

claimInput.items = new List<ConnectApi.ClaimItemInputRepresentation>{item1, item2};

```

```
// Add claimInput to input map
input.put('claimInput', claimInput);

// Prepare output map
Map<String, Object> output = new Map<String, Object>();

// Execute updateClaim action
Map<String, Object> args = new Map<String, Object>{ 'input' => input, 'output' => output
};
ConnectApi.CreateClaimOutputRepresentation claimData =
    (ConnectApi.CreateClaimOutputRepresentation) callable.call('updateClaim', args);

// Log results
System.debug('Claim Data: ' + JSON.serializePretty(claimData));
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
}
```

UpdateClaimCoverage(claimId, coverageId, UpdateClaimCoverageInput)

Update the details of a specific coverage in a claim.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.ClaimCoverageRepresentation UpdateClaimCoverage(String claimId,
String coverageId, ConnectApi.UpdateClaimCoverageInputRepresentation
UpdateClaimCoverageInput)
```

Parameters

claimId

Type: String

ID of the claim.

coverageId

Type: String

ID of the claim coverage to update.

UpdateClaimCoverageInput

Type: [ConnectApi.UpdateClaimCoverageInputRepresentation](#)

Details of the claim coverage.

Return Value

Type: `ConnectApi.ClaimCoverageRepresentation`

Example

The Apex example shows how to use the `ConnectApi.InsuranceClaimFamily.UpdateClaimCoverage()` method to update a claim coverage for an existing claim.

Class Definition

```
public with sharing class UpdateClaimCoverageCallable implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'updateClaimCoverage') {
            // Retrieve the coverage input data from the input map
            ConnectApi.UpdateClaimCoverageInputRepresentation coverageInput =
            (ConnectApi.UpdateClaimCoverageInputRepresentation) inputMap.get('coverageInput');
            System.debug('Start of connect API call to update claim coverage');

            // Initialize the coverage output representation
            ConnectApi.ClaimCoverageRepresentation coverageData;

            // Call the Connect API to update the claim coverage
            try {
                coverageData =
                ConnectApi.ClaimFamily.updateClaimCoverage('0Zkxx00000000JNCAY', '0kPxx00000000RREAY',
                coverageInput);
                System.debug('Claim coverage updated successfully: ' + coverageData);

                // Put the coverage data in the output map
                outputMap.put('coverageData', coverageData);
            }
            catch (Exception e) {
                System.debug('Error while updating claim coverage: ' + e.getMessage());
                outputMap.put('error', e.getMessage());
            }

            // Return the coverage data
            return coverageData;
        }
    }
}
```

```
        // If the action is not recognized, return null
        return null;
    }
}
```

Usage Example

This example shows how to call the

```
UpdateClaimCoverageCallable
```

class and include the required input data.

```
UpdateClaimCoverageCallable callable = new UpdateClaimCoverageCallable();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();

// Create and populate the UpdateClaimCoverageInputRepresentation object
ConnectApi.UpdateClaimCoverageInputRepresentation coverageInput = new
ConnectApi.UpdateClaimCoverageInputRepresentation();

// Set optional fields for update
coverageInput.lossReserveAmount = 30000.00;
coverageInput.expenseReserveAmount = 3000.00;

// Set additional fields
Map<String, Object> additionalFields = new Map<String, Object>();
additionalFields.put('Description', 'Updated coverage for vehicle damage claim');
coverageInput.additionalFields = additionalFields;

// Add the coverageInput to the input map
input.put('coverageInput', coverageInput);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'updateClaimCoverage' action
Object result = callable.call('updateClaimCoverage', args);

// Retrieve the coverage data from the result
ConnectApi.ClaimCoverageRepresentation coverageData =
```

```
(ConnectApi.ClaimCoverageRepresentation) result;

// Output the coverage data to the debug log
System.debug('Updated Coverage Data: ' + JSON.serializePretty(coverageData));

// Check for errors in the output map
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
}
```

ConnectApi Input Classes

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

Input classes are concrete unless marked abstract in this documentation. Concrete input classes have public constructors that have no parameters.

Some methods have parameters that are typed with an abstract class. You must pass in an instance of a concrete child class for these parameters.

Most input class properties can be set. Read-only properties are noted in this documentation.

Insurance Claim includes these Apex input classes.

[ConnectApi.CalculateAdjustmentsInputRepresentation](#)

Input representation for calculating adjustments for claim loss items.

[ConnectApi.ClaimAdditionalFieldInputRep](#)

Input representation for the details of the additional field.

[ConnectApi.ClaimInputRepresentation](#)

Input representation for creating or updating a claim.

[ConnectApi.ClaimAttributeInputRepresentation](#)

Input representation for claim attributes.

[ConnectApi.ClaimAttributeInputRep](#)

Invocable action input representation for claim attributes.

[ConnectApi.ClaimItemInputRepresentation](#)

Input representation for claim items.

[ConnectApi.ClaimItemInputRep](#)

Invocable action input representation for claim item.

[ConnectApi.ClaimItemRelatedObjectInputRep](#)

Invocable action input representation for the related object in claim items.

[ConnectApi.ClaimOptionsInputRepresentation](#)

Input representation for claim options.

[ConnectApi.ClaimOptionInputRep](#)

Invocable action input representation for claim options

[ConnectApi.ClaimParticipantInputRepresentation](#)

Input representation for a claim participant.

[ConnectApi.ClaimParticipantInputRep](#)

Invocable Action Input representation for Claim Participant

[ConnectApi.ClaimRelatedObjectInputRepresentation](#)

Input representation for a related object on a claim.

[ConnectApi.CreateClaimInputRepresentation](#)

Input representation for the Create Claim action.

[ConnectApi.CreateClaimCoverageInputRepresentation](#)

Input representation for creating a claim coverage.

[ConnectApi.CreateClaimCovPaymentDetailInputRep](#)

Input representation for creating a claim coverage payment detail record.

[ConnectApi.CreatePolicyLimitsInputRepresentation](#)

Input representation for creating insurance policy limit records.

[ConnectApi.CreatePolicyLimitsInputRep](#)

Input representation for the apex ConnectApi.CreatePolicyLimitsInputRep record that contains the details of the policy limits to be created.

[ConnectApi.DeleteClaimCvrPymtDtlInputRep](#)

Input representation for deleting claim coverage payment detail operations

[ConnectApi.EditClaimCvrPaymentDetailInputRep](#)

Input representation for editing claim coverage payment detail.

[ConnectApi.GetClaimInputRepresentation](#)

Input representation to get the details of a claim.

[ConnectApi.GetInsPlcyCurrStndInputRep](#)

An Apex ConnectApi.GetInsPlcyLmtCurrStndInputRep record that contains the details needed to retrieve the consumed and available limits for the policy or claim.

[ConnectApi.IPLStandingCalcOptions](#)

Invocable action input representation for options to calculate standings.

[ConnectApi.InsPolicyLmtCurrentStandingsInputRep](#)

Input representation for the current standings of the insurance policy limits.

[ConnectApi.LimitStandingOptionInputRepresentation](#)

Input representation for limit standing options.

[ConnectApi.OpenClaimCoverageInputRep](#)

Input representation for the open claim coverage action.

[ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep](#)

Input representation for the ex gratia payment for the claim coverage payment detail.

[ConnectApi.ProcessPolicyLimitsInputRepresentation](#)

Input representation for processing policy limits for a claim loss item.

[ConnectApi.RecalculateAdjustmentsInputRep](#)

Input representation for recalculating the adjustments for existing payment details.

[ConnectApi.UpdateClaimInputRepresentation](#)

Invocable action input representation for the update claim action.

[ConnectApi.UpdateClaimCoverageInputRepresentation](#)

Input representation for updating a claim coverage.

[ConnectApi.UpdateClaimCoverageInputRep](#)

Input representation for the update claim coverage action.

[ConnectApi.VerifyPolicyCoverageInputRepresentation](#)

Input representation for verifying a policy coverage's applicability for a specified loss date.

[ConnectApi.VerifyPolicyCoverageInputRep](#)

Invocable action input representation details to verify the coverage of a specific insurance policy.

ConnectApi.CalculateAdjustmentsInputRepresentation

Input representation for calculating adjustments for claim loss items.

Property	Type	Description	Required or Optional	Available Version
claimedAmount	Double	Amount that's requested by the claimant when submitting the loss.	Required	65.0
benefitName	String	Name of the specific benefit or sub-coverage that's associated with the coverage, as defined in the product attribute mapping.	Optional	66.0

ConnectApi.ClaimAdditionalFieldInputRep

Input representation for the details of the additional field.

Property	Type	Description	Required or Optional	Available Version
key	String	API name of key.	Optional	65.0
value	String	Corresponding value for the key.	Optional	65.0

ConnectApi.ClaimInputRepresentation

Input representation for creating or updating a claim.

Property	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	Object	Additional fields for the claim.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
attributes	~~the Claim Input Representation~~	List of attributes for the claim product.	Optional	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't accept time of incident. To specify both the date and time of loss, use the <code>lossDate</code> parameter.	Optional	66.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Type of reason for the claim.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0
incidentTypeId	String	ID of the incident type.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	~~the Claim Input Representation~~	Claim items that are associated with the claim.	Optional	65.0
lossDate	Date	Date on which the loss occurred.	Optional	65.0
lossType	String	Type of loss for which the claim is raised.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	~~the Claim Input Representation~~	Options to include during the claim creation or update process.	Optional	65.0
participants	~~the Claim Input Representation~~	Participants who are associated with the claim.	Optional	65.0
productCode	String	Product code of the claim product.	Optional	65.0

ConnectApi.ClaimAttributeInputRepresentation

Input representation for claim attributes.

Property	Type	Description	Required or Optional	Available Version
apiName	String	API name of the claim attribute.	Optional	65.0
value	String	Value for the claim attribute.	Optional	65.0

ConnectApi.ClaimAttributeInputRep

Invocable action input representation for claim attributes.

Property	Type	Description	Required or Optional	Available Version
apiName	String	API Name of the claim attribute.	Optional	65.0
value	String	Corresponding value for the claim attribute.	Optional	65.0

ConnectApi.ClaimItemInputRepresentation

Input representation for claim items.

Property	Type	Description	Required or Optional	Available Version
action	String	Action to perform on the record, such as create or update.	Optional	65.0
additionalFields	Object	Additional standard or custom fields for the claim item.	Optional	65.0
attributes	The ClaimItemRepresentation on page 213>	Attributes of the claim item.	Optional	65.0
category	String	Category of the claim item.	Optional	65.0
instanceKey	String	Unique identifier of the claim item.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage that applies to the claim item.	Optional	65.0
insuredItemId	String	ID of the insured item that's involved in the claim.	Optional	65.0
name	String	Name of the claim item.	Optional	65.0
participantInstanceKey	String	Unique identifier that's used to link a claim item to a specific participant.	Optional	65.0
productCode	String	Product code of the claim item product.	Optional	65.0
relatedObjects	The ClaimItemRepresentation on page 217>	Related claim item records.	Optional	65.0

ConnectApi.ClaimItemInputRep

Invocable action input representation for claim item.

Property	Type	Description	Required or Optional	Available Version
action	String	Action to perform on the record, such as add, update, or delete.	Optional	65.0
additionalFields	The ClaimItemInputRep on page 214>	Additional fields for the claim item.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
attributes	~~ConnectApi.ClaimItemRep~~	Attributes that are assigned to claim item.	Optional	65.0
category	String	Category of claim item.	Optional	65.0
instanceKey	String	Unique identifier for claim item.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage that applies to the claim item.	Optional	65.0
insuredItemId	String	ID of the insured item that's involved in the claim.	Optional	65.0
name	String	Name of the claim item.	Optional	65.0
participantInstanceKey	String	Unique identifier that's used to link a claim item to a specific participant.	Optional	65.0
productCode	String	Product code of the claim item product.	Optional	65.0
relatedObjects	~~ConnectApi.ClaimItemRep~~	Related claim item records.	Optional	65.0

ConnectApi.ClaimItemRelatedObjectInputRep

Invocable action input representation for the related object in claim items.

Property	Type	Description	Required or Optional	Available Version
recordId	String	ID of the related claim item record.	Optional	65.0
recordObjName	String	Object name of the related claim item record.	Optional	65.0

ConnectApi.ClaimOptionsInputRepresentation

Input representation for claim options.

Property	Type	Description	Required or Optional	Available Version
execute Configuration Rules	String	Specifies whether to execute configuration rules or not.	Optional	65.0

ConnectApi.ClaimOptionInputRep

Invocable action input representation for claim options

Property	Type	Description	Required or Optional	Available Version
~~executeConfigurationRules~~	String	Specifies whether to execute configuration rules or not.	Optional	65.0

ConnectApi.ClaimParticipantInputRepresentation

Input representation for a claim participant.

Property	Type	Description	Required or Optional	Available Version
accountId	String	Account ID of the claim participant.	Optional	65.0
action	String	Action to perform on the record, such as create or update.	Optional	65.0
additionalFields	Object	Additional fields for the claim record to create or update the record.	Optional	65.0
contactId	String	Contact ID of the claim participant.	Optional	65.0
instanceKey	String	Unique instance key of the claim participant.	Optional	65.0
~~insurancePolicyParticipantId~~	String	ID of the insurance policy participant that's related to the claim.	Optional	65.0
isInjured	String	Specifies if the participant is injured or not.	Optional	65.0
roles	List<String>	Roles of the claim participant.	Optional	65.0

ConnectApi.ClaimParticipantInputRep

Invocable Action Input representation for Claim Participant

Property	Type	Description	Required or Optional	Available Version
accountId	String	Account ID that's associated with the claim participant.	Optional	65.0
action	String	Action to perform on the record, such as create or update.	Optional	65.0
additionalFields	~~Object~~ List<String>	Additional standard or custom fields to add details for the claim participant.	Optional	65.0
contactId	String	Contact ID that's associated with the participant.	Optional	65.0
instanceKey	String	Unique identifier for the participant.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
insurancePolicyParticipantId	String	ID of the insurance policy participant that's related to the claim.	Optional	65.0
isInjured	String	Specifies if the participant is injured or not.	Optional	65.0
roles	List<String>	Roles of the claim participant.	Optional	65.0

ConnectApi.ClaimRelatedObjectInputRepresentation

Input representation for a related object on a claim.

Property	Type	Description	Required or Optional	Available Version
recordId	String	ID of the related claim item record.	Optional	65.0
recordObjName	String	Object name of the related claim item record.	Optional	65.0

ConnectApi.CreateClaimInputRepresentation

Input representation for the Create Claim action.

Property	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	String	Additional standard or custom fields that are used to create or update insurance claim. ConnectApi.ClaimInputRepresentation on page 212>	Optional	65.0
attributes	String	Attributes of the claim product. ConnectApi.ClaimInputRepresentation on page 213>	Optional	65.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Category that the claim reason belongs to.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0
incidentTypeId	String	ID for the type of incident in the claim.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	String	Details of the items that are associated with the claim. ConnectApi.ClaimInputRepresentation on page 214>	Optional	65.0
lossDate	String	Date on which the loss occurred.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
lossType	String	Type of the loss for which the claim is initiated.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	ClaimOptionsInputRep on page 215	Options for creating or updating a claim.	Optional	65.0
participants	CreateClaimParticipantRep on page 216>	Claim participants that are associated with the claim.	Optional	65.0
productCode	String	Product code of the claim product.	Required to create a claim	65.0

ConnectApi.CreateClaimCoverageInputRepresentation

Input representation for creating a claim coverage.

Property	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional fields for the claim coverage.	Optional	65.0
claimItemId	String	ID of the claim item.	Optional	65.0
claimParticipantId	String	ID of the claim participant.	Optional	65.0
currencyIsoCode	String	Currency ISO code for reserve adjustments.	Optional	65.0
expenseReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
insurancePolicyAssetId	String	ID of the insurance policy asset.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage.	Optional	65.0
lossReserveAmount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0
name	String	Name of the claim coverage.	Optional	65.0
reserveProcessingMode	String	Determines how financial reserves are handled when a claim coverage is created.	Optional	65.0

ConnectApi.CreateClaimCovPaymentDetailInputRep

Input representation for creating a claim coverage payment detail record.

Property	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields that are used to create the claim coverage payment detail record.	Optional	65.0
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	Optional	65.0
benefitName	String	Sub-coverage information.	Optional	65.0
claimedAmount	Double	Amount of the claim.	Optional	65.0
coveragePaymentType	String	Specifies the category of the payment. The value must be either Loss or Expense. Loss refers to the payment related to the policyholder's covered damages. Expense refers to costs associated with claim processing, such as vendor fees.	Optional	65.0
description	String	Description of the payment detail.	Optional	65.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	65.0
recipientId	String	ID of the recipient.	Optional	65.0

ConnectApi.CreatePolicyLimitsInputRepresentation

Input representation for creating insurance policy limit records.

Property	Type	Description	Required or Optional	Available Version
attributeScopes	List<String>	List of attribute scopes for which the policy limit records must be created.	Optional	65.0

ConnectApi.CreatePolicyLimitsInputRep

Input representation for the apex ConnectApi.CreatePolicyLimitsInputRep record that contains the details of the policy limits to be created.

Property	Type	Description	Required or Optional	Available Version
attributeScopes	List<String>	Attribute scopes for which policy limit records must be created.	Optional	65.0
policyId	String	ID of the insurance policy for which the limits must be generated.	Required	65.0

ConnectApi.DeleteClaimCvrPymtDtlInputRep

Input representation for deleting claim coverage payment detail operations

Property	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim whose coverage payment detail must be deleted.	Required	65.0
coverageId	String	ID of the claim coverage record.	Required	65.0
paymentDetailId	String	ID of the claim coverage payment detail record.	Required	65.0

ConnectApi.EditClaimCvrPaymentDetailInputRep

Input representation for editing claim coverage payment detail.

Property	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional standard or custom fields that are used to update the claim coverage payment detail record.	Optional	66.0
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	Optional	66.0
claimedAmount	Double	Amount of the claim.	Optional	66.0
description	String	Description of the payment detail.	Optional	66.0
limitUnitCount	Integer	Number of units for this claim coverage payment detail, such as the number of days for a rental car expense. The default value is 1.	Optional	66.0
name	String	Name of the claim coverage payment detail record.	Optional	66.0
recipientId	String	ID of the person or entity who will receive the payment, such as the policyholder, a third-party claimant, or a service vendor. For a Loss item, the recipient is the policyholder or a third party. For an Expense item, the recipient ID represents a vendor or internal staff member.	Optional	66.0
status	String	Current status of the claim coverage payment detail.	Optional	66.0

ConnectApi.GetClaimInputRepresentation

Input representation to get the details of a claim.

Property	Type	Description	Required or Optional	Available Version
claimId	String	ID of the claim to get the details for.	Required	65.0

ConnectApi.GetInsPclyCurrStndInputRep

An Apex ConnectApi.GetInsPclyLmtCurrStndInputRep record that contains the details needed to retrieve the consumed and available limits for the policy or claim.

Property	Type	Description	Required or Optional	Available Version
claimCoverageId	String	ID of the claim coverage record.	Optional	65.0
claimId	String	ID of the claim for which you want to get the current standings.	Required to get the standings for a claim.	65.0
options	ConnectApi.StandingCalcOptions	Options to include pending values and filter coverage.	Optional	65.0
policyAssetId	String	ID of the insurance policy asset.	Optional	65.0
policyCoverageId	String	ID of the policy coverage record.	Optional	65.0
policyId	String	ID of the insurance policy for which you want to get the current standings.	Required to get the standings for a policy.	65.0
policyParticipantId	String	ID of the policy participant.	Optional	65.0

ConnectApi.IPLStandingCalcOptions

Invocable action input representation for options to calculate standings.

Property	Type	Description	Required or Optional	Available Version
shouldIncludePending	Boolean	Indicates whether the pending values must be included in the response (<code>true</code>) or not (<code>false</code>) The default value is <code>false</code> .	Optional	65.0
viewOption	String	Filter values for coverages. When a <code>claimId</code> is specified in the request, valid values are:	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
		<code>showAllPolicyCoverages</code>—Retrieves all policy limits for the root insurance policy and all associated policy coverages. This provides a comprehensive view of all coverage terms. <code>showOpenClaimPolicyCoverages</code>—(Default) Retrieves the policy limits that are associated with the open claim coverages. Additionally, the claim view includes all policy-level and claim-level limits, ensuring visibility into limits that impact the claim, even if no open coverage exists. When a <code>policyId</code> is specified in the request, valid values are: <code>showAllPolicyCoverages</code>—(Default) Retrieves all policy limit standings, regardless of whether the claim has open coverages. This option provides a full view of the policy's limits without restriction to active claims. <code>showPolicyLevelLimitsOnly</code>—Retrieves policy limit standings at the policy and policy coverage level, but excludes limits that are associated with policy participants or policy assets. Use this option when only high-level policy limits are needed without participant or asset specific details.		

ConnectApi.InsPolicyLmtCurrentStandingsInputRep

Input representation for the current standings of the insurance policy limits.

Property	Type	Description	Required or Optional	Available Version
<code>claimCoverageId</code>	String	Unique ID of the Claim Coverage.	Optional	65.0
<code>claimId</code>	String	ID of the claim.	Optional	65.0
<code>options</code>	ClaimInputRep on page 223	Options for the calculation.	Optional	65.0
<code>policyAssetId</code>	String	ID of the asset.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
policyCoverageId	String	ID of the coverage.	Optional	65.0
policyId	String	ID of the insurance policy.	Optional	65.0
policyParticipantId	String	ID of the participant.	Optional	65.0
trackingDate	Date	Date on which the standing is calculated. The default value is the current system date.	Optional	66.0

ConnectApi.LimitStandingOptionInputRepresentation

Input representation for limit standing options.

Property	Type	Description	Required or Optional	Available Version
shouldIncludePendingAmount	Boolean	Indicates whether the pending amount must be included in the calculated output (<code>true</code>) or not (<code>false</code>).	Optional	65.0
viewOption	String	View options.	Optional	65.0

ConnectApi.OpenClaimCoverageInputRep

Input representation for the open claim coverage action.

Property	Type	Description	Required or Optional	Available Version
additionalFields	~~String~~ AdditionalField	Additional standard or custom fields to open the claim coverage.	Optional	65.0
claimId	String	ID of the claim record.	Required	65.0
claimItemId	String	ID of the claim item.	Required	65.0
claimParticipantId	String	ID of the claim participant that's associated with the claim coverage.	Required	65.0
currencyIsoCode	String	Currency ISO code for reserve adjustments.	Optional	65.0
expenseReserveAmount	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
insurancePolicyAssetId	String	ID of the asset covered by the insurance policy that's associated with the claim coverage.	Optional	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
lossReserveAmount	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0
name	String	Name of the claim coverage.	Optional	65.0
reserveProcessingMode	String	Determines how financial reserves are handled when a claim coverage is created.	Optional	65.0

ConnectApi.PayExGratiaClaimCvrPaymentDetailInputRep

Input representation for the ex gratia payment for the claim coverage payment detail.

Property	Type	Description	Required or Optional	Available Version
reason	String	Reason for the ex gratia payment.	Optional	66.0

ConnectApi.ProcessPolicyLimitsInputRepresentation

Input representation for processing policy limits for a claim loss item.

Property	Type	Description	Required or Optional	Available Version
adjustedAmount	Double	Amount after adjustments are made based on the current standings of the policy limits.	Optional	65.0
benefitName	String	Name of the specific benefit or sub-coverage that the loss item applies to.	Optional	65.0
limitUnitCount	Integer	Unit count that the payment detail represents. Default value is 1.	Optional	65.0

ConnectApi.RecalculateAdjustmentsInputRep

Input representation for recalculating the adjustments for existing payment details.

Property	Type	Description	Required or Optional	Available Version
claimedAmount	Double	Amount requested by the claimant when submitting the loss.	Optional	66.0
paymentDetailId	String	ID of the claim coverage payment detail for recalculating adjustments.	Optional	66.0

ConnectApi.UpdateClaimInputRepresentation

Invocable action input representation for the update claim action.

Property	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account that's associated with the claim.	Optional	65.0
additionalFields	CreateClaimInputRep	Additional standard or custom fields that are used to create or update insurance claim.	Optional	65.0
attributes	CreateClaimInputRep	Attributes of the claim product.	Optional	65.0
claimId	String	ID of the claim.	Required	65.0
claimReason	String	Reason for the claim.	Optional	65.0
claimReasonType	String	Category that the claim reason belongs to.	Optional	65.0
claimType	String	Type of claim.	Optional	65.0
incidentTypeId	String	ID for the type of incident in the claim.	Optional	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	Optional	65.0
items	CreateClaimInputRep	Details of the items that are associated with the claim.	Optional	65.0
lossDate	String	Date on which the loss occurred.	Optional	65.0
lossType	String	Type of the loss for which the claim is initiated.	Optional	65.0
name	String	Name of the claim.	Optional	65.0
options	ClaimOptionsInputRep	Options for creating or updating a claim.	Optional	65.0
participants	CreateClaimInputRep	Claim participants that are associated with the claim.	Optional	65.0
productCode	String	Product code of the claim product.	Optional	65.0

ConnectApi.UpdateClaimCoverageInputRepresentation

Input representation for updating a claim coverage.

Property	Type	Description	Required or Optional	Available Version
additionalFields	Object	Additional fields for the coverage.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
~~expenseReserveAmount~~	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
~~lossReserveAmount~~	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0

ConnectApi.UpdateClaimCoverageInputRep

Input representation for the update claim coverage action.

Property	Type	Description	Required or Optional	Available Version
~~additionalFields~~ ~~Optional~~	~~String~~	Additional standard or custom fields to update the record.	Optional	65.0
claimId	String	ID of the claim.	Required	65.0
coverageId	String	ID of the coverage within the claim.	Required	65.0
~~expenseReserveAmount~~	Double	Amount that's reserved to cover potential expenses while handling the claim, such as administrative or legal costs.	Optional	65.0
~~lossReserveAmount~~	Double	Amount that's reserved to cover a potential payment for a loss.	Optional	65.0

ConnectApi.VerifyPolicyCoverageInputRepresentation

Input representation for verifying a policy coverage's applicability for a specified loss date.

Property	Type	Description	Required or Optional	Available Version
~~insuredParticipantId~~	String	ID of the insured asset or participant that's related to the policy coverage.	Optional	65.0
lossDate	String	Date on which the incident occurred.	Optional	65.0

ConnectApi.VerifyPolicyCoverageInputRep

Invocable action input representation details to verify the coverage of a specific insurance policy.

Property	Type	Description	Required or Optional	Available Version
coverageId	String	Insurance policy coverage ID.	Required	65.0

Property	Type	Description	Required or Optional	Available Version
<code>insuredAssetParticipantId</code>	String	ID of the insured asset or participant that's related to the policy coverage.	Optional	65.0
<code>lossDate</code>	String	Date on which the incident occurred.	Optional	65.0
<code>policyId</code>	String	ID of the insurance policy.	Required	65.0

ConnectApi Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

All properties are read-only, except for instances of output classes created within test code.

All output classes are concrete unless marked abstract in this documentation.

All concrete output classes have no-argument constructors that you can invoke only from test code. See [Testing ConnectApi Code](#).

Insurance Claim includes these Apex output classes.

[ConnectApi.CalculateAdjustmentsRepresentation](#)

Output representation of the details for calculating the adjustments for the claim loss item.

[ConnectApi.CancelClaimCoveragePaymentDetailRep](#)

Output representation for the details of the cancel claim coverage payment detail record.

[ConnectApi.ClaimAttributeOutputRepresentation](#)

Output representation for a claim attribute.

[ConnectApi.ClaimCovPaymentAdjDetailsRep](#)

Output representation for the claim coverage payment adjustment details.

[ConnectApi.ClaimCoverageRepresentation](#)

Output representation for a claim coverage operation.

[ConnectApi.ClaimDetailOutputRep](#)

Invocable action output representation for claim details.

[ConnectApi.ClaimItemOutputRepresentation](#)

Output representation for a claim item.

[ConnectApi.ClaimItemOutputRep](#)

Invocable action output representation for claim item.

[ConnectApi.ClaimItemOutputs](#)

Invocable action output representation for a wrapper that contains a list of claim items.

[ConnectApi.ClaimOutputRepresentation](#)

Output representation for a retrieved claim.

[ConnectApi.ClaimParticipantOutputRepresentation](#)

Output representation for a claim participant.

[ConnectApi.ClaimParticipantOutputRep](#)

Invocable action output representation for claim participant.

[ConnectApi.ClaimParticipantOutputs](#)

Invocable action output representation for a wrapper that contains a list of claim participants.

[ConnectApi.ClaimRelatedObjectOutputRepresentation](#)

Output representation for a related object.

[ConnectApi.CreateClaimCoveragePaymentDetailRep](#)

Output representation for the details of the create claim coverage payment detail record.

[ConnectApi.CreateClaimOutputRepresentation](#)

Output representation for a create or update claim operation.

[ConnectApi.CreatePolicyLimitsRepresentation](#)

Output representation of the created insurance policy limit records.

[ConnectApi.DeleteClaimCoveragePaymentDetailRep](#)

Output representation for the Delete Claim Coverage Payment Detail resource.

[ConnectApi.EditClaimCoveragePaymentDetailRep](#)

Output representation for editing claim coverage payment detail.

[ConnectApi.ErrorDetail](#)

Output representation of an error.

[ConnectApi.GetClaimOutputRepresentation](#)

Output representation for the Get Claim action.

[ConnectApi.GetInsPlcyCurrStndOutputRep](#)

An Apex ConnectApi.GetInsPlcyCurrStndOutputRep record that contains the details of the consumed and available limits that are retrieved for the policy or claim.

[ConnectApi.IPLStandingCalcAmounts](#)

Invocable action output representation for insurance policy limit standing calculation amounts.

[ConnectApi.IPLStandingCalcCoverageInfo](#)

Output representation for insurance policy limit standing calculation.

[ConnectApi.IPLStandingCalcLimitAttributes](#)

Invocable action output representation for limit attributes.

[ConnectApi.IPLStandingCalcOutputNode](#)

Node representation for the Insurance Policy Term Current Standing invocable action node.

[ConnectApi.IPLStandingCalcValues](#)

Invocable Action Output representation for IPL Standing Calculation Values

[ConnectApi.IPLStandingMappedScopes](#)

Invocable action output representation for insurance policy limit standing mapped scopes.

[ConnectApi.InsPolicyLimitTrackingDetailsRep](#)

Output representation for the insurance policy limit tracking details.

[ConnectApi.InsPolicyLmtCalcStandingRepresentation](#)

Output representation for the current standings of the insurance policy limits.

[ConnectApi.InsPolicyLmtCalcStandingNode](#)

Node representation for an insurance policy limit standing calculation.

[ConnectApi.PayClaimCoveragePaymentDetailRep](#)

Output representation for a claim coverage payment detail operation.

[ConnectApi.PayExGratiaClaimCvrPaymentDetailRep](#)

Output representation for the details of the ex gratia payment that's made for the claim coverage payment detail.

[ConnectApi.ProcessPolicyLimitsRepresentation](#)

Output representation for processing the policy limits for a claim loss item.

[ConnectApi.VerifyPolicyCoverageRepresentation](#)

Output representation of a policy coverage's applicability for a specified loss date.

[ConnectApi.VerifyPolicyCoverageOutputRep](#)

Invocable action output representation for verify policy coverage.

ConnectApi.CalculateAdjustmentsRepresentation

Output representation of the details for calculating the adjustments for the claim loss item.

Property Name	Type	Description	Available Version
adjustedAmount	Double	Amount after adjustments are made based on the current standings of the policy limits.	65.0
adjustmentReason	String	Reason for the claim coverage payment detail adjustment, such as deductible, copay, or coinsurance.	65.0
claimCoverageAdjustments	ConnectApi.ClaimCoverageAdjustmentRep on page 230>	List of claim coverage payment adjustment details.	65.0
errors	ConnectApi.ErrorDetail on page 235>	List of errors that are associated with the API request.	65.0
insPolicyLimitTrackings	ConnectApi.InsPolicyLimitTrackingRep on page 238>	List of insurance policy limit tracking details.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.CancelClaimCoveragePaymentDetailRep

Output representation for the details of the cancel claim coverage payment detail record.

Property Name	Type	Description	Available Version
errors	ConnectApi.ErrorDetail on page 235>	Details of the errors that are associated with the API request.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0
paymentDetailId	String	ID of the claim coverage payment detail for which the payment is canceled.	65.0

ConnectApi.ClaimAttributeOutputRepresentation

Output representation for a claim attribute.

Property Name	Type	Description	Available Version
apiName	String	API name of the attribute.	65.0
value	String	Value for the claim attribute.	65.0

ConnectApi.ClaimCovPaymentAdjDetailsRep

Output representation for the claim coverage payment adjustment details.

Property Name	Type	Description	Available Version
adjustedAmount	Double	Amount deducted from the Posted Amount based on the respective attribute category.	65.0
adjustmentReason	String	Reason for the claim coverage payment adjustment, such as deductible, copay, or coinsurance.	65.0

ConnectApi.ClaimCoverageRepresentation

Output representation for a claim coverage operation.

Property Name	Type	Description	Available Version
coverageId	String	ID of the claim coverage record.	65.0
errors	like <code>ConnectApi.ErrorDetail</code> on page 235>	Details of the errors that are associated with the API request.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.ClaimDetailOutputRep

Invocable action output representation for claim details.

Property Name	Type	Description	Available Version
accountId	String	ID of the associated account.	65.0
attributes	ConnectApi.ClaimAttributeOutputs	Attributes of the claim product.	65.0
claimReason	String	Reason for which the claim is raised.	65.0
claimReasonType	String	Type of reason for the claim.	65.0
claimType	String	Type of claim.	65.0
incidentTypeId	String	ID for the type of incident in the claim.	65.0

Property Name	Type	Description	Available Version
lossType	String	Type of loss for which the claim is initiated.	65.0

ConnectApi.ClaimItemOutputRepresentation

Output representation for a claim item.

Property Name	Type	Description	Available Version
additionalFields	Object	Additional fields of the claim item.	65.0
attributes	ConnectApi.ClaimItemOutputRepresentation on page 230>	Attributes of the claim item.	65.0
category	String	Category of the claim item.	65.0
id	String	ID of the claim item.	65.0
instanceKey	String	Unique instance key of the claim item.	65.0
insurancePolicyCoverageId	String	ID of the insurance policy coverage.	65.0
insuredItemId	String	ID of the insured item involved.	65.0
name	String	Name of the claim item.	65.0
participantInstanceKey	String	Instance key of the claim participant.	65.0
productCode	String	Product code of the claim item.	65.0
relatedObjects	ConnectApi.ClaimItemOutputRepresentation on page 233>	Claim item related objects.	65.0

ConnectApi.ClaimItemOutputRep

Invocable action output representation for claim item.

Property Name	Type	Description	Available Version
category	String	Category of the claim item.	65.0
claimItemDetails	ConnectApi.ClaimItemDetailOutputRep	Details of the claim items.	65.0
id	String	ID of the claim item.	65.0
instanceKey	String	Unique instance key of the claim item.	65.0
insuredItemId	String	ID of the insured item involved.	65.0
name	String	Name of the claim item.	65.0
productCode	String	Product code of the claim item.	65.0

ConnectApi.ClaimItemOutputs

Invocable action output representation for a wrapper that contains a list of claim items.

Property Name	Type	Description	Available Version
claimItems	ConnectApi.ClaimItemOutputRepresentation	List of claim items.	65.0

ConnectApi.ClaimOutputRepresentation

Output representation for a retrieved claim.

Property Name	Type	Description	Available Version
accountId	String	ID of the associated account.	65.0
attributes	ConnectApi.ClaimAttributeRepresentation	Attributes of the claim product.	65.0
claimId	String	ID of the claim.	65.0
claimLossDate	String	Date when the loss occurred. This parameter doesn't show the time of incident.	66.0
claimReason	String	Reason for which the claim is raised.	65.0
claimReasonType	String	Type of reason for the claim.	65.0
claimType	String	Type of claim.	65.0
contextId	String	ID of the context.	65.0
errors	ConnectApi.ErrorDetail on page 235>	Errors that are encountered during the processing of the API request.	65.0
incidentTypeId	String	ID for the type of incident in the claim.	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	65.0
items	ConnectApi.ClaimItemOutputRepresentation on page 231>	Details of the claim items.	65.0
lossDate	Date	Date on which the incident took place.	65.0
lossType	String	Type of loss for which the claim is initiated.	65.0
name	String	Name of the claim.	65.0
participants	ConnectApi.ClaimParticipantOutputRepresentation on page 232>	Details of the claim participants.	65.0
productCode	String	Product code of the claim product.	65.0

ConnectApi.ClaimParticipantOutputRepresentation

Output representation for a claim participant.

Property Name	Type	Description	Available Version
accountId	String	Account ID of the claim participant.	65.0
contactId	String	Contact ID of the claim participant.	65.0
id	String	ID of the claim participant.	65.0
instanceKey	String	Instance key of the claim participant.	65.0
insurancePolicyParticipantId	String	Insurance policy participant ID of the claim participant.	65.0
roles	List<String>	Claim participant roles.	65.0

ConnectApi.ClaimParticipantOutputRep

Invocable action output representation for claim participant.

Property Name	Type	Description	Available Version
accountId	String	Account ID of the claim participant.	65.0
contactId	String	Contact ID of the claim participant.	65.0
id	String	ID of the claim participant.	65.0
instanceKey	String	Instance key of the claim participant.	65.0
insurancePolicyParticipantId	String	Insurance policy participant ID of the claim participant.	65.0
roles	List<String>	Roles of the claim.	65.0

ConnectApi.ClaimParticipantOutputs

Invocable action output representation for a wrapper that contains a list of claim participants.

Property Name	Type	Description	Available Version
claimParticipants	List<ConnectApi.ClaimParticipantOutputRep>	List of claim participants.	65.0

ConnectApi.ClaimRelatedObjectOutputRepresentation

Output representation for a related object.

Property Name	Type	Description	Available Version
relatedRecordId	String	ID of the related object.	65.0
relatedRecordObjName	String	API name of the related object.	65.0

ConnectApi.CreateClaimCoveragePaymentDetailRep

Output representation for the details of the create claim coverage payment detail record.

Property Name	Type	Description	Available Version
adjustedAmount	Double	Adjusted amount of the claim coverage payment detail.	65.0
adjustedReason	String	Reason for the adjustment.	65.0
errors	link to ConnectApi.ErrorDetail on page 235	List of errors, including detailed information about each error.	65.0
insurancePolicyLimitIds	List<String>	IDs of the insurance policy limit tracking records.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or (<code>not</code>).	65.0
paymentDetailId	String	ID of the claim coverage payment detail record that's created.	65.0

ConnectApi.CreateClaimOutputRepresentation

Output representation for a create or update claim operation.

Property Name	Type	Description	Available Version
claimId	String	ID of the created or updated claim.	65.0
errors	link to ConnectApi.ErrorDetail on page 235	Details of the errors that are associated with the API request.	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.CreatePolicyLimitsRepresentation

Output representation of the created insurance policy limit records.

Property Name	Type	Description	Available Version
errors	link to ConnectApi.ErrorDetail on page 235	Errors that are encountered during the processing of the API request.	65.0
policyLimitIds	List<String>	IDs for the insurance policy limit records that are created.	65.0

ConnectApi.DeleteClaimCoveragePaymentDetailRep

Output representation for the Delete Claim Coverage Payment Detail resource.

Property Name	Type	Description	Available Version
errors	link<ConnectApi.ErrorDetail>	Details of the errors that are associated with the API request.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0
paymentDetailId	String	ID of the claim coverage payment detail record.	65.0

ConnectApi.EditClaimCoveragePaymentDetailRep

Output representation for editing claim coverage payment detail.

Property Name	Type	Description	Available Version
adjustedAmount	Double	Final amount after all deductions, such as deductibles or co-pays. This field is returned when a Loss item is created and adjustments are applied.	65.0
adjustedReason	String	Reason for the claim coverage payment detail adjustment, such as deductible, copay, or coinsurance.	65.0
errors	link<ConnectApi.ErrorDetail> on page 235	List of errors, including detailed information about each error when the API operation fails.	65.0
insurancePolicyLimitTrackingIds	List<String>	Comma-separated IDs of the insurance policy limit tracking records.	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.ErrorDetail

Output representation of an error.

Property Name	Type	Description	Available Version
code	String	Code of the error.	65.0
message	String	Message of the error.	65.0

ConnectApi.GetClaimOutputRepresentation

Output representation for the Get Claim action.

Property Name	Type	Description	Available Version
claimDetails	ConnectApi.ClaimDetailOutputRep	Details of the claim.	65.0

Property Name	Type	Description	Available Version
claimId	String	ID of the claim.	65.0
contextId	String	ID of the context that's associated with the claim.	65.0
insurancePolicyId	String	ID of the insurance policy on which the claim is raised.	65.0
lossDate	String	Date on which the incident took place.	65.0
name	String	Name of the claim.	65.0
productCode	String	Product code of the claim product.	65.0

ConnectApi.GetInsPclyCurrStndOutputRep

An Apex ConnectApi.GetInsPclyCurrStndOutputRep record that contains the details of the consumed and available limits that are retrieved for the policy or claim.

Property Name	Type	Description	Available Version
iplStandingCalcOutputNodes	ConnectApi.IPLStandingCalcOutputNode	List of policy limit current standing nodes.	65.0

ConnectApi.IPLStandingCalcAmounts

Invocable action output representation for insurance policy limit standing calculation amounts.

Property Name	Type	Description	Available Version
initialAmount	Double	Configured amount.	65.0
pendingAmount	Double	Amount that's pending.	65.0
predictedRemainingAmount	Double	Remaining amount to carry forward for the next payment after calculating the pending amount.	65.0
remainingAmount	Double	Remaining amount to carry forward for the next payment.	65.0
totalUsedAmount	Double	Total amount that's used till date.	65.0

ConnectApi.IPLStandingCalcCoverageInfo

Output representation for insurance policy limit standing calculation.

Property Name	Type	Description	Available Version
claimCoverageId	String	ID of the related claim coverage.	65.0
claimId	String	ID of the related claim.	65.0
policyAssetId	String	ID of the related policy asset.	65.0
policyCoverageId	String	ID of the related policy coverage.	65.0

Property Name	Type	Description	Available Version
policyParticipantId	String	ID of the related policy participant.	65.0
subCoverageName	String	Name of the related sub coverage.	65.0

ConnectApi.IPLStandingCalcLimitAttributes

Invocable action output representation for limit attributes.

Property Name	Type	Description	Available Version
applicableActions	List<String>	List of actions that initiate the attribute processing.	65.0
category	String	Category of the limit.	65.0
durationType	String	Duration type over which the attribute consumption is actively monitored during processing.	65.0
itemType	String	Type of the item.	65.0
policyLimitId	String	ID of the insurance policy limit record.	65.0
policyLimitStandingKey	String	Unique identifier for the insurance policy limit standing.	65.0
scope	String	Scope where the limit is applied.	65.0

ConnectApi.IPLStandingCalcOutputNode

Node representation for the Insurance Policy Term Current Standing invocable action node.

Property Name	Type	Description	Available Version
amounts	ConnectApi.IPLStandingCalcAmounts	Details of the calculation amounts.	65.0
coverageInfo	ConnectApi.IPLStandingCalcCoverageInfo	Details of the claim coverage.	65.0
limitAttributes	ConnectApi.IPLStandingCalcLimitAttributes	Attribute details of the insurance policy limit.	65.0
mappedScopes	Map<String, ConnectApi.IPLStandingCalcTypes >	Details of the mapped scopes.	65.0
values	ConnectApi.IPLStandingCalcValues	Calculation values.	65.0

ConnectApi.IPLStandingCalcValues

Invocable Action Output representation for IPL Standing Calculation Values

Property Name	Type	Description	Available Version
initialValue	Integer	Configured value of the limit.	65.0
pendingValue	Integer	Value that's pending.	65.0

Property Name	Type	Description	Available Version
predictedRemainingValue	Integer	Remaining value to carry forward for the next payment after calculating the pending value.	65.0
remainingValue	Integer	Remaining value to carry forward for the next payment.	65.0
totalUsedValue	Integer	Total value used till date.	65.0

ConnectApi.IPLStandingMappedScopes

Invocable action output representation for insurance policy limit standing mapped scopes.

Property Name	Type	Description	Available Version
parentMappedScopeId	String	ID of the parent product attribute mapped scope that's associated with the product attribute to define parent-child relationship between attribute mappings.	65.0
prodAttrMappedScopeId	String	ID of the product attribute mapped scope that defines how a product attribute is associated with a specific scope, such as a product type or category.	65.0

ConnectApi.InsPolicyLimitTrackingDetailsRep

Output representation for the insurance policy limit tracking details.

Property Name	Type	Description	Available Version
benefitName	String	Name of the benefit that's associated with the policy limit tracking.	65.0
calcInitialAmount	Double	Calculated currency amount that's used as the base value for tracking policy attribute consumption.	65.0
calcInitialValue	Integer	Count that's used as the base for tracking policy attribute consumption.	65.0
category	String	Category of the associated attribute.	65.0
policyLimitId	String	ID of the insurance policy limit record.	65.0
postedAmount	Double	Amount that's requested by the claimant when submitting the loss.	65.0
postedValue	Integer	Count that's populated when submitting the loss.	65.0
remainingAmount	Double	Remaining amount to carry forward for the next loss processing.	65.0
remainingValue	Integer	Remaining value to carry forward for the next loss processing.	65.0

Property Name	Type	Description	Available Version
scope	String	Scope for tracking the associated attribute.	65.0
totalUsedAmount	Double	Total amount that's used to date.	65.0
totalUsedValue	Integer	Total value that's used to date.	65.0

ConnectApi.InsPolicyLmtCalcStandingRepresentation

Output representation for the current standings of the insurance policy limits.

Property Name	Type	Description	Available Version
errors	List<ConnectApi.ErrorDetail> on page 235>	Errors that are encountered during the processing of the API request.	65.0
insPolicyLmtCalcStandingNodes	ConnectApi.InsPolicyLmtCalcStandingNode on page 239>	List of policy limit current standing nodes.	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.InsPolicyLmtCalcStandingNode

Node representation for an insurance policy limit standing calculation.

Property Name	Type	Description	Available Version
applicableActions	List<String>	List of actions that initiate attribute processing.	65.0
category	String	Category of the limit.	65.0
claimCoverageId	String	ID of the related claim coverage.	65.0
claimId	String	ID of the related claim.	65.0
durationType	String	Duration type over which the attribute consumption is actively monitored during processing.	65.0
initialAmount	Double	Configured amount.	65.0
initialValue	Integer	Configured value of the limit.	65.0
itemType	String	Type of the item.	65.0
parentMappedScopeId	String	ID of the parent product attribute mapped scope that's associated with the product attribute to define a parent-child relationship between attribute mappings.	65.0
pendingAmount	Double	Amount that's pending.	65.0
pendingValue	Integer	Value that's pending.	65.0

Property Name	Type	Description	Available Version
policyAssetId	String	ID of the related policy asset.	65.0
policyCoverageId	String	ID of the related policy coverage.	65.0
policyLimitId	String	ID of the insurance policy limit record.	65.0
policyLimitStandingKey	String	Unique identifier for the insurance policy limit standing.	65.0
policyParticipantId	String	ID of the related policy participant.	65.0
predictedRemainingAmount	Double	Remaining amount to carry forward for the next payment after calculating the pending amount.	65.0
predictedRemainingValue	Integer	Remaining value to carry forward for the next payment after calculating the pending value.	65.0
prodAttrMappedScopeId	String	ID of the product attribute mapped scope that defines how a product attribute is associated with a specific scope, such as a product type or category.	65.0
remainingAmount	Double	Remaining amount to carry forward for the next payment.	65.0
remainingValue	Integer	Remaining value to carry forward for the next payment.	65.0
scope	String	Scope where the limit is applied.	65.0
subCoverageName	String	Name of the sub-coverage.	65.0
totalUsedAmount	Double	Total amount that's used to date.	65.0
totalUsedValue	Integer	Total value that's used to date.	65.0
trackingEndDate	Date	End date of the duration period that's used to calculate the current standing.	66.0
trackingStartDate	Date	Start date of the duration period that's used to calculate the current standing.	66.0

ConnectApi.PayClaimCoveragePaymentDetailRep

Output representation for a claim coverage payment detail operation.

Property Name	Type	Description	Available Version
errors	link to ConnectApi.ErrorDetail on page 235	Details of the errors that are associated with the API request.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.PayExGratiaClaimCvrPaymentDetailRep

Output representation for the details of the ex gratia payment that's made for the claim coverage payment detail.

Property Name	Type	Description	Available Version
claimPaymentSummaryId	String	ID of the claim payment summary record that's added or updated. This ID links individual claim coverage payment details to the corresponding summary of payments that are processed for a claim.	66.0
consumedLimitCount	Integer	Total number of coverage units or limit counts that are utilized by this service or transaction.	66.0
errors	link to ConnectApi.ErrorDetail on page 235	Error details that are shown when the API operation fails.	66.0
exceededAmount	Double	Excess amount that's paid above the limits.	66.0
exceededCount	Integer	Number of coverage units or counts that are paid above the policy limits. This occurs when payments exceed the allowable limits, such as in discretionary ex gratia payments.	66.0
insPolicyLimitTrackingIds	List<String>	IDs of the insurance policy limit tracking records that are created during the ex gratia payment process.	66.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	66.0
paidAmount	Double	Total amount that's paid towards ex gratia.	66.0

ConnectApi.ProcessPolicyLimitsRepresentation

Output representation for processing the policy limits for a claim loss item.

Property Name	Type	Description	Available Version
errors	link to ConnectApi.ErrorDetail on page 235	Details of the errors that are associated with the processing of the API request.	65.0
exceededAmount	Double	Amount that exceeds the remaining available policy limit in the specified limit currency.	66.0
exceededCount	Integer	Count of coverage units beyond the policy's allowed unit limit.	66.0
insPolicyLimitTrackings	link to ConnectApi.PolicyLimitTrackingDetail on page 238	List of insurance policy limit tracking details.	65.0
isLimitExceeded	Boolean	Indicates whether a limit is exceeded (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.VerifyPolicyCoverageRepresentation

Output representation of a policy coverage's applicability for a specified loss date.

Property Name	Type	Description	Available Version
errors	Link to ConnectApi.ErrorDetail on page 235	Errors that are encountered during the processing of the API request.	65.0
isSuccess	Boolean	Indicates whether the policy coverage is valid for the date on which the incident occurred (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.VerifyPolicyCoverageOutputRep

Invocable action output representation for verify policy coverage.

Property Name	Type	Description	Available Version
errors	Link to ConnectApi.ErrorDetailOutputRep on page 235	List of errors that are encountered during the API operation.	65.0
isSuccess	Boolean	Indicates whether the policy coverage is valid for the date on which the incident occurred (<code>true</code>) or not (<code>false</code>).	65.0

Insurance Claims Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Claims](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Claims

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Claim exposes additional `actionType` values for the `FlowActionCall` Metadata type.

Field Name	Field Type	Description
actionType	InvokableActionType (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Claims include: • <code>calculateAdjustments</code>—Calculate the adjusted amount for a loss item and generate data for Insurance Policy Limit Tracking and Claim Coverage Payment Adjustment records. • <code>cancelClaimCvrPaymentDetail</code>—Cancel payment for a claim coverage payment detail record. • <code>createClaim</code>—Create a claim. • <code>createClaimCvrPaymentDetail</code>—Create a claim coverage payment detail record. • <code>createPolicyLimits</code>—Create insurance policy limit records for policy coverage and root attributes. • <code>deleteClaimCvrPaymentDetail</code>—Remove unpaid claim coverage payment details by updating the status of the claim coverage payment detail record to <code>Deleted</code>. • <code>getClaim</code>—Get the details of a claim. • <code>getInsPolicyCurrentStandings</code>—Get the current standings of the insurance policy. • <code>openClaimCoverage</code>—Open a claim coverage. • <code>payClaimCvrPaymentDetail</code>—Initiate the payment for a claim coverage payment detail record, track the policy limit consumption for the payout, and update the status of the claim coverage payment detail record status to <code>Paid</code> upon successful completion. • <code>updateClaim</code>—Update a claim. • <code>updateClaimCoverage</code>—Update a claim coverage. • <code>verifyPolicyCoverage</code>—Verify whether a policy coverage is applicable for a specified loss date under an insurance policy, so that only eligible coverages are processed for claims.

Insurance Claims Developer Journey

Learn how to effectively use the Insurance Claims Connect APIs, along with the associated invocable actions, to automate end-to-end claims processing workflows, from initial coverage verification to final payment settlement or cancellation by using Salesforce Flows and Omniscripts.

This guide walks you through the complete claims lifecycle, explaining:

- The business scenario behind a claim.
- The sequence of API calls across the Claims and Financial domains.
- The data states and system transitions at each step.
- The standard claim resolution flow for successful payments.

- The exception paths for voiding or canceling payments.

Key TERMS	DESCRIPTIONS
Coverage Validation	Confirms policy status, coverage type, and claim eligibility.
Claim Coverage Payment Detail (CCPD)	The payment record that’s created for each coverage-level payment request.
Claim Status	Represents lifecycle stages, such as Submitted, Pending Review, Approved, and Paid.
Deductible	The portion of loss the policyholder pays before insurer coverage applies.
Limit	The maximum payout that’s allowed for a policy coverage type.

Claims Processing Journey

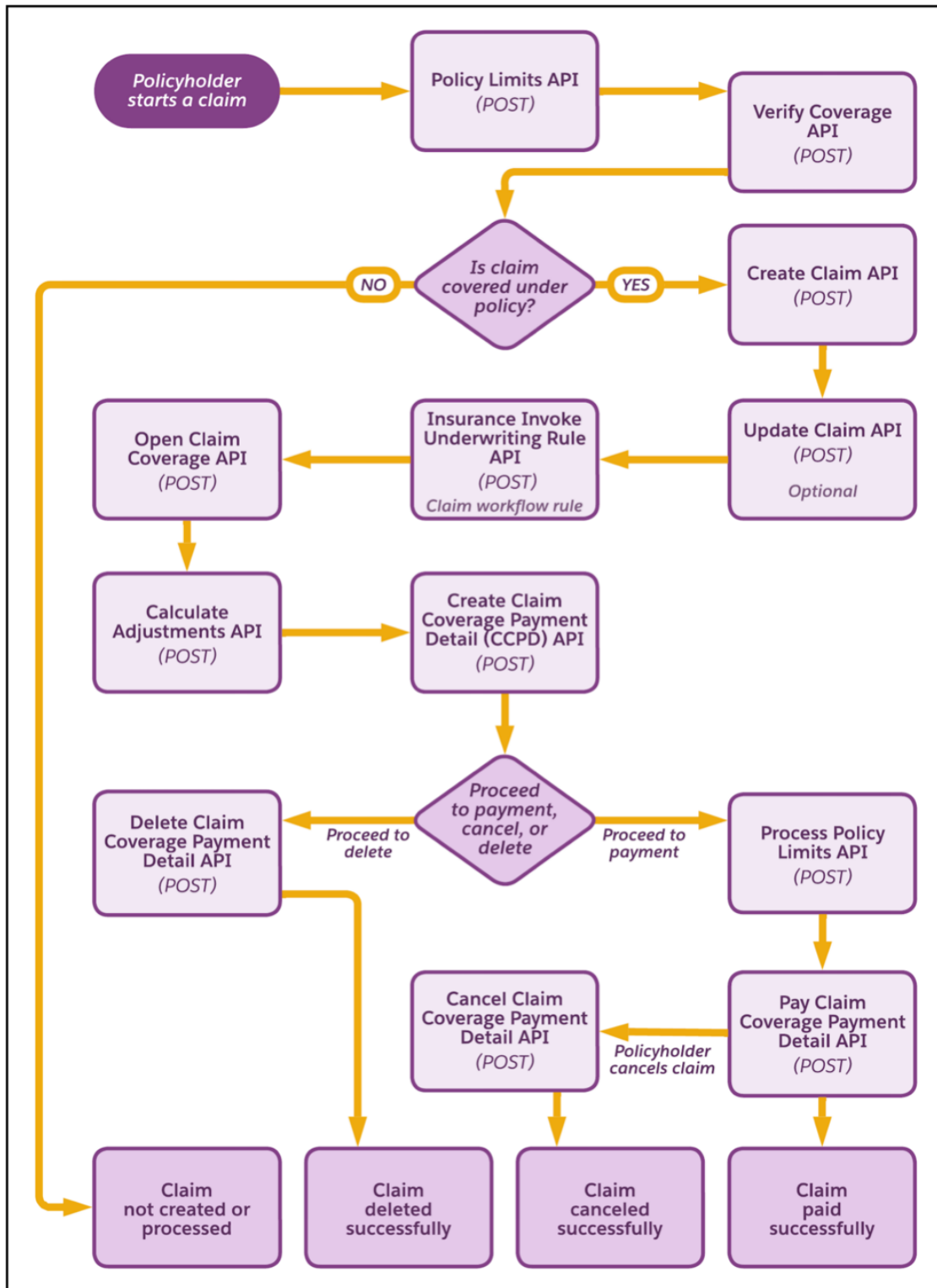
A claim is a formal request submitted by a policyholder to receive benefits or compensation for a covered loss under an active insurance policy. A claim journey begins when the policyholder reports an incident, such as damage, injury, or another qualifying event that they believe is covered by their policy.

Claims Processing Journey

A claim is a formal request submitted by a policyholder to receive benefits or compensation for a covered loss under an active insurance policy. A claim journey begins when the policyholder reports an incident, such as damage, injury, or another qualifying event that they believe is covered by their policy.

A valid claim requires an active insurance policy, a reported loss date, and an incident that falls within the policy's covered benefits. Once initiated, the claim process includes creating policy limits for the policyholder, verifying policy coverage, creating a claim record, collecting supporting claim items and participant details, routing the claim for adjuster review, assessing eligibility and damages, performing financial calculations, and determining the payout amount. The process concludes with issuing payment for the claim. As an exception, the process also concludes when the customer withdraws the claim or when the payment is voided because it was initiated based on inaccurate data.

This diagram outlines the sequence of operations that are involved in the claim process.



1. Create Policy Limits—Generate a record of the trackable policy or coverage attributes, such as deductibles and limits, and store the record details in a separate entity. You can create policy limits by either:

- Making a POST request on the [Policy Limits](#) Connect API, or

- Using the [Create Policy Limits](#) invocable action.

Result: The system retrieves the financial rules for the policyholder's specific policy, including deductible and coverage limits.

2. Verify Policy Coverage—Validate whether the policy covers the type of claim being submitted by either:

- Making a POST request on the [Verify Policy Coverage](#) Connect API, or
- Using the [Verify Coverage](#) invocable action.

Specify the `policyId` and `coverageId` during the API call.

Result: The system returns coverage details indicating whether the claim is eligible under the policy's coverage rules or not.

3. Create Claim—Create a new claim for the covered incident, including related details, such as involved items and participants. Create a claim by either:

- Making a POST request on the [Claim](#) Connect API, or
- Using the [Create Claim](#) invocable action.

Result: A unique claim ID is generated. The claim is submitted for review to a claims adjuster.

4. (Optional) Update Claim—Update an existing claim, including its items and participant details by either:

- Making a PATCH request on the [Claim](#) Connect API, or
- Using the [Update Claim](#) invocable action.

Result: The claim record is updated, and its status changes from New to Pending Review.

5. Invoke Claim Workflow Rules—Route the claim for automated or manual review by triggering the underwriting API by either:

- Making a POST request on the [Insurance Invoke Underwriting Rule](#) Connect API, or
- Using the [Invoke Underwriting Rules](#) invocable action.

Result: The system evaluates claim parameters, such as claim amount and type, and assigns it to a claims adjuster or workflow queue.

6. Open Claim Coverage—Open a claim coverage record that tracks the coverage-level reserves, payments, and other financial transactions, and maps them to the corresponding insurance policy coverage, asset, or participant. Open a claim coverage by either:

- Making a POST request on the [Claim Coverage](#) Connect API, or
- Using the [Open Claim Coverage](#) invocable action.

Result: The claim coverage record is created that's used to store and track the associated payment details for the claim.

7. Calculate Adjustments—Calculate how much the insurer must pay after applying adjustments to the policy's deductible, copay, coinsurance, and out-of-pocket maximums (OOPM). Calculate the adjusted amount by either:

- Making a POST request on the [Calculate Adjustments](#) Connect API, or
- Using the [Calculate Adjustments](#) invocable action.

Result: Financial details are computed and made available for payment generation.

8. Create Claim Coverage Payment Detail (CCPD)—Create a claim coverage payment detail record capturing the amount requested by the claimant. The record is used to track payment requests and associated claim coverage details. Create a CCPD by either:

- Making a POST request on the [Create Claim Coverage Payment Detail](#) Connect API, or
- Using the [Create Claim Coverage Payment Detail](#) invocable action.

Result: A claim coverage payment detail (CCPD) record is created for the calculated payout amount.

Standard Claim Resolution Flow (Successful Payment)

When the reviewer approves the claim, use this sequence to process and complete payment.

- **(Optional) Validate Payment Limits**—Before the payment is executed, the system must validate that the payout amount is within the policy's applicable coverage limits, such as the collision limit or other coverage-specific maximums. This prevents overpayments and protects the insurer from financial leakage.

Make a POST request to the [Process Policy Limits](#) Connect API to ensure the payout is within policy limits.

- **Execute Payment Transaction** Initiate the payment process by either:
 - Making a POST request on the [Pay Claim Coverage Payment Detail](#) Connect API, or
 - Using the [Pay Claim Coverage Payment Detail](#) invocable action.



Note: When the Pay Claim Coverage Payment Detail Connect API is executed, it automatically invokes the Process Policy Limits Connect API to check the correctness of the payout amount.

Result: The claim payment is processed successfully, and the policyholder receives a payment confirmation notification. The claim is marked Closed.

Exception Flows

Void Payment Record—The reviewer approves the claim based on inaccurate data. Before payment processing, the admin must void the incorrect CCPD record that's already created. Invalidate the payment record by either:

- Making a POST request on the [Delete Claim Coverage Payment Detail](#) Connect API, or
- Using the [Delete Claim Coverage Payment Detail](#) invocable action.

The payment instruction is permanently deleted from the financial system, and the claim must be re-evaluated.

Cancel Payment—After the Pay CCPD record is created and payment is initiated, policyholder withdraws the claim before payout. Cancel the payment by either:

- Making a POST request on the [Cancel Claim Coverage Payment Detail](#) Connect API, or
- Using the [Cancel Claim Coverage Payment Detail](#) invocable action.

The payment instruction is formally canceled. The claim is logged as Canceled at Customer's Request and closed with status Withdrawn. Any policy-related financial updates are rolled back, and the original policyholder information remains unchanged.

CHAPTER 4 Insurance Group Benefits

In this chapter ...

- [Insurance Group Benefits Business APIs](#)
- [Insurance Group Benefits Apex Reference](#)
- [Insurance Group Benefits Standard Invocable Actions](#)
- [Insurance Group Benefits Metadata API Types](#)

Group benefits are insurance plans offered to and covering a group of people, such as employees and their dependents. Manage these insurance offerings in a rapidly evolving marketplace with Group Benefits on the Digital Insurance Platform.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Insurance Group Benefits Business APIs

Use the Group Benefits API resources to effectively manage the insurance plans offered to groups, such as employees and their dependents.

Available in: Lightning Experience

Available in: **Professional**, **Enterprise**, and **Unlimited** Editions of Digital Insurance with Digital Insurance Product Admin Add On, Digital Insurance Group Benefits AddOn, Digital Insurance Group Enrollment AddOn, Digital Insurance Group Benefits Customer Community AddOn, and Digital Insurance Group Benefits Partner Community AddOn licenses

[Resources](#)

Learn more about the available insurance group benefits API resources.

[Requests](#)

Learn more about the available requests of insurance group benefits APIs.

[Responses](#)

Learn more about the available response bodies of insurance group benefits APIs.

Resources

Learn more about the available insurance group benefits API resources.

[Insurance Person Accounts and Users \(POST\)](#)

Create person accounts and portal users for group census members. The Person Account is used for linking to the appropriate policy for policy association, and the corresponding portal user is created to allow the member to log in to the community portal.

[Insurance Contacts and Users \(POST\)](#)

Create contacts and portal users for group census members. The Contact record is used for linking to the appropriate policy for policy association, and the corresponding portal user is created to allow the member to log in to the community portal.

[Insurance Member Rating \(POST\)](#)

Calculate pricing along with employee and employer contribution for individual members of a group census based on plan and coverages within the plan.

[Insurance Group Census Members \(POST, PATCH, DELETE\)](#)

Add (POST) and update (PATCH) group census members. Delete all census members for the specified group census ID.

[Insurance Group Census Members Delete \(POST\)](#)

Delete one or more census members of the specified group census ID.

[Insurance Bulk Census Management \(POST\)](#)

Process group census data in bulk.

[Insurance Census Summary Sync \(POST\)](#)

Synchronize and recalculate the summary data for a group census.

[Insurance Member Plans \(POST, PATCH\)](#)

Create and update insurance plans for a group census member.

[Insurance Member Plans Delete \(POST\)](#)

Delete specified plans for a group census member.

[Insurance Member Eligible Plans \(GET\)](#)

Get the contract group plans for which a group census member and their dependents are eligible.

[Insurance Plans Eligibility Check \(POST, PATCH\)](#)

Evaluate product configuration rules for a group census member based on their selected plan and coverage options during initial enrollment (POST). Reevaluate plan eligibility for a group census member when plans are added or removed (PATCH).

[Insurance Contract from Quote \(POST\)](#)

Create a group benefits contract from a quote. This API supports asynchronous operation for long-running conversions.

[Insurance Group Rating \(POST\)](#)

Rate a group benefits quote to calculate premiums for all quote line items. Rate specific root plans to calculate premiums for their associated quote line items.

[Individual Enrollment \(POST\)](#)

Enroll a group census member in a group benefit insurance plan, creating the necessary policy records.

SEE ALSO:

[Salesforce Help: Explore Group Benefits in Digital Insurance](#)

Insurance Person Accounts and Users (POST)

Create person accounts and portal users for group census members. The Person Account is used for linking to the appropriate policy for policy association, and the corresponding portal user is created to allow the member to log in to the community portal.

After the records are created and required permission sets are assigned, members can log into the customer community portal to select plans and complete their individual enrollment based on the digital workflow that's configured by the carrier.

Special Access Rule

To use this API, the org must be enabled with Person Account and you must have Census Management user access.

Resource

```
/connect/insurance/accounts
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/accounts?createPortalUsers=false
```

Available version

65.0

HTTP methods

POST

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
createPortalUsers	Boolean	Indicates whether to create portal users for the new person accounts (<code>true</code>) or not (<code>false</code>). The default value is <code>true</code> .	Optional	65.0

Request body for POST**JSON example**

This is a sample JSON example with GroupCensusId.

```
{
  "groupCensusId": "0rfxx0000000007AAA",
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzx",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

This is a sample JSON example with GroupCensusMembers.

```
{
  "groupCensusMemberIdList": {
    "values": [
      "0r6xx0000000002AAA",
      "0r6xx0000000004rAAA"
    ]
  },
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzx",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

```
}
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldMappings	Map<String, String>	Map of field API names from the Group Census Member object (key) to the Account object (value).	Optional	65.0
groupCensusId	String	ID of the group census that contains the members to process.	Optional	65.0
groupCensusMemberIdList	String[]	IDs of the group census members for whom you want to create person accounts and users. If not provided, all members of the specified group census are processed.	Optional	65.0
matchingKeyList	String[]	List of field API names on the Group Census Member object that are used to identify and prevent the creation of duplicate person accounts. The default fields are FirstName, LastName, PersonBirthdate, PersonEmail, and PersonGender.	Optional	65.0
personAccountRecordType	String	Developer name of the person account record type to use for the new accounts.	Optional	65.0
userDetails	Portal User Creation Input	Details to create portal users.	Required if the <code>createPortalUsers</code> request parameter is set to <code>true</code> .	65.0

Response body for POST

[Insurance Person Accounts and Users Creation Response](#)

SEE ALSO:

[Salesforce Help: Configure Group Benefits Enrollment](#)

Insurance Contacts and Users (POST)

Create contacts and portal users for group census members. The Contact record is used for linking to the appropriate policy for policy association, and the corresponding portal user is created to allow the member to log in to the community portal.

Special Access Rule

To use this API, the org must be enabled with Person Account and you must have Census Management user access.

Resource`/connect/insurance/contacts`**Resource example**`https://yourInstance.salesforce.com/services/data/v/connect/insurance/contacts?createPortalUsers=false`**Available version**

65.0

HTTP methods

POST

Request parameters for POST

Parameter Name	Type	Description	Required or Optional	Available Version
createPortalUsers	Boolean	Indicates whether to create portal users for the new contacts (true) or not (false). The default value is true.	Optional	65.0

Request body for POST**JSON example**

This is a sample JSON example with GroupCensusId.

```
{
  "groupCensusId": "0rfxx0000000007AAA",
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzxC",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

This is a sample JSON example with GroupCensusMembers.

```
{
  "groupCensusMemberIdList": {
    "values": [
      "0r6xx0000000002AAA",
      "0r6xx0000000004rAAA"
    ]
  },
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzx",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldMappings	Map	Map of field API names from the Group Census Member object (key) to the Contact object (value).	Optional	65.0
groupCensusId	String	ID of the group census.	Optional	65.0
groupCensusMemberIdList	String[]	IDs of the group census members for whom you want to create contacts and users. If not provided, all members of the specified group census are processed.	Optional	65.0
matchingKeyList	String[]	List of field API names on the Group Census Member object that are used to identify and prevent the creation of duplicate contact records. The default fields are FirstName, LastName, Birthdate, Email, and Gender.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
userDetails	Portal User Creation Input	Details to create portal users. This property is required if the <code>createPortalUsers</code> parameter in the method call is set to <code>true</code> .	Optional	65.0

Response body for POST

[Insurance Contacts and Users Creation Response](#)

Insurance Member Rating (POST)

Calculate pricing along with employee and employer contribution for individual members of a group census based on plan and coverages within the plan.

The prices and contribution are calculated at coverage level and then rolled up to plan and member level.

Special Access Rule

To use this API, the org must be enabled with Person Account and you must have Census Management user access.

Resource

```
/connect/insurance/group-census-members/memberId/action/rate
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/group-census-members/0r6RM0000000h1pYAA/action/rate
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
memberId	String	ID of the primary group census member for which the rating must be calculated.	Required	65.0

Request body for POST**JSON example**

```
{
  "effectiveDate": "2025-10-01",
  "members": [
    {
      "memberId": "0r6RM0000000h1pYAA",
      "plans": [
        {
```

```
{
  "planId": "0j5RM00000000a1YAA",
  "planCoverageIds": [
    "0k8RM000000001b2ZAA"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the member's rating is calculated.	Optional	65.0
members	Insurance Member Rating Details Input[]	Details of the members and their plans to be rated.	Required	65.0

Response body for POST

[Insurance Member Rating](#)

Insurance Group Census Members (POST, PATCH, DELETE)

Add (POST) and update (PATCH) group census members. Delete all census members for the specified group census ID.

Resource

/connect/insurance/census-management/group-censuses/**groupCensusId**/group-census-members

Resource example

https://~~yourInstance~~.salesforce.com/services/data/v/connect/insurance/census-management/group-censuses/~~0j5RM00000000a1YAA~~/group-census-members

Available version

65.0

HTTP methods

POST, PATCH, DELETE

Path parameter for POST, PATCH, and DELETE

Parameter Name	Type	Description	Required or Optional	Available Version
groupCensusId	String	ID of the group census.	Required	65.0

Request body for POST**JSON example**

```
{
  "members": {
    "values": [
      {
        "name": "member1",
        "firstName": "fm",
        "lastName": "lm",
        "email": "cci@test.com",
        "sourceSystemIdentifier": "1.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
      },
      {
        "name": "member2",
        "firstName": "fm2",
        "lastName": "lm2",
        "email": "cci@test.com",
        "sourceSystemIdentifier": "2.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
      },
      {
        "name": "dep1",
        "firstName": "depfm",
        "lastName": "depfm",
        "email": "cci@test.com",
        "associationWithPrimaryMbr": "Child",
        "relationshipToPrimaryMember": "Child",
        "sourceSystemIdentifier": "1.Child",
        "primaryMemberSource": "1.primary"
      },
      {
        "name": "dep2",
        "firstName": "depfm2",
        "lastName": "depfm2",
        "email": "cci@test.com",
        "associationWithPrimaryMbr": "Child",
        "relationshipToPrimaryMember": "Child",
        "sourceSystemIdentifier": "2.Child",
        "primaryMemberSource": "2.primary"
      }
    ]
  },
  "duplicateDetectionFields": {
    "values": [
      "firstName"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
duplicate Detection Fields	String[]	List of field API names that are used to generate a unique key for identifying duplicate members along with default values. The default fields for duplicate detection are <code>FirstName</code> , <code>LastName</code> , <code>AssociationWithPrimaryMbr</code> , and <code>GroupCensusId</code> .	Optional	65.0
flowName	String	Name of the flow to use for processing group member data. Default value is <code>Process Group Census Member Data</code> .	Optional	65.0
members	Group Census Member Input []	List of group census members to add or update.	Required	65.0

Response body for POST

[Insurance Group Census Member Management](#)

Request body for PATCH**JSON example**

```
{
  "members": {
    "values": [
      {
        "name": "member1",
        "firstName": "fm",
        "lastname": "lm",
        "email": "cci@test.com",
        "sourceSystemIdentifier": "1.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
      },
      {
        "name": "member2",
        "firstName": "fm2",
        "lastname": "lm2",
        "email": "cci@test.com",
        "sourceSystemIdentifier": "2.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
      }
    ]
  }
}
```



```

        {
            "name": "dep1",
            "firstName": "depfm",
            "lastName": "depfm",
            "email": "cci@test.com",
            "associationWithPrimaryMbr": "Child",
            "relationshipToPrimaryMember": "Child",
            "sourceSystemIdentifier": "1.Child",
            "primaryMemberSource": "1.primary"
        },
        {
            "name": "dep2",
            "firstName": "depfm2",
            "lastName": "depfm2",
            "email": "cci@test.com",
            "associationWithPrimaryMbr": "Child",
            "relationshipToPrimaryMember": "Child",
            "sourceSystemIdentifier": "2.Child",
            "primaryMemberSource": "2.primary"
        }
    ]
},
"duplicateDetectionFields": {
    "values": [
        "firstName"
    ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
duplicate Detection Fields	String[]	List of field API names that are used to generate a unique key for identifying duplicate members along with default values. The default fields for duplicate detection are <code>FirstName</code>, <code>LastName</code>, <code>AssociationWithPrimaryMbr</code>, and <code>GroupCensusId</code>.	Optional	65.0
flowName	String	Name of the flow to use for processing group member data. Default value is <code>Process Group Census Member Data</code>.	Optional	65.0
members	Group Census Member Input []	List of group census members to add or update.	Required	65.0

Response body for PATCH[Insurance Group Census Member Management](#)**Response body for DELETE**[Insurance Group Census Member Management](#)

Insurance Group Census Members Delete (POST)

Delete one or more census members of the specified group census ID.

Resource

```
/connect/insurance/census-management/group-censuses/groupCensusId/group-census-members/delete
```

Resource example

```
https://yourinstance.salesforce.com/services/data/v/connect/insurance/census-management/group-censuses/0r6RM0000000h1pYAA0r6RM0000000h1pYAA/group-census-members/delete
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
groupCensusId	String	ID of the group census.	Required	65.0

Request body for POST**JSON example**

```
{
  "memberIds": [
    "0r6RM0000000h1pYAA"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
memberIds	String[]	IDs of the group census member that must be deleted.	Required	65.0

Response body for POST[Insurance Group Census Member Management](#)

Insurance Bulk Census Management (POST)

Process group census data in bulk.

This API efficiently manages large volumes of member data that are linked to a specific group census ID, simplifying the enrollment and update process for group insurance plans. When processing enrollment census data, this API also updates the member plan details.

Resource

```
/connect/insurance/census-management/group-censuses/groupCensusId/group-census-members/bulk-process
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/census-management/group-censuses/01h00000001234/group-census-members/bulk-process
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
groupCensusId	String	ID of the group census.	Required	65.0

Request body for POST

JSON example

```
{
  "batchSize": 200,
  "contentDocumentId": "069SG000003gLazYAE",
  "contentDocumentVersionId": "0688d0000025fA3AAI",
  "duplicateDetectionFields": {
    "values": [
      "FirstName",
      "Lastname"
    ]
  },
  "csvColumnMapping": {
    "data": {
      "First Name": "FirstName",
      "Last Name": "Lastname",
      "Email": "Email",
      "Birthdate": "Birthdate",
      "Gender": "Gender",
      "Relationship to Primary Member": "AssociationWithPrimaryMbr",
      "Medical": "ContractGroupPlan",
      "Dental": "ContractGroupPlan",
      "Group Class": "GroupClassName"
    }
  },
  "shouldExecuteConfigurationRule": false
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of CSV records to process in each batch. The default size for Quote Census is 2000. The default size for Enrollment Census is 25.	Optional	65.0
contentDocumentId	String	ID of the content document (CSV file) to process.	Required	65.0
contentDocumentVersionId	String	ID of the specific version of the content document.	Optional	65.0
csvColumnMapping	Map<String, String>	Map where the key is the CSV column header and the value is the API name of the corresponding field in the GroupCensusMember object.	Required	65.0
duplicateDetectionFields	String[]	List of field API names that are used to generate a unique key for identifying duplicate members along with the default values. The default fields that are used to detect duplication are <code>FirstName</code> , <code>LastName</code> , <code>groupCensusId</code> , and <code>AssociationWithPrimaryMbr</code> .	Optional	65.0
shouldExecuteConfigurationRule	Boolean	Indicates whether to execute the configuration rules after processing the data (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Optional	65.0

Response body for POST

[Bulk Group Census Management](#)

Insurance Census Summary Sync (POST)

Synchronize and recalculate the summary data for a group census.

This API uses the group census ID to synchronize the group census summary and group census group class summary data. This API doesn't require a request payload.

Resource

```
/connect/insurance/census-management/group-censuses/groupCensusId/sync-summary
```



```
{
  "contractGroupPlanId": "OrgSG00000000e8jYAA"
},
{
  "contractGroupPlanId": "OrgSG00000000eALYAY"
}
]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
groupCensusMemberPlans	Group Census Member Plan Input[]	List of plans to add to the member.	Required	65.0

Response body for POST[Insurance Add Member Plans](#)**Request body for PATCH****JSON example**

```
{
  "addedPlans": {
    "values": [
      {
        "contractGroupPlanId": "OrgSG00000000e8jYAA"
      },
      {
        "contractGroupPlanId": "OrgSG00000000eALYAY"
      }
    ]
  },
  "deletedPlans": {
    "values": [
      "OrgSG00000000eALYAY",
      "OrgSG00000000e8jYAA"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
addedPlans	Group Census Member Plan Input[]	List of plans to add to the member.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
deletedPlans	String[]	List of group census member plan IDs to delete.	Optional	65.0

Response body for PATCH

Insurance Update Member Plans

Insurance Member Plans Delete (POST)

Delete specified plans for a group census member.

Resource

```
/connect/insurance/census-management/group-census-members/memberId/plans/action/delete
```

Resource example

<https://yourInstance.salesforce.com/services/data/v/connect/insurance/census-management/group-census-members/06R000000ch1pVA/plans/action/delete>

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
memberId	String	ID of the group census member for whom insurance plans must be deleted.	Required	65.0

Request body for POST

JSON example

```
{
  "plans": {
    "values": [
      "0rgSG0000000eALYAY",
      "0rgSG0000000e8jYAA"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
plans	String[]	List of group census member plan IDs to delete. If you specify the root plan ID, all the associated details will be deleted.	Required	65.0

Response body for POST

[Delete Member Plans](#)

Insurance Member Eligible Plans (GET)

Get the contract group plans for which a group census member and their dependents are eligible.

Resource

```
/connect/insurance/census-management/group-census-members/memberId/eligible-plans
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/census-management/group-census-members/06900000010124/eligible-plans
```

Available version

65.0

HTTP methods

GET

Path Parameter for GET

Name	Type	Description	Required or Optional	Available Version
memberId	String	ID of the group census member.	Required	65.0

Request parameter for GET

Parameter Name	Type	Description	Required or Optional	Available Version
shouldInclude Dependents	Boolean	Indicates whether to include the eligible plans for the member's dependents in the response (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .	Optional	65.0

Request body for PATCH**JSON example**

```
{
  "contextId": "42ce00ea6a44dcc0d308ba1ea7c4c9bb455bd14bb0405ff03dca1b5371c42cda",
  "addedPlans": [
    {
      "planId": "0gpp00000004C9wAAE",
      "coverages": [
        {
          "planId": "0rgxx0000000001AAA"
        }
      ]
    }
  ],
  "deletedPlans": []
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
addedPlans	Insurance Group Plan Input []	List of contract group plans to add to the member's enrollment for eligibility checking.	Optional	66.0
contextId	String	ID of the context.	Required	66.0
deletedPlans	Insurance Group Plan Input []	List of contract group plans to remove from the member's enrollment.	Optional	66.0

Response body for PATCH[Insurance Plans Eligibility Details](#)

Insurance Contract from Quote (POST)

Create a group benefits contract from a quote. This API supports asynchronous operation for long-running conversions.

Resource

```
/connect/insurance/contracts
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/contracts
```

Available version

65.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "quoteId": "0Q0SG000000DQA00AO",
  "contractStartDate": "2025-03-11",
  "contractEndDate": "2026-03-11",
  "enrollmentStartDate": "2025-01-01",
  "enrollmentEndDate": "2025-02-01",
  "contractTerm": 1,
  "additionalFields": [
    {
      "entity": "Contract",
      "fields": {
        "data": [
          {
            "key": "CustomerSignedTitle",
            "value": "Mr"
          }
        ]
      }
    },
    {
      "entity": "ContractGroupPlan",
      "instanceKey": "d5861be1-9920-42d9-8f43-093b0a709d35",
      "fields": {
        "data": [
          {
            "key": "EligibilityCriteria",
            "value": "testcriteria"
          }
        ]
      }
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Contract Item []	List of additional fields to add to contract items.	Optional	65.0
contractEndDate	String	End date of the contract.	Required	65.0
contractStartDate	String	Start date of the contract.	Required	65.0
contractTerm	Integer	Contract terms in months.	Optional	65.0
enrollmentEndDate	String	End date for plan enrollment.	Optional	65.0
enrollmentStartDate	String	Start date for plan enrollment.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
quoteId	String	ID of the quote to create the contract from.	Required	65.0

Response body for POST[Contract Creation Response](#)

Insurance Group Rating (POST)

Rate a group benefits quote to calculate premiums for all quote line items. Rate specific root plans to calculate premiums for their associated quote line items.

Resource

```
/connect/insurance/quotes/quoteId/group-rating
```

Resource Example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/quotes/0008d00000004sQzCAI/group-rating
```

Available version

65.0

HTTP methods

POST

Path Parameter for POST

Name	Type	Description	Required or Optional	Available Version
quoteId	String	ID of the group benefits quote.	Required	65.0

Request body for POST**JSON example**

```
{
  "batchSize": 200,
  "quoteLineItemIds": [
    "0qvp00000004C9wAAE"
  ],
  "ratingDate": "2025-01-01"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of records to process in each batch for the rating calculation. The default value is 5.	Optional	65.0
quoteLineItemIds	String[]	List of quote line item IDs of the root plan.	Optional	65.0
ratingDate	String	Date to use for the rating calculation. If not specified, the API uses the effective date of the quote.	Optional	65.0

Response body for POST[Group Rating Response](#)**Individual Enrollment (POST)**

Enroll a group census member in a group benefit insurance plan, creating the necessary policy records.

Resource

```
/connect/insurance/group-census-members/memberId/policies
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/group-census-members/067M0000000h1pYAA/policies
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
memberId	String	ID of the group census member.	Required	65.0

Request body for POST**JSON example**

```
{  
  "flowName": "MyEnrollmentFlow",  
  "plan": "0gpp00000004C9wAAE"  
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
flowName	String	API name of an optional flow to run as part of the enrollment process. The default value is Process Group Census Member Data.	Optional	65.0
plan	String	ID of the root contract group plan.	Required	65.0

Response body for POST[Individual Enrollment Response](#)

Requests

Learn more about the available requests of insurance group benefits APIs.

[Insurance Person Accounts and Users Creation Input](#)

Input representation for creating person accounts and portal users from group census members.

[Portal User Creation Input](#)

Input representation for the details to create a portal user.

[Insurance Member Rating Input](#)

Input representation for rating the individual members within a group.

[Insurance Member Rating Details Input](#)

Input representation for an individual member to be rated.

[Insurance Member Rating Plans Input](#)

Input representation for an individual rating plan.

[Insurance Contacts and Users Creation Input](#)

Input representation for creating contacts and portal users from group census members.

[Bulk Group Census Management Input](#)

Input representation to bulk-process group census data from a CSV file.

[Insurance Group Census Member Management Input](#)

Input representation for the details to add or update group census members.

[Group Census Member Input](#)

Input representation for the details of the group census member.

[Additional Fields Map Value](#)

Input representation for additional fields map value.

[Insurance Update Member Plans Input](#)

Input representation for updating the plans that are associated with a census member.

[Insurance Add Member Plans Input](#)

Input representation for adding plans to a census member.

[Group Census Member Plan Input](#)

Input representation for a single plan that's associated with a group census member.

[Delete Member Plans Input](#)

Input representation for deleting the insurance plans for a census member.

[Insurance Plans Eligibility Create Context Input](#)

Input representation for checking plan eligibility by creating a new context.

[Insurance Plans Eligibility Update Context Input](#)

Input representation for checking plan eligibility by updating an existing context.

[Insurance Group Plan Input](#)

Input representation for group census member plans and their coverages.

[Contract Creation Input](#)

Input representation for creating group benefits contract.

[Contract Item](#)

Input representation for additional fields to be added to contract items.

[Apex String](#)

Input representation for a key-value pair.

[Group Rating Input](#)

Input representation for the Group Benefits Group Rating API.

[Individual Enrollment Input](#)

Input representation for individual enrollment.

[Insurance Delete Census Member Input](#)

Input representation to delete census member.

Insurance Person Accounts and Users Creation Input

Input representation for creating person accounts and portal users from group census members.

JSON example

This is a sample JSON example with GroupCensusId.

```
{
  "groupCensusId": "0rffxx0000000007AAA",
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
}
```

```

"userDetails": {
  "profileId": "00exx000000mzxC",
  "localeSidKey": "en_US",
  "timeZoneSidKey": "America/New York",
  "languageLocaleKey": "en_US",
  "emailEncodingKey": "UTF-8"
}
}

```

This is a sample JSON example with GroupCensusMembers.

```

{
  "groupCensusMemberIdList": {
    "values": [
      "0r6xx0000000002AAA",
      "0r6xx0000000004rAAA"
    ]
  },
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzxC",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldMappings	Map<String, String>	Map of field API names from the Group Census Member object (key) to the Account object (value).	Optional	65.0
groupCensusId	String	ID of the group census that contains the members to process.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
groupCensusMemberIdList	String[]	IDs of the group census members for whom you want to create person accounts and users. If not provided, all members of the specified group census are processed.	Optional	65.0
matchingKeyList	String[]	List of field API names on the Group Census Member object that are used to identify and prevent the creation of duplicate person accounts. The default fields are FirstName, LastName, PersonBirthdate, PersonEmail, and PersonGender.	Optional	65.0
personAccountRecordType	String	Developer name of the person account record type to use for the new accounts.	Optional	65.0
userDetails	Portal User Creation Input	Details to create portal users.	Required if the <code>createPortalUsers</code> request parameter is set to <code>true</code> .	65.0

Portal User Creation Input

Input representation for the details to create a portal user.

JSON example

```
{
  "profileId": "00e123AbcDefGhiJkl",
  "emailEncodingKey": "ISO-8859-1",
  "languageLocaleKey": "en_US",
  "localeSidKey": "en_US",
  "timeZoneSidKey": "Atlanta"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
emailEncodingKey	String	Email encoding for the user, such as UTF-8 or ISO-8859-1.	Optional	65.0
languageLocaleKey	String	Language for the user, such as en_US for U.S. English.	Optional	65.0
localeSidKey	String	Locale for the user, which affects the format of dates, times, and numbers.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
profileId	String	ID of the profile to assign to the new portal user.	Required	65.0
timeZoneSidKey	String	Time zone for the user, such as America or Los_Angeles.	Optional	65.0

Insurance Member Rating Input

Input representation for rating the individual members within a group.

JSON example

```
{
  "effectiveDate": "2025-10-01",
  "members": [
    {
      "memberId": "0r6RM0000000h1pYAA",
      "plans": [
        {
          "planId": "0j5RM0000000a1YAA",
          "planCoverageIds": [
            "0k8RM00000001b2ZAA"
          ]
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the member's rating is calculated.	Optional	65.0
members	Insurance Member Rating Details Input[]	Details of the members and their plans to be rated.	Required	65.0

Insurance Member Rating Details Input

Input representation for an individual member to be rated.

Properties

Name	Type	Description	Required or Optional	Available Version
memberId	String	ID of the member.	Required	65.0
plans	Insurance Member Rating Plans Input[]	Plan details for individual rating.	Required	65.0

Insurance Member Rating Plans Input

Input representation for an individual rating plan.

Properties

Name	Type	Description	Required or Optional	Available Version
planCoverageIds	String[]	Coverage IDs within the plan.	Optional	65.0
planId	String	ID of the plan.	Required	65.0

Insurance Contacts and Users Creation Input

Input representation for creating contacts and portal users from group census members.

JSON example

This is a sample JSON example with GroupCensusId.

```
{
  "groupCensusId": "0rfxx0000000007AAA",
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzxC",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

This is a sample JSON example with GroupCensusMembers.

```
{
  "groupCensusMemberIdList": {
    "values": [
      "0r6xx0000000002AAA",
      "0r6xx0000000004rAAA"
    ]
  },
  "matchingKeyList": {
    "values": [
      "FirstName",
      "LastName"
    ]
  },
  "fieldMappings": {
    "data": {
      "FirstName": "FirstName",
      "Lastname": "LastName"
    }
  },
  "userDetails": {
    "profileId": "00exx000000mzxC",
    "localeSidKey": "en_US",
    "timeZoneSidKey": "America/New York",
    "languageLocaleKey": "en_US",
    "emailEncodingKey": "UTF-8"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
fieldMappings	Map	Map of field API names from the Group Census Member object (key) to the Contact object (value).	Optional	65.0
groupCensusId	String	ID of the group census.	Optional	65.0
groupCensusMemberIdList	String[]	IDs of the group census members for whom you want to create contacts and users. If not provided, all members of the specified group census are processed.	Optional	65.0
matchingKeyList	String[]	List of field API names on the Group Census Member object that are used to identify and prevent the creation of duplicate contact records. The default fields are FirstName, LastName, Birthdate, Email, and Gender.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
userDetails	Portal User Creation Input	Details to create portal users. This property is required if the <code>createPortalUsers</code> parameter in the method call is set to <code>true</code> .	Optional	65.0

Bulk Group Census Management Input

Input representation to bulk-process group census data from a CSV file.

JSON example

```
{
  "batchSize": 200,
  "contentDocumentId": "069SG000003gLazYAE",
  "contentDocumentVersionId": "0688d0000025fA3AAI",
  "duplicateDetectionFields": {
    "values": [
      "FirstName",
      "Lastname"
    ]
  },
  "csvColumnMapping": {
    "data": {
      "First Name": "FirstName",
      "Last Name": "Lastname",
      "Email": "Email",
      "Birthdate": "Birthdate",
      "Gender": "Gender",
      "Relationship to Primary Member": "AssociationWithPrimaryMbr",
      "Medical": "ContractGroupPlan",
      "Dental": "ContractGroupPlan",
      "Group Class": "GroupClassName"
    }
  },
  "shouldExecuteConfigurationRule": false
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of CSV records to process in each batch. The default size for Quote Census is 2000. The default size for Enrollment Census is 25.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
content DocumentId	String	ID of the content document (CSV file) to process.	Required	65.0
contentDocument VersionId	String	ID of the specific version of the content document.	Optional	65.0
csvColumnMapping	Map<String, String>	Map where the key is the CSV column header and the value is the API name of the corresponding field in the GroupCensusMember object.	Required	65.0
duplicate Detection Fields	String[]	List of field API names that are used to generate a unique key for identifying duplicate members along with the default values. The default fields that are used to detect duplication are <code>FirstName</code>, <code>LastName</code>, <code>groupCensusId</code>, and <code>AssociationWithPrimaryMbr</code>.	Optional	65.0
shouldExecute Configuration Rule	Boolean	Indicates whether to execute the configuration rules after processing the data (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code>.	Optional	65.0

Insurance Group Census Member Management Input

Input representation for the details to add or update group census members.

JSON example

```
{
  "members": {
    "values": [
      {
        "name": "member1",
        "firstName": "fm",
        "lastName": "lm",
        "email": "cci@test.com",
        "sourceSystemIdentifier": "1.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
      },
      {
        "name": "member2",
        "firstName": "fm2",
        "lastName": "lm2",
        "email": "cci@test.com",

```

```

        "sourceSystemIdentifier": "2.primary",
        "primaryMemberSource": null,
        "relationshipToPrimaryMember": "Self",
        "associationWithPrimaryMbr": "Self"
    },
    {
        "name": "dep1",
        "firstName": "depfm",
        "lastname": "depfm",
        "email": "cci@test.com",
        "associationWithPrimaryMbr": "Child",
        "relationshipToPrimaryMember": "Child",
        "sourceSystemIdentifier": "1.Child",
        "primaryMemberSource": "1.primary"
    },
    {
        "name": "dep2",
        "firstName": "depfm2",
        "lastname": "depfm2",
        "email": "cci@test.com",
        "associationWithPrimaryMbr": "Child",
        "relationshipToPrimaryMember": "Child",
        "sourceSystemIdentifier": "2.Child",
        "primaryMemberSource": "2.primary"
    }
]
},
"duplicateDetectionFields": {
    "values": [
        "firstName"
    ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
duplicate Detection Fields	String[]	List of field API names that are used to generate a unique key for identifying duplicate members along with default values. The default fields for duplicate detection are <code>FirstName</code>, <code>LastName</code>, <code>AssociationWithPrimaryMbr</code>, and <code>GroupCensusId</code>.	Optional	65.0
flowName	String	Name of the flow to use for processing group member data. Default value is <code>Process Group Census Member Data</code>.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
members	Group Census Member Input[]	List of group census members to add or update.	Required	65.0

Group Census Member Input

Input representation for the details of the group census member.

JSON example

```
{
  "members": [
    {
      "accountId": "0015j00000AbCdEfGHI",
      "additionalFields": [
        {
          "key": "EmployeeNumber__c",
          "value": "E12345"
        }
      ],
      "annualEligibleSalary": 82000.5,
      "associationWithPrimaryMbr": null,
      "birthdate": "1990-08-25",
      "city": "Austin",
      "contactId": "0035j00000KlMnOpQR",
      "country": "USA",
      "dependentCount": 0,
      "email": "alex.taylor@example.com",
      "firstName": "Alex",
      "fullTimeEquivalent": 1,
      "gender": "Non-Binary",
      "groupCensusId": "0r65j00000StUvWxYZ",
      "groupClassId": "a015j00000XyZ12aBC",
      "groupClassName": "Full-Time Salaried",
      "grpCensusGrpClassSummaryId": "a025j00000GhIjKkLM",
      "id": "0r65j00000StUvWxYZ",
      "isOptOutAllPlans": false,
      "isPortalUser": false,
      "joinDate": "2023-03-15",
      "lastname": "Taylor",
      "memberKey": "TAYLOR-ALEX-19900825",
      "memberType": "Primary",
      "name": "Alex Taylor",
      "optOutPlanTypes": null,
      "phoneNumber": "512-555-0199",
      "policyStartDate": "2025-01-01",
      "postalCode": "78701",
      "primaryGroupCensusMemberId": null,
      "primaryMemberCategory": "Full-Time",
      "primaryMemberSource": "Workday",
      "relationshipToPrimaryMember": "Self",
      "smokerStatus": "Non-Smoker",
    }
  ]
}
```



```

    "sourceSystemIdentifier": "WD-EMP-67890",
    "state": "TX",
    "status": "Active",
    "street": "456 Congress Ave"
  }
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	Person account ID.	Optional	65.0
additionalFields	Additional Fields Map Value[]	List of key-value pairs for populating custom fields or other fields.	Optional	65.0
annualEligibleSalary	Double	Annual salary that's eligible for benefits calculations.	Optional	65.0
associationWithPrimaryMbr	String	Association to primary member for dependents, such as Spouse, Child.	Optional	65.0
birthdate	String	Date of birth of the member, in YYYY-MM-DD format.	Optional	65.0
city	String	City of the member's address.	Optional	65.0
contactId	String	ID of the Contact record of the member.	Optional	65.0
country	String	Country of the member's address.	Optional	65.0
dependentCount	Integer	Number of dependents that are associated with the member.	Optional	65.0
email	String	Email address of the member.	Optional	65.0
firstName	String	First name of the member.	Required to add member	65.0
fullTimeEquivalent	Double	Work capacity of the member, where 1.0 indicates a full-time employee.	Optional	65.0
gender	String	Gender of the member.	Optional	65.0
groupCensusId	String	ID of the Group Census that the member belongs to.	Optional	65.0
groupClassId	String	ID for the specific class or category within the group that the member belongs to.	Optional	65.0
groupClassName	String	Name of the group class.	Optional	65.0
grpCensusGrpClassSummaryId	String	ID of the related group census group class summary record.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
id	String	ID of the group census member to update the member details.	Required to update member	65.0
isOptOutAllPlans	Boolean	Indicates whether the member has opted out of all available insurance plans (<code>true</code>) or not (<code>false</code>).	Optional	65.0
isPortalUser	Boolean	Indicates whether a community portal user must be created for this member (<code>true</code>) or not (<code>false</code>).	Optional	65.0
joinDate	String	Date the member joined the group or company.	Optional	65.0
lastname	String	Last name of the member.	Required to add member	65.0
memberKey	String	Unique, externally defined key that's used for identifying and preventing duplicate members, often created by concatenating several fields.	Optional	65.0
memberType	String	Type of member, such as Primary or Dependent.	Optional	65.0
name	String	Full name of the member.	Optional	65.0
optOutPlanTypes	String	Types of plans the member has opted out of.	Optional	65.0
phoneNumber	String	Phone number of the member.	Optional	65.0
policyStartDate	String	Date when the member's policy coverage begins.	Optional	65.0
postalCode	String	Postal or ZIP code of the member's address.	Optional	65.0
primaryGroupCensusMemberId	String	ID of the primary group census member the dependents are associated with.	Optional	65.0
primaryMemberCategory	String	Category of the primary member.	Optional	65.0
primaryMemberSource	String	System or source from which the member's data originated.	Optional	65.0
relationshipToPrimaryMember	String	Member's relationship to the primary insured.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
smokerStatus	String	Smoking status of the member, such as Smoker or Non-smoker.	Optional	65.0
sourceSystem Identifier	String	ID from an external source system, that's used for integration purposes.	Optional	65.0
state	String	State or province the member belongs to.	Optional	65.0
status	String	Current status of the member.	Optional	65.0
street	String	Street address of the member.	Optional	65.0

Additional Fields Map Value

Input representation for additional fields map value.

JSON example

```
{
  "key": "EmployeeNumber__c",
  "value": "E12345"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additional FieldsMapValue	Object	Key-value pairs for populating custom fields or other fields.	Optional	65.0

Insurance Update Member Plans Input

Input representation for updating the plans that are associated with a census member.

JSON example

```
{
  "addedPlans": {
    "values": [
      {
        "contractGroupPlanId": "0rgSG0000000e8jYAA"
      },
      {
        "contractGroupPlanId": "0rgSG0000000eALYAY"
      }
    ]
  },
  "deletedPlans": {
    "values": [
      "0rgSG0000000eALYAY",
    ]
  }
}
```

```
{
  "OrgSG00000000e8jYAA"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
addedPlans	Group Census Member Plan Input[]	List of plans to add to the member.	Optional	65.0
deletedPlans	String[]	List of group census member plan IDs to delete.	Optional	65.0

Insurance Add Member Plans Input

Input representation for adding plans to a census member.

JSON example

```
{
  "groupCensusMemberPlans": {
    "values": [
      {
        "contractGroupPlanId": "OrgSG00000000e8jYAA"
      },
      {
        "contractGroupPlanId": "OrgSG00000000eALYAY"
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
groupCensusMemberPlans	Group Census Member Plan Input[]	List of plans to add to the member.	Required	65.0

Group Census Member Plan Input

Input representation for a single plan that's associated with a group census member.

JSON example

```
{
  "contractGroupPlanId": "0s1xx000000001aEAA",
}
```

```
{
  "name": "Gold Tier Medical Plan"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contractGroupPlanId	String	ID of the contract group plan.	Required	65.0
groupCensusMemberId	String	ID of the group census member.	Optional	65.0
id	String	ID of the group census member plan record.	Required to update plan	65.0
name	String	Name of the group census member plan.	Optional	65.0

Delete Member Plans Input

Input representation for deleting the insurance plans for a census member.

JSON example

```
{
  "plans": {
    "values": [
      "0rgSG00000000eALYAY",
      "0rgSG00000000e8jYAA"
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
plans	String[]	List of group census member plan IDs to delete. If you specify the root plan ID, all the associated details will be deleted.	Required	65.0

Insurance Plans Eligibility Create Context Input

Input representation for checking plan eligibility by creating a new context.

JSON example

```
{
  "plans": [
    {
      "planId": "0gpp00000004C9wAAE",
      "coverages": []
    }
  ]
}
```

```

    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
plans	Insurance Group Plan Input[]	List of contract group plans for initial eligibility validation.	Required	66.0

Insurance Plans Eligibility Update Context Input

Input representation for checking plan eligibility by updating an existing context.

JSON example

```

{
  "contextId": "42ce00ea6a44dcc0d308ba1ea7c4c9bb455bd14bb0405ff03dca1b5371c42cda",
  "addedPlans": [
    {
      "planId": "0gpp00000004C9wAAE",
      "coverages": [
        {
          "planId": "0rgxx0000000001AAA"
        }
      ]
    }
  ],
  "deletedPlans": []
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
addedPlans	Insurance Group Plan Input[]	List of contract group plans to add to the member's enrollment for eligibility checking.	Optional	66.0
contextId	String	ID of the context.	Required	66.0
deletedPlans	Insurance Group Plan Input[]	List of contract group plans to remove from the member's enrollment.	Optional	66.0

Insurance Group Plan Input

Input representation for group census member plans and their coverages.

JSON example

```
{
  "planId": "0gpp00000004C9wAAE",
  "coverages": [
    {
      "planId": "0gqp00000004C9wAAE"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
coverages	Insurance Group Plan Input[]	List of coverages that are associated with this plan.	Optional	65.0
planId	String	ID of the contract group plan or coverage.	Required	65.0

Contract Creation Input

Input representation for creating group benefits contract.

JSON example

```
{
  "quoteId": "0Q0SG000000DQA00AO",
  "contractStartDate": "2025-03-11",
  "contractEndDate": "2026-03-11",
  "enrollmentStartDate": "2025-01-01",
  "enrollmentEndDate": "2025-02-01",
  "contractTerm": 1,
  "additionalFields": [
    {
      "entity": "Contract",
      "fields": {
        "data": [
          {
            "key": "CustomerSignedTitle",
            "value": "Mr"
          }
        ]
      }
    }
  ],
  {
    "entity": "ContractGroupPlan",
    "instanceKey": "d5861be1-9920-42d9-8f43-093b0a709d35",
    "fields": {
      "data": [
        {
          "key": "EligibilityCriteria",
          "value": "testcriteria"
        }
      ]
    }
  }
}
```

```

    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Contract Item []	List of additional fields to add to contract items.	Optional	65.0
contractEndDate	String	End date of the contract.	Required	65.0
contractStartDate	String	Start date of the contract.	Required	65.0
contractTerm	Integer	Contract terms in months.	Optional	65.0
enrollmentEndDate	String	End date for plan enrollment.	Optional	65.0
enrollmentStartDate	String	Start date for plan enrollment.	Optional	65.0
quoteId	String	ID of the quote to create the contract from.	Required	65.0

Contract Item

Input representation for additional fields to be added to contract items.

JSON example

```

{
  "entity": "ContractGroupPlan",
  "fields": [
    {
      "key": "CustomField__c",
      "value": "CustomValue"
    }
  ],
  "instanceKey": "d5861be1-9920-42d9-8f43-093b0a709d35"
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
entity	String	API name of the entity to add fields to, such as ContractGroupPlan.	Optional	65.0
fields	Apex String []	List of key-value pairs that represent the fields and their values for the entity.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
instanceKey	String	Unique instance key of the Quote Line Item.	Optional	65.0

Apex String

Input representation for a key-value pair.

JSON example

```
{
  "key": "FieldName__c",
  "value": "FieldValue"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
key	String	API name of the field.	Optional	65.0
value	String	Value to assign to the field.	Optional	65.0

Group Rating Input

Input representation for the Group Benefits Group Rating API.

JSON example

```
{
  "batchSize": 200,
  "quoteLineItemIds": [
    "0qvp00000004C9wAAE"
  ],
  "ratingDate": "2025-01-01"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of records to process in each batch for the rating calculation. The default value is 5.	Optional	65.0
quoteLineItemIds	String[]	List of quote line item IDs of the root plan.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
ratingDate	String	Date to use for the rating calculation. If not specified, the API uses the effective date of the quote.	Optional	65.0

Individual Enrollment Input

Input representation for individual enrollment.

JSON example

```
{
  "flowName": "MyEnrollmentFlow",
  "plan": "0gpp00000004C9wAAE"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
flowName	String	API name of an optional flow to run as part of the enrollment process. The default value is Process Group Census Member Data.	Optional	65.0
plan	String	ID of the root contract group plan.	Required	65.0

Insurance Delete Census Member Input

Input representation to delete census member.

JSON example

```
{
  "memberIds": [
    "0r6RM0000000h1pYAA"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
memberIds	String[]	IDs of the group census member that must be deleted.	Required	65.0

Responses

Learn more about the available response bodies of insurance group benefits APIs.

[Insurance Person Accounts and Users Creation Response](#)

Output representation of the person account and user creation process.

[Insurance Member Account and User Details](#)

Output representation of a member's person account and portal user details that are created.

[Insurance Error](#)

Output representation for an error that occurred during a census management operation.

[Insurance Member Rating](#)

Output representation of the rating for individual members.

[Insurance Member Rating Details](#)

Output representation of an individual member's rating results.

[Insurance Plan Rating](#)

Output representation of an individual plan's pricing result.

[Insurance Coverage Rating](#)

Output representation of an individual coverage's pricing result.

[Insurance Contribution](#)

Output representation of contribution information in an individual rating.

[Insurance Object Map](#)

Output representation for a map of attributes with string keys and object values.

[Insurance Contacts and Users Creation Response](#)

Output representation of the contact and user creation process.

[Insurance Member Contact and User Details](#)

Output representation of a member's contact and portal user details that's created.

[Bulk Group Census Management](#)

Output representation for the bulk processing of employee census data.

[Insurance Group Census Member Management](#)

Output representation for the details of the group census members.

[Insurance Census Summary Sync Details](#)

Output representation of an employee census summary synchronization.

[Insurance Group Census Details](#)

Output representation of group census summary data.

[Group Census Group Class Summary](#)

Output representation of census summary data aggregated by group class.

[Insurance Update Member Plans](#)

Output representation for the update member plans operation.

[Insurance Add Member Plans](#)

Output representation for the add member plans operation.

[Delete Member Plans](#)

Output representation for the member plan deletion process.

[Insurance Member Eligible Plans Response](#)

Output representation for a member's eligible plans.

[Members Plans](#)

Output representation for a list of members with their eligible plans.

[Insurance Member Plans](#)

Output representation of the Insurance Get Member Plans service for an individual member.

[Insurance Plan List](#)

Output representation for a list of contract group plans.

[Insurance Plan Details](#)

Output representation for contract group plans.

[Insurance Plans Eligibility Details](#)

Output representation of the plan eligibility check for an.

[Eligible Plan Details](#)

Output representation for the details of the eligible plans.

[Insurance Message Output](#)

Output representation of an insurance message, which can be informational or an error.

[Insurance Group Plans](#)

Output representation of an insurance group plan.

[Contract Creation Response](#)

Output representation of the group benefits Quote-to-Contract process.

[Group Rating Response](#)

Output representation of the Group Benefits Group Rating process.

[Individual Enrollment Response](#)

Output representation for the Individual Enrollment API.

[Family Policy Details](#)

Output representation for individual group policy information.

Insurance Person Accounts and Users Creation Response

Output representation of the person account and user creation process.

JSON example

This is a sample JSON example for a successful API request.

```
{
  "accountsAndUsersDetails": [
    {
      "groupCensusId": "0rffx00000001Qv",
      "groupCensusMemberId": "0r6xx00000002uR",
      "personAccountId": "001xx000003GeCV"
    },
    {
      "groupCensusId": "0rffx00000001Qv",
```

```

    "groupCensusMemberId": "0r6xx00000002uU",
    "personAccountId": "001xx000003GeCW"
  }
],
"errors": [],
"isSuccess": true
}

```

This is a sample JSON example with error.

```

{
  "accountsAndUsersDetails": [
    {
      "groupCensusId": "0rfxx00000001Qr",
      "groupCensusMemberId": "0r6xx00000002sB",
      "personAccountId": "001xx000003GeE7"
    },
    {
      "groupCensusId": "0rfxx00000001Qr",
      "groupCensusMemberId": "0r6xx00000002sC",
      "personAccountId": "001xx000003GeE8"
    }
  ],
  "errors": [
    {
      "additionalInfo": {},
      "code": "SF-0004-0047",
      "message": "Ensure to set the isPortalUser flag to true for group census members 0r6xx00000002sB,0r6xx00000002sC."
    }
  ],
  "isSuccess": true
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
accountsAndUsersDetails	Insurance Member Account and User Details[]	Details of the objects that contain the IDs of the person accounts and users that are created and mapped to their source group census members.	Small, 65.0	65.0
errors	Insurance Error[]	Details of errors that occurred during the operation. Each error includes details about why a specific group census member failed to process.	Small, 65.0	65.0
groupCensusId	String	ID of the group census that's processed.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the overall operation was successful (<code>true</code>) or not (<code>false</code>). If any member fails to process, this value is <code>false</code> .	Small, 65.0	65.0

Insurance Member Account and User Details

Output representation of a member's person account and portal user details that are created.

JSON example

```
{
  "groupCensusId": "0rfAbcDefGhiJklMno",
  "groupCensusMemberId": "0r6PrimaryMemberId",
  "personAccountId": "003abc123DEF456ghi",
  "userId": "005jkl789MNO012pqr"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
groupCensusId	String	ID of the group census for the specific member.	Small, 65.0	65.0
groupCensusMemberId	String	ID of the source group census member who's processed.	Small, 65.0	65.0
personAccountId	String	ID of the person account that's created.	Small, 65.0	65.0
userId	String	ID of the portal user that's created. This property is null if portal user creation isn't requested.	Small, 65.0	65.0

Insurance Error

Output representation for an error that occurred during a census management operation.

JSON example

```
{
  "code": "RESOURCE_NOT_FOUND",
  "message": "The requested resource does not exist",
  "additionalInfo": {
    "memberId": "0r6RM000000h1pYAA"
  }
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
additionalInfo	Map	Map of additional details about the error, such as the ID of the record that caused the failure.	Small, 65.0	65.0
code	String	Error code.	Small, 65.0	65.0
message	String	Error message.	Small, 65.0	65.0

Insurance Member Rating

Output representation of the rating for individual members.

JSON example

```
{
  "errors": [],
  "id": "c639a6ec-11d1-4c66-b997-3fff967f052d",
  "pricingResults": [
    {
      "errors": [],
      "memberId": "Or6DR0000002UmvYAE",
      "plans": [
        {
          "contributions": {
            "employeePremiumAmount": 51.04,
            "employerPremiumAmount": 5.67,
            "errors": []
          },
          "coverages": [
            {
              "contributions": {
                "employeePremiumAmount": 51.04,
                "employerPremiumAmount": 5.67,
                "errors": [],
                "groupClassContributionId": "OrFDR00000002DM2AY"
              },
              "coverageId": "OrgDR00000001iVPYAY",
              "errors": [],
              "proratedRates": {
                "data": {
                  "NetUnitPrice": 56.71,
                  "NetTotalPrice": 56.71,
                  "StandardQLITaxAmount": 0
                }
              },
              "rates": {
                "data": {
                  "NetUnitPrice": 112.5,
                  "NetTotalPrice": 112.5,
                  "StandardQLITaxAmount": 0
                }
              }
            }
          ],
          "errors": [],
          "id": "OrgDR00000001iVLYAY",
          "proratedRates": {
            "data": {
              "NetUnitPrice": 56.71,
              "NetTotalPrice": 56.71,
              "StandardQLITaxAmount": 0
            }
          }
        }
      ],
      "errors": []
    }
  ],
  "errors": []
}
```

```

      "rates": {
        "data": {
          "NetUnitPrice": 112.5,
          "NetTotalPrice": 112.5,
          "StandardQLITaxAmount": 0
        }
      }
    }
  ]
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	Error details.	Small, 65.0	65.0
id	String	Unique ID that's generated during the member rating transaction.	Small, 65.0	65.0
pricingResults	Insurance Member Rating Details[]	Details of pricing results for a member's rating calculation.	Small, 65.0	65.0

Insurance Member Rating Details

Output representation of an individual member's rating results.

JSON example

```

{
  "memberId": "0r6PrimaryMemberId",
  "plans": [
    {
      "id": "0kfMedicalPlanId01",
      "rates": {
        "data": {
          "total": 350
        }
      },
      "proratedRates": {
        "data": {
          "total": 350
        }
      },
      "contributions": {
        "employeePremiumAmount": 100,
        "employerPremiumAmount": 250,
        "groupClassContributionId": "gcc-001-a",
        "errors": []
      },
      "coverages": [
        {
          "coverageId": "cov-hosp-01",

```



```
      "rates": {
        "data": {
          "total": 350
        }
      },
      "proratedRates": {
        "data": {
          "total": 350
        }
      },
      "contributions": {
        "employeePremiumAmount": 100,
        "employerPremiumAmount": 250,
        "groupClassContributionId": "gcc-001-a",
        "errors": []
      }
    ],
    "errors": []
  }
],
"errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	Error details.	Small, 65.0	65.0
memberId	String	ID of the member whose rating is calculated.	Small, 65.0	65.0
plans	Insurance Plan Rating[]	Rating details for the plan.	Small, 65.0	65.0

Insurance Plan Rating

Output representation of an individual plan's pricing result.

JSON example

```
{
  "id": "OkfMedicalPlanId01",
  "rates": {
    "data": {
      "total": 350
    }
  },
  "proratedRates": {
    "data": {
      "total": 350
    }
  },
  "contributions": {
    "employeePremiumAmount": 100,
```

```

    "employerPremiumAmount": 250,
    "groupClassContributionId": "gcc-001-a",
    "errors": []
  },
  "coverages": [
    {
      "coverageId": "cov-hosp-01",
      "rates": {
        "data": {
          "total": 350
        }
      },
      "proratedRates": {
        "data": {
          "total": 350
        }
      },
      "contributions": {
        "employeePremiumAmount": 100,
        "employerPremiumAmount": 250,
        "groupClassContributionId": "gcc-001-a",
        "errors": []
      }
    }
  ],
  "errors": []
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
contributions	Insurance Contribution	Contribution details for the plan, if applicable.	Small, 65.0	65.0
coverages	Insurance Coverage Rating[]	Coverage details for the insurance plan.	Small, 65.0	65.0
errors	Insurance Error[]	Error details.	Small, 65.0	65.0
id	String	ID of the plan for which the rating is calculated.	Small, 65.0	65.0
rates	Insurance Object Map	Rate details for each plan.	Small, 65.0	65.0
proratedRates	Insurance Object Map	Prorated rates for the specified plan.	Small, 65.0	65.0

Insurance Coverage Rating

Output representation of an individual coverage's pricing result.

Property Name	Type	Description	Filter Group and Version	Available Version
contributions	Insurance Contribution	Contribution details for the coverage, if applicable.	Small, 65.0	65.0
coverageId	String	ID of the coverage.	Small, 65.0	65.0
errors	Insurance Error[]	Error details.	Small, 65.0	65.0
proratedRates	Insurance Object Map	Prorated rates for the coverage.	Small, 65.0	65.0
rates	Insurance Object Map	Rate details for the coverage.	Small, 65.0	65.0

Insurance Contribution

Output representation of contribution information in an individual rating.

Property Name	Type	Description	Filter Group and Version	Available Version
employee PremiumAmount	Double	Premium amount that's contributed by an employee.	Small, 65.0	65.0
employer PremiumAmount	Double	Premium amount that's contributed by an employer.	Small, 65.0	65.0
errors	Insurance Error[]	Error details.	Small, 65.0	65.0
groupClass ContributionId	String	ID of the contribution record.	Small, 65.0	65.0

Insurance Object Map

Output representation for a map of attributes with string keys and object values.

JSON example

```
{
  "data": {
    "attributeKey": "attributeValue"
  }
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
data	Map<String, Object>	Map of string keys and object values.	Small, 65.0	65.0

Insurance Contacts and Users Creation Response

Output representation of the contact and user creation process.

JSON example

This is a sample JSON example for a successful API request.

```
{
  "isSuccess": true,
  "groupCensusId": "0rfAbcDefGhiJklMno",
  "contactsAndUsersDetails": [
    {
      "groupCensusId": "0rfAbcDefGhiJklMno",
      "groupCensusMemberId": "0r6PrimaryMemberId",
      "contactId": "003abc123DEF456ghi"
    },
    {
      "groupCensusId": "0rfAbcDefGhiJklMno",
      "groupCensusMemberId": "0r6DependentMemberId",
      "contactId": "003stu456VWX789yz"
    }
  ],
  "errors": []
}
```

This is a sample JSON example with error.

```
{
  "contactsAndUsersDetails": [
    {
      "groupCensusId": "0rfxx00000001Qr",
      "groupCensusMemberId": "0r6xx00000002sB",
      "contactId": "001xx000003Gd15AAC"
    },
    {
      "groupCensusId": "0rfxx00000001Qr",
      "groupCensusMemberId": "0r6xx00000002sC",
      "contactId": "001xx000003Gd15BAC"
    }
  ],
  "errors": [
    {
      "additionalInfo": {},
      "code": "SF-0004-0047",
      "message": "Ensure to set the isPortalUser flag to true for group census members 0r6xx00000002sB,0r6xx00000002sC."
    }
  ],
  "isSuccess": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
contactsAndUsersDetails	Insurance Member Contact and User Details[]	Details of contacts and users that are created.	Small, 65.0	65.0
errors	Insurance Error[]	Errors that occurred during the process.	Small, 65.0	65.0
groupCensusId	String	ID of the group census that's processed.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0

Insurance Member Contact and User Details

Output representation of a member's contact and portal user details that's created.

JSON example

```
{
  "contactId": "003abc123DEF456ghi",
  "groupCensusId": "0rfAbcDefGhiJklMno",
  "groupCensusMemberId": "0r6PrimaryMemberId",
  "userId": "005jkl789MNO012pqr"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
contactId	String	ID of the contact record that's created.	Small, 65.0	65.0
groupCensusId	String	ID of the group census for the specific member.	Small, 65.0	65.0
groupCensusMemberId	String	ID of the group census member who's processed.	Small, 65.0	65.0
userId	String	ID of the portal user that's created.	Small, 65.0	65.0

Bulk Group Census Management

Output representation for the bulk processing of employee census data.

JSON example

```
{
  "errors": [],
  "isSuccess": true,
  "requestId": "a1Bxx000000001aEAA"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	ConnectApiResponse	List of errors that contains a message and an error code.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the request is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
requestId	String	ID of the asynchronous job request.	Small, 65.0	65.0

Insurance Group Census Member Management

Output representation for the details of the group census members.

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	List of errors that are associated with the API operation.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
memberIds	String[]	List of IDs for the group census members that are created, updated, or deleted.	Small, 65.0	65.0

Insurance Census Summary Sync Details

Output representation of an employee census summary synchronization.

JSON example

```
{
  "isSuccess": true,
  "errors": null,
  "groupCensus": {
    "id": "a0B1x000000XXXXX",
    "memberOptOutCount": 5,
    "mbrWithoutDependentCount": 20,
    "memberWithOneChildCount": 15,
    "mbrWithMoreThan1ChldCount": 10,
    "mbrWithMoreThan2ChldCount": 5,
    "memberWithDependentCount": 30,
    "memberWithSpouseCount": 25,
    "totalMemberPlusDependent": 85,
    "totalDependents": 45,
    "totalMembers": 40,
    "fullTimeMemberCount": 35,
    "partTimeMemberCount": 5,
    "grpCensusGrpClassSummaryList": [
      {
        "id": "a0C1x000000YYYYY",
```

```
    "groupCensusId": "a0B1x000000XXXXX",
    "groupClassId": "a0D1x000000ZZZZZ",
    "memberOptOutCount": 2,
    "mbrWithoutDependentCount": 10,
    "memberWithOneChildCount": 5,
    "mbrWithMoreThan1ChldCount": 3,
    "mbrWithMoreThan2ChldCount": 2,
    "memberWithDependentCount": 10,
    "memberWithSpouseCount": 8,
    "totalMemberPlusDependent": 28,
    "totalDependents": 15,
    "totalMembers": 13,
    "fullTimeMemberCount": 12,
    "partTimeMemberCount": 1
  }
]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Link to Error Report	List of errors that occurred during processing.	Small, 65.0	65.0
groupCensus	Insurance Group Census Details	Summary details for the group census.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the operation is successful (true) or not (false).	Small, 65.0	65.0

Insurance Group Census Details

Output representation of group census summary data.

JSON example

```
{
  "groupCensus": {
    "id": "a0B1x000000XXXXX",
    "memberOptOutCount": 5,
    "mbrWithoutDependentCount": 20,
    "memberWithOneChildCount": 15,
    "mbrWithMoreThan1ChldCount": 10,
    "mbrWithMoreThan2ChldCount": 5,
    "memberWithDependentCount": 30,
    "memberWithSpouseCount": 25,
    "totalMemberPlusDependent": 85,
    "totalDependents": 45,
    "totalMembers": 40,
    "fullTimeMemberCount": 35,
    "partTimeMemberCount": 5,
    "grpCensusGrpClassSummaryList": [
      {
```

```

    "id": "a0C1x000000YYYYY",
    "groupCensusId": "a0B1x000000XXXXX",
    "groupClassId": "a0D1x000000ZZZZZ",
    "memberOptOutCount": 2,
    "mbrWithoutDependentCount": 10,
    "memberWithOneChildCount": 5,
    "mbrWithMoreThan1ChldCount": 3,
    "mbrWithMoreThan2ChldCount": 2,
    "memberWithDependentCount": 10,
    "memberWithSpouseCount": 8,
    "totalMemberPlusDependent": 28,
    "totalDependents": 15,
    "totalMembers": 13,
    "fullTimeMemberCount": 12,
    "partTimeMemberCount": 1
  }
]
}
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
fullTimeMemberCount	Integer	Total number of full-time members.	Small, 65.0	65.0
grpCensus GrpClass SummaryList	Group Census Group Class Summary[]	List of census summaries that are aggregated by group class.	Small, 65.0	65.0
id	String	ID of the group census summary.	Small, 65.0	65.0
mbrWith MoreThan1 ChldCount	Integer	Total number of members who have more than one child dependent.	Small, 65.0	65.0
mbrWith MoreThan2 ChldCount	Integer	Total number of members who have more than two child dependents.	Small, 65.0	65.0
mbrWithout Dependent Count	Integer	Total number of members who have no dependents.	Small, 65.0	65.0
memberOpt OutCount	Integer	Total number of members who have opted out of all insurance plans.	Small, 65.0	65.0
memberWith DependentCount	Integer	Total number of members who have at least one dependent of any type.	Small, 65.0	65.0
memberWith OneChildCount	Integer	Total number of members who have exactly one child dependent.	Small, 65.0	65.0
memberWith SpouseCount	Integer	Total number of members who have a spouse listed as a dependent.	Small, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
partTime MemberCount	Integer	Total number of part-time members.	Small, 65.0	65.0
totalDependents	Integer	Total of all dependent records across all members in the census.	Small, 65.0	65.0
totalMember PlusDependent	Integer	Total of all members and all their dependents.	Small, 65.0	65.0
totalMembers	Integer	Total number of primary members in the census.	Small, 65.0	65.0

Group Census Group Class Summary

Output representation of census summary data aggregated by group class.

JSON example

```
{
  "groupCensus": {
    "id": "a0B1x000000XXXXX",
    "memberOptOutCount": 5,
    "mbrWithoutDependentCount": 20,
    "memberWithOneChildCount": 15,
    "mbrWithMoreThan1ChldCount": 10,
    "mbrWithMoreThan2ChldCount": 5,
    "memberWithDependentCount": 30,
    "memberWithSpouseCount": 25,
    "totalMemberPlusDependent": 85,
    "totalDependents": 45,
    "totalMembers": 40,
    "fullTimeMemberCount": 35,
    "partTimeMemberCount": 5,
    "grpCensusGrpClassSummaryList": [
      {
        "id": "a0C1x000000YYYYY",
        "groupCensusId": "a0B1x000000XXXXX",
        "groupClassId": "a0D1x000000ZZZZZ",
        "memberOptOutCount": 2,
        "mbrWithoutDependentCount": 10,
        "memberWithOneChildCount": 5,
        "mbrWithMoreThan1ChldCount": 3,
        "mbrWithMoreThan2ChldCount": 2,
        "memberWithDependentCount": 10,
        "memberWithSpouseCount": 8,
        "totalMemberPlusDependent": 28,
        "totalDependents": 15,
        "totalMembers": 13,
        "fullTimeMemberCount": 12,
        "partTimeMemberCount": 1
      }
    ]
  }
}
```

```

    ]
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
fullTime MemberCount	Integer	Total number of full-time members for this group class.	Small, 65.0	65.0
groupCensusId	String	ID of the group census.	Small, 65.0	65.0
groupClassId	String	ID of the group class.	Small, 65.0	65.0
id	String	ID of the group census group class summary record.	Small, 65.0	65.0
mbrWithMore Than1Chld Count	Integer	Total number of members who have more than one child dependent for this group class.	Small, 65.0	65.0
mbrWithMore Than2ChldCount	Integer	Total number of members who have more than two child dependents for this group class.	Small, 65.0	65.0
mbrWithout DependentCount	Integer	Total number of members who have no dependents for this group class.	Small, 65.0	65.0
memberOptOut Count	Integer	Total number of members who have actively opted out of all insurance plans for this group class.	Small, 65.0	65.0
memberWith DependentCount	Integer	Total number of members who have at least one dependent of any type, such as spouse or children for this group class.	Small, 65.0	65.0
memberWith OneChildCount	Integer	Total number of members who have exactly one child dependent for this group class.	Small, 65.0	65.0
memberWith SpouseCount	Integer	Total number of members who have a spouse listed as a dependent for this group class.	Small, 65.0	65.0
partTime MemberCount	Integer	Total number of part-time members for this group class.	Small, 65.0	65.0
totalDependents	Integer	Total number of dependents for this group class.	Small, 65.0	65.0
totalMember PlusDependent	Integer	Total of all members and all their dependents for this group class.	Small, 65.0	65.0
totalMembers	Integer	Total number of members for this group class.	Small, 65.0	65.0

Insurance Update Member Plans

Output representation for the update member plans operation.

JSON example

```
{
  "errors": [],
  "failedPlans": [],
  "isSuccess": true,
  "memberId": "0r6xx000000001aEAA",
  "updatedMemberPlans": [
    "0r8xx000000003jEAA"
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	List of errors that occurred for the API request.	Small, 65.0	65.0
failedPlans	String[]	List of plan IDs that failed to be updated.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the overall operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
memberId	String	ID of the group census member.	Small, 65.0	65.0
updatedMemberPlans	String[]	List of IDs for the member plan records that are successfully updated.	Small, 65.0	65.0

Insurance Add Member Plans

Output representation for the add member plans operation.

JSON example

```
{
  "errors": [],
  "isSuccess": true,
  "memberId": "0r6xx000000001aEAA",
  "memberPlans": [
    "0r8xx000000004kEAA",
    "0r8xx000000005lEAA"
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	List of errors that occurred during the API operation.	Small, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
isSuccess	Boolean	Indicates whether the API executed successfully (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
memberId	String	ID of the group census member.	Small, 65.0	65.0
memberPlans	String[]	List of IDs for the member plans that are successfully added.	Small, 65.0	65.0

Delete Member Plans

Output representation for the member plan deletion process.

JSON example

```
{
  "errors": [],
  "isSuccess": true,
  "memberId": "0r6p00000004C9wAAE"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error []	List of errors occurred during the API operation.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the API executed successfully (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
memberId	String	ID of the group census member.	Small, 65.0	65.0

Insurance Member Eligible Plans Response

Output representation for a member's eligible plans.

JSON example

This is a sample JSON for a successful API request.

```
{
  "memberId": "0r6xx0000000001AAA",
  "membersPlansDetails": {
    "values": [
      {
        "memberId": "0r6xx0000000001AAA",
        "plans": {
          "Medical": {
            "values": [
              {
                "attributes": [],
                "coverages": [
                  {
```

```

        "attributes": [],
        "coverages": [],
        "isOptional": false,
        "isPrimarySubscribed": false,
        "isSubscribed": false,
        "planId": "Orgxx000000001dAAA",
        "planName": "Annual Health Checkups"
      },
      {
        "attributes": [],
        "coverages": [],
        "isOptional": false,
        "isPrimarySubscribed": false,
        "isSubscribed": false,
        "planId": "Orgxx000000003FAAQ",
        "planName": "OPD Consults"
      }
    ],
    "isOptional": false,
    "isPrimarySubscribed": false,
    "isSubscribed": false,
    "planId": "Orgxx000000001AAA",
    "planName": "Bajaj Medical Premium"
  }
]
}
},
{
  "memberId": "0r6xx0000000002AAB",
  "plans": {
    "Medical": {
      "values": [
        {
          "attributes": [],
          "coverages": [
            {
              "attributes": [],
              "coverages": [],
              "isOptional": false,
              "isPrimarySubscribed": false,
              "isSubscribed": false,
              "planId": "Orgxx000000001dAAA",
              "planName": "Annual Health Checkups"
            }
          ],
          "isOptional": false,
          "isPrimarySubscribed": false,
          "isSubscribed": false,
          "planId": "Orgxx000000001AAA",
          "planName": "Bajaj Medical Premium"
        }
      ]
    }
  }
}

```

```

    }
  }
]
}
}

```

This is a sample JSON with error.

```

{
  "memberId": "0r6xx0000000001AAA",
  "membersPlansDetails": {
    "values": [
      {
        "error": {
          "additionalInfo": {},
          "code": "SF-0001-0005",
          "message": "Specify a valid input for memberId."
        },
        "memberId": "0r6xx0000000001AAA",
        "plans": {}
      }
    ]
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
memberId	String	ID for the primary group census member.	Small, 65.0	65.0
membersPlansDetails	Members Plans	List of members with their eligible plans or errors.	Small, 65.0	65.0

Members Plans

Output representation for a list of members with their eligible plans.

Property Name	Type	Description	Filter Group and Version	Available Version
values	Insurance Member Plans[]	List of members (primary and dependent) and their associated eligible plans or errors.	Small, 65.0	65.0

Insurance Member Plans

Output representation of the Insurance Get Member Plans service for an individual member.

JSON example

```

{
  "error": null,
  "memberId": "0r6xx0000000001AAA",
  "plans": {

```

```

    "Medical": {
      "values": [
        {
          "attributes": [],
          "coverages": [
            {
              "attributes": [],
              "coverages": [],
              "isOptional": false,
              "isPrimarySubscribed": false,
              "isSubscribed": false,
              "planId": "Orgxx000000001dAAA",
              "planName": "Annual Health Checkups"
            },
            {
              "attributes": [],
              "coverages": [],
              "isOptional": false,
              "isPrimarySubscribed": false,
              "isSubscribed": false,
              "planId": "Orgxx000000003FAAQ",
              "planName": "OPD Consults"
            }
          ],
          "isOptional": false,
          "isPrimarySubscribed": false,
          "isSubscribed": false,
          "planId": "Orgxx000000001AAA",
          "planName": "Bajaj Medical Premium"
        }
      ]
    }
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
error	Insurance Error	Error details if an error occurred while getting plans for the member.	Small, 65.0	65.0
memberId	String	ID of the group census member.	Small, 65.0	65.0
plans	Map<String, Insurance Plan List >	Map of eligible plans, where the key is the plan category, such as Medical and Dental, and the value is a list of plans in that category.	Small, 65.0	65.0

Insurance Plan List

Output representation for a list of contract group plans.

Property Name	Type	Description	Filter Group and Version	Available Version
values	Insurance Plan Details[]	List of contract group plan values.	Small, 65.0	65.0

Insurance Plan Details

Output representation for contract group plans.

JSON example

```
{
  "attributes": [],
  "coverages": [],
  "isOptional": false,
  "isPrimarySubscribed": true,
  "isSubscribed": true,
  "planDescription": "Gold medical coverage",
  "planId": "0gpp00000004C9wAAE",
  "planName": "Gold Plan"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
attributes	Object Map[]	Attributes for the plan.	Small, 65.0	65.0
coverages	Insurance Plan Details[]	Coverages for the contract group plan.	Small, 65.0	65.0
isOptional	Boolean	Indicates if the plan is optional for the group census member (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
isPrimary Subscribed	Boolean	Indicates if the primary group census member is subscribed to the plan (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
isSubscribed	Boolean	Indicates if the group census member is subscribed to the plan (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
planDescription	String	Description of the contract group plan.	Small, 65.0	65.0
planId	String	ID of the contract group plan.	Small, 65.0	65.0
planName	String	Name of the contract group plan.	Small, 65.0	65.0

Insurance Plans Eligibility Details

Output representation of the plan eligibility check for an.

JSON example

```
{
  "contextId": "42ce00ea6a44dcc0d308balea7c4c9bb455bd14bb0405ff03dca1b5371c42cda",
  "isSuccess": true,
  "errors": [],
  "memberId": "0r6RM0000000hlpYAA",
  "planEligibilityDetails": {
    "plans": [
      {
        "planId": "plan123",
        "isOptional": false,
        "productCategory": "Medical",
        "planName": "plan123",
        "attributes": [],
        "coverages": [
          {
            "planId": "coverageA",
            "isOptional": false,
            "productCategory": "Medical",
            "planName": "plan123"
          },
          {
            "planId": "coverageB",
            "isOptional": false,
            "productCategory": "Medical",
            "planName": "plan123"
          }
        ]
      }
    ]
  }
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
contextId	String	ID of the context that's used for the eligibility check.	Small, 66.0	66.0
errors	Insurance Error[]	List of errors that occurred during the eligibility check.	Small, 66.0	66.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	Small, 66.0	66.0
memberId	String	ID of the group census member.	Small, 66.0	66.0
planEligibility Details	Eligible Plan Details	Details of the eligible plans.	Small, 66.0	66.0

Eligible Plan Details

Output representation for the details of the eligible plans.

JSON example

```
{
  "messages": [],
  "plans": [{
    "isOptional": false,
    "productCategory": "Medical",
    "planName": "plan123",
    "attributes": [],
    "coverages": [
      {
        "planId": "coverageA",
        "isOptional": false,
        "productCategory": "Medical",
        "planName": "plan123"
      },
      {
        "planId": "coverageB",
        "isOptional": true,
        "productCategory": "Medical",
        "planName": "plan123"
      }
    ]
  }
]}
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
messages	Insurance Message []	List of informational or error messages related to the eligibility check.	Small, 66.0	66.0
plans	Insurance Group Plans []	List of contract group plans that the member is eligible for.	Small, 66.0	66.0

Insurance Message Output

Output representation of an insurance message, which can be informational or an error.

JSON example

```
{
  "message": "Plan is eligible for this member.",
  "type": "Info"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Content of the message.	Small, 65.0	65.0
type	String	Type of message, such as Info or Error.	Small, 65.0	65.0

Insurance Group Plans

Output representation of an insurance group plan.

JSON example

```
{
  "isOptional": false,
  "productCategory": "Medical",
  "planName": "plan123",
  "attributes": [],
  "coverages": [
    {
      "planId": "coverageA",
      "isOptional": false,
      "productCategory": "Medical",
      "planName": "plan123"
    },
    {
      "planId": "coverageB",
      "isOptional": false,
      "productCategory": "Medical",
      "planName": "plan123"
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
attributes	Insurance Object Map[]	List of attributes that are associated with the contract group plan.	Small, 66.0	66.0
coverages	Insurance Group Plans[]	List of coverages that are associated with this plan.	Small, 66.0	66.0
isOptional	Boolean	Indicates whether the plan is optional (<code>true</code>) or not (<code>false</code>).	Small, 66.0	66.0
planId	String	ID of the contract group plan.	Small, 66.0	66.0
planName	String	Name of the contract group plan.	Small, 66.0	66.0
productCategory	String	Category of the contract group plan.	Small, 66.0	66.0

Contract Creation Response

Output representation of the group benefits Quote-to-Contract process.

JSON example

```
{
  "errors": [],
  "isSuccess": true,
  "message": "Contract creation process has been initiated.",
}
```

```
{
  "requestId": "707p00000004C9wAAE"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	ConnectApiErrorResponse	List of errors that occurred during processing.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the contract creation is successfully initiated (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
message	String	Status message for the contract creation process.	Small, 65.0	65.0
requestId	String	ID of the asynchronous bulk request job.	Small, 65.0	65.0

Group Rating Response

Output representation of the Group Benefits Group Rating process.

JSON example

```
{
  "errors": [],
  "isSuccess": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	ConnectApiErrorResponse	List of errors that occurred during processing.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the group rating process is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0

Individual Enrollment Response

Output representation for the Individual Enrollment API.

JSON example

```
{
  "errors": [],
  "isSuccess": true,
  "policies": [
    {
      "plan": "Gold Plan",
      "policy": "0YTp00000004C9wAAE"
    }
  ]
}
```

```

    ]
  }

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Error[]	List of errors that are encountered during the enrollment process.	Small, 65.0	65.0
isSuccess	Boolean	Indicates whether the enrollment process is successful (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
policies	Family Policy Details[]	List of policies that are created as part of the enrollment process.	Small, 65.0	65.0

Family Policy Details

Output representation for individual group policy information.

JSON example

```

{
  "plan": "Gold Plan",
  "policy": "0YTp00000004C9wAAE"
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
plan	String	Name of the plan that's associated with the policy.	Small, 65.0	65.0
policy	String	ID of the insurance policy that's created.	Small, 65.0	65.0

Insurance Group Benefits Apex Reference

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. Insurance Group Benefits Apex reference includes built-in Apex classes and interfaces, grouped by namespace.

[ConnectApi Namespace](#)

The `ConnectApi` namespace (also called `Connect` in Apex) in Insurance Group Benefits provides classes for accessing the same data available in the Insurance Group Benefits Business APIs.

ConnectApi Namespace

The `ConnectApi` namespace (also called `Connect` in Apex) in Insurance Group Benefits provides classes for accessing the same data available in the Insurance Group Benefits Business APIs.

For information about working with the `ConnectApi` classes, see [Connect in Apex](#).

[InsuranceGroupBenefitsFamily Class](#)

Effectively manage the insurance plans that are offered to groups, such as employees and their dependents, by using the Group Benefits feature on the Digital Insurance platform. Simplify the Group Benefits workflow through a unified platform for all processes, including census management, quote configuration and pricing, contract creation, and enrollment.

[ConnectApi Input Classes](#)

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

[ConnectApi Output Classes](#)

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

InsuranceGroupBenefitsFamily Class

Effectively manage the insurance plans that are offered to groups, such as employees and their dependents, by using the Group Benefits feature on the Digital Insurance platform. Simplify the Group Benefits workflow through a unified platform for all processes, including census management, quote configuration and pricing, contract creation, and enrollment.

Namespace

`ConnectApi`

[InsuranceGroupBenefitsFamily Methods](#)

These methods are for `InsuranceGroupBenefitsFamily`. All methods are static.

InsuranceGroupBenefitsFamily Methods

These methods are for `InsuranceGroupBenefitsFamily`. All methods are static.

[postInsuranceAddMembers\(censusMgmtInputParam, groupCensusId\)](#)

Add new group census members.

[postInsuranceAddMemberPlans\(insuranceAddMemberPlansInputRepresentation, memberId\)](#)

Add plans for a group census member.

[postInsuranceBulkAddMembers\(bulkCensusMgmtInputParam, groupCensusId\)](#)

Add or update group census member data in bulk.

[postInsuranceCheckPlansEligibility\(plansEligibilityInputParam, memberId\)](#)

Check the eligibility of a group census member for a given set of plans and create a new eligibility context.

[patchInsuranceCheckPlansEligibility\(plansEligibilityInputParamWithContextId, memberId\)](#)

Check the eligibility of a group census member for a set of added and deleted plans within an existing context.

[postInsuranceCreateAccounts\(createAccountsInputParam, createPortalUsers\)](#)

Create person accounts and portal users for group census members.

[postInsuranceCreateContacts\(createContactsInputParam, createPortalUsers\)](#)

Create contacts and portal users for group census members.

[deleteInsuranceDeleteAllMembers\(groupCensusId\)](#)

Delete all census members for the specified group census ID.

[postInsuranceDeleteMemberPlans\(insuranceDeleteMemberPlansInputRepresentation, memberId\)](#)

Delete the specified plans for a group census member.

[postInsuranceGroupRating\(gbGroupRatingInputParam, quoteId\)](#)

Rate a group benefits quote to calculate premiums for all quote line items.

[postInsuranceIndividualEnrollment\(insuranceIndividualEnrollmentInputParam, memberId\)](#)

Enroll a group census member in one or more insurance plans, creating the necessary policy records.

[getInsuranceMemberEligiblePlans\(memberId, shouldIncludeDependents\)](#)

Get the enrollment plans for which a group census member and their dependents are eligible.

[postInsuranceMembersRating\(insuranceMemberRatingInputParam, memberId\)](#)

Calculate pricing for individual members of a group census, with the support for plan and individual coverages within the plan.

[postInsuranceSyncCensusSummary\(groupCensusId\)](#)

Synchronize and recalculate the summary data for a group census.

[patchInsuranceUpdateMembers\(censusMgmtInputParam, groupCensusId\)](#)

Update group census members.

[patchInsuranceUpdateMemberPlans\(insuranceUpdateMemberPlansInput, memberId\)](#)

Update the plans that are associated with a group census member.

postInsuranceAddMembers (censusMgmtInputParam, groupCensusId)

Add new group census members.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceCensusMgmtOutputRepresentation
postInsuranceAddMembers (ConnectApi.InsuranceCensusMgmtInputRepresentation
censusMgmtInputParam, String groupCensusId)
```

Parameters

censusMgmtInputParam

Type: [ConnectApi.InsuranceCensusMgmtInputRepresentation](#)

The input representation containing the census members to add.

groupCensusId

Type: String

ID of the group census.

Return Value

Type: [ConnectApi.InsuranceCensusMgmtOutputRepresentation](#)

postInsuranceAddMemberPlans(**insuranceAddMemberPlansInputRepresentation**, **memberId**)

Add plans for a group census member.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceAddMemberPlansOutputRepresentation
postInsuranceAddMemberPlans (ConnectApi.InsuranceAddMemberPlansInputRepresentation
insuranceAddMemberPlansInputRepresentation, String memberId)
```

Parameters

insuranceAddMemberPlansInputRepresentation

Type: [ConnectApi.InsuranceAddMemberPlansInputRepresentation](#)

Details to add plans.

memberId

Type: String

ID of the group census member to add plans for.

Return Value

Type: [ConnectApi.InsuranceAddMemberPlansOutputRepresentation](#)

Example

Here's an example of how to invoke the Insurance Add Member Plans API from Apex code.

```
// Add Member Plan(s) for a Group Census Member via Connect API (simple version)

// 1) Create the top-level input
ConnectApi.InsuranceAddMemberPlansInputRepresentation input = new
ConnectApi.InsuranceAddMemberPlansInputRepresentation();

// 2) Build one or more member plan entries
//   Required: contractGroupId (the plan you want to add for the member)
//   Tip: You don't need to set groupCensusMemberId on each plan here because it's passed
//       as a separate argument.
List<ConnectApi.GroupCensusMemberPlanInputRepresentation> plans = new
List<ConnectApi.GroupCensusMemberPlanInputRepresentation>();
```



```

ConnectApi.GroupCensusMemberPlanInputRepresentation plan1 = new
ConnectApi.GroupCensusMemberPlanInputRepresentation();
plan1.contractGroupPlanId = '0rgxx00000003kAAA'; // e.g., a2Cxx0000001234AAA
// Optional fields:
// plan1.name = 'Employee Medical Gold';
// plan1.id = 'existingMemberPlanIdIfUpdating'; // not used for add
plans.add(plan1);

ConnectApi.GroupCensusMemberPlanInputRepresentation plan2 = new
ConnectApi.GroupCensusMemberPlanInputRepresentation();
plan2.contractGroupPlanId = '0rgxx00000003jAAA';
// If adding more plans, repeat and add to the list
plans.add(plan2);

// 3) Wrap plans list into the input
//     The wrapper expects a list container object
input.groupCensusMemberPlans = plans;

// 4) Execute the POST API
//     URL param is the Group Census Member Id the plans will be added to
String memberId = '0r6xx00000009CzAAI';

try {
    ConnectApi.InsuranceAddMemberPlansOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceAddMemberPlans(input, memberId);

    System.debug('Add Member Plans result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error while adding member plans: ' + e.getMessage());
}

```

postInsuranceBulkAddMembers (bulkCensusMgmtInputParam, groupCensusId)

Add or update group census member data in bulk.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.InsuranceBulkCensusMgmtOutputRepresentation
postInsuranceBulkAddMembers (ConnectApi.InsuranceBulkCensusMgmtInputRepresentation
bulkCensusMgmtInputParam, String groupCensusId)

```

Parameters

*bulkCensusMgmtInputParam*Type: [ConnectApi.InsuranceBulkCensusMgmtInputRepresentation](#)

Input representation for the bulk census management process.

groupCensusId

Type: String

ID of the group census to process members for.

Return Value

Type: [ConnectApi.InsuranceBulkCensusMgmtOutputRepresentation](#)

Usage

This method corresponds to the

`/connect/insurance/census-management/group-censuses/groupCensusId/group-census-members/bulk-process`
REST API endpoint.**postInsuranceCheckPlansEligibility(plansEligibilityInputParam, memberId)**

Check the eligibility of a group census member for a given set of plans and create a new eligibility context.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsurancePlansEligibilityOutputRepresentation  
postInsuranceCheckPlansEligibility(ConnectApi.InsurancePlansEligibilityPostInputRepresentation  
plansEligibilityInputParam, String memberId)
```

Parameters

*plansEligibilityInputParam*Type: [ConnectApi.InsurancePlansEligibilityPostInputRepresentation](#)

Input representation that contains the list of plans to check.

memberId

Type: String

ID of the group census member whose eligibility must be checked.

Return Value

Type: [ConnectApi.InsurancePlansEligibilityOutputRepresentation](#)

Example

```
// Plans Eligibility (POST) via Connect API

// 1) Build the input with plans
ConnectApi.InsurancePlansEligibilityPostInputRepresentation input =
    new ConnectApi.InsurancePlansEligibilityPostInputRepresentation();

// Each plan requires a Contract Group Plan Id
ConnectApi.InsuranceGroupPlanInputRepresentation plan1 =
    new ConnectApi.InsuranceGroupPlanInputRepresentation();
plan1.planId = '0rgSG000000fXPYAY';

// Optionally add coverages under the plan if applicable
// plan1.coverages = new List<ConnectApi.InsuranceGroupPlanInputRepresentation>{ /* coverage
    items */ };

// Provide one or more plans
input.plans = new List<ConnectApi.InsuranceGroupPlanInputRepresentation>{ plan1 };

// 2) Execute the POST with the Group Census Member Id
String memberId = '0r6SG0000008n3CYAQ';

try {
    ConnectApi.InsurancePlansEligibilityOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceCheckPlansEligibility(
            input,
            memberId
        );

    System.debug('Eligibility (POST) result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Eligibility (POST) error: ' + e.getMessage());
}
```

patchInsuranceCheckPlansEligibility(plansEligibilityInputParamWithContextId, memberId)

Check the eligibility of a group census member for a set of added and deleted plans within an existing context.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsurancePlansEligibilityOutputRepresentation
patchInsuranceCheckPlansEligibility(ConnectApi.InsurancePlansEligibilityPatchInputRepresentation
plansEligibilityInputParamWithContextId, String memberId)
```

Parameters

*plansEligibilityInputParamWithContextId*Type: [ConnectApi.InsurancePlansEligibilityPatchInputRepresentation](#)

Input representation that contains the context ID and the plans that must be added or removed.

memberId

Type: String

ID of the group census member whose eligibility must be checked.

Return Value

Type: [ConnectApi.InsurancePlansEligibilityOutputRepresentation](#)

Example

```
// Plans Eligibility (PATCH) via Connect API

// 1) Build the patch input
ConnectApi.InsurancePlansEligibilityPatchInputRepresentation input =
    new ConnectApi.InsurancePlansEligibilityPatchInputRepresentation();

// Required: contextId from a prior eligibility session (POST response)
input.contextId = '0000000i14tq18g00291764218838971ecb8e4a0905c434cbe3f9beeb9f0739e';

// Plans to add
ConnectApi.InsuranceGroupPlanInputRepresentation addPlan =
    new ConnectApi.InsuranceGroupPlanInputRepresentation();
addPlan.planId = 'OrgSG0000000fjBYAQ';
input.addedPlans = new List<ConnectApi.InsuranceGroupPlanInputRepresentation>{ addPlan };

// Plans to remove
ConnectApi.InsuranceGroupPlanInputRepresentation removePlan =
    new ConnectApi.InsuranceGroupPlanInputRepresentation();
removePlan.planId = 'OrgSG0000000fU1YAI';
input.deletedPlans = new List<ConnectApi.InsuranceGroupPlanInputRepresentation>{ removePlan
};

// 2) Execute the PATCH with the same Group Census Member Id
String memberId = '0r6SG00000091EbYAI';

try {
    ConnectApi.InsurancePlansEligibilityOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.patchInsuranceCheckPlansEligibility(
            input,
            memberId
        );

    System.debug('Eligibility (PATCH) result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Eligibility (PATCH) error: ' + e.getMessage());
}
```

postInsuranceCreateAccounts(createAccountsInputParam, createPortalUsers)

Create person accounts and portal users for group census members.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceAccountsAndUsersCreationRepresentation
postInsuranceCreateAccounts (ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation
createAccountsInputParam, Boolean createPortalUsers)
```

Parameters

createAccountsInputParam

Type: [ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation](#)

Details to create person accounts.

createPortalUsers

Type: Boolean

Specifies whether to create portal users for the new person accounts (`true`) or not (`false`). The default is `false`.

Return Value

Type: [ConnectApi.InsuranceAccountsAndUsersCreationRepresentation](#)

Example

Here's an example of how to invoke the Create Accounts API from Apex code.

```
// Accounts and (optional) Portal Users creation via Connect API (simple version)

// 1) Create the top-level input
ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation input =
    new ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation();

// Provide either a Group Census Id OR a list of Group Census Member Ids

// Option A: Use the Group Census Id (creates accounts for all eligible members)
// input.groupCensusId = '0rfSG0000000vw7YAA';

// Option B: Or specify specific Group Census Member Ids
input.groupCensusMemberIdList = new List<String>{ '0r6SG00000008uTRYAY' };

// Optional: Person Account Record Type Name
input.personAccountRecordType = 'Person Account';
```

```
// Optional: Matching keys (used to find existing Person Accounts before creating new ones)
// input.matchingKeyList = new List<String>{ 'Email', 'ExternalId__c' };

// Optional: Field mappings from Group Census Member fields -> Account fields
// Example: map census FirstName/LastName to Person Account fields
// input.fieldMappings = new Map<String, String>{
//     'FirstName' => 'FirstName',
//     'LastName'  => 'LastName',
//     'Email'     => 'PersonEmail'
// };

// Optional: Provide portal user defaults (only applied if createPortalUsers = true)
ConnectApi.PortalUserCreationInputRepresentation userDefaults =
    new ConnectApi.PortalUserCreationInputRepresentation();
userDefaults.profileId = '00eSG00000JZg4t';
// userDefaults.localeSidKey = 'en_US';
// userDefaults.timeZoneSidKey = 'America/Los_Angeles';
// userDefaults.languageLocaleKey = 'en_US';
// userDefaults.emailEncodingKey = 'UTF-8';
input.userDetails = userDefaults;

// 2) Execute the POST API
Boolean createPortalUsers = true; // set false to only create accounts

try {
    ConnectApi.InsuranceAccountsAndUsersCreationRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceCreateAccounts(
            input,
            createPortalUsers
        );

    System.debug('Accounts and Users creation result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error during accounts/users creation: ' + e.getMessage());
}
```

postInsuranceCreateContacts(createContactsInputParam, createPortalUsers)

Create contacts and portal users for group census members.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceContactsAndUsersCreationRepresentation
postInsuranceCreateContacts(ConnectApi.InsuranceContactsAndUsersCreationInputRepresentation
createContactsInputParam, Boolean createPortalUsers)
```

Parameters

*createContactsInputParam*Type: [ConnectApi.InsuranceContactsAndUsersCreationInputRepresentation](#)

Details to create contacts and users.

createPortalUsers

Type: Boolean

Indicates whether to create portal users for the new contacts (true) or not (false). The default value is false.

Return Value

Type: [ConnectApi.InsuranceContactsAndUsersCreationRepresentation](#)

Example

Here's an example of how to invoke the Insurance Create Contracts API from Apex code.

```
// Create Group Benefits Contract via Connect API (simple version)

// 1) Create the top-level input
ConnectApi.InsuranceContractCreationInputRepresentation input =
    new ConnectApi.InsuranceContractCreationInputRepresentation();

// Required: Quote Id to convert
input.quoteId = '0Q0xx0000004C9QCAU';

// Optional dates (ISO-8601 strings)
input.contractStartDate = '2026-01-01';
input.contractEndDate   = '2026-12-31';
input.enrollmentStartDate = '2025-12-01';
input.enrollmentEndDate   = '2025-12-31';
// Optional: contract term in months
// input.contractTerm = 12;

// 2) (Optional) Additional fields per contract item (e.g., per quote line)
// Build a map-like structure using InsuranceApexStringMapInputRepresentation
List<ConnectApi.InsuranceApexStringRepresentation> fieldPairs = new
List<ConnectApi.InsuranceApexStringRepresentation>();
ConnectApi.InsuranceApexStringRepresentation kv1 = new
ConnectApi.InsuranceApexStringRepresentation();
kv1.key = 'CustomerSignedTitle';
kv1.value = 'Mr';
fieldPairs.add(kv1);

// One contract item (repeat and add to the list for more)
ConnectApi.ContractItem item = new ConnectApi.ContractItem();
item.entity = 'Contract'; // e.g., 'ContractLineItem__c' (use your org's API name)
// item.instanceKey = 'QLI-1'; // unique identification key from Quote
// Line Item
item.fields = fieldPairs;

// Wrap into list and assign to input.additionalFields
input.additionalFields = new List<ConnectApi.ContractItem>{ item };
```

```
// 3) Execute the Contract Creation API
try {
    ConnectApi.InsuranceContractCreationOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceGBQuoteToContract(input);

    System.debug('Contract creation result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error while creating contract: ' + e.getMessage());
}
```

deleteInsuranceDeleteAllMembers (groupCensusId)

Delete all census members for the specified group census ID.

API Version

65.0

Requires Chatter

No

Signature

```
public static void deleteInsuranceDeleteAllMembers (String groupCensusId)
```

Parameters

groupCensusId

Type: String

ID of the group census whose members must be deleted.

Return Value

Type: [ConnectApi.InsuranceCensusMgmtOutputRepresentation](#)

postInsuranceDeleteMemberPlans (insuranceDeleteMemberPlansInputRepresentation, memberId)

Delete the specified plans for a group census member.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation
postInsuranceDeleteMemberPlans (ConnectApi.InsuranceDeleteMemberPlansInputRepresentation
insuranceDeleteMemberPlansInputRepresentation, String memberId)
```

Parameters

insuranceDeleteMemberPlansInputRepresentation
 Type: [ConnectApi.InsuranceDeleteMemberPlansInputRepresentation](#)
 The input representation containing the list of plans to delete.

memberId
 Type: String
 ID of the group census member whose plans are being deleted.

Return Value

Type: [ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation](#)

Example

Here's an example of how to invoke the Insurance Delete Member Plans API from Apex code.

```
// Delete Member Plan(s) for a Group Census Member via Connect API (simple version)

// 1) Create the top-level input
ConnectApi.InsuranceDeleteMemberPlansInputRepresentation input = new
ConnectApi.InsuranceDeleteMemberPlansInputRepresentation();
// 2) Provide the list of existing Group Census Member Plan record Ids to delete
// Uses a simple string list wrapper
List<String> planIdsToDelete = new List<String>{
    '0rgxx000000003LAAQ',
    '0rgxx000000003UAAQ'
};
input.plans = planIdsToDelete;
// 3) Execute the POST Delete Member Plans API
// URL param is the Group Census Member Id whose plans you want to delete
String memberId = '0r6xx00000004LGAAAY';
try {
    ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceDeleteMemberPlans(input,
memberId);
    System.debug('Delete Member Plans result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error while deleting member plans: ' + e.getMessage());
}
```

postInsuranceGroupRating (gbGroupRatingInputParam, quoteId)

Rate a group benefits quote to calculate premiums for all quote line items.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceGroupRatingOutputRepresentation
postInsuranceGroupRating (ConnectApi.InsuranceGroupRatingInputRepresentation
gbGroupRatingInputParam, String quoteId)
```

Parameters

gbGroupRatingInputParam

Type: [ConnectApi.InsuranceGroupRatingInputRepresentation](#)

The details required for the group rating process.

quoteId

Type: String

ID of the quote to perform the group rating on.

Return Value

Type: [ConnectApi.InsuranceGroupRatingOutputRepresentation](#)

Example

Here's an example of how to invoke the Insurance Group Rating API from Apex code.

```
// Group Rating via Connect API (simple version)

// 1) Build the input
ConnectApi.InsuranceGroupRatingInputRepresentation input =
    new ConnectApi.InsuranceGroupRatingInputRepresentation();

// Required: Quote Line Item Ids to rate
// input.quoteLineItemIds = new List<String>{
//     '0QLxx00000001A1AAY', '0QLxx00000001A2AAY'
// };

// Optional: Rating date (ISO-8601, e.g., 2025-11-26)
input.ratingDate = '2025-11-26T14:19:30.000';

// Optional: Batch size (process quote line items in batches)
// input.batchSize = 100;

// 2) Execute the POST API with the Quote Id
String quoteId = '0Q0SG000000PcEz0AK'; // Quote Id (type 0Q0)

try {
```

```

    ConnectApi.InsuranceGroupRatingOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceGroupRating(
            input,
            quoteId
        );

    System.debug('Group Rating success: ' + result.isSuccess);
    System.debug('Errors: ' + JSON.serializePretty(result.errors));
} catch (Exception e) {
    System.debug('Group Rating error: ' + e.getMessage());
}

```

postInsuranceIndividualEnrollment(insuranceIndividualEnrollmentInputParam, memberId)

Enroll a group census member in one or more insurance plans, creating the necessary policy records.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.InsuranceIndividualEnrollmentOutputRepresentation
postInsuranceIndividualEnrollment (ConnectApi.InsuranceIndividualEnrollmentInputRepresentation
insuranceIndividualEnrollmentInputParam, String memberId)

```

Parameters

insuranceIndividualEnrollmentInputParam

Type: [ConnectApi.InsuranceIndividualEnrollmentInputRepresentation](#)

Details of the plans to enroll the member in.

memberId

Type: String

ID of the Group Census Member to enroll.

Return Value

Type: [ConnectApi.InsuranceIndividualEnrollmentOutputRepresentation](#)

Example

Here's an example of how to invoke the Individual Enrollment API from Apex code.

```

// Individual Enrollment via Connect API (simple version)

// 1) Create the top-level input

```

```

ConnectApi.InsuranceIndividualEnrollmentInputRepresentation input =
    new ConnectApi.InsuranceIndividualEnrollmentInputRepresentation();

// Required: target plan (Contract Group Plan Id)
input.plan = '0rgxx0000000CcAAI';

// Optional: specify a processing flow if your org uses one
// input.flowName = 'DefaultEnrollmentFlow';

// 2) Execute the POST API with the Group Census Member Id
String memberId = '0r6xx00000007G2AAI';

try {
    ConnectApi.InsuranceIndividualEnrollmentOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceIndividualEnrollment(input,
            memberId);

    System.debug('Individual Enrollment result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error during individual enrollment: ' + e.getMessage());
}

```

getInsuranceMemberEligiblePlans(memberId, shouldIncludeDependents)

Get the enrollment plans for which a group census member and their dependents are eligible.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.InsuranceGetMembersPlans getInsuranceMemberEligiblePlans(String
memberId, Boolean shouldIncludeDependents)

```

Parameters

memberId

Type: String

ID of the group census member.

shouldIncludeDependents

Type: Boolean

Indicates whether to include eligible plans for the member's dependents in the response (true) or not (false).

Return Value

Type: [ConnectApi.InsuranceGetMembersPlans](#)

Example

Here's an example of how to invoke the Insurance Get Member Eligible Plans API from Apex code.

```
// Get Member Eligible Plans via Connect API (simple version)

// Replace with an existing Group Census Member Id
String memberId = '0r6xx00000007BrAAI';

// Optional: include dependents' plans (requires API version >= 258)
Boolean includeDependents = true;

try {
    ConnectApi.InsuranceGetMembersPlans result =
        ConnectApi.InsuranceGroupBenefitsFamily.getInsuranceMemberEligiblePlans(memberId,
            includeDependents);

    System.debug('Eligible Plans response: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error while fetching eligible plans: ' + e.getMessage());
}
```

postInsuranceMembersRating(insuranceMemberRatingInputParam, memberId)

Calculate pricing for individual members of a group census, with the support for plan and individual coverages within the plan.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceMemberRatingRepresentation
postInsuranceMembersRating(ConnectApi.InsuranceMemberRatingInputRepresentation
insuranceMemberRatingInputParam, String memberId)
```

Parameters

insuranceMemberRatingInputParam

Type: [ConnectApi.InsuranceMemberRatingInputRepresentation](#)

Input representation for rating the members.

memberId

Type: String

ID of the group census member.

Return Value

Type: [ConnectApi.InsuranceMemberRatingRepresentation](#)

Example

Here's an example of how to invoke the Insurance Member Rating API from Apex code.

```
// Individual Member Rating via Connect API (simple version)

// 1) Build the input
ConnectApi.InsuranceMemberRatingInputRepresentation input =
    new ConnectApi.InsuranceMemberRatingInputRepresentation();

// Optional: Effective date (ISO-8601)
input.effectiveDate = '2025-11-26T15:26:00.000';

// Members to rate (one or more)
ConnectApi.InsuranceMemberRatingDetailsInputRepresentation memberDetails =
    new ConnectApi.InsuranceMemberRatingDetailsInputRepresentation();
memberDetails.memberId = '0r6SG0000008n8XYAQ'; // member to rate

// Plan(s) to rate for this member
ConnectApi.InsuranceMemberRatingPlansInputRepresentation plan =
    new ConnectApi.InsuranceMemberRatingPlansInputRepresentation();
plan.planId = '0rgSG0000000fgIY AQ';
plan.planCoverageIds = new List<String>{
    '0rgSG0000000fgJY AQ'
};

// Assign plans to the member and add to input
memberDetails.plans = new List<ConnectApi.InsuranceMemberRatingPlansInputRepresentation>{
    plan };
input.members = new List<ConnectApi.InsuranceMemberRatingDetailsInputRepresentation>{
    memberDetails };

// 2) Execute the POST API with the primary Group Census Member Id (family anchor)
String primaryMemberId = '0r6SG0000008n8OY AQ';

try {
    ConnectApi.InsuranceMemberRatingRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceMembersRating(
            input,
            primaryMemberId
        );

    System.debug('Member Rating ID: ' + result.id);
    System.debug('Pricing Results: ' + JSON.serializePretty(result.pricingResults));
    System.debug('Errors: ' + JSON.serializePretty(result.errors));
} catch (Exception e) {
    System.debug('Member Rating error: ' + e.getMessage());
}
```

postInsuranceSyncCensusSummary (groupCensusId)

Synchronize and recalculate the summary data for a group census.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceCensusSummarySyncOutputRepresentation  
postInsuranceSyncCensusSummary (ConnectApi.InsuranceCensusSummarySyncInputRepresentation  
censusSummarySyncInputParam, String groupCensusId)
```

Parameters

groupCensusId

Type: String

ID of the group census to synchronize.

Return Value

Type: [ConnectApi.InsuranceCensusSummarySyncOutputRepresentation](#)

Example

Here's an example of how to invoke the Insurance Sync Census Summary API from Apex code.

```
// Census Summary Sync via Connect API (simple version)  
  
// 1) Create the top-level input (no fields required)  
ConnectApi.InsuranceCensusSummarySyncInputRepresentation input =  
    new ConnectApi.InsuranceCensusSummarySyncInputRepresentation();  
  
// 2) Execute the POST API with the Group Census Id  
String groupCensusId = '0rffx000000000RAAQ';  
  
try {  
    ConnectApi.InsuranceCensusSummarySyncOutputRepresentation result =  
        ConnectApi.InsuranceGroupBenefitsFamily.postInsuranceSyncCensusSummary(input,  
        groupCensusId);  
  
    System.debug('Census Summary Sync result: ' + JSON.serializePretty(result));  
} catch (Exception e) {  
    System.debug('Error during Census Summary Sync: ' + e.getMessage());  
}
```

patchInsuranceUpdateMembers(censusMgmtInputParam, groupCensusId)

Update group census members.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceCensusMgmtOutputRepresentation  
patchInsuranceUpdateMembers (ConnectApi.InsuranceCensusMgmtInputRepresentation  
censusMgmtInputParam, String groupCensusId)
```

Parameters

censusMgmtInputParam

Type: [ConnectApi.InsuranceCensusMgmtInputRepresentation](#)

Details to update the census members.

groupCensusId

Type: String

ID of the group census.

Return Value

Type: [ConnectApi.InsuranceCensusMgmtOutputRepresentation](#)

patchInsuranceUpdateMemberPlans (insuranceUpdateMemberPlansInput, memberId)

Update the plans that are associated with a group census member.

API Version

65.0

Requires Chatter

No

Signature

```
public static ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation  
patchInsuranceUpdateMemberPlans (ConnectApi.InsuranceUpdateMemberPlansInputRepresentation  
insuranceUpdateMemberPlansInput, String memberId)
```

Parameters

insuranceUpdateMemberPlansInput

Type: [ConnectApi.InsuranceUpdateMemberPlansInputRepresentation](#)

Details to update the insurance plans for the census member.

memberId

Type: String

ID of the group census member whose plans must be updated.

Return Value

Type: `ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation`

Example

Here's an example of how to invoke the Insurance Update Member Plans API from Apex code.

```
// Update Member Plan(s) for a Group Census Member via Connect API (simple version)

// 1) Create the top-level input
ConnectApi.InsuranceUpdateMemberPlansInputRepresentation input = new
ConnectApi.InsuranceUpdateMemberPlansInputRepresentation();

// 2) Plans to ADD (provide Contract Group Plan Ids to add)
List<ConnectApi.GroupCensusMemberPlanInputRepresentation> added = new
List<ConnectApi.GroupCensusMemberPlanInputRepresentation>();

ConnectApi.GroupCensusMemberPlanInputRepresentation addPlan1 = new
ConnectApi.GroupCensusMemberPlanInputRepresentation();
addPlan1.contractGroupPlanId = '0rgxx000000003UAAQ'; // e.g., a2Cxx0000001234AAA
added.add(addPlan1);

// Assign added plans
input.addedPlans = added;

// 3) Plans to DELETE (provide existing Group Census Member Plan record Ids)
input.deletedPlans = new List<String>{
    // 'REPLACE_WITH_MEMBER_PLAN_ID_1',
    // 'REPLACE_WITH_MEMBER_PLAN_ID_2'
};

// 4) Execute the PATCH API with the Group Census Member Id
String memberId = '0r6xx000000004LGAAY';

try {
    ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation result =
        ConnectApi.InsuranceGroupBenefitsFamily.patchInsuranceUpdateMemberPlans(input,
memberId);

    System.debug('Update Member Plans result: ' + JSON.serializePretty(result));
} catch (Exception e) {
    System.debug('Error while updating member plans: ' + e.getMessage());
}
```

ConnectApi Input Classes

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

Input classes are concrete unless marked abstract in this documentation. Concrete input classes have public constructors that have no parameters.

Some methods have parameters that are typed with an abstract class. You must pass in an instance of a concrete child class for these parameters.

Most input class properties can be set. Read-only properties are noted in this documentation.

Insurance Group Benefits includes these Apex input classes.

[ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation](#)

Input representation for creating person accounts and portal users from group census members.

[ConnectApi.PortalUserCreationInputRepresentation](#)

Input representation containing the details for creating a portal user.

[ConnectApi.InsuranceContactsAndUsersCreationInputRepresentation](#)

Input representation to create contacts and portal users.

[ConnectApi.InsuranceMemberRatingInputRepresentation](#)

Input representation for the individual member rating API.

[ConnectApi.InsuranceMemberRatingDetailsInputRepresentation](#)

Input representation for the details of an individual member to be rated.

[ConnectApi.InsuranceMemberRatingPlansInputRepresentation](#)

Input representation for plan details used in individual rating.

[ConnectApi.InsuranceBulkCensusMgmtInputRepresentation](#)

Input representation for managing employee census data in bulk.

[ConnectApi.InsuranceCensusMgmtInputRepresentation](#)

Input representation for adding or updating census members.

[ConnectApi.GroupCensusMemberInputRepresentation](#)

Input representation for a group census member.

[ConnectApi.InsuranceUpdateMemberPlansInputRepresentation](#)

Input representation for updating the plans for a group census member.

[ConnectApi.InsuranceAddMemberPlansInputRepresentation](#)

Input representation for adding plans to a group census member.

[ConnectApi.GroupCensusMemberPlanInputRepresentation](#)

Input representation for a single plan that's associated with a group census member.

[ConnectApi.InsuranceDeleteMemberPlansInputRepresentation](#)

Input representation for deleting a member's plans.

[ConnectApi.InsurancePlansEligibilityPatchInputRepresentation](#)

Input representation for checking plan eligibility by updating an existing context.

[ConnectApi.InsurancePlansEligibilityPostInputRepresentation](#)

Input representation for checking plan eligibility by creating a new context.

[ConnectApi.InsuranceGroupPlanInputRepresentation](#)

Input representation for a contract group plan.

[ConnectApi.InsuranceGroupRatingInputRepresentation](#)

Input representation for rating a group benefits quote.

[ConnectApi.InsuranceIndividualEnrollmentInputRepresentation](#)

Input representation for enrolling an individual member.

[ConnectApi.InsuranceContractCreationInputRepresentation](#)

Input representation for creating contract.

[ConnectApi.ContractItem](#)

Addition fields to be added to the contract items.

[ConnectApi.InsuranceApexStringRepresentation](#)

Input representation for the Apex enabled string.

ConnectApi.InsuranceAccountsAndUsersCreationInputRepresentation

Input representation for creating person accounts and portal users from group census members.

Property	Type	Description	Required or Optional	Available Version
fieldMappings	Map<String, String>	Map of field API names from the Group Census Member object (key) to the Account object (value).	Optional	65.0
groupCensusId	String	ID of the group census that contains the members to process.	Required	65.0
groupCensusMemberIdList	List<String>	List of group census member IDs to be processed. If not provided, all members of the specified group census are processed.	Optional	65.0
matchingKeyList	List<String>	List of field API names on the Group Census Member object used to identify and prevent the creation of duplicate person accounts.	Optional	65.0
personAccountRecordType	String	Developer name of the person account record type to use for the new accounts.	Optional	65.0
userDetails	ConnectApi.PortalUserCreationInputRepresentation	Details to create portal users. This property is required if the <code>createPortalUsers</code> parameter in the method call is set to <code>true</code> .	Optional	65.0

ConnectApi.PortalUserCreationInputRepresentation

Input representation containing the details for creating a portal user.

Property	Type	Description	Required or Optional	Available Version
emailEncodingKey	String	Email encoding for the user, such as UTF-8 or ISO-8859-1.	Required	65.0
languageLocaleKey	String	Language for the user, such as en_US for U.S. English.	Required	65.0

Property	Type	Description	Required or Optional	Available Version
localeSidKey	String	Locale for the user, which affects the format of dates, times, and numbers.	Required	65.0
profileId	String	ID of the profile to assign to the new portal user.	Required	65.0
timeZoneSidKey	String	Time zone for the user, such as America or Los_Angeles.	Required	65.0

ConnectApi.InsuranceContactsAndUsersCreationInputRepresentation

Input representation to create contacts and portal users.

Property	Type	Description	Required or Optional	Available Version
fieldMappings	Map<String,String>	Map of field API names from the Group Census Member object (key) to the Contact object (value).	Optional	65.0
groupCensusId	String	ID of the group census.	Required	65.0
groupCensusMemberIdList	List<String>	List of group census member IDs.	Optional	65.0
matchingKeyList	List<String>	List of matching keys from Group Census Member to find existing contacts.	Optional	65.0
userDetails	PortalUserCreationInputRepresentation	Details to create portal users. This property is required if the <code>createPortalUsers</code> parameter in the method call is set to <code>true</code>	Optional	65.0

ConnectApi.InsuranceMemberRatingInputRepresentation

Input representation for the individual member rating API.

Property	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the member's rating is calculated.	Required	65.0
members	InsuranceMemberRatingDetailsInputRepresentation on page 343>	Details of the members and their plans to be rated.	Required	65.0

ConnectApi.InsuranceMemberRatingDetailsInputRepresentation

Input representation for the details of an individual member to be rated.

Property	Type	Description	Required or Optional	Available Version
memberId	String	ID of the member.	Required	65.0
plans	InsuranceMemberRatingPlansInputRepresentation	Plan details for rating.	Required	65.0

ConnectApi.InsuranceMemberRatingPlansInputRepresentation

Input representation for plan details used in individual rating.

Property	Type	Description	Required or Optional	Available Version
planCoverageIds	List<String>	Coverage IDs within the plan.	Optional	65.0
planId	String	ID of the plan.	Required	65.0

ConnectApi.InsuranceBulkCensusMgmtInputRepresentation

Input representation for managing employee census data in bulk.

Property	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of records to process in each batch.	Optional	65.0
contentDocumentId	String	ID of the content document (CSV file) that contains the census data.	Required	65.0
contentDocumentVersionId	String	ID of the specific version of the content document.	Optional	65.0
csvColumnMapping	Map<String,String>	Map where the key is the CSV column header and the value is the API name of the corresponding field in the GroupCensusMember object.	Required	65.0
duplicateDetectionFields	List<String>	List of field API names that are used to generate a unique key for identifying duplicate members.	Optional	65.0
skipBulkConfigurationRules	Boolean	Indicates whether to execute the configuration rules after processing the data (<code>true</code>) or not (<code>false</code>). The default value is <code>False</code> . This property is applicable only for the census management enrollment API.	Optional	65.0

ConnectApi.InsuranceCensusMgmtInputRepresentation

Input representation for adding or updating census members.

Property	Type	Description	Required or Optional	Available Version
duplicateDetectionFields	List<String>	List of field API names that are used to generate a unique key for identifying duplicate members.	Optional	65.0
flowName	String	Name of the flow to use for processing group member data.	Optional	65.0
members	~~GroupCensusMemberInputRepresentation~~	List of group census members to add or update.	Required	65.0

ConnectApi.GroupCensusMemberInputRepresentation

Input representation for a group census member.

This class contains properties for all the fields on the Group Census Member object.

Property	Type	Description	Required or Optional	Available Version
accountId	String	Person account ID.	Optional	65.0
additionalFields	~~CustomAdditionalFields~~	List of key-value pairs for populating custom fields or other fields.	Optional	65.0
annualEligibleSalary	Double	Annual salary that's eligible for benefits calculations.	Optional	65.0
associationWithPrimaryMember	String	Association to primary member for dependents, such as Spouse, Child.	Optional	65.0
birthdate	String	Member's date of birth in YYYY-MM-DD format.	Optional	65.0
city	String	City of the member's address.	Optional	65.0
contactId	String	ID of the Contact record of the member.	Optional	65.0
country	String	Country of the member's address.	Optional	65.0
dependentCount	Integer	Number of dependents that are associated with the member.	Optional	65.0
email	String	Email address of the member.	Optional	65.0
firstName	String	First name of the member.	Required to add a member	65.0
fullTimeEquivalent	Double	Work capacity of the member, where 1.0 indicates a full-time employee.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
gender	String	Gender of the member.	Optional	65.0
groupCensusId	String	ID of the Group Census that the member belongs to.	Optional	65.0
groupClassId	String	ID for the specific class or category within the group that the member belongs to.	Optional	65.0
groupClassName	String	Name of the group class.	Optional	65.0
groupCensusSummaryId	String	ID of the related group census group class summary record.	Optional	65.0
id	String	ID of the group census member.	Required to update member details	65.0
isOptOutAllPlans	Boolean	Indicates whether the member has opted out of all available insurance plans (<code>true</code>) or not (<code>false</code>).	Optional	65.0
isPortalUser	Boolean	Indicates whether a community portal user must be created for this member (<code>true</code>) or not (<code>false</code>).	Optional	65.0
joinDate	String	Date the member joined the group or company.	Optional	65.0
lastname	String	Last name of the member.	Required to add member	65.0
memberKey	String	Unique, externally defined key that's used for identifying and preventing duplicate members, often created by concatenating several fields.	Optional	65.0
memberType	String	Type of member, such as Primary or Dependent.	Optional	65.0
name	String	Full name of the member.	Optional	65.0
optOutPlanTypes	String	Types of plans the member has opted out of.	Optional	65.0
phoneNumber	String	Phone number of the member.	Optional	65.0
policyStartDate	String	Date when the member's policy coverage begins.	Optional	65.0
postalCode	String	Postal or ZIP code of the member's address.	Optional	65.0
primaryGroupCensusId	String	ID of the primary group census member the dependents are associated with.	Optional	65.0
primaryMemberCategory	String	Category of the primary member.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
primaryMemberSource	String	System or source from which the member's data originated.	Optional	65.0
relationshipPrimaryMember	String	Member's relationship to the primary insured.	Optional	65.0
smokerStatus	String	Smoking status of the member, such as Smoker or Non-smoker.	Optional	65.0
sourceSystemIdentifier	String	ID from an external source system, that's used for integration purposes.	Optional	65.0
state	String	State or province the member belongs to.	Optional	65.0
status	String	Current status of the member.	Optional	65.0
street	String	Street address of the member.	Optional	65.0

ConnectApi.InsuranceUpdateMemberPlansInputRepresentation

Input representation for updating the plans for a group census member.

Property	Type	Description	Required or Optional	Available Version
addedPlans	~~GroupCensusMemberPlanRepresentation~~	List of plans to add for the member.	Optional	65.0
deletedPlans	List<String>	List of group census member plan IDs to delete.	Optional	65.0

ConnectApi.InsuranceAddMemberPlansInputRepresentation

Input representation for adding plans to a group census member.

Property	Type	Description	Required or Optional	Available Version
groupCensusMemberPlans	~~GroupCensusMemberPlanRepresentation~~	List of plan details to add for the member.	Required	65.0

ConnectApi.GroupCensusMemberPlanInputRepresentation

Input representation for a single plan that's associated with a group census member.

Property	Type	Description	Required or Optional	Available Version
contractGroupPlanId	String	ID of the contract group plan.	Required	65.0
groupCensusMemberId	String	ID of the group census member.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
id	String	ID of the group census member plan record.	Required to update plan	65.0
name	String	Name of the group census member plan.	Optional	65.0

ConnectApi.InsuranceDeleteMemberPlansInputRepresentation

Input representation for deleting a member's plans.

Property	Type	Description	Required or Optional	Available Version
plans	List<String>	List of group census member plan IDs to delete.	Required	65.0

ConnectApi.InsurancePlansEligibilityPatchInputRepresentation

Input representation for checking plan eligibility by updating an existing context.

Property	Type	Description	Required or Optional	Available Version
addedPlans	~~String~~ List<String>	List of contract group plans to add to the eligibility check.	Optional	65.0
contextId	String	ID of the existing eligibility context to update.	Required	65.0
deletedPlans	~~String~~ List<String>	List of contract group plans to remove from the eligibility check.	Optional	65.0

ConnectApi.InsurancePlansEligibilityPostInputRepresentation

Input representation for checking plan eligibility by creating a new context.

Property	Type	Description	Required or Optional	Available Version
plans	~~String~~ List<String>	List of contract group plans to check for eligibility.	Required	65.0

ConnectApi.InsuranceGroupPlanInputRepresentation

Input representation for a contract group plan.

Property	Type	Description	Required or Optional	Available Version
coverages	ConnectApi.ContractItem on page 347>	List of coverages that are associated with this plan.	Optional	65.0
planId	String	ID of the contract group plan.	Required	65.0

ConnectApi.InsuranceGroupRatingInputRepresentation

Input representation for rating a group benefits quote.

Property	Type	Description	Required or Optional	Available Version
batchSize	Integer	Number of records to process in each batch for the rating calculation.	Optional	65.0
quoteLineItemIds	List<String>	List of quote line item IDs to be included in the rating process.	Optional	65.0
ratingDate	String	Date to use for the rating calculation. If not specified, the API uses the effective date of the quote.	Optional	65.0

ConnectApi.InsuranceIndividualEnrollmentInputRepresentation

Input representation for enrolling an individual member.

Property	Type	Description	Required or Optional	Available Version
flowName	String	API name of an optional flow to run as part of the enrollment process.	Optional	65.0
plan	String	Comma-separated string of contract group plan IDs to enroll the member in.	Required	65.0

ConnectApi.InsuranceContractCreationInputRepresentation

Input representation for creating contract.

Property	Type	Description	Required or Optional	Available Version
additionalFields	ConnectApi.ContractItem on page 349>	List of additional fields to add to contract items.	Optional	65.0
contractEndDate	String	End date of the contract.	Required	65.0
contractStartDate	String	Start date of the contract.	Required	65.0

Property	Type	Description	Required or Optional	Available Version
contractTerm	Integer	Contract terms in months.	Optional	65.0
enrollmentEndDate	String	End date for plan enrollment.	Optional	65.0
enrollmentStartDate	String	Start date for plan enrollment.	Optional	65.0
quoteId	String	ID of the quote to create the contract from.	Required	65.0

ConnectApi.ContractItem

Addition fields to be added to the contract items.

Property	Type	Description	Required or Optional	Available Version
entity	String	API name of the entity to add fields to, such as ContractGroupPlan.	Optional	65.0
fields	ConnectApi.StringRepresentation on page 349>	List of key-value pairs that represent the fields and their values for the entity.	Optional	65.0
instanceKey	String	Unique instance key of the Quote Line Item.	Optional	65.0

ConnectApi.InsuranceApexStringRepresentation

Input representation for the Apex enabled string.

Property	Type	Description	Required or Optional	Available Version
key	String	API name of the field.	Optional	65.0
value	String	Value to assign to the field.	Optional	65.0

ConnectApi Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

All properties are read-only, except for instances of output classes created within test code.

All output classes are concrete unless marked abstract in this documentation.

All concrete output classes have no-argument constructors that you can invoke only from test code. See [Testing ConnectApi Code](#).

Insurance Group Benefits includes these Apex output classes.

[ConnectApi.InsuranceAccountsAndUsersCreationRepresentation](#)

Output representation for the person account and user creation process.

[ConnectApi.InsuranceErrorRepresentation](#)

Output representation for an error that occurred during an insurance process.

[ConnectApi.InsuranceMemberAccountAndUserDetailsRepresentation](#)

Output representation that maps a source group census member to the IDs of the newly created person account and portal user.

[ConnectApi.InsuranceContactsAndUsersCreationRepresentation](#)

Output representation of the contact and user creation process.

[ConnectApi.InsuranceMemberContactAndUserDetailsRepresentation](#)

Output representation of the created contact and portal user details for a group census member.

[ConnectApi.InsuranceMemberRatingRepresentation](#)

Output representation of the rating results for individual members.

[ConnectApi.InsuranceMemberRatingDetailsRepresentation](#)

Output representation of rating details for an individual member.

[ConnectApi.InsurancePlanRatingRepresentation](#)

Output representation of pricing results for an individual plan.

[ConnectApi.InsuranceCoverageRatingRepresentation](#)

Output representation of pricing results for an individual coverage.

[ConnectApi.InsuranceContributionRepresentation](#)

Output representation of contribution information in individual rating.

[ConnectApi.InsuranceObjectMapRepresentation](#)

Output representation for attributes and other mapped data.

[ConnectApi.InsuranceBulkCensusMgmtOutputRepresentation](#)

Output representation for the bulk employee census management process.

[ConnectApi.InsuranceCensusMgmtOutputRepresentation](#)

Output representation for the insurance census management process.

[ConnectApi.InsuranceCensusSummarySyncOutputRepresentation](#)

Output representation for the insurance employee census summary synchronization process.

[ConnectApi.InsuranceGroupCensusOutputRepresentation](#)

Output representation of a group census summary.

[ConnectApi.InsuranceGrpCensusGrpClassSummaryOutputRepresentation](#)

Output representation of census summary data aggregated by group class.

[ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation](#)

Output representation for the member plan update process.

[ConnectApi.InsuranceAddMemberPlansOutputRepresentation](#)

Output representation for adding plans to a member.

[ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation](#)

Output representation for the member plan deletion process.

[ConnectApi.InsuranceGetMembersPlans](#)

Output representation for a member's eligible plans.

[ConnectApi.InsuranceMembersPlansDetailsListRepresentation](#)

A wrapper for a list of members and their eligible plans.

[ConnectApi.InsuranceGetMemberPlansOutputRepresentation](#)

Output representation of eligible plans for a single member.

[ConnectApi.InsurancePlanListOutputRepresentation](#)

A wrapper for a list of insurance plans.

[ConnectApi.InsurancePlanOutputRepresentation](#)

Output representation of an insurance plan.

[ConnectApi.InsurancePlansEligibilityOutputRepresentation](#)

Output representation for the employee plan eligibility check.

[ConnectApi.EligiblePlanDetailsOutputRepresentation](#)

Output representation containing the details of eligible plans.

[ConnectApi.InsuranceGroupPlanOutputRepresentation](#)

Output representation of an insurance group plan.

[ConnectApi.InsuranceMessageOutputRepresentation](#)

Output representation of insurance message.

[ConnectApi.InsuranceGroupRatingOutputRepresentation](#)

Output representation for the group benefits rating process.

[ConnectApi.InsuranceIndividualEnrollmentOutputRepresentation](#)

Output representation for the individual enrollment process.

[ConnectApi.FamilyPolicyInfoOutputRepresentation](#)

Output representation of the created policy information for an enrollment.

ConnectApi.InsuranceAccountsAndUsersCreationRepresentation

Output representation for the person account and user creation process.

Property Name	Type	Description	Available Version
accountsAndUsersDetails	ConnectApi.AccountAndUserDetailsRepresentation on page 352>	Details of the objects that contain the IDs of the person accounts and users that are created and mapped to their source group census members.	65.0
errors	ConnectApi.InsuranceErrorRepresentation on page 351>	Details of errors that occurred during the operation. Each error includes details about why a specific group census member failed to process.	65.0
groupCensusId	String	ID of the group census that's processed.	65.0
isSuccess	Boolean	Indicates whether the overall operation was successful (<code>true</code>) or not (<code>false</code>). If any member fails to process, this value is <code>false</code> .	65.0

ConnectApi.InsuranceErrorRepresentation

Output representation for an error that occurred during an insurance process.

Property Name	Type	Description	Available Version
additionalInfo	Map<String, Object>	Map of additional details about the error, such as the ID of the record that caused the failure.	65.0

Property Name	Type	Description	Available Version
code	String	Error code.	65.0
message	String	Detailed error message.	65.0

ConnectApi.InsuranceMemberAccountAndUserDetailsRepresentation

Output representation that maps a source group census member to the IDs of the newly created person account and portal user.

Property Name	Type	Description	Available Version
groupCensusId	String	ID of the group census.	65.0
groupCensusMemberId	String	ID of the source group census member.	65.0
personAccountId	String	ID of the person account that's created.	65.0
userId	String	ID of the portal user that's created. This property is null if portal user creation isn't requested.	65.0

ConnectApi.InsuranceContactsAndUsersCreationRepresentation

Output representation of the contact and user creation process.

Property Name	Type	Description	Available Version
contactsAndUsersDetails	Details of the contacts and users that are created.	Details of the contacts and users that are created.	65.0
errors	Errors that occurred during the operation.	Errors that occurred during the operation.	65.0
groupCensusId	String	ID of the group census.	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.InsuranceMemberContactAndUserDetailsRepresentation

Output representation of the created contact and portal user details for a group census member.

Property Name	Type	Description	Available Version
contactId	String	ID of the created or existing contact.	65.0
groupCensusId	String	ID of the group census.	65.0
groupCensusMemberId	String	ID of the group census member.	65.0
userId	String	ID of the created portal user.	65.0

ConnectApi.InsuranceMemberRatingRepresentation

Output representation of the rating results for individual members.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceErrorsRepresentation on page 351>	Error details.	65.0
id	String	A unique identifier for this rating representation.	65.0
pricingResults	ConnectApi.InsurancePricingResultsRepresentation on page 353>	Details of pricing results from individual ratings.	65.0

ConnectApi.InsuranceMemberRatingDetailsRepresentation

Output representation of rating details for an individual member.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceErrorsRepresentation on page 351>	List of error details.	65.0
memberId	String	ID of the member.	65.0
plans	ConnectApi.InsurancePlanRatingRepresentation on page 353>	Rating details for the plan.	65.0

ConnectApi.InsurancePlanRatingRepresentation

Output representation of pricing results for an individual plan.

Property Name	Type	Description	Available Version
contributions	ConnectApi.InsuranceContributionRepresentation on page 354	Contribution details for the plan, if applicable.	65.0
coverages	ConnectApi.InsuranceCoverageRatingRepresentation on page 353>	Coverage details for the plan.	65.0
errors	ConnectApi.InsuranceErrorsRepresentation on page 351>	Error details.	65.0
id	String	ID of the plan.	65.0
proratedRates	ConnectApi.InsuranceProratedRatesRepresentation on page 354	Prorated rates for the plan.	65.0
rates	ConnectApi.InsuranceRateRepresentation on page 354	Rate details for the plan.	65.0

ConnectApi.InsuranceCoverageRatingRepresentation

Output representation of pricing results for an individual coverage.

Property Name	Type	Description	Available Version
contributions	ConnectApi.InsuranceContributionRepresentation on page 354	Contribution details for the coverage, if applicable.	65.0
coverageId	String	ID of the coverage.	65.0
errors	ConnectApi.InsuranceErrorResponse on page 351>	Error details.	65.0
proratedRates	ConnectApi.InsuranceObjectMapRepresentation	Prorated rates for the coverage.	65.0
rates	ConnectApi.InsuranceObjectMapRepresentation on page 354	Rate details for the coverage.	65.0

ConnectApi.InsuranceContributionRepresentation

Output representation of contribution information in individual rating.

Property Name	Type	Description	Available Version
employeePremiumAmount	Double	Premium amount that's contributed by an employee.	65.0
employerPremiumAmount	Double	Premium amount that's contributed by an employer.	65.0
errors	ConnectApi.InsuranceErrorResponse on page 351>	Error details.	65.0
groupClassContributionId	String	ID of the contribution record.	65.0

ConnectApi.InsuranceObjectMapRepresentation

Output representation for attributes and other mapped data.

Property Name	Type	Description	Available Version
data	Map<String, Object>	Map with a string key and an object value.	65.0

ConnectApi.InsuranceBulkCensusMgmtOutputRepresentation

Output representation for the bulk employee census management process.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceErrorResponse	List of errors that occurred during processing.	65.0
isSuccess	Boolean	Indicates whether the bulk operation is successfully initiated (<code>true</code>) or not (<code>false</code>).	65.0
requestId	String	ID of the asynchronous bulk request.	65.0

ConnectApi.InsuranceCensusMgmtOutputRepresentation

Output representation for the insurance census management process.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceCensusMgmtOutputRepresentation	List of errors that occurred during processing.	65.0
isSuccess	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	65.0
memberIds	List<String>	IDs of the group census members that are successfully created or updated.	65.0

ConnectApi.InsuranceCensusSummarySyncOutputRepresentation

Output representation for the insurance employee census summary synchronization process.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceCensusSummarySyncOutputRepresentation	List of errors that occurred during synchronization.	65.0
groupCensus	ConnectApi.InsuranceCensusSummarySyncOutputRepresentation on page 355	Details of the synchronized group census summary.	65.0
isSuccess	Boolean	Indicates whether the synchronization is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.InsuranceGroupCensusOutputRepresentation

Output representation of a group census summary.

Property Name	Type	Description	Available Version
fullTimeMemberCount	Integer	Total number of full-time members.	65.0
groupCensusGroupClassSummaryList	ConnectApi.InsuranceGroupCensusOutputRepresentation on page 356>	List of census summaries that are aggregated by group class.	65.0
id	String	ID of the group census summary.	65.0
memberWithMoreThan1ChildCount	Integer	Total number of members who have more than one child dependent.	65.0
memberWithMoreThan2ChildCount	Integer	Total number of members who have more than two child dependents.	65.0
memberWithoutDependentCount	Integer	Total number of members who have no dependents.	65.0
memberOptOutCount	Integer	Total number of members who have actively opted out of all insurance plans.	65.0
memberWithDependentCount	Integer	Total number of members who have at least one dependent of any type.	65.0

Property Name	Type	Description	Available Version
memberWithOneChildCount	Integer	Total number of members who have exactly one child dependent.	65.0
memberWithSpouseCount	Integer	Total number of members who have a spouse listed as a dependent.	65.0
partTimeMemberCount	Integer	Total number of part-time members.	65.0
totalDependents	Integer	Total of all dependent records across all members in the census.	65.0
totalMemberPlusDependent	Integer	Total of all members and all their dependents.	65.0
totalMembers	Integer	Total number of primary members in the census.	65.0

ConnectApi.InsuranceGrpCensusGrpClassSummaryOutputRepresentation

Output representation of census summary data aggregated by group class.

Property Name	Type	Description	Available Version
fullTimeMemberCount	Integer	Total number of full-time members for this group class.	65.0
groupCensusId	String	ID of the group census.	65.0
groupClassId	String	ID of the group class.	65.0
id	String	ID of the group census group class summary record.	65.0
memberWithMoreThan1ChildCount	Integer	Total number of members who have more than one child dependent for this group class.	65.0
memberWithMoreThan2ChildCount	Integer	Total number of members who have more than two child dependents for this group class.	65.0
memberWithoutDependentCount	Integer	Total number of members who have no dependents for this group class.	65.0
memberOptOutCount	Integer	Total number of members who have actively opted out of all insurance plans for this group class.	65.0
memberWithDependentCount	Integer	Total number of members who have at least one dependent of any type, such as spouse or children for this group class.	65.0
memberWithOneChildCount	Integer	Total number of members who have exactly one child dependent for this group class.	65.0
memberWithSpouseCount	Integer	Total number of members who have a spouse listed as a dependent for this group class.	65.0
partTimeMemberCount	Integer	Total number of part-time members for this group class.	65.0

Property Name	Type	Description	Available Version
totalDependents	Integer	Total number of dependents for this group class.	65.0
totalMemberPlusDependent	Integer	Total of all members and all their dependents for this group class.	65.0
totalMembers	Integer	Total number of members for this group class.	65.0

ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation

Output representation for the member plan update process.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceUpdateMemberPlansOutputRepresentation	List of errors that occurred during the update.	65.0
failedPlans	List<String>	List of plan IDs that failed to be added or deleted.	65.0
isSuccess	Boolean	Indicates whether the overall operation is successful (<code>true</code>) or not (<code>false</code>).	65.0
memberId	String	ID of the group census member.	65.0
updatedMemberPlans	List<String>	List of IDs for the member plan records that are successfully updated.	65.0

ConnectApi.InsuranceAddMemberPlansOutputRepresentation

Output representation for adding plans to a member.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceAddMemberPlansOutputRepresentation on page 351>	List of errors that occurred while adding plans.	64.0
isSuccess	Boolean	Indicates whether the API executed successfully.	64.0
memberId	String	ID of the group census member.	64.0
memberPlans	List<String>	List of IDs for the member plan records that are successfully added.	64.0

ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation

Output representation for the member plan deletion process.

Property Name	Type	Description	Available Version
errors	ConnectApi.InsuranceDeleteMemberPlansOutputRepresentation	List of errors that occurred during the deletion process.	65.0

Property Name	Type	Description	Available Version
isSuccess	Boolean	Indicates whether the API executed successfully (<code>true</code>) or not (<code>false</code>).	65.0
memberId	String	ID of the group census member.	65.0

ConnectApi.InsuranceGetMembersPlans

Output representation for a member's eligible plans.

Property Name	Type	Description	Available Version
memberId	String	ID of the primary group census member.	65.0
membersPlansDetails	ConnectApi.InsuranceMembersPlansDetailsListRepresentation on page 358	List of the eligible plans for the primary member and any dependents.	65.0

ConnectApi.InsuranceMembersPlansDetailsListRepresentation

A wrapper for a list of members and their eligible plans.

Property Name	Type	Description	Available Version
values	ConnectApi.InsuranceMembersPlansDetailsListRepresentation on page 358	List of members (primary and dependent) and their associated eligible plans or errors.	65.0

ConnectApi.InsuranceGetMemberPlansOutputRepresentation

Output representation of eligible plans for a single member.

Property Name	Type	Description	Available Version
error	ConnectApi.InsuranceErrorRepresentation on page 351	Error details that are associated with the API request.	65.0
memberId	String	ID of the group census member (primary or dependent).	65.0
plans	Map<String, ConnectApi.InsurancePlanListOutputRepresentation> on page 358	Map of eligible plans, where the key is the plan category, such as Medical and Dental, and the value is a list of plans in that category.	65.0

ConnectApi.InsurancePlanListOutputRepresentation

A wrapper for a list of insurance plans.

Property Name	Type	Description	Available Version
values	ConnectApi.InsurancePlanOutputRepresentation on page 359>	List of insurance plans.	65.0

ConnectApi.InsurancePlanOutputRepresentation

Output representation of an insurance plan.

Property Name	Type	Description	Available Version	
attributes	ConnectApi.InsurancePlanAttributesRepresentation on page 354>	Attributes for the plan.	65.0	65.0
coverages	ConnectApi.InsurancePlanCoverageRepresentation on page 359>	Coverages for the plan.	65.0	65.0
isOptional	Boolean	Indicates if the plan is optional for the group census member (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
isPrimarySubscribed	Boolean	Indicates if the primary group census member is subscribed to the plan (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
isSubscribed	Boolean	Indicates if the group census member is subscribed to the plan (<code>true</code>) or not (<code>false</code>).	Small, 65.0	65.0
planDescription	String	Description of the contract group plan.	Small, 65.0	65.0
planId	String	ID of the contract group plan.	Small, 65.0	65.0
planName	String	Name of the contract group plan.	Small, 65.0	65.0

ConnectApi.InsurancePlansEligibilityOutputRepresentation

Output representation for the employee plan eligibility check.

Property Name	Type	Description	Available Version
contextId	String	ID of the context that's used for the eligibility check.	65.0
errors	ConnectApi.InsurancePlanErrorsRepresentation on page 351>	List of errors that occurred during the eligibility check.	65.0
isSuccess	Boolean	Indicates whether the API operation is successful (<code>true</code>) or not (<code>false</code>).	65.0
memberId	String	ID of the group census member.	65.0
planEligibilityDetails	ConnectApi.InsurancePlanEligibilityDetailsRepresentation on page 360	Details of the plan eligibility check.	65.0

ConnectApi.EligiblePlanDetailsOutputRepresentation

Output representation containing the details of eligible plans.

Property Name	Type	Description	Available Version
messages	ConnectApi.MessageOutputRepresentation	List of informational or warning messages related to the eligibility results.	65.0
plans	ConnectApi.PlanOutputRepresentation	List of the contract group plans that the member is eligible for.	65.0

ConnectApi.InsuranceGroupPlanOutputRepresentation

Output representation of an insurance group plan.

Property Name	Type	Description	Available Version
attributes	ConnectApi.AttributeOutputRepresentation on page 354>	List of attributes for the contract group plan.	65.0
coverages	ConnectApi.CoverageOutputRepresentation on page 360>	List of coverages for the plan.	65.0
isOptional	Boolean	Indicates whether the plan is optional (<code>true</code>) or not (<code>false</code>).	65.0
planId	String	ID of the contract group plan.	65.0
planName	String	Name of the contract group plan.	65.0
productCategory	String	Category of the contract group plan.	65.0

ConnectApi.InsuranceMessageOutputRepresentation

Output representation of insurance message.

Property Name	Type	Description	Available Version
message	String	Content of the message.	65.0
type	String	Type of message, such as <code>Info</code> or <code>Error</code> .	65.0

ConnectApi.InsuranceGroupRatingOutputRepresentation

Output representation for the group benefits rating process.

Property Name	Type	Description	Available Version
errors	ConnectApi.ErrorOutputRepresentation on page 351>	List of errors that are encountered during the rating process.	65.0

Property Name	Type	Description	Available Version
isSuccess	Boolean	Indicates whether the group rating process is successful (<code>true</code>) or not (<code>false</code>).	65.0

ConnectApi.InsuranceIndividualEnrollmentOutputRepresentation

Output representation for the individual enrollment process.

Property Name	Type	Description	Available Version
errors	ConnectApi.EnrollmentOutputRepresentation on page 351>	List of errors that are encountered during the enrollment process.	65.0
isSuccess	Boolean	Indicates whether the enrollment process is successful (<code>true</code>) or not (<code>false</code>).	65.0
policies	ConnectApi.FamilyPolicyInfoRepresentation on page 361>	List of policies that are created as part of the enrollment process.	65.0

ConnectApi.FamilyPolicyInfoOutputRepresentation

Output representation of the created policy information for an enrollment.

Property Name	Type	Description	Available Version
plan	String	Name of the plan that's associated with the policy.	65.0
policy	String	ID of the insurance policy that's created.	65.0

Insurance Group Benefits Standard Invocable Actions

Use the standard invocable actions available with Insurance Group Benefits to effectively manage the insurance plans offered to groups, such as employees and their dependents.

Available in: Lightning Experience

Available in: **Professional**, **Enterprise**, and **Unlimited** Editions of Digital Insurance with Digital Insurance Product Admin Add On, Digital Insurance Group Benefits AddOn, Digital Insurance Group Enrollment AddOn, Digital Insurance Group Benefits Customer Community AddOn, and Digital Insurance Group Benefits Partner Community AddOn licenses

[Create Contacts and Users from Group Census Members Action](#)

Create contacts and portal users for group census or group census members.

[Create Contract From Quote Action](#)

Create a group insurance contract from a quote.

[Create Person Accounts and Users from Group Census Members Action](#)

Create person accounts and portal users for group census or group census members. This action supports bulk processing, custom field mapping, and duplicate detection.

[Get Group Census Members Action](#)

Get the group census member records from the content document in the specified asynchronous bulk request item record.

[Get Group Census Members and Plans Action](#)

Get the group census member records and associated plans from the content document in the specified asynchronous bulk request item record.

[Get Member Ratings and Contributions Action](#)

Get standard and prorated insurance rates for the specified group census members and their selected plan coverages, including contributions.

[Process Member Enrollment Action](#)

Create insurance policies for specified group census members as per their selected plans.

[Save Group Census Members Data Action](#)

Create or update the group census member records for primary members, update the IDs of the dependent members, and create or update groups census member records for the dependent members.

[Save Group Census Member Plans Data Action](#)

Validate and save group census member plans for the new members. Update the group census member plans for existing members.

[Validate Group Census Members Action](#)

Validate the specified group census member records and assign default values to the specified fields.

[Validate Group Census Members Plans Action](#)

Validate the specified group census member plan records and assign default values to the specified fields.

Create Contacts and Users from Group Census Members Action

Create contacts and portal users for group census or group census members.

This action supports bulk processing, custom field mapping, and duplicate record detection. It converts insurance group census members to Salesforce Contacts, enables member self-service through portal users, supports B2C enrollment workflows, and maintains relationship data for benefits administration and household grouping.

This action is available in API version 65.0 and later.

Special Access Rules

The org must have Digital Insurance Group Benefits enabled. To use this action, you need Census Management user access.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/createCtctUsrFrGrpCens`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
<code>contactFieldsMapping</code>	Type Apex-defined Description Map of Group Census Member Object fields to Contact fields.
<code>createPortalUsers</code>	Type boolean Description Indicates whether the portal users must be created (<code>true</code>) or not (<code>false</code>).
<code>emailEncodingKey</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Email encoding key to use to create the user records. For example, ISO-8859-1 or UTF-8.
<code>groupCensusId</code>	Type string Description Required if <code>groupCensusMembers</code> isn't specified. ID of the group census to create contacts and portal users for the associated group census members.
<code>groupCensusMembers</code>	Type object Description Required if <code>groupCensusId</code> isn't specified. Group census member records to create contacts and portal users for.
<code>localeSidKey</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Locale for the user, which affects the format of dates, times, and numbers.
<code>languageLocaleKey</code>	Type string

Input	Details
	Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Language to use to create the user records. For example, <code>fr</code> for French or <code>zh_Cw</code> for Chinese (Traditional).
<code>matchingKeysList</code>	Type string Description Comma-separated list of contact object fields to detect duplicate contact records.
<code>profileId</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. ID of the profile to create the user records for.
<code>timeZoneSidKey</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Time zone for the user, such as <code>America</code> or <code>Los_Angeles</code>.

Outputs

Output	Details
<code>createContactsOutputRep</code>	Type Apex-defined Description An apex <code>ind_ins_grpben__ContactsAndUsersCreationOutput</code> record that contains the details of the results.

Example

Sample Request

```
{
  "inputs": [
    {
      "groupCensusMembers": [
        {
          "Id": "0r6xx00000001tuAAA"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "Id": "0r6xx00000001tvAAA"
    }
  ],
  "matchingKeysList": [
    "Email",
    "LastName"
  ],
  "contactFieldsMapping": [
    {
      "key": "FirstName__c",
      "value": "FirstName"
    },
    {
      "key": "Email__c",
      "value": "Email"
    }
  ],
  "createPortalUsers": true,
  "profileId": "00exx000000n0ywAAA",
  "timeZoneSidKey": "America/New_York",
  "localeSidKey": "en_US",
  "emailEncodingKey": "UTF-8",
  "languageLocaleKey": "en_US"
}
]
}

```

Sample Response

```

{
  "outputs": [
    {
      "createContactsOutputRep": {
        "memberContactDetails": [
          {
            "groupCensusMemberId": "0r6xx00000001tuAAA",
            "contactId": "003xx000004WhsgAAC",
            "userId": "005xx000001X8vwAAC"
          }
        ],
        "errors": [
          {
            "code": "SF-0004-0058",
            "message": "Contact already exists for member",
            "additionalInfo": [
              {
                "key": "memberId",
                "value": "0r6xx00000001tvAAA"
              }
            ]
          }
        ]
      }
    }
  ],
}

```

```
{
  "isSuccess": true
}
```

Create Contract From Quote Action

Create a group insurance contract from a quote.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/createContractFromQuote

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
<code>insuranceContractCreationInputRepresentation</code>	Type Apex-defined Description Required. An Apex ConnectApi.InsuranceContractCreationInputRepresentation object that contains the details of the contract to be created.

Outputs

Output	Details
<code>requestId</code>	Type string Description ID of the asynchronous request that's created when the contract service is initiated.

Example

Sample Request

```
{
  "inputs": [
    {
      "insuranceContractCreationInputRepresentation": {
        "quoteId": "0Q0SG000000DQA00AO",
        "contractStartDate": "2025-03-11",
        "contractEndDate": "2026-03-11",
        "enrollmentStartDate": "2025-01-01",
        "enrollmentEndDate": "2025-02-01",
        "contractTerm": 1,
        "additionalFields": [
          {
            "entity": "Contract",
            "fields": {
              "data": [
                {
                  "key": "CustomerSignedTitle",
                  "value": "Mr"
                }
              ]
            }
          },
          {
            "entity": "ContractGroupPlan",
            "instanceKey": "d5861be1-9920-42d9-8f43-093b0a709d35",
            "fields": {
              "data": [
                {
                  "key": "EligibilityCriteria",
                  "value": "testcriteria"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "createContractFromQuote",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "requestId": "15Uxx0000004XYZABC"
    }
  }
]
```

```
}  
]
```

Create Person Accounts and Users from Group Census Members Action

Create person accounts and portal users for group census or group census members. This action supports bulk processing, custom field mapping, and duplicate detection.

This action supports bulk processing, custom field mapping, and duplicate record detection. It converts insurance group census members to Salesforce Person Accounts, enables member self-service through portal users, supports B2C enrollment workflows, and maintains relationship data for benefits administration and household grouping.

This action is available in API version 65.0 and later.

Special Access Rules

The org must have Person Accounts and Digital Insurance Group Benefits enabled. To use this action, you need Census Management user access.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/createAcctUsrFrGrpCens`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
accountFieldsMapping	Type Apex-defined Description Map of Group Census Member Object fields to Account fields.
createPortalUsers	Type boolean Description Indicates whether portal users must be created (<i>true</i>) or not (<i>false</i>).
emailEncodingKey	Type string

Input	Details
	Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Email encoding key to use to create the user records. For example, ISO-8859-1 or UTF-8.
<code>groupCensusId</code>	Type string Description Required if a list of <code>groupCensusMembers</code> isn't specified. ID of the group census to create person accounts and portal users for the associated group census members.
<code>groupCensusMembers</code>	Type object Description Required if <code>groupCensusId</code> isn't specified. Group census member records to create person accounts and portal users.
<code>languageLocaleKey</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Language to use to create the user records. For example, <code>fr</code> for French or <code>zh_Cw</code> for Chinese (Traditional).
<code>localeSidKey</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. Locale for the user, which affects the format of dates, times, and numbers.
<code>matchingKeysList</code>	Type string Description Comma-separated list of account object fields to detect duplicate account records.
<code>profileId</code>	Type string Description Required if <code>createPortalUsers</code> is set to <code>true</code>. ID of the profile to create the user records for.

Input	Details
recordTypeName	Type string Description API name of the record type to create the person accounts for.
timeZoneSidKey	Type string Description Required if createPortalUsers is set to true. Time zone for the user, such as America or Los_Angeles.

Outputs

Output	Details
createAccountsOutputRep	Type Apex-defined Description An Apex ind_ins_grpben__AccountsAndUsersCreationOutput record that contains the details of the results.

Example

Sample Request

```
{
  "inputs": [
    {
      "groupCensusId": "Orfxx000000006dAAA",
      "recordTypeName": "IndividualMember",
      "matchingKeysList": ["PersonEmail", "LastName", "PersonBirthdate"],
      "accountFieldsMapping": [
        { "key": "FirstName__c", "value": "FirstName" },
        { "key": "LastName__c", "value": "LastName" },
        { "key": "Email__c", "value": "PersonEmail" },
        { "key": "Birthdate__c", "value": "PersonBirthdate" }
      ],
      "createPortalUsers": true,
      "profileId": "00exx000000n0ywAAA",
      "timeZoneSidKey": "America/New_York",
      "localeSidKey": "en_US",
      "emailEncodingKey": "UTF-8",
      "languageLocaleKey": "en_US"
    }
  ]
}
```



```

    ]
  }

```

Sample Response

```

{
  "outputs": [
    {
      "AccountsAndUsersCreationOutput": {
        "groupCensusId": "0rfxx000000006dAAA",
        "memberAccountDetails": [
          {
            "groupCensusMemberId": "0r6xx00000001tqAAA",
            "accountId": "001xx000003Ga7KAAS",
            "userId": "005xx000001X8vwAAC"
          }
        ],
        "errors": [
          {
            "code": "SF-0004-0054",
            "message": "Person Account already exists for member",
            "additionalInfo": [
              {
                "key": "memberId",
                "value": "0r6xx00000001u4AAA"
              }
            ]
          }
        ]
      },
      "isSuccess": true
    }
  ]
}

```

Get Group Census Members Action

Get the group census member records from the content document in the specified asynchronous bulk request item record.

The action processes bulk member data that are uploaded through census management workflows, validates the data, identifies duplicate records, and returns structured member records for saving or further processing. It returns all transformed member records along with detailed error information for any records that fail validation.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

```
/services/data/v/actions/standard/getGroupCensusMembers
```

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
asyncBulkRequestItem	Type subject Description Required. Asynchronous bulk request item record that contains the metadata of the CSV file. This property must include the <code>AdditionalInformation</code> field with serialized JSON that contains these details. • <code>contentDocumentId</code>—ID of the content document • <code>contentDocumentVersion</code>—Version number of the content document • <code>csvColumnMapping</code>—Mapping of CSV columns to target field names • <code>duplicateDetectionFields</code>—List of fields that are used to detect duplicates • <code>groupCensusId</code>—ID of the parent group census • <code>csvHeader</code>—Header row from the CSV file • <code>startRowNumber</code>—First CSV row to process • <code>endRowNumber</code>—Last CSV row to process • <code>duplicateRowNumbersList</code>—List of row numbers identified as duplicates

Outputs

Output	Details
duplicateDetectionKeysList	Type string Description Comma-separated list of group census member fields that are used to identify unique group census member records. It's shown if duplicate detection fields are configured in the request.
errorsData	Type Apex-defined Description Errors for the records that failed validation or parsing. Each error includes the row number, error code, error message, and field-level details.

Output	Details
groupCensusId	Type string Description ID of the group census record that's associated with the group census member records that the action returned.
groupCensusMembers	Type subject Description Group census member records that the action returns. Each record contains mapped field values from CSV columns. Records with errors include an error field with the error message. Maximum number of records that are returned is 2000.

Example

POST

This sample request is for the Get Group Census Members action.

```
{
  "inputs": [
    {
      "AdditionalInformation":
      "{employeeBulkRequestInfo:{groupCensusId:0rfSG0000000vclYAA,contentDocumentName:Archit50Records
      With groupClass - Archit50Records With groupClass
      0),contentDocumentId:0rfSG0000000vclYAA,duplicateDetectionFields:{Age,FirstName,LastName,AssociationWithPrimaryMor,Group
      Name:FirstName,Last
      Name:LastName,Gender:Gender,Birthdate:Birthdate,SmokerStatus:SmokerStatus,Relationship:AssociationWithPrimaryMor,Group
      Class:GroupClassName),executeRules:false},startRowIndex:1,endRowIndex:27,startByteOffset:83,endByteOffset:1717,csvHeader:First
      Name,Last Name,Gender,Birthdate,Relationship,SmokerStatus,Group
      Class,Errors,duplicateRowNumbersList:[],shouldExecuteConfigurationRule:false}",
      "Id": "18cSG0000003DgfyAE"
    }
  ]
}
```

This sample response is for the Get Group Census Members action.

```
[
  {
    "actionName": "getGroupCensusMembers",
    "errors": null,
    "isSuccess": true,
    "outputValues": {
      "duplicateDetectionKeysList": [],
      "groupCensusId": "0rfSG0000000zyY9YAI",
      "groupCensusMembers": [
        {
```

```
      "Email": "ross.farley_0001@insgroup.co.in",
      "AssociationWithPrimaryMbr": "Self",
      "FirstName": "Ross",
      "LastName": "Farley",
      "Country": "Health Comprehensive|Chronic Disease",
      "Gender": "Male",
      "Birthdate": "08/07/1981"
    },
    {
      "Email": "teagan.wallace_0001@gmail.com",
      "AssociationWithPrimaryMbr": "Spouse",
      "FirstName": "Teagan",
      "LastName": "Wallace",
      "Country": "Health Comprehensive|Critical Illness & Surgery",
      "Gender": "Female",
      "Birthdate": "08/07/1986"
    }
  ]
}
```

Get Group Census Members and Plans Action

Get the group census member records and associated plans from the content document in the specified asynchronous bulk request item record.

The action processes bulk member data that are uploaded through census management workflows. This action validates the data, identifies duplicates, and returns structured member records for further processing. This action returns all transformed records, along with detailed error information for any records that fail validation.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getGroupCensusMembersAndPlans

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
asyncBulkRequestItem	Type subject

Input**Details****Description**

Required.

Asynchronous bulk request item record that contains the metadata about the CSV upload.

This property must have the `AdditionalInformation` field with serialized JSON that contains these details.

- `contentDocumentId`—ID of content document
- `contentDocumentVersion`—Version of content document
- `csvColumnMapping`—Map of CSV column to field name
- `duplicateDetectionFields`—List of fields for duplicate detection
- `groupCensusId`—Parent Group Census ID
- `csvHeader`—CSV header row
- `startRowNumber`—Starting row to process
- `endRowNumber`—Ending row to process
- `duplicateRowNumbersList`—Pre-identified duplicate rows

Outputs**Output****Details**

`duplicateDetectionKeysList`

Type

string

Description

Comma-separated list of group census member fields that can be used to identify unique group census member records.

This property is returned if duplicate detection fields are configured in the request.

`errorsData`

Type

Apex-defined

Description

Error records that occurred during the retrieval of group census member and plan records.

`groupCensusId`

Type

string

Description

Salesforce ID of the GroupCensus record which these members belong to. This property is used to associate members with their parent census.

Output	Details
groupCensusMembers	Type subject Description Collection of transformed GroupCensusMember SObject records that are parsed from the CSV file. Each record includes field values that are mapped from the corresponding CSV columns. Records with errors include an error field containing the error message. The maximum limit size is 2000.
groupCensusMemberPlans	Type subject Collection of plan records for group census members.
shouldExecuteConfigurationRule	Type boolean Description Indicates whether the product configuration rules must be executed for member plans in the enrollment census (<code>true</code>) or not (<code>false</code>).

Example

Sample Request

```
{
  "AdditionalInformation": {
    "employeeBulkRequestInfo": {
      "groupCensusId": "0rfSG0000000vc1YAA",
      "contentDocumentName": "Archit50Records With groupClass - Archit50Records With
groupClass (1)",
      "contentDocumentId": "069SG0000041wPdYAI",
      "contentDocumentVersion": "068SG000004PIp3YAG",
      "duplicateDetectionFields": [
        "Gender",
        "FirstName",
        "Lastname",
        "AssociationWithPrimaryMbr",
        "GroupCensusId"
      ],
      "batchSize": null,
      "csvColumnMapping": {
        "First Name": "FirstName",
        "Last Name": "Lastname",
        "Gender": "Gender",
        "Birthdate": "Birthdate",
        "SmokerStatus": "SmokerStatus",
        "Relationship": "AssociationWithPrimaryMbr",
        "Group Class": "GroupClassName"
      }
    },
  },
}
```

```

        "executeRules": false
    },
    "startRowNumber": 1,
    "endRowNumber": 27,
    "startByteOffset": 83,
    "endByteOffset": 1717,
    "csvHeader": "First Name,Last Name,Gender,Birthdate,Relationship,SmokerStatus,Group
Class,Errors",
    "duplicateRowNumbersList": [],
    "shouldExecuteConfigurationRule": false
},
    "Id": "18cSG0000003DgfyYAE"
}

```

Sample Response

```

[
  {
    "actionName": "getGroupCensusMembersAndPlans",
    "errors": null,
    "isSuccess": true,
    "outputValues": [
      {
        "GroupClassName": "Test_MGR",
        "AssociationWithPrimaryMbr": "Self",
        "FirstName": "Employee1",
        "Lastname": "Salesforce1",
        "Gender": "Male",
        "Birthdate": "1987-12-04",
        "SmokerStatus": "Tobacco"
      },
      {
        "AssociationWithPrimaryMbr": "Spouse",
        "FirstName": "Employee2",
        "Lastname": "Salesforce2",
        "Gender": "Female",
        "Birthdate": "1982-11-30",
        "SmokerStatus": "Tobacco"
      },
      {
        "AssociationWithPrimaryMbr": "Child",
        "FirstName": "Employee3",
        "Lastname": "Salesforce3",
        "Gender": "Male",
        "Birthdate": "1988-11-18",
        "SmokerStatus": "NonTobacco"
      }
    ]
  }
]

```

Get Member Ratings and Contributions Action

Get standard and prorated insurance rates for the specified group census members and their selected plan coverages, including contributions.

This action retrieves member-specific coverage selections, validates them against the root plan hierarchy, and invokes the rating service to calculate premiums and contributions. The action returns pricing results with rates at member, plan, and coverage levels, along with hierarchical error information. The action is currently used in enrollment flows.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getMbrsRatingAndContribution

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
groupCensusMembers	Type subject Description Required. Collection of group census member records for whom rates must be calculated. This action retrieves the member IDs from these records. The maximum list size is 2000.
contractGroupPlanId	Type string Description Required. ID of the root contract group plan for which the rates and contributions must be calculated. This ID represents the primary plan under which coverages are organized. The action validates that selected coverages that belong to this plan hierarchy.
effectiveDate	Type date

Input	Details
	Description Effective date for rating calculations in yyyy-MM-dd format. If specified, the rating service uses this date to lookup the active pricing procedure and perform proration calculations. If not specified, contract start date is used as default.

Outputs

Output	Details
memberRatingOutputRep	Type Apex-defined Description An Apex ind_ins_grpben__MembersPlansRatingOutput record that contains pricing results for the requested members, plans, and coverages. The Apex class includes standard rates, prorated rates, contributions, and hierarchical error information at the request, member, plan, and coverage levels.

Example

Sample Request

```
{
  "inputs": [
    {
      "groupCensusMembers": [
        {
          "Id": "0r6SG0000008OFjYAM",
          "Name": "John Doe",
          "FirstName": "John",
          "LastName": "Doe",
          "Email": "john.doe@example.com",
          "BirthDate": "1985-03-15"
        },
        {
          "Id": "0r6SG0000008OFjYAN",
          "Name": "Jane Smith",
          "FirstName": "Jane",
          "LastName": "Smith",
          "Email": "jane.smith@example.com",
          "BirthDate": "1990-07-22"
        }
      ],
      "contractGroupPlanId": "0rgSG0000000ef1YAA",
      "effectiveDate": "2025-10-15"
    }
  ]
}
```

```
]
}
```

Sample Response

```
[
  {
    "actionName": "getMbrsRatingAndContributions",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "memberRatingOutputRep": {
        "pricingResults": [
          {
            "plans": [
              {
                "rates": [
                  {
                    "value": 60,
                    "key": "NetUnitPrice"
                  }
                ],
                "proratedRates": [
                  {
                    "value": 60,
                    "key": "NetUnitPrice"
                  }
                ],
                "planId": "0rgSG0000000eezYAA",
                "errors": [],
                "coverages": [
                  {
                    "rates": [
                      {
                        "value": 20,
                        "key": "NetUnitPrice"
                      }
                    ],
                    "proratedRates": [
                      {
                        "value": 20,
                        "key": "NetUnitPrice"
                      }
                    ],
                    "errors": [],
                    "coverageId": "0rgSG0000000ef1YAA",
                    "contributions": {
                      "groupClassContributionId": "0rFSG0000000Beb2AE",
                      "errors": [],
                      "employerPremiumAmount": 2,
                      "employeePremiumAmount": 18
                    }
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  }
]
```

```

        ],
        "contributions": {
            "groupClassContributionId": null,
            "errors": [],
            "employerPremiumAmount": 2,
            "employeePremiumAmount": 58
        }
    },
    ],
    "memberId": "0r6SG0000008R33YAE",
    "errors": []
}
],
"errors": []
}
},
"sortOrder": -1,
"version": 1
}
]

```

Process Member Enrollment Action

Create insurance policies for specified group census members as per their selected plans.

The action retrieves member-specific coverage selections, validates them against the plan selection and checks for duplicate policies if specified in action parameter, and generates insurance policies, coverages, participants and related objects. The action returns list of insurance policy IDs that are generated or error if they occur during the processing.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/processMemberEnrollment

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer ***token***

Inputs

Input	Details
contractGroupPlanId	Type string

Input	Details
	Description Required ID of the Contract Group Plan that's used to retrieve associated product and coverage details.
<code>createRelatedObjectRecords</code>	Type boolean Description Indicates whether to create related records for each insurance policy participant (<code>true</code>) or not (<code>false</code>). If set to <code>true</code> , two insurance policy participant related object records are created per participant. The first record maps the participant to the associated Group Census Member, and the second record links the participant to either a Person Account or a Contact, based on their participant type.
<code>duplicateCheckRequired</code>	Type boolean Description Indicates whether to check for duplicate policies (<code>true</code>) or not (<code>false</code>). The default value is <code>true</code> . If set to <code>true</code> and a policy already exists for the same member and product category, this invocable action shows an error.
<code>groupCensusMembers</code>	Type subject Description Collection of group census member records that represent the members that must be enrolled.
<code>pricingResult</code>	Type Apex-defined Description Collection of Apex <code>ConnectApi.MembersPlansRatingOutput</code> records that contain pricing results for the requested members.

Outputs

Output	Details
<code>insurancePolicyIds</code>	Type string

Output**Details****Description**

IDs of the insurance policy records that are created.

Example

Sample Request

```
{
  "inputs": [
    {
      "groupCensusMembers": [
        {
          "Id": "0r6SG00000093MrYAI",
          "Name": "John Doe",
          "FirstName": "John",
          "LastName": "Doe",
          "Email": "john.doe@example.com",
          "BirthDate": "1998-03-15"
        }
      ],
      "contractGroupPlanId": "0rgSG0000000iaXYAQ",
      "pricingResult": {
        "pricingResults": [
          {
            "plans": [
              {
                "rates": [
                  {
                    "value": 60,
                    "key": "NetUnitPrice"
                  },
                  {
                    "value": 0,
                    "key": "StandardQLITaxAmount"
                  }
                ],
                "proratedRates": [
                  {
                    "value": 60,
                    "key": "NetUnitPrice"
                  },
                  {
                    "value": 0,
                    "key": "StandardQLITaxAmount"
                  }
                ]
              },
              {
                "planId": "0rgSG0000000iaXYAQ",
                "errors": [],
                "coverages": [
                  {
                    "rates": [
```

```

    {
      "value": 0,
      "key": "StandardQLITaxAmount"
    },
    {
      "value": 20,
      "key": "NetUnitPrice"
    }
  ],
  "proratedRates": [
    {
      "value": 0,
      "key": "StandardQLITaxAmount"
    },
    {
      "value": 20,
      "key": "NetUnitPrice"
    }
  ],
  "errors": [],
  "coverageId": "OrgSG0000000iaZYAQ",
  "contributions": {
    "groupClassContributionId": "OrFSG0000000Beb2AE",
    "errors": [],
    "employerPremiumAmount": 2,
    "employeePremiumAmount": 18
  }
},
{
  "rates": [
    {
      "value": 0,
      "key": "StandardQLITaxAmount"
    },
    {
      "value": 40,
      "key": "NetUnitPrice"
    }
  ],
  "proratedRates": [
    {
      "value": 0,
      "key": "StandardQLITaxAmount"
    },
    {
      "value": 40,
      "key": "NetUnitPrice"
    }
  ],
  "errors": [],
  "coverageId": "OrgSG0000000iaaYAA",
  "contributions": {
    "groupClassContributionId": null,
    "errors": [],

```

```

        "employerPremiumAmount": 0,
        "employeePremiumAmount": 40
      }
    ],
    "contributions": {
      "groupClassContributionId": null,
      "errors": [],
      "employerPremiumAmount": 2,
      "employeePremiumAmount": 58
    }
  ],
  "memberId": "0r6SG00000093MrYAI",
  "errors": []
}
]
}

```

Sample Response

```

[
  {
    "actionName": "processMemberEnrollment",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "insurancePolicyIds": ["0YTSG000000QzO14AK"]
    },
    "sortOrder": -1,
    "version": 1
  }
]

```

Save Group Census Members Data Action

Create or update the group census member records for primary members, update the IDs of the dependent members, and create or update groups census member records for the dependent members.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/saveGroupCensusMembersData

Formats

JSON

HTTP Methods

POST

AuthenticationAuthorization: Bearer *token***Inputs**

Input	Details
asyncBulkRequestId	Type string Description ID of the async bulk request item to use as reference for creating insurance async bulk record details records.
groupCensusId	Type string Description ID of the parent group census record that's associated with the group census member records.
groupCensusMembers	Type subject Description Collection of group census member records to be save. Maximum limit size is 2000.
operation	Type string Description Required if <code>asyncBulkRequestId</code> isn't specified. Operation to perform on the group census member records. Valid values are Add and Update.
shouldLogErrors	Type boolean Description Indicates whether errors that occur while saving group census member plans must be recorded in a content document (<code>true</code>) or not (<code>false</code>). The default values is <code>false</code> .

Outputs

Output	Details
errorsData	Type Apex-defined Description Collection of error records for synchronous Add Members and Update Members services. Maximum limit is 2000.
savedGroupCensusMembers	Type subject Description Collection of group census member records that are created or updated during the execution of the action. It contains both primary and dependent members that are successfully processed. Maximum limit is 2000.

Example

Sample Request

```
{
  "inputs": [
    {
      "asyncBulkRequestId": "15Uxx0000004CfoEAE",
      "groupCensusId": "0rfxx0000000005AAA",
      "groupCensusMembers": [
        {
          "Name": "IronMan",
          "FirstName": "Iron",
          "LastName": "Man",
          "Gender": "Male",
          "GroupClassId": "0rExx0000000001EAA",
          "GroupClassName": "Executive",
          "GroupCensusId": "0rfxx0000000005AAA",
          "Email": "test1@gtest.com",
          "AssociationWithPrimaryMbr": "Self",
          "MemberKey": "c01c21aecc57e259f2faeb9484149c8e"
        }
      ]
    }
  ]
}
```

Sample Response

```
{
  "actionName": "saveGroupCensusMembersData",
  "savedGroupCensusMembers": [
```

```
{
  {
    "attributes": {
      "type": "GroupCensusMember"
    },
    "Email": "test2@gtest.com",
    "AssociationWithPrimaryMbr": "Spouse",
    "FirstName": "Pepper",
    "MemberKey": "ad8b415683d7e6b6cc8d5a975c8cf75b",
    "Lastname": "Potts",
    "Gender": "Female",
    "GroupCensusId": "0rfxx0000000005AAA",
    "Name": "PepperPotts"
  }
}
```

Save Group Census Member Plans Data Action

Validate and save group census member plans for the new members. Update the group census member plans for existing members. This action processes member enrollment information, validates plan assignments, and handles primary and dependent member relationships. It manages plan updates, including plan additions and deletions, and supports bulk processing of member plans with comprehensive error handling and logging. This action is available in API version 66.0 and later.

Special Access Rules

To use this invocable action, you must have Digital Insurance Group Benefits Census Management user access

Supported REST HTTP Methods

- URI**
/services/data/v/actions/standard/saveGrpCensusMbrPlansData
- Formats**
JSON, XML
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer *token*

Inputs

Input	Details
asyncBulkRequestId	Type string

Input	Details
	Description Required. ID of the asynchronous bulk request item to use for tracking and error logging.
groupCensusId	Type string Description Required. ID of the group census record.
groupCensusMemberPlans	Type subject Description Required. List of group census member plan records to save.
groupCensusMembers	Type subject Description Required. List of group census member records to process.
shouldLogErrors	Type boolean Description Required. Indicates whether errors that occur while saving group census member plans are recorded in a content document (<code>true</code>) or not (<code>false</code>).

Outputs

Output	Details
errorsData	Type Apex-defined Description Error details for the failed records.

Output**Details**

groupCensusMemberPlans

Type

subject

Description

Details of group census member plan records that are successfully saved.

Example

POST

This sample request is for the Save Group Census Member Plans Data action.

```
{
  "inputs": [
    {
      "groupCensusMembers": [
        {
          "Id": "a1B0000000000001",
          "MemberRefId": "M001",
          "PrimaryGroupCensusMemberId": null,
          "GroupCensusId": "a3B0000000000001",
          "MemberKey": "EMP001",
          "AssociationWithPrimaryMbr": "Self"
        },
        {
          "Id": "a1B0000000000002",
          "MemberRefId": "M002",
          "PrimaryGroupCensusMemberId": "a1B0000000000001",
          "GroupCensusId": "a3B0000000000001",
          "MemberKey": "DEP001",
          "AssociationWithPrimaryMbr": "Spouse"
        }
      ],
      "groupCensusMembersPlans": [
        {
          "ContractGroupPlanId": "p1B0000000000001",
          "PlanName": "Health Plan A",
          "PlanType": "Medical",
          "GroupCensusMember": "a1B0000000000001",
          "MemberRefId": "M001"
        },
        {
          "ContractGroupPlanId": "p1B0000000000002",
          "PlanName": "Dental Plan B",
          "PlanType": "Dental",
          "GroupCensusMember": "a1B0000000000001",
          "MemberRefId": "M001"
        },
        {
          "ContractGroupPlanId": "p1B0000000000001",
          "PlanName": "Health Plan A",
```

```

        "PlanType": "Medical",
        "GroupCensusMember": "a1B0000000000002",
        "MemberRefId": "M002"
    }
],
"shouldLogErrors": true,
"asyncBulkRequestItemId": "a2B0000000000001",
"groupCensusId": "a3B0000000000001"
}
]
}

```

This sample response is for the Save Group Census Member Plans Data action.

```

{
  "outputs": [
    {
      "savedGroupCensusMembersPlans": [
        {
          "Id": "a4B0000000000001",
          "Name": "Plan-001",
          "GroupCensusMemberId": "a1B0000000000001",
          "ContractGroupPlanId": "p1B0000000000001",
          "CreatedDate": "2025-01-27T10:30:00.000Z",
          "CreatedById": "0050000000000001"
        },
        {
          "Id": "a4B0000000000002",
          "Name": "Plan-002",
          "GroupCensusMemberId": "a1B0000000000001",
          "ContractGroupPlanId": "p1B0000000000002",
          "CreatedDate": "2025-01-27T10:30:00.000Z",
          "CreatedById": "0050000000000001"
        },
        {
          "Id": "a4B0000000000003",
          "Name": "Plan-003",
          "GroupCensusMemberId": "a1B0000000000002",
          "ContractGroupPlanId": "p1B0000000000001",
          "CreatedDate": "2025-01-27T10:30:00.000Z",
          "CreatedById": "0050000000000001"
        }
      ],
      "errors": [
        {
          "errorMessage": "Invalid plan assignment for member M003",
          "recordId": "M003",
          "errorType": "VALIDATION_ERROR",
          "fieldName": "ContractGroupPlanId"
        }
      ],
      "isSuccess": true
    }
  ]
}

```

Validate Group Census Members Action

Validate the specified group census member records and assign default values to the specified fields.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/validateGroupCensusMembers

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
duplicateDetectionKeysList	Type string Description Comma-separated list of field names that are used to identify the unique group census member records. For example, <code>FirstName</code> , <code>LastName</code> , <code>GroupCensusId</code> , and <code>AssociationWithPrimaryMbr</code> .
groupCensusId	Type string Description ID of the parent group census record that's associated with the group census member records.
groupCensusMembers	Type subject Description Collection of group census member records that must be validated. Maximum list size is 2000.

Outputs

Output	Details
errorsData	Type Apex-defined Description Details of the validation errors that occurred while validating the group census member records. Maximum limit is 2000.
validatedGroupCensusMembers	Type subject Description Collection of group census member records that are validated and assigned with default values. Records with validation errors aren't shown. Maximum limit is 2000.

Example

Sample Request

```
{
  "inputs": [
    {
      "groupCensusId": "0rfDR0000000Cn6YAE",
      "groupCensusMembers": [
        {
          "Name": "RossFarley",
          "FirstName": "Ross",
          "LastName": "Farley",
          "Gender": "Male",
          "GroupClassName": "Manager",
          "Email": "ross.farley_0001@insgroup.co.in",
          "AssociationWithPrimaryMbr": "Self"
        },
        {
          "Name": "Fakeem Huber",
          "FirstName": "Fakeem",
          "LastName": "Huber",
          "Gender": "Male",
          "GroupClassName": "Associate",
          "Email": "fakeem.huber_0432@insgroup.co.in",
          "AssociationWithPrimaryMbr": "Self"
        }
      ]
    }
  ]
}
```

Sample Response

```
{
  "validatedGroupCensusMembers": [
    {
      "attributes": {
        "type": "GroupCensusMember"
      },
      "Email": "ross.farley_0001@insgroup.co.in",
      "AssociationWithPrimaryMbr": "Self",
      "FirstName": "Ross",
      "MemberKey": "a66e329bd537b9b0be9d04f326eed94",
      "RelationshipToPrimaryMember": "Self",
      "LastName": "Farley",
      "Gender": "Male",
      "GroupClassId": "0rEDR000000085k2AA",
      "GroupCensusId": "0rfDR0000000Cn6YAE",
      "Name": "RossFarley"
    },
    {
      "attributes": {
        "type": "GroupCensusMember"
      },
      "Email": "fakeem.huber_0432@insgroup.co.in",
      "AssociationWithPrimaryMbr": "Self",
      "FirstName": "Fakeem",
      "MemberKey": "fa26049af8a10b70bd68ce9a6912425f",
      "RelationshipToPrimaryMember": "Self",
      "LastName": "Huber",
      "Gender": "Male",
      "GroupClassId": "0rEDR000000085p2AA",
      "GroupCensusId": "0rfDR0000000Cn6YAE",
      "Name": "Fakeem Huber"
    }
  ]
}
```

Validate Group Census Members Plans Action

Validate the specified group census member plan records and assign default values to the specified fields.

This action is available in API version 66.0 and later.

Special Access Rules

To use this invocable action, you must have Digital Insurance Group Benefits Census Management user access and the org must be enabled with the Digital Insurance Group Benefits feature.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/validateGrpCensusMembersPlans

Formats

JSON, XML

HTTP Methods

POST

AuthenticationAuthorization: Bearer *token*

Inputs

You must specify one of these input parameters for the validation process.

- contextId
- groupCensusMembers
- memberPlansToDelete
- memberPlansToValidate

Input	Details
contextId	Type string Description ID of the context definition that contains information about root plans and coverages that are configured for the current list of group census members.
groupCensusMembers	Type subject Description Group census member records that must be validated.
memberPlansToDelete	Type subject Description Group census member plan records that must be deleted.
memberPlansToValidate	Type subject Description Group census member plan records that must be validated.
shouldExecuteConfigurationRule	Type boolean

Input	Details
	Description Indicates whether the product configuration rules must be executed for member plans in the enrollment census (<code>true</code>) or not (<code>false</code>).

Outputs

Output	Details
<code>errorsData</code>	Type Apex-defined Description Validation errors that occurred while validating the group census member and plans records.
<code>validatedGroupCensusMemberPlans</code>	Type subject Description Group census member plan records that are validated and assigned with default values.
<code>validatedGroupCensusMembers</code>	Type subject Description Group census member records that are validated and assigned with default values.

Example

POST

This sample request is for the Validate Group Census Members Plans action.

```
{
  "inputs": [
    {
      "groupCensusMembers": [
        {
          "Id": "a1b2c3d4e5f6g7h8",
          "MemberRefId": "M001",
          "FirstName": "John",
          "LastName": "Doe",
          "GroupCensus": "gc_12345"
        },
        {
          "Id": "b2c3d4e5f6g7h8i9",
          "MemberRefId": "M002",
          "FirstName": "Jane",

```

```

        "LastName": "Smith",
        "GroupCensus": "gc_12345"
      }
    ],
    "memberPlansToValidate": [
      {
        "ContractGroupPlan": "plan_medical_001",
        "GroupCensusMember": "a1b2c3d4e5f6g7h8",
        "MemberRefId": "M001"
      },
      {
        "ContractGroupPlan": "plan_dental_001",
        "GroupCensusMember": "a1b2c3d4e5f6g7h8",
        "MemberRefId": "M001"
      }
    ],
    "memberPlansToDelete": [
      {
        "ContractGroupPlan": "plan_old_001",
        "GroupCensusMember": "b2c3d4e5f6g7h8i9",
        "MemberRefId": "M002"
      }
    ],
    "contextId": "ctx_validation_12345",
    "shouldExecuteConfigurationRule": true
  }
]
}

```

This sample response is for the Validate Group Census Members Plans action.

```

{
  "outputs": [
    {
      "validatedGroupCensusMembers": [
        {
          "Id": "a1b2c3d4e5f6g7h8",
          "MemberRefId": "M001",
          "FirstName": "John",
          "LastName": "Doe",
          "Information": "Plans validated successfully",
          "CensusMgmtUploadMemError": null
        },
        {
          "Id": "b2c3d4e5f6g7h8i9",
          "MemberRefId": "M002",
          "FirstName": "Jane",
          "LastName": "Smith",
          "Information": "Plan deletion validated",
          "CensusMgmtUploadMemError": null
        }
      ],
      "validatedGroupCensusMemberPlans": [
        {
          "ContractGroupPlan": "plan_medical_001",

```

```

        "GroupCensusMember": "a1b2c3d4e5f6g7h8",
        "MemberRefId": "M001",
        "PlanType": "AUTO_ADDED"
    },
    {
        "ContractGroupPlan": "plan_dental_001",
        "GroupCensusMember": "a1b2c3d4e5f6g7h8",
        "MemberRefId": "M001",
        "PlanType": "AUTO_ADDED"
    },
    {
        "ContractGroupPlan": "plan_old_001",
        "GroupCensusMember": "b2c3d4e5f6g7h8i9",
        "MemberRefId": "M002",
        "CensusMgmtUploadMemError": "Plan successfully marked for deletion"
    }
],
"errorsData": []
}
]
}

```

Insurance Group Benefits Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Group Benefits](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Group Benefits

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Group Benefits exposes additional `actionType` values for the FlowActionCall Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	InvokableActionType (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Group Benefits include: <code>createCtctUsrFrGrpCens</code>—Create contacts and portal users for group census or group census members.

Field Name	Field Type	Description
		• <code>createContractFromQuote</code>—Create a group insurance contract from a quote. • <code>createAcctUsrFrGrpCens</code>—Create person accounts and portal users for group census or group census members. • <code>getGroupCensusMembersAndPlans</code>—Get the group census member records and associated plans from the content document in the specified asynchronous bulk request item record. • <code>getMbrsRatingAndContribution</code>—Get standard and prorated insurance rates for the specified group census members and their selected plan coverages, including contributions. • <code>processMemberEnrollment</code>—Process enrolled group census member and their selected plan to create insurance policies and related objects. • <code>saveGroupCensusMembersData</code>—Create or update the group census member records for primary members, update the IDs of the dependent members, and create or update groups census member records for the dependent members. • <code>validateGroupCensusMembers</code>—Validate the specified group census member records and assign default values to the specified fields.

CHAPTER 5 Insurance Policy Administration

In this chapter ...

- [Insurance Policy Business APIs](#)
- [Insurance Policy Standard Invocable Actions](#)
- [Insurance Policy Administration Metadata API Types](#)
- [Insurance Policy Standard Objects](#)
- [Insurance Policy Administration Apex Reference](#)
- [Insurance Policy Developer Journey](#)

Use the Insurance Policy developer resources to manage insurance policies comprehensively, for different transaction types, such as issuance, endorsement, renewal, and cancellation. Each transaction type creates an InsurancePolicyTransaction record.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Insurance Policy Business APIs

Manage policy transactions such as issuance, endorsement, renewal, and cancellation. Calculate term taxes based on transaction types and policy durations.

This table lists the available Insurance Policy Administration resources.

Resource	Description
/connect/insurance/policies/policyId/cancel (POST)	Cancel an insurance policy version. The API takes the effective date of cancellation from the input JSON.
/connect/insurance/policies/policyId/endorse (POST)	Endorse the latest version of the insurance policy based on the context ID and effective date.
/connect/insurance/policies/policyId?_qs=contextOnly=true (GET)	Get the ID of the context and the insurance policy details. This includes related assets, participants, and coverages. Use query parameters to request additional information.
/connect/insurance/policies (POST)	Use the context ID provided in the input JSON to generate a new insurance policy.
/connect/insurance/policies/policyId/renew (POST)	Use the context ID of an existing policy to renew a policy. Modify the renewal date and also add optional fields and additional information in the input JSON.
/connect/insurance/policies/<i>policyId</i>/out-of-sequence-endorsement (POST)	Create an out-of-sequence endorsement for an insurance policy.
/connect/insurance/policies/<i>policyId</i>/reinstate (POST)	Reinstate a previously canceled insurance policy.
/connect/insurance/policies/multi-root (POST)	Asynchronously create a multi-root insurance policy along with its child policies. A multi-root policy includes multiple child policies of different product types under a single parent policy.
/connect/insurance/policies/multi-root/endorse (POST)	Asynchronously endorse a multi-root insurance policy along with its child policies.

Resource	Description
/connect/insurance/policies/multi-root/renew (POST)	Asynchronously renew a multi-root insurance policy, create renewed records for the parent policy and the associated child policies.

[Resources](#)

Learn more about the available Insurance Policy API resources.

[Requests](#)

Learn more about the available Insurance Policy requests.

[Responses](#)

Learn more about the available response bodies of Insurance Policy APIs.

Resources

Learn more about the available Insurance Policy API resources.

[Cancel Insurance Policy \(POST\)](#)

Cancel an insurance policy version. The API takes the effective date of cancellation from the input JSON. You can advance the cancellation date within the policy term, but you can't postpone the cancellation.

[Endorse Insurance Policy \(POST\)](#)

Endorse the latest version of the insurance policy based on the context ID and effective date. You can also include other optional fields and additional details in the input JSON.

[Insurance Policy Details \(GET\)](#)

Get the context ID and the insurance policy details. This includes related assets, participants, and coverages. Use query parameters to request additional information.

[Issue Insurance Policy \(POST\)](#)

Use the context ID provided in the input JSON to generate a new insurance policy.

[Out-of-Sequence Endorse Policy \(POST\)](#)

Create an out-of-sequence endorsement for an insurance policy.

[Reinstate Insurance Policy \(POST\)](#)

Reinstate a previously canceled insurance policy.

[Renew Insurance Policy \(POST\)](#)

Use the context ID of an existing policy to renew a policy. Modify the renewal date and also add optional fields and additional information in the input JSON.

[Multi-Root Policy Issuance \(POST\)](#)

Asynchronously create a multi-root insurance policy along with its child policies. A multi-root policy includes multiple child policies of different product types under a single parent policy.

[Multi-Root Policy Endorsement \(POST\)](#)

Asynchronously endorse a multi-root insurance policy along with its child policies.

[Multi-Root Policy Renewal \(POST\)](#)

Asynchronously renew a multi-root insurance policy, create renewed records for the parent policy and the associated child policies.

Cancel Insurance Policy (POST)

Cancel an insurance policy version. The API takes the effective date of cancellation from the input JSON. You can advance the cancellation date within the policy term, but you can't postpone the cancellation.

Resource

```
/connect/insurance/policies/policyId/cancel
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx000000006TCAQ/cancel
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "effectiveDate": "2024-09-08",
  "transactionRecord":{
    "name" : "Cancel Transaction"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the cancellation of the policy goes into effect.	Required	63.0
transactionRecord	Insurance Policy Transaction Input	The JSON structure that overrides the transaction information.	Optional	63.0

Response body for POST

[Cancel Policy](#)

Endorse Insurance Policy (POST)

Endorse the latest version of the insurance policy based on the context ID and effective date. You can also include other optional fields and additional details in the input JSON.

Resource

```
/connect/insurance/policies/policyId/endorse
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTxx00000000WHGAY/endorse
```

Available version

63.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "contextId": "54920265842e1b61496c38b7e2202f6f0aff8123ad2f19fe17ac68f7fc1c60b3",
  "effectiveDate": "2024-01-01",
  "insurancePolicy": {
    "policyName": "p2_endorse_name",
    "policyNumber": "p2_endorse_number"
  },
  "transactionRecord": {
    "name": "New Endorsement"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoRoot",
          "fields": {
            "Checkbox__c": "true",
            "Currency__c": 123456,
            "Date__c": "2024-11-30",
            "DateTime__c": "2024-12-31T01:34:00.000Z",
            "Email__c": "testuser@sf.com"
          }
        }
      },
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoDriver",
          "fields": {
            "TextEncr__c": "233345675676",
            "Url__c": "https://thisistest.com",
            "Time__c": "12:05:00.000Z"
          }
        }
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> , <code>InsurancePolicyCoverage</code> , and <code>InsurancePolicyParticipant</code> objects. The values for <code>additionalFields</code> take precedence over the context or the <code>insurancePolicy</code> JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Context ID of the quote that's used for endorsement.	Required	63.0
effectiveDate	String	Date from when the endorsed policy version is effective.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that overrides the name and policy number of the endorsed policy version.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that overrides the transaction information.	Optional	63.0

Response body for POST[Policy](#)**Insurance Policy Details (GET)**

Get the context ID and the insurance policy details. This includes related assets, participants, and coverages. Use query parameters to request additional information.

Resource

```
/connect/insurance/policies/policyId?_qs=contextOnly=true

/connect/insurance/policies/policyId?_qs=expand[attributes]=true

/connect/insurance/policies/policyId?_qs=expand[attributes]=true&expand[surcharges]=true

/connect/insurance/policies/policyId?_qs=expand[additionalFields] [fieldSet] [$in]={fieldSetIds}

/connect/insurance/policies/policyId?_qs=expand[additionalFields] [fieldSet] [$eq]={fieldSetId}

/connect/insurance/policies/policyId?_qs=expand[additionalFields] [fieldSet] [$eq]={fieldSetId};contextOnly=true

/connect/insurance/policies/policyId?_qs=expand[additionalFields] [fieldSet] [$eq]={fieldSetId};expand[attributes]=true

/connect/insurance/policies/policyId?_qs=expand[additionalFields] [fieldSet] [$eq]={fieldSetId};expand[attributes]=true&expand[surcharges]=true
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx0000000001GAA

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTS0000000233R4AQ?_qs=expand[attributes]=true

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx0000000001GAA?_qs=expand[attributes]=true&expand[surcharges]=true

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTS0000000233R4AQ?_qs=contextOnly=true

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTS0000000233R4AQ?_qs=expand[additionalFields] [fieldSet] [$in]=0XSG0000000003PAG,0XSG0000000003PAG

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTS0000000233R4AQ?_qs=expand[additionalFields] [fieldSet] [$in]=0XSG0000000003PAG,0XSG0000000003PAG;contextOnly=true

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTS0000000233R4AQ?_qs=expand[additionalFields] [fieldSet] [$in]=0XSG0000000003PAG,0XSG0000000003PAG;expand[attributes]=true

https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx0000000001GAA?_qs=expand[surcharges]=true
```

Available version

63.0

HTTP methods

GET

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
connectDynamicParameters	Map<String,String>	Map of connect dynamic parameters.	Optional	63.0

Response body for GET

[Get Policy](#)

Issue Insurance Policy (POST)

Use the context ID provided in the input JSON to generate a new insurance policy.

Resource

```
/connect/insurance/policies
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "contextId": "3561319f267fd53380bc38c049a01c8089876e833a321860766d35a9c40b67df",
  "insurancePolicy": {
    "policyName": "Mr. Bret Lees Policy",
    "policyNumber": "Policy-Bret-OneYear",
    "effectiveFromDate": "2024-01-01",
    "effectiveToDate": "2024-10-31"
  },
  "transactionRecord": {
    "name": "New Policy"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoRoot",
          "fields": {
            "Checkbox__c": "true",
            "Currency__c": 123456,
            "Date__c": "2024-11-30",
            "DateTime__c": "2024-12-31T01:34:00.000Z",
            "Email__c": "testuser@sf.com"
          }
        }
      }
    ]
  }
}

{
  "additionalFieldsMapValue": {
    "instanceKey": "autoDriver",
    "fields": {
      "TextEncr__c": "233345675676",
      "Url__c": "https://thisistest.com",
      "Time__c": "12:05:00.000Z"
    }
  }
}
```

```

    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> , <code>InsurancePolicyCoverage</code> , and <code>InsurancePolicyParticipant</code> objects. The values for <code>additionalFields</code> take precedence over the context or the <code>insurancePolicy</code> JSON. Address and Geolocation field types are not supported as additional fields. Add the <code>SourceQuotedId</code> standard field to the <code>additionalFields</code> option when the context ID in the input JSON for the Issue Insurance Policy API or the Multiroot Issue Insurance Policy API is generated by the Create Insurance Quote API.	Optional	63.0
contextId	String	Context ID of the policy or quote that's used to create the policy.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that's used to create the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that's used to override transaction information.	Optional	63.0

Response body for POST

[Policy](#)

Out-of-Sequence Endorse Policy (POST)

Create an out-of-sequence endorsement for an insurance policy.

Resource

```
/connect/insurance/policies/policyId/out-of-sequence-endorsement
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/00000000000000000000000000000000/out-of-sequence-endorsement
```

Available version

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of the insurance policy to create the out-of-sequence endorsement for.	Required	65.0

Request body for POST**JSON example**

```
{
  "contextId": "00000000i18tq18g00291750179239892137c4250a82940e49987f6d97559b4de",
  "effectiveDate": "2025-03-05",
  "insurancePolicy": {
    "policyName": "Life Insurance",
    "policyNumber": "LI000aax8967245gj"
  },
  "fieldSetIds": {
    "policyFieldSetId": "0IXxx0000004CYqGAM",
    "assetFieldSetId": "0IXxx0000004CaSGAU",
    "participantFieldSetId": "0IXxx0000004CdgGAE",
    "coverageFieldSetId": "0IXxx0000004Cc4GAE"
  },
  "transactionRecord": {
    "name": "Custom OOSE transaction"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context that's used to create an endorsement policy version.	Required	65.0
effectiveDate	String	Date from which the out-of-sequence endorsement applies. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set <code>effectiveDate</code> to 2025-03-01, the	Required	65.0

Name	Type	Description	Required or Optional	Available Version
		endorsement applies starting March 1, 2025, and the API re-rates the policy from 2025-03-01 forward.		
fieldSetIds	Additional Fields Field Set Input[]	Field set IDs of the additional fields that are associated with the insurance policy.	Optional	65.0
insurancePolicy	Insurance Policy Input[]	Details of the insurance policy to create the out-of-sequence endorsement for. It contains fields and values from the Insurance Policy object.	Optional	65.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that's used to override transaction information.	Optional	65.0

Response body for POST[Out-Of-Sequence Endorse Policy](#)**Reinstate Insurance Policy (POST)**

Reinstate a previously canceled insurance policy.

Resource`/connect/insurance/policies/policyId/reinstate`**Resource example**`https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTx0000004CdeFGH/reinstate`**Available version**

65.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of the insurance policy to be reinstated.	Required	65.0

Request body for POST**JSON example**

```
{
  "effectiveDate": "2025-09-15",
  "insurancePolicy": {
```



```

    "policyName": "Reinstate Auto Policy"
  },
  "transactionRecord": {
    "name": "Reinstatement Transaction"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "ReinstatementReason": "Payment Received"
        }
      }
    ]
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>additionalFields</code>	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> , <code>InsurancePolicyCoverage</code> , and <code>InsurancePolicyParticipant</code> objects. The values for <code>additionalFields</code> take precedence over the context or the <code>insurancePolicy</code> parameter. Address and Geolocation field types are not supported as additional fields.	Optional	65.0
<code>effectiveDate</code>	String	Date from the policy reinstatement becomes effective. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set <code>effectiveDate</code> to 2025-03-01, the reinstatement applies starting March 1, 2025.	Optional	65.0
<code>insurancePolicy</code>	Insurance Policy Input	Details of the insurance policy. It contains fields and values from the Insurance Policy object.	Optional	65.0
<code>transactionRecord</code>	Insurance Policy Transaction Input	JSON structure that's used to override transaction information.	Optional	65.0

Response body for POST[Policy](#)

Renew Insurance Policy (POST)

Use the context ID of an existing policy to renew a policy. Modify the renewal date and also add optional fields and additional information in the input JSON.

Resource

```
/connect/insurance/policies/policyId/renew
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTBxx000000000WEGAY/renew
```

Available version

63.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "contextId": "54920265842e1b61496c38b7e2202f6f0aff8123ad2f19fe17ac68f7fc1c60b3",
  "insurancePolicy": {
    "policyName": "Demo Test Policy 9",
    "policyNumber": "test renew policy",
    "effectiveFromDate": "2025-03-15",
    "effectiveToDate": "2026-03-15"
  },
  "transactionRecord": {
    "name": "Renew Payment"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoRoot",
          "fields": {
            "Checkbox__c": "true",
            "Currency__c": 123456,
            "Date__c": "2024-11-30",
            "DateTime__c": "2024-12-31T01:34:00.000Z",
            "Email__c": "testuser@sf.com"
          }
        }
      }
    ]
  },
  {
    "additionalFieldsMapValue": {
      "instanceKey": "autoDriver",
      "fields": {
```

```
        "TextEncr__c": "233345675676",
        "Url__c": "https://thisistest.com",
        "Time__c": "12:05:00.000Z"
      }
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that’s used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Priced context ID of the policy or quote that’s used to renew the policy.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that’s used to renew the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that’s used to override the transaction information.	Optional	63.0

Response body for POST [Policy](#)

Multi-Root Policy Issuance (POST)

Asynchronously create a multi-root insurance policy along with its child policies. A multi-root policy includes multiple child policies of different product types under a single parent policy.

Resource

```
/connect/insurance/policies/multi-root
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/multi-root
```

Available version

66.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "contextId": "3561319f267fd53380bc38c049a01c8089876e833a321860766d35a9c40b67df",

  "insurancePolicy": {
    "policyName": "Mr. Bret Lees Policy",
    "policyNumber": "Policy-Bret-OneYear",
    "effectiveFromDate": "2024-01-01",
    "effectiveToDate": "2024-10-31"
  },
  "transactionRecord": {
    "name": "New Policy"
  },
  "additionalFields":{
    "additionalFieldsList":[
      {
        "additionalFieldsMapValue":{
          "instanceKey":"autoRoot",
          "fields":{
            "Checkbox__c":"true",
            "Currency__c":123456,
            "Date__c":"2024-11-30",
            "DateTime__c":"2024-12-31T01:34:00.000Z",
            "Email__c":"testuser@sf.com"
          }
        }
      }
    ]
  },
  {
    "additionalFieldsMapValue":{
      "instanceKey":"autoDriver",
      "fields":{
        "TextEncr__c":"233345675676",
        "Url__c":"https://thisistest.com",
        "Time__c":"12:05:00.000Z"
      }
    }
  }
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>additionalFields</code>	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> , <code>InsurancePolicyCoverage</code> , and <code>InsurancePolicyParticipant</code> objects. The values for <code>additionalFields</code> take precedence over the context or the <code>insurancePolicy</code> JSON. Address and Geolocation field types are not supported as additional fields. Add the <code>SourceQuotedId</code> standard field to the <code>additionalFields</code> option when the context ID in the input JSON for the Issue Insurance Policy API or the Multiroot Issue Insurance Policy API is generated by the Create Insurance Quote API.	Optional	63.0
<code>contextId</code>	String	Context ID of the policy or quote that's used to create the policy.	Required	63.0
<code>insurancePolicy</code>	Insurance Policy Input[]	JSON structure that's used to create the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
<code>transactionRecord</code>	Insurance Policy Transaction Input[]	JSON structure that's used to override transaction information.	Optional	63.0

Response body for POST

Multi-Root Policy Asynchronous Response

Multi-Root Policy Endorsement (POST)

Asynchronously endorse a multi-root insurance policy along with its child policies.

Resource

```
/connect/insurance/policies/policyId/multi-root/endorse
```

Resource example

<https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YTex00000000WHGAY/multi-root/endorse>

Available version

66.0

HTTP methods

POST

Path parameter for POST

Parameter Name	Type	Description	Required or Optional	Available Version
policyId	String	ID of the policy that must be endorsed along with its associated child policies.	Required	66.0

Request body for POST

JSON example

```
{
  "contextId": "54920265842e1b61496c38b7e2202f6f0aff8123ad2f19fe17ac68f7fc1c60b3",
  "effectiveDate": "2024-01-01",
  "insurancePolicy": {
    "policyName": "p2_endorse_name",
    "policyNumber": "p2_endorse_number"
  },
  "transactionRecord": {
    "name": "New Endorsement"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoRoot",
          "fields": {
            "Checkbox__c": "true",
            "Currency__c": 123456,
            "Date__c": "2024-11-30",
            "DateTime__c": "2024-12-31T01:34:00.000Z",
            "Email__c": "testuser@sf.com"
          }
        }
      },
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoDriver",
          "fields": {
            "TextEncr__c": "233345675676",
            "Url__c": "https://thisistest.com",
            "Time__c": "12:05:00.000Z"
          }
        }
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Context ID of the quote that's used for endorsement.	Required	63.0
effectiveDate	String	Date from when the endorsed policy version is effective.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that overrides the name and policy number of the endorsed policy version.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that overrides the transaction information.	Optional	63.0

Response body for POST[Multi-Root Policy Asynchronous Response](#)**Multi-Root Policy Renewal (POST)**

Asynchronously renew a multi-root insurance policy, create renewed records for the parent policy and the associated child policies.

Resource

```
/connect/insurance/policies/policyId/multi-root/renew
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/policies/0YDxx000000000W0H3AY/multi-root/renew
```

Path parameter for POST

Parameter Name	Type	Description	Required or Optional
policyId	String	ID of the policy that must be renewed along with its associated child policies.	Required

Available version

66.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "contextId": "54920265842e1b61496c38b7e2202f6f0aff8123ad2f19fe17ac68f7fc1c60b3",
  "insurancePolicy": {
    "policyName": "Demo Test Policy 9",
    "policyNumber": "test renew policy",
    "effectiveFromDate": "2025-03-15",
    "effectiveToDate": "2026-03-15"
  },
  "transactionRecord":{
    "name": "Renew Payment"
  },
  "additionalFields":{
    "additionalFieldsList":[
      {
        "additionalFieldsMapValue":{
          "instanceKey":"autoRoot",
          "fields":{
            "Checkbox__c":"true",
            "Currency__c":123456,
            "Date__c":"2024-11-30",
            "DateTime__c":"2024-12-31T01:34:00.000Z",
            "Email__c":"testuser@sf.com"
          }
        }
      }
    ]
  },
  {
    "additionalFieldsMapValue":{
      "instanceKey":"autoDriver",
      "fields":{
        "TextEncr__c":"233345675676",
        "Url__c":"https://thisistest.com",
        "Time__c":"12:05:00.000Z"
      }
    }
  }
}
```



```
}
  }
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>additionalFields</code>	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> , <code>InsurancePolicyCoverage</code> , and <code>InsurancePolicyParticipant</code> objects. The values for <code>additionalFields</code> take precedence over the context or the <code>insurancePolicy</code> JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
<code>contextId</code>	String	Priced context ID of the policy or quote that's used to renew the policy.	Required	63.0
<code>insurancePolicy</code>	Insurance Policy Input[]	JSON structure that's used to renew the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
<code>transactionRecord</code>	Insurance Policy Transaction Input[]	JSON structure that's used to override the transaction information.	Optional	63.0

Response body for POST

[Multi-Root Policy Asynchronous Response](#)

Requests

Learn more about the available Insurance Policy requests.

[Additional Fields Field Set Input](#)

Input representation for the list of the field set IDs for asset, coverage, participant, and policy objects.

[Additional Fields Map Value](#)

Input representation for the `AdditionalFieldsMapValue` wrapper object that's used as child wrapper of `AdditionalFieldsMap`.

[Cancel Policy Input](#)

Input Representation of cancel policy details.

Endorse Policy Input

Input Representation of endorse policy details.

Insurance Policy Input

Input representation of insurance policy details.

Insurance Policy Transaction Input

Input representation of the insurance policy transaction details.

Issue Policy Input

Input representation of the request to issue an insurance policy.

Out-of-Sequence Endorse Policy Input

Input representation of the details for creating the out-of-sequence endorsement.

Reinstate Policy Input

Input representation for reinstating an insurance policy.

Additional Fields Field Set Input

Input representation for the list of the field set IDs for asset, coverage, participant, and policy objects.

JSON example

```
{
  "policyFieldSetId": "0IXxx0000004CYqGAM",
  "assetFieldSetId": "0IXxx0000004CaSGAU",
  "participantFieldSetId": "0IXxx0000004CdgGAE",
  "coverageFieldSetId": "0IXxx0000004Cc4GAE"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
assetFieldSetId	String	ID of the field set to get the fields that are associated with the asset object.	Optional	65.0
coverageFieldSetId	String	ID of the field set to get the fields that are associated with the coverage object.	Optional	65.0
participantFieldSetId	String	ID of the field set to get the fields that are associated with the participant object.	Optional	65.0
policyFieldSetId	String	ID of the field set to get the fields that are associated with the policy object.	Optional	65.0

Additional Fields Map Value

Input representation for the AdditionalFieldsMapValue wrapper object that's used as child wrapper of AdditionalFieldsMap.

JSON example

```
additionalFields:{
  "additionalFieldsList": [
```

```

    {
      "additionalFieldsMapValue":{
        "instanceKey":"autoRoot",
        "fields":{
          "Checkbox__c":"true",
          "Currency__c":123456,
          "Date__c":"2024-11-30",
          "DateTime__c":"2024-12-31T01:34:00.000Z",
          "Email__c":"testuser@sf.com"
        }
      }
    }
  ]
}

{
  "additionalFieldsMapValue":{
    "instanceKey":"autoDriver",
    "fields":{
      "TextEncr__c":"233345675676",
      "Url__c":"https://thisistest.com",
      "Time__c":"12:05:00.000Z"
    }
  }
}
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFieldsMapValue	Object	JSON map that contains the fields to be updated.	Required	63.0

Cancel Policy Input

Input Representation of cancel policy details.

JSON example

```
{
  "effectiveDate": "2024-09-08",
  "transactionRecord":{
    "name" : "Cancel Transaction"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the cancellation of the policy goes into effect.	Required	63.0

Name	Type	Description	Required or Optional	Available Version
transactionRecord	Insurance Policy Transaction Input[]	The JSON structure that overrides the transaction information.	Optional	63.0

Endorse Policy Input

Input Representation of endorse policy details.

JSON example

```
{
  "contextId": "54920265842e1b61496c38b7e2202f6f0aff8123ad2f19fe17ac68f7fc1c60b3",
  "effectiveDate": "2024-01-01",
  "insurancePolicy": {
    "policyName": "p2_endorse_name",
    "policyNumber": "p2_endorse_number"
  },
  "transactionRecord": {
    "name": "New Endorsement"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoRoot",
          "fields": {
            "Checkbox__c": "true",
            "Currency__c": 123456,
            "Date__c": "2024-11-30",
            "DateTime__c": "2024-12-31T01:34:00.000Z",
            "Email__c": "testuser@sf.com"
          }
        }
      },
      {
        "additionalFieldsMapValue": {
          "instanceKey": "autoDriver",
          "fields": {
            "TextEncr__c": "233345675676",
            "Url__c": "https://thisistest.com",
            "Time__c": "12:05:00.000Z"
          }
        }
      }
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Context ID of the quote that's used for endorsement.	Required	63.0
effectiveDate	String	Date from when the endorsed policy version is effective.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that overrides the name and policy number of the endorsed policy version.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that overrides the transaction information.	Optional	63.0

Insurance Policy Input

Input representation of insurance policy details.

JSON example

```
insurancePolicy": {
  "policyName": "Mr. Bret Lees Policy",
  "policyNumber": "Policy-Bret-OneYear",
  "effectiveFromDate": "2024-01-01",
  "effectiveToDate": "2024-10-31",
  "referencePolicyNumber": "00AXSYH567INS"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Date that the policy is effective from. This property isn't applicable to the policy endorsement and reinstatement APIs.	Optional	63.0
effectiveToDate	String	Expiration date of the policy. This property isn't applicable to the policy endorsement and reinstatement APIs.	Optional	63.0
policyName	String	Name of the insurance policy.	Optional	63.0
policyNumber	String	Number of the policy.	Optional	63.0
referencePolicy Number	String	Reference number that's used to link policies across terms. If not specified, policies are linked by the insurance policy ID. This property is applicable to the Issue Policy API and supports both single-root and multi-root policies.	Optional	65.0

Insurance Policy Transaction Input

Input representation of the insurance policy transaction details.

JSON example

```
transaction": {  
  "name": Cancel Transaction  
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
name	String	Name of the transaction, such as Cancel Transaction.	Optional	63.0

Issue Policy Input

Input representation of the request to issue an insurance policy.

JSON example

```
{
  "contextId": "3561319f267fd53380bc38c049a01c8089876e833a321860766d35a9c40b67df",
  "insurancePolicy": {
    "policyName": "Mr. Bret Lees Policy",
    "policyNumber": "Policy-Bret-OneYear",
    "effectiveFromDate": "2024-01-01",
    "effectiveToDate": "2024-10-31"
  },
  "transactionRecord": {
    "name": "New Policy"
  },
  "additionalFields":{
    "additionalFieldsList":[
      {
        "additionalFieldsMapValue":{
          "instanceKey":"autoRoot",
          "fields":{
            "Checkbox__c":"true",
            "Currency__c":123456,
            "Date__c":"2024-11-30",
            "DateTime__c":"2024-12-31T01:34:00.000Z",
            "Email__c":"testuser@sf.com"
          }
        }
      }
    ]
  },
  {
    "additionalFieldsMapValue":{
      "instanceKey":"autoDriver",
      "fields":{
        "TextEncr__c":"233345675676",
        "Url__c":"https://thisistest.com",
        "Time__c":"12:05:00.000Z"
      }
    }
  }
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that’s used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
		additionalFields take precedence over the context or the insurancePolicy JSON. Address and Geolocation field types are not supported as additional fields. Add the SourceQuoted standard field to the additionalFields option when the context ID in the input JSON for the Issue Insurance Policy API or the Multiroot Issue Insurance Policy API is generated by the Create Insurance Quote API.		
contextId	String	Context ID of the policy or quote that's used to create the policy.	Required	63.0
insurancePolicy	Insurance Policy Input[]	JSON structure that's used to create the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that's used to override transaction information.	Optional	63.0

Out-of-Sequence Endorse Policy Input

Input representation of the details for creating the out-of-sequence endorsement.

JSON example

```
{
  "contextId": "0000000i18tq18g00291750179239892137c4250a82940e49987f6d97559b4de",
  "effectiveDate": "2025-03-05",
  "insurancePolicy": {
    "policyName": "Life Insurance",
    "policyNumber": "LI000aax8967245gj"
  },
  "fieldSetIds": {
    "policyFieldSetId": "0IXxx0000004CYqGAM",
    "assetFieldSetId": "0IXxx0000004CaSGAU",
    "participantFieldSetId": "0IXxx0000004CdGAE",
    "coverageFieldSetId": "0IXxx0000004Cc4GAE"
  },
  "transactionRecord": {
    "name": "Custom OOSE transaction"
  }
}
```


Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context that's used to create an endorsement policy version.	Required	65.0
effectiveDate	String	Date from which the out-of-sequence endorsement applies. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set <code>effectiveDate</code> to 2025-03-01, the endorsement applies starting March 1, 2025, and the API re-rates the policy from 2025-03-01 forward.	Required	65.0
fieldSetIds	Additional Fields Field Set Input[]	Field set IDs of the additional fields that are associated with the insurance policy.	Optional	65.0
insurancePolicy	Insurance Policy Input[]	Details of the insurance policy to create the out-of-sequence endorsement for. It contains fields and values from the Insurance Policy object.	Optional	65.0
transactionRecord	Insurance Policy Transaction Input[]	JSON structure that's used to override transaction information.	Optional	65.0

Reinstate Policy Input

Input representation for reinstating an insurance policy.

JSON example

```
{
  "effectiveDate": "2025-09-15",
  "insurancePolicy": {
    "policyName": "Reinstate Auto Policy"
  },
  "transactionRecord": {
    "name": "Reinstatement Transaction"
  },
  "additionalFields": {
    "additionalFieldsList": [
      {
        "additionalFieldsMapValue": {
          "ReinstatementReason": "Payment Received"
        }
      }
    ]
  }
}
```

```
  }  
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Additional Fields Map Value[]	JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy parameter. Address and Geolocation field types are not supported as additional fields.	Optional	65.0
effectiveDate	String	Date from the policy reinstatement becomes effective. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set effectiveDate to 2025-03-01, the reinstatement applies starting March 1, 2025.	Optional	65.0
insurancePolicy	Insurance Policy Input	Details of the insurance policy. It contains fields and values from the Insurance Policy object.	Optional	65.0
transactionRecord	Insurance Policy Transaction Input	JSON structure that's used to override transaction information.	Optional	65.0

Responses

Learn more about the available response bodies of Insurance Policy APIs.

[Cancel Policy](#)

Output representation of cancel policy details.

[Get Policy](#)

Output representation of policy details.

[Policy](#)

Output representation of policy details.

[Refund](#)

Output representation of refund details.

[Out-of-Sequence Endorse Policy](#)

Output representation of the details for the out of sequence policy endorsement process.

[Error Details](#)

Output representation for the details of the error code and error message.

[Error Response](#)

Output representation of the error details.

[Multi-Root Policy Asynchronous Response](#)

Details of the multi-root policy that's processed asynchronously.

Cancel Policy

Output representation of cancel policy details.

JSON example

```
{
  "impactedPolicyVersions": [
    "0YTx000000006TGAQ"
  ],
  "refund": {
    "fee": 15.71,
    "premium": 31.42,
    "tax": 15.71
  }
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
error	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
impactedPolicyVersions	String[]	List of impacted policy versions.	Small, 63.0	63.0
refund	Refund	Refund associated with cancellation operation.	Small, 63.0	63.0

Get Policy

Output representation of policy details.

JSON example

This example shows a sample response when you pass a policy ID in the get policy request.

```
{
  "insurancepolicy" : {
    "id" : "xxxxxxxxXxx",
    "policyName" : "policyName",
    "policyNumber" : "PN-10012",
  }
}
```

```

"universalPolicyNumber" : "TNS-123981",
"effectiveFromDate" : "2023/01/01", //ISO format - yyyy-mm-dd
"effectiveToDate" : "2023/12/31",
..
..
"insurancePolicyCoverages" : [
    {
        "coverageName" : "Collision",
        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
    },
    {
        "coverageName" : "Theft Protection",
        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
    }
]
"insurancePolicyAssets" : [
    {
        "assetName" : "Audi A3",
        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
        "insurancePolicyCoverages" : [
            {
                "coverageName" : "Collision",
                "effectiveFromDate" : "2023/01/01",
                "effectiveToDate" : "2023/12/31",
                ..
                ..
            },
            {
                "coverageName" : "Theft Protection",
                "effectiveFromDate" : "2023/01/01",
                "effectiveToDate" : "2023/12/31",
                ..
                ..
            }
        ]
    },
    {
        "assetName" : "Audi A4",
        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
        "insurancePolicyCoverages" : [
            {
                "coverageName" : "Collision",

```

```

        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
    },
    {
        "coverageName" : "Theft Protection",
        "effectiveFromDate" : "2023/01/01",
        "effectiveToDate" : "2023/12/31",
        ..
        ..
    }
]
}
]
}
}

```

This example shows a sample response when you pass the `contextOnly` query parameter as true in the request. When the `contextOnly` parameter is true, the API returns a context ID along with the insurance policy details.

```

{
  "contextId": "e02036acf464f323f197c200895fec48a6a62ae3c711b3603609f08817c4af48",
  "insurancePolicy": {}
}

```

This example shows a sample response when `expand[attributes]` query parameter is set to true.

```

{
  "insurancePolicy": {
    "effectiveFromDate": "2024-01-01",
    "effectiveToDate": "2024-09-29",
    "originalEffectiveFromDate": "2024-01-01",
    "originalEffectiveToDate": "2024-12-31",
    "active": true,
    ....
  },
  "grossWrittenPremium": 1678.28,
  "insurancePolicyAssets": [
    {
      "effectiveFromDate": "2024-01-01",
      "effectiveToDate": "2024-09-29",
      "assetName": "Auto Bundle",
      "id": "0YWSG000000N4f4AE",
      ....
    },
    {
      "effectiveFromDate": "2024-01-01",
      "effectiveToDate": "2024-09-29",
      "id": "0aoSG000000sNvhYAE",
      ....
    }
  ],
  "insurancePolicyAssets": [],
  "insurancePolicyParticipants": [
    {
      "effectiveFromDate": "2024-01-01",
      "effectiveToDate": "2024-09-29",
      "id": "0aoSG000000sNvhYAE",
      ....
    }
  ]
}

```

```

.....

    "insurancePolicyAssets": [],
    "insurancePolicyCoverages": [],
    "insurancePolicyParticipants": [],
    "insurancePolicyParticipantAttributes": [
      {
        "id": "19USG0000004r382AA",
        "attributeDefinitionId": "0tjSG00000049D8YAI",
        "attributeName": "DriverAccidents",
        "attributeValue": "2.0",
        "insurancePolicyParticipantId": "0aoSG000000sNvhYAE"
      }
    ],
    "additionalFields": {}
  },
  "insurancePolicyCoverages": [
    {
      "coverageName": "Comprehensive",
      "id": "0cYSG0000005gt32AA",
      "insurancePolicyAssetId": "0YWSG0000000N4f4AE",
      "insurancePolicyId": "0YTSG000000233R4AQ",
      "name": "IPC-00000034",
.....
.....

    "insurancePolicyCoverageAttributes": [
      {
        "id": "19VSG0000004r2w2AA",
        "attributeDefinitionId": "0tjSG00000049DjYAI",
        "attributeName": "LimitBI",
        "attributeValue": "100",
        "insurancePolicyCoverageId": "0cYSG0000005gt32AA"
      },
.....
.....

      {
        "id": "19VSG0000004r342AA",
        "attributeDefinitionId": "0tjSG00000049DXAY",
        "attributeName": "RentalNumDays",
        "attributeValue": "10",
        "insurancePolicyCoverageId": "0cYSG0000005gt32AA"
      }
    ],
    "additionalFields": {}
  },
  "insurancePolicyAssetAttributes": [
    {
      "id": "19QSG0000004r2X2AQ",
      "attributeDefinitionId": "0tjSG00000049E4YAI",
      "attributeName": "Body Type",
      "attributeValue": "NA",
      "insurancePolicyAssetId": "0YWSG0000000N4f4AE"
    },

```

```

.....
.....
        {
            "id": "19QSG0000004r2Y2AQ",
            "attributeDefinitionId": "0tjSG00000049DIYAY",
            "attributeName": "Alternative Fuel",
            "attributeValue": "NA",
            "insurancePolicyAssetId": "0YWSG0000000N4f4AE"
        }
    ],
    "additionalFields": {}
}
],
"additionalFields": {}
}
}

```

This example shows a sample response for the get policy request with additional fields. The request returns a list of field-set IDs when the operator is \$in and a single field-set ID when the operator is \$eq.

```

// GET
/contract/policies/0YSG00000233R4AQ?expand[additionalFields][fieldSet][$in]=0XSG000001AD34AG,0XSG000001A6f4AG,0XSG000001A634AG,0XSG000001A5H4AW

{
  "insurancePolicy": {
    "effectiveFromDate": "2024-01-01",
    "effectiveToDate": "2024-09-29",
    ....
    ....
    "insurancePolicyAssets": [
      {
        "effectiveFromDate": "2024-01-01",
        "effectiveToDate": "2024-09-29",
        "assetName": "Auto Bundle",
        "id": "0YWSG0000000N4f4AE",
        "assetCode": "BMW",
        ....
      },
      {
        "effectiveFromDate": "2024-01-01",
        "effectiveToDate": "2024-09-29",
        "assetName": "Auto Bundle",
        "id": "0YWSG0000000N4f4AE",
        "assetCode": "BMW",
        ....
      }
    ],
    "insurancePolicyAssets": [],
    "insurancePolicyParticipants": [
      {
        "effectiveFromDate": "2024-01-01",
        "effectiveToDate": "2024-09-29",
        "id": "0aoSG000000sNvhYAE",
        "insurancePolicyId": "0YTSG0000000233R4AQ",
        "name": "IPP-00000032",
        ....
        ....
      }
    ],
    "insurancePolicyParticipantAttributes": [],
    "additionalFields": {
      "InsurancePolicyId": "0YTSG0000000233R4AQ",
      "EffectiveFromDate": "2024-01-01",
      ....
    }
  }
}

```

```
        "Id": "0aoSG000000sNvhYAE"
      },
    ],
    "insurancePolicyCoverages": [
      {
        "coverageName": "Comprehensive",
        "id": "0cYSG0000005gt32AA",
        "insurancePolicyAssetId": "0YWSG0000000N4f4AE",
        ....
      },
      {
        "insurancePolicyCoverageAttributes": [],
        "additionalFields": {
          "CoverageName": "Comprehensive",
          "Id": "0cYSG0000005gt32AA"
        }
      }
    ],
    "insurancePolicyAssetAttributes": [],
    "additionalFields": {
      "Name": "IPA-00000035"
    }
  }
},
"additionalFields": {
  "CheckboxField_c": true,
  "EffectiveFromDate": "2024-01-01",
  "Name": "Admin User Policy001"
}
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
contextId	String	ID of the context.	Small, 63.0	63.0
error	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
insurancePolicy	Map<String, Object>	Details of insurance policy.	Small, 63.0	63.0

Policy

Output representation of policy details.

JSON example

This request shows a sample success response.

```
{
  "policyId": "0YTxx0000000253GAA"
}
```


Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response	List of errors that occurred during the processing of the API request.	Small, 63.0	63.0
originalPolicyId	String	ID of the original policy.	Small, 63.0	63.0
policyId	String	ID of the issued or endorsed policy.	Small, 63.0	63.0

Refund

Output representation of refund details.

JSON example

```
refund": {
  "fee": 15.71,
  "premium": 31.42,
  "tax": 15.71
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
fee	Double	Refund fee amount after cancellation.	Small, 63.0	63.0
premium	Double	Refund premium amount after cancellation.	Small, 63.0	63.0
tax	Double	Refund tax amount after cancellation.	Small, 63.0	63.0

Out-of-Sequence Endorse Policy

Output representation of the details for the out of sequence policy endorsement process.

JSON Example

```
{
  "error": [
    {
      "detail": "Looks like you don't have access to this feature. Your Salesforce admin can help with that.",
      "errors": [
        {
          "code": "INTERNAL_ERROR",
          "message": "Something went wrong. Try again later or contact your Salesforce admin for help."
        }
      ],
      "instance": "",
      "title": "Access Denied"
    }
  ],
  "title": "Access Denied"
}
```

```
{
  "jobIdentifier": "15Uxx0000004CUWEA2"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
error	Error Response[]	Error details when the out-of-sequence endorsement operation fails.	Small, 65.0	65.0
jobIdentifier	String	ID of the job that's created for the out-of-sequence endorsement process.	Small, 65.0	65.0

Error Details

Output representation for the details of the error code and error message.

JSON Example

```
{
  "code": "DATE_FIELD_MUST_BE_IN_ISO_FORMAT",
  "message": "Specify a date in the yyyy-MM-dd format."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
code	String	Error code.	Small, 62.0	62.0
message	String	Error message.	Small, 62.0	62.0

Error Response

Output representation of the error details.

JSON Example

```
{
  "detail": "Looks like you don't have access to this feature. Your Salesforce admin can help with that.",
  "errors": [
    {
      "code": "INTERNAL_ERROR",
      "message": "Something went wrong. Try again later or contact your Salesforce admin for help."
    }
  ],
  "instance": "",
  "title": "Access Denied"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
detail	String	Description of the error.	Small, 62.0	62.0
errors	Error Details[]	Error details for the API process.	Small, 62.0	62.0
instance	String	Instance where the error occurred.	Small, 62.0	62.0
title	String	Short label for the error type.	Small, 62.0	62.0

Multi-Root Policy Asynchronous Response

Details of the multi-root policy that's processed asynchronously.

JSON example

```
{
  "error": {
    "detail": "Error detail message.",
    "errors": [
      {
        "code": "ERROR_CODE",
        "message": "Error message."
      }
    ],
    "instance": "Instance info",
    "title": "Error Title"
  },
  "jobIdentifier": "08Pxx0000000001EAA"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
error	Error Output	List of errors that occurred during the processing of the API request.	Small, 66.0	66.0
jobIdentifier	String	ID of the asynchronous job that's generated to process the multi-root policy along with its child policies.	Small, 66.0	66.0

Insurance Policy Standard Invocable Actions

Use the standard invocable actions available with Insurance Policy to create, manage, and endorse insurance policies.

[Cancel Insurance Policy Action](#)

Cancel an insurance policy by using a set of user inputs that represent policy details.

[Create Out-of-sequence Policy Endorsement Action](#)

Create an out-of-sequence endorsement for an insurance policy.

[Endorse Insurance Policy Action](#)

Endorse an insurance policy by using a set of user inputs that represent policy details.

[Endorse Multi-Root Insurance Policy Action](#)

Create an endorsement for an existing multi-root insurance policy to apply updates or changes to the policy and its associated child policies.

[Get Insurance Policy Action](#)

Get the details of an insurance policy.

[Issue Insurance Policy Action](#)

Issue an insurance policy by using a set of user inputs that represent policy details.

[Issue Multi-Root Insurance Policy Action](#)

Create a new multi-root insurance policy along with its associated child policies asynchronously.

[Reinstate Insurance Policy Action](#)

Reinstate a previously canceled insurance policy.

[Renew Insurance Policy Action](#)

Renew an insurance policy by using a set of user inputs that represent policy details.

[Renew Insurance Policies Action](#)

Process the renewal of insurance policies in bulk.

[Renew Multi-Root Insurance Policy Action](#)

Renew an existing multi-root insurance policy along with its associated child policies asynchronously.

Cancel Insurance Policy Action

Cancel an insurance policy by using a set of user inputs that represent policy details.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/cancelInsurancePolicy`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer token

Inputs

Input	Details
cancelPolicyIAInputRep	Type Apex class Description An Apex ConnectApi.CancelPolicyIAInputRep record that contains details of the policy to be canceled.

Outputs

Output	Details
cancelPolicyIAOutputRep	Type Apex class Description An Apex ConnectApi.CancelPolicyIAOutputRep record that contains the details of the policy that was canceled.

Example

Sample Request

```
{
  "inputs": [
    {
      "CancelPolicyIAInputRep": {
        "policyId": "0YTxx00000001H3GAI",
        "effectiveDate": "2024-09-08",
        "transactionRecord": {
          "name": "Cancel Transaction"
        }
      }
    }
  ]
}
```

Sample Response

```
{
  "actionName": "cancelInsurancePolicy",
  "errors": null,
  "invocationId": null,
  "isSuccess": true,
  "outcome": null,
  "outputValues": {
```

```
"CancelPolicyIAOutputRep": {
  "impactedPolicyVersions": [
    "0YTSG000000PgSH4A0"
  ],
  "refund": {
    "premium": "133.4",
    "tax": "21.3",
    "fee": "32.3"
  },
  "sortOrder": -1,
  "version": 1
}
}
```

Create Out-of-sequence Policy Endorsement Action

Create an out-of-sequence endorsement for an insurance policy.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

- URI
- /services/data/v/actions/standard/createOutOfSeqPlcyEndorsement
- Formats
- JSON
- HTTP Methods
- POST
- Authentication
- Authorization: Bearer *token*

Inputs

Input	Details
outOfSequencePolicyEndorsementInputRep	Type Apex class Description Required. An Apex ConnectApi.OutOfSequenceEndorsePolicyInputRep record that contains the details to create the out-of-sequence policy endorsement.

Outputs

Output	Details
jobIdentifier	Type string Description ID of the job that's created for the out-of-sequence endorsement process.

Example

POST

This is a sample request to call this invocable action.

```
{
  "inputs": [
    {
      "contextId": "0000000i18tq18g00291750179239892137c4250a82940e49987f6d97559b4de",
      "effectiveDate": "2025-03-05",
      "policyId": ""
      "insurancePolicy": {
        "policyName": "Life Insurance",
        "policyNumber": "LI000aax8967245gj"
      },
      "fieldSetIds": {
        "policyFieldSetId": "0IXxx0000004CYqGAM",
        "assetFieldSetId": "0IXxx0000004CaSGAU",
        "participantFieldSetId": "0IXxx0000004CdGAE",
        "coverageFieldSetId": "0IXxx0000004Cc4GAE"
      },
      "transactionRecord": {
        "name": "Custom OOSE transaction"
      }
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
public with sharing class OutOfSequenceEndorsementInvocable {

    /**
     * Execute OOSE with custom parameters
     * Usage: OutOfSequenceEndorsement.executeOOSEWithParams('policyId', 'effectiveDate',
     'contextId');
     */
    public static void executeOOSEWithParams(String policyId, String effectiveDate, String
contextId) {
        // Validate input parameters
        if (String.isBlank(policyId) || String.isBlank(effectiveDate) ||
String.isBlank(contextId)) {
            System.debug('ERROR: All parameters (policyId, effectiveDate, contextId)
```

```

are required');
        return;
    }

    // OOSE IA - Create the standard Invocable Action for out of sequence endorsement

    Invocable.Action action =
    Invocable.Action.createStandardAction('createOutOfSeqPlcyEndorsement');

    // Instantiate the IA input object for OOSE
    ConnectApi.outOfSequencePolicyEndorsementInputRep policyInput =
        new ConnectApi.outOfSequencePolicyEndorsementInputRep();

    // Set parameters
    policyInput.effectiveDate = effectiveDate;
    policyInput.contextId = contextId;
    policyInput.policyId = policyId;

    // Populate Insurance Policy details (using IA classes)
    ConnectApi.InsPolicyIAInputRep insurancePolicy =
        new ConnectApi.InsPolicyIAInputRep();
    insurancePolicy.policyName = 'Lisa Auto Policy';
    insurancePolicy.policyNumber = 'Lisa Auto Policy';
    policyInput.insurancePolicy = insurancePolicy;

    // Set field sets (using IA field set representation)
    ConnectApi.FieldSetIdsInputRep fieldSetIds =
        new ConnectApi.FieldSetIdsInputRep();
    fieldSetIds.policyFieldSetId = '0IXZM0000008P684AE'; // Replace with your policy
    field set ID
    fieldSetIds.assetFieldSetId = '0IXZM0000004DDC4A2'; // Replace with your asset
    field set ID
    fieldSetIds.participantFieldSetId = '0IXZM0000004DDH4A2'; // Replace with your
    participant field set ID
    fieldSetIds.coverageFieldSetId = '0IXZM0000008P6D4AU'; // Replace with your
    coverage field set ID
    policyInput.fieldSetIds = fieldSetIds;

    // Populate Transaction record (using IA transaction representation)
    ConnectApi.InsPolicyTrxnIAInputRep transactionRecord =
        new ConnectApi.InsPolicyTrxnIAInputRep();
    transactionRecord.name = 'Custom OOSE Transaction';
    policyInput.transactionRecord = transactionRecord;

    // Set invocation parameters
    action.setInvocationParameter('outOfSequencePolicyEndorsementInputRep',
    policyInput);

    System.debug('Starting OOSE Invocable Action (createOutOfSeqPlcyEndorsement)...');

    System.debug('Policy ID: ' + policyId);
    System.debug('Effective Date: ' + effectiveDate);
    System.debug('Context ID: ' + contextId);

```



```

        try {
            // Invoke the action
            List<Invocable.Action.Result> results = action.invoke();

            System.debug('OOSE Invocable Action result = ' + results);

            // Process results
            for (Invocable.Action.Result result : results) {
                if (result.isSuccess()) {
                    System.debug('SUCCESS: Custom OOSE completed successfully!');
                    Object outputValue =
result.getOutputParameters().get('jobIdentifier');
                    if (outputValue != null) {
                        System.debug('Job Identifier: ' + outputValue);
                    } else {
                        System.debug('OOSE job started - result: ' +
result.getOutputParameters());
                    }
                } else {
                    System.debug('ERROR: Custom OOSE failed');
                    System.debug('Error Details: ' + result.getErrors());
                }
            }

        } catch (Exception e) {
            System.debug('EXCEPTION in Custom OOSE Invocable Action: ' + e.getMessage());
        }
    }
}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */

OutOfSequenceEndorsementInvocable.executeOOSEWithParams(
    '0YTZM000000000xG4AQ', // policyId
    '2025-10-01',           // effectiveDate
    '0000000i18tq18g002917575994042598cecb38226a341dd96349ee513c7c1e5' // contextId
);

```

This is a sample response when you call this action.

```

[
  {
    "actionName": "createOutOfSeqPlcyEndorsement",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "jobIdentifier": "15Uxx0000004CUWEA2"
    },
    "sortOrder": -1,
  }
]

```

```
    "version": 1
  }
]
```

Endorse Insurance Policy Action

Endorse an insurance policy by using a set of user inputs that represent policy details.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/endorseInsurancePolicy

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer token

Inputs

Input	Details
endorsePolicyIAInputRep	Type Apex class Description An Apex ConnectApi.EndorsePolicyIAInputRep record that contains the details of the policy to be endorsed.

Outputs

Output	Details
endorsedPolicyId	Type String Description The ID of the endorsed insurance policy.

Example

Sample Request

```
{
  "inputs": [
    {
      "EndorsePolicyIAInputRep": {
        "contextId":
"5dbc40312f69b2d8cb4e7042a700d06f32fc5b9df7de3eabc94aa19a66ed77b3",
        "policyId": "0YTxx00000001IfGAI",
        "effectiveDate" : "2024-06-06",
        "insurancePolicy": {
          "policyName": "Endorsed Test Auto Ins policy",
          "policyNumber": "PLC5678987"
        },
        "transactionRecord": {
          "name": "Endorsed Policy Transaction"
        },
        "additionalInput": [
          {
            "instanceKey": "AutoBundle2",
            "additionalFieldsList": [
              {
                "key": "AssetDescription__c",
                "value": "Test_AssetDescription__c"
              },
              {
                "key": "SourceSystem",
                "value": "Test_SourceSystem"
              }
            ]
          },
          {
            "instanceKey": "AutoDriver",
            "additionalFieldsList": [
              {
                "key": "ParticipantName",
                "value": "Test_ParticipantName"
              },
              {
                "key": "Participant_Hometown__c",
                "value": "Test_Participant_Hometown__c"
              }
            ]
          }
        ],
        "instanceKey": "AutoRoot",
        "additionalFieldsList": [
          {
            "key": "Primary_Policy_Holder__c",
            "value": "Test_Primary_Policy_Holder__c"
          },
          {
            "key": "PolicyDescription",

```

```

        "value": "Test_PolicyDescription"
      }
    ],
    },
    {
      "instanceKey": "AutoBundle2_AutoComp1",
      "additionalFieldsList": [
        {
          "key": "Coverage_Description__c",
          "value": "Test_Coverage_Description__c"
        },
        {
          "key": "Description",
          "value": "Test_Description"
        }
      ]
    }
  ]
}

```

Sample Response

```

[
  {
    "actionName": "endorseInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "endorsedPolicyId": "0YTSG000000PgVV4A0"
    },
    "sortOrder": -1,
    "version": 1
  }
]

```

Endorse Multi-Root Insurance Policy Action

Create an endorsement for an existing multi-root insurance policy to apply updates or changes to the policy and its associated child policies.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/endorseMultiRootInsPolicy

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
endorseMultiRootPolicyInputRep	Type Apex-defined Description Required. An Apex ConnectApi.EndorseMultiRootPolicyInputRep record that contains the details of the policy to be endorsed.

Outputs

Output	Details
jobIdentifier	Type string Description A unique system-generated ID for the job that processes the endorsement of the multi-root insurance policy.

Example

POST

This sample request is for the Endorse Multi-Root Insurance Policy action.

```
{
  "inputs": [
    {
      "endorseMultiRootPolicyInputRep": {
        "contextId": "5dbc40312f69b2d8cb4e7042a700d06f32fc5b9df7de3eabc94aa19a66ed77b3",

        "policyId": "0YTx00000001IfGAI",
        "effectiveDate": "2024-06-06",
        "insurancePolicy": {
          "policyName": "Endorsed Test Auto Ins policy",
          "policyNumber": "PLC5678987"
        }
      },
    },
  ],
}
```

```

"transactionRecord": {
  "name": "Endorsed Policy Transaction"
},
"additionalInput": [
  {
    "instanceKey": "AutoBundle2",
    "additionalFieldsList": [
      {
        "key": "AssetDescription__c",
        "value": "Test_AssetDescription__c"
      },
      {
        "key": "SourceSystem",
        "value": "Test_SourceSystem"
      }
    ]
  },
  {
    "instanceKey": "AutoDriver",
    "additionalFieldsList": [
      {
        "key": "ParticipantName",
        "value": "Test_ParticipantName"
      },
      {
        "key": "Participant_Hometown__c",
        "value": "Test_Participant_Hometown__c"
      }
    ]
  },
  {
    "instanceKey": "AutoRoot",
    "additionalFieldsList": [
      {
        "key": "Primary_Policy_Holder__c",
        "value": "Test_Primary_Policy_Holder__c"
      },
      {
        "key": "PolicyDescription",
        "value": "Test_PolicyDescription"
      }
    ]
  },
  {
    "instanceKey": "AutoBundle2_AutoComp1",
    "additionalFieldsList": [
      {
        "key": "Coverage_Description__c",
        "value": "Test_Coverage_Description__c"
      },
      {
        "key": "Description",
        "value": "Test_Description"
      }
    ]
  }
]

```

```

        ]
    }
}
}
}
]
}

```

This is a sample code to call the Endorse Multi-Root Insurance Policy action from Apex.

```

public with sharing class MultiRootPolicyEndorsementInvocable {

    /**
     * Execute Multi-Root Policy Endorsement with parameters map
     * Usage: MultiRootPolicyEndorsementInvocable.executeMultiRootEndorsement(paramsMap);
     *
     * @param params Map of parameters:
     * - contextId (String, REQUIRED)
     * - policyId (String, REQUIRED)
     * - effectiveDate (String, REQUIRED)
     * - policyName (String, optional)
     * - policyNumber (String, optional)
     * - transactionName (String, optional)
     * - additionalFields (Map<String, Map<String, String>>, optional)
     */
    public static void executeMultiRootEndorsement(Map<String, Object> params) {
        // Extract and validate contextId (REQUIRED)
        String contextId = (String) params.get('contextId');
        if (String.isBlank(contextId)) {
            System.debug('ERROR: contextId is required');
            return;
        }

        // Extract and validate policyId (REQUIRED)
        String policyId = (String) params.get('policyId');
        if (String.isBlank(policyId)) {
            System.debug('ERROR: policyId is required');
            return;
        }

        // Extract and validate effectiveDate (REQUIRED)
        String effectiveDate = (String) params.get('effectiveDate');
        if (String.isBlank(effectiveDate)) {
            System.debug('ERROR: effectiveDate is required');
            return;
        }

        // Multi-Root Endorse Policy IA - Create the standard Invocable Action
        Invocable.Action action =
        Invocable.Action.createStandardAction('endorseMultiRootInsPolicy');

        // Instantiate the IA input object for Multi-Root Policy Endorsement
        // IMPORTANT: Use EndorseMultiRootPolicyInputRep, NOT EndorsePolicyIAInputRep
    }
}

```

```

ConnectApi.EndorseMultiRootPolicyInputRep policyInput =
    new ConnectApi.EndorseMultiRootPolicyInputRep();

// Set context ID (REQUIRED)
policyInput.contextId = contextId;

// Set policy ID (REQUIRED)
policyInput.policyId = policyId;

// Set effectiveDate (REQUIRED)
policyInput.effectiveDate = effectiveDate;

// Extract optional parameters
String policyName = (String) params.get('policyName');
String policyNumber = (String) params.get('policyNumber');
String transactionName = (String) params.get('transactionName');
Map<String, Map<String, String>> additionalFields =
    (Map<String, Map<String, String>>) params.get('additionalFields');

// Populate Insurance Policy details (using IA classes) - all optional
if (!String.isBlank(policyName) || !String.isBlank(policyNumber)) {

    ConnectApi.InsPolicyIAInputRep insurancePolicy =
        new ConnectApi.InsPolicyIAInputRep();

    if (!String.isBlank(policyName)) {
        insurancePolicy.policyName = policyName;
    }
    if (!String.isBlank(policyNumber)) {
        insurancePolicy.policyNumber = policyNumber;
    }
    policyInput.insurancePolicy = insurancePolicy;
}

// Populate Transaction record (using IA transaction representation) - optional
if (!String.isBlank(transactionName)) {
    ConnectApi.InsPolicyTrxnIAInputRep transactionRecord =
        new ConnectApi.InsPolicyTrxnIAInputRep();
    transactionRecord.name = transactionName;
    policyInput.transactionRecord = transactionRecord;
}

// Populate additional input for multi-root policies - optional
if (additionalFields != null && !additionalFields.isEmpty()) {
    List<ConnectApi.InsPolicyAddlInputIARep> additionalInputList = new
List<ConnectApi.InsPolicyAddlInputIARep>();

    for (String instanceKey : additionalFields.keySet()) {
        ConnectApi.InsPolicyAddlInputIARep additionalInput =
            new ConnectApi.InsPolicyAddlInputIARep();

        // Set instance key (e.g., "AutoSilver", "familyHealth")
        additionalInput.instanceKey = instanceKey;
    }
}

```



```

        // Build additional fields list for this instance
        Map<String, String> fieldsMap = additionalFields.get(instanceKey);
        if (fieldsMap != null && !fieldsMap.isEmpty()) {
            List<ConnectApi.AdditionalFieldIARep> fieldsList = new
List<ConnectApi.AdditionalFieldIARep>();

            for (String fieldKey : fieldsMap.keySet()) {
                ConnectApi.AdditionalFieldIARep field = new
ConnectApi.AdditionalFieldIARep();
                field.key = fieldKey;
                field.value = fieldsMap.get(fieldKey);
                fieldsList.add(field);
            }

            additionalInput.additionalFieldsList = fieldsList;
        }

        additionalInputList.add(additionalInput);
    }

    policyInput.additionalInput = additionalInputList;
}

// Set invocation parameters
// IMPORTANT: Parameter name is endorseMultiRootPolicyInputRep (camelCase)
action.setInvocationParameter('endorseMultiRootPolicyInputRep', policyInput);

System.debug('Starting Multi-Root Policy Endorsement Invocable Action
(endorseMultiRootInsPolicy)...');
System.debug('Context ID: ' + contextId);
if (!String.isBlank(policyName)) {
    System.debug('Policy Name: ' + policyName);
}

try {
    // Invoke the action
    List<Invocable.Action.Result> results = action.invoke();

    System.debug('Multi-Root Policy Endorsement IA result = ' + results);

    // Process results
    for (Invocable.Action.Result result : results) {
        if (result.isSuccess()) {
            System.debug('SUCCESS: Multi-Root Policy Endorsement completed
successfully!');
            Object outputValue =
result.getOutputParameters().get('jobIdentifier');
            if (outputValue != null) {
                System.debug('Job Identifier: ' + outputValue);
            } else {
                System.debug('Multi-Root Policy Endorsement job started - result:
' + result.getOutputParameters());
            }
        }
    }
}

```

```

        } else {
            System.debug('ERROR: Multi-Root Policy Endorsement failed');
            System.debug('Error Details: ' + result.getErrors());
        }
    }

    } catch (Exception e) {
        System.debug('EXCEPTION in Multi-Root Policy Endorsement Invocable Action:
' + e.getMessage());
        System.debug('Stack Trace: ' + e.getStackTraceString());
    }
}

}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */
Map<String, Object> params = new Map<String, Object>();
params.put('contextId',
'00000000i18tq18g002917629276616310d09368a7cb24db687f6fd95af4b9954');
params.put('policyId', '0YTSM0000001MzN4AU');
params.put('policyName', 'James Multi-Root Policy Apex IA');
params.put('effectiveDate', '2026-08-01');
params.put('transactionName', 'Endorsement Second');

// Build additional fields configuration
Map<String, Map<String, String>> additionalFields = new Map<String, Map<String,
String>>();

// Configuration for AutoSilver instance
Map<String, String> autoFields = new Map<String, String>();
autoFields.put('PolicyName', 'James Auto Policy IA Apex');
autoFields.put('Name', 'James Auto Policy');
additionalFields.put('AutoSilver', autoFields);

// Configuration for FamilyHealth instance
Map<String, String> healthFields = new Map<String, String>();
healthFields.put('PolicyName', 'James Medical Policy IA Apex');
healthFields.put('Name', 'James Medical Policy');
healthFields.put('Email__c', 'medical@james.co');
additionalFields.put('familyHealth', healthFields);

params.put('additionalFields', additionalFields);

MultiRootPolicyEndorsementInvocable.executeMultiRootEndorsement(params);

```

This sample response is for the Endorse Multi-Root Insurance Policy action.

```

[
  {
    "actionName": "endorseMultiRootInsPolicy",
    "errors": null,
    "invocationId": null,

```

```
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "jobIdentifier": "750x00000000001AAA"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Get Insurance Policy Action

Get the details of an insurance policy.
This action is available in API version 63.0 and later.

Supported REST HTTP Methods

- URI**
`/services/data/v/actions/standard/getInsurancePolicy`
- Formats**
JSON
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer token

Inputs

Input	Details
insurancePolicyId	Type String Description The ID of the insurance policy to be retrieved.

Outputs

Output	Details
contextId	Type String Description

Output**Details**

The context ID of the retrieved insurance policy.

Example

Sample Request

```
{
  "inputs": [
    {
      "policyId": "0YTxx00000001H3GAI"
    }
  ]
}
```

Sample Response

```
[
  {
    "actionName": "getInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "contextId": "0000000i18tq18g00291753332964253e35d47202c0c428785435cdfa2f44ae8"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Issue Insurance Policy Action

Issue an insurance policy by using a set of user inputs that represent policy details.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/issueInsurancePolicy

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer token

Inputs

Input	Details
issuePolicyIAInputRep	Type Apex class Description An Apex ConnectApi.IssuePolicyIAInputRep record that contains the details of the policy to be issued.

Outputs

Output	Details
issuedPolicyId	Type String Description The ID of the issued policy.

Usage

Sample Request

```
{
  "inputs": [
    {
      "IssuePolicyIAInputRep": {
        "contextId":
"c98a08a0a9725b7a7d42c35d02e49d80c77a5aae4ea687b0bf09a7997d2af3d3",
        "insurancePolicy": {
          "policyName": "Test Auto Ins policy Via REST",
          "policyNumber": "PLC5678987",
          "effectiveFromDate" : "2024-01-01",
          "effectiveToDate" : "2024-12-31",
          "standardFee": "1011"
        },
        "transactionRecord": {
          "name": "Sold Policy Transaction"
        },
        "additionalInput": [
          {
            "instanceKey": "AutoBundle2",
            "additionalFieldsList": [
              {
                "key":"AssetDescription__c",
```

```

        "value": "Test_AssetDescription__c"
    },
    {
        "key": "SourceSystem",
        "value": "Test_SourceSystem"
    }
]
},
{
    "instanceKey": "AutoDriver",
    "additionalFieldsList": [
        {
            "key": "ParticipantName",
            "value": "Test_ParticipantName"
        },
        {
            "key": "Participant_Hometown__c",
            "value": "Test_Participant_Hometown__c"
        }
    ]
},
{
    "instanceKey": "AutoRoot",
    "additionalFieldsList": [
        {
            "key": "Primary_Policy_Holder__c",
            "value": "Test_Primary_Policy_Holder__c"
        },
        {
            "key": "PolicyDescription",
            "value": "Test_PolicyDescription"
        }
    ]
},
{
    "instanceKey": "AutoBundle2_AutoCompl",
    "additionalFieldsList": [
        {
            "key": "Coverage_Description__c",
            "value": "Test_Coverage_Description__c"
        },
        {
            "key": "Description",
            "value": "Test_Description"
        }
    ]
}
}
]
}
}
]
```

Sample Response

```
[
  {
    "actionName": "issueInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "issuedPolicyId": "0YTSG000000PgSH4A0"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Issue Multi-Root Insurance Policy Action

Create a new multi-root insurance policy along with its associated child policies asynchronously.
This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/issueMultiRootInsurancePolicy

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
issueMultiRootPolicyInputRep	Type Apex-defined Description Required. An Apex ConnectApi.IssueMultiRootPolicyInputRep record that contains the details to create a new multi-root insurance policy along with its child policies.

Outputs

Output	Details
jobIdentifier	Type string Description A unique system-generated ID for the asynchronous job that processes the issuance of a new multi-root insurance policy.

Example

POST

This sample request is for the Issue Multi-Root Insurance Policy action.

```
{
  "inputs": [
    {
      "issueMultiRootPolicyInputRep": {
        "contextId": "c98a08a0a9725b7a7d42c35d02e49d80c77a5aae4ea687b0bf09a7997d2af3d3",

        "insurancePolicy": {
          "policyName": "Test Auto Ins policy Via REST",
          "policyNumber": "PLC5678987",
          "effectiveFromDate": "2024-01-01",
          "effectiveToDate": "2024-12-31",
          "referencePolicyNumber": "1011"
        },
        "transactionRecord": {
          "name": "Sold Policy Transaction"
        },
        "additionalInput": [
          {
            "instanceKey": "AutoBundle2",
            "additionalFieldsList": [
              {
                "key": "AssetDescription__c",
                "value": "Test_AssetDescription__c"
              },
              {
                "key": "SourceSystem",
                "value": "Test_SourceSystem"
              }
            ]
          },
          {
            "instanceKey": "AutoDriver",
            "additionalFieldsList": [
              {
                "key": "ParticipantName",
                "value": "Test_ParticipantName"
              }
            ]
          }
        ]
      }
    }
  ]
}
```



```

    },
    {
      "key": "Participant_Hometown__c",
      "value": "Test_Participant_Hometown__c"
    }
  ]
},
{
  "instanceKey": "AutoRoot",
  "additionalFieldsList": [
    {
      "key": "Primary_Policy_Holder__c",
      "value": "Test_Primary_Policy_Holder__c"
    },
    {
      "key": "PolicyDescription",
      "value": "Test_PolicyDescription"
    }
  ]
},
{
  "instanceKey": "AutoBundle2_AutoComp1",
  "additionalFieldsList": [
    {
      "key": "Coverage_Description__c",
      "value": "Test_Coverage_Description__c"
    },
    {
      "key": "Description",
      "value": "Test_Description"
    }
  ]
}
]
}
}
}
}
```

This is a sample code to call the Issue Multi-Root Insurance Policy action from Apex.

```
public with sharing class MultiRootPolicyIssuanceInvocable {

    /**
     * Execute Multi-Root Policy Issuance with parameters map
     * Usage: MultiRootPolicyIssuanceInvocable.executeMultiRootIssuance(paramsMap);
     *
     * @param params Map of parameters:
     *     - contextId (String, REQUIRED)
     *     - policyName (String, optional)
     *     - policyNumber (String, optional)
     *     - referencePolicyNumber (String, optional)
     *     - effectiveFromDate (String, optional)
     *     - effectiveToDate (String, optional)
     *     - transactionName (String, optional)
     */
}
```

```

    * - additionalFields (Map<String, Map<String, String>>, optional)
    */
    public static void executeMultiRootIssuance(Map<String, Object> params) {
        // Extract and validate contextId (REQUIRED)
        String contextId = (String) params.get('contextId');
        if (String.isBlank(contextId)) {
            System.debug('ERROR: contextId is required');
            return;
        }

        // Multi-Root Issue Policy IA - Create the standard Invocable Action
        Invocable.Action action =
        Invocable.Action.createStandardAction('issueMultiRootInsurancePolicy');

        // Instantiate the IA input object for Multi-Root Policy Issuance
        // IMPORTANT: Use IssueMultiRootPolicyInputRep, NOT IssuePolicyIAInputRep
        ConnectApi.IssueMultiRootPolicyInputRep policyInput =
            new ConnectApi.IssueMultiRootPolicyInputRep();

        // Set context ID (REQUIRED)
        policyInput.contextId = contextId;

        // Extract optional parameters
        String policyName = (String) params.get('policyName');
        String policyNumber = (String) params.get('policyNumber');
        String referencePolicyNumber = (String) params.get('referencePolicyNumber');
        String effectiveFromDate = (String) params.get('effectiveFromDate');
        String effectiveToDate = (String) params.get('effectiveToDate');
        String transactionName = (String) params.get('transactionName');
        Map<String, Map<String, String>> additionalFields =
            (Map<String, Map<String, String>>) params.get('additionalFields');

        // Populate Insurance Policy details (using IA classes) - all optional
        if (!String.isBlank(policyName) || !String.isBlank(policyNumber) ||
            !String.isBlank(referencePolicyNumber) || !String.isBlank(effectiveFromDate)
||
            !String.isBlank(effectiveToDate)) {

            ConnectApi.InsPolicyIAInputRep insurancePolicy =
                new ConnectApi.InsPolicyIAInputRep();

            if (!String.isBlank(policyName)) {
                insurancePolicy.policyName = policyName;
            }
            if (!String.isBlank(policyNumber)) {
                insurancePolicy.policyNumber = policyNumber;
            }
            if (!String.isBlank(referencePolicyNumber)) {
                insurancePolicy.referencePolicyNumber = referencePolicyNumber;
            }
            if (!String.isBlank(effectiveFromDate)) {
                insurancePolicy.effectiveFromDate = effectiveFromDate;
            }
            if (!String.isBlank(effectiveToDate)) {

```

```

        insurancePolicy.effectiveToDate = effectiveToDate;
    }

    policyInput.insurancePolicy = insurancePolicy;
}

// Populate Transaction record (using IA transaction representation) - optional
if (!String.isBlank(transactionName)) {
    ConnectApi.InsPolicyTrxnIAInputRep transactionRecord =
        new ConnectApi.InsPolicyTrxnIAInputRep();
    transactionRecord.name = transactionName;
    policyInput.transactionRecord = transactionRecord;
}

// Populate additional input for multi-root policies - optional
if (additionalFields != null && !additionalFields.isEmpty()) {
    List<ConnectApi.InsPolicyAddlInputIARep> additionalInputList = new
List<ConnectApi.InsPolicyAddlInputIARep>();

    for (String instanceKey : additionalFields.keySet()) {
        ConnectApi.InsPolicyAddlInputIARep additionalInput =
            new ConnectApi.InsPolicyAddlInputIARep();

        // Set instance key (e.g., "AutoSilver", "FamilyHealth")
        additionalInput.instanceKey = instanceKey;

        // Build additional fields list for this instance
        Map<String, String> fieldsMap = additionalFields.get(instanceKey);
        if (fieldsMap != null && !fieldsMap.isEmpty()) {
            List<ConnectApi.AdditionalFieldIARep> fieldsList = new
List<ConnectApi.AdditionalFieldIARep>();

            for (String fieldKey : fieldsMap.keySet()) {
                ConnectApi.AdditionalFieldIARep field = new
ConnectApi.AdditionalFieldIARep();
                field.key = fieldKey;
                field.value = fieldsMap.get(fieldKey);
                fieldsList.add(field);
            }

            additionalInput.additionalFieldsList = fieldsList;
        }

        additionalInputList.add(additionalInput);
    }

    policyInput.additionalInput = additionalInputList;
}

// Set invocation parameters
// IMPORTANT: Parameter name is issueMultiRootPolicyInputRep (camelCase)
action.setInvocationParameter('issueMultiRootPolicyInputRep', policyInput);

```

```

        System.debug('Starting Multi-Root Policy Issuance Invocable Action
(issueMultiRootInsurancePolicy)...');
        System.debug('Context ID: ' + contextId);
        if (!String.isBlank(policyName)) {
            System.debug('Policy Name: ' + policyName);
        }

        try {
            // Invoke the action
            List<Invocable.Action.Result> results = action.invoke();

            System.debug('Multi-Root Policy Issuance IA result = ' + results);

            // Process results
            for (Invocable.Action.Result result : results) {
                if (result.isSuccess()) {
                    System.debug('SUCCESS: Multi-Root Policy Issuance completed
successfully!');
                    Object outputValue =
result.getOutputParameters().get('jobIdentifier');
                    if (outputValue != null) {
                        System.debug('Job Identifier: ' + outputValue);
                    } else {
                        System.debug('Multi-Root Policy Issuance job started - result:
' + result.getOutputParameters());
                    }
                } else {
                    System.debug('ERROR: Multi-Root Policy Issuance failed');
                    System.debug('Error Details: ' + result.getErrors());
                }
            }

        } catch (Exception e) {
            System.debug('EXCEPTION in Multi-Root Policy Issuance Invocable Action: '
+ e.getMessage());
            System.debug('Stack Trace: ' + e.getStackTraceString());
        }
    }
}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */
Map<String, Object> params = new Map<String, Object>();
params.put('contextId', <your-context-id-here>);
params.put('policyName', 'James Multi-Root Policy');
params.put('policyNumber', 'James Multi-Root Policy');
params.put('referencePolicyNumber', 'James1234');
params.put('effectiveFromDate', '2026-01-01');
params.put('effectiveToDate', '2026-10-31');
params.put('transactionName', 'Sold Policy Transaction');

```

```
// Build additional fields configuration
Map<String, Map<String, String>> additionalFields = new Map<String, Map<String, String>>();

// Configuration for AutoSilver instance
Map<String, String> autoFields = new Map<String, String>();
autoFields.put('PolicyName', 'James Auto Policy');
autoFields.put('Name', 'James Auto Policy');
additionalFields.put(<your-instanceKey-goes-here>, autoFields);

// Configuration for FamilyHealth instance
Map<String, String> healthFields = new Map<String, String>();
healthFields.put('PolicyName', 'James Medical Policy');
healthFields.put('Name', 'James Medical Policy');
additionalFields.put(<your-instanceKey-goes-here>, healthFields);

params.put('additionalFields', additionalFields);

MultiRootPolicyIssuanceInvocable.executeMultiRootIssuance(params);
```

This sample response is for the Issue Multi-Root Insurance Policy action.

```
[
  {
    "actionName": "issueMultiRootInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "jobIdentifier": "55SG000000PgSH4A0"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Reinstate Insurance Policy Action

Reinstate a previously canceled insurance policy.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/reinstateInsurancePolicy

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
reinstatePolicyInputRep	Type Apex class Description Required. An Apex ConnectApi__ReinstatePolicyInputRep record that contains the policy details to reinstate the insurance policy.

Outputs

Output	Details
reinstatedPolicyId	Type string Description ID of the reinstated insurance policy.

Example

POST

This is a sample request to call this invocable action.

```
{
  "inputs": [
    {
      "ReinstatePolicyInputRep": {
        "policyId": "0YTxx0000004CdeFGH",
        "effectiveDate": "2024-12-30",
        "insurancePolicy": {
          "policyName": "Auto Insurance",
          "policyNumber": "PLC5678987",
          "effectiveFromDate": "2024-01-01",
          "effectiveToDate": "2024-12-31",
          "referencePolicyNumber": "PLC56011213"
        },
      },
      "transactionRecord": {
        "name": "Cancel Policy Transaction"
      },
      "additionalInput": [
        {
```

```

        "instanceKey": "AutoBundle2",
        "additionalFieldsList": [
            {
                "key": "AssetDescription__c",
                "value": "Test_AssetDescription__c"
            },
            {
                "key": "SourceSystem",
                "value": "Test_SourceSystem"
            }
        ]
    },
    {
        "instanceKey": "AutoDriver",
        "additionalFieldsList": [
            {
                "key": "ParticipantName",
                "value": "Test_ParticipantName"
            },
            {
                "key": "Participant_Hometown__c",
                "value": "Test_Participant_Hometown__c"
            }
        ]
    },
    {
        "instanceKey": "AutoRoot",
        "additionalFieldsList": [
            {
                "key": "Primary_Policy_Holder__c",
                "value": "Test_Primary_Policy_Holder__c"
            },
            {
                "key": "PolicyDescription",
                "value": "Test_PolicyDescription"
            }
        ]
    },
    {
        "instanceKey": "AutoBundle2_AutoCompl",
        "additionalFieldsList": [
            {
                "key": "Coverage_Description__c",
                "value": "Test_Coverage_Description__c"
            },
            {
                "key": "Description",
                "value": "Test_Description"
            }
        ]
    }
]
}
}

```

```

    ]
}

```

This is a sample request to call this invocable action from Apex code.

```

// Create the standard Invocable Action for policy reinstatement
Invocable.Action action =
Invocable.Action.createStandardAction('reinstateInsurancePolicy');

// Instantiate the input object for the policy reinstatement
ConnectApi.ReinstatePolicyInputRep policyInput = new ConnectApi.ReinstatePolicyInputRep();

// Required: Set the Policy ID of the policy to be reinstated
policyInput.policyId = '0YTxx00000000PpGAI';

// Optional: Populate Insurance Policy details
ConnectApi.InsPolicyIAInputRep insurancePolicy = new ConnectApi.InsPolicyIAInputRep();
insurancePolicy.policyName = 'Health Insurance Policy Reinstated';
insurancePolicy.policyNumber = 'HIPE12';

// Assign the insurance policy input to the main input object
policyInput.insurancePolicy = insurancePolicy;

// Optional: Add additional input fields (custom fields, etc.)
List<ConnectApi.AdditionalFieldIARep> additionalFields = new
List<ConnectApi.AdditionalFieldIARep>();

ConnectApi.AdditionalFieldIARep additionalFieldElement = new
ConnectApi.AdditionalFieldIARep();
additionalFieldElement.key = 'Currency__c';
additionalFieldElement.value = '6700';

additionalFields.add(additionalFieldElement);

// Create a wrapper for the additional input fields per policy instance

List<ConnectApi.InsPolicyAddlInputIARep> addlInputIARep = new
List<ConnectApi.InsPolicyAddlInputIARep>();

ConnectApi.InsPolicyAddlInputIARep addlInputIARepElement = new
ConnectApi.InsPolicyAddlInputIARep();
addlInputIARepElement.instanceKey = 'AutoSilver'; // Logical key representing the policy
instance
addlInputIARepElement.additionalFieldsList = additionalFields;

addlInputIARep.add(addlInputIARepElement);

// Assign additional input to the main input object
policyInput.additionalInput = addlInputIARep;

// Optional: Populate Transaction record information
ConnectApi.InsPolicyTrxnIAInputRep transactionRecord = new
ConnectApi.InsPolicyTrxnIAInputRep();
transactionRecord.name = 'Apex IA Transaction'; // A label for the transaction
policyInput.transactionRecord = transactionRecord;

```



```
action.setInvocationParameter('ReinstatePolicyInputRep', policyInput);
List<Invocable.Action.Result> results = action.invoke();

System.debug('Reinstate Invocable Action result = ' + results);
```

This is a sample response when you call this action.

```
[
  {
    "actionName": "reinstateInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "reinstatedPolicyId": "15Uxx0000004CUWEA2"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Renew Insurance Policy Action

Renew an insurance policy by using a set of user inputs that represent policy details.
This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/renewInsurancePolicy

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer token

Inputs

Input	Details
renewPolicyIAInputRep	Type Apex class Description

Input	Details
	An Apex ConnectApi.RenewPolicyIAInputRep record that contains the details of the policy to be renewed.

Outputs

Output	Details
renewedPolicyId	Type String Description The ID of the renewed policy.

Example

Sample Request

```
{
  "inputs": [
    {
      "RenewPolicyIAInputRep": {
        "contextId": "d767b27ec095f9f95c5a34299e3bd6f507df23e8d0692dfed17f98dcb5b661b6",
        "policyId": "0YTxx00000001ovGAA",
        "insurancePolicy": {
          "policyName": "Renewed_Auto_Policy_00002",
          "policyNumber": "Renewed Auto Policy 00002",
          "effectiveFromDate": "2025-01-01",
          "effectiveToDate": "2025-12-31"
        },
        "transactionRecord": {
          "name": "Renew Transaction"
        },
        "additionalInput": [
          {
            "instanceKey": "AutoBundle2",
            "additionalFieldsList": [
              {
                "key": "AssetDescription__c",
                "value": "Test_AssetDescription__c"
              },
              {
                "key": "SourceSystem",
                "value": "Test_SourceSystem"
              }
            ]
          },
          {
            "instanceKey": "AutoDriver",
```

```

    "additionalFieldsList": [
      {
        "key": "ParticipantName",
        "value": "Test_ParticipantName"
      },
      {
        "key": "Participant_Hometown__c",
        "value": "Test_Participant_Hometown__c"
      }
    ]
  },
  {
    "instanceKey": "AutoRoot",
    "additionalFieldsList": [
      {
        "key": "Primary_Policy_Holder__c",
        "value": "Test_Primary_Policy_Holder__c"
      },
      {
        "key": "PolicyDescription",
        "value": "Test_PolicyDescription"
      }
    ]
  },
  {
    "instanceKey": "AutoBundle2_AutoComp1",
    "additionalFieldsList": [
      {
        "key": "Coverage_Description__c",
        "value": "Test_Coverage_Description__c"
      },
      {
        "key": "Description",
        "value": "Test_Description"
      }
    ]
  }
]
}
}
}

```

Sample Response

```
[
  {
    "actionName": "renewInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "renewedPolicyId": "0YTSG000000PgX74AK"
```

```
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Renew Insurance Policies Action

Process the renewal of insurance policies in bulk.
This action is available in API version 64.0 and later.

Supported REST HTTP Methods

- URI**
/services/data/v/actions/standard/renewInsurancePolicies
- Formats**
JSON, XML
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer *token*

Inputs

Input	Details
renewInsurancePoliciesIAInputRep	Type Apex-defined Description Required. A collection of Apex input records that contain details of the insurance policies to be renewed. See ConnectApi.RenewInsurancePoliciesIAInputRep on page 530 for the list of input parameters.

Outputs

No output representation as it's an async invocable action, which publishes the request to a queue for processing.

Example

POST
This example shows a sample request for the Renew Insurance Policies action.

```
{
  "inputs": [
    {
```

```

    "RenewInsurancePoliciesIAInputRep": {
      "policyId": "0YTIU000000xUi14AE",
      "jobIdentifier": "testJob368",
      "executeConfigurationRules": true,
      "executeQualificationRules": true,
      "issueQuote": false,
      "sameTenure": true
      "priceBookId": "01sxx0000005pyfAAA",
      "productSellingModel": "One Time",
      "flowAPINameWithNamespace": "abc",
      "fieldSetIds" : {
        "policyFieldSetId": "0YTIU000000xUi14AE",
        "assetFieldSetId" : "0YTIU000000xUi14AE",
        "participantFieldSetId" : "0YTIU000000xUi14AE",
        "coverageFieldSetId" : "0YTIU000000xUi14AE",
      }
    }
  }
}
]
```

Renew Multi-Root Insurance Policy Action

Renew an existing multi-root insurance policy along with its associated child policies asynchronously.
This action is available in API version 66.0 and later.

Supported REST HTTP Methods

- URI**
/services/data/v/actions/standard/renewMultiRootInsurancePolicy
- Formats**
JSON, XML
- HTTP Methods**
POST
- Authentication**
Authorization: Bearer *token*

Inputs

Input	Details
renewMultiRootPolicyInputRep	Type Apex-defined Description Required. An Apex ConnectApi.RenewMultiRootPolicyInputRep record that contains the details to renew the multi-root insurance policy.

Outputs

Output	Details
jobIdentifier	Type string Description A unique system-generated ID for the job that processes the renewal of the multi-root insurance policy.

Example

POST

This sample request is for the Renew Multi-Root Insurance Policy action.

```
{
  "inputs": [
    {
      "renewMultiRootPolicyInputRep": {
        "contextId": "d767b27ec095f9f95c5a34299e3bd6f507df23e8d0692dfed17f98dcb5b661b6",

        "policyId": "0YTxx00000001ovGAA",
        "insurancePolicy": {
          "policyName": "Renewed_Auto_Policy_00002",
          "policyNumber": "Renewed Auto Policy 00002",
          "effectiveFromDate": "2025-01-01",
          "effectiveToDate": "2025-12-31"
        },
        "transactionRecord": {
          "name": "Renew Transaction"
        },
        "additionalInput": [
          {
            "instanceKey": "AutoBundle2",
            "additionalFieldsList": [
              {
                "key": "AssetDescription__c",
                "value": "Test_AssetDescription__c"
              },
              {
                "key": "SourceSystem",
                "value": "Test_SourceSystem"
              }
            ]
          },
          {
            "instanceKey": "AutoDriver",
            "additionalFieldsList": [
              {
                "key": "ParticipantName",
                "value": "Test_ParticipantName"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

    },
    {
      "key": "Participant_Hometown__c",
      "value": "Test_Participant_Hometown__c"
    }
  ]
},
{
  "instanceKey": "AutoRoot",
  "additionalFieldsList": [
    {
      "key": "Primary_Policy_Holder__c",
      "value": "Test_Primary_Policy_Holder__c"
    },
    {
      "key": "PolicyDescription",
      "value": "Test_PolicyDescription"
    }
  ]
},
{
  "instanceKey": "AutoBundle2_AutoComp1",
  "additionalFieldsList": [
    {
      "key": "Coverage_Description__c",
      "value": "Test_Coverage_Description__c"
    },
    {
      "key": "Description",
      "value": "Test_Description"
    }
  ]
}
]
}
}
}
}
```

This is a sample code to call the Renew Multi-Root Insurance Policy action from Apex.

```
public with sharing class MultiRootPolicyRenewalInvocable {

    /**
     * Execute Multi-Root Policy Renewal with parameters map
     * Usage: MultiRootPolicyRenewalInvocable.executeMultiRootRenewal(paramsMap);
     *
     * @param params Map of parameters:
     *   - contextId (String, REQUIRED)
     *   - policyId (String, REQUIRED)
     *   - policyName (String, optional)
     *   - effectiveFromDate (String, optional)
     *   - effectiveToDate (String, optional)
     *   - transactionName (String, optional)
     *   - additionalFields (Map<String, Map<String, String>>, optional)
     */
}
```

```

*/
public static void executeMultiRootRenewal(Map<String, Object> params) {
    // Extract and validate contextId (REQUIRED)
    String contextId = (String) params.get('contextId');
    if (String.isBlank(contextId)) {
        System.debug('ERROR: contextId is required');
        return;
    }

    // Extract and validate policyId (REQUIRED)
    String policyId = (String) params.get('policyId');
    if (String.isBlank(policyId)) {
        System.debug('ERROR: policyId is required');
        return;
    }

    // Multi-Root Renew Policy IA - Create the standard Invocable Action
    Invocable.Action action =
    Invocable.Action.createStandardAction('renewMultiRootInsurancePolicy');

    // Instantiate the IA input object for Multi-Root Policy Renewal
    // IMPORTANT: Use RenewMultiRootPolicyInputRep
    ConnectApi.RenewMultiRootPolicyInputRep policyInput =
        new ConnectApi.RenewMultiRootPolicyInputRep();

    // Set context ID and policy ID (REQUIRED)
    policyInput.contextId = contextId;
    policyInput.policyId = policyId;

    // Extract optional parameters
    String policyName = (String) params.get('policyName');
    String effectiveFromDate = (String) params.get('effectiveFromDate');
    String effectiveToDate = (String) params.get('effectiveToDate');
    String transactionName = (String) params.get('transactionName');
    Map<String, Map<String, String>> additionalFields =
        (Map<String, Map<String, String>>) params.get('additionalFields');

    // Populate Insurance Policy details (using IA classes) - all optional
    if (!String.isBlank(policyName) || !String.isBlank(effectiveFromDate) ||
        !String.isBlank(effectiveToDate)) {

        ConnectApi.InsPolicyIAInputRep insurancePolicy =
            new ConnectApi.InsPolicyIAInputRep();

        if (!String.isBlank(policyName)) {
            insurancePolicy.policyName = policyName;
        }
        if (!String.isBlank(effectiveFromDate)) {
            insurancePolicy.effectiveFromDate = effectiveFromDate;
        }
        if (!String.isBlank(effectiveToDate)) {
            insurancePolicy.effectiveToDate = effectiveToDate;
        }
    }
}

```



```

        policyInput.insurancePolicy = insurancePolicy;
    }

    // Populate Transaction record (using IA transaction representation) - optional
    if (!String.isBlank(transactionName)) {
        ConnectApi.InsPolicyTrxnIAInputRep transactionRecord =
            new ConnectApi.InsPolicyTrxnIAInputRep();
        transactionRecord.name = transactionName;
        policyInput.transactionRecord = transactionRecord;
    }

    // Populate additional input for multi-root policies - optional
    if (additionalFields != null && !additionalFields.isEmpty()) {
        List<ConnectApi.InsPolicyAddlInputIARep> additionalInputList = new
List<ConnectApi.InsPolicyAddlInputIARep>();

        for (String instanceKey : additionalFields.keySet()) {
            ConnectApi.InsPolicyAddlInputIARep additionalInput =
                new ConnectApi.InsPolicyAddlInputIARep();

            // Set instance key (e.g., "AutoSilver", "FamilyHealth")
            additionalInput.instanceKey = instanceKey;

            // Build additional fields list for this instance
            Map<String, String> fieldsMap = additionalFields.get(instanceKey);
            if (fieldsMap != null && !fieldsMap.isEmpty()) {
                List<ConnectApi.AdditionalFieldIARep> fieldsList = new
List<ConnectApi.AdditionalFieldIARep>();

                for (String fieldKey : fieldsMap.keySet()) {
                    ConnectApi.AdditionalFieldIARep field = new
ConnectApi.AdditionalFieldIARep();
                    field.key = fieldKey;
                    field.value = fieldsMap.get(fieldKey);
                    fieldsList.add(field);
                }

                additionalInput.additionalFieldsList = fieldsList;
            }

            additionalInputList.add(additionalInput);
        }

        policyInput.additionalInput = additionalInputList;
    }

    // Set invocation parameters
    // IMPORTANT: Parameter name is renewMultiRootPolicyInputRep (camelCase)
    action.setInvocationParameter('renewMultiRootPolicyInputRep', policyInput);

    System.debug('Starting Multi-Root Policy Renewal Invocable Action
(renewMultiRootInsurancePolicy)...');
    System.debug('Context ID: ' + contextId);

```

```

        System.debug('Policy ID: ' + policyId);
        if (!String.isBlank(policyName)) {
            System.debug('Policy Name: ' + policyName);
        }

        try {
            // Invoke the action
            List<Invocable.Action.Result> results = action.invoke();

            System.debug('Multi-Root Policy Renewal IA result = ' + results);

            // Process results
            for (Invocable.Action.Result result : results) {
                if (result.isSuccess()) {
                    System.debug('SUCCESS: Multi-Root Policy Renewal completed
successfully!');
                    Object outputValue =
result.getOutputParameters().get('jobIdentifier');
                    if (outputValue != null) {
                        System.debug('Job Identifier: ' + outputValue);
                    } else {
                        System.debug('Multi-Root Policy Renewal job started - result:
' + result.getOutputParameters());
                    }
                } else {
                    System.debug('ERROR: Multi-Root Policy Renewal failed');
                    System.debug('Error Details: ' + result.getErrors());
                }
            }

        } catch (Exception e) {
            System.debug('EXCEPTION in Multi-Root Policy Renewal Invocable Action: ' +
e.getMessage());
            System.debug('Stack Trace: ' + e.getStackTraceString());
        }
    }
}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */
Map<String, Object> params = new Map<String, Object>();
params.put('contextId', <your-context-id-here>); // REQUIRED
params.put('policyId', <your-policy-id-here>); // REQUIRED
params.put('policyName', 'Renewed Multi-Root Policy');
params.put('effectiveFromDate', '2026-01-01');
params.put('effectiveToDate', '2027-01-01');
params.put('transactionName', 'Multi-Root Annual Renewal with Updates');

// Build additional fields configuration
Map<String, Map<String, String>> additionalFields = new Map<String, Map<String,
String>>();

```

```
// Configuration for AutoSilver instance
Map<String, String> autoFields = new Map<String, String>();
autoFields.put('PolicyName', 'James Auto Policy');
autoFields.put('Name', 'James Auto Policy');
additionalFields.put(<your-instanceKey-goes-here>, autoFields);

// Configuration for FamilyHealth instance
Map<String, String> healthFields = new Map<String, String>();
healthFields.put('PolicyName', 'James Medical Policy');
healthFields.put('Name', 'James Medical Policy');
additionalFields.put(<your-instanceKey-goes-here>, healthFields);

params.put('additionalFields', additionalFields);

MultiRootPolicyRenewalInvocable.executeMultiRootRenewal(params);
```

This sample response is for the Renew Multi-Root Insurance Policy action.

```
[
  {
    "actionName": "renewMultiRootInsurancePolicy",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "jobIdentifier": "XYTSG000000PgX74AK"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Insurance Policy Administration Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Policy Administration](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Policy Administration

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Policy Administration exposes additional `actionType` values for the `FlowActionCall` Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	<code>InvokableActionType</code> (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Policy Administration include: • <code>cancelInsurancePolicy</code>—Cancel an insurance policy by using a set of user inputs that represent policy details. • <code>createOutOfSeqPlcyEndorsement</code>—Create an out-of-sequence endorsement for an insurance policy. • <code>endorseInsurancePolicy</code>—Endorse an insurance policy by using a set of user inputs that represent policy details. • <code>endorseMultiRootInsPolicy</code>— • <code>getInsurancePolicy</code>—Get the details of an insurance policy. • <code>issueInsurancePolicy</code>—Issue an insurance policy by using a set of user inputs that represent policy details. • <code>issueMultiRootInsPolicy</code>— • <code>reinstateInsurancePolicy</code>—Reinstate a previously canceled insurance policy. • <code>renewInsurancePolicy</code>—Renew an insurance policy by using a set of user inputs that represent policy details. • <code>renewMultiRootInsPolicy</code>— • <code>renewInsurancePolicies</code>—Renew insurance policies in bulk.

Insurance Policy Standard Objects

The Insurance Policy data model provides objects and fields to manage insurance policy transactions.

[InsProductClauseVariableMap](#)

Represents the mapping between a token defined in an insurance clause and the product attributes that populate the token value at runtime. For example, a Free Look clause allows a customer to cancel a policy within a certain number of days. The number of days is a variable (or, token) that's populated using product-specific attributes. This object is available in API version 65.0 and later.

[InsuranceClause](#)

Defines the terms and conditions associated with a product. For example, an Avocation Clause excludes coverage if the insured engages in a risky hobby. This object is available in API version 65.0 and later.

[InsurancePolicyProductClause](#)

Represents all clauses associated with the products that are applicable to a specific insurance policy. The object contains one record for each clause associated with a product that is applicable to a policy. This object is available in API version 65.0 and later.

[InsuranceProductClause](#)

Represents a junction between a product and an insurance clause. This association defines the clauses applicable for the specific products. For example, an Excluded Driver clause applies to auto and motorcycle insurance products, but not to home insurance products. This object is available in API version 65.0 and later.

[QuoteLineItemInsProductClause](#)

Represents all clauses associated with the products that are applicable to a specific quote line item. For example, if auto insurance quote "ABC" includes both a Free Look clause and an Excluded Driver clause, the object will have two records, one for each clause applicable to the quote line item. This object is available in API version 65.0 and later.

InsProductClauseVariableMap

Represents the mapping between a token defined in an insurance clause and the product attributes that populate the token value at runtime. For example, a Free Look clause allows a customer to cancel a policy within a certain number of days. The number of days is a variable (or, token) that's populated using product-specific attributes. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
Attribute	Type string Properties Create, Filter, Group, Sort, Update Description The name of the context attribute that's mapped to the variable.
InsuranceProductClauseId	Type reference Properties Create, Filter, Group, Sort Description This field is a relationship field. Relationship Name InsuranceProductClause Relationship Type Master-detail Refers To InsuranceProductClause (the master object)

Field	Details
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (LastReferencedDate), but not viewed it.
Name	Type string Properties Autonumber, Defaulted on create, Filter, idLookup, Sort Description The name of the insurance product clause mapping.
VariableDataType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort, Update Description The type of the variable used in the clause template. Possible values are: • boolean • currency • date • datetime • number • percent • picklist • reference • string

Field	Details
VariableName	Type string Properties Create, Filter, Group, Sort, Update Description The name of the variable used in the clause template.

InsuranceClause

Defines the terms and conditions associated with a product. For example, an Avocation Clause excludes coverage if the insured engages in a risky hobby. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ApiName	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The unique name of the insurance clause.
Code	Type string Properties Create, Filter, Group, Sort, Update Description The unique code that represents the insurance clause.
ContentText	Type textarea Properties Create, Nillable, Update

Field	Details
	Description The template, containing variables, used to generate the clause statement.
CreationMethod	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Specifies how the insurance clause is added. Possible values are: • AutoAdded • Manual
EffectiveDate	Type date Properties Create, Filter, Group, Sort, Update Description The date when the insurance clause becomes active.
ExpirationDate	Type date Properties Create, Filter, Group, Sort, Update Description The date when insurance clause becomes inactive.
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort

Field	Details
	Description The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (<code>LastReferencedDate</code>), but not viewed it.
Name	Type string Properties Create, Filter, Group, idLookup, Sort, Update Description The name of the insurance clause.
OwnerId	Type reference Properties Create, Defaulted on create, Filter, Group, Sort, Update Description This field is a polymorphic relationship field. Relationship Name Owner Refers To Group, User
Type	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort, Update Description Specifies the type of insurance clause. Possible values are: • <code>Clause</code> • <code>Exclusion</code>
Variables	Type textarea Properties Create, Nillable, Update Description The variables (tokens) in the <code>ContentText</code> that are replaced with values when generating the clause statement.

InsurancePolicyProductClause

Represents all clauses associated with the products that are applicable to a specific insurance policy. The object contains one record for each clause associated with a product that is applicable to a policy. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ApplicableObjectType	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort Description Specifies the type of object the insurance policy product clause applies to. Possible values are: - InsurancePolicyAsset—Insurance Policy Asset - InsurancePolicyCoverage—Insurance Policy Coverage - InsurancePolicyParticipant—Insurance Policy Participant
ClauseText	Type textarea Properties Create, Nillable, Update Description The generated clause text based on rule evaluation.
ConstraintModelDeveloperName	Type string Properties Create, Filter, Group, Nillable, Sort Description The developer name of the constraint model.
CreationMethod	Type picklist Properties Create, Filter, Group, Nillable, Restricted picklist, Sort

Field	Details
	Description Indicates how the insurance policy product clause is added. Possible values are: - AutoAdded - Manual
Description	Type string Properties Create, Filter, Group, Nillable, Sort Description The description of the insurance policy product clause.
EffectiveDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the insurance policy product clause becomes active.
ExpirationDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the insurance policy product clause becomes inactive.
InsurancePolicyAssetId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the policy asset associated with the insurance policy product clause. This field is a relationship field. Relationship Name InsurancePolicyAsset Refers To InsurancePolicyAsset
InsurancePolicyCoverageId	Type reference

Field	Details
	Properties Create, Filter, Group, Nillable, Sort Description The ID of the policy coverage associated with the insurance policy product clause. This field is a relationship field. Relationship Name InsurancePolicyCoverage Refers To InsurancePolicyCoverage
InsurancePolicyId	Type reference Properties Create, Filter, Group, Sort Description The ID of the policy associated with the insurance policy product clause. This field is a relationship field. Relationship Name InsurancePolicy Relationship Type Master-detail Refers To InsurancePolicy (the master object)
InsurancePolicyParticipantId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The ID of the participant associated with the insurance policy product clause. This field is a relationship field. Relationship Name InsurancePolicyParticipant Refers To InsurancePolicyParticipant
InsuranceProductClauseId	Type reference Properties Create, Filter, Group, Nillable, Sort

Field	Details
	Description The ID of the insurance product clause associated with the insurance policy product clause. This field is a relationship field. Relationship Name InsuranceProductClause Refers To InsuranceProductClause
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (LastReferencedDate), but not viewed it.
Name	Type string Properties Create, Filter, Group, idLookup, Sort Description The name of the insurance policy product clause.
SourceSystem	Type string Properties Create, Filter, Group, Nillable, Sort Description The name of the system from where the insurance policy product clause originated.
SourceSystemIdentifier	Type string

Field	Details
	Properties Create, Filter, Group, Nillable, Sort
	Description The unique identifier of the insurance policy product clause record in the source system.

InsuranceProductClause

Represents a junction between a product and an insurance clause. This association defines the clauses applicable for the specific products. For example, an Excluded Driver clause applies to auto and motorcycle insurance products, but not to home insurance products. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ApiName	Type string
	Properties Create, Filter, Group, Nillable, Sort
	Description The API name of the insurance product clause.
EffectiveDate	Type date
	Properties Create, Filter, Group, Sort, Update
	Description The insurance clause associated with the insurance product clause.
ExpirationDate	Type date
	Properties Create, Filter, Group, Sort, Update
	Description The date when the insurance clause becomes active.

Field	Details
InsuranceClauseId	Type reference Properties Create, Filter, Group, Sort Description This field is a relationship field. Relationship Name InsuranceClause Relationship Type Master-detail Refers To InsuranceClause (the master object)
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (<code>LastReferencedDate</code>), but not viewed it.
Name	Type string Properties Create, Filter, Group, idLookup, Sort Description The name of the insurance product clause.
ProductPath	Type string

Field	Details
	Properties Create, Filter, Group, Sort Description The path that contains the product IDs that roll up to the parent product.
ProductRelatedComponentId	Type reference Properties Create, Filter, Group, Nillable, Sort Description The product-related component associated with the insurance product clause. This field is a relationship field. Relationship Name ProductRelatedComponent Refers To ProductRelatedComponent
RootProductId	Type reference Properties Create, Filter, Group, Sort Description The ID of the parent (root) product that this product rolls up to. This field is a relationship field. Relationship Name RootProduct Refers To Product2
RuleKey	Type string Properties Create, Filter, Nillable, Sort Description The key of the variable defined in CML (Constraint Modeling Language) that represents the constraint for the insurance product clause.

QuoteLineItemInsProductClause

Represents all clauses associated with the products that are applicable to a specific quote line item. For example, if auto insurance quote "ABC" includes both a Free Look clause and an Excluded Driver clause, the object will have two records, one for each clause applicable to the quote line item. This object is available in API version 65.0 and later.

Supported Calls

`create()`, `delete()`, `describeLayout()`, `describeSObjects()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `undelete()`, `update()`, `upsert()`

Fields

Field	Details
ClauseText	Type textarea Properties Create, Update Description The generated clause text based on rule evaluation.
ConstraintModelDeveloperName	Type string Properties Create, Filter, Group, Nillable, Sort Description The developer name of the the constraint model used to generate clauses.
CreationMethod	Type picklist Properties Create, Filter, Group, Restricted picklist, Sort Description Indicates how the quote item insurance product clause is added. Possible values are: • AutoAdded • Manual
Description	Type string Properties Create, Filter, Group, Nillable, Sort Description The description of the quote item insurance product clause.

Field	Details
EffectiveDate	Type date Properties Create, Filter, Group, Sort, Update Description The date when the quote item insurance product clause becomes active.
ExpirationDate	Type date Properties Create, Filter, Group, Nillable, Sort, Update Description The date when the quote item insurance product clause becomes inactive.
InsuranceProductClauseId	Type reference Properties Create, Filter, Group, Sort Description The ID of the insurance product clause associated with the quote item insurance product clause. This field is a relationship field. Relationship Name InsuranceProductClause Refers To InsuranceProductClause
LastReferencedDate	Type dateTime Properties Filter, Nillable, Sort Description The timestamp when the current user last accessed this record indirectly, for example, through a list view or related record.
LastViewedDate	Type dateTime Properties Filter, Nillable, Sort

Field	Details
	Description The timestamp when the current user last viewed this record or list view. If this value is null, the user might have only accessed this record or list view (<code>LastReferencedDate</code>), but not viewed it.
Name	Type string Properties Create, Filter, Group, idLookup, Sort Description The name of the quote item insurance product clause.
QuoteLineItemId	Type reference Properties Create, Filter, Group, Sort Description The quote line item associated with the quote item insurance product clause. This field is a relationship field. Relationship Name QuoteLineItem Relationship Type Master-detail Refers To QuoteLineItem (the master object)

Insurance Policy Administration Apex Reference

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Salesforce Platform server, in conjunction with calls to the API. Insurance Policy Administration Apex reference includes built-in Apex classes and interfaces, grouped by namespace.

[ConnectApi Namespace](#)

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Policy Administration provides classes for accessing the same data available in the Insurance Policy Administration Business APIs.

ConnectApi Namespace

The `ConnectApi` namespace (also called Connect in Apex) in Insurance Policy Administration provides classes for accessing the same data available in the Insurance Policy Administration Business APIs.

For information about working with the `ConnectApi` classes, see [Connect in Apex](#).

[InsurancePolicyAdminFamily Class](#)

Manage insurance policies comprehensively, for different transaction types, such as issuance, endorsement, renewal, and cancellation.

[ConnectApi Input Classes](#)

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

[ConnectApi Output Classes](#)

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

InsurancePolicyAdminFamily Class

Manage insurance policies comprehensively, for different transaction types, such as issuance, endorsement, renewal, and cancellation.

Namespace

`ConnectApi`

[InsurancePolicyAdminFamily Methods](#)

These methods are for `InsurancePolicyAdminFamily`. All methods are static.

InsurancePolicyAdminFamily Methods

These methods are for `InsurancePolicyAdminFamily`. All methods are static.

[CancelPolicy\(CancelPolicyInput, policyId\)](#)

Cancel an insurance policy version. The API takes the effective date of cancellation from the input JSON. You can advance the cancellation date within the policy term, but you can't postpone the cancellation.

[EndorsePolicy\(EndorsePolicyInput, policyId\)](#)

Endorse the latest version of the insurance policy based on the context ID and effective date. You can also include other optional fields and additional details in the input JSON.

[GetPolicy\(policyId, connectDynamicParameters\)](#)

Get details of an insurance policy. Pass additional context or metadata to customize the information that's fetched.

[IssuePolicy\(IssuePolicyInput\)](#)

Use the context ID provided in the input JSON to generate a new insurance policy.

[MultiRootEndorsePolicy\(EndorsePolicyInput, policyId\)](#)

Resource for the Insurance Multi Root Endorsement Policy Connect API POST Method.

[MultiRootIssuePolicy\(IssuePolicyInput\)](#)

Asynchronously create a multi-root insurance policy along with its child policies.

[MultiRootRenewPolicy\(RenewPolicyInput, policyId\)](#)

Asynchronously renew a multi-root insurance policy, create renewed records for the parent policy and the associated child policies. Use the priced context ID for the renewal process.

[OutOfSequenceEndorsePolicy\(OutOfSequenceEndorsePolicyInput, policyId\)](#)

Create an out-of-sequence endorsement for an insurance policy.

[ReinstatePolicy\(ReinstatePolicyInput, policyId\)](#)

Reinstate a previously canceled insurance policy.

[RenewPolicy\(RenewPolicyInput, policyId\)](#)

Use the context ID of an existing policy to renew a policy. Modify the renewal date, and add optional fields and additional information in the input JSON.

CancelPolicy(CancelPolicyInput, policyId)

Cancel an insurance policy version. The API takes the effective date of cancellation from the input JSON. You can advance the cancellation date within the policy term, but you can't postpone the cancellation.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.CancelPolicyOutputRepresentation  
CancelPolicy(ConnectApi.CancelPolicyInputRepresentation CancelPolicyInput, String  
policyId)
```

```
ConnectApi.InsurancePolicyAdminFamily, CancelPolicy,  
[ConnectApi.CancelPolicyInputRepresentation, String],  
ConnectApi.CancelPolicyOutputRepresentation
```

Parameters

CancelPolicyInput

Type: [ConnectApi.CancelPolicyInputRepresentation](#) on page 522

Description

policyId

Type: String

Input representation of the request to cancel a policy.

Return Value

Type: [ConnectApi.CancelPolicyOutputRepresentation](#) on page 531

Usage

Send the cancellation effective date and transaction record in the request to cancel a policy.

EndorsePolicy(EndorsePolicyInput, policyId)

Endorse the latest version of the insurance policy based on the context ID and effective date. You can also include other optional fields and additional details in the input JSON.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.PolicyOutputRepresentation  
EndorsePolicy(ConnectApi.EndorsePolicyInputRepresentation EndorsePolicyInput, String  
policyId)
```

```
ConnectApi.InsurancePolicyAdminFamily, EndorsePolicy,  
[ConnectApi.EndorsePolicyInputRepresentation, String], ConnectApi.PolicyOutputRepresentation
```

Parameters

EndorsePolicyInput

Type: [ConnectApi.EndorsePolicyInputRepresentation](#) on page 523

Input representation of the request to endorse a policy.

policyId

Type: String

ID of the policy to be endorsed.

Return Value

Type: [ConnectApi.PolicyOutputRepresentation](#) on page 533

GetPolicy(policyId, connectDynamicParameters)

Get details of an insurance policy. Pass additional context or metadata to customize the information that's fetched.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.GetPolicyOutputRepresentation GetPolicy(String policyId,
Map<String,String> connectDynamicParameters)
```

```
ConnectApi.InsurancePolicyAdminFamily, GetPolicy, [String, Map<String,String>],
ConnectApi.GetPolicyOutputRepresentation
```

Parameters

policyId

Type: String

ID of the policy for which the details are to be fetched.

connectDynamicParameters

Type: Map<String,String>

Map of connect dynamic parameters to fetch the policy details.

Return Value

Type: [ConnectApi.GetPolicyOutputRepresentation](#) on page 532

Usage

Use this API to fetch the insurance policy details. This includes related assets, participants, and coverages. Use query parameters to request additional information.

- When the `contextOnly` parameter is true, the API returns a context ID along with the insurance policy details.
- When you send a request with additional fields with the operator `$in`, it returns a list of field-set IDs. When the operator is `$eq`, the request returns a single field-set ID.

For examples, see [Get Policy](#).

IssuePolicy(IssuePolicyInput)

Use the context ID provided in the input JSON to generate a new insurance policy.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.PolicyOutputRepresentation
IssuePolicy(ConnectApi.IssuePolicyInputRepresentation IssuePolicyInput)
```

```
ConnectApi.InsurancePolicyAdminFamily, IssuePolicy,
[ConnectApi.IssuePolicyInputRepresentation], ConnectApi.PolicyOutputRepresentation
```

Parameters

*IssuePolicyInput*Type: [ConnectApi.IssuePolicyInputRepresentation](#)

Input representation of the request to issue an insurance policy.

Return Value

Type: [ConnectApi.PolicyOutputRepresentation](#) on page 533**MultiRootEndorsePolicy (EndorsePolicyInput, policyId)**

Resource for the Insurance Multi Root Endorsement Policy Connect API POST Method.

API Version

66.0

Requires Chatter

No

Signature

```
public static ConnectApi.PolicyAsyncRepresentation
MultiRootEndorsePolicy (ConnectApi.EndorsePolicyInputRepresentation EndorsePolicyInput,
String policyId)
```

Parameters

*EndorsePolicyInput*Type: [ConnectApi.EndorsePolicyInputRepresentation](#)

Details to endorse the multi-root insurance policy along with its child policies.

policyId

Type: String

ID of the insurance policy that must be endorsed along with its child policies.

Return Value

Type: [ConnectApi.PolicyAsyncRepresentation](#) on page 532

Example

This is a sample code that calls the Multi-Root Policy Endorsement API from Apex.

```
public with sharing class MultiRootPolicyEndorsement implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
    }
}
```



```

Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

// Check the action type
if (action == 'multiRootEndorsePolicy') {
    System.debug('Start of connect API call to multi-root endorse policy');

    // Get input parameters from the input map
    String contextId = (String) inputMap.get('contextId');
    String policyId = (String) inputMap.get('policyId');
    String policyName = (String) inputMap.get('policyName');
    String policyNumber = (String) inputMap.get('policyNumber');
    String effectiveDate = (String) inputMap.get('effectiveDate');
    String transactionName = (String) inputMap.get('transactionName');
    Map<String, Object> additionalFields = (Map<String, Object>)
inputMap.get('additionalFields');

    System.debug('Context ID: ' + contextId);
    System.debug('Policy ID: ' + policyId);
    System.debug('Policy Name: ' + policyName);
    System.debug('Policy Number: ' + policyNumber);
    System.debug('Effective Date: ' + effectiveDate);
    System.debug('Transaction Name: ' + transactionName);
    System.debug('Additional Fields: ' + additionalFields);

    // Initialize the job identifier
    String jobIdentifier = null;

    // Validate required parameters
    if (String.isBlank(contextId)) {
        System.debug('ERROR: Context ID is required');
        outputMap.put('error', 'Context ID is required');
        return null;
    }

    try {

        // Step 1: Create Multi-Root Policy Endorsement input using the provided
context ID

        ConnectApi.EndorsePolicyInputRepresentation policyInput =
            new ConnectApi.EndorsePolicyInputRepresentation();

        // Populate fields for InsurancePolicyInputRepresentation
        ConnectApi.InsurancePolicyInputRepresentation insurancePolicy =
            new ConnectApi.InsurancePolicyInputRepresentation();

        // Set policy basic fields (all optional except contextId)
        if (!String.isBlank(policyName)) {
            insurancePolicy.policyName = policyName;
        }
        if (!String.isBlank(policyNumber)) {
            insurancePolicy.policyNumber = policyNumber;
        }
    }
}

```

```

        policyInput.insurancePolicy = insurancePolicy;

        // Populate transaction record (optional)
        if (!String.isBlank(transactionName)) {
            ConnectApi.InsurancePolicyTransactionInputRepresentation
transactionRecord =
                new ConnectApi.InsurancePolicyTransactionInputRepresentation();
            transactionRecord.name = transactionName;
            policyInput.transactionRecord = transactionRecord;
        }

        // Use the provided context ID (REQUIRED)
        policyInput.contextId = contextId;

        // Use the provided Effective Date (REQUIRED)
        policyInput.effectiveDate = effectiveDate;

        // Populate additional fields for child policies
        if (additionalFields != null && !additionalFields.isEmpty()) {
            System.debug('Processing additional fields for child policies');
            policyInput.additionalFields =
buildAdditionalFieldsList(additionalFields);
        }

        // Step 2: Execute Multi-Root Policy Endorsement with the provided context
ID
        System.debug('Executing Multi-Root Policy Endorsement with context ID: '
+ contextId + ' policy ID:' + policyId);

        Id idPolicyId = Id.valueOf(policyId);

        // Call the Connect API - Multi-Root Endorsement is ASYNCHRONOUS and returns
a job identifier
        ConnectApi.PolicyAsyncRepresentation endorseResult =
            ConnectApi.InsurancePolicyAdminFamily.multiRootEndorsePolicy(policyInput,
idPolicyId);

        // Get the job identifier
        jobIdentifier = endorseResult.jobIdentifier;

        // Log success
        System.debug('Multi-root policy endorsement job started successfully. Job
ID: ' + jobIdentifier);
        System.debug('Context: ' + contextId);
        if (!String.isBlank(policyName)) {
            System.debug('Policy Name: ' + policyName);
        }

        // Put the job identifier in the output map
        outputMap.put('jobIdentifier', jobIdentifier);
        outputMap.put('endorsementResult', endorseResult);

    } catch (Exception e) {
        System.debug('Error while executing multi-root policy endorsement: ' +

```

```

e.getMessage());
        System.debug('Stack Trace: ' + e.getStackTraceString());
        outputMap.put('error', e.getMessage());
    }

    // Return the job identifier
    return jobIdIdentifier;
}

// If the action is not recognized, return null
return null;
}

/**
 * Build additional fields list for multi-root policies
 * Converts from Map structure to List<ConnectApi.AdditionalFieldsMapValue>
 *
 * Input structure:
 * {
 *     "additionalFieldsList": [
 *         {
 *             "additionalFieldsMapValue": {
 *                 "instanceKey": "AutoSilver",
 *                 "fields": { "PolicyName": "James Auto Policy" }
 *             }
 *         }
 *     ]
 * }
 */
private static List<ConnectApi.AdditionalFieldsMapValue>
buildAdditionalFieldsList(Map<String, Object> additionalFieldsInput) {
    List<ConnectApi.AdditionalFieldsMapValue> fieldsList = new
List<ConnectApi.AdditionalFieldsMapValue>();

    // Get the additionalFieldsList array
    List<Object> additionalFieldsList = (List<Object>)
additionalFieldsInput.get('additionalFieldsList');

    if (additionalFieldsList != null && !additionalFieldsList.isEmpty()) {
        for (Object item : additionalFieldsList) {
            Map<String, Object> itemMap = (Map<String, Object>) item;
            Object mapValue = itemMap.get('additionalFieldsMapValue');

            if (mapValue != null) {
                // Create ConnectApi.AdditionalFieldsMapValue instance
                ConnectApi.AdditionalFieldsMapValue fieldMapValue = new
ConnectApi.AdditionalFieldsMapValue();

                // Set the additionalFieldsMapValue property
                // This property accepts Object type which will be the Map containing
                instanceKey and fields
                fieldMapValue.additionalFieldsMapValue = mapValue;
            }
        }
    }
}

```

```

        fieldsList.add(fieldMapValue);
    }
}

return fieldsList;
}
}

/*
 * USAGE EXAMPLE:
 * Multi-Root Policy Endorsement with Additional Fields
 */

// Create instance of the callable class
MultiRootPolicyEndorsement callable = new MultiRootPolicyEndorsement();

// Prepare the input arguments with additional fields
Map<String, Object> input = new Map<String, Object>();
input.put('contextId', '0000000i18tq18g002917629263342460535451b25974898aae9a44aab48ef3c');
// REQUIRED - Replace with your context ID
input.put('policyId', '0YTSM0000001MrJ4AU'); // REQUIRED - Replace with your policy ID
input.put('policyName', 'James Multi-Root Policy');
input.put('policyNumber', 'James Multi-Root Policy');
input.put('referencePolicyNumber', 'James1234');
input.put('effectiveDate', '2025-08-30'); // REQUIRED
input.put('transactionName', 'Endorsed Policy Transaction');

// Build additional fields for child policies
// Structure matches the JSON payload exactly
Map<String, Object> additionalFields = new Map<String, Object>();
List<Object> additionalFieldsList = new List<Object>();

// Child Policy 1: Auto Insurance
Map<String, Object> autoPolicy = new Map<String, Object>();
Map<String, Object> autoMapValue = new Map<String, Object>();
autoMapValue.put('instanceKey', 'AutoSilver');
Map<String, Object> autoFields = new Map<String, Object>();
autoFields.put('PolicyName', 'James Auto Policy');
autoFields.put('Name', 'James Auto Policy');
autoMapValue.put('fields', autoFields);
autoPolicy.put('additionalFieldsMapValue', autoMapValue);
additionalFieldsList.add(autoPolicy);

// Child Policy 2: Health Insurance
Map<String, Object> healthPolicy = new Map<String, Object>();
Map<String, Object> healthMapValue = new Map<String, Object>();
healthMapValue.put('instanceKey', 'FamilyHealth');

Map<String, Object> healthFields = new Map<String, Object>();
healthFields.put('PolicyName', 'James Medical Policy');
healthFields.put('Name', 'James Medical Policy');
healthMapValue.put('fields', healthFields);

```

```

healthPolicy.put('additionalFieldsMapValue', healthMapValue);
additionalFieldsList.add(healthPolicy);

additionalFields.put('additionalFieldsList', additionalFieldsList);
input.put('additionalFields', additionalFields);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'multiRootEndorsePolicy' action
Object result = callable.call('multiRootEndorsePolicy', args);

// Check results
System.debug('Job Identifier: ' + result);
if (output.containsKey('error')) {
    System.debug('ERROR: ' + output.get('error'));
} else if (output.containsKey('endorsementResult')) {
    System.debug('SUCCESS: Multi-root policy endorsement job started with ID: ' +
output.get('jobIdentifier'));
    System.debug('This job will endorse the input policy and apply the changes using the
context, with additional fields on parent and child policies.');
```

MultiRootIssuePolicy (IssuePolicyInput)

Asynchronously create a multi-root insurance policy along with its child policies.

API Version

66.0

Requires Chatter

No

Signature

```

public static ConnectApi.PolicyAsyncRepresentation
MultiRootIssuePolicy(ConnectApi.IssuePolicyInputRepresentation IssuePolicyInput)
```

Parameters

IssuePolicyInput

Type: [ConnectApi.IssuePolicyInputRepresentation](#)

Details to create the multi-root insurance policy.

Return Value

Type: [ConnectApi.PolicyAsyncRepresentation](#) on page 532

Example

This is a sample code that calls the Multi-Root Issue Policy API from Apex.

```
public with sharing class MultiRootPolicyIssuance implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'multiRootIssuePolicy') {
            System.debug('Start of connect API call to multi-root issue policy');

            // Get input parameters from the input map
            String contextId = (String) inputMap.get('contextId');
            String policyName = (String) inputMap.get('policyName');
            String policyNumber = (String) inputMap.get('policyNumber');
            String referencePolicyNumber = (String) inputMap.get('referencePolicyNumber');

            String effectiveFromDate = (String) inputMap.get('effectiveFromDate');
            String effectiveToDate = (String) inputMap.get('effectiveToDate');
            String transactionName = (String) inputMap.get('transactionName');
            Map<String, Object> additionalFields = (Map<String, Object>)
inputMap.get('additionalFields');

            System.debug('Context ID: ' + contextId);
            System.debug('Policy Name: ' + policyName);
            System.debug('Policy Number: ' + policyNumber);
            System.debug('Reference Policy Number: ' + referencePolicyNumber);
            System.debug('Effective From Date: ' + effectiveFromDate);
            System.debug('Effective To Date: ' + effectiveToDate);
            System.debug('Transaction Name: ' + transactionName);
            System.debug('Additional Fields: ' + additionalFields);

            // Initialize the job identifier
            String jobIdentifier = null;

            // Validate required parameters
            if (String.isBlank(contextId)) {
                System.debug('ERROR: Context ID is required');
                outputMap.put('error', 'Context ID is required');
                return null;
            }

            try {

                // Step 1: Create Multi-Root Policy Issuance input using the provided
context ID
```

```

ConnectApi.IssuePolicyInputRepresentation policyInput =
    new ConnectApi.IssuePolicyInputRepresentation();

// Populate fields for InsurancePolicyInputRepresentation
ConnectApi.InsurancePolicyInputRepresentation insurancePolicy =
    new ConnectApi.InsurancePolicyInputRepresentation();

// Set policy basic fields (all optional except contextId)
if (!String.isBlank(policyName)) {
    insurancePolicy.policyName = policyName;
}
if (!String.isBlank(policyNumber)) {
    insurancePolicy.policyNumber = policyNumber;
}
if (!String.isBlank(referencePolicyNumber)) {
    insurancePolicy.referencePolicyNumber = referencePolicyNumber;
}
if (!String.isBlank(effectiveFromDate)) {
    insurancePolicy.effectiveFromDate = effectiveFromDate;
}
if (!String.isBlank(effectiveToDate)) {
    insurancePolicy.effectiveToDate = effectiveToDate;
}

policyInput.insurancePolicy = insurancePolicy;

// Populate transaction record (optional)
if (!String.isBlank(transactionName)) {
    ConnectApi.InsurancePolicyTransactionInputRepresentation
transactionRecord =
        new ConnectApi.InsurancePolicyTransactionInputRepresentation();
    transactionRecord.name = transactionName;
    policyInput.transactionRecord = transactionRecord;
}

// Use the provided context ID (REQUIRED)
policyInput.contextId = contextId;

// Populate additional fields for child policies
if (additionalFields != null && !additionalFields.isEmpty()) {
    System.debug('Processing additional fields for child policies');
    policyInput.additionalFields =
buildAdditionalFieldsList(additionalFields);
}

// Step 2: Execute Multi-Root Policy Issuance with the provided context
ID
System.debug('Executing Multi-Root Policy Issuance with context ID: ' +
contextId);

// Call the Connect API - Multi-Root Issuance is ASYNCHRONOUS and returns
a job identifier
ConnectApi.PolicyAsyncRepresentation issuanceResult =
    ConnectApi.InsurancePolicyAdminFamily.multiRootIssuePolicy(policyInput);

```

```

        // Get the job identifier
        jobIdentifier = issuanceResult.jobIdentifier;

        // Log success
        System.debug('Multi-root policy issuance job started successfully. Job ID: ' + jobIdentifier);
        System.debug('Context: ' + contextId);
        if (!String.isBlank(policyName)) {
            System.debug('Policy Name: ' + policyName);
        }

        // Put the job identifier in the output map
        outputMap.put('jobIdentifier', jobIdentifier);
        outputMap.put('issuanceResult', issuanceResult);

    } catch (Exception e) {
        System.debug('Error while executing multi-root policy issuance: ' + e.getMessage());
        System.debug('Stack Trace: ' + e.getStackTraceString());
        outputMap.put('error', e.getMessage());
    }

    // Return the job identifier
    return jobIdentifier;
}

// If the action is not recognized, return null
return null;
}

/**
 * Build additional fields list for multi-root policies
 * Converts from Map structure to List<ConnectApi.AdditionalFieldsMapValue>
 *
 * Input structure:
 * {
 *   "additionalFieldsList": [
 *     {
 *       "additionalFieldsMapValue": {
 *         "instanceKey": "AutoSilver",
 *         "fields": { "PolicyName": "James Auto Policy" }
 *       }
 *     }
 *   ]
 * }
 */
private static List<ConnectApi.AdditionalFieldsMapValue>
buildAdditionalFieldsList(Map<String, Object> additionalFieldsInput) {
    List<ConnectApi.AdditionalFieldsMapValue> fieldsList = new
List<ConnectApi.AdditionalFieldsMapValue>();

    // Get the additionalFieldsList array

```



```

        List<Object> additionalFieldsList = (List<Object>)
additionalFieldsInput.get('additionalFieldsList');

        if (additionalFieldsList != null && !additionalFieldsList.isEmpty()) {
            for (Object item : additionalFieldsList) {
                Map<String, Object> itemMap = (Map<String, Object>) item;
                Object mapValue = itemMap.get('additionalFieldsMapValue');

                if (mapValue != null) {
                    // Create ConnectApi.AdditionalFieldsMapValue instance
                    ConnectApi.AdditionalFieldsMapValue fieldMapValue = new
ConnectApi.AdditionalFieldsMapValue();

                    // Set the additionalFieldsMapValue property
                    // This property accepts Object type which will be the Map containing
instanceKey and fields
                    fieldMapValue.additionalFieldsMapValue = mapValue;

                    fieldsList.add(fieldMapValue);
                }
            }
        }

        return fieldsList;
    }
}

/*
 * USAGE EXAMPLE:
 * Multi-Root Policy Issuance with Additional Fields
 */

// Create instance of the callable class
MultiRootPolicyIssuance callable = new MultiRootPolicyIssuance();

// Prepare the input arguments with additional fields
Map<String, Object> input = new Map<String, Object>();
input.put('contextId', <your-context-id-here>); // REQUIRED - Replace with your context
ID
input.put('policyName', 'James Multi-Root Policy');
input.put('policyNumber', 'James Multi-Root Policy');
input.put('referencePolicyNumber', 'James1234');
input.put('effectiveFromDate', '2026-01-01');
input.put('effectiveToDate', '2026-10-31');
input.put('transactionName', 'Sold Policy Transaction');

// Build additional fields for child policies
// Structure matches the JSON payload exactly
Map<String, Object> additionalFields = new Map<String, Object>();
List<Object> additionalFieldsList = new List<Object>();

// Child Policy 1: Auto Insurance
Map<String, Object> autoPolicy = new Map<String, Object>();
Map<String, Object> autoMapValue = new Map<String, Object>();

```

```

autoMapValue.put('instanceKey', <instance-key>);
Map<String, Object> autoFields = new Map<String, Object>();
autoFields.put('PolicyName', 'James Auto Policy');
autoFields.put('Name', 'James Auto Policy');
autoMapValue.put('fields', autoFields);
autoPolicy.put('additionalFieldsMapValue', autoMapValue);
additionalFieldsList.add(autoPolicy);

// Child Policy 2: Health Insurance
Map<String, Object> healthPolicy = new Map<String, Object>();
Map<String, Object> healthMapValue = new Map<String, Object>();
healthMapValue.put('instanceKey', <instance-key>);

Map<String, Object> healthFields = new Map<String, Object>();
healthFields.put('PolicyName', 'James Medical Policy');
healthFields.put('Name', 'James Medical Policy');
healthMapValue.put('fields', healthFields);
healthPolicy.put('additionalFieldsMapValue', healthMapValue);
additionalFieldsList.add(healthPolicy);

additionalFields.put('additionalFieldsList', additionalFieldsList);
input.put('additionalFields', additionalFields);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'multiRootIssuePolicy' action
Object result = callable.call('multiRootIssuePolicy', args);

// Check results
System.debug('Job Identifier: ' + result);
if (output.containsKey('error')) {
    System.debug('ERROR: ' + output.get('error'));
} else if (output.containsKey('jobIdentifier')) {
    System.debug('SUCCESS: Multi-root policy issuance job started with ID: ' +
output.get('jobIdentifier'));
    System.debug('This job will create a parent policy and child policies for Auto and
Health insurance.');
```

MultiRootRenewPolicy(RenewPolicyInput, policyId)

Asynchronously renew a multi-root insurance policy, create renewed records for the parent policy and the associated child policies. Use the priced context ID for the renewal process.

API Version

66.0

Requires Chatter

No

Signature

```
public static ConnectApi.PolicyAsyncRepresentation
MultiRootRenewPolicy (ConnectApi.RenewPolicyInputRepresentation RenewPolicyInput, String
policyId)
```

Parameters

RenewPolicyInput

Type: [ConnectApi.RenewPolicyInputRepresentation](#) on page 529

Input representation of the request to renew an existing multi-root policy.

policyId

Type: String

ID of the multi-root policy that must be renewed along with its child policies.

Return Value

Type: [ConnectApi.PolicyAsyncRepresentation](#) on page 532

Example

This is a sample code that calls the Multi-Root Policy Renewal API from Apex.

```
public with sharing class MultiRootPolicyRenewal implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'multiRootRenewPolicy') {
            System.debug('Start of connect API call to multi-root renew policy');

            // Get input parameters from the input map
            String contextId = (String) inputMap.get('contextId');
            String policyId = (String) inputMap.get('policyId');
            String policyName = (String) inputMap.get('policyName');
            String policyNumber = (String) inputMap.get('policyNumber');
            String effectiveFromDate = (String) inputMap.get('effectiveFromDate');
            String effectiveToDate = (String) inputMap.get('effectiveToDate');
            String transactionName = (String) inputMap.get('transactionName');
            Map<String, Object> additionalFields = (Map<String, Object>)
inputMap.get('additionalFields');

            System.debug('Context ID: ' + contextId);
            System.debug('Policy ID: ' + policyId);
```

```

System.debug('Policy Name: ' + policyName);
System.debug('Policy Number: ' + policyNumber);
System.debug('Effective From Date: ' + effectiveFromDate);
System.debug('Effective To Date: ' + effectiveToDate);
System.debug('Transaction Name: ' + transactionName);
System.debug('Additional Fields: ' + additionalFields);

// Initialize the job identifier
String jobIdentifier = null;

// Validate required parameters
if (String.isBlank(contextId)) {
    System.debug('ERROR: Context ID is required');
    outputMap.put('error', 'Context ID is required');
    return null;
}

if (String.isBlank(policyId)) {
    System.debug('ERROR: Policy ID is required');
    outputMap.put('error', 'Policy ID is required');
    return null;
}

try {
    // Step 1: Create Multi-Root Policy Renewal input using the provided context
    ID ConnectApi.RenewPolicyInputRepresentation renewalInput =
        new ConnectApi.RenewPolicyInputRepresentation();

    // Use the provided context ID (REQUIRED)
    renewalInput.contextId = contextId;

    // Populate fields for InsurancePolicyInputRepresentation (nested object)

    ConnectApi.InsurancePolicyInputRepresentation insurancePolicy =
        new ConnectApi.InsurancePolicyInputRepresentation();

    // Set policy basic fields (all optional)
    if (!String.isBlank(policyName)) {
        insurancePolicy.policyName = policyName;
    }
    if (!String.isBlank(policyNumber)) {
        insurancePolicy.policyNumber = policyNumber;
    }

    // Set effective dates on the insurancePolicy object (not on renewalInput
    directly)
    if (!String.isBlank(effectiveFromDate)) {
        insurancePolicy.effectiveFromDate = effectiveFromDate;
    }
    if (!String.isBlank(effectiveToDate)) {
        insurancePolicy.effectiveToDate = effectiveToDate;
    }
}

```

```

        // Only set insurancePolicy if any fields were set
        if (!String.isBlank(policyName) || !String.isBlank(policyNumber) ||
            !String.isBlank(effectiveFromDate) || !String.isBlank(effectiveToDate))
        {
            renewalInput.insurancePolicy = insurancePolicy;
        }

        // Populate transaction record (optional)
        if (!String.isBlank(transactionName)) {
            ConnectApi.InsurancePolicyTransactionInputRepresentation
transactionRecord =
                new ConnectApi.InsurancePolicyTransactionInputRepresentation();
            transactionRecord.name = transactionName;
            renewalInput.transactionRecord = transactionRecord;
        }

        // Populate additional fields for child policies
        if (additionalFields != null && !additionalFields.isEmpty()) {
            System.debug('Processing additional fields for child policies');
            renewalInput.additionalFields =
buildAdditionalFieldsList(additionalFields);
        }

        // Step 2: Execute Multi-Root Policy Renewal with the provided context ID
        and policy ID
        System.debug('Executing Multi-Root Policy Renewal with context ID: ' +
contextId + ' and policy ID: ' + policyId);

        // Call the Connect API - Multi-Root Renewal is ASYNCHRONOUS and returns
a job identifier
        // NOTE: Parameter order is INPUT first, then policyId
        ConnectApi.PolicyAsyncRepresentation renewalResult =
            ConnectApi.InsurancePolicyAdminFamily.multiRootRenewPolicy(renewalInput,
policyId);

        // Get the job identifier
        jobIdIdentifier = renewalResult.jobIdentifier;

        // Log success
        System.debug('Multi-root policy renewal job started successfully. Job ID:
' + jobIdIdentifier);
        System.debug('Context: ' + contextId);
        System.debug('Policy ID: ' + policyId);

        // Put the job identifier in the output map
        outputMap.put('jobIdentifier', jobIdIdentifier);
        outputMap.put('renewalResult', renewalResult);

    } catch (Exception e) {
        System.debug('Error while executing multi-root policy renewal: ' +
e.getMessage());
        System.debug('Stack Trace: ' + e.getStackTraceString());
        outputMap.put('error', e.getMessage());
    }

```

```

    }

    // Return the job identifier
    return jobIdIdentifier;
}

// If the action is not recognized, return null
return null;
}

/**
 * Build additional fields list for multi-root policies
 * Converts from Map structure to List<ConnectApi.AdditionalFieldsMapValue>
 *
 * Input structure:
 * {
 *   "additionalFieldsList": [
 *     {
 *       "additionalFieldsMapValue": {
 *         "instanceKey": "AutoSilver",
 *         "fields": { "EffectiveFromDate": "2025-01-01", "EffectiveToDate":
"2026-01-01"}
 *       }
 *     }
 *   ]
 * }
 */
private static List<ConnectApi.AdditionalFieldsMapValue>
buildAdditionalFieldsList(Map<String, Object> additionalFieldsInput) {
    List<ConnectApi.AdditionalFieldsMapValue> fieldsList = new
List<ConnectApi.AdditionalFieldsMapValue>();

    // Get the additionalFieldsList array
    List<Object> additionalFieldsList = (List<Object>)
additionalFieldsInput.get('additionalFieldsList');

    if (additionalFieldsList != null && !additionalFieldsList.isEmpty()) {
        for (Object item : additionalFieldsList) {
            Map<String, Object> itemMap = (Map<String, Object>) item;
            Object mapValue = itemMap.get('additionalFieldsMapValue');

            if (mapValue != null) {
                // Create ConnectApi.AdditionalFieldsMapValue instance
                ConnectApi.AdditionalFieldsMapValue fieldMapValue = new
ConnectApi.AdditionalFieldsMapValue();

                // Set the additionalFieldsMapValue property
                // This property accepts Object type which will be the Map containing
instanceKey and fields
                fieldMapValue.additionalFieldsMapValue = mapValue;

                fieldsList.add(fieldMapValue);
            }
        }
    }
}

```

```

        }
    }

    return fieldsList;
}
}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */

// Create instance of the callable class
MultiRootPolicyRenewal callable = new MultiRootPolicyRenewal();

// Build additional fields for child policies
// Structure matches the JSON payload exactly
Map<String, Object> additionalFields = new Map<String, Object>();
List<Object> additionalFieldsList = new List<Object>();

// Child Policy 1: Auto Insurance
Map<String, Object> autoPolicy = new Map<String, Object>();
Map<String, Object> autoMapValue = new Map<String, Object>();
autoMapValue.put('instanceKey', 'AutoSilver');
Map<String, Object> autoFields = new Map<String, Object>();
autoFields.put('PolicyName', 'James Auto Policy');
autoFields.put('Name', 'James Auto Policy');
autoMapValue.put('fields', autoFields);
autoPolicy.put('additionalFieldsMapValue', autoMapValue);
additionalFieldsList.add(autoPolicy);

// Child Policy 2: Health Insurance
Map<String, Object> healthPolicy = new Map<String, Object>();
Map<String, Object> healthMapValue = new Map<String, Object>();
healthMapValue.put('instanceKey', 'FamilyHealth');

Map<String, Object> healthFields = new Map<String, Object>();
healthFields.put('PolicyName', 'James Medical Policy');
healthFields.put('Name', 'James Medical Policy');
healthMapValue.put('fields', healthFields);
healthPolicy.put('additionalFieldsMapValue', healthMapValue);
additionalFieldsList.add(healthPolicy);

additionalFields.put('additionalFieldsList', additionalFieldsList);
input.put('additionalFields', additionalFields);

// Prepare the input arguments - WITH ADDITIONAL FIELDS
Map<String, Object> input = new Map<String, Object>();
input.put('contextId', <your-context-id-here>); // REQUIRED
input.put('policyId', <your-policy-id-here>); // REQUIRED
input.put('effectiveFromDate', '2026-01-01'); // Optional
input.put('effectiveToDate', '2027-01-01'); // Optional
input.put('policyName', 'Renewed Multi-Root Policy');
input.put('transactionName', 'Multi-Root Annual Renewal with Updates');

```

```

input.put('additionalFields', additionalFields);

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'multiRootRenewPolicy' action
Object result = callable.call('multiRootRenewPolicy', args);

// Check results
if (output.containsKey('error')) {
    System.debug('ERROR: ' + output.get('error'));
} else if (output.containsKey('jobIdentifier')) {
    String jobId = (String) output.get('jobIdentifier');
    System.debug('SUCCESS: Multi-root renewal job started with Job ID: ' + jobId);
    System.debug('Use this job ID to track the renewal progress');
}

```

OutOfSequenceEndorsePolicy (OutOfSequenceEndorsePolicyInput, policyId)

Create an out-of-sequence endorsement for an insurance policy.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.OutOfSequenceEndorsePolicyOutputRepresentation
OutOfSequenceEndorsePolicy(ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation
OutOfSequenceEndorsePolicyInput, String policyId)

```

Parameters

OutOfSequenceEndorsePolicyInput

Type: [ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation](#)

Details to create the out-of-sequence endorsement.

policyId

Type: String

ID of the insurance policy to create the out-of-sequence endorsement for.

Return Value

Type: [ConnectApi.OutOfSequenceEndorsePolicyOutputRepresentation](#)

Example

Here's an example of how to invoke the Out-of-Sequence Policy API by using Apex.

```
/*
 * Out of Sequence Policy Endorsement Service using Connect API
 *
 * IMPORTANT: Out-of-sequence endorsement is an asynchronous process and returns a job
 identifier
 * for tracking the request.
 */
public with sharing class OutOfSequenceEndorsement implements System.Callable {

    public Object call(String action, Map<String, Object> args) {
        // Retrieve the input and output maps from the arguments
        Map<String, Object> inputMap = (Map<String, Object>) args.get('input');
        Map<String, Object> outputMap = (Map<String, Object>) args.get('output');

        // Check the action type
        if (action == 'outOfSequenceEndorsePolicy') {
            System.debug('Start of connect API call to out of sequence endorse policy');

            // Get input parameters from the input map
            String policyId = (String) inputMap.get('policyId');
            String effectiveDate = (String) inputMap.get('effectiveDate');
            String contextId = (String) inputMap.get('contextId');

            System.debug('Policy ID: ' + policyId);
            System.debug('Effective Date: ' + effectiveDate);
            System.debug('Context ID: ' + contextId);

            // Initialize the job identifier
            String jobIdIdentifier = null;

            // Validate required parameters
            if (String.isBlank(policyId)) {
                System.debug('ERROR: Policy ID is required');
                outputMap.put('error', 'Policy ID is required');
                return null;
            }
            if (String.isBlank(effectiveDate)) {
                System.debug('ERROR: Effective Date is required');
                outputMap.put('error', 'Effective Date is required');
                return null;
            }
            if (String.isBlank(contextId)) {
                System.debug('ERROR: Context ID is required');
                outputMap.put('error', 'Context ID is required');
                return null;
            }
        }
    }
}
```

```

try {

    // Step 1: Create OOSE input using the provided context ID
    ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation policyInput =
    new ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation();

    // Populate fields for InsurancePolicyInputRepresentation
    ConnectApi.InsurancePolicyInputRepresentation insurancePolicy =
    new ConnectApi.InsurancePolicyInputRepresentation();
    insurancePolicy.policyName = 'Lisa Auto Policy';
    insurancePolicy.policyNumber = 'Lisa Auto Policy';
    policyInput.insurancePolicy = insurancePolicy;

    // Populate transaction record
    ConnectApi.InsurancePolicyTransactionInputRepresentation transactionRecord =
    new ConnectApi.InsurancePolicyTransactionInputRepresentation();
    transactionRecord.name = 'Custom OOSE Transaction';
    policyInput.transactionRecord = transactionRecord;

    // Set the effective date for the out of sequence endorsement
    policyInput.effectiveDate = effectiveDate;

    // Use the provided context ID
    policyInput.contextId = contextId;

    // Set field sets
    ConnectApi.AdditionalFieldsFieldSetInputRepresentation fieldSetIds =
    new ConnectApi.AdditionalFieldsFieldSetInputRepresentation();
    fieldSetIds.policyFieldSetId = '0IXZM0000008P684AE'; // Replace with your policy field
set ID
    fieldSetIds.assetFieldSetId = '0IXZM0000004DDC4A2'; // Replace with your asset field set
ID
    fieldSetIds.participantFieldSetId = '0IXZM0000004DDH4A2'; // Replace with your participant
field set ID
    fieldSetIds.coverageFieldSetId = '0IXZM0000008P6D4AU'; // Replace with your coverage field
set ID
    policyInput.fieldSetIds = fieldSetIds;

    // Step 2: Execute OOSE with the provided context ID
    System.debug('Executing OOSE with context ID: ' + contextId);

    // Call the Connect API directly - OOSE is ASYNCHRONOUS and returns a job identifier
    ConnectApi.OutOfSequenceEndorsePolicyOutputRepresentation ooseResult =
    ConnectApi.InsurancePolicyAdminFamily.outOfSequenceEndorsePolicy(policyInput, policyId);

    // Get the job identifier
    jobIdIdentifier = ooseResult.jobIdentifier;

    // Log success
    System.debug('Out-of-sequence endorsement job started successfully. Job ID: ' +
jobIdentifier);
    System.debug('Policy: ' + policyId + ' | Context: ' + contextId + ' | Effective Date: '
+ effectiveDate);
}

```

```

// Put the job identifier in the output map
outputMap.put('jobIdentifier', jobIdentifier);
outputMap.put('ooseResult', ooseResult);

} catch (Exception e) {
System.debug('Error while executing out-of-sequence endorsement: ' + e.getMessage());
outputMap.put('error', e.getMessage());
}

// Return the job identifier
return jobIdentifier;
}

// If the action is not recognized, return null
return null;
}
}

/*
 * USAGE EXAMPLE:
 * Copy and paste this code into Developer Console Execute Anonymous Window
 */

// Create instance of the callable class
OutOfSequenceEndorsement callable = new OutOfSequenceEndorsement();

// Prepare the input arguments
Map<String, Object> input = new Map<String, Object>();
input.put('policyId', '0YTxx00000000MbGAI'); // Replace with your policy ID
input.put('effectiveDate', '2025-06-01'); // Replace with your desired effective date
input.put('contextId', 'your-context-id-here'); // Replace with your context ID

// Prepare the output map
Map<String, Object> output = new Map<String, Object>();

// Prepare the args map
Map<String, Object> args = new Map<String, Object>();
args.put('input', input);
args.put('output', output);

// Call the 'outOfSequenceEndorsePolicy' action
Object result = callable.call('outOfSequenceEndorsePolicy', args);

// Retrieve the job identifier from the result
String jobIdentifier = (String) result;

// Output the job identifier to the debug log
System.debug('Job Identifier: ' + jobIdentifier);

// Check for errors in the output map
if (output.containsKey('error')) {
    System.debug('Error: ' + output.get('error'));
} else if (output.containsKey('jobIdentifier')) {

```

```

    System.debug('SUCCESS: OOSE job started with ID: ' + output.get('jobIdentifier'));
    System.debug('Monitor this job ID for completion status.');
```

ReinstatePolicy(ReinstatePolicyInput, policyId)

Reinstate a previously canceled insurance policy.

API Version

65.0

Requires Chatter

No

Signature

```

public static ConnectApi.PolicyOutputRepresentation
ReinstatePolicy(ConnectApi.ReinstatePolicyInputRepresentation ReinstatePolicyInput,
String policyId)
```

Parameters

ReinstatePolicyInput

Type: [ConnectApi.ReinstatePolicyInputRepresentation](#)

Input representation for reinstating an insurance policy.

policyId

Type: String

ID of the insurance policy to be reinstated.

Return Value

Type: [ConnectApi.PolicyOutputRepresentation](#)

Example

Here's an example of how to invoke the Reinstate Policy API by using Apex.

```

// Create and populate the ReinstatePolicyInputRepresentation object
ConnectApi.ReinstatePolicyInputRepresentation policyInput = new
ConnectApi.ReinstatePolicyInputRepresentation();

// Populate fields for InsurancePolicyInputRepresentation
ConnectApi.InsurancePolicyInputRepresentation insurancePolicy = new
ConnectApi.InsurancePolicyInputRepresentation();
insurancePolicy.policyName = 'Health Insurance Policy Reinstated';
insurancePolicy.policyNumber = 'HIPE12';

// Create the list that will hold the AdditionalFieldsMapValue wrappers.
List<ConnectApi.AdditionalFieldsMapValue> additionalFieldsValueList = new
```

```

List<ConnectApi.AdditionalFieldsMapValue>();

// 1. Create the map containing the actual field data for ONE item.
Map<String, Object> dataForFirstMapValue = new Map<String, Object>();
dataForFirstMapValue.put('instanceKey', 'AutoSilver'); // Example key/value from your JSON
structure
Map<String, Object> innerFieldsMap = new Map<String, Object>();
innerFieldsMap.put('Currency__c', '4456'); // Example inner field value
dataForFirstMapValue.put('fields', innerFieldsMap); // Put the inner fields map here

// 2. Create an instance of the AdditionalFieldsMapValue wrapper for this item.
ConnectApi.AdditionalFieldsMapValue firstMapValueWrapper = new
ConnectApi.AdditionalFieldsMapValue();

// 3. Assign the map data (created in step 1) to the *correct property* on the wrapper
instance.
firstMapValueWrapper.additionalFieldsMapValue = dataForFirstMapValue;

// 4. Add the populated wrapper instance to the list.
additionalFieldsValueList.add(firstMapValueWrapper);

// --- End of Populating the First Item ---

// --- If you need more items in the list, repeat steps 1-4 for each ---

// Assign the populated additional object to the policyInput
policyInput.additionalFields = additionalFieldsValueList;

policyInput.insurancePolicy = insurancePolicy;

// Optionally populate InsurancePolicyTransactionInputRepresentation
ConnectApi.InsurancePolicyTransactionInputRepresentation transactionRecord = new
ConnectApi.InsurancePolicyTransactionInputRepresentation();
policyInput.transactionRecord = transactionRecord;

// Optionally set the effective date
// policyInput.effectiveDate = '2025-06-01';

try {
    // Call the Connect API directly
    ConnectApi.PolicyOutputRepresentation policyData =
        ConnectApi.InsurancePolicyAdminFamily.reinstatePolicy(policyInput,
        '0YTx0000000MbGAI');

    // Output the policy data to the debug log
    System.debug('Policy reinstated successfully: ' + JSON.serializePretty(policyData));
} catch (Exception e) {
    // Log the error
    System.debug('Error while reinstating policy: ' + e.getMessage());
}

```

RenewPolicy(RenewPolicyInput, policyId)

Use the context ID of an existing policy to renew a policy. Modify the renewal date, and add optional fields and additional information in the input JSON.

API Version

63.0

Requires Chatter

No

Signature

```
public static ConnectApi.PolicyOutputRepresentation
RenewPolicy(ConnectApi.RenewPolicyInputRepresentation RenewPolicyInput, String policyId)
ConnectApi.InsurancePolicyAdminFamily, RenewPolicy,
[ConnectApi.RenewPolicyInputRepresentation, String], ConnectApi.PolicyOutputRepresentation
```

Parameters

RenewPolicyInput

Type: [ConnectApi.RenewPolicyInputRepresentation](#) on page 529

Input representation of the request to renew a policy.

policyId

Type: String

ID of the insurance policy that's to be renewed.

Return Value

Type: [ConnectApi.PolicyOutputRepresentation](#) on page 533

ConnectApi Input Classes

Some `ConnectApi` methods take arguments that are instances of `ConnectApi` input classes.

Input classes are concrete unless marked abstract in this documentation. Concrete input classes have public constructors that have no parameters.

Some methods have parameters that are typed with an abstract class. You must pass in an instance of a concrete child class for these parameters.

Most input class properties can be set. Read-only properties are noted in this documentation.

Insurance Policy Administration includes these Apex input classes.

[ConnectApi.AdditionalFieldIARep](#)

Input representation for the additional fields.

[ConnectApi.AdditionalFieldsMapValue](#)

AdditionalFieldsMapValue wrapper object for use as child wrapper of AdditionalFieldsMap.

[ConnectApi.AdditionalFieldsFieldSetInputRepresentation](#)

Input Representation for Additional Fields Field Set

[ConnectApi.CancelPolicyInputRepresentation](#)

Input representation of the request to cancel a policy.

[ConnectApi.EndorsePolicyInputRepresentation](#)

Input representation of the request to endorse a policy. Context ID of the quote that's used for endorsement and date that the endorsed policy version is effective from must be passed in the request.

[ConnectApi.FieldSetIDsInputRep](#)

List of the field set IDs for asset, coverage, participant, and policy objects.

[ConnectApi.InsPolicyAddlInputIRep](#)

Input representation for the additional fields of an insurance policy.

[ConnectApi.InsPolicyIInputRep](#)

Input representation details of an insurance policy.

[ConnectApi.InsurancePolicyInputRepresentation](#)

Input representation details of an insurance policy.

[ConnectApi.InsPolicyTrxnIInputRep](#)

Input representation details of an insurance policy transaction.

[ConnectApi.InsurancePolicyTransactionInputRepresentation](#)

Input representation details of an insurance policy transaction.

[ConnectApi.IssuePolicyInputRepresentation](#)

Input representation of the request to issue an insurance policy.

[ConnectApi.OutOfSequencePolicyEndorsementInputRep](#)

Input representation of the details to perform an out-of-sequence endorsement on an insurance policy.

[ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation](#)

Input representation of the details to perform an out-of-sequence endorsement on an insurance policy.

[ConnectApi.ReinstatePolicyInputRep](#)

Input representation for the Reinstate Insurance Policy Invocable Action.

[ConnectApi.ReinstatePolicyInputRepresentation](#)

Input representation for reinstating an insurance policy.

[ConnectApi.RenewPolicyInputRepresentation](#)

Input representation of the request to renew a policy.

[ConnectApi.RenewInsurancePoliciesIInputRep](#)

An Apex ConnectApi.RenewInsurancePoliciesIInputRep record that contains the details of the insurance policies to be renewed.

ConnectApi.AdditionalFieldIInputRep

Input representation for the additional fields.

Property	Type	Description	Required or Optional	Available Version
key	String	Name of key.	Optional	63.0
value	String	Value for the key.	Optional	63.0

ConnectApi.AdditionalFieldsMapValue

AdditionalFieldsMapValue wrapper object for use as child wrapper of AdditionalFieldsMap.

Property	Type	Description	Required or Optional	Available Version
additionalFieldsMapValue	Object	Additional fields map containing fields to be updated.	Required	63.0

ConnectApi.AdditionalFieldsFieldSetInputRepresentation

Input Representation for Additional Fields Field Set

Property	Type	Description	Required or Optional	Available Version
assetFieldSetId	String	ID of the field set to get the fields that are associated with the asset object.	Optional	65.0
coverageFieldSetId	String	ID of the field set to get the fields that are associated with the coverage object.	Optional	65.0
participantFieldSetId	String	ID of the field set to get the fields that are associated with the participant object.	Optional	65.0
policyFieldSetId	String	ID of the field set to get the fields that are associated with the policy object.	Optional	65.0

ConnectApi.CancelPolicyInputRepresentation

Input representation of the request to cancel a policy.

Property	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date on which the cancellation of the policy goes into effect.	Required	63.0
transactionRecord	TransactionRecord on page 525	The JSON structure that overrides the transaction information.	Optional	63.0

ConnectApi.EndorsePolicyInputRepresentation

Input representation of the request to endorse a policy. Context ID of the quote that's used for endorsement and date that the endorsed policy version is effective from must be passed in the request.

Property	Type	Description	Required or Optional	Available Version
additionalFields	ConnectApi.AdditionalFieldsMapValue on page 522>	The JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Context ID of the quote that's used for endorsement.	Required	63.0
effectiveDate	String	Date from when endorsed policy version is effective from.	Required	63.0
insurancePolicy	InsurancePolicyInputRepresentation on page 525	The JSON structure that overrides the name and policy number of the endorsed policy version.	Optional	63.0
transactionRecord	InsurancePolicyTransactionInputRepresentation on page 525	The JSON structure that overrides the transaction information.	Optional	63.0

ConnectApi.FieldSetIdsInputRep

List of the field set IDs for asset, coverage, participant, and policy objects.

Property	Type	Description	Required or Optional	Available Version
assetFieldSetId	String	ID of the field set to get the fields associated with the asset object.	Optional	64.0
coverageFieldSetId	String	ID of the field set to get the fields associated with the coverage object.	Optional	64.0
participantFieldSetId	String	ID of the field set to get the fields associated with the participant object.	Optional	64.0

Property	Type	Description	Required or Optional	Available Version
policyFieldSetId	String	ID of the field set to get the fields associated with the policy object.	Optional	64.0

ConnectApi.InsPolicyAddlInputIARep

Input representation for the additional fields of an insurance policy.

Property	Type	Description	Required or Optional	Available Version
additionalFieldsList	ConnectApi.AdditionalFieldIARep	Additional fields for the insurance policy.	Optional	63.0
instanceKey	String	Instance key for the additional fields of the insurance policy.	Optional	63.0

ConnectApi.InsPolicyIAInputRep

Input representation details of an insurance policy.

Property	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Date from when the policy is effective. This property isn't applicable to the out-of-sequence endorsement and policy reinstatement APIs.	Optional	63.0
effectiveToDate	String	Date until when the policy is effective. This property isn't applicable to the out-of-sequence endorsement and policy reinstatement APIs.	Optional	63.0
policyName	String	Name of the insurance policy.	Optional	63.0
policyNumber	String	Number of the policy.	Optional	63.0
referencePolicyNumber	String	Reference number that's used to link policies across terms. If not specified, policies are linked by the insurance policy ID.	Optional	65.0

ConnectApi.InsurancePolicyInputRepresentation

Input representation details of an insurance policy.

Property	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Date from when the policy is effective. This property isn't applicable to the out-of-sequence endorsement and policy reinstatement APIs.	Optional	63.0
effectiveToDate	String	Date until when the policy is effective. This property isn't applicable to the out-of-sequence endorsement and policy reinstatement APIs.	Optional	63.0
policyName	String	Name of the insurance policy.	Optional	63.0
policyNumber	String	Number of the policy.	Optional	63.0
referencePolicy Number	String	Reference number that's used to link policies across terms. If not specified, policies are linked by the insurance policy ID. This property isn't applicable to the out-of-sequence endorsement and policy reinstatement APIs.	Optional	65.0

ConnectApi.InsPolicyTrxnInputRep

Input representation details of an insurance policy transaction.

Property	Type	Description	Required or Optional	Available Version
name	String	Name of the transaction, such as Cancel Transaction.	Optional	63.0

ConnectApi.InsurancePolicyTransactionInputRepresentation

Input representation details of an insurance policy transaction.

Property	Type	Description	Required or Optional	Available Version
name	String	Name of the transaction, such as Cancel Transaction.	Optional	63.0

ConnectApi.IssuePolicyInputRepresentation

Input representation of the request to issue an insurance policy.

Property	Type	Description	Required or Optional	Available Version
additionalFields	List< ConnectApi.AdditionalFieldMetadata on page 522>	The JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. Address and Geolocation field types are not supported as additional fields. Add the SourceQuoted standard field to the additionalFields option when the context ID in the input JSON for the Issue Insurance Policy API is generated by the Create Insurance Quote API.	Optional	63.0
contextId	String	Context ID of the quote that's used to create the policy.	Required	63.0
insurancePolicy	InsurancePolicyInputRepresentation on page 524	The JSON structure that's used to create the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
transactionRecord	InsurancePolicyTransactionInputRepresentation on page 525	The JSON structure that's used to override transaction information.	Optional	63.0

ConnectApi.OutOfSequencePolicyEndorsementInputRep

Input representation of the details to perform an out-of-sequence endorsement on an insurance policy.

Property	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context that's used to create an endorsement policy version.	Required	65.0
effectiveDate	String	Date on which the out of sequence endorsement operation happens.	Required	65.0
fieldSetIds	FieldSetIdsInputRep	Field set IDs of the additional fields that are associated with the insurance policy.	Optional	65.0
insurancePolicy	InsPolicyIAInputRep	Details of the insurance policy to create the out-of-sequence endorsement for.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
policyId	String	ID of the insurance policy to create the out-of-sequence endorsement for.	Required	65.0
transactionRecord	InsurancePolicyTransactionInputRepresentation	JSON structure that's used to override the transaction information.	Optional	65.0

ConnectApi.OutOfSequenceEndorsePolicyInputRepresentation

Input representation of the details to perform an out-of-sequence endorsement on an insurance policy.

Property	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context that's used to create an endorsement policy version.	Required	65.0
effectiveDate	String	Date from which the out-of-sequence endorsement applies. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set <code>effectiveDate</code> to 2025-03-01, the endorsement applies starting March 1, 2025, and the API re-rates the policy from 2025-03-01 forward.	Required	65.0
fieldSetIds	AdditionalFieldsFieldSetInputRepresentation	Field set IDs of the additional fields that are associated with the insurance policy.	Optional	65.0
insurancePolicy	InsurancePolicyInputRepresentation	Details of the insurance policy to create the out-of-sequence endorsement for.	Optional	65.0
transactionRecord	InsurancePolicyTransactionInputRepresentation	JSON structure that's used to override the transaction information.	Optional	65.0

ConnectApi.ReinstatePolicyInputRep

Input representation for the Reinstate Insurance Policy Invocable Action.

Property	Type	Description	Required or Optional	Available Version
additionalInput	ConnectApi.InsPolicyAddlInputIARep	JSON structure that's used to add custom or standard fields to the <code>InsurancePolicy</code> , <code>InsurancePolicyAsset</code> ,	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
		InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalInput take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional input.		
effectiveDate	String	Date from when the policy is reinstated. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set effectiveDate to 2025-03-01, the reinstatement applies starting March 1, 2025.	Optional	65.0
insurancePolicy	InsPolicyIAInputRep	Details of the insurance policy.	Optional	65.0
policyId	String	ID of the canceled insurance policy to be reinstated.	Required	65.0
transactionRecord	InsPolicyTrxnIAInputRep	JSON structure that's used to override the transaction information.	Optional	65.0

ConnectApi.ReinstatePolicyInputRepresentation

Input representation for reinstating an insurance policy.

Property	Type	Description	Required or Optional	Available Version
additionalFields	ConnectApi.AdditionalFieldsMapValue	JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	65.0

Property	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date from which the policy reinstatement becomes effective. This date must be within the policy term. For example, if the policy term is 2025-01-01 to 2025-12-31, and you set effectiveDate to 2025-03-01, the reinstatement applies starting March 1, 2025.	Optional	65.0
insurancePolicy	InsurancePolicyInputRepresentation	Details of the insurance policy.	Optional	65.0
transactionRecord	InsurancePolicyTransactionInputRepresentation	JSON structure that's used to override the transaction information.	Optional	65.0

ConnectApi.RenewPolicyInputRepresentation

Input representation of the request to renew a policy.

Property	Type	Description	Required or Optional	Available Version
additionalFields	List< ConnectApi.AdditionalFieldValue on page 522>	The JSON structure that's used to add custom or standard fields to the InsurancePolicy, InsurancePolicyAsset, InsurancePolicyCoverage, and InsurancePolicyParticipant objects. The values for additionalFields take precedence over the context or the insurancePolicy JSON. This is applicable only for the endorsed policy version. Address and Geolocation field types are not supported as additional fields.	Optional	63.0
contextId	String	Context ID of the quote that's used to renew the policy.	Required	63.0
insurancePolicy	InsurancePolicyInputRepresentation on page 524	The JSON structure that's used to renew the policy. It contains fields and values from the Insurance Policy object.	Optional	63.0
transactionRecord	InsurancePolicyTransactionInputRepresentation on page 525	The JSON structure that's used to override the transaction information.	Optional	63.0

ConnectApi.RenewInsurancePoliciesInputRep

An Apex `ConnectApi.RenewInsurancePoliciesInputRep` record that contains the details of the insurance policies to be renewed.

Property	Type	Description	Required or Optional	Available Version
<code>executeConfigurationRules</code>	Boolean	Indicates whether the configuration rules for renewal must be executed (<code>true</code>) or not (<code>false</code>).	Optional	64.0
<code>executeQualificationRules</code>	Boolean	Indicates whether the qualification rules for renewal must be executed (<code>true</code>) or not (<code>false</code>).	Required for quote renewals.	64.0
<code>fieldSetIds</code>	Field SetIdsInputRep on page 523	Input representation of field set IDs.	Optional	64.0
<code>flowAPINameWithNamespace</code>	String	The API name of the flow to execute changes before each policy or quote renewal process. Include the namespace prefix, if any.	Optional	64.0
<code>issueQuote</code>	Boolean	Indicates whether the renewal quote must be issued (<code>true</code>) or not (<code>false</code>).	Optional	64.0
<code>jobIdentifier</code>	String	A unique ID of the batch job that's used to monitor and generate reports using Custom Report Type. The maximum length is 50 characters.	Required	64.0
<code>policyId</code>	String	ID of the insurance policy to be renewed.	Required	64.0
<code>priceBookId</code>	String	ID of the price book that's used to renew policies. Defaults to the Standard Price Book record ID if no value is provided.	Required for quote renewals.	64.0
<code>productSellingModel</code>	String	The model that's used to offer the product, such as one-time or term-based. The default is one-time if no value is provided.	Required for quote renewals.	64.0
<code>sameTenure</code>	Boolean	Indicates whether the same tenure must be set for renewal as original policy (<code>true</code>) or not (<code>false</code>).	Optional	64.0

ConnectApi Output Classes

Most `ConnectApi` methods return instances of `ConnectApi` output classes.

All properties are read-only, except for instances of output classes created within test code.

All output classes are concrete unless marked abstract in this documentation.

All concrete output classes have no-argument constructors that you can invoke only from test code. See [Testing ConnectApi Code](#).

Insurance Policy Administration includes these Apex output classes.

[ConnectApi.CancelPolicyOutputRepresentation](#)

Output representation details of cancel policy, such as impacted policy versions and refund associated with the cancellation.

[ConnectApi.ErrorDetailsOutputRepresentation](#)

Output representation of the error details, such as error code and error message.

[ConnectApi.ErrorOutputRepresentation](#)

Output representation details of the error encountered while processing the request.

[ConnectApi.GetPolicyOutputRepresentation](#)

Output representation of the policy details.

[ConnectApi.OutOfSequenceEndorsePolicyOutputRepresentation](#)

Output representation for out of sequence endorsement output representation.

[ConnectApi.PolicyAsyncRepresentation](#)

Output representation for multi-root policy renewal asynchronous operations.

[ConnectApi.PolicyOutputRepresentation](#)

Output representation of the policy details.

[ConnectApi.RefundOutputRepresentation](#)

Output representation of the refund payment.

ConnectApi.CancelPolicyOutputRepresentation

Output representation details of cancel policy, such as impacted policy versions and refund associated with the cancellation.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorOutputRepresentation on page 531	List of errors encountered during the processing of the API request.	63.0
impactedPolicyVersions	List<String>	List of impacted policy versions.	63.0
refund	ConnectApi.RefundOutputRepresentation on page 533	Refund associated with cancellation operation.	63.0

ConnectApi.ErrorDetailsOutputRepresentation

Output representation of the error details, such as error code and error message.

Property Name	Type	Description	Available Version
code	String	The error code thrown when the request fails.	63.0
message	String	Message stating the reason for error, if any.	63.0

ConnectApi.ErrorOutputRepresentation

Output representation details of the error encountered while processing the request.

Property Name	Type	Description	Available Version
detail	String	Description of the error.	63.0
errors	List< ConnectApi.ErrorDetailsOutputRepresentation on page 531>	Output representation of the error details.	63.0
instance	String	Instance where the error occurred.	63.0
title	String	Short label of the error type.	63.0

ConnectApi.GetPolicyOutputRepresentation

Output representation of the policy details.

Property Name	Type	Description	Available Version
contextId	String	ID of the context.	63.0
error	ConnectApi.ErrorOutputRepresentation on page 531	List of errors encountered during the processing of the API request.	63.0
insurancePolicy	Map<String, Object>	Details of insurance policy.	63.0

ConnectApi.OutOfSequenceEndorsePolicyOutputRepresentation

Output representation for out of sequence endorsement output representation.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorOutputRepresentation	List of errors when the out-of-sequence endorsement operation fails.	65.0
jobIdentifier	String	ID of the job that's created for the out-of-sequence endorsement process.	65.0

ConnectApi.PolicyAsyncRepresentation

Output representation for multi-root policy renewal asynchronous operations.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorOutputRepresentation	List of errors that occurred during the processing of the API request.	66.0
jobIdentifier	String	ID of the asynchronous job that's generated to process the multi-root policy.	66.0

ConnectApi.PolicyOutputRepresentation

Output representation of the policy details.

Property Name	Type	Description	Available Version
error	ConnectApi.ErrorOutputRepresentation	List of errors encountered during the processing of the API request.	63.0
originalPolicyId	String	ID of the original policy.	63.0
policyId	String	ID of the issued or endorsed policy.	63.0

ConnectApi.RefundOutputRepresentation

Output representation of the refund payment.

Property Name	Type	Description	Available Version
fee	Double	Refund fee amount after cancellation.	63.0
premium	Double	Refund premium amount after cancellation.	63.0
tax	Double	Refund tax amount after cancellation.	63.0

Insurance Policy Developer Journey

To manage insurance policies by using Salesforce Quoting, use the Quoting APIs to generate a quote context, which serves as the foundation for policy operations. For managing policies without Salesforce Quoting, use the Rating API to create a price-adjusted context. This price-adjusted context provides the necessary rated data to manage insurance policies without depending on a formal quote record.

[Create and Manage Insurance Policies by Using Salesforce Quoting](#)

Learn how to effectively use the Insurance Policy and Quoting Connect APIs, along with the associated invocable actions, to build comprehensive user journeys by using Salesforce Flows and Omniscripts.

[Create and Manage Insurance Policies Without Using Salesforce Quoting](#)

Learn how to use the Insurance Policy APIs and invocable actions to manage the policy lifecycle when Salesforce quoting isn't used, or when a third-party system handles the quoting process.

[Cancel Policy](#)

The policy cancellation process addresses the termination of an insurance policy before its scheduled expiration date. Common reasons for canceling a policy include not paying the premiums or a direct cancellation request from the policyholder.

Create and Manage Insurance Policies by Using Salesforce Quoting

Learn how to effectively use the Insurance Policy and Quoting Connect APIs, along with the associated invocable actions, to build comprehensive user journeys by using Salesforce Flows and Omniscripts.

The Salesforce Insurance Policy and Quoting APIs, and the related invocable actions provide a complete set of tools to manage the entire insurance policy lifecycle. These APIs support key policy administration processes, such as policy issuance, endorsement, renewal, and

cancellation. By using these APIs and invocable actions, you can create user journeys for insurance representatives who use the Salesforce console, as well as the policyholders and partners who access the services through community portals.

Endorse Policy

Policy endorsement, also known as Mid-Term Adjustment (MTA), is the process of making modifications to an active insurance policy during its term. Common modifications include adding or removing coverage, or updating policy attributes, such as limits and deductibles. Endorsements usually affect the policy price and create a new policy version instead of updating the existing record. This new version includes the applied changes, along with a recalculated standard premium. It also contains a prorated term premium that's calculated based on the effective date of the endorsement.

Issue Policy

The policy issuance process begins after the policyholder accepts the quote provided by the insurance company.

Renew Policy

Policy renewal is the process of extending insurance coverage for an additional policy term, typically as the current policy approaches its effective-to date. A key part of renewal is that a renewed policy creates a new policy record.

Create Out of Sequence Policy Endorsement

Perform a backdated change on a policy in the past, current, or future policy term, even when another policy version with an effective date later than the endorsement date already exists. The impacted policy versions are canceled and replaced with new, repriced versions that include the change. The premium impact for each affected policy term is captured in an individual out-of-sequence endorsement transaction.

Issue Multi-Root Policy

The policy issuance process begins after the policyholder accepts the quote provided by the insurance company. This process creates a parent-child insurance policy structure in which a parent policy serves as a lifecycle container for one or more insured child policies.

Endorse Multi-Root Policy

Modifies an active multi-root insurance policy by creating new versions of the parent policy and all child policies, regardless of which child policy is changed. The resulting policy versions include recalculated standard and prorated premiums based on the endorsement effective date.

Renew Multi-Root Policy

Renew a multi-root policy to create a new policy term for the parent policy and all associated child policies. The parent-child policy structure is preserved in the renewed term.

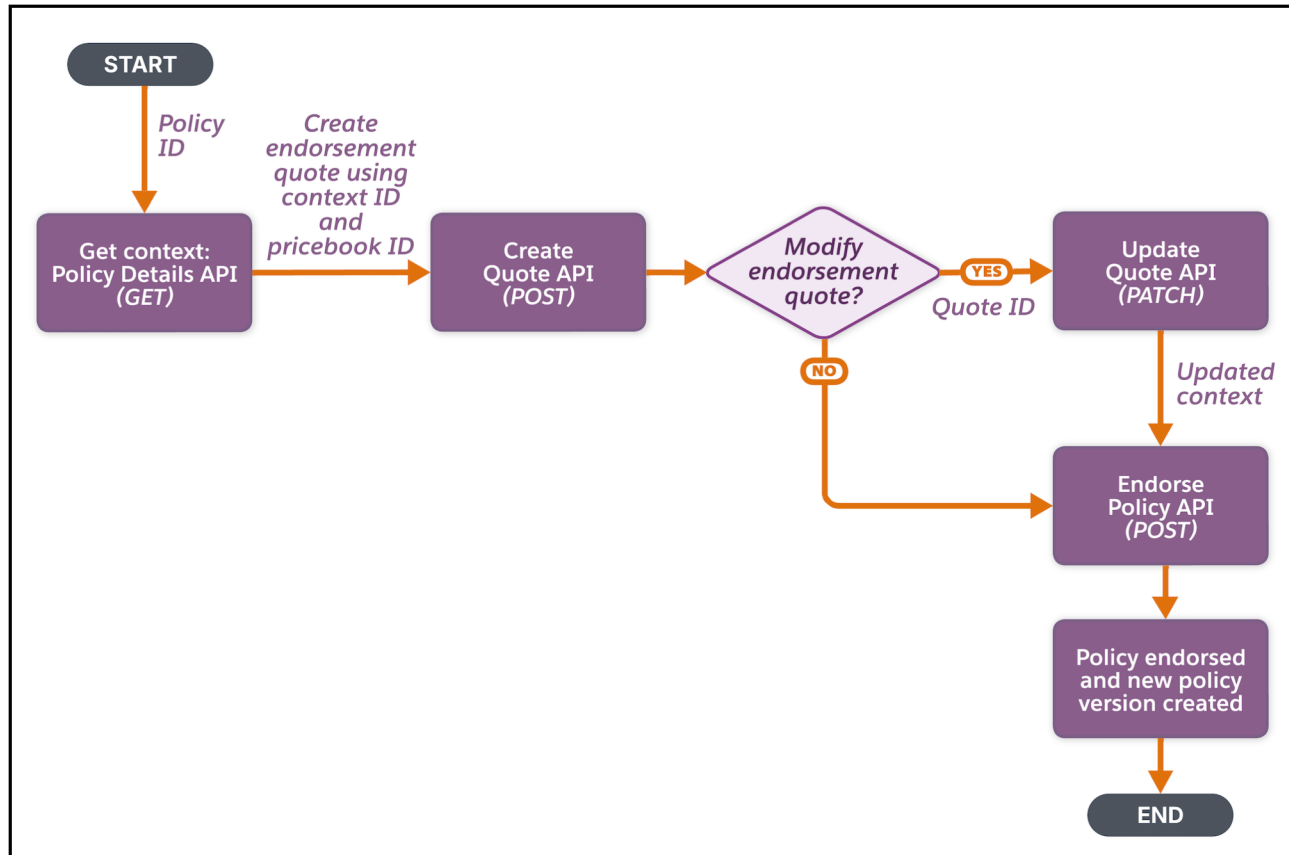
Endorse Policy

Policy endorsement, also known as Mid-Term Adjustment (MTA), is the process of making modifications to an active insurance policy during its term. Common modifications include adding or removing coverage, or updating policy attributes, such as limits and deductibles. Endorsements usually affect the policy price and create a new policy version instead of updating the existing record. This new version includes the applied changes, along with a recalculated standard premium. It also contains a prorated term premium that's calculated based on the effective date of the endorsement.

Make sure you have these details.

- Policy ID of the active insurance policy to be endorsed.
- Pricebook ID is a required input for generating an endorsement quote.
- Details of the modifications to be made to the policy, such as specific coverages to add or remove, and new limit amounts.

This diagram outlines the sequence of endorsing a policy, where any changes are initially processed through a quote to assess the pricing impact.



1. Get Policy Context—Use the policy ID to get the context either by:

- Making a GET request on the [Insurance Policy Details](#) Connect API, or
- Using the [Get Insurance Policy](#) invocable action.

2. Create Endorsement Quote—Use the policy context to create the endorsement quote either by:

- Making a POST request on the [Insurance Quote](#) Connect API, or
- Using the [Create Insurance Quote](#) invocable action.

Specify the `contextId` and `pricebookId` during the API call. You can update the header-level fields of the quote, such as the quote name or effective date for the endorsement, by specifying these values as the input parameters during the API call.

A new quote record is generated for the endorsement. After the policyholder confirms the updated quote price, proceed with the next steps in the endorsement workflow.

3. (Optional)Modify Endorsement Quote—If policyholders want to modify the endorsement quote, make a PATCH request on the [Insurance Quote Details](#) Connect API with the required changes.

An updated, price-adjusted context is returned in the response.

4. Endorse Policy—Use the retrieved context to endorse a policy either by:

- Making a POST request on the [Endorse Insurance Policy](#) Connect API, or
- Using the [Endorse Insurance Policy](#) invocable action.

You can change the context fields, such as the policy name or policy number, directly during the API call. You can also provide a name for the transaction record that's created along with the endorsement.

This transaction record documents the financial details and audit information associated with the MTA.

A new endorsed policy version is created.

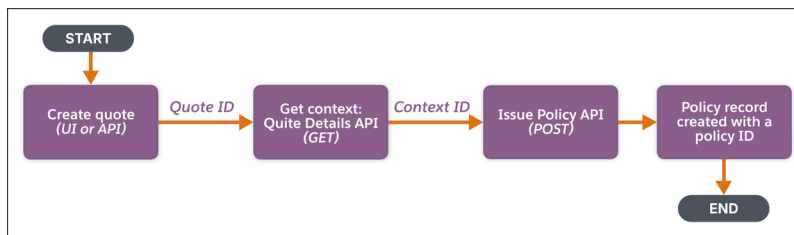
Expected Results:

- A new policy version record is created in Salesforce, which includes the endorsed changes.
- A new policy version is created with an updated effective-from date. The effective-to date of the existing policy version is adjusted to an end date just before the new version begins, ensuring continuous but non-overlapping coverage.
- The new policy version has the updated premium information, including the new standard premium and the prorated term premium, accurately calculated and stored.
- A transaction record is created and linked to the endorsement activity.

Issue Policy

The policy issuance process begins after the policyholder accepts the quote provided by the insurance company.

This diagram outlines the sequence of issuing a policy based on a quote.



1. Create Quote—Create a quote either by:

- Making a POST request on the [Insurance Quote](#) Connect API, or
- Using the [Create Insurance Quote](#) invocable action, or
- Using the [Salesforce Quote User Interface](#).

This process creates the quote and generates the quote ID.

2. Get Quote Context—Use the quote ID to retrieve the quote details either by:

- Making a GET request on the [Insurance Quote Details](#) Connect API, or
- Using the [Get Insurance Quote Details](#) invocable action.

This request returns the context ID in the response that contains the context of the quote.

3. Issue Policy—Use the context details, specifically the context ID, to issue policy either by:

- Making a POST request on the [Issue Insurance Policy](#) Connect API, or
- Using the [Issue Insurance Policy](#) invocable action.

You can override the context data, such as policy name, policy number, effective-from date, effective-to date, and any other additional fields to provide the values for the custom fields during the API call. You can also specify a name for the transaction record that's created as part of the policy issuance process. This transaction record is used to track an audit or as a financial marker.

This request creates a new policy record.

Expected Results:

- A new policy record is created and saved in Salesforce. The policy record includes the details, such as the policy name, number (if provided or created during the API call), policy start date, chosen coverage, and the price.
- Any custom field values that are provided during the API call are saved to the new policy record.
- A transaction record is created and linked to the policy.

Renew Policy

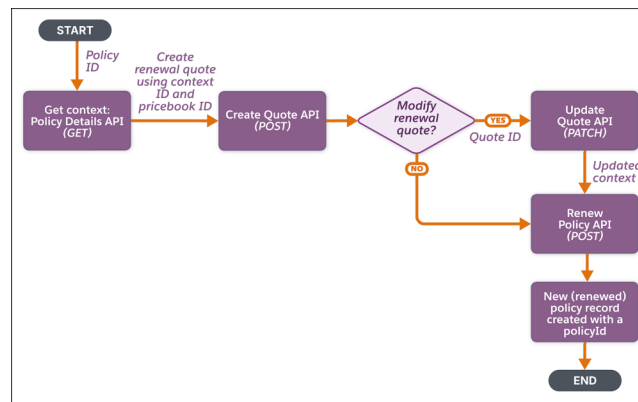
Policy renewal is the process of extending insurance coverage for an additional policy term, typically as the current policy approaches its effective-to date. A key part of renewal is that a renewed policy creates a new policy record.

Make sure you have these details.

- Policy ID of the active insurance policy that's to be renewed.
- Pricebook ID as it's a required input for generating an endorsement quote.

If the policyholder changes their information, it affects the terms or prices of the renewed policy. This journey describes renewing a policy where a formal renewal quote is generated. After the customer confirms the renewal quote, a new policy record is created for the new renewal term.

This diagram outlines the sequence of renewing an existing policy.



- 1. Get Policy Context**—Use the policy ID to get the context either by:
 - Making a GET request on the [Insurance Policy Details](#) Connect API, or
 - Using the [Get Insurance Policy](#) invocable action.
- 2. Create Renewal Quote**—Use the policy context to create a renewal quote either by:
 - Making a POST request on the [Insurance Quote](#) Connect API, or
 - Using the [Create Insurance Quote Action](#) invocable action.

Specify the `pricebookId` during the API call. You can change the header-level fields of the quote, such as the quote name or the start date of the renewal term by specifying these values as the input parameters during the API call.

A new quote record is generated for the policy renewal. After the customer confirms the updated quote price, proceed with the next steps in the renewal workflow.

- 3. (Optional)Modify Renewal Quote**—If policyholders want to modify the endorsement quote, make a PATCH request on the [Insurance Quote Details](#) Connect API with the desired changes. An updated, price-adjusted context is returned in the response.
- 4. Renew Policy**—Use the retrieved context to create a new policy record for the renewal term either by:
 - Making a POST request on the [Renew Insurance Policy](#) Connect API, or
 - Using the [Renew Insurance Policy Action](#) invocable action.

You can change or confirm the policy name, policy number, and effective start and end dates for the renewed policy directly during the API call. Additionally, you can also specify a name for the transaction record that's created along with the renewal record. This transaction record captures the financial details or the audit information related to the renewal record.

Expected Results:

- A brand new policy record is created in Salesforce, representing the coverage for the new policy term. The new policy record includes the details, such as any changed terms, coverages, prices, and the new start and end dates.
- A transaction record is created and linked to the new policy.

Create Out of Sequence Policy Endorsement

Perform a backdated change on a policy in the past, current, or future policy term, even when another policy version with an effective date later than the endorsement date already exists. The impacted policy versions are canceled and replaced with new, repriced versions that include the change. The premium impact for each affected policy term is captured in an individual out-of-sequence endorsement transaction.

Make sure you have the following details.

- Policy ID of the active insurance policy to be endorsed.
- Context ID of the endorsement quote created from the policy version that needs modification.
- Effective date for the endorsement must fall within a valid policy term.
- (Optional) FieldSetIds to copy selected fields (policy, asset, participant, coverage) from the cancelled versions to the new versions.

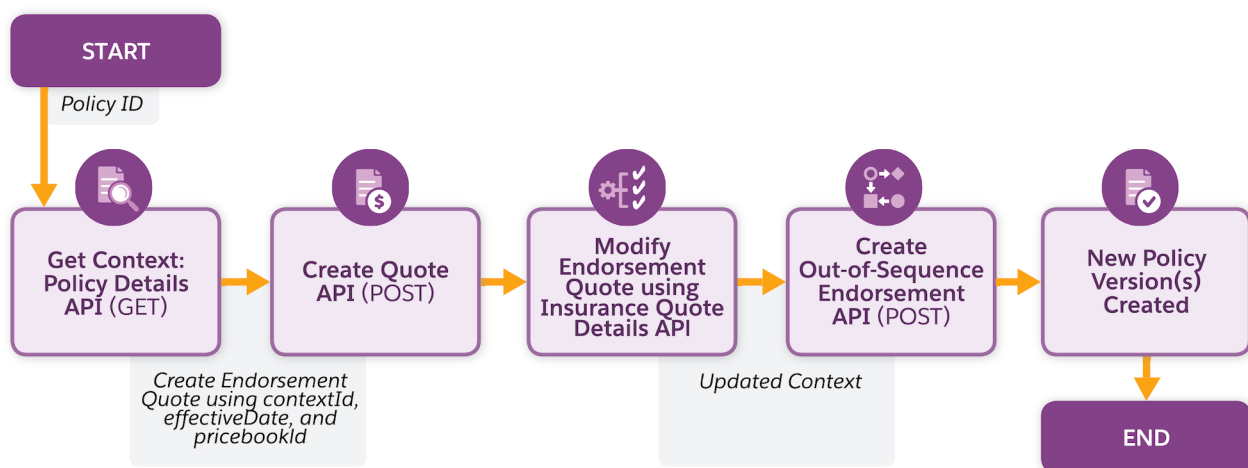
Example

If a policy has three active versions (V1, V2, and V3) within a single term, and you need to make a change from an effective date which lies in the tenure of V2. You can prepare the context for OOSE using the Quote API. Get the context from policy V2. Use this context to make changes and create an updated quote with the createUpdateQuote APIs. Retrieve the quote context, and then pass this context to OOSE.

Important

When you retrieve the policy context and run a product-rating call, it uses the effective date from the policy context to calculate premiums and taxes. If you later endorse the policy with a different effective date or start date, the endorsed policy still consumes the tax calculations from the earlier product-rating call. To avoid incorrect results, the effective date in the product-rating call must align with the effective date range of the endorsed policy. This ensures that the premiums and taxes are calculated correctly for the endorsement's term.

This diagram outlines the sequence for creating an out-of-sequence policy endorsement using a quote.

**1. Get Policy Context**—Use the policy ID to get the context either by:

- Making a GET request on the [Insurance Policy Details](#) Connect API, or

- Using the [Get Insurance Policy](#) invocable action.

2. Create Endorsement Quote—Use the policy context to create a renewal quote either by:

- Making a POST request on the [Insurance Quote](#) Connect API, or
- Using the [Create Insurance Quote Action](#) invocable action.

Specify the `contextId`, `effectiveDate`, and `pricebookId` during the API call.

A new quote record is generated for the policy renewal.

3. (Optional)Modify Endorsement Quote—If policyholders want to modify the endorsement quote, make a PATCH request on the [Insurance Quote Details](#) Connect API with the required changes.

An updated, price-adjusted context is returned in the response.

4. Submit out of sequence endorsement request—Use the retrieved context to endorse a policy either by:

- Making a POST request on the Out-of-Sequence Endorsement Connect API, or
- Using the Create Out of Sequence Policy Endorsement invocable action.

Specify the `policyId`, `contextId`, and `effectiveDate` and optional details such as custom transaction name, policy name, policy number, and `fieldSetIds` during the API call.

This request returns a `jobIdentifier` that you can use to track the async process through a custom report.

- Identifies and fetches the impacted policy versions.
- Locks impacted policy versions to prevent any other operation on the impacted policies when the OOSE request is in progress.
- Applies deltas (add, update, delete) across all impacted versions and reprices the context.
- Unlocks the impacted policy versions and sends a success or failure notification.
- Publishes a platform event that subscribers can use for tracking status.

Expected Results:

- Creates a new policy starting from the effective date of the OOSE endorsement.
- Cancels impacted future versions and replaces them with updated versions.
- Sets the effective-to date of older versions to `endorsementDate - 1` and resets the term premiums and taxes to zero.
- Recalculates premiums and taxes on the new versions and prorates them based on the endorsement date.
- Creates a transaction record for each policy term and links it to the original policy version of that term for auditing and financial tracking.

Issue Multi-Root Policy

The policy issuance process begins after the policyholder accepts the quote provided by the insurance company. This process creates a parent-child insurance policy structure in which a parent policy serves as a lifecycle container for one or more insured child policies.

Make sure you have these details before issuing a multi-root policy:

- The Quote ID of the finalized multi-root quote.
 - A fully priced context to pass to the Multi-Root Policy Issuance API.
 - For multi-root quotes configured through the Quote UI, click the Reprice All button on the Transaction Line Item Table (TLT) after all products are configured to ensure pricing and aggregate amounts are correctly reflected at the quote level.
- 1. Create Quote**—Create a quote by using one of the following methods
- Make a POST request on the [Insurance Quote](#) connect API.
 - Use the [Create Insurance Quote](#) invocable action.

- Use the [Salesforce Quote User Interface](#).

Creates the quote and generates the quote ID.

2. Get Quote Context—Retrieve the quote context by using the quote ID through one of the following methods

- Make a GET request to the [Insurance Quote Details](#) Connect API
- Use the [Get Insurance Quote Details](#) invocable action.

Returns the context ID that contains the quote context.

3. Issue Multi-Root Policy—Use the context details, specifically the context ID, to issue multi-root policy by using one of the following methods

- Make a POST request to the [Multi-Root Policy Issuance](#) connect API.
- Use the [Issue Multiroot Policy](#) invocable action.
- You can override the context data, such as policy name, policy number, effective-from date, effective-to date, and any other additional fields, to provide values for the custom fields during the API call. You can also specify a name for the transaction record that's created as part of the policy issuance process. This transaction record is used for audit tracking and financial reference.
- Pricing does not occur during this lifecycle event. The context that you pass to this request must already be fully priced.
- All policy lifecycle operations occur only on the parent policy and not on child policies. The parent policy isn't associated with any product and doesn't have any line items or attributes, except for its associated child policies.

- Returns an Async Request ID as a response to track the job status.
- Creates a multi-root policy with the following structure:
 - A parent policy that represents the policy lifecycle.
 - One or more insured child policies, created for each root product bundle and linked to the parent policy.
- Rolls up premiums and taxes from child policies to the parent policy for a consolidated financial view.
- Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.
- Sends a success or failure notification after the process completes.

Endorse Multi-Root Policy

Modifies an active multi-root insurance policy by creating new versions of the parent policy and all child policies, regardless of which child policy is changed. The resulting policy versions include recalculated standard and prorated premiums based on the endorsement effective date.

Make sure you have these details before endorsing a multi-root policy

- The Policy ID of the active parent insurance policy to be endorsed.
 - The Pricebook ID required to generate the endorsement quote.
 - Details of the modifications to apply to the policy, such as coverages to add or remove and updated limit amounts.
 - For multi-root quotes configured through the Quote UI, click the Reprice All button on the Transaction Line Item Table (TLT) after all products are configured to ensure pricing and aggregate amounts are correctly reflected at the quote level.
- 1. Get Policy Context**—Retrieve the policy context by using the policy ID through one of the following methods:
- Make a GET request to the [Insurance Policy Details](#) Connect API.
 - Use the [Get Insurance Policy](#) invocable action.

2. Create Endorsement Quote—Use the policy context to create the endorsement quote container by using one of the following methods:

- Make a POST request to the [CreateUpdateQuote](#) Connect API.
- Use the [Create Insurance Quote](#) invocable action.

Specify the `contextId` and `pricebookId` during the API call.

Creates an endorsement quote associated with the active policy. After the policyholder confirms the updated quote price, proceed with the next steps in the endorsement workflow.

3. Create or Update Quote (PATCH)—Configure and price the endorsement changes by making a PATCH request to the [CreateUpdateQuote](#) API using the `quoteId` and user inputs.

 **Note:** For multi-root policies, ensure pricing is recalculated across all root products to accurately compute aggregate totals and prorated term premiums.

4. Endorse Multi-Root Policy—Use the updated context to endorse the policy by using one of the following methods:

- Make a POST request to the [Multi-Root Policy Endorsement](#) Connect API.
- Use the [Endorse Multi-Root Policy](#) invocable action.
- Returns an Async Request ID in the response to track the job status.
- Creates a new version of the parent policy and all associated child policies.
- Recalculates standard and prorated premiums based on the endorsement effective date and rolls them up to the parent policy.
- Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.
- Generates transaction records that document the financial impact of the endorsement.

Renew Multi-Root Policy

Renew a multi-root policy to create a new policy term for the parent policy and all associated child policies. The parent–child policy structure is preserved in the renewed term.

Make sure you have these details before renewing a multi-root policy:

- The Policy ID of the active parent insurance policy to be renewed.
- The Pricebook ID required to generate the renewal quote.

1. Get Policy Context—Retrieve the policy context by using the policy ID through one of the following methods:

- Make a GET request to the [Insurance Policy Details](#) Connect API.
- Use the [Get Insurance Policy](#) invocable action.

Returns the context required to generate the renewal quote.

2. Create Renewal Quote—Use the policy context to create the renewal quote by using one of the following methods:

- Make a POST request to the [CreateUpdateQuote](#) Connect API.
- Use the [Create Insurance Quote](#) invocable action and specify the `contextId` and `pricebookId`.

Creates a renewal quote associated with the active policy.

3. Modify Renewal Quote (Optional)—Configure changes to the renewal terms by making a PATCH request to the [CreateUpdateQuote](#) Connect API with the desired updates.

 **Note:** For multi-root policies, ensure pricing is recalculated across all root products to accurately compute aggregate totals and prorated term premiums.

4. Renew Multi-Root Policy—Use the updated context to renew the policy by using one of the following methods:
 - Make a POST request to the [Multi-Root Policy Renewal](#) Connect API.
 - Use the [Renew Multi-Root Policy](#) invocable action.
- Returns an Async Request ID in the response to track the job status.
- Creates a new parent policy record and corresponding child policy records for the new term.
- Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.
- Consolidates financial details at the parent policy level.

Create and Manage Insurance Policies Without Using Salesforce Quoting

Learn how to use the Insurance Policy APIs and invocable actions to manage the policy lifecycle when Salesforce quoting isn't used, or when a third-party system handles the quoting process.

[Endorse Policy](#)

Policy endorsement, also known as a Mid-Term Adjustment (MTA), is a process where you change an active insurance policy. The endorsement process changes the policy version and the premiums. This user journey shows you how to endorse a policy when Salesforce's quoting capabilities aren't used.

[Issue Policy](#)

Issue a new insurance policy when quoting is managed outside Salesforce, by a third-party system. This process results in the creation of a policy record in Salesforce, formally establishing coverage for the customer.

[Renew Policy](#)

Policy renewal extends insurance coverage for an additional term, creating a new policy record. This journey details renewing a policy without using Salesforce quoting, relying on the Rating API for pricing the new term.

[Create Out of Sequence Policy Endorsement](#)

Perform a backdated change on a policy in the past, current, or future policy term, even when another policy version with an effective date later than the endorsement date already exists. The impacted policy versions are canceled and replaced with new, repriced versions that include the change. The premium impact for each affected policy term is captured in an individual out-of-sequence endorsement transaction.

[Issue Multi-Root Policy](#)

Creates a new multi-root insurance policy when quoting is managed outside Salesforce by a third-party system. This process uses the Rating API to generate a priced context and create a parent-child policy structure without persisting a quote record.

[Endorse Multi-Root Policy](#)

Perform a mid-term adjustment on an active multi-root policy when pricing is handled through the Rating API instead of a Salesforce Quote. This process bypasses the creation of a Quote record while ensuring accurate pricing and versioned updates to the parent and child policies.

[Renew Multi-Root Policy](#)

Creates a new policy term for a multi-root policy when pricing is handled through the Rating API instead of a quote. This process bypasses the creation of a quote record while generating new parent and child policy records for the renewed term.

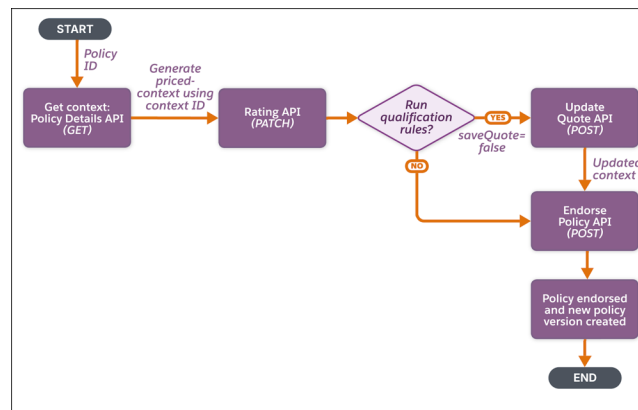
Endorse Policy

Policy endorsement, also known as a Mid-Term Adjustment (MTA), is a process where you change an active insurance policy. The endorsement process changes the policy version and the premiums. This user journey shows you how to endorse a policy when Salesforce's quoting capabilities aren't used.

Make sure you have these details.

- Policy ID of the active insurance policy to be endorsed.
- Details of the modifications to be made, such as changes to coverages, limits, and deductibles.

This diagram outlines the sequence of endorsing a policy, where any changes are initially processed through the Rating API to generate a price-adjusted context. The context contains the calculated premium and other relevant details that are required to endorse the policy. This process bypasses the need for a formal Salesforce Quote record for pricing.



1. Get Policy Context—Use the policy ID to get the policy context either by:

- Making a GET request on the [Insurance Policy Details](#) Connect API, or
- Using the [Get Insurance Policy](#) invocable action.

2. Generate Price-adjusted Context—Use the context ID and make a PATCH request on the [Insurance Product Ratings](#) Connect API.

The price-adjusted context record contains the premium impact of the proposed modifications.

3. (Optional) Run Qualification Rules—If customers want to determine the policyholder's eligibility for a product, run the qualification rules either by:

- Making a POST request on the [Insurance Quote](#) Connect API, or
- Using the [Create Insurance Quote](#) invocable action.

Set the `saveQuote` parameter to `false`. It doesn't save the quote and returns an updated context with the eligibility details.

4. Endorse Policy—Use the retrieved context to endorse a policy either by:

- Making a POST request on the [Endorse Insurance Policy](#) Connect API, or
- Using the [Endorse Insurance Policy](#) invocable action.

You can change the context fields, such as the policy name or policy number, directly during the API call. You can also provide a name for the transaction record that's created along with the endorsement. This transaction record captures the financial details or the audit information related to the MTA.

A new endorsed policy version is created.

Expected Results:

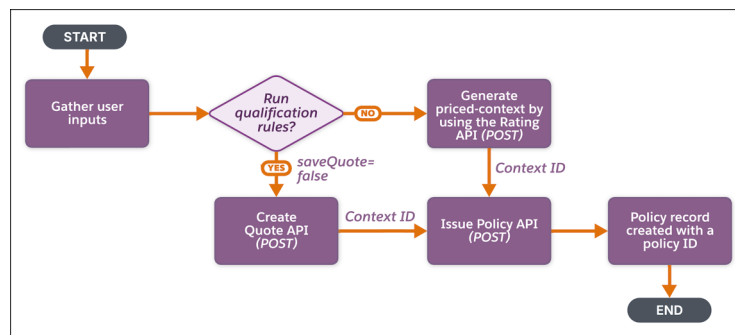
- A new policy version record is created in Salesforce, which includes the endorsed changes.
- A new policy version is created with an updated effective-from date. The effective-to date of the existing policy version is adjusted to an end date just before the new version begins, ensuring continuous but non-overlapping coverage.
- The new policy version has the updated premium information, including the new standard premium and the prorated term premium, accurately calculated and stored.
- A transaction record is created and linked to the endorsement activity.

Issue Policy

Issue a new insurance policy when quoting is managed outside Salesforce, by a third-party system. This process results in the creation of a policy record in Salesforce, formally establishing coverage for the customer.

Use the Rating API to generate a price-adjusted context. This price-adjusted context contains the calculated premium and other relevant details that are required to issue a policy. This process bypasses the need for a formal Salesforce Quote record for pricing.

This diagram outlines the sequence of issuing a policy without using Salesforce quote, rather by using the Rating API to generate a price-adjusted context that's used to issue a policy.



1. (Optional) Run Qualification Rules—If customers want to determine the policyholder's eligibility for a product, run the qualification rules either by:

- Making a POST request on the [Insurance Quote](#) Connect API, or
- Using the [Create Insurance Quote](#) invocable action.

Set the `saveQuote` parameter to `false`. It doesn't save the quote and returns an updated context with the eligibility details.

2. Generate Price-adjusted Context—Make a POST request on the [Insurance Product Ratings](#) Connect API.

You must include `additionalFields` in the request payload with either a `QuoteAccount` or an `Opportunity ID` that's linked to an account to generate the context.

3. Issue Policy—Use the price-adjusted context record to issue a policy either by:

- Making a POST request on the [Issue Insurance Policy](#) Connect API, or
- Using the [Issue Insurance Policy](#) invocable action.

You can override the context data, such as policy name, policy number, effective-from date, effective-to date, and any other additional fields to provide the values for the custom fields during the API call. You can also specify a name for the transaction record that's created as part of the policy issuance process. This transaction record is used to track an audit or as a financial marker.

This request creates a new policy record.

Expected Results:

- A new policy record is created and saved in Salesforce. The policy record includes the details, such as the policy name, number (if provided or created during the API call), policy start date, the coverage you chose, and the price.
- Any custom field values that are provided during the API call are saved to the new policy record.
- A transaction record is created and linked to the policy.

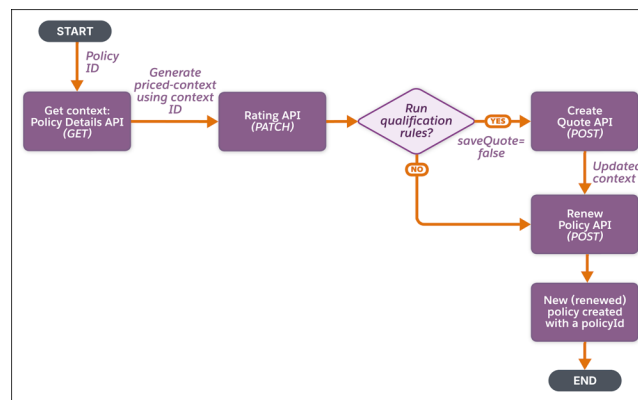
Renew Policy

Policy renewal extends insurance coverage for an additional term, creating a new policy record. This journey details renewing a policy without using Salesforce quoting, relying on the Rating API for pricing the new term.

Make sure you have a valid policy ID of the active insurance policy that's due for renewal.

Use the Rating API to generate a price-adjusted context. This price-adjusted context contains the calculated premium and other relevant details that are required to renew the policy. This process bypasses the need for a formal Salesforce Quote record for pricing.

This diagram outlines the sequence of renewing a policy, where any changes are initially processed through the Rating API to generate a price-adjusted context. The context contains the calculated premium and other relevant details that are required to renew the policy. This process bypasses the need for a formal Salesforce Quote record for pricing.



- 1. Get Policy Context**—Use the policy ID to get the policy context either by:
 - Making a GET request on the [Insurance Policy Details](#) Connect API, or
 - Using the [Get Insurance Policy](#) invocable action.
- 2. Generate Price-adjusted Context**—Make a PATCH request on the [Insurance Product Ratings](#) Connect API. The price-adjusted context record contains the premium impact of the proposed modifications.
- 3. (Optional)Run Qualification Rules**—If customers want to determine the policyholder's eligibility for a product, run the qualification rules either by:
 - Making a POST request on the [Insurance Quote](#) Connect API, or
 - Using the [Create Insurance Quote](#) invocable action.

Set the `saveQuote` parameter to `false`. It doesn't save the quote and returns an updated context with the eligibility details.
- 4. Renew Policy**—Use the retrieved context to create a new policy record for the renewal term either by:
 - Making a POST request on the [Renew Insurance Policy](#) Connect API, or
 - Using the [Renew Insurance Policy Action](#) invocable action.

You can change or confirm the policy name, policy number, and effective start and end dates for the renewed policy directly during the API call. Additionally, you can also specify a name for the transaction record that's created along with the renewal record. This transaction record captures the financial details or the audit information related to the renewal record.

Expected Results:

- A brand new policy record is created in Salesforce, representing the coverage for the new policy term. The new policy record includes the details, such as any changed terms, coverages, prices, and the new start and end dates.
- A transaction record is created and linked to the new policy.

Create Out of Sequence Policy Endorsement

Perform a backdated change on a policy in the past, current, or future policy term, even when another policy version with an effective date later than the endorsement date already exists. The impacted policy versions are canceled and replaced with new, repriced versions that include the change. The premium impact for each affected policy term is captured in an individual out-of-sequence endorsement transaction.

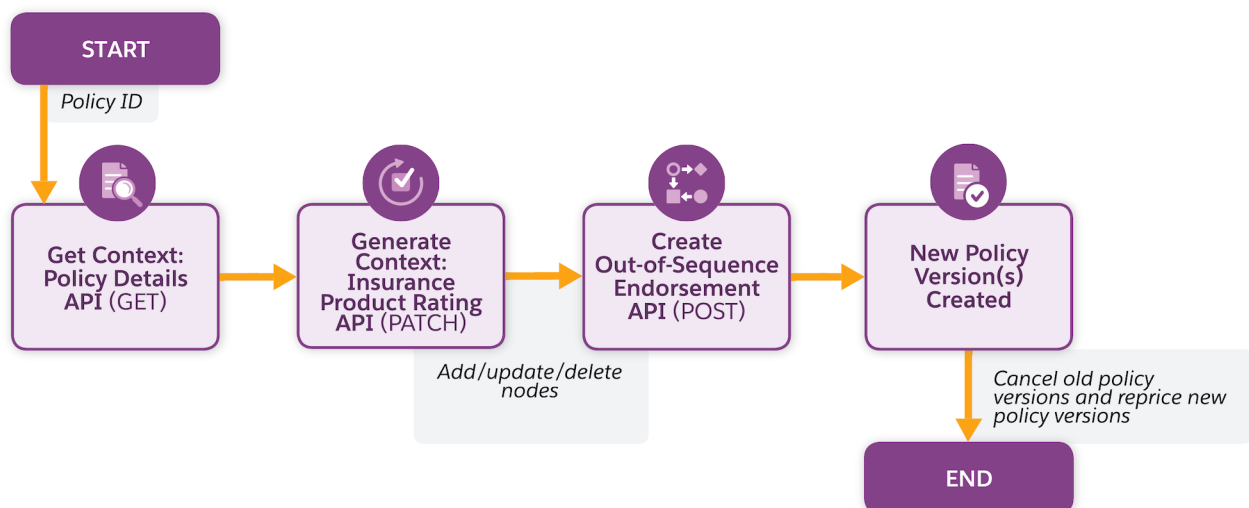
Make sure you have the following details.

- Policy ID of the active insurance policy to be endorsed.
- Effective date for the endorsement must fall within a valid policy term.
- (Optional) FieldSetIds to copy selected fields (policy, asset, participant, coverage) from cancelled versions to the new versions.

Example

If a policy has three active versions (V1, V2, and V3) within a single term, and you need to make a change from an effective date which lies in the tenure of V2. You can prepare the context for out-of-sequence endorsement using the Rating API. Get the context from policy V2 and pass this context to the patch rating API. Make additions, deletions, or updates to generate an updated context and then pass this context to out-of-sequence endorsement request.

This diagram outlines the sequence for creating an out-of-sequence policy endorsement without using a quote.



1. Get Policy Context—Use the policy ID to get the context either by:

- Making a GET request on the [Insurance Policy Details](#) Connect API, or
- Using the [Get Insurance Policy](#) invocable action.

2. Generate Price-Adjusted Context—Use the context ID and make a PATCH request on the [Insurance Product Ratings](#) Connect API.

- a. Modify coverages, limits, or deductibles in the payload.

The returned context includes updated premiums and tax details.

3. Submit out of sequence endorsement request—Use the retrieved context to endorse a policy either by:

- Making a POST request on the Out-of-Sequence Endorsement Connect API, or
- Using the Create Out of Sequence Policy Endorsement invocable action.

Specify the `policyId`, `contextId`, `effectiveDate` and transaction details during the API call.

The request returns a `jobIdentifier` that you can use to track the async process through a custom report.

- Identifies and fetches the impacted policy versions.
- Locks impacted policy versions to prevent any other operation on the impacted policies when the OOSE request is in progress.
- Applies deltas (add, update, delete) across all impacted versions and reprices the context.
- Unlocks the impacted policy versions and sends a success or failure notification.
- Publishes a platform event that subscribers can use for tracking status.

Expected Results:

- Creates a new policy starting from the effective date of the OOSE endorsement.
- Cancels impacted future versions and replaces them with updated versions.
- Sets the effective-to date of older versions to `endorsementDate - 1` and resets the term premiums and taxes to zero.
- Recalculates premiums and taxes on the new versions and prorates them based on the endorsement date.
- Creates a transaction record for each policy term and links it to the original policy version of that term for auditing and financial tracking.

Issue Multi-Root Policy

Creates a new multi-root insurance policy when quoting is managed outside Salesforce by a third-party system. This process uses the Rating API to generate a priced context and create a parent-child policy structure without persisting a quote record.

Make sure you have these details before issuing a multi-root policy:

- Inputs provided to the Issue Multi-Root Policy API in addition to the context ID. These inputs include policy-level details such as policy name, effective start and end dates.
 - The context is priced using the Rating API before calling the Issue Multi-Root Policy API.
- 1. (Optional) Run Qualification Rules**—To determine eligibility before pricing, run qualification rules by making a POST request on the [CreateUpdateQuote](#) API with `saveQuote=false`.
Returns an updated context that contains eligibility results without saving a Quote record.
 - 2. Generate Price-adjusted Context**—Generate a price-adjusted context by using one of the following methods:
 - Make a POST request to the [Insurance Product Rating](#) Connect API.
 - Make a POST or PATCH request to the [CreateUpdateQuote](#) API with `saveQuote=false`.
 Returns a price-adjusted context that includes calculated premiums required for policy issuance.
 - 3. Issue Multi-Root Policy**—Use the context details, specifically the context ID, along with user inputs to issue a multi-root policy by using one of the following methods:
 - Making a POST request on the [Multi-Root Policy Issuance](#) Connect API.

- Use the [Issue Multiroot Policy](#) invocable action.
- Returns an Async Request ID as a response to track the job status.
- Creates a multi-root policy with the following structure:
 - A parent policy that represents the policy lifecycle.
 - One or more insured child policies are created for each root product bundle and linked to the parent policy.
- Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.

Endorse Multi-Root Policy

Perform a mid-term adjustment on an active multi-root policy when pricing is handled through the Rating API instead of a Salesforce Quote. This process bypasses the creation of a Quote record while ensuring accurate pricing and versioned updates to the parent and child policies.

Make sure you have these details before endorsing a multi-root policy:

- The Policy ID of the active parent insurance policy to be endorsed.
 - Details of the modifications to apply to the policy.
1. Get Policy Context—Retrieve the policy context by using the policy ID through one of the following methods:
 - Make a GET request to the [Insurance Policy Details](#) Connect API.
 - Use the [Get Insurance Policy](#) invocable action.

Returns the context required to evaluate and price the endorsement changes.
 2. Generate Price-Adjusted Context—Generate a priced context for the proposed changes by making a PATCH request to the [Insurance Product Rating](#) Connect API using the context ID.

Returns a price-adjusted context that contains the premium impact of the proposed modifications.
 3. (Optional) Run Qualification Rules—Determine eligibility for the proposed changes by making a POST request to the [CreateUpdateQuote](#) Connect API with saveQuote=false.

Returns an updated context containing eligibility results without saving a Quote record.
 4. Endorse Multi-Root Policy—Use the updated context to endorse the policy by using one of the following methods:
 - Make a POST request to the [Multi-Root Policy Endorsement](#) Connect API.
 - Use the [Endorse Multi-Root Policy](#) invocable action.
 - Returns an Async Request ID in the response to track the job status.
 - Creates new versions of the parent policy and all associated child policies with updated effective dates and premiums.
 - Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.
 - Completes the operation asynchronously and notifies the user upon success or failure.

Renew Multi-Root Policy

Creates a new policy term for a multi-root policy when pricing is handled through the Rating API instead of a quote. This process bypasses the creation of a quote record while generating new parent and child policy records for the renewed term.

Make sure you have these details before renewing a multi-root policy:

- The Policy ID of the active parent insurance policy to be renewed.

1. Get Policy Context—Retrieve the policy context by using the policy ID through one of the following methods:
 - Make a GET request to the [Insurance Policy Details](#) Connect API.
 - Use the [Get Insurance Policy](#) invocable action.Returns the context required to price and renew the policy.
2. Generate Price-Adjusted Context—Generate a priced context for the new term by making a PATCH request to the [Insurance Product Rating](#) Connect API using the context ID.
Returns a price-adjusted context that contains the premium calculation for the renewed term.
3. (Optional) Run Qualification Rules—Determine eligibility for renewal by making a POST request to the [CreateUpdateQuote](#) Connect API with saveQuote=false.
Returns an updated context containing eligibility results without saving a Quote record.
4. Renew Multi-Root Policy—Use the updated context to renew the policy by using one of the following methods:
 - Make a POST request to the [Renew Multi-Root Policy](#) Connect API.
 - Use the [Renew Multi-Root Policy](#) invocable action.
 - Returns an Async Request ID in the response to track the job status.
 - Creates new parent and child policy records asynchronously for the new term.
 - Creates a corresponding transaction record for each child policy. In addition, creates a parent policy transaction that represents the rolled-up premium from all associated child policy transactions.
 - Creates transaction records associated with the renewed policy term.

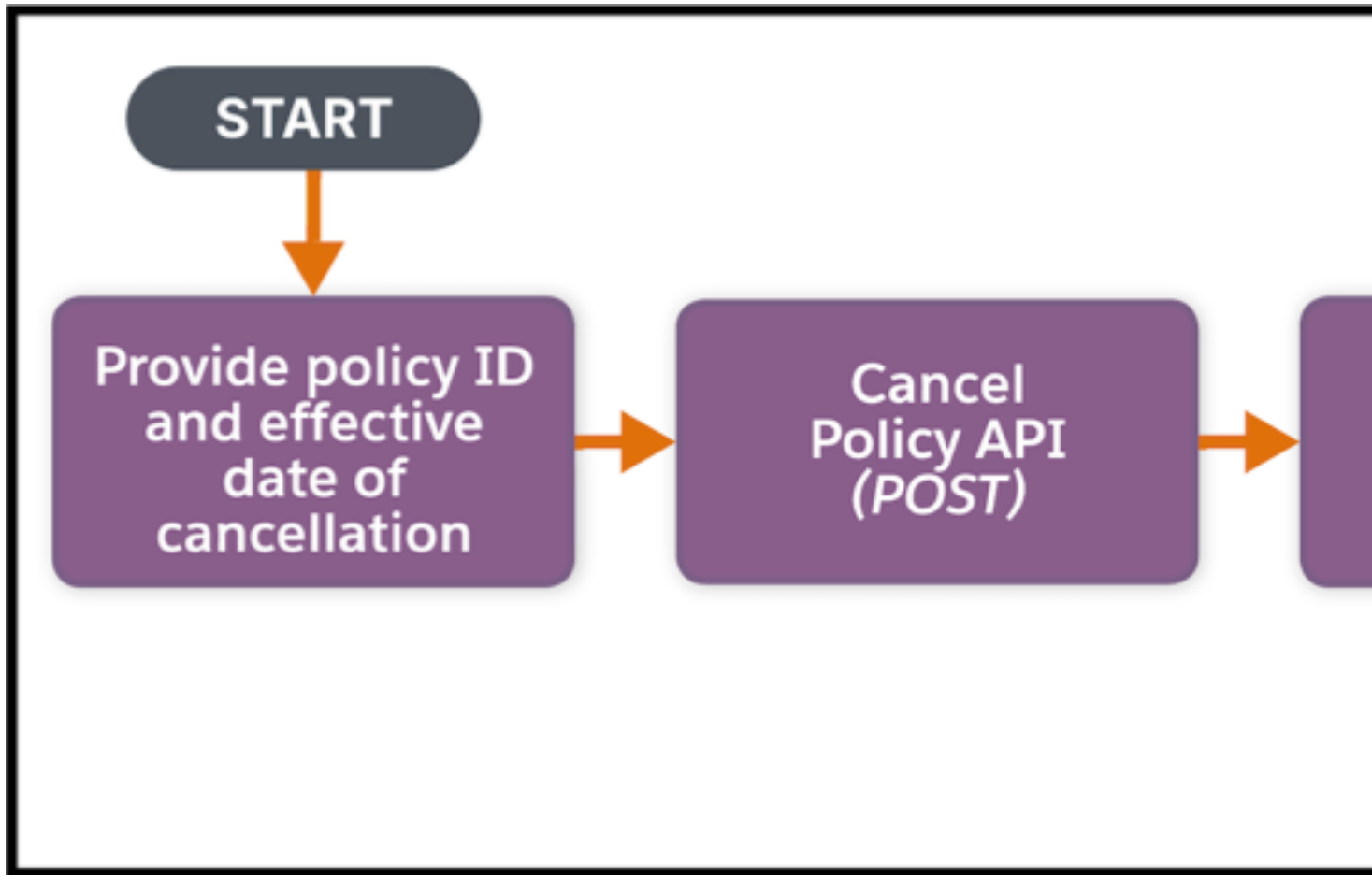
Cancel Policy

The policy cancellation process addresses the termination of an insurance policy before its scheduled expiration date. Common reasons for canceling a policy include not paying the premiums or a direct cancellation request from the policyholder.

Make sure you have these details.

- Policy ID of the active insurance policy to be canceled.
- Effective date of cancellation. This date determines when coverage ceases and is a required input for the policy cancellation API.

This diagram outlines a sequence of operations involved in canceling a policy.



1. Use the policy ID to cancel the policy either by:
 - Making a POST request on the [Cancel Insurance Policy Connect API](#), or
 - Using the [Cancel Insurance Policy](#) invocable action.

You can specify a name for the transaction record that's created as part of the cancellation process. The transaction record is used for tracking financial implications, such as refunds.

This process returns the details of the policy versions that are canceled, which include the current version and any future-dated versions. Details of any applicable refunds, such as premium refunds and tax refunds, calculated by the system based on the cancellation date and policy terms are also returned by the API.

Expected Results:

- The specified policy is marked as `Canceled` in Salesforce.
- The policy's effective-to-date field is updated to the day immediately preceding the effective date of cancellation that's specified during the API call.
- Any future-dated policy versions associated with the canceled policy are also automatically canceled.
- Information regarding applicable premium and tax refunds are shown in the API response, which can then be used for downstream financial processing.
- A transaction record is created and linked to the cancellation event.

CHAPTER 6 Insurance Product Administration

In this chapter ...

- [Insurance Product Administration Business APIs](#)
- [Insurance Product Administration Invocable Actions](#)
- [Insurance Product Administration Metadata API Types](#)

Use the Insurance Product Administration developer resources to define the products you sell to customers, including their classifications, categories, and attributes. Create customized price adjustment methods and pricing procedures. Define rules that determine how your products behave when they're quoted, sold, and serviced.

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Insurance Product Administration Business APIs

Design, manage, and administer insurance products to meet the business requirements.

This table lists the available Insurance Product Administration resources.

Resource	Description
/connect/insurance/underwriting-rules (POST)	Create an underwriting rule for a root product based on the selected state definition, transition, and object name.
/connect/insurance/underwriting-rule-group-status/uwrGroupId (PATCH)	Update the status of an underwriting rule group.
/connect/insurance/underwriting-rules/underwritingRuleId (PATCH)	Update an existing underwriting rule for a root product.

Resources

Learn more about the available Insurance Product Administration API resources.

Requests

Learn more about the available Insurance Product Administration requests.

Responses

Learn more about the available response bodies of Insurance Product Administration APIs.

Resources

Learn more about the available Insurance Product Administration API resources.

Insurance Underwriting Rules (POST)

Create an underwriting rule for a root product based on the selected state definition, transition, and object name.

Insurance Underwriting Rule Group Status (PATCH)

Update the status of an underwriting rule group.

Insurance Underwriting Rules (PATCH)

Update an existing underwriting rule for a root product.

Insurance Underwriting Rules (POST)

Create an underwriting rule for a root product based on the selected state definition, transition, and object name.

Resource

```
/connect/insurance/underwriting-rules
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/underwriting-rules
```

Available version

63.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "name": "Automatically Approve Quote",
  "apiName": "approveQuoteBasedonComprehensiveCoverageDeductibles",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "status": "Draft",
  "description": "Rule description",
  "underwritingRuleGroup": {
    "underwritingRuleGroupId": "1KQ000000MKJIK", // Optional,
    // if underwritingRuleGroupId exist, then use it as parent instead of create
    a new record.
    "stageTransitionId": "0jjDR00000000VGAAA", // State Transition id.
    "stageTransitionName": "Draft to Approved",
    "objectType": "Quote",
    "recordType": "--Master--",
    "fromStage": "Draft",
    "toStage": "Approved",
    "rootProductId": "01tDR000000EMslyAG" // Auto Gold.
    "criteriaExpression": "1" // Exmple "(1 AND 2) OR 3" if more than 1 rules.
  }
  "productPath": "01tDR000000EMslyAG/11BDR00000000T72AI", // "Auto Gold/Auto
Vehicle"
  "sequence": 1,
  "ruleCriteria": [
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslyAG", // Auto Gold.
      "productRelatedComponent" : // Need More details on it.
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
      "sourceDataType": "String",

```

```

    "sourceValues": [
      "11BDR00000000T72AI" // Auto Vehicle.
    ],
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Comprehensive Deductible",
          "type": "Attribute",
          "attributePicklistValueId": null,
          "attributeId": "0tjDR00000000UaYAI",
          "dataType": "Number",
          "values": [
            "$5000"
          ]
        }
      ],
      "rootObjectId": "01tDR0000000EMs1YAG", // Auto Bundle
      "productRelatedComponent" : // Need More details on it.
      "criteriaIndex": 2,
      "sourceContextTagName": "Product",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "01tDR0000000EMsoYAG" // Comprehensive Product.
      ]
    },
    // EvaluationSuccessTaskGroup and EvaluationFailureTaskGroup are optional
    // Fulfillment plan design time entities will be created.
    "evaluationSuccessTaskGroup": // Create FulfillmentStepDefinitionGroup records

    {
      "name": "QuoteApproveSuccessTaskGroup",
      "tasks": [ // create FulfillmentStepDefinition records
        {
          "name": "KYC check flow", // FulfillmentStepDefinition API Names
          "type": "AutoTask", // FulfillmentStepDefinition type
          "flowApiName": "KYC check flow", // Flow API Names
        }
      ],
    },
    "evaluationFailureTaskGroup": //Similar to EvaluationSuccessTaskGroup
    {
      "name": "QuoteApproveFailureTaskGroup",
      "tasks": [
        {
          "name": "Vehicle ineligible",
          "type": "AutoTask",
          "flowApiName": "Mail Notification - Vehicle ineligible",
        }
      ]
    }

```



```

    ]
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
apiName	String	Unique API name of the underwriting rule.	Required	63.0
description	String	Description of the underwriting rule	Optional	63.0
effectiveFromDate	String	Date from which the underwriting rule is effective.	Required	63.0
effectiveToDate	String	Date until which the underwriting rule is effective.	Optional	63.0
evaluationFailedTaskGroup	Evaluation Task Group Input[]	Input details for failed evaluation tasks.	Optional	63.0
evaluationSuccessfulTaskGroup	Evaluation Task Group Input[]	Input details for successful evaluation tasks.	Optional	63.0
name	String	Name of the underwriting rule	Required	63.0
productPath	String	Name of the product path.	Required	63.0
ruleCriteria	Insurance Rule Criteria Input[]	Rule criteria for the underwriting rule.	Required	63.0
status	String	Status of the underwriting rule. Possible values are: • Active • Inactive • Draft	Optional	63.0
underwritingRuleGroup	Insurance Underwriting Rule Group Input[]	The UnderwritingRuleGroup input data.	Required	63.0

Response body for POST[Insurance Create Underwriting Rules](#)**Insurance Underwriting Rule Group Status (PATCH)**

Update the status of an underwriting rule group.

Resource

```

/connect/insurance/underwriting-rule-group-status/uwrGroupId

```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/underwriting-rule-group-status/urwGroupId
```

Available version

63.0

HTTP methods

PATCH

Request body for PATCH

JSON example

```
{
  "status": "Active"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
status	String	The status of the underwriting rule group.	Required	63.0

Response body for PATCH

[Insurance Underwriting Rule Group Update Status](#)

Insurance Underwriting Rules (PATCH)

Update an existing underwriting rule for a root product.

Resource

```
/connect/insurance/underwriting-rules/underwritingRuleId
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v66.0/connect/insurance/underwriting-rules/underwritingRuleId
```

Available version

63.0

HTTP methods

PATCH

Request body for PATCH

JSON example

```
{
  "name": "Automatically Approve Quote",
  "apiName": "approveQuoteBasedonComprehensiveCoverageDeductibles",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "productPath": "01tDR000000EMsLYAG/11BDR00000000T72AI", // "Auto Gold/Auto Vehicle"
```

```

    "ruleCriteria": [ // ruleCriteria will be replaced as a whole. Updating only
one condition is not supported
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGYAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG", // Auto Gold.
      "productRelatedComponent" : // Need More details on it.
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "11BDR00000000T72AI" // Auto Vehicle.
      ]
    },
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Comprehensive Deductible",
          "type": "Attribute",
          "attributePicklistValueId": null,
          "attributeId": "0tjDR00000000UaYAI",
          "dataType": "Number",
          "values": [
            "$5000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG", // Auto Bundle
      "productRelatedComponent" : // Need More details on it.
      "criteriaIndex": 2,
      "sourceContextTagName": "Product",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "01tDR000000EMsoYAG" // Comprehensive Product.
      ]
    }
  ]
}

```

```

],
// EvaluationSuccessTaskGroup and EvaluationFailureTaskGroup are optional
// Fulfillment plan design time entities will be created.
"EvaluationSuccessTaskGroup": // Create FulfillmentStepDefinitionGroup records

{
  "name": "QuoteApproveSuccessTaskGroup",
  "tasks": [ // create FulfillmentStepDefinition records
    {
      "name": "KYC check flow", // FulfillmentStepDefinition API Names
      "type": "AutoTask", // FulfillmentStepDefinition type
      "flowApiName": "KYC check flow", // Flow API Names
    }
  ]
},
"EvaluationFailureTaskGroup": //Similar to EvaluationSuccessTaskGroup
{
  "name": "QuoteApproveFailureTaskGroup",
  "tasks": [
    {
      "name": "Vehicle ineligible",
      "type": "AutoTask",
      "flowApiName": "Mail Notification - Vehicle ineligible",
    }
  ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
description	String	Description of the underwriting rule.	Optional	63.0
effectiveFromDate	String	Date from which the underwriting rule is effective.	Optional	63.0
effectiveToDate	String	Date until which the underwriting rule is effective.	Optional	63.0
evaluationFailureTaskGroup	Evaluation Task Group Input[]	Input details for failed evaluation tasks.	Optional	63.0
evaluationSuccessTaskGroup	Evaluation Task Group Input[]	Input details for successful evaluation tasks.	Optional	63.0
name	String	Name of the underwriting rule.	Optional	63.0
productPath	String	Name of the product path.	Optional	63.0
ruleCriteria	Insurance Rule Criteria Input[]	Rule criteria for the underwriting rule.	Required	63.0

Name	Type	Description	Required or Optional	Available Version
status	String	Status of the underwriting rule. Possible values are: • Active • Inactive • Draft	Optional	63.0

Response body for PATCH[Insurance Update Underwriting Rules](#)

Requests

Learn more about the available Insurance Product Administration requests.

[Evaluation Task Group Input](#)

Input representation details of the evaluation task group.

[Evaluation Task Input](#)

Input representation details of the evaluation task.

[Insurance Create Underwriting Rules Input](#)

Create Underwriting Rule Input Representation

[Insurance Rule Condition Input](#)

Input representation details of insurance rule condition.

[Insurance Rule Criteria Input](#)

Input representation details of the insurance rule criteria.

[Insurance Underwriting Rule Group Input](#)

Input representation details of the insurance underwriting rule group.

[Insurance Underwriting Rule Group Update Status Input](#)

Input representation of the request to update insurance underwriting rule group status.

[Insurance Update Underwriting Rules Input](#)

Update Underwriting Rule Input Representation

Evaluation Task Group Input

Input representation details of the evaluation task group.

JSON example

```
// EvaluationSuccessTaskGroup and EvaluationFailureTaskGroup are optional
// Fulfillment plan design time entities will be created.
"evaluationSuccessTaskGroup": // Create FulfillmentStepDefinitionGroup records
{
  "name": "QuoteApproveSuccessTaskGroup",
```

```
"tasks": [ // create FulfillmentStepDefinition records
{
  "name": "KYC check flow", // FulfillmentStepDefinition API Names
  "type": "AutoTask", // FulfillmentStepDefinition type
  "flowApiName": "KYC check flow", // Flow API Names
}
],
"evaluationFailureTaskGroup": //Similar to EvaluationSuccessTaskGroup
{
  "name": "QuoteApproveFailureTaskGroup",
  "tasks": [
    {
      "name": "Vehicle ineligible",
      "type": "AutoTask",
      "flowApiName": "Mail Notification - Vehicle ineligible",
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
name	String	Name of the evaluation task group.	Required	63.0
tasks	Evaluation Task Input[]	List of tasks to be executed for this evaluation task group.	Required	63.0

Evaluation Task Input

Input representation details of the evaluation task.

JSON example

```
"tasks": [
{
  "name": "Vehicle ineligible",
  "type": "AutoTask",
  "flowApiName": "Mail Notification - Vehicle ineligible",
}
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
flowApiName	String	API name of the associated flow when the task type is AutoTask..	Required	63.0
name	String	Name of the evaluation task.	Required	63.0

Name	Type	Description	Required or Optional	Available Version
type	String	Type of the evaluation task. Possible values are: • AutoTask • Manual	Required	63.0

Insurance Create Underwriting Rules Input

Create Underwriting Rule Input Representation

JSON example

```
{
  "name": "Automatically Approve Quote",
  "apiName": "approveQuoteBasedonComprehensiveCoverageDeductibles",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "status": "Draft",
  "description": "Rule description",
  "underwritingRuleGroup": {
    "underwritingRuleGroupId": "1KQ000000MKJIK", // Optional,
    // if underwritingRuleGroupId exist, then use it as parent instead of create a
    new record.
    "stageTransitionId": "0jjDR00000000VGAAA", // State Transition id.
    "stageTransitionName": "Draft to Approved",
    "objectType": "Quote",
    "recordType": "--Master--",
    "fromStage": "Draft",
    "toStage": "Approved",
    "rootProductId": "01tDR000000EMslyAG" // Auto Gold.
    "criteriaExpression": "1" // Exmple "(1 AND 2) OR 3" if more than 1 rules.
  }
  "productPath": "01tDR000000EMslyAG/11BDR00000000T72AI", // "Auto Gold/Auto Vehicle"

  "sequence": 1,
  "ruleCriteria": [
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGyAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ],
  "rootObjectId": "01tDR000000EMslYAG", // Auto Gold.
  "productRelatedComponent" : // Need More details on it.
  "criteriaIndex": 1,
  "sourceContextTagName": "ProductBasedOn",
  "sourceOperator": "Equals",
  "sourceDataType": "String",
  "sourceValues": [
    "11BDR00000000T72AI" // Auto Vehicle.
  ]
},
{
  "conditions": [
    {
      "contextTagName": "SalesTransactionItemAttribute",
      "operator": "LessThan",
      "conditionIndex": 1,
      "attributeName": "Comprehensive Deductible",
      "type": "Attribute",
      "attributePicklistValueId": null,
      "attributeId": "0tjDR00000000UaYAI",
      "dataType": "Number",
      "values": [
        "$5000"
      ]
    }
  ],
  "rootObjectId": "01tDR000000EMslYAG", // Auto Bundle
  "productRelatedComponent" : // Need More details on it.
  "criteriaIndex": 2,
  "sourceContextTagName": "Product",
  "sourceOperator": "Equals",
  "sourceDataType": "String",
  "sourceValues": [
    "01tDR000000EMsoYAG" // Comprehensive Product.
  ]
}
],
// EvaluationSuccessTaskGroup and EvaluationFailureTaskGroup are optional
// Fulfillment plan design time entities will be created.
"evaluationSuccessTaskGroup": // Create FulfillmentStepDefinitionGroup records
{
  "name": "QuoteApproveSuccessTaskGroup",
  "tasks": [ // create FulfillmentStepDefinition records
    {
      "name": "KYC check flow", // FulfillmentStepDefinition API Names
      "type": "AutoTask", // FulfillmentStepDefinition type
      "flowApiName": "KYC check flow", // Flow API Names
    }
  ]
},
"evaluationFailureTaskGroup": //Similar to EvaluationSuccessTaskGroup
{

```



```

    "name": "QuoteApproveFailureTaskGroup",
    "tasks": [
      {
        "name": "Vehicle ineligible",
        "type": "AutoTask",
        "flowApiName": "Mail Notification - Vehicle ineligible",
      }
    ]
  }
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
apiName	String	Unique API name of the underwriting rule.	Required	63.0
description	String	Description of the underwriting rule	Optional	63.0
effectiveFromDate	String	Date from which the underwriting rule is effective.	Required	63.0
effectiveToDate	String	Date until which the underwriting rule is effective.	Optional	63.0
evaluationFailureTaskGroup	Evaluation Task Group Input[]	Input details for failed evaluation tasks.	Optional	63.0
evaluationSuccessTaskGroup	Evaluation Task Group Input[]	Input details for successful evaluation tasks.	Optional	63.0
name	String	Name of the underwriting rule	Required	63.0
productPath	String	Name of the product path.	Required	63.0
ruleCriteria	Insurance Rule Criteria Input[]	Rule criteria for the underwriting rule.	Required	63.0
status	String	Status of the underwriting rule. Possible values are: • Active • Inactive • Draft	Optional	63.0
underwritingRuleGroup	Insurance Underwriting Rule Group Input[]	The UnderwritingRuleGroup input data.	Required	63.0

Insurance Rule Condition Input

Input representation details of insurance rule condition.

JSON example

```

"conditions": [
  {
    "contextTagName": "SalesTransactionItemAttribute",
    "operator": "LessThan",
    "conditionIndex": 1,
    "attributeName": "Annual Milage",
    "attributePicklistValueId": null,
    "type": "Attribute",
    "attributeId": "0tjDR00000000VGyAY",
    "dataType": "Number",
    "values": [
      "50000"
    ]
  }
]

```

Properties

Name	Type	Description	Required or Optional	Available Version
attributeId	String	ID of the attribute.	Required	63.0
attributeName	String	Name of the attribute.	Required	63.0
attributePicklistValueId	String	ID of the attribute picklist value.	Required	63.0
conditionIndex	Integer	Index of the rule condition.	Required	63.0
contextTagName	String	Tag name of the context.	Required	63.0
dataType	String	Data type of the attribute.	Required	63.0
operator	String	Type of operator. Possible values are: - Equals - LessThan	Required	63.0
type	String	Type of rule condition.	Required	63.0
values	String[]	List of values.	Required	63.0

Insurance Rule Criteria Input

Input representation details of the insurance rule criteria.

JSON example

```

"ruleCriteria": [
  {
    "conditions": [
      {
        "contextTagName": "SalesTransactionItemAttribute",
        "operator": "LessThan",

```

```

    "conditionIndex": 1,
    "attributeName": "Annual Milage",
    "attributePicklistValueId": null,
    "type": "Attribute",
    "attributeId": "0tjDR00000000VGYAY",
    "dataType": "Number",
    "values": [
      "50000"
    ]
  }
]

```

Properties

Name	Type	Description	Required or Optional	Available Version
conditions	Insurance Rule Condition Input[]	List of rule conditions.	Optional	63.0
criteriaIndex	Integer	Index of the rule criteria.	Required	63.0
rootObjectId	String	ID of the root object.	Required	63.0
sourceContextTagName	String	Name of the source context tag.	Required	63.0
sourceDataType	String	Source data type. Possible values are: - String - Integer	Required	63.0
sourceOperator	String	Source operator. Possible value: Equals	Required	63.0
sourceValues	String[]	List of source values.	Required	63.0

Insurance Underwriting Rule Group Input

Input representation details of the insurance underwriting rule group.

JSON example

```

{
  "underwritingRuleGroup": {
    "underwritingRuleGroupId": "1KQ000000MKJIK", // Optional,
    // if underwritingRuleGroupId exist, then use it as parent instead of create a
    new record.
    "stageTransitionId": "0jjDR00000000VGAAA", // State Transition id.
    "stageTransitionName": "Draft to Approved",
    "objectType": "Quote",
    "recordType": "--Master--",
    "fromStage": "Draft",
    "toStage": "Approved",
    "rootProductId": "01tDR000000EMSlYAG" // Auto Gold.
  }
}

```

```
"criteriaExpression": "1" // Exmple "(1 AND 2) OR 3" if more than 1 rules.
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
criteriaExpression	String	Criteria Expression for the underwriting rule group. For example, (1 AND 2) OR 3.	Required	63.0
fromStage	String	Name of the stage the record is moving from.	Optional	63.0
objectType	String	Object type for which the underwriting rule group is defined.	Required	63.0
recordType	String	Record type associated with the object type for which the underwriting rule group is defined.	Optional	63.0
rootProductId	String	ID of the root product.	Required	63.0
rulesetStatus	String	Status of the dynamic ruleset. Possible values are: - Active - Inactive - Draft	Required	63.0
stageTransitionId	String	ID of the stage transition.	Required	63.0
stageTransitionName	String	Name of the source stage transition.	Optional	63.0
toStage	String	Name of the stage the record is moving to.	Optional	63.0
underwritingRuleGroupId	String	ID of the underwriting rule group.	Optional	63.0

Insurance Underwriting Rule Group Update Status Input

Input representation of the request to update insurance underwriting rule group status.

JSON example

```
{
  "status": "Active"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
status	String	The status of the underwriting rule group.	Required	63.0

Insurance Update Underwriting Rules Input

Update Underwriting Rule Input Representation

JSON example

```
{
  "name": "Automatically Approve Quote",
  "apiName": "approveQuoteBasedonComprehensiveCoverageDeductibles",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "productPath": "01tDR000000EMslyAG/11BDR00000000T72AI", // "Auto Gold/Auto Vehicle"

  "ruleCriteria": [ // ruleCriteria will be replaced as a whole. Updating only
one condition is not supported
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGyAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslyAG", // Auto Gold.
      "productRelatedComponent" : // Need More details on it.
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "11BDR00000000T72AI" // Auto Vehicle.
      ],
      {
        "conditions": [
          {
            "contextTagName": "SalesTransactionItemAttribute",
            "operator": "LessThan",
            "conditionIndex": 1,
            "attributeName": "Comprehensive Deductible",
            "type": "Attribute",
            "attributePicklistValueId": null,
            "attributeId": "0tjDR00000000UaYAI",
            "dataType": "Number",
            "values": [
              "$5000"
            ]
          }
        ]
      }
    ]
  }
}
```

```

    }
  ],
  "rootObjectId": "01tDR000000EMslYAG", // Auto Bundle
  "productRelatedComponent" : // Need More details on it.
  "criteriaIndex": 2,
  "sourceContextTagName": "Product",
  "sourceOperator": "Equals",
  "sourceDataType": "String",
  "sourceValues": [
    "01tDR000000EMsoYAG" // Comprehensive Product.
  ]
}
],
// EvaluationSuccessTaskGroup and EvaluationFailureTaskGroup are optional
// Fulfillment plan design time entities will be created.
"EvaluationSuccessTaskGroup": // Create FulfillmentStepDefinitionGroup records
{
  "name": "QuoteApproveSuccessTaskGroup",
  "tasks": [ // create FulfillmentStepDefinition records
    {
      "name": "KYC check flow", // FulfillmentStepDefinition API Names
      "type": "AutoTask", // FulfillmentStepDefinition type
      "flowApiName": "KYC check flow", // Flow API Names
    }
  ],
},
"EvaluationFailureTaskGroup": //Similar to EvaluationSuccessTaskGroup
{
  "name": "QuoteApproveFailureTaskGroup",
  "tasks": [
    {
      "name": "Vehicle ineligible",
      "type": "AutoTask",
      "flowApiName": "Mail Notification - Vehicle ineligible",
    }
  ]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
description	String	Description of the underwriting rule.	Optional	63.0
effectiveFromDate	String	Date from which the underwriting rule is effective.	Optional	63.0
effectiveToDate	String	Date until which the underwriting rule is effective.	Optional	63.0
evaluationFailureTaskGroup	Evaluation Task Group Input[]	Input details for failed evaluation tasks.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
evaluationSuccessfulGroup	Evaluation Task Group Input[]	Input details for successful evaluation tasks.	Optional	63.0
name	String	Name of the underwriting rule.	Optional	63.0
productPath	String	Name of the product path.	Optional	63.0
ruleCriteria	Insurance Rule Criteria Input[]	Rule criteria for the underwriting rule.	Required	63.0
status	String	Status of the underwriting rule. Possible values are: • Active • Inactive • Draft	Optional	63.0

Responses

Learn more about the available response bodies of Insurance Product Administration APIs.

[Insurance Create Underwriting Rules](#)

Output representation details of insurance create underwriting rules request.

[Insurance Underwriting Rule Group Update Status](#)

Output representation of insurance underwriting rule group update status request.

[Insurance Underwriting Rules Error](#)

Details of the error response.

[Insurance Update Underwriting Rules](#)

Output representation details of the insurance update underwriting rules request.

Insurance Create Underwriting Rules

Output representation details of insurance create underwriting rules request.

JSON example

This example shows a sample success response.

```
{
  "underwritingRuleGroupId": "u8ixx0000007MnhEAI",
  "underwritingRuleId": "11Oxx0000007MnhEAE",
  "apiName": "ExampleApiName",
  "isSuccess": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
apiName	String	API name of the underwriting rule.	Big, 63.0	63.0
errors	Insurance Underwriting Rules Error[]	List of errors encountered during the processing of the API request.	Big, 63.0	63.0
isSuccess	Boolean	Indicates whether the request is successful (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
underwritingRuleGroupId	String	ID of the underwriting rule group.	Big, 63.0	63.0
underwritingRuleId	String	ID of the underwriting rule.	Big, 63.0	63.0

Insurance Underwriting Rule Group Update Status

Output representation of insurance underwriting rule group update status request.

JSON example

```
{
  "success": "true",
  "underwritingRuleGroupId": "1KQxx0000004C92GAE"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
success	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
underwritingRuleGroupId	String	ID of the underwriting rule group whose ruleset status is updated.	Big, 63.0	63.0

Insurance Underwriting Rules Error

Details of the error response.

Property Name	Type	Description	Filter Group and Version	Available Version
code	String	Error code.	Big, 63.0	63.0
message	String	Message stating the reason for error, if any.	Big, 63.0	63.0

Insurance Update Underwriting Rules

Output representation details of the insurance update underwriting rules request.

JSON example

This example shows a sample success response.

```
{
  "underwritingRuleId": "11pxx0000005RE4AAM",
  "isActive": false,
  "isSuccess": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Insurance Underwriting Rules Error[]	List of errors encountered during the processing of the API request.	Big, 63.0	63.0
isActive	Boolean	Indicates whether the rule is active (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
isSuccess	Boolean	Indicates whether the request is successful (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
underwritingRuleId	String	ID of the underwriting rule.	Big, 63.0	63.0

Insurance Product Administration Invocable Actions

Learn more about the standard invocable actions available with Insurance Product Administration.

[Invoke Underwriting Rules Action](#)

Invoke the underwriting rules for different insurance records during the run time.

Invoke Underwriting Rules Action

Invoke the underwriting rules for different insurance records during the run time.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

`/services/data/v/actions/standard/invokeUnderwritingRules`

Formats

JSON

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
invokeUnderwritingRuleInputRep	Type Apex-defined Description Required. An Apex ConnectApi.InvokeUnderwritingRuleInputRep record that contains the details of the object records to which the underwriting rules must be applied.

Outputs

Output	Details
invokeUnderwritingRuleOputRep	Type Apex-defined Description An Apex ConnectApi.InvokeUnderwritingRuleOputRep record that contains the details of the object records to which the underwriting rules are applied.

Example

Sample Request

```
{
  "inputs": [
    {
      "invokeUnderwritingRuleInputRep": {
        "objectId": "0Zkxx000000001qCAA",
        "toStage": "Open"
      }
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
public class InsuranceUnderwritingRuleApexService implements Callable {

    private static final String METHOD_INVOKE_UW_RULE = 'invokeUnderwritingRule';
    private static final String OUTPUT_NAME = 'invokeUnderwritingRuleOputRep';

    /**
     * @description Implementation of Callable interface for OmniStudio
     * @param action Method name to invoke
     * @param args Map containing input, output, and options
     * @return Object result of the method invocation
     */
}
```

```

    */
    public Object call(String action, Map<String, Object> args) {
        System.debug('InsuranceQuoteApexService: call ' + action);
        Boolean response = false;
        Map<String, Object> input = (Map<String, Object>) args.get('input');
        Map<String, Object> output = (Map<String, Object>) args.get('output');
        Map<String, Object> options = (Map<String, Object>) args.get('options');

        if (METHOD_INVOKE_UW_RULE.equals(action)) {
            if (!input.containsKey('objectId') || input.get('objectId') == null) {
                throw new IllegalArgumentException('objectId is missing or null');
            }
            if (!input.containsKey('toStage') || input.get('toStage') == null) {
                throw new IllegalArgumentException('toStage is missing or null');
            }
            String objectId = (String)input.get('objectId');
            String toStage = (String)input.get('toStage');
            Object ruleResult = invokeUWRules(objectId, toStage);
            output.put('ruleResult', ruleResult);
            response = true;
        }
        return response;
    }

    public static Object invokeUWRules(String objectId, String toStage) {
        System.debug('InsuranceUnderwritingRuleApexService: objectId:' + objectId + ',
toStage : ' + toStage);

        Object outputRep;

        try {

            System.debug('Calling Invocable Action: invokeUnderwritingRules');

            // Create and configure the invocable action
            Invocable.Action action =
            Invocable.Action.createStandardAction('invokeUnderwritingRules');

            // Set required invocation parameters
            ConnectApi.invokeUnderwritingRuleInputRep inputRep = new
            ConnectApi.invokeUnderwritingRuleInputRep();
            inputRep.objectId = objectId;
            inputRep.toStage = toStage;
            action.setInvocationParameter('invokeUnderwritingRuleInputRep', inputRep);

            // Execute the action
            List<Invocable.Action.Result> results = action.invoke();

            if (results != null && !results.isEmpty()) {
                Invocable.Action.Result result = results[0];

                if (result.isSuccess()) {

```

```

        System.debug('Invoke UW Rule successfully from Invocable Action');

        // Extract key information from the result
        Map<String, Object> outputParams = result.getOutputParameters();
        if (outputParams != null) {
            outputRep = outputParams.get(OUTPUT_NAME);
            System.debug('output: ' + JSON.serialize(outputRep));

            return outputRep;
        }
        } else {
            System.debug('invoke UW Rule fail: ' + result);
            return result.getErrors();
        }
    }

    } catch (Exception e) {
        throw e;
    }

    return outputRep;
}

```

Sample Response

```

[
  {
    "actionName": "invokeUnderwritingRules",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "invokeUnderwritingRuleOputRep": {
        "objectId": "0Zkxx000000001qCAA",
        "objectName": "Claim",
        "fromStage": "Draft",
        "toStage": "Open",
        "stageTransitionName": "Draft to Open",
        "stageTransitionUnderwritingEvaluationId": "1Mjxx00000004C0CA2",
        "isSuccess": true,
        "rulesetResults": [
          {
            "productId": "01tSM0000050FyGYAW",
            "evaluationCriteria": "1 OR 2",
            "isSuccess": true,
            "ruleResults": [
              {
                "ruleApiName": "CollisionRule",
                "isSuccess": true
              },
              {
                "ruleApiName": "AutoSilverRule",
                "isSuccess": true
              }
            ]
          }
        ]
      }
    }
  }
]

```

```

    ]
  },
  {
    "productId": "01tSM000005OFyJYAW",
    "evaluationCriteria": "1",
    "isSuccess": true,
    "ruleResults": [
      {
        "ruleApiName": "MedicalRule1",
        "isSuccess": true
      }
    ]
  }
]
}
},
"sortOrder": -1,
"version": 1
}
]

```

Insurance Product Administration Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Product Administration](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Product Administration

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Product Administration exposes additional `actionType` values for the FlowActionCall Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	InvokableActionType (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Product Administration include: <code>invokeUnderwritingRules</code>—Invoke the underwriting rules for different insurance records during the run time.

CHAPTER 7 Insurance Quoting

In this chapter ...

- [Insurance Quoting Business APIs](#)
- [Insurance Quoting Standard Invocable Actions](#)
- [Insurance Quoting Metadata API Types](#)
- [Insurance Quoting Developer Journey](#)

Use the Insurance Quoting developer resources to create, update, configure, and delete quotes and quote line items.

SEE ALSO:

[Salesforce Help: Insurance Quoting](#)

EDITIONS

Available in: Lightning Experience

Available in: **Enterprise**, **Unlimited**, and **Developer** Editions

Insurance Quoting Business APIs

Create, update, configure, and delete quotes and quote line items. Use the APIs to create guided flows for your external end users.

This table lists the available Insurance Quoting resources.

Resource	Description
/connect/insurance/quotes (POST)	Generate a quote or quote-ready context in multiple ways.
/connect/insurance/product-rating (POST, PATCH)	Rate products or update product ratings for use in Insurance quoting processes.
/connect/insurance/product-rating/get-rating-input (POST)	Generate rating input details from a list of Product IDs, and use the generated rating input as the request body for the Insurance Product Rating API to evaluate product pricing.
/connect/insurance/product-surcharges/productSurchargeId (PATCH)	Update or fetch a product surcharge for a root product.
/connect/insurance/product-surcharges/bulk (POST)	Create one or more product surcharges to associate with a root product.
/connect/insurance/quotes/quoteId (GET, PATCH)	Update or get details for a specified quote ID.
/connect/insurance/underwriting-rules/invoke (POST)	Invoke underwriting rules for different object records during run time.
/connect/insurance/product-clauses/add-eligible-product-clauses (POST)	Add selected eligible product clauses to a quote or context.
/connect/insurance/product-clauses/eligible-product-clauses (POST)	Get eligible product clauses based on quoteld and contextId.
/connect/insurance/product-clauses/productClauseId (PATCH)	Update an existing product clause for a root product.
/connect/insurance/product-clauses/bulk (POST)	Create one or multiple product clauses at once.
/connect/insurance/product-clauses/clause-generation (POST)	Generate product clauses based on quote and context information.

[Resources](#)

Learn more about the available Insurance Quoting API resources.

[Requests](#)

Learn more about the available Insurance Quoting requests.

[Responses](#)

Learn more about the available response bodies of Insurance Quoting APIs.

Resources

Learn more about the available Insurance Quoting API resources.

[Insurance Invoke Underwriting Rule \(POST\)](#)

Invoke underwriting rules for different object records during run time.

[Insurance Product Rating \(POST, PATCH\)](#)

Rate products or update product ratings in insurance quoting processes.

[Insurance Product Rating Details \(POST\)](#)

Generate rating input details from a list of Product IDs, and use the generated rating input as the request body for the Insurance Product Rating API to evaluate product pricing.

[Insurance Product Surcharge \(PATCH\)](#)

Update or fetch a product surcharge for a root product.

[Insurance Product Surcharge Bulk \(POST\)](#)

Create one or more product surcharges to associate with a root product.

[Insurance Quote \(POST\)](#)

Generate a quote or quote-ready context in multiple ways. The API accepts user inputs, generates context, runs rules and pricing, prorates, and then returns a quote ID. The API can also use a context ID to run pricing and rules, update the context, and then return a quote ID. Alternatively, the API can accept user inputs, generate context, run rules and pricing, prorate, and then return only the context ID.

[Insurance Quote Details \(GET, PATCH\)](#)

Get or update quote details by the specified quote ID.

[Insurance Eligible Product Clauses Add \(POST\)](#)

Add selected eligible product clauses to a quote or context.

[Insurance Eligible Product Clauses Get \(POST\)](#)

Get eligible product clauses based on quoteld and contextld.

[Insurance Product Clause \(PATCH\)](#)

Update an existing product clause for a root product.

[Insurance Product Clause Bulk \(POST\)](#)

Create one or multiple product clauses at once.

[Insurance Product Clause Generation \(POST\)](#)

Generate product clauses based on quote and context information.

Insurance Invoke Underwriting Rule (POST)

Invoke underwriting rules for different object records during run time.

Resource

```
/connect/insurance/underwriting-rules/invoke
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/underwriting-rules/invoke
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "objectId": "110xx0000007MnhEAE",
  "toStage": "Approved"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
objectId	String	ID of the insurance object(Quote or Insurance Policy ID).	Required	63.0
toStage	String	Name of the stage the record is moving to.	Required	63.0

Response body for POST

[Insurance Invoke Underwriting Rule](#)

Insurance Product Rating (POST, PATCH)

Rate products or update product ratings in insurance quoting processes.

Resource

```
/connect/insurance/product-rating
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-rating
```

Available version

63.0

HTTP methods

POST, PATCH

Request body for POST

JSON example

```
{
  "transactionType": "AutoTransactionType",
  "additionalFields": {
    "QuoteAccount": "001ZM000002ZNnBYAW"
  },
  "ratingOptions": {
    "executeConfigurationRules": false
  },
  "ratingInputs": [
    {
      "productCode": "auto",
      "instanceKeys": [
        "Auto1"
      ],
      "attributes": {
        "Make": "BMW",
        "AutoValue": "20000",
        "Year": "2024",
        "Model": "SUV"
      }
    },
    {
      "productCode": "comprehensive",
      "instanceKeys": [
        "Auto1",
        "Comprehensive"
      ],
      "attributes": {
        "Limit": "2000",
        "Deductible": "100"
      }
    }
  ],
  "rootProductCodes": [
    "auto-ECO1x",
    "auto-ECO2x"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Map of additional fields that must be updated in the quote context.	Required if the generated context ID is used to issue a policy	63.0

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date used to find the active pricing procedure for pricing. The default value is the current date.	Optional	63.0
ratingInputs	Rating Input Node []	JSON that contains the attribute values of each context node.	Required	63.0
ratingOptions	Rating Options Input []	Options for rating.	Optional	63.0
rootProductCodes	String []	Root product codes that are used with the <code>ratingInputs</code> to create rating result. The created rating result is used as request in the insurance product rating API.	Optional	66.0
transactionType	String	Name of the transaction processing type.	Required	63.0

Response body for POST[Product Rating Response](#)**Request body for PATCH****JSON example**

```
{
  "contextId": "a49103b233638b918217c74eced69220e534e160c3d3d35bc8cf6608515aeab3",
  "ratingOptions": {
    "executePricing": true,
    "executeConfigurationRules": false
  },
  "updatedNodes": [
    {
      "instanceKeys": [
        "AutoSilver",
        "BMW",
        "Tim1"
      ],
      "productCode": "driver",
      "attributes": {
        "DriverAccidentPoints": 1,
        "AgeFirstLicensed": 50,
        "DriverMVRPoints": 1
      },
      "targetRecords": [
        "003DR000002ov8NYAQ",
        "001DR000003avQCYAY"
      ]
    }
  ],
  "addedNodes": [
```

```

{
  "instanceKeys": [
    "AutoSilver",
    "Tesla"
  ],
  "productCode": "auto",
  "attributes": {
    "Year": "2022",
    "Model": "Y"
  }
},
{
  "instanceKeys": [
    "AutoSilver",
    "Tesla",
    "collision1"
  ],
  "productCode": "collision",
  "attributes": {
    "Limit": 500,
    "Deductible": 1000
  }
},
{
  "instanceKeys": [
    "AutoSilver",
    "Tesla",
    "Driver1"
  ],
  "productCode": "driver",
  "attributes": {
    "DriverAccidentPoints": 5,
    "AgeFirstLicensed": 30,
    "DriverMVRPoints": 1
  },
  "targetRecords": [
    "003DR000002oxCfYAI",
    "001DR000003azdpYAA"
  ]
},
{
  "instanceKeys": [
    "AutoSilver",
    "Audi",
    "Collision2"
  ],
  "productCode": "collision",
  "attributes": {
    "Limit": 500,
    "Deductible": 1000
  }
}
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
addedNodes	Rating Input Node []	List of added nodes.	Optional	63.0
additionalFields	Map<String, Object>	Map of additional fields that must be updated in the quote context.	Optional	63.0
contextId	String	Context ID.	Required	63.0
deletedNodes	Rating Input Node []	List of deleted nodes.	Optional	63.0
effectiveDate	String	Date used to find the active pricing procedure for pricing. The default value is the current date.	Optional	63.0
ratingDate	String	Date that's used to find the active pricing procedure.	Optional	63.0
ratingOptions	Rating Options Input []	Options for rating.	Optional	63.0
updatedNodes	Rating Input Node []	List of updated nodes.	Optional	63.0

Response body for PATCH[Product Rating Response](#)**Insurance Product Rating Details (POST)**

Generate rating input details from a list of Product IDs, and use the generated rating input as the request body for the Insurance Product Rating API to evaluate product pricing.

This API retrieves product catalog information and constructs a hierarchical rating input structure based on product relationships. It generates product-specific attributes that's formatted specifically for the [Insurance Product Rating API](#).

Resource

```
/connect/insurance/product-rating/get-rating-input
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-rating/get-rating-input
```

Available version

66.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "productIds": [
```

```
    "01txx0000006i3DAAQ"
  ],
  "salesTransactionTypeId": "0YWxx000000001GAAQ",
  "transactionType": "Auto",
  "ratingOptions": {
    "executeConfigurationRules": true,
    "executePricing": true,
    "returnContextJSON": false,
    "returnProductDetails": false
  },
  "effectiveDate": "2025-01-01T00:00:00.000Z",
  "ratingDate": "2025-01-01T00:00:00.000Z",
  "additionalFields": {
    "customField1": "value1"
  },
  "rootProductCodes": [
    "AUTO_POLICY"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Additional fields to generate rating input.	Optional	66.0
effectiveDate	Date	Date that's used to find the active pricing procedure for pricing. The default value is the current date.	Optional	66.0
productIds	String[]	IDs of the products for which you want to generate rating input.	Required	66.0
ratingDate	Date	Date that's used to calculate rating.	Optional	66.0
ratingOptions	Rating Options Input	Options to control rating behavior.	Optional	66.0
rootProduct Codes	String[]	Root product codes that are used to create rating inputs.	Optional	66.0
salesTransaction TypeId	String	ID of the sales transaction type for which you want to generate rating request payload.	Optional	66.0
transaction Type	String	Name of the transaction processing type.	Optional	66.0

Response body for POST
[Product Rating Details Response](#)

Insurance Product Surcharge (PATCH)

Update or fetch a product surcharge for a root product.

Resource

```
/connect/insurance/product-surcharges/productSurchargeId
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-surcharges/productSurchargeId
```

Available version

63.0

HTTP methods

PATCH

Request body for PATCH

JSON example

```
{
  "name": "AutoVehicleTax",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "isProrationEligible": true,
  "isRefundEligible": true,
  "isActive": true,
  "productPath": "01tDR000000EMslYAG/11BDR00000000T72AI", // "Auto Gold/Auto
Vehicle"
  "ruleDefinition": [ // ruleDefinition will be replaced as a whole. Updating
only one condition is not supported
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGYAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG", // Auto Gold.
      "productRelatedComponent" : "",
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "11BDR00000000T72AI" // Auto Vehicle.
      ]
    }
  ]
}
```

```

    },
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Comprehensive Deductible",
          "type": "Attribute",
          "attributePicklistValueId": null,
          "attributeId": "0tjDR00000000UaYAI",
          "dataType": "Number",
          "values": [
            "$5000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG", // Auto Bundle
      "productRelatedComponent" : "",
      "criteriaIndex": 2,
      "sourceContextTagName": "Product",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "01tDR000000EMsoYAG" // Comprehensive Product.
      ]
    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Effective from date of the product surcharge.	Optional	63.0
effectiveToDate	String	Effective to date of the product surcharge.	Optional	63.0
isActive	Boolean	Indicates whether the product surcharge is active (<code>true</code>) or not (<code>false</code>). Currently, only active product surcharges are supported.	Optional	63.0
isProrationAllowed	Boolean	Indicates whether the product surcharge is allowed for proration (<code>true</code>) or not (<code>false</code>).	Optional	63.0
isRefundAllowed	Boolean	Indicates whether the product surcharge is allowed for refund (<code>true</code>) or not (<code>false</code>).	Optional	63.0
name	String	Name of the product surcharge.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
productPath	String	A path containing the product IDs that roll up to the parent product.	Optional	63.0
productRelatedComponentId	String	Product related component associated with the product surcharge.	Optional	63.0
ruleApiName	String	Rule API name associated with product surcharge. You can set the ruleApiName with the POST or PATCH method. Once set, this parameter can't be changed. If you set the ruleApiName in the request body, you must also set the ruleCriteria.	Optional	63.0
ruleCriteria		Rule criteria for the dynamic rule.	Optional	63.0

Response body for PATCH

[Product Surcharge Patch](#)

Insurance Product Surcharge Bulk (POST)

Create one or more product surcharges to associate with a root product.

Resource

```
/connect/insurance/product-surcharges/bulk
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-surcharges/bulk
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "records": [
    {
      "name": "AutoVehicleTax",
      "surchargeId": "01cLO000000EMslOKL",
      "rootProductId": "01tDR000000EMslYAG",
      "productPath": "01tDR000000EMslYAG/11BDR00000000T72AI",
      "effectiveFromDate": "2024-05-30T10:30:00",
      "effectiveToDate": "2024-11-01T10:30:00",
      "isProrationEligible": true,
      "isRefundEligible": true,
      "isActive": true,
    }
  ]
}
```

```

"ruleApiName": "autoVehicleTaxRule",
"ruleCriteria": [
  {
    "conditions": [
      {
        "contextTagName": "SalesTransactionItemAttribute",
        "operator": "LessThan",
        "conditionIndex": 1,
        "attributeName": "Annual Milage",
        "attributePicklistValueId": null,
        "type": "Attribute",
        "attributeId": "0tjDR00000000VGYAY",
        "dataType": "Number",
        "values": [
          "50000"
        ]
      }
    ],
    "rootObjectId": "01tDR000000EMslYAG",
    "productRelatedComponent": "",
    "criteriaIndex": 1,
    "sourceContextTagName": "ProductBasedOn",
    "sourceOperator": "Equals",
    "sourceDataType": "String",
    "sourceValues": [
      "11BDR00000000T72AI"
    ]
  },
  {
    "conditions": [
      {
        "contextTagName": "SalesTransactionItemAttribute",
        "operator": "LessThan",
        "conditionIndex": 1,
        "attributeName": "Comprehensive Deductible",
        "type": "Attribute",
        "attributePicklistValueId": null,
        "attributeId": "0tjDR00000000UaYAI",
        "dataType": "Number",
        "values": [
          "$5000"
        ]
      }
    ],
    "rootObjectId": "01tDR000000EMslYAG",
    "productRelatedComponent": "",
    "criteriaIndex": 2,
    "sourceContextTagName": "Product",
    "sourceOperator": "Equals",
    "sourceDataType": "String",
    "sourceValues": [
      "01tDR000000EMsoYAG"
    ]
  }
]

```

```
    ]
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
records	Product Surcharge Post Input[]	List of the product surcharges.	Required	63.0

Response body for POST

[Product Surcharge List](#)

Insurance Quote (POST)

Generate a quote or quote-ready context in multiple ways. The API accepts user inputs, generates context, runs rules and pricing, prorates, and then returns a quote ID. The API can also use a context ID to run pricing and rules, update the context, and then return a quote ID. Alternatively, the API can accept user inputs, generate context, run rules and pricing, prorate, and then return only the context ID.

Resource

```
/connect/insurance/quotes
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/quotes
```

Available version

63.0

HTTP methods

POST

Request body for POST

JSON example

```
{
  "quoteName": "Auto Insurance Quote",
  "effectiveDate": "2024-06-30T00:00:00.000Z",
  "expirationDate": "2025-07-28T00:00:00.000Z",
  "opportunityId": "006xx000001a2odAAA",
  "transactionType": "AutoTransactionType",
  "salesTransactionTypeId": "1Chxx0000004C92CAE",
  "priceBookId": "01sxx0000005ptpAAA",
  "configOptions": {
    "executePricing": false,
    "validateProductCatalog": false,
    "executeConfigurationRules": false,
    "executeQualificationRules": false
  },
  "options": {
```

```

    "saveQuote": true,
    "productSellingModel": "One Time"
  },
  "quoteInputs": [
    {
      "productCode": "autoRoot",
      "instanceKeys": [
        "AutoRoot"
      ],
      "items": [
        {
          "productCode": "autoBundle",
          "instanceKeys": [
            "AutoRoot",
            "AutoBundle1"
          ],
          "attributes": {
            "avMake": "BMW",
            "avValue": "20000",
            "avYear": "2020",
            "AutoValueFactor": 5,
            "AutoAgeFactor": 6
          }
        }
      ],
    },
    {
      "productCode": "autoDriver",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle1",
        "AutoDriver"
      ],
      "relatedRecords": [
        {
          "relatedRecordId": "001xx000003GYiEAAW",
          "relatedRecordObjName": "Account"
        },
        {
          "relatedRecordId": "001xx000003GYiEAAV",
          "relatedRecordObjName": "Contact"
        }
      ]
    },
    {
      "productCode": "autoComp",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle1",
        "AutoComp"
      ],
      "attributes": {
        "CompBasePremium": "750"
      }
    },
    {

```

```

    "productCode": "autoBundle",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2"
    ],
    "attributes": {
      "avMake": "Audi",
      "avValue": "20000",
      "avYear": "2020",
      "AutoValueFactor": 5,
      "AutoAgeFactor": 6
    }
  },
  {
    "productCode": "autoDriver",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2",
      "AutoDriver"
    ],
    "relatedRecords": [
      {
        "relatedRecordId": "001xx000003GYiEAAW",
        "relatedRecordObjName": "Account"
      },
      {
        "relatedRecordId": "001xx000003GYiEAAV",
        "relatedRecordObjName": "Contact"
      }
    ]
  },
  {
    "productCode": "autoCollision",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2",
      "AutoCollision1"
    ],
    "attributes": {
      "CollisionBasePremium": "120"
    }
  }
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account associated with the quote that's being created.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Additional fields that are used to create insurance quote.	Optional	63.0
configOptions	Insurance Quote Config Options[]	The JSON structure that includes the options for configuration.	Optional	63.0
contextId	String	ID of the context that's used for creating the quote.	Optional	63.0
effectiveDate	String	Date that the quote is effective from.	Required	63.0
expirationDate	String	Expiration date of the quote.	Optional	63.0
opportunityId	String	ID of the opportunity associated with the quote.	Optional	63.0
options	Insurance Quote Options[]	Insurance Quote Options.	Optional	63.0
priceBookId	String	ID of the pricebook associated with the quote.	Optional	63.0
pricingProcedure	String	Pricing Procedure for insurance quote.	Optional	63.0
quoteInputs	Insurance Quote Input Node[]	The JSON structure that contains user inputs for the products for which the quote is being created.	Required	63.0
quoteName	String	Name of the quote.	Optional	63.0
ratingDate	String	Date of the rating quote.	Optional	63.0
salesTransactionId	String	ID of the sales transaction type.	Optional	63.0
transactionType	String	Type of the transaction.	Optional	63.0

Response body for POST[Insurance Quote](#)**Insurance Quote Details (GET, PATCH)**

Get or update quote details by the specified quote ID.

You can get the desired quote details by using the filter and non-filter request parameters. Use the term `filter` to differentiate between the filter and non-filter parameters. See, Resource example.



Note: A filter parameter is used to show the results that match a certain criteria, for example `ProductSpec=InsuredItem` shows the quote details for the insured item only. A non-filter parameter controls the behavior or format of the response, for example `returnChildrenNodes=true` shows the child nodes of the `SalesTransactionItem`. In the default context definition, a `SalesTransactionItem` in context represents a `QuoteLineItem`.

These are the operators that are supported to filter the results.

- eq—Equal
- ne—Not equal
- lt—Less than
- lte—Less than or equal to
- gt—Greater than
- gte—Greater than or equal to

Resource

```
/connect/insurance/quotes/quoteId

/connect/insurance/quotes/quoteId?contextId={value}&filter[{field1}]{operator1}={value1}&filter[{field2}]{operator2}={value2}&key={value}&includeChildren={true}
```

The `field1` parameter refers to the `ProductSpec` filter parameter, and the `field2` parameter refers to the valid context attribute of `SalesTransactionItem` in the target context definition. You can add multiple filter and non-filter parameters to get the desired result.

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/quotes/0Q0DR0000000xFW0AI

https://yourInstance.salesforce.com/services/data/v/connect/insurance/quotes/0Q0DR0000000xFW0AI?contextId=00000000000000000000000000000000&filter[ProductSpec]eq=00000000000000000000000000000000&filter[SalesTransactionItem]lt=00000000000000000000000000000000&key=00000000000000000000000000000000&includeChildren=true
```

Available version
63.0

HTTP methods
GET, PATCH

Request parameters for GET

Parameter Name	Type	Description	Required or Optional	Available Version
<code>connectParameters</code>	Map<String,String>	Specify these filter and non-filter request parameters to get the desired results. • <code>contextId</code>—Non-filter parameter to get the quote details for the specified context ID. • <code>ProductSpec</code>—Filter parameter to get the quote details for the specified product spec. Valid values are: – Coverage – InsuredItem – InsuredParty – Product • <code>returnChildrenNodes</code>—Non-filter parameter that retrieves all the <code>SalesTransactionItems</code> in	Optional	63.0

Parameter Name	Type	Description	Required or Optional	Available Version
		the hierarchy, starting from each matching <code>SalesTransactionItem</code> and includes all child nodes. Use the <code>returnChildrenNodes</code> parameter with filter parameters. <code>includeClause</code>—Non-filter parameter that includes the associated product clauses of the <code>SalesTransactionItems</code> in the response body. Use the <code>includeClause</code> parameter with filter parameters.		

Response body for GET[Get Quote](#)**Request body for PATCH****JSON example**

```
{
  "configOptions": {
    "executePricing": false,
    "validateProductCatalog": false,
    "executeConfigurationRules": false,
    "executeQualificationRules": false
  },
  "additionalFields": {
    "Status": "Draft",
    "ValidFromDate": "2025-10-29T00:00:00.000Z"
  },
  "addedNodes": [
    {
      "productCode": "autoCollision",
      "instanceKeys": [
        "autoBundle",
        "autoCollision"
      ],
      "additionalFields": {
        "StartDate": "2025-09-29T00:00:00.000Z",
        "EndDate": "2025-10-29T00:00:00.000Z",
        "Discount": 40
      }
    }
  ],
  "updatedNodes": [
    {
```



```
    "instanceKeys": [
      "autoBundle"
    ],
    "productCode": "autoBundle",
    "attributes": {
      "avMake": "AudiQ14"
    },
    "additionalFields": {
      "StartDate": "2025-09-29T00:00:00.000Z",
      "EndDate": "2025-10-29T00:00:00.000Z",
      "Discount": 50
    }
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
addedNodes	Insurance Quote Input Node[]	List of added nodes for insurance quote.	Optional	63.0
additionalFields	Map<String, Object>	Insurance quote additional fields.	Optional	63.0
configOptions	Insurance Quote Config Options[]	Configurator options for insurance quote.	Optional	63.0
deletedNodes	Insurance Quote Input Node[]	List of deleted nodes from insurance quote.	Optional	63.0
pricingProcedure	String	Pricing procedure for insurance quote.	Optional	63.0
updatedNodes	Insurance Quote Input Node[]	List of updated nodes on insurance quote.	Optional	63.0

Response body for PATCH

[Insurance Quote](#)

Insurance Eligible Product Clauses Add (POST)

Add selected eligible product clauses to a quote or context.

Resource

```
/connect/insurance/product-clauses/add-eligible-product-clauses
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-clauses/add-eligible-product-clauses
```

Available version

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "quoteId": "0Q0xx0000000001AAA",
  "quoteLineItemId": "0QLxx0000000002BBB",
  "contextId": "0j5xx0000000003CCC",
  "instanceKey": ["Auto", "Liability"],
  "insuranceProductClauses": [
    {
      "insuranceProductClauseId": "0j4xx0000000004DDD",
      "effectiveDate": "2025-01-01",
      "expirationDate": "2025-12-31"
    }
  ],
  "options": {
    "saveContext": false
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context.	Required if quoteId isn't specified	65.0
instanceKey	String[]	List of product instance IDs from the root product to the target quote line item.	Required if quoteLineItemId isn't specified	65.0
insurance Product Clauses	Insurance Product Clause Input[]	List of insurance product clauses to add.	Required	65.0
options	Add Eligible Product Clauses Options Input	Options for adding eligible product clauses.	Optional	65.0
quoteId	String	ID of the quote.	Required if contextId isn't specified	65.0
quoteLine ItemId	String	ID of the quote line item.	Required if instanceKey isn't specified	65.0

Response body for POST[Add Eligible Product Clauses Details](#)

Insurance Eligible Product Clauses Get (POST)

Get eligible product clauses based on quoteld and contextId.

Resource

```
/connect/insurance/product-clauses/eligible-product-clauses
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-clauses/eligible-product-clauses
```

Available version

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "contextId": "00000000i18tq18g0025175944829641764efd423f7dc44b98dd1e829086efc59",
  "instanceKey": [
    "a3Nxx000000CmikEAC"
  ],
  "quoteId": "0Q0xx000000Dm3jCAC",
  "quoteLineItemId": "0QLxx000000E5d1OAC"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of context.	Required if quoteId isn't specified	65.0
instanceKey	String[]	List of product instance IDs from the root product to the target quote line item.	Required if quoteLineItemId isn't specified	65.0
quoteId	String	ID of the quote.	Required if contextId isn't specified	65.0
quoteLineItemId	String	ID of the quote line item.	Required if instanceKey isn't specified	65.0

Response body for POST[Insurance Get Eligible Product Clauses](#)

Insurance Product Clause (PATCH)

Update an existing product clause for a root product.

Resource

```
/connect/insurance/product-clauses/productClauseId
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-clauses/0j4xx0000000001AAA
```

Available version

65.0

HTTP methods

PATCH

Path parameter for PATCH

Parameter Name	Type	Description	Required or Optional	Available Version
productClauseId	String	ID of the product clause that must be updated.	Required	65.0

Request body for PATCH**JSON example**

```
{
  "effectiveDate": "2026-01-01",
  "expirationDate": "2026-12-31",
  "variableMappings": [
    {
      "id": "a1Bxx0000000001AAA",
      "variable": "LimitAmount",
      "attribute": "NewCoverageLimit"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date from which the product clause is valid.	Optional	65.0
expirationDate	String	Date from which the product clause is no longer valid.	Optional	65.0

Name	Type	Description	Required or Optional	Available Version
variable Mappings	Product Clause Variable Mapping Input[]	Variable mappings for the product clause that must be updated.	Optional	65.0

Response body for PATCH[Product Clause Update Details](#)

Insurance Product Clause Bulk (POST)

Create one or multiple product clauses at once.

Resource`/connect/insurance/product-clauses/bulk`**Resource example**`https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-clauses/bulk`**Available version**

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "productClauses": [
    {
      "apiName": "AutoGlassClause_2025",
      "name": "Auto Glass Clause",
      "effectiveDate": "2025-01-01",
      "expirationDate": "2025-12-31",
      "insuranceClauseId": "0j1xx0000000001AAA",
      "rootProduct": "01t0000000003CCC",
      "productPath": "01t0000000003DDD",
      "variableMappings": [
        {
          "variable": "DeductibleAmount",
          "variableType": "Currency",
          "attribute": "WaiverAmount"
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
productClauses	Product Clause Input[]	Details to create product clauses.	Required	65.0

Response body for POST[Product Clause Creation Details](#)

Insurance Product Clause Generation (POST)

Generate product clauses based on quote and context information.

Resource

```
/connect/insurance/product-clauses/clause-generation
```

Resource example

```
https://yourInstance.salesforce.com/services/data/v/connect/insurance/product-clauses/clause-generation
```

Available version

65.0

HTTP methods

POST

Request body for POST**JSON example**

```
{
  "quoteId": "0Q0xx0000000001AAA",
  "contextId": "0000000i18tq18g0025175944829641764efd423f7dc44b98dd1e829086efc59",
  "quoteLineItemIds": [
    "0QLxx0000000003CCC"
  ],
  "instanceKeys": [
    {
      "instanceKey": [ "Auto", "Liability" ]
    }
  ],
  "options": {
    "generateClausesAsync": true,
    "generateManualClauses": false,
    "saveContext": true
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context based on which the product clause is generated.	Required if quoteId isn't specified	65.0
instanceKeys	Instance Key Input[]	List of instance keys.	Optional	65.0
options	Clause Generation Options Input	Options for clause generation.	Optional	65.0
quoteId	String	ID of the quote.	Required if contextId isn't specified	65.0
quoteLineItemIds	String[]	IDs of the quote line items.	Optional	65.0

Response body for POST[Product Clause Generation Details](#)

Requests

Learn more about the available Insurance Quoting requests.

[Add Eligible Product Clause Input](#)

Input representation for adding eligible product clauses.

[Add Eligible Product Clause Options Input](#)

Input representation for options when adding eligible product clauses.

[Bulk Product Clause Input](#)

Input representation for creating one or a list of product clauses.

[Clause Generation Options Input](#)

Input representation for clause generation options.

[Instance Key Input](#)

Input representation for a single instance key array.

[Insurance Get Eligible Clauses Input](#)

Input representation for get eligible insurance product clauses.

[Insurance Product Clause Input](#)

Input representation for a single insurance product clause.

[Insurance Quote Config Options](#)

Input representation of insurance quote line item related objects.

[Insurance Quote Options](#)

Input representation of insurance quote options.

[Insurance Quote Related Record Input](#)

Input representation of insurance quote line item related objects.

[Insurance Invoke Underwriting Rule Input](#)

Input representation for invoke underwriting rule operation.

[Insurance Quote Input Node](#)

Input representation of root quote line items and child quote line items of insurance quote.

[Insurance Quote Patch Input](#)

Input representation for insurance quote patch operation.

[Insurance Quote Post Input](#)

Input representation details to create insurance quote.

[Product Clause Update Input](#)

Input representation for updating a product clause.

[Product Clause Variable Mapping Input](#)

Input representation for mapping an insurance clause variable to an insurance product variable.

[Product Rating Patch Input](#)

Input representation of product rating update details.

[Product Rating Post Input](#)

Input representation of the product rating details.

[Product Rating Request Input](#)

Input representation for getting rating input from product IDs.

[Product Surcharge Bulk Post Input](#)

Input representation of the details of the product surcharge records.

[Product Surcharge Patch Input](#)

Input representation for product surcharge patch request.

[Product Surcharge Post Input](#)

Input representation of product surcharge details.

[Product Clause Input](#)

Input representation for the details of a single product clause to be created.

[Rating Input Node](#)

Input representation of details of context node.

[Rating Options Input](#)

Input representation details of rating service options.

Add Eligible Product Clause Input

Input representation for adding eligible product clauses.

JSON example

```
{
  "quoteId": "0Q0xx0000000001AAA",
  "quoteLineItemId": "0QLxx0000000002BBB",
  "contextId": "0j5xx0000000003CCC",
```



```

"instanceKey": ["Auto", "Liability"],
"insuranceProductClauses": [
  {
    "insuranceProductClauseId": "0j4xx0000000004DDD",
    "effectiveDate": "2025-01-01",
    "expirationDate": "2025-12-31"
  }
],
"options": {
  "saveContext": false
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of the context.	Required if quoteId isn't specified	65.0
instanceKey	String[]	List of product instance IDs from the root product to the target quote line item.	Required if quoteLineItemId isn't specified	65.0
insurance Product Clauses	Insurance Product Clause Input[]	List of insurance product clauses to add.	Required	65.0
options	Add Eligible Product Clauses Options Input	Options for adding eligible product clauses.	Optional	65.0
quoteId	String	ID of the quote.	Required if contextId isn't specified	65.0
quoteLine ItemId	String	ID of the quote line item.	Required if instanceKey isn't specified	65.0

Add Eligible Product Clause Options Input

Input representation for options when adding eligible product clauses.

JSON example

```

{
  "saveContext": false
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
saveContext	Boolean	Indicates whether to save the context (true) or not (false). The default value is false.	Optional	65.0

Bulk Product Clause Input

Input representation for creating one or a list of product clauses.

JSON example

```
{
  "productClauses": [
    {
      "apiName": "AutoGlassClause_2025",
      "name": "Auto Glass Clause",
      "effectiveDate": "2025-01-01",
      "expirationDate": "2025-12-31",
      "insuranceClauseId": "0j1xx0000000001AAA",
      "rootProduct": "01t000000003CCC",
      "productPath": "01t000000003DDD",
      "variableMappings": [
        {
          "variable": "DeductibleAmount",
          "variableType": "Currency",
          "attribute": "WaiverAmount"
        }
      ]
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
productClauses	Product Clause Input[]	Details to create product clauses.	Required	65.0

Clause Generation Options Input

Input representation for clause generation options.

JSON example

```
{
  "generateClausesAsync": true,
  "generateManualClauses": false,
}
```

```
{
  "saveContext": true
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
generateClausesAsync	Boolean	Indicates whether to generate clauses asynchronously (<code>true</code>) or not (<code>false</code>).	Optional	65.0
generateManualClauses	Boolean	Indicates whether to generate clauses manually (<code>true</code>) or not (<code>false</code>).	Optional	65.0
saveContext	Boolean	Indicates whether to save the context while generating the product clause (<code>true</code>) or not (<code>false</code>).	Optional	65.0

Instance Key Input

Input representation for a single instance key array.

JSON example

```
{
  "instanceKey": [ "Auto", "Liability" ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
instanceKey	String[]	List of instance key strings that define a path.	Required	65.0

Insurance Get Eligible Clauses Input

Input representation for get eligible insurance product clauses.

JSON example

```
{
  "contextId": "0000000i18tq18g0025175944829641764efd423f7dc44b98dd1e829086efc59",
  "instanceKey": [
    "a3Nxx000000CmikEAC"
  ],
  "quoteId": "0Q0xx000000Dm3jCAC",
  "quoteLineItemId": "0QLxx000000E5d1OAC"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
contextId	String	ID of context.	Required if quoteId isn't specified	65.0
instanceKey	String[]	List of product instance IDs from the root product to the target quote line item.	Required if quoteLineItemId isn't specified	65.0
quoteId	String	ID of the quote.	Required if contextId isn't specified	65.0
quoteLineItemId	String	ID of the quote line item.	Required if instanceKey isn't specified	65.0

Insurance Product Clause Input

Input representation for a single insurance product clause.

JSON example

```
{
  "effectiveDate": "2025-10-27T12:00:00.000Z",
  "expirationDate": "2026-10-27T12:00:00.000Z",
  "insuranceProductClauseId": "a4Lxx000000Cde1EAC"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date from which the product clause is valid.	Optional	65.0
expirationDate	String	Date from which the product clause is no longer valid.	Optional	65.0
insuranceProductClauseId	String	ID of the insurance product clause.	Optional	65.0

Insurance Quote Config Options

Input representation of insurance quote line item related objects.

JSON example

```
"configOptions": {
  "executePricing": false,
  "validateProductCatalog": false,
  "executeConfigurationRules": false,
  "executeQualificationRules": false
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>executeConfigurationRules</code>	Boolean	Indicates whether to run product configuration rules as part of insurance quote (<code>true</code>) or not (<code>false</code>).	Optional	63.0
<code>executePricing</code>	Boolean	Indicates whether to execute NGP Pricing as part of insurance quote (<code>true</code>) or not (<code>false</code>). This parameter is always set to <code>false</code> for Group Benefits rating.	Optional	63.0
<code>executeQualificationRules</code>	Boolean	Indicates whether to execute qualification rules as part of insurance quote (<code>true</code>) or not (<code>false</code>).	Optional	63.0
<code>validateProductCatalog</code>	Boolean	Indicates whether to execute product catalog validation as part of insurance quote (<code>true</code>) or not (<code>false</code>).	Optional	63.0

Insurance Quote Options

Input representation of insurance quote options.

JSON example

```
options: {
  "saveQuote": true,
  "productSellingModel": "One Time"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>productSellingModel</code>	String	Determines the pricebookId for the products.	Required	63.0

Name	Type	Description	Required or Optional	Available Version
saveQuote	Boolean	Indicates whether to save insurance quote or only generate context.	Required	63.0

Insurance Quote Related Record Input

Input representation of insurance quote line item related objects.

JSON example

```
relatedRecords": [
  {
    "relatedRecordId": "001xx000003GYiEAAW",
    "relatedRecordObjName": "Account"
  },
  {
    "relatedRecordId": "001xx000003GYiEAAV",
    "relatedRecordObjName": "Contact"
  }
]
```

Properties

Name	Type	Description	Required or Optional	Available Version
relatedRecordId	String	ID of the related insurance quote record.	Required	63.0
relatedRecordObjName	String	Object API name of the related insurance quote record.	Required	63.0

Insurance Invoke Underwriting Rule Input

Input representation for invoke underwriting rule operation.

JSON example

```
{
  "objectId": "110xx0000007MnhEAE",
  "toStage": "Approved"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
objectId	String	ID of the insurance object(Quote or Insurance Policy ID).	Required	63.0
toStage	String	Name of the stage the record is moving to.	Required	63.0

Insurance Quote Input Node

Input representation of root quote line items and child quote line items of insurance quote.

JSON example

```
quoteInputs: [
  {
    "productCode": "autoRoot",
    "instanceKeys": [
      "AutoRoot"
    ],
    "items": [
      {
        "productCode": "autoBundle",
        "instanceKeys": [
          "AutoRoot",
          "AutoBundle1"
        ],
        "attributes": {
          "avMake": "BMW",
          "avValue": "20000",
          "avYear": "2020",
          "AutoValueFactor": 5,
          "AutoAgeFactor": 6
        }
      },
      {
        "productCode": "autoDriver",
        "instanceKeys": [
          "AutoRoot",
          "AutoBundle1",
          "AutoDriver"
        ],
        "relatedRecords": [
          {
            "relatedRecordId": "001xx000003GYiEAAW",
            "relatedRecordObjName": "Account"
          },
          {
            "relatedRecordId": "001xx000003GYiEAAV",
            "relatedRecordObjName": "Contact"
          }
        ]
      }
    ],
  },
  {
    "productCode": "autoComp",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle1",
      "AutoComp"
    ],
    "attributes": {
      "CompBasePremium": "750"
    }
  }
]
```

```

    },
    {
      "productCode": "autoBundle",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle2"
      ],
      "attributes": {
        "avMake": "Audi",
        "avValue": "20000",
        "avYear": "2020",
        "AutoValueFactor": 5,
        "AutoAgeFactor": 6
      }
    },
    {
      "productCode": "autoDriver",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle2",
        "AutoDriver"
      ],
      "relatedRecords": [
        {
          "relatedRecordId": "001xx000003GYiEAAW",
          "relatedRecordObjName": "Account"
        },
        {
          "relatedRecordId": "001xx000003GYiEAAV",
          "relatedRecordObjName": "Contact"
        }
      ]
    },
    {
      "productCode": "autoCollision",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle2",
        "AutoCollision1"
      ],
      "attributes": {
        "CollisionBasePremium": "120"
      }
    }
  ]
}
]

```


Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Insurance quote line items.	Optional	63.0
attributes	Map<String, Object>	Insurance quote attributes key-value pairs.	Optional	63.0
instanceKeys	String[]	Product instance identifier of the quote line item.	Required	63.0
items	String[]	The JSON structure that contains user inputs for the products for which the quote is created.	Optional	63.0
priceBookEntryId	String	ID of the pricebook associated with the quote.	Optional	63.0
productCode	String	Product code of the associated product.	Required	63.0
relatedRecords	Insurance Quote Related Record Input []	List of the related insurance quote records.	Required	62.0

Insurance Quote Patch Input

Input representation for insurance quote patch operation.

JSON example

```
{
  "configOptions": {
    "executePricing": false,
    "validateProductCatalog": false,
    "executeConfigurationRules": false,
    "executeQualificationRules": false
  },
  "additionalFields": {
    "Status": "Draft",
    "ValidFromDate": "2025-10-29T00:00:00.000Z"
  },
  "addedNodes": [
    {
      "productCode": "autoCollision",
      "instanceKeys": [
        "autoBundle",
        "autoCollision"
      ],
      "additionalFields": {
        "StartDate": "2025-09-29T00:00:00.000Z",
        "EndDate": "2025-10-29T00:00:00.000Z",

```

```
      "Discount": 40
    }
  }
],
"updatedNodes": [
  {
    "instanceKeys": [
      "autoBundle"
    ],
    "productCode": "autoBundle",
    "attributes": {
      "avMake": "AudiQ14"
    },
    "additionalFields": {
      "StartDate": "2025-09-29T00:00:00.000Z",
      "EndDate": "2025-10-29T00:00:00.000Z",
      "Discount": 50
    }
  }
]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
addedNodes	Insurance Quote Input Node[]	List of added nodes for insurance quote.	Optional	63.0
additionalFields	Map<String, Object>	Insurance quote additional fields.	Optional	63.0
configOptions	Insurance Quote Config Options[]	Configurator options for insurance quote.	Optional	63.0
deletedNodes	Insurance Quote Input Node[]	List of deleted nodes from insurance quote.	Optional	63.0
pricingProcedure	String	Pricing procedure for insurance quote.	Optional	63.0
updatedNodes	Insurance Quote Input Node[]	List of updated nodes on insurance quote.	Optional	63.0

Insurance Quote Post Input

Input representation details to create insurance quote.

JSON example

```
{
  "quoteName": "Auto Insurance Quote",
  "effectiveDate": "2024-06-30T00:00:00.000Z",
  "expirationDate": "2025-07-28T00:00:00.000Z",
  "opportunityId": "006xx000001a2odAAA",
}
```

```

"transactionType": "AutoTransactionType",
"salesTransactionTypeId": "1Chxx0000004C92CAE",
"priceBookId": "01sxx0000005ptpAAA",
"configOptions": {
  "executePricing": false,
  "validateProductCatalog": false,
  "executeConfigurationRules": false,
  "executeQualificationRules": false
},
"options": {
  "saveQuote": true,
  "productSellingModel": "One Time"
},
"quoteInputs": [
  {
    "productCode": "autoRoot",
    "instanceKeys": [
      "AutoRoot"
    ],
    "items": [
      {
        "productCode": "autoBundle",
        "instanceKeys": [
          "AutoRoot",
          "AutoBundle1"
        ],
        "attributes": {
          "avMake": "BMW",
          "avValue": "20000",
          "avYear": "2020",
          "AutoValueFactor": 5,
          "AutoAgeFactor": 6
        }
      }
    ],
    {
      "productCode": "autoDriver",
      "instanceKeys": [
        "AutoRoot",
        "AutoBundle1",
        "AutoDriver"
      ],
      "relatedRecords": [
        {
          "relatedRecordId": "001xx000003GYiEAAW",
          "relatedRecordObjName": "Account"
        },
        {
          "relatedRecordId": "001xx000003GYiEAAV",
          "relatedRecordObjName": "Contact"
        }
      ]
    }
  ],
  {
    "productCode": "autoComp",

```

```

    "instanceKeys": [
      "AutoRoot",
      "AutoBundle1",
      "AutoComp"
    ],
    "attributes": {
      "CompBasePremium": "750"
    }
  },
  {
    "productCode": "autoBundle",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2"
    ],
    "attributes": {
      "avMake": "Audi",
      "avValue": "20000",
      "avYear": "2020",
      "AutoValueFactor": 5,
      "AutoAgeFactor": 6
    }
  },
  {
    "productCode": "autoDriver",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2",
      "AutoDriver"
    ],
    "relatedRecords": [
      {
        "relatedRecordId": "001xx000003GYiEAAW",
        "relatedRecordObjName": "Account"
      },
      {
        "relatedRecordId": "001xx000003GYiEAAV",
        "relatedRecordObjName": "Contact"
      }
    ]
  },
  {
    "productCode": "autoCollision",
    "instanceKeys": [
      "AutoRoot",
      "AutoBundle2",
      "AutoCollision1"
    ],
    "attributes": {
      "CollisionBasePremium": "120"
    }
  }
]
}

```

```

    ]
  }

```

Properties

Name	Type	Description	Required or Optional	Available Version
accountId	String	ID of the account associated with the quote that's being created.	Optional	63.0
additionalFields	Map<String, Object>	Additional fields that are used to create insurance quote.	Optional	63.0
configOptions	Insurance Quote Config Options []	The JSON structure that includes the options for configuration.	Optional	63.0
contextId	String	ID of the context that's used for creating the quote.	Optional	63.0
effectiveDate	String	Date that the quote is effective from.	Required	63.0
expirationDate	String	Expiration date of the quote.	Optional	63.0
opportunityId	String	ID of the opportunity associated with the quote.	Optional	63.0
options	Insurance Quote Options []	Insurance Quote Options.	Optional	63.0
priceBookId	String	ID of the pricebook associated with the quote.	Optional	63.0
pricingProcedure	String	Pricing Procedure for insurance quote.	Optional	63.0
quoteInputs	Insurance Quote Input Node []	The JSON structure that contains user inputs for the products for which the quote is being created.	Required	63.0
quoteName	String	Name of the quote.	Optional	63.0
ratingDate	String	Date of the rating quote.	Optional	63.0
salesTransactionId	String	ID of the sales transaction type.	Optional	63.0
transactionType	String	Type of the transaction.	Optional	63.0

Product Clause Update Input

Input representation for updating a product clause.

JSON example

```

{
  "effectiveDate": "2026-01-01",
  "expirationDate": "2026-12-31",
  "variableMappings": [

```

```
{
  {
    "id": "a1Bxx0000000001AAA",
    "variable": "LimitAmount",
    "attribute": "NewCoverageLimit"
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveDate	String	Date from which the product clause is valid.	Optional	65.0
expirationDate	String	Date from which the product clause is no longer valid.	Optional	65.0
variable Mappings	Product Clause Variable Mapping Input[]	Variable mappings for the product clause that must be updated.	Optional	65.0

Product Clause Variable Mapping Input

Input representation for mapping an insurance clause variable to an insurance product variable.

JSON example

```
{
  "id": "a1Bxx0000000001AAA",
  "variable": "LimitAmount",
  "variableType": "Currency",
  "attribute": "CoverageLimit"
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
attribute	String	Attribute that's mapped to the variable of the Insurance Product Clause record.	Optional	65.0
id	String	ID of the Insurance Product Clause Variable Mapping record.	Optional	65.0
variable	String	Variable that's associated with the Insurance Product Clause record.	Required	65.0
variableType	String	Type of the variable that's associated with the attribute.	Optional	65.0

Product Rating Patch Input

Input representation of product rating update details.

JSON example

```
{
  "contextId": "a49103b233638b918217c74eced69220e534e160c3d3d35bc8cf6608515aeab3",
  "ratingOptions": {
    "executePricing": true,
    "executeConfigurationRules": false
  },
  "updatedNodes": [
    {
      "instanceKeys": [
        "AutoSilver",
        "BMW",
        "Tim1"
      ],
      "productCode": "driver",
      "attributes": {
        "DriverAccidentPoints": 1,
        "AgeFirstLicensed": 50,
        "DriverMVRPoints": 1
      },
      "targetRecords": [
        "003DR000002ov8NYAQ",
        "001DR000003avQCYAY"
      ]
    }
  ],
  "addedNodes": [
    {
      "instanceKeys": [
        "AutoSilver",
        "Tesla"
      ],
      "productCode": "auto",
      "attributes": {
        "Year": "2022",
        "Model": "Y"
      }
    },
    {
      "instanceKeys": [
        "AutoSilver",
        "Tesla",
        "collision1"
      ],
      "productCode": "collision",
      "attributes": {
        "Limit": 500,
        "Deductible": 1000
      }
    }
  ],
}
```

```

    {
      "instanceKeys": [
        "AutoSilver",
        "Tesla",
        "Driver1"
      ],
      "productCode": "driver",
      "attributes": {
        "DriverAccidentPoints": 5,
        "AgeFirstLicensed": 30,
        "DriverMVRPoints": 1
      },
      "targetRecords": [
        "003DR000002oxCfYAI",
        "001DR000003azdpYAA"
      ]
    },
    {
      "instanceKeys": [
        "AutoSilver",
        "Audi",
        "Collision2"
      ],
      "productCode": "collision",
      "attributes": {
        "Limit": 500,
        "Deductible": 1000
      }
    }
  ]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
addedNodes	Rating Input Node []	List of added nodes.	Optional	63.0
additionalFields	Map<String, Object>	Map of additional fields that must be updated in the quote context.	Optional	63.0
contextId	String	Context ID.	Required	63.0
deletedNodes	Rating Input Node []	List of deleted nodes.	Optional	63.0
effectiveDate	String	Date used to find the active pricing procedure for pricing. The default value is the current date.	Optional	63.0
ratingDate	String	Date that's used to find the active pricing procedure.	Optional	63.0
ratingOptions	Rating Options Input []	Options for rating.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
updatedNodes	Rating Input Node []	List of updated nodes.	Optional	63.0

Product Rating Post Input

Input representation of the product rating details.

JSON example

```
{
  "transactionType": "AutoTransactionType",
  "additionalFields": {
    "QuoteAccount": "001ZM000002ZNnBYAW"
  },
  "ratingOptions": {
    "executeConfigurationRules": false
  },
  "ratingInputs": [
    {
      "productCode": "auto",
      "instanceKeys": [
        "Auto1"
      ],
      "attributes": {
        "Make": "BMW",
        "AutoValue": "20000",
        "Year": "2024",
        "Model": "SUV"
      }
    },
    {
      "productCode": "comprehensive",
      "instanceKeys": [
        "Auto1",
        "Comprehensive"
      ],
      "attributes": {
        "Limit": "2000",
        "Deductible": "100"
      }
    }
  ],
  "rootProductCodes": [
    "auto-ECO1x",
    "auto-ECO2x"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Map of additional fields that must be updated in the quote context.	Required if the generated context ID is used to issue a policy	63.0
effectiveDate	String	Date used to find the active pricing procedure for pricing. The default value is the current date.	Optional	63.0
ratingInputs	Rating Input Node []	JSON that contains the attribute values of each context node.	Required	63.0
ratingOptions	Rating Options Input []	Options for rating.	Optional	63.0
rootProductCodes	String []	Root product codes that are used with the <code>ratingInputs</code> to create rating result. The created rating result is used as request in the insurance product rating API.	Optional	66.0
transactionType	String	Name of the transaction processing type.	Required	63.0

Product Rating Request Input

Input representation for getting rating input from product IDs.

JSON example

```
{
  "productIds": [
    "01txx0000006i3DAAQ"
  ],
  "salesTransactionTypeId": "0YWxx000000001GAAQ",
  "transactionType": "Auto",
  "ratingOptions": {
    "executeConfigurationRules": true,
    "executePricing": true,
    "returnContextJSON": false,
    "returnProductDetails": false
  },
  "effectiveDate": "2025-01-01T00:00:00.000Z",
  "ratingDate": "2025-01-01T00:00:00.000Z",
  "additionalFields": {
    "customField1": "value1"
  },
  "rootProductCodes": [
    "AUTO_POLICY"
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
additionalFields	Map<String, Object>	Additional fields to generate rating input.	Optional	66.0
effectiveDate	Date	Date that's used to find the active pricing procedure for pricing. The default value is the current date.	Optional	66.0
productIds	String[]	IDs of the products for which you want to generate rating input.	Required	66.0
ratingDate	Date	Date that's used to calculate rating.	Optional	66.0
ratingOptions	Rating Options Input	Options to control rating behavior.	Optional	66.0
rootProduct Codes	String[]	Root product codes that are used to create rating inputs.	Optional	66.0
salesTransaction TypeId	String	ID of the sales transaction type for which you want to generate rating request payload.	Optional	66.0
transaction Type	String	Name of the transaction processing type.	Optional	66.0

Product Surcharge Bulk Post Input

Input representation of the details of the product surcharge records.

JSON example

```
{
  "records": [
    {
      "name": "AutoVehicleTax",
      "surchargeId": "01cLO000000EMslOKL",
      "rootProductId": "01tDR000000EMslYAG",
      "productPath": "01tDR000000EMslYAG/11BDR00000000T72AI",
      "effectiveFromDate": "2024-05-30T10:30:00",
      "effectiveToDate": "2024-11-01T10:30:00",
      "isProrationEligible": true,
      "isRefundEligible": true,
      "isActive": true,
      "ruleApiName": "autoVehicleTaxRule",
      "ruleCriteria": [
        {
          "conditions": [
            {
              "contextTagName": "SalesTransactionItemAttribute",
```

```

        "operator": "LessThan",
        "conditionIndex": 1,
        "attributeName": "Annual Milage",
        "attributePicklistValueId": null,
        "type": "Attribute",
        "attributeId": "0tjDR00000000VGAY",
        "dataType": "Number",
        "values": [
            "50000"
        ]
    },
    ],
    "rootObjectId": "01tDR000000EMslYAG",
    "productRelatedComponent": "",
    "criteriaIndex": 1,
    "sourceContextTagName": "ProductBasedOn",
    "sourceOperator": "Equals",
    "sourceDataType": "String",
    "sourceValues": [
        "11BDR00000000T72AI"
    ]
},
{
    "conditions": [
        {
            "contextTagName": "SalesTransactionItemAttribute",
            "operator": "LessThan",
            "conditionIndex": 1,
            "attributeName": "Comprehensive Deductible",
            "type": "Attribute",
            "attributePicklistValueId": null,
            "attributeId": "0tjDR00000000UaYAI",
            "dataType": "Number",
            "values": [
                "$5000"
            ]
        }
    ],
    "rootObjectId": "01tDR000000EMslYAG",
    "productRelatedComponent": "",
    "criteriaIndex": 2,
    "sourceContextTagName": "Product",
    "sourceOperator": "Equals",
    "sourceDataType": "String",
    "sourceValues": [
        "01tDR000000EMsoYAG"
    ]
}
]
}
]
}
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
records	Product Surcharge Post Input[]	List of the product surcharges.	Required	63.0

Product Surcharge Patch Input

Input representation for product surcharge patch request.

JSON example

```
{
  "name": "AutoVehicleTax",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "isProrationEligible": true,
  "isRefundEligible": true,
  "isActive": true,
  "productPath": "01tDR000000EMslYAG/11BDR00000000T72AI", // "Auto Gold/Auto Vehicle"

  "ruleDefinition": [ // ruleDefinition will be replaced as a whole. Updating only
    one condition is not supported
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGYAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG", // Auto Gold.
      "productRelatedComponent" : "",
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
      "sourceDataType": "String",
      "sourceValues": [
        "11BDR00000000T72AI" // Auto Vehicle.
      ]
    },
    {
      "conditions": [
        {
```

```

    "contextTagName": "SalesTransactionItemAttribute",
    "operator": "LessThan",
    "conditionIndex": 1,
    "attributeName": "Comprehensive Deductible",
    "type": "Attribute",
    "attributePicklistValueId": null,
    "attributeId": "0tjDR00000000UaYAI",
    "dataType": "Number",
    "values": [
    "$5000"
    ]
  },
  "rootObjectId": "01tDR000000EMsIYAG", // Auto Bundle
  "productRelatedComponent" : "",
  "criteriaIndex": 2,
  "sourceContextTagName": "Product",
  "sourceOperator": "Equals",
  "sourceDataType": "String",
  "sourceValues": [
    "01tDR000000EMsoYAG" // Comprehensive Product.
  ]
}
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Effective from date of the product surcharge.	Optional	63.0
effectiveToDate	String	Effective to date of the product surcharge.	Optional	63.0
isActive	Boolean	Indicates whether the product surcharge is active (<code>true</code>) or not (<code>false</code>). Currently, only active product surcharges are supported.	Optional	63.0
isProrationAllowed	Boolean	Indicates whether the product surcharge is allowed for proration (<code>true</code>) or not (<code>false</code>).	Optional	63.0
isRefundAllowed	Boolean	Indicates whether the product surcharge is allowed for refund (<code>true</code>) or not (<code>false</code>).	Optional	63.0
name	String	Name of the product surcharge.	Optional	63.0
productPath	String	A path containing the product IDs that roll up to the parent product.	Optional	63.0

Name	Type	Description	Required or Optional	Available Version
productRelatedComponentId	String	Product related component associated with the product surcharge.	Optional	63.0
ruleApiName	String	Rule API name associated with product surcharge. You can set the ruleApiName with the POST or PATCH method. Once set, this parameter can't be changed. If you set the ruleApiName in the request body, you must also set the ruleCriteria.	Optional	63.0
ruleCriteria		Rule criteria for the dynamic rule.	Optional	63.0

Product Surcharge Post Input

Input representation of product surcharge details.

JSON example

```
{
  "name": "AutoVehicleTax",
  "surchargeId": "01cLO000000EMslOKL",
  "rootProductId": "01tDR000000EMslYAG",
  "productPath": "01tDR000000EMslYAG/11BDR00000000T72AI",
  "effectiveFromDate": "2024-05-30T10:30:00",
  "effectiveToDate": "2024-11-01T10:30:00",
  "isProrationEligible": true,
  "isRefundEligible": true,
  "isActive": true,
  "ruleApiName": "autoVehicleTaxRule",
  "ruleCriteria": [
    {
      "conditions": [
        {
          "contextTagName": "SalesTransactionItemAttribute",
          "operator": "LessThan",
          "conditionIndex": 1,
          "attributeName": "Annual Milage",
          "attributePicklistValueId": null,
          "type": "Attribute",
          "attributeId": "0tjDR00000000VGYAY",
          "dataType": "Number",
          "values": [
            "50000"
          ]
        }
      ],
      "rootObjectId": "01tDR000000EMslYAG",
      "productRelatedComponent": "",
      "criteriaIndex": 1,
      "sourceContextTagName": "ProductBasedOn",
      "sourceOperator": "Equals",
    }
  ]
}
```

```

        "sourceDataType": "String",
        "sourceValues": [
            "11BDR00000000T72AI"
        ]
    }

```

Properties

Name	Type	Description	Required or Optional	Available Version
effectiveFromDate	String	Effective from date of the product surcharge.	Optional	63.0
effectiveToDate	String	Effective to date of the product surcharge.	Optional	63.0
isActive	Boolean	Indicates whether the product surcharge is active (<code>true</code>) or not (<code>false</code>). Currently, only active product surcharges are supported	Optional	63.0
isProrationAllowed	Boolean	Indicates whether the product surcharge is allowed for proration (<code>true</code>) or not (<code>false</code>).	Optional	63.0
isRefundAllowed	Boolean	Indicates whether the product surcharge is allowed for refund (<code>true</code>) or not (<code>false</code>).	Optional	63.0
name	String	Name of the product surcharge.	Required	63.0
productPath	String	A path containing the product IDs that roll up to the parent product.	Required	63.0
productRelatedComponentId	String	ID of the product related component associated with the product surcharge.	Optional	63.0
rootProductId	String	ID of the root product associated with the product surcharge.	Required	63.0
ruleApiName	String	Rule API name associated with product surcharge.	Optional	63.0
ruleCriteria		Rule criteria of the dynamic rule for the product surcharge.	Optional	63.0
surchargeId	String	ID of the surcharge associated with the product surcharge.	Required	63.0

Product Clause Input

Input representation for the details of a single product clause to be created.

JSON example

```
{
  "apiName": "DeductibleWaiver_2025",
  "name": "Deductible Waiver Clause",
  "effectiveDate": "2025-01-01",
  "expirationDate": "2025-12-31",
  "insuranceClauseId": "0j1xx0000000004DDD",
  "rootProduct": "01t0000000003CCC",
  "productPath": "01t0000000003DDD",
  "variableMappings": [
    {
      "variable": "DeductibleAmount",
      "variableType": "Currency",
      "attribute": "WaiverAmount"
    }
  ]
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
apiName	String	API name of the product clause.	Required	65.0
effectiveDate	String	Date from which the clause is valid.	Required	65.0
expirationDate	String	Date from which the clause is no longer valid.	Required	65.0
insuranceClauseId	String	Insurance clause component that's associated with the product clause.	Required	65.0
name	String	Name of the product clause.	Required	65.0
productPath	String	Product path of the product clause.	Required	65.0
productRelatedComponentId	String	Product-related component that's associated with the product clause.	Optional	65.0
rootProduct	String	Root product of the product clause.	Required	65.0
variableMappings	Product Clause Variable Mapping Input[]	List of variable mappings for the product clause.	Optional	65.0

Rating Input Node

Input representation of details of context node.

JSON example

```
{
  "ratingInputs": [
    {
```

```

    "instanceKeys": [
      "AutoRoot"
    ],
    "productCode": "AUTOROOT",
    "attributes": {
      "Deductible": 1000
    }
  },
  {
    "instanceKeys": [
      "AutoRoot",
      "Audi"
    ],
    "productCode": "AUTO",
    "attributes": {
      "Mileage": 20000
    }
  },
  {
    "instanceKeys": [
      "AutoRoot",
      "Audi",
      "Tom"
    ],
    "productCode": "DRIVER",
    "attributes": {
      "DriverAge": 50,
      "DriverAccident": 0
    },
    "targetRecords": [
      "003xx000004WkvEAAS"
    ]
  }
]
}

```

Properties

Name	Type	Description	Required or Optional	Available Version
<code>additionalFields</code>	<code>Map<String, Object></code>	Map of additional fields that must be updated in the quote context.	Optional	63.0
<code>attributes</code>	<code>Map<String, Object></code>	Map of attributes key-value pairs that's used to determine pricing.	Optional	63.0
<code>instanceKeys</code>	<code>String[]</code>	List of instance keys for context path.	Required	63.0
<code>productCode</code>	<code>String</code>	Unique code to identify a product.	Required	63.0
<code>reusable</code>	<code>Boolean</code>	Indicate whether the node is reusable across multiple roots (<code>true</code>) or not (<code>false</code>).	Optional	66.0

Name	Type	Description	Required or Optional	Available Version
targetRecords	String[]	List of target records.	Optional	63.0

Rating Options Input

Input representation details of rating service options.

JSON example

```
{
  "ratingOptions": {
    "executeConfigurationRules": true,
    "executePricing": true,
    "returnContextJSON": true,
    "returnProductDetails": true
  }
}
```

Properties

Name	Type	Description	Required or Optional	Available Version
executeConfigurationRules	Boolean	Indicates whether to execute configuration rules.	Optional	63.0
executePricing	Boolean	Indicates whether to execute pricing.	Optional	63.0
pricingProcedure	String	Name of the associated pricing procedure. This property is deprecated in API version 66.0 and later.	Optional	63.0
returnContextJSON	Boolean	Indicates whether context details must be shown in the response (<code>true</code>) or not (<code>false</code>).	Optional	66.0
returnProductDetails	Boolean	Indicates whether product details must be shown in the response (<code>true</code>) or not (<code>false</code>).	Optional	66.0
returnRatingResults	Boolean	Indicates whether rating results must be shown in the response (<code>true</code>) or not (<code>false</code>).	Optional	66.0

Responses

Learn more about the available response bodies of Insurance Quoting APIs.

[Add Eligible Product Clauses Details](#)

Output representation for adding eligible product clauses.

[Context JSON](#)

Output representation for the context JSON.

[Error Detail](#)

Output representation of the error details.

[Get Quote](#)

Output representation of the details of the get quote details.

[Insurance App Usage Assignment](#)

Output representation for the insurance app usage assignment.

[Insurance Configurator Message](#)

Output representation of configurator messages.

[Insurance Get Eligible Product Clauses](#)

Output representation for eligible clauses.

[Insurance Invoke Underwriting Rule](#)

Output representation for invoke underwriting rule response.

[Insurance Invoke Underwriting Rule Set Result](#)

Output representation for invoke underwriting rule set result response.

[Insurance Product Clause Details](#)

Output representation for insurance product clause.

[Insurance Quote](#)

Output representation of the details of the insurance quote.

[Insurance Sales Transaction](#)

Output representation for the insurance sales transaction.

[Insurance Sales Transaction Item](#)

Output representation for the insurance sales transaction item.

[Insurance Sales Transaction Item Attribute](#)

Output representation for the details of the insurance sales transaction item attribute.

[Insurance Sales Transaction Item Clause](#)

Output representation for the details of the insurance sales transaction item clause.

[Insurance Sales Transaction Item Relationship](#)

Output representation for the details of the Insurance Sales Transaction Item Relationship.

[Insurance Sales Transaction Item Related Object](#)

Output representation for the details of the Insurance Sales Transaction Item Related Object.

[Insurance Sales Transaction Item Tax](#)

Output representation for the details of the Insurance Sales Transaction Item Tax.

[Insurance Underwriting Rule Result](#)

Underwriting rule evaluation result.

[Insurance UI Treatments](#)

Output representation for the details of the insurance UI treatments.

[Product Clause Generation Details](#)

Output representation of the details for a clause generation request.

[Product Clause Creation Details](#)

Output representation for the bulk creation of product clauses.

[Product Clauses](#)

Output representation for a successfully created product clause.

[Product Clause Error](#)

Output representation of an error that occurred during the product clause operation.

[Product Clause Update Details](#)

Output representation of the details for an updated product clause.

[Product Rating Response](#)

Output representation for the Product Rating Connect API.

[Product Surcharge List](#)

Output Representation for List of ProductSurcharge

[Product Surcharge Patch](#)

Output representation of product surcharge update.

[Product Rating Details Response](#)

Output representation that contains the constructed rating request payload.

[Rating Input Node Response](#)

Output representation of a single node in the rating input hierarchy.

[Rating Options](#)

Output representation of the rating options that are returned in the response.

[Rating Result](#)

Output representation for the details of the product rating pricing results.

Add Eligible Product Clauses Details

Output representation for adding eligible product clauses.

JSON example

```
{
  "success": true,
  "quoteLineItemInsProductClauseIds": [
    "a1Cxx0000000001AAA"
  ],
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Product Clause Error[]	List of errors that occurred during processing.	Big, 65.0	65.0

Property Name	Type	Description	Filter Group and Version	Available Version
quoteLineItem InsProduct ClauseIds	String[]	List of unique IDs for all insurance product clauses that are created for the specified quote line item	Big, 65.0	65.0
success	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Context JSON

Output representation for the context JSON.

Property Name	Type	Description	Filter Group and Version	Available Version
salesTransactions	Insurance Sales Transaction[]	List of JSON representations for the context node.	Small, 66.0	66.0

Error Detail

Output representation of the error details.

Property Name	Type	Description	Filter Group and Version	Available Version
code	String	Error code.	Big, 63.0	63.0
message	String	Message stating the reason for error, if any.	Big, 63.0	63.0

Get Quote

Output representation of the details of the get quote details.

JSON example

This example shows a sample response when you pass a quote ID in the get quote details request.

```
{
  "contextId": "fa917d24acfaa40c14943d7009f096b2952c974a6e940d1302c465cd3abae3d",
  "errors": [],
  "insuranceContext": {
    "SalesTransaction": [
      {
        "Status": "Draft",
        "Account": null,
        "BillingCity": null,
        "Subtotal": 1350,
        "businessObjectType": "Quote",
        "SalesTransactionName": "Multi Auto",
        "TotalAmount": 1350,

```

```

"ShippingStreet": null,
"SalesTransactionItem": [
  {
    "ProrationPolicy": "0muxx000000004sAAA",
    "ProductName": "Medical Payments",
    "ProductCode": "STA-3",
    "ProductSellingModel": "0jPxx000000004sEAA",
    "SpecificationType": "Coverage",
    "TaxTreatment": "1ttxx000000004rAAA",
    "SalesTransactionItemParent": "0Q0xx0000004CE1CAM",
    "Quantity": 1,
    "Product": "01txx0000006i7KAAQ",
    "businessObjectType": "QuoteLineItem",
    "PricebookEntry": "0luxx0000008ybrAAA",
    "PricingTermCount": 1,
    "StartDate": "2023-10-19T00:00:00Z",
    "NetUnitPrice": 750,
    "UnitPrice": 750,
    "id": "0QLxx0000004CIpGAM"
  },
  {
    "ProrationPolicy": null,
    "ProductName": "Collision",
    "ProductCode": "STA-2",
    "ProductSellingModel": "0jPxx000000004tEAA",
    "SpecificationType": "Coverage",
    "TaxTreatment": "1ttxx000000004rAAA",
    "SalesTransactionItemParent": "0Q0xx0000004CE1CAM",
    "Quantity": 1,
    "Product": "01txx0000006i7JAAQ",
    "businessObjectType": "QuoteLineItem",
    "PricebookEntry": "0luxx0000008ybsAAA",
    "PricingTermCount": 1,
    "StartDate": null,
    "NetUnitPrice": 500,
    "UnitPrice": 500,
    "id": "0QLxx0000004CIqGAM"
  }
],
"BillingCountry": null,
"BillingStreet": null,
"Pricebook": "01sxx0000005pyfAAA",
"Opportunity": null,
"ShippingPostalCode": null,
"SalesTransactionSource": "0Q0xx0000004CE1CAM",
"ShippingCountry": null,
"ShippingCity": null,
"ShippingState": null,
"BillingPostalCode": null,
"id": "0Q0xx0000004CE1CAM",
"Contract": null,
"BillingState": null

```

```

]

```

} }

This example shows a sample response when you pass the filter and non-filter request parameters.

```
//GET
/asset/transaction/00000000i18tq18g002917545349255616564529f415d45c4b1b1ea08747a2195?filePolicy[e]=FreeLookPeriodClause
{
  "contextId": "00000000i18tq18g002917545349255616564529f415d45c4b1b1ea08747a2195",
  "errors": [],
  "insuranceContext": {
    "SalesTransaction": [
      {
        "Status": "Draft",
        "Account": null,
        "BillingCity": null,
        "Subtotal": 1350,
        "businessObjectType": "Quote",
        "SalesTransactionName": "Multi Auto",
        "TotalAmount": 1350,
        "ShippingStreet": null,
        "SalesTransactionItem": [
          {
            "businessObjectType": "QuoteLineItem",
            "Product": "01txx0000006i3EAAQ",
            "DYN__License_State": null,
            "ValidationResult": null,
            "DYN__AssetRecordName": "02ixx0000004HI3AAM",
            "PolicyPremium": null,
            "SalesTransactionSourceAsset": null,
            "PeriodBoundaryStartMonth": null,
            "ProratedQLITaxAmount": 0,
            "id": "0QLxx0000004CozGAE",
            "STICurrencyIsoCode__std": null,
            "IntermediateVariableOne": null,
            "ProductCode": "auto",
            "SalesTrxnItemRelatedObject": [
              {
                "RelatedObject": "02ixx0000004HI3AAM",
                "ParentReference": "0QLxx0000004CozGAE",
                "RelatedObjectName": "Asset",
                "QuoteLineItem": "0QLxx0000004CozGAE",
                "businessObjectType": "QuoteLineItemRelatedObject",
                "id": "1A2xx0000004CNZCA2",
                "UsageType": "ExtendedAttribute"
              }
            ]
          },
          {
            "SalesTransactionItemInsClause__std": [
              {
                "SalesTranxItemInsClauseConstraintModelName__std": "Q2Blitz_V1",
                "ParentReference": "0QLSB000000b8Q34AI",
                "SalesTranxItemInsClauseName__std": "FreeLookPeriodClause",
                "businessObjectType": "QuoteLineItemInsProductClause",

```



```

        "SalesTranxItemInsClauseText__std": "On Collision, Auto Added, only
eligible when collision deductible is 100. Deductible is 100.",
        "InsuranceProductClause__std": "1T8SB00000000Xt0AI",
        "id": "9TeSB000004Ny7d0AC",
        "SalesTranxItemInsClauseEffectiveDate__std": "2025-08-01",
        "SalesTranxItemInsClauseExpirationDate__std": "2025-08-31",
        "SalesTranxItemInsClauseCreationMethod__std": "AutoAdded",
        "SalesTranxItemInsClauseParent__std": "0QLSB0000000b8Q34AI",
        "SalesTranxItemInsClauseDescription__std": null
    }
],
"Subtotal": 320,
"LineItemPath": "0QLxx0000004Cp1GAE/0QLxx0000004CozGAE",
"PricebookEntry": "0luxx0000008yXWAAy",
"DiscountAmount": null,
"NetUnitPrice": 320,
"DYN__Is_Electric_Vehicle": false,
"ParentReference": "0Q0xx0000004CIiCAM",
"DYN__Model": "A3",
"EndQuantity": 1,
"NetTotalPrice": 320,
"TotalLineAmount": 320,
"UnitPrice": 320,
"DYN__Auto_Value": 100000,
"SalesTrxnItemRelationship": [
    {
        "ParentReference": "0QLxx0000004CozGAE",
        "AssociatedItem": "0QLxx0000004CozGAE",
        "MainItemRole": "Bundle",
        "ProductRelatedComponent": "0dSxx000000001EEAQ",
        "businessObjectType": "QuoteLineRelationship",
        "AssociatedItemPricing": "IncludedInBundlePrice",
        "ProductRelationshipType": "0yoxx0000000001AAA",
        "RootItemProductCode": "autoSilver",
        "IsPriceInclusive": true,
        "RootItem": "0QLxx0000004Cp1GAE",
        "RootItemProductSellingModel": "0jPxx0000000001EAA",
        "MainItem": "0QLxx0000004Cp1GAE",
        "AssociatedQuantScaleMethod": "Proportional",
        "id": "0yQxx00000000RSEAY",
        "AssociatedItemRole": "BundleComponent",
        "SalesTrnItemRelationshipParent": "0Q0xx0000004CIiCAM",
        "MainItemProduct": "01txx0000006i3DAAQ",
        "MainItemProductSellingModel": "0jPxx0000000001EAA",
        "RootItemProduct": "01txx0000006i3DAAQ"
    }
],
"ProductSpec": "InsuredItem",
"SalesTransactionItemAttribute": [
    {
        "AttributeValue": null,
        "ParentReference": "0QLxx0000004CozGAE",
        "hasObjectRef": null,
        "businessObjectType": "QuoteLineItemAttribute",
    }
]

```

```

        "AttributeDefinitionCode": "LicenseNumber",
        "SalesTransactionItemAttrParent": "0QLxx0000004CozGAE",
        "AttributeDeveloperName": "License_Number",
        "AttributeKey": "0tjxx0000000013AAA",
        "AttributePicklistValue": null,
        "IsPriceImpacting": null,
        "AttributeName": "License Number",
        "id": "0zuxx00000001vWAAQ"
    }
],
"BillingCountry": null,
"StandardTaxAmount": 0,
"BillingStreet": null,
"PriceAdjustmentSchedule": null,
"SalesTransactionSource": "0Q0xx0000004CIiCAM",
"TotalStandardAmount": 640,
"EffectiveDate": null
}
],
"DeletedSTIIds": []
}
]
}
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
contextId	String	ID of the insurance context.	Small, 63.0	63.0
errors	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
insuranceContext	Map<String, Object>	List of insurance context details.	Small, 63.0	63.0

Insurance App Usage Assignment

Output representation for the insurance app usage assignment.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the application context of a sales transaction record.	Small, 66.0	66.0

Insurance Configurator Message

Output representation of configurator messages.

Property Name	Type	Description	Filter Group and Version	Available Version
message	String	Insurance quote configurator message.	Small, 63.0	63.0
type	String	Type of message.	Small, 63.0	63.0
recordId	String	Record ID.	Small, 66.0	66.0

Insurance Get Eligible Product Clauses

Output representation for eligible clauses.

JSON example

```
{
  "errors": [],
  "insuranceProductClauses": [
    {
      "category": "Manual",
      "clauseName": "Manual Clause 1",
      "description": "This is a manually added clause.",
      "effectiveDate": "2025-10-01T12:00:00.000Z",
      "expirationDate": "2026-10-01T12:00:00.000Z",
      "insuranceProductClauseId": "a4Lxx000000Cde1EAC",
      "type": "Clause"
    }
  ],
  "success": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Product Clause Error[]	List of errors that occurred during the API operation.	Big, 65.0	65.0
insurance Product Clauses	Insurance Product Clause Details[]	List of insurance product clauses.	Big, 65.0	65.0
success	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Insurance Invoke Underwriting Rule

Output representation for invoke underwriting rule response.

JSON example

This example shows a sample success response.

```
{
  "objectId": "0Q0SM0000001uL70AI",
  "objectName": "Quote",
```

```

"stageTransitionName": "Draft To Submitted",
"fromStage": "Draft",
"toStage": "Submitted",
"ruleSetResults": [
  {
    "productId": "01tSM0000050FyGYAW",
    "evaluationCriteria": "1 OR 2",
    "isSuccess": true,
    "ruleResults": [
      {
        "ruleApiName": "CollisionRule",
        "isSuccess": true
      },
      {
        "ruleApiName": "AutoSilverRule",
        "isSuccess": true
      }
    ]
  },
  {
    "productId": "01tSM0000050FyJYAW",
    "evaluationCriteria": "1",
    "isSuccess": true,
    "ruleResults": [
      {
        "ruleApiName": "MedicalRule1",
        "isSuccess": true
      }
    ]
  }
],
"errors": [],
"isSuccess": true
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
fromStage	String	The from stage.	Small, 63.0	63.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>). This parameter is set to <code>true</code> when all individual rule set results are successful.	Small, 63.0	63.0
objectId	String	ID of the quote or insurance policy.	Small, 63.0	63.0
objectName	String	Name of the object (Quote or Insurance Policy).	Small, 63.0	63.0

Property Name	Type	Description	Filter Group and Version	Available Version
ruleSetResult	Insurance Underwriting Rule Set Results []	List of underwriting rule set evaluation results.	Small, 66.0	66.0
stageTransition Name	String	ID of the stage transition.	Small, 63.0	63.0
stageTransition Underwriting EvaluationId	String	ID of the underwriting rule evaluation.	Small, 63.0	63.0
toStage	String	The to stage.	Small, 63.0	63.0

Insurance Invoke Underwriting Rule Set Result

Output representation for invoke underwriting rule set result response.

JSON example

This example shows a sample success response.

```
{
  "rulesetResults": [
    {
      "productId": "01tSM0000050FyGYAW",
      "evaluationCriteria": "1 OR 2",
      "isSuccess": true,
      "ruleResults": [
        {
          "ruleApiName": "CollisionRule",
          "isSuccess": true
        },
        {
          "ruleApiName": "AutoSilverRule",
          "isSuccess": false
        }
      ]
    },
    {
      "productId": "01tSM0000050FyJYAW",
      "evaluationCriteria": "1",
      "success": true,
      "ruleResults": [
        {
          "ruleApiName": "MedicalRule1",
          "isSuccess": true
        }
      ]
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
productId	String	ID of the product that's evaluated by the rule set.	Small, 66.0	66.0
evaluationCriteria	String	Evaluation criteria expression that's used during the rule set evaluation.	Small, 66.0	66.0
ruleResult	Insurance Underwriting Rule Results[]	List of underwriting rule evaluation results.	Small, 66.0	66.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 66.0	66.0

Insurance Product Clause Details

Output representation for insurance product clause.

JSON example

```
{
  "category": "Manual",
  "clauseName": "Manual Clause 1",
  "description": "This is a manually added clause.",
  "effectiveDate": "2025-10-01T12:00:00.000Z",
  "expirationDate": "2026-10-01T12:00:00.000Z",
  "insuranceProductClauseId": "a4Lxx000000Cde1EAC",
  "type": "Clause"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
category	String	Category of the insurance clause, such as manual or auto-added.	Big, 65.0	65.0
clauseName	String	Name of the insurance product clause.	Big, 65.0	65.0
description	String	Description of the insurance product clause.	Big, 65.0	65.0
effectiveDate	String	Date from which the product clause is valid.	Big, 65.0	65.0
expirationDate	String	Date from when the product clause is no longer valid.	Big, 65.0	65.0
insuranceProductClauseId	String	ID of the insurance product clause.	Big, 65.0	65.0
type	String	Type of insurance product provision, such as clause or exclusion.	Big, 65.0	65.0

Insurance Quote

Output representation of the details of the insurance quote.

JSON example

```
{
  "quoteId" : "0Q0xx0000004E2mCAE",
  "contextId" : "98977d60d58a614485280fe430afded87290baa9a6ad6bfec889df75ef11008e"
  "errors" : [],
  "success" : true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
configMessages	Insurance Configurator Message[]	List of insurance quote configurator messages.	Small, 63.0	63.0
contextId	String	ID of the context.	Small, 63.0	63.0
errors	Error Response	List of errors encountered during the processing of the API request.	Small, 63.0	63.0
quoteDetails	Map<String, Object>	Details of the insurance quote.	Small, 63.0	63.0
quoteId	String	ID of the quote.	Small, 63.0	63.0
success	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0

Insurance Sales Transaction

Output representation for the insurance sales transaction.

Property Name	Type	Description	Filter Group and Version	Available Version
appUsageAssignments	Insurance App Usage Assignment[]	Application context for a sales transaction record.	Small, 66.0	66.0
fields	Map<String, Object>	A key-value map that represents sales transaction record.	Small, 66.0	66.0
salesTransactionItems	Insurance Sales Transaction Items[]	List of individual line items that are associated with the sales transaction. In the default context definition, a sales transaction item represents a quote line item.	Small, 66.0	66.0

Insurance Sales Transaction Item

Output representation for the insurance sales transaction item.

Property Name	Type	Description	Filter Group and Version	Available Version
childNodes	Insurance Sales Transaction Item[]	List of child sales transaction items.	Small, 66.0	66.0
fields	Map<String, Object>	A key-value map that represents sales transaction item record.	Small, 66.0	66.0
~~salesTransactionAttributes~~	Insurance Sales Transaction Item Attribute[]	List of dynamic attributes of the sales transaction item in key-value pairs.	Small, 66.0	66.0
~~salesTransactionClasses~~	Insurance Sales Transaction Item Clause[]	List of clauses of the sales transaction item in key-value pairs.	Small, 66.0	66.0
~~salesTransactionRelatedObjects~~	Insurance Sales Transaction Item Related Object[]	List of related objects of the sales transaction item in key-value pairs.	Small, 66.0	66.0
~~salesTransactionRelationships~~	Insurance Sales Transaction Item Relationship[]	List of records that represent the relationship between the sales transaction items.	Small, 66.0	66.0
~~salesTransactionTaxes~~	Insurance Sales Transaction Item Tax[]	List of tax records of the sales transaction items.	Small, 66.0	66.0

Insurance Sales Transaction Item Attribute

Output representation for the details of the insurance sales transaction item attribute.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the dynamic attributes of the sales transaction item.	Small, 66.0	66.0

Insurance Sales Transaction Item Clause

Output representation for the details of the insurance sales transaction item clause.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the clauses of the sales transaction item.	Small, 66.0	66.0

Insurance Sales Transaction Item Relationship

Output representation for the details of the Insurance Sales Transaction Item Relationship.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the relationship between the sales transaction items.	Small, 66.0	66.0

Insurance Sales Transaction Item Related Object

Output representation for the details of the Insurance Sales Transaction Item Related Object.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the related objects of the sales transaction item.	Small, 66.0	66.0

Insurance Sales Transaction Item Tax

Output representation for the details of the Insurance Sales Transaction Item Tax.

Property Name	Type	Description	Filter Group and Version	Available Version
fields	Map<String, Object>	A key-value map that represents the tax record of the sales transaction item.	Small, 66.0	66.0

Insurance Underwriting Rule Result

Underwriting rule evaluation result.

Property Name	Type	Description	Filter Group and Version	Available Version
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Small, 63.0	63.0
ruleApiName	String	API name of the dynamic rule.	Small, 63.0	63.0

Insurance UI Treatments

Output representation for the details of the insurance UI treatments.

JSON example

This example shows a sample success response.

```
{
  "uiTreatments": [
    {
      "details": {
        "attributeId": "0tjxx0000000007AAA",
        "stiId": "0QLxx0000004CU0GAM",
        "attributePicklistValueId": "0v6xx0000000005AAA"
      },
      "uiTreatmentTarget": "Attribute_Picklist_Value",
      "uiTreatmentType": "Hide"
    },
    {
      "details": {
        "stiId": "ref_f0f2da7b_c431_482d_bf4b_599052f3a2e1"
      },
      "uiTreatmentTarget": "Component",
      "uiTreatmentType": "Disable"
    }
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
details	Map<String,String>	Key-value pair that specifies the items to apply the rules on, which includes these details. - ID of the sales transaction item - ID of the attribute - ID of the attribute picklist value	Small, 66.0	66.0
uiTreatmentTarget	String	Target of the UI treatment. Valid values are: - Component—Represents a product option or bundle component. - Quantity—Represents a quantity field. - Attribute—Represents a certain attribute of the product. - Attribute_Picklist_Value—Represents one of the picklist values of a product attribute.	Small, 66.0	66.0

Property Name	Type	Description	Filter Group and Version	Available Version
uiTreatmentType	String	Type of UI treatment to be performed. Valid values are: • <code>Hide</code>—Hide the associated target. • <code>Disable</code>—Disable the associated target.	Small, 66.0	66.0

Product Clause Generation Details

Output representation of the details for a clause generation request.

JSON example

```
{
  "success": true,
  "asyncBulkRequestId": "750xx0000000001AAA",
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
asyncBulkRequestId	String	Asynchronous bulk request ID for tracking the clause generation process.	Big, 65.0	65.0
errors	Product Clause Error[]	List of errors that occurred during processing.	Big, 65.0	65.0
success	Boolean	Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0

Product Clause Creation Details

Output representation for the bulk creation of product clauses.

JSON example

```
{
  "isSuccess": true,
  "productClauses": [
    {
      "productClauseId": "0j4xx0000000001AAA",
      "apiName": "AutoGlassClause_2025"
    }
  ],
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Product Clause Error[]	List of errors that occurred during processing.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the bulk creation operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0
productClauses	Product Clauses []	List of product clauses that are successfully created.	Big, 65.0	65.0

Product Clauses

Output representation for a successfully created product clause.

JSON example

```
{
  "productClauseId": "0j4xx0000000001AAA",
  "apiName": "AutoGlassClause_2025"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
apiName	String	API name of the product clause that's created.	Big, 65.0	65.0
productClauseId	String	ID of the product clause that's created.	Big, 65.0	65.0

Product Clause Error

Output representation of an error that occurred during the product clause operation.

JSON example

```
{
  "code": "DUPLICATE_API_NAME",
  "message": "A product clause with this API name already exists."
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
code	String	Error code that identifies the type of error.	Big, 65.0	65.0
message	String	Message that describes the error.	Big, 65.0	65.0

Product Clause Update Details

Output representation of the details for an updated product clause.

JSON example

```
{
  "isSuccess": true,
  "productClauseId": "0j4xx0000000001AAA",
  "errors": []
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Product Clause Error[]	List of errors that occurred during processing.	Big, 65.0	65.0
isSuccess	Boolean	Indicates whether the update operation is successful (<code>true</code>) or not (<code>false</code>).	Big, 65.0	65.0
productClauseId	String	ID of the product clause that's updated.	Big, 65.0	65.0

Product Rating Response

Output representation for the Product Rating Connect API.

JSON example

This example shows a sample success response.

```
{
  "contextId": "11jxx0000006i5gAAA",
  "pricingResult": {
    "Economy Renter 1": {
      "NetUnitPrice": 1000,
      "childNodes": []
    },
    "Economy Renter 2": {
      "NetUnitPrice": 2000,
      "childNodes": []
    }
  },
  "contextJSON": {
    "SalesTransaction": [
      {
        "Account": null,
        "businessObjectType": "Quote",
        "SalesTransactionName": "Test Quote",
        "ProratedAmount": 112.25,
        "StartDate": "2025-07-01",
        "CurrencyIsoCode__std": "USD",
        "InsuranceQuoteType": "New Business",
        "TransactionType": "AutoTransactionType",
        "TotalProratedAmount": 132.25,
        "Pricebook": "01sSG000009SJvNYAW",

```

```

    "Opportunity": null,
    "TotalAmount": 277.15,
    "EndDate": "2025-08-31",
    "SalesTransactionItem": [
      {
        "businessObjectType": "QuoteLineItem",
        "Product": "01tSG000008KnvSYAS",
        "id": "0QLSG000000dIx4AE",
        "ProductCode": "Auto Silver",
        "SalesTransactionItemAttribute": [
          {
            "AttributeKey": "0tjSG0000006q8GYAQ",
            "AttributeValue": "6.0",
            "ParentReference": "0QLSG000000dIx4AE",
            "hasObjectRef": null,
            "AttributePicklistValue": null,
            "IsPriceImpacting": null,
            "businessObjectType": "QuoteLineItemAttribute",
            "AttributeName": "Driver MVR Points",
            "id": "0zuSG000002If0SYAS",
            "AttributeDefinitionCode": "DriverMVRPoints",
            "SalesTransactionItemAttrParent": "0QLSG000000dIx4AE",
            "AttributeDeveloperName": "Driver_MVR_Points"
          }
        ],
        "LineItemPath": "0QLSG000000dIy04AE/0QLSG000000dIx4AE",
        "PricebookEntry": "01uSG000003chfXYAQ",
        "NetUnitPrice": 79.77,
        "ParentReference": "0Q0SG000000P3XN0A0",
        "UnitPrice": 469.6,
        "InstanceKey": "673ec36c-7d4a-4e8a-999a-6cf16a299f4a",
        "childNodes": [
          {
            "businessObjectType": "QuoteLineItem",
            "Product": "01tSG000008KnvSYAS",
            "id": "0QLSG000000dIx4AE",
            "ProductCode": "medPay",
            "SalesTransactionItemAttribute": [],
            "childNodes": []
          }
        ]
      }
    ],
    "TotalStandardAmount": 680.8
  }
],
},
"productDetails": {
  "auto": {
    "attributeCategories": [
      {
        "attributes": [
          {
            "attributeCategoryId": "0v3SG0000001ZToYAM",

```

```

      "attributeNameOverride": "Bodily Injury Per Person Limit",
      "attributePicklist": {
        "id": "0v5SG0000004YXzYAM",
        "values": [
          {
            "code": "BIPD500",
            "displayValue": "$500",
            "id": "0v6SG0000006Vl2YAE",
            "name": "BIPD500",
            "sequence": 1,
            "status": "Active",
            "textValue": "500.0"
          }
        ]
      },
      "code": "BodilyInjuryPerPersonLimit",
      "dataType": "PICKLIST",
      "developerName": "Bodily_Injury_Per_Person_Limit",
      "id": "0tjSG0000006q7RYAQ",
      "isCloneable": false,
      "isConfigurable": false,
      "isHidden": true,
      "isPriceImpacting": false,
      "isReadOnly": false,
      "isRequired": false,
      "label": "Bodily Injury Per Person Limit",
      "name": "Bodily Injury Per Person Limit",
      "status": "Active"
    }
  ],
  "code": "Auto Term",
  "name": "Auto Term"
}
],
"id": "01tSG0000008KnvBYAS",
"isActive": true,
"isAssetizable": true,
"isConfigurable": true,
"isSoldOnlyWithOtherProds": false,
"name": "Auto",
"nodeType": "bundleProduct",
"productClassification": {
  "id": "11BSG00000LARcM2AX"
},
"productCode": "auto"
}
},
"uiTreatments": [
  {
    "details": {
      "attributeId": "0tjxx0000000007AAA",
      "stiId": "0QLxx0000004CU0GAM",
      "attributePicklistValueId": "0v6xx0000000005AAA"
    }
  },

```

```

    "uiTreatmentTarget": "Attribute_Picklist_Value",
    "uiTreatmentType": "Hide"
  },
  {
    "details": {
      "stiId": "ref_f0f2da7b_c431_482d_bf4b_599052f3a2e1"
    },
    "uiTreatmentTarget": "Component",
    "uiTreatmentType": "Disable"
  }
],
"ratingResult": {
  "childNodes": [
    {
      "childNodes": [
        {
          "childNodes": [],
          "details": {
            "NetUnitPrice": 590.4,
            "InstanceKey": "bipd1",
            "ProductCode": "bipd",
            "NetTotalPrice": 590.4,
            "ProratedQLITaxAmount": 0
          }
        }
      ]
    },
    {
      "details": {
        "NetUnitPrice": 1485.84,
        "InstanceKey": "AutoSilver",
        "ProductCode": "autoSilver",
        "NetTotalPrice": 1485.84,
        "ProratedQLITaxAmount": 0
      }
    }
  ]
}
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
configMessages	Insurance Config Message []	List of configurator messages.	Small, 63.0	63.0
contextId	String	ID of the context.	Small, 61.0	61.0
contextJSON	Context JSON []	Serialized JSON representation of the context.	Small, 66.0	66.0
errors		List of errors that are encountered during the processing of the API request.	Small, 61.0	61.0
pricingResult	Map<String, Object>	Details of the pricing result.	Small, 61.0	61.0

Property Name	Type	Description	Filter Group and Version	Available Version
productDetails	Product[]	Details of the list of products that are included in the context JSON.	Small, 66.0	66.0
ratingResult	Rating Result[]	Rating results after executing pricing with the context.	Small, 66.0	66.0
uiTreatments	Insurance UI Treatments[]	Details of the insurance UI configuration.	Small, 66.0	66.0

Product Surcharge List

Output Representation for List of ProductSurcharge

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Map<String, ErrorDetailOutput>	List of errors encountered during the processing of the API request.	Big, 63.0	63.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
records	String[]	Product surcharge records that are created.	Big, 63.0	63.0

Product Surcharge Patch

Output representation of product surcharge update.

JSON example

This request shows a sample success response.

```
{
  "errors": []
  "isSuccess": true
  "productSurchargeId": "1Xrxx000000003F"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
errors	Error Detail Output[]	List of errors encountered during the processing of the API request.	Big, 63.0	63.0
isSuccess	Boolean	Indicates whether the API request is successful (<code>true</code>) or not (<code>false</code>).	Big, 63.0	63.0
productSurchargeId	String	ID of the product surcharge.	Big, 63.0	63.0

Product Rating Details Response

Output representation that contains the constructed rating request payload.

JSON example

```
{
  "salesTransactionTypeId": "0YWxx000000001GAAQ",
  "transactionType": "NewBusiness",
  "ratingInputs": [
    {
      "instanceKeys": [
        "autoSilver",
        "auto",
        "comprehensive"
      ],
      "productCode": "comprehensive",
      "attributes": {
        "NumberOfDays": "",
        "Deductible": "100",
        "BodilyInjuryPerPersonLimit": "",
        "PropertyDamagePerAccidentLimit": "",
        "Limit": "1000",
        "BodilyInjuryPerAccidentLimit": ""
      },
      "additionalFields": {}
    }
  ],
  "ratingOptions": {
    "executeConfigurationRules": true,
    "executePricing": true,
    "returnContextJSON": false,
    "returnProductDetails": false
  },
  "effectiveDate": "2025-01-01T00:00:00.000Z",
  "ratingDate": "2025-01-01T00:00:00.000Z",
  "additionalFields": {
    "customField1": "value1"
  },
  "rootProductCodes": [
    "AUTO_POLICY"
  ]
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
additionalFields	Map<String, Object>	Additional custom fields that are used to generate rating request payload.	Big, 66.0	66.0
effectiveDate	Date	Date that's used to find the active pricing procedure for pricing. The default value is the current date.	Big, 66.0	66.0
ratingDate	Date	Date that's used to calculate rating.	Big, 66.0	66.0

Property Name	Type	Description	Filter Group and Version	Available Version
ratingInputs	Rating Input Node Response[]	List of the rating input nodes in the hierarchical structure that contains the attribute values of each context node.	Big, 66.0	66.0
ratingOptions	Rating Options	Rating options that are used to control the rating behavior for the Insurance Product Rating API.	Big, 66.0	66.0
rootProductCodes	String[]	Root product codes that the products belong to for which the rating request payload is created.	Big, 66.0	66.0
salesTransactionTypeId	String	ID of the sales transaction type.	Big, 66.0	66.0
transactionType	String	Name of the transaction processing type.	Big, 66.0	66.0

Rating Input Node Response

Output representation of a single node in the rating input hierarchy.

JSON example

```
{
  "additionalFields": {},
  "attributes": {},
  "instanceKeys": [
    "Key1"
  ],
  "productCode": "ProductCode1"
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
additionalFields	Map<String, Object>	Map of additional fields that must be updated in the quote context.	Big, 66.0	66.0
attributes	Map<String, Object>	Map of product-specific attributes in key-value pairs that are used to determine pricing.	Big, 66.0	66.0
instanceKeys	String[]	List of the instance keys for the product hierarchy.	Big, 66.0	66.0
productCode	String	Unique code to identify a product.	Big, 66.0	66.0

Rating Options

Output representation of the rating options that are returned in the response.

JSON example

```
{
  "executeConfigurationRules": true,
  "executePricing": true,
  "returnContextJSON": true,
  "returnProductDetails": true
}
```

Property Name	Type	Description	Filter Group and Version	Available Version
executeConfigurationRules	Boolean	Indicates whether to execute the configuration rules (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0
executePricing	Boolean	Indicates whether to execute the pricing (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0
returnContextJSON	Boolean	Indicates whether to return the context JSON (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0
returnProductDetails	Boolean	Indicates whether to return the product details (<code>true</code>) or not (<code>false</code>).	Big, 66.0	66.0

Rating Result

Output representation for the details of the product rating pricing results.

JSON example

```
{
  "ratingResult": {
    "childNodes": [
      {
        "childNodes": [
          {
            "childNodes": [],
            "details": {
              "NetUnitPrice": 590.4,
              "InstanceKey": "bipd1",
              "ProductCode": "bipd",
              "NetTotalPrice": 590.4,
              "ProratedQLITaxAmount": 0
            }
          }
        ],
        "details": {
          "NetUnitPrice": 1485.84,
          "InstanceKey": "AutoSilver",
          "ProductCode": "autoSilver",
          "NetTotalPrice": 1485.84,
          "ProratedQLITaxAmount": 0
        }
      }
    ]
  }
}
```

```

    ]
  }
}

```

Property Name	Type	Description	Filter Group and Version	Available Version
childNodes	Rating Result[]	List of child nodes for which the rating results are calculated.	Small, 66.0	66.0
details	Map<String, Object>	Details of the rating result of the parent node.	Small, 66.0	66.0

Insurance Quoting Standard Invocable Actions

Learn more about the standard invocable actions available with Insurance Quoting.

[Add Eligible Insurance Clauses Action](#)

Add eligible insurance clauses to a quote or context. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID along with insurance product clause details.

[Create Insurance Quote Action](#)

Create an Insurance quote by using a context ID or a set of user inputs that represent quote details.

[Create Insurance Rating Action](#)

Create a runtime context that contains the details to create a quote.

[Generate Insurance Clauses Action](#)

Generate insurance clauses based on the specified quote and context information. You must specify either quote ID or context ID to generate insurance clauses.

[Get Eligible Insurance Clauses Action](#)

Get eligible insurance clauses based on quote ID and context ID. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID to get the eligible insurance clauses.

[Get Insurance Quote Details Action](#)

Get the details of an insurance quote.

[Get Insurance Rating Input Action](#)

Get the rating input data for the insurance products to support the pricing calculations and coverage analysis.

[Reprice Insurance Product Action](#)

Recalculate the price of insurance products based on user inputs.

[Update Insurance Quote Action](#)

Update an existing insurance quote by adding, updating, or deleting nodes and optionally executing pricing and rules.

Add Eligible Insurance Clauses Action

Add eligible insurance clauses to a quote or context. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID along with insurance product clause details.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/addEligibleInsuranceClauses

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
contextId	Type string Description Required if quoteId isn't specified. Context ID of the insurance quote.
instanceKey	Type string Description Required if quoteLineItemId isn't specified. Unique key that identifies the product to which a quote line item belongs.
insuranceProductClauses	Type string Description Required. JSON object with the details of the insurance product clauses to add.
optionFlags	Type Apex-defined Description An Apex runtime_industries_insurance.AddEligibleInsuranceClausesOptions record that contains the details to add eligible clauses to an insurance product.

Input	Details
quoteId	Type string Description Required if contextId isn't specified. ID of the insurance quote to add eligible clauses.
quoteLineItemId	Type string Description Required if instanceKey isn't specified. ID of the quote line items for adding eligible clauses.

Outputs

Output	Details
errorsList	Type string Description Errors that are encountered during the operation.
isSuccess	Type boolean Description Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).
quoteLineItemInsProductClauseIds	Type string Description IDs of the quote line item insurance product clauses.

Example

Sample Request

```
{
  "inputs": [
    {
      "quoteId": "0Q0xx0000000001AAA",
      "quoteLineItemId": "0QLxx0000000002BBB",
    }
  ]
}
```

```

    "contextId": "0j5xx0000000003CCC",
    "instanceKey": [
      "Auto",
      "Liability"
    ],
    "insuranceProductClauses": [
      {
        "insuranceProductClauseId": "0j4xx0000000004DDD",
        "effectiveDate": "2025-01-01",
        "expirationDate": "2025-12-31"
      }
    ],
    "optionFlags": {
      "saveContext": true
    }
  }
]
}

```

This is a sample request to call this invocable action from Apex code.

```

public class AddEligibleInsuranceClausesInvoke {
    private static final String ACTION_NAME = 'addEligibleInsuranceClauses';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Either quoteId or contextId is required
            setInvocationParameter(action, inputs, 'quoteId', false);
            setInvocationParameter(action, inputs, 'contextId', false);

            // Either quoteLineItemId or instanceKey is required
            setInvocationParameter(action, inputs, 'quoteLineItemId', false);
            setInvocationParameter(action, inputs, 'instanceKey', false);

            // insuranceProductClauses expects JSON
            setInvocationParameter(action, inputs, 'insuranceProductClauses', true);

            // Optional optionFlags map
            if (inputs != null && inputs.containsKey('optionFlags')) {
                Map<String, Object> addClauseOptions =
                    (Map<String, Object>) inputs.get('optionFlags');
                if (addClauseOptions != null) {
                    runtime_industries_insurance.AddEligibleInsuranceClausesOptions
apexOptions =
                        new
runtime_industries_insurance.AddEligibleInsuranceClausesOptions();

```



```

        if (addClauseOptions.containsKey('saveContext')) {
            apexOptions.SaveContext =
                (Boolean) addClauseOptions.get('saveContext');
        }
        action.setInvocationParameter('addClauseOptions', apexOptions);
    }
}

List<Invocable.Action.Result> results = action.invoke();
if (results != null && !results.isEmpty() && results[0].isSuccess()) {
    output.put('output', results[0].getOutputParameters());
    output.put('success', true);
    return true;
}

output.put('success', false);
if (results != null && !results.isEmpty()) {
    output.put('errors', results[0].getErrors());
}
return false;
} catch (Exception e) {
    output.put('success', false);
    output.put('errorMessage', e.getMessage());
    return false;
}
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String attribute,
    Boolean serializeJson
) {
    if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
!= null) {
        if (serializeJson) {
            action.setInvocationParameter(attribute,
JSON.serialize(payload.get(attribute)));
        } else {
            action.setInvocationParameter(attribute, payload.get(attribute));
        }
    }
}
}
}

```

Sample Response

```

[
  {
    "actionName": "addEligibleInsuranceClauses",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {

```

```
    "isSuccess": true,
    "quoteLineItemInsProductClauseIds": [
      "a1Cxx0000000001AAA"
    ],
    "errorsList": []
  },
  "sortOrder": -1,
  "version": 1
}
```

Create Insurance Quote Action

Create an Insurance quote by using a context ID or a set of user inputs that represent quote details.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/createInsuranceQuote

Formats

JSON

HTTP Methods

GET, POST

Authentication

Bearer Token

Inputs

If you specify quoteInputs (but not contextId) in the request body, you must provide quoteName, priceBookId, and effectiveDate as input parameters.

If you specify contextId (but not quoteInputs) in the request body, you must provide these:

- either quoteName or SalesTransactionName within additionalFields, and
- either priceBookId or PriceBook within additionalFields

Input	Details
accountId	Type String Description ID of the account that's associated with the quote being created.
additionalFields	Type String Description Optional. Map of additional fields that must be updated in the quote context.

Input	Details
contextId	Type String Description Required if quoteInputs isn't specified. Context ID that's used to issue the quote.
effectiveDate	Type Date Description Date that the quote becomes effective.
expirationDate	Type Date Description Optional. Expiration date of the quote.
opportunityId	Type String Description Optional. ID of the opportunity that's associated with the quote.
pricebookId	Type String Description Required if either quoteInputs or contextId is specified. ID of the pricebook that's associated with the quote.
pricingProcedureApiName	Type String Description Optional. API name of the pricing procedure used to price the product.
quoteInputs	Type String Description Required if contextId isn't specified. Set of user inputs on products that represent the quote details.
quoteOptions	Type Apex-defined

Input	Details
	Description An Apex runtime_industries_insurance.CreateInsuranceQuoteOptions record that contains details to create the quote.
quoteName	Type String Description Name of the quote that must be created.
ratingDate	Type Date Description Optional. Date to find the active pricing procedure.

Outputs

Output	Details
contextId	Type String Description Context ID of the insurance quote that's created.
quoteId	Type String Description ID of the insurance quote that's created.

Example

Sample request where `quoteInputs` is specified in the request body.

```
{
  "inputs": [
    {
      "quoteName": "Auto Insurance Quote",
      "priceBookId" : "01sxx0000005pyfAAA",
      "accountId": "001xx000003GYwrAAG",
      "effectiveDate": "2024-10-11",
      "opportunityId": "006xx000001a2tbAAA",
      "quoteOptions": {
        "productSellingModel": "One Time",
        "saveQuote": true
      }
    }
  ]
}
```

```
    },  
    "quoteInputs":  
    "[REDACTED]"  
  }  
]  
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for createInsuranceQuote invocable action
public class CreateInsuranceQuoteInvoke {
    private static final String ACTION_NAME = 'createInsuranceQuote';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Required inputs (serialize where needed)
            setInvocationParameter(action, inputs, 'quoteInputs', true);
            setInvocationParameter(action, inputs, 'quoteName', false);
            setInvocationParameter(action, inputs, 'priceBookId', false);
            setInvocationParameter(action, inputs, 'accountId', false);
            setInvocationParameter(action, inputs, 'effectiveDate', false);

            // Optional inputs
            setInvocationParameter(action, inputs, 'contextId', false);
            setInvocationParameter(action, inputs, 'expirationDate', false);
            setInvocationParameter(action, inputs, 'additionalFields', true);
            setInvocationParameter(action, inputs, 'opportunityId', false);

            // quoteOptions is optional but supported
            if (inputs != null && inputs.containsKey('quoteOptions')) {
                Map<String, Object> optionsFromInput =
                    (Map<String, Object>) inputs.get('quoteOptions');
                if (optionsFromInput != null) {
                    runtime_industries_insurance.CreateInsuranceQuoteOptions quoteOptions
=
                        new runtime_industries_insurance.CreateInsuranceQuoteOptions();
                    if (optionsFromInput.containsKey('executePricing')) {
                        quoteOptions.executePricing =
                            (Boolean) optionsFromInput.get('executePricing');
                    }
                    if (optionsFromInput.containsKey('validateProductCatalog')) {
                        quoteOptions.validateProductCatalog =
                            (Boolean) optionsFromInput.get('validateProductCatalog');
                    }
                }
            }
        }
    }
}
```

```

        if (optionsFromInput.containsKey('executeConfigurationRules')) {
            quoteOptions.executeConfigurationRules =
                (Boolean) optionsFromInput.get('executeConfigurationRules');
        }
        if (optionsFromInput.containsKey('executeQualificationRules')) {
            quoteOptions.executeQualificationRules =
                (Boolean) optionsFromInput.get('executeQualificationRules');
        }
        if (optionsFromInput.containsKey('saveQuote')) {
            quoteOptions.saveQuote =
                (Boolean) optionsFromInput.get('saveQuote');
        }
        if (optionsFromInput.containsKey('productSellingModel')) {
            quoteOptions.productSellingModel =
                (String) optionsFromInput.get('productSellingModel');
        }
        action.setInvocationParameter('quoteOptions', quoteOptions);
    }
}

List<Invocable.Action.Result> results = action.invoke();
if (results != null && !results.isEmpty() && results[0].isSuccess()) {
    output.put('output', results[0].getOutputParameters());
    output.put('success', true);
    return true;
}

output.put('success', false);
if (results != null && !results.isEmpty()) {
    output.put('errors', results[0].getErrors());
}
return false;
} catch (Exception e) {
    output.put('success', false);
    output.put('errorMessage', e.getMessage());
    return false;
}
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String attribute,
    Boolean serializeJson
) {
    if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
!= null) {
        if (serializeJson) {
            action.setInvocationParameter(attribute,
JSON.serialize(payload.get(attribute)));
        } else {
            action.setInvocationParameter(attribute, payload.get(attribute));
        }
    }
}

```

```
    }
}
```

Sample Response

```
[
  {
    "actionName": "createInsuranceQuote",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "contextId": "00000000i18tq18g00291753491065771466a599fe8a2440890b4f3d8b4ef40b5",
      "quoteId": "0Q0SG0000000O0UD0A4"
    },
    "sortOrder": -1,
    "version": 1
  }
]
```

Create Insurance Rating Action

Create a runtime context that contains the details to create a quote.
This action is available in API version 66.0 and later.

Supported REST HTTP Methods

- URI**
/services/data/v/actions/standard/createInsuranceRating
- Formats**
JSON
- HTTP Methods**
POST
- Authentication**
Bearer Token

Inputs

Input	Details
additionalFields	TypeString DescriptionAdditional fields to include in the rating context.
effectiveDate	TypeDate

Input	Details
	Description Required. Date from which the insurance policy coverage becomes effective.
executeRating	Type String Description Indicates whether rating must be executed (<code>true</code>) or not (<code>false</code>). The default value is <code>true</code> .
ratingDate	Type Date Description Date that's used for rating calculations and rule evaluations. If not specified, the current date is used.
ratingInputs	Type Map<String, Object> Description Required. JSON string that contains the rating input nodes with product codes and attributes.
ratingOptions	Type Apex-defined Description An Apex runtime_industries_insurance.CreateInsuranceRatingOptions record that contains rating execution options to create insurance rating.
rootProductCodes	Type String Description Array of root product codes to include in the rating process.
salesTransactionTypeId	Type String Description ID of the sales transaction type.
transactionType	Type String Description Type of insurance transaction, such as NewBusiness, Renewal, and Endorsement.

Outputs

Output	Details
contextId	Type String Description Context ID of the insurance quote.
pricingResults	Type Map<String, Object> Description Recalculated pricing details of the insurance product.
productRatingOutput	Type Apex-defined Description An Apex ConnectApi__ProductRatingOutputRepresentation record that contains the product rating details.

Example

Sample Request

```
{
  "inputs": [
    {
      "effectiveDate": "2024-10-18",
      "ratingInputs":
      "[{"insurance":["Auto","Auto","No Drive"],"product":["at Drive"],"attributes":{"DriveAccidents":4,"tagCode":["03x0004W7X"]}",
      "ratingOptions": {
        "returnRatingResults": true,
        "returnContextJson": true,
        "returnProductDetails": true
      }
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for createInsuranceRating invocable action
public class CreateInsuranceRatingInvoke {
    private static final String ACTION_NAME = 'createInsuranceRating';

    public static Boolean invokeMethod(
        Map<String, Object> inputs,
```

```

        Map<String, Object> output,
        Map<String, Object> options
    ) {
        Map<String, Object> postPayload = (Map<String, Object>)
getValueFromMap('postPayload', inputs, options);
        System.debug('Rating post Input: ' + postPayload);

        if (postPayload == null) {
            output.put('success', false);
            output.put('errorMessage', 'postPayload is required for rating patch');
            output.put('errorType', 'VALIDATION_ERROR');
            return false;
        }

        try {
            System.debug('Calling Invocable Action: createInsuranceRating');

            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            System.debug('RatingInputs: ' + postPayload.get('ratingInputs'));
            setInvocationParameter(action, postPayload, 'ratingInputs', true);

            action.setInvocationParameter('transactionType',
postPayload.get('transactionType'));

            setInvocationParameter(action, postPayload, 'additionalFields', true);

            action.setInvocationParameter('effectiveDate', postPayload.get('effectiveDate'));

            if (postPayload.containsKey('ratingOptions')) {
                System.debug('Options:' + postPayload.get('ratingOptions'));
                Map<String, Object> ratingOptions = (Map<String, Object>)
postPayload.get('ratingOptions');
                action.setInvocationParameter('ratingOptions', ratingOptions);
            }

            List<Invocable.Action.Result> results = action.invoke();

            if (results != null && !results.isEmpty()) {
                Invocable.Action.Result result = results[0];

                if (result.isSuccess()) {
                    System.debug('Rating post successfully via Invocable Action');
                    output.put('ratingData', result.getOutputParameters());

                    List<Invocable.Action.Error> actionErrors = result.getErrors();
                    if (actionErrors != null && !actionErrors.isEmpty()) {
                        output.put('success', false);
                        output.put('error', result.getErrors());
                        return false;
                    }
                    output.put('success', true);
                }
            }
        }
    }
}

```

```

        return true;

    } else {
        List<Invocable.Action.Error> actionErrors = result.getErrors();
        String errorMessage = 'Rating post Invocable Action failed';

        if (actionErrors != null && !actionErrors.isEmpty()) {
            List<String> errorMessages = new List<String>();
            List<Map<String, Object>> errorDetailsList = new List<Map<String,
Object>>();

            for (Invocable.Action.Error actionError : actionErrors) {
                errorMessages.add(actionError.getMessage());
                errorDetailsList.add(new Map<String, Object>{
                    'message' => actionError.getMessage(),
                    'code' => actionError.getCode()
                });
            }

            errorMessage = String.join(errorMessages, ' ');
            output.put('actionErrors', errorDetailsList);
        }

        output.put('success', false);
        output.put('errorMessage', errorMessage);

        System.debug(LoggingLevel.ERROR, 'Rating post Invocable Action failed:
' + errorMessage);
        return false;
    }

    } else {
        output.put('success', false);
        output.put('errorMessage', 'Rating post Invocable Action returned null or
empty results');
        System.debug(LoggingLevel.WARN, 'Rating post Invocable Action returned
null/empty results');
        return false;
    }

    } catch (Exception e) {
        String errorMsg = 'Exception: ' + e.getMessage();
        output.put('success', false);
        output.put('errorMessage', errorMsg);
        output.put('errorType', 'EXCEPTION');
        System.debug(LoggingLevel.ERROR, 'Exception while rating post: ' + errorMsg);

        return false;
    }
}

private static Object getValueFromMap(String key, Map<String, Object> inputs, Map<String,
Object> options) {
    if (inputs != null && inputs.get(key) != null &&

```

```

String.isNotBlank(String.valueOf(inputs.get(key)))) {
    return inputs.get(key);
}
else if (options != null && options.get(key) != null &&
String.isNotBlank(String.valueOf(options.get(key)))) {
    return options.get(key);
}

return null;
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String inputAttribute,
    Boolean serializeJSON
) {
    if (payload.containsKey(inputAttribute) && payload.get(inputAttribute) != null) {

        System.debug(inputAttribute + ': ' +
JSON.serialize(payload.get(inputAttribute)));
        if (serializeJSON) {
            action.setInvocationParameter(inputAttribute,
JSON.serialize(payload.get(inputAttribute)));
        } else {
            action.setInvocationParameter(inputAttribute, payload.get(inputAttribute));
        }
    } else {
        System.debug('payload does not contain attribute: ' + inputAttribute);
    }
}
}

```

Sample Response

```

[
  {
    "actionName": "repriceInsuranceProduct",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "contextId": "273e01fdfb6f9f21428869eb784ee756976afb0e9bdc49f2a360be397a10ca12",
      "productRatingOutput": {
        "ratingResult": {
          "childNodes": [
            {
              "childNodes": [
                {
                  "childNodes": [
                    {
                      "childNodes": [],
                      "details": {

```

```

        "NetUnitPrice": null,
        "InstanceKey": "Driver",
        "ProductCode": "driver",
        "NetTotalPrice": 0,
        "Driver_Accident_Points": 15,
        "ProratedQLITaxAmount": 0,
        "Driver_MVR_Points": 8,
        "Age_First_Licensed": 16
    }
},
{
    "childNodes": [],
    "details": {
        "NetUnitPrice": 501.84,
        "InstanceKey": "Comprehensive",
        "ProductCode": "comprehensive",
        "NetTotalPrice": 501.84,
        "ProratedQLITaxAmount": 0
    }
}
],
"details": {
    "NetUnitPrice": 501.84,
    "InstanceKey": "Auto1",
    "ProductCode": "auto",
    "NetTotalPrice": 501.84,
    "ProratedQLITaxAmount": 0
}
},
"details": {
    "NetUnitPrice": 1485.84,
    "InstanceKey": "AutoSilver",
    "ProductCode": "autoSilver",
    "NetTotalPrice": 1485.84,
    "ProratedQLITaxAmount": 0
}
},
"details": {}
},
"productDetails": [
{
    "productSpecificationType": {
        "productSpecificationRecordType": {},
        "name": "InsuredParty"
    },
    "childProducts": [],
    "additionalFields": {
        "ProductCode": "driver",
        "SpecificationType": "InsuredParty",
        "BasedOnId": "11Bxx000002C1kZEAS",
        "Name": "Driver"
    }
},

```

```

"availabilityDate": null,
"description": null,
"productInformation": null,
"childVariationIds": [],
"isActive": true,
"qualificationContext": null,
"productClassification": {
  "id": "11Bxx000002C1kZEAS"
},
"productComponentGroups": [],
"configureDuringSale": "Allowed",
"productRelatedComponent": null,
"id": "01txx0000006i3iAAA",
"prices": [],
"isComponentRequired": null,
"productType": null,
"isAssetizable": true,
"isSoldOnlyWithOtherProds": false,
"attributeCategories": [
  {
    "code": "Driving History",
    "records": [
      {
        "isRequired": false,
        "code": "AgeFirstLicensed",
        "additionalFields": {},
        "attributeCategoryId": "0v3xx0000000001AAA",
        "hidden": false,
        "isConfigurable": false,
        "dataType": "NUMBER",
        "label": "Age First Licensed",
        "isReadOnly": false,
        "isCloneable": false,
        "name": "Age First Licensed",
        "id": "0tjxx0000000001FAQ",
        "attributeNameOverride": "Age First Licensed",
        "isPriceImpacting": true,
        "developerName": "Age_First_Licensed",
        "status": "Active"
      }
    ],
    "name": "Driving History"
  }
],
"nodeType": "simpleProduct",
"displayUrl": null,
"productSellingModelOptions": [
  {
    "productSellingModel": {
      "sellingModelType": "OneTime",
      "name": "One Time",
      "id": "0jPxx0000000001EAA",
      "doesAutoRenewByDefault": false,
      "status": "Active"
    }
  }
]

```

```

    },
    "productSellingModelId": "0jPxx0000000001EAA",
    "isDefault": true,
    "productId": "01txx00000006i3iAAA",
    "id": "0iOxx000000000zEAA"
  }
],
"productQuantity": null,
"productCode": "driver",
"name": "Driver"
}
],
"contextJSON": {
  "salesTransactions": [
    {
      "salesTransactionItems": [
        {
          "salesTransactionItemClauses": [],
          "childNodes": [
            {
              "salesTransactionItemClauses": [],
              "childNodes": [],
              "salesTransactionItemAttributes": [
                {
                  "fields": {
                    "AttributeKey": "0tjxx000000000PAAQ",
                    "AttributeValue": "50",
                    "ParentReference": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",

                    "hasObjectRef": null,
                    "AttributePicklistValue": "0v6xx000000000ZAAQ",
                    "IsPriceImpacting": false,
                    "businessObjectType": "QuoteLineItemAttribute",
                    "AttributeName": "Deductible",
                    "AttributeDefinitionCode": "Deductible",
                    "id": "ref_fdd78e30_1258_4ea3_9f56_c74245ae58d6",
                    "SalesTransactionItemAttrParent":
"ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
                    "AttributeDeveloperName": "Deductible"
                  }
                }
              ]
            }
          ]
        }
      ]
    }
  ],
  "salesTransactionItemRelatedObjects": [],
  "salesTransactionItemTaxes": [],
  "fields": {
    "LevelTwoBasePremium": 120,
    "IntermediateVariableTwo": null,
    "Product": "01txx00000006i3UAAQ",
    "businessObjectType": "QuoteLineItem",
    "DYN__License_State": null,
    "RevenueCloudPackagingFlag": "C",
    "ValidationResult": null,

```

```

        "DYN__AssetRecordName": null,
        "PolicyPremium": 1485.84,
        "PeriodBoundaryStartMonth": null,
        "SalesTransactionSourceAsset": null,
        "ProratedQLITaxAmount": 0,
        "id": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
        "TaxTreatment": null,
        "Subtotal": null,
        "LineItemPath":
"ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc/ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
        "DiscountAmount": null,
        "PricebookEntry": null,
        "NetUnitPrice": 590.4,
        "ParentSalesTransactionItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",

        "ObligatedAmount": null,
        "DYN__Age_First_Licensed": null,
        "LevelThreePremium": null,
        "ProductName": "Bodily Injury & Property Damage",
        "ProformaBillingPeriodAmount": null,
        "CommissionAmount": null,
        "ProductSpec": "Coverage",
        "DYN__Last_Name": null,
        "PricingTermCount": null,
        "DYN__ContactRecordName": null,
        "RoundedLineAmount": null,
        "SalesTransactionAction": null,
        "TransactionType": null,
        "PricingTermUnit": null,
        "BasisTransactionItem": null,
        "PricingSource": null,
        "StockKeepingUnit": null,
        "DYN__Year": null,
        "PartnerUnitPrice": null,
        "ItemTotalAdjustmentAmount": null,
        "ArePartialPeriodsAllowed": false,
        "DYN__Bodily_Injury_Per_Person_Limit": 1000,
        "DYN__Colour": null,
        "FactorP": null,
        "FactorO": null,
        "CustomProductName": null,
        "SalesTransactionItemParent": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",

        "Quantity": 1,
        "ContractId": null,
        "ItemGroupSummarySubtotal": null,
        "DYN__AccountRecordName": null,
        "DYN__E_Mail": null,
        "DYN__First_Name": null,
        "DYN__Bodily_Injury_Per_Accident_Limit": 1000,
        "DYN__Driver_Accident_Points": null,
        "SalesItemType": null,
        "ItemPath": "01txx0000006i3DAAQ/01txx0000006i3UAAQ"
    },

```



```

    "salesTransactionItemRelationships": [
      {
        "fields": {
          "ParentReference": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
          "AssociatedItem": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
          "MainItemRole": null,
          "ProductRelatedComponent": "0dSxx000000000XEAQ",
          "businessObjectType": "QuoteLineRelationship",
          "AssociatedItemPricing": "IncludedInBundlePrice",
          "ProductRelationshipType": "0yoxx0000000001AAA",
          "RootItemProductCode": "autoSilver",
          "IsPriceInclusive": true,
          "RootItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
          "RootItemProductSellingModel": null,
          "AssociatedQuantScaleMethod": null,
          "MainItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
          "AssociatedItemRole": null,
          "SalesTrnItemRelationshipParent": null,
          "id": "ref_80623cb4_3abd_4ee0_876a_c5ca9ecc07fc",
          "MainItemProduct": "01txx0000006i3DAAQ",
          "MainItemProductSellingModel": null,
          "RootItemProduct": "01txx0000006i3DAAQ"
        }
      }
    ],
    "salesTransactionItemAttributes": [],
    "salesTransactionItemRelatedObjects": [],
    "salesTransactionItemTaxes": [],
    "fields": {
      "LevelTwoBasePremium": null,
      "IntermediateVariableTwo": null,
      "Product": "01txx0000006i3DAAQ",
      "businessObjectType": "QuoteLineItem",
      "DYN__License_State": null,
      "RevenueCloudPackagingFlag": "C",
      "ValidationResult": null,
      "DYN__AssetRecordName": null,
      "PolicyPremium": 1485.84,
      "PeriodBoundaryStartMonth": null,
      "SalesTransactionSourceAsset": null,
      "ProratedQLITaxAmount": 0,
      "id": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
      "DYN__Age": null,
      "DYN__Property_Damage_Per_Accident_Limit": null,
      "DYN__Deductible": null,
      "STICurrencyIsoCode__std": null,
      "IntermediateVariableOne": null,
      "BillingFrequency": null,
      "DYN__State": null,
      "DYN__DOB": null,
      "ProductCode": "autoSilver",
      "DerivedPricingAttribute": false,

```

```

        "ItemPath": "01txx0000006i3DAAQ"
      },
      "salesTransactionItemRelationships": []
    }
  ],
  "fields": {
    "Account": null,
    "BillingCity": null,
    "businessObjectType": "Quote",
    "SalesTransactionName": null,
    "StatusCode": null,
    "ProratedAmount": 1485.84,
    "StartDate": "2025-11-04",
    "CurrencyIsoCode__std": null,
    "BillingPostalCode": null,
    "id": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",
    "BillingState": null,
    "CensusRatingType__std": null,
    "TransactionType": "AutoTransactionType",
    "UserProfile": "System Administrator",
    "QuoteAccount": null,
    "TotalProratedAmount": 1485.84,
    "CensusType__std": null,
    "GroupCensus": null,
    "SalesTransactionType": null,
    "Pricebook": null,
    "Opportunity": null,
    "ShippingCountry": null,
    "ShippingCity": null,
    "StandardAmount": 1485.84,
    "ProratedTaxAmount": 0,
    "RatingDate": null,
    "SalesTransactionSource": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",
    "EffectiveDate": "2025-11-04T00:00:00.000Z",
    "TotalStandardAmount": 1485.84
  },
  "appUsageAssignments": []
}
},
"sortOrder": -1,
"version": 1
}
]

```

Generate Insurance Clauses Action

Generate insurance clauses based on the specified quote and context information. You must specify either quote ID or context ID to generate insurance clauses.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/generateInsuranceClauses

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
contextId	Type string Description Required if quoteId isn't specified. Context ID of the insurance quote.
instanceKeys	Type string Description Required if quoteLineItemId isn't specified. Unique keys that identify the products to which quote line items belong.
optionFlags	Type Apex-defined Description An Apex runtime_industries_insurance.GenerateInsuranceClausesOptions record that contains the configuration and execution options to generate the insurance clauses.
quoteId	Type string Description Required if contextId isn't specified. ID of the insurance quote for adding eligible clauses.
quoteLineItemIds	Type string

Input	Details
	Description Required if quoteLineItemId isn't specified. IDs of the quote line item for the quote.

Outputs

Output	Details
asyncBulkRequestId	Type string Description ID of the asynchronous bulk request.
errorsList	Type string Description Errors that are encountered during the operation.
isSuccess	Type boolean Description Indicates whether the operation is successful (true) or not (false).

Example

Sample Request

```
{
  "inputs": [
    {
      "quoteId": "0Q0xx0000000001AAA",
      "contextId": "0000000i18tq18g0025175944829641764efd423f7dc44b98dd1e829086efc59",
      "quoteLineItemIds": [
        "0QLxx0000000003CCC"
      ],
      "instanceKeys": [
        {
          "instanceKey": [
            "Auto",
            "Liability"
          ]
        }
      ]
    }
  ],
}
```

```

    "optionFlags": {
      "generateClausesAsync": true,
      "generateManualClauses": false,
      "saveContext": true
    }
  }
]
}

```

This is a sample request to call this invocable action from Apex code.

```

// Sample Apex call for generateInsuranceClauses invocable action
public class GenerateInsuranceClausesInvoke {
    private static final String ACTION_NAME = 'generateInsuranceClauses';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Either quoteId or contextId is required
            setInvocationParameter(action, inputs, 'quoteId', false);
            setInvocationParameter(action, inputs, 'contextId', false);

            // Optional inputs
            setInvocationParameter(action, inputs, 'quoteLineItemIds', false);
            setInvocationParameter(action, inputs, 'instanceKeys', false);

            // Optional optionFlags map
            if (inputs != null && inputs.containsKey('optionFlags')) {
                Map<String, Object> generateClauseOptions =
                    (Map<String, Object>) inputs.get('optionFlags');
                if (generateClauseOptions != null) {
                    runtime_industries_insurance.GenerateInsuranceClausesOptions apexOptions
=
                    new runtime_industries_insurance.GenerateInsuranceClausesOptions();

                    if (generateClauseOptions.containsKey('saveContext')) {
                        apexOptions.SaveContext =
                            (Boolean) generateClauseOptions.get('saveContext');
                    }
                    if (generateClauseOptions.containsKey('generateManualClauses')) {
                        apexOptions.GenerateManualClauses =
                            (Boolean) generateClauseOptions.get('generateManualClauses');
                    }
                    if (generateClauseOptions.containsKey('generateClausesAsync')) {
                        apexOptions.generateClausesAsync =
                            (Boolean) generateClauseOptions.get('generateClausesAsync');
                    }
                }
            }
        }
    }
}

```

```

        action.setInvocationParameter('optionFlags', apexOptions);
    }
}

List<Invocable.Action.Result> results = action.invoke();
if (results != null && !results.isEmpty() && results[0].isSuccess()) {
    output.put('output', results[0].getOutputParameters());
    output.put('success', true);
    return true;
}

output.put('success', false);
if (results != null && !results.isEmpty()) {
    output.put('errors', results[0].getErrors());
}
return false;
} catch (Exception e) {
    output.put('success', false);
    output.put('errorMessage', e.getMessage());
    return false;
}
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String attribute,
    Boolean serializeJson
) {
    if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
!= null) {
        if (serializeJson) {
            action.setInvocationParameter(attribute,
JSON.serialize(payload.get(attribute)));
        } else {
            action.setInvocationParameter(attribute, payload.get(attribute));
        }
    }
}
}

```

Sample Response

```

[
  {
    "actionName": "generateInsuranceClauses",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "isSuccess": true,
      "asyncBulkRequestId": "750xx0000000001AAA",
      "errorsList": []
    },
  },
]

```

```
    "sortOrder": -1,
    "version": 1
  }
]
```

Get Eligible Insurance Clauses Action

Get eligible insurance clauses based on quote ID and context ID. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID to get the eligible insurance clauses.

This action is available in API version 65.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getEligibleInsuranceClauses

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer *token*

Inputs

Input	Details
contextId	Type string Description Required if quoteId isn't specified. Context ID of the insurance quote.
instanceKey	Type string Description Required if quoteLineItemId isn't specified. Unique key that identifies the product to which a quote line item belongs.
quoteId	Type string Description Required if contextId isn't specified. ID of the insurance quote to get eligible clauses.

Input	Details
quoteLineItemId	Type string Description Required if <code>instanceKey</code> isn't specified. ID of the quote line items to get eligible clauses.

Outputs

Output	Details
eligibleClauses	Type string Description Details of insurance product clauses.
errorsList	Type string Description Errors that are encountered during the operation.
isSuccess	Type boolean Description Indicates whether the operation is successful (<code>true</code>) or not (<code>false</code>).

Example

Sample Request

```
{
  "inputs": [
    {
      "contextId": "0000000i18tq18g0025175944829641764efd423f7dc44b98dd1e829086efc59",
      "instanceKey": [
        "a3Nxx000000CmikEAC"
      ],
      "quoteId": "0Q0xx000000Dm3jCAC",
      "quoteLineItemId": "0QLxx000000E5d1OAC"
    }
  ]
}
```


This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for getEligibleInsuranceClauses invocable action
public class GetEligibleInsuranceClausesInvoke {
    private static final String ACTION_NAME = 'getEligibleInsuranceClauses';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Either quoteId or contextId is required
            setInvocationParameter(action, inputs, 'quoteId', false);
            setInvocationParameter(action, inputs, 'contextId', false);

            // Either quoteLineItemId or instanceKey is required
            setInvocationParameter(action, inputs, 'quoteLineItemId', false);
            setInvocationParameter(action, inputs, 'instanceKey', false);

            List<Invocable.Action.Result> results = action.invoke();
            if (results != null && !results.isEmpty() && results[0].isSuccess()) {
                output.put('output', results[0].getOutputParameters());
                output.put('success', true);
                return true;
            }

            output.put('success', false);
            if (results != null && !results.isEmpty()) {
                output.put('errors', results[0].getErrors());
            }
            return false;
        } catch (Exception e) {
            output.put('success', false);
            output.put('errorMessage', e.getMessage());
            return false;
        }
    }

    private static void setInvocationParameter(
        Invocable.Action action,
        Map<String, Object> payload,
        String attribute,
        Boolean serializeJson
    ) {
        if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
!= null) {
            if (serializeJson) {
                action.setInvocationParameter(attribute,
JSON.serialize(payload.get(attribute)));
            } else {

```

```

        action.setInvocationParameter(attribute, payload.get(attribute));
    }
}
}
}

```

Sample Response

```

[
  {
    "actionName": "getEligibleInsuranceClauses",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "getEligibleClausesOutputRep": {
        "errors": [],
        "insuranceProductClauses": [
          {
            "category": "Manual",
            "clauseName": "mxtestIC",
            "description": "two variable",
            "effectiveDate": "2025-10-01",
            "expirationDate": "2025-12-31",
            "insuranceProductClauseId": "...",
            "type": "Clause"
          }
        ],
        "success": true
      }
    },
    "sortOrder": -1,
    "version": 1
  }
]

```

Get Insurance Quote Details Action

Get the details of an insurance quote.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getInsuranceQuoteDetails

Formats

JSON

HTTP Methods

GET, POST

Authentication
Bearer Token

Inputs

Input	Details
quoteId	Type String Description Required. ID of the insurance quote to get the details for.
filters	Type String Description Optional. The filters used to filter the insurance quote or line items associated with the insurance quote.

Outputs

Output	Details
contextId	Type String Description Context ID of the insurance quote that's created.
contextJson	Type Map<String, Object> Description List of insurance context details.

Example

Sample Request

```
{
  "inputs":
  [
    {
      "quoteId": "0Q0xx0000004CXECA2"
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for getInsuranceQuoteDetails invocable action
public class GetInsuranceQuoteDetailsInvoke {
    private static final String ACTION_NAME = 'getInsuranceQuoteDetails';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Optional inputs
            setInvocationParameter(action, inputs, 'quoteId', false);
            setInvocationParameter(action, inputs, 'filters', false);

            List<Invocable.Action.Result> results = action.invoke();
            if (results != null && !results.isEmpty() && results[0].isSuccess()) {
                output.put('output', results[0].getOutputParameters());
                output.put('success', true);
                return true;
            }

            output.put('success', false);
            if (results != null && !results.isEmpty()) {
                output.put('errors', results[0].getErrors());
            }
            return false;
        } catch (Exception e) {
            output.put('success', false);
            output.put('errorMessage', e.getMessage());
            return false;
        }
    }

    private static void setInvocationParameter(
        Invocable.Action action,
        Map<String, Object> payload,
        String attribute,
        Boolean serializeJson
    ) {
        if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
!= null) {
            if (serializeJson) {
                action.setInvocationParameter(attribute,
JSON.serialize(payload.get(attribute)));
            } else {
                action.setInvocationParameter(attribute, payload.get(attribute));
            }
        }
    }
}
```

```

    }
}

```

Sample Response

```

[
  {
    "actionName": "getInsuranceQuoteDetails",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "contextId": "0000000i18tq18g00291753491555889e8ef669a59f0453fade5acbe4595a3d6",
      "contextJson": {
        "SalesTransaction": [
          {
            "Status": "Draft",
            "Account": null,
            "BillingCity": null,
            "Subtotal": 1350,
            "businessObjectType": "Quote",
            "SalesTransactionName": "Multi Auto",
            "TotalAmount": 1350,
            "ShippingStreet": null,
            "SalesTransactionItem": [
              {
                "ProrationPolicy": "0muxx000000004sAAA",
                "ProductName": "Medical Payments",
                "ProductCode": "STA-3",
                "ProductSellingModel": "0jPxx000000004sEAA",
                "SpecificationType": "Coverage",
                "TaxTreatment": "1ttxx000000004rAAA",
                "SalesTransactionItemParent": "0Q0xx0000004CE1CAM",
                "Quantity": 1,
                "Product": "01txx00000006i7KAAQ",
                "businessObjectType": "QuoteLineItem",
                "PricebookEntry": "0luxx00000008ybrAAA",
                "PricingTermCount": 1,
                "StartDate": "2023-10-19T00:00:00Z",
                "NetUnitPrice": 750,
                "UnitPrice": 750,
                "id": "0QLxx0000004CIpGAM"
              },
              {
                "ProrationPolicy": null,
                "ProductName": "Collision",
                "ProductCode": "STA-2",
                "ProductSellingModel": "0jPxx000000004tEAA",
                "SpecificationType": "Coverage",
                "TaxTreatment": "1ttxx000000004rAAA",
                "SalesTransactionItemParent": "0Q0xx0000004CE1CAM",
                "Quantity": 1,
                "Product": "01txx00000006i7JAAQ",
                "businessObjectType": "QuoteLineItem",

```

```

        "PricebookEntry": "01uxx0000008ybsAAA",
        "PricingTermCount": 1,
        "StartDate": null,
        "NetUnitPrice": 500,
        "UnitPrice": 500,
        "id": "0QLxx0000004CIqGAM"
    }
],
"BillingCountry": null,
"BillingStreet": null,
"Pricebook": "01sxx0000005pyfAAA",
"Opportunity": null,
"ShippingPostalCode": null,
"SalesTransactionSource": "0Q0xx0000004CE1CAM",
"ShippingCountry": null,
"ShippingCity": null,
"ShippingState": null,
"BillingPostalCode": null,
"id": "0Q0xx0000004CE1CAM",
"Contract": null,
"BillingState": null
}
]
}
},
"sortOrder": -1,
"version": 1
}
]

```

Get Insurance Rating Input Action

Get the rating input data for the insurance products to support the pricing calculations and coverage analysis.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/getInsuranceRatingInput

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
additionalFields	Type string Description JSON string of additional fields to include in the rating context.
effectiveDate	Type date Description Effective date for the rating calculation.
productIds	Type string Description Required. IDs of the products to generate the rating input for.
ratingDate	Type date Description Date to use for rating calculations.
rootProductCodes	Type string Description List of root product codes.
salesTransactionTypeId	Type string Description ID of the sales transaction type.
transactionType	Type string Description Name of the transaction type.

Outputs

Output	Details
getRatingInputOutputRep	Type Apex-defined Description An Apex ConnectApi__GetRatingInputOutputRep record that contains the details of the rating input for insurance products.

Example

POST

This sample request is for the Get Insurance Rating Input action.

```
{
  "inputs": [
    {
      "additionalFields": {
        "QuoteAccount": "001ZM000002ZNnBYAW"
      },
      "effectiveDate": "2024-01-01",
      "productIds": [
        "01tRM0000000001"
      ],
      "ratingDate": "2024-01-01",
      "rootProductCodes": [
        "ProductCode1"
      ],
      "salesTransactionTypeId": "0k5RM0000000001",
      "transactionType": "NewBusiness"
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample implementation to call insurance get rating input invocable action in Apex
public class InsuranceGetRatingInputApexService implements Callable {

    private static final String GET_RATING_INPUT_METHOD = 'getRatingInput';

    /**
     * @description Implementation of Callable interface for OmniStudio
     * @param action Method name to invoke
     * @param args Map containing input, output, and options
     * @return Object result of the method invocation
     */
    public Object call(String action, Map<String, Object> args) {
        System.debug('InsuranceGetRatingInputApexService: call ' + action);
    }
}
```



```

    Map<String, Object> input = (Map<String, Object>) args.get('input');
    Map<String, Object> output = (Map<String, Object>) args.get('output');
    Map<String, Object> options = (Map<String, Object>) args.get('options');

    return invokeMethod(action, input, output, options);
}
// Invoke methods, and if the method is not supported, it will be ignored.
public Boolean invokeMethod(
    String methodName,
    Map<String, Object> inputs,
    Map<String, Object> output,
    Map<String, Object> options
) {
    Boolean response = false;

    System.debug('Insurance Get Rating Input Apex Service - Method: ' + methodName);

    try {
        if (GET_RATING_INPUT_METHOD.equals(methodName)) {
            response = getRatingInput(inputs, output, options);
        } else {
            throw new UnsupportedOperationException('this method name is incorrect
or not supported');
        }
    } catch (Exception e) {
        response = false;
        output.put('success', false);
        output.put('errorMessage', 'Exception occurred: ' + e.getMessage());
        System.debug(LoggingLevel.ERROR, 'Exception in invokeMethod: ' + methodName
+ ', Error: ' + e.getMessage());
        System.debug(LoggingLevel.ERROR, 'Stack trace: ' + e.getStackTraceString());
    }
    System.debug('Output from invoke:' + output);
    return response;
}

private Boolean getRatingInput(Map<String, Object> inputs, Map<String, Object>
output, Map<String, Object> options) {

    Map<String, Object> getRatingInputPayload = inputs;
    // Handle null input
    if (getRatingInputPayload == null) {
        output.put('success', false);
        output.put('errorMessage', 'ratingInput is required in the payload for
rating');
        output.put('errorType', 'VALIDATION_ERROR');
        return false;
    }
    System.debug('Create Rating Input: ' + JSON.serialize(getRatingInputPayload));

    try {
        System.debug('Calling Invocable Action: getInsuranceRatingInput');

```

```

        // Create and configure the invocable action
        Invocable.Action action =
        Invocable.Action.createStandardAction('getInsuranceRatingInput');

        // productIds is required
        setInvocationParameter(action, getRatingInputPayload, 'productIds', true);

        // salesTransactionTypeId is optional
        setInvocationParameter(action, getRatingInputPayload,
'salesTransactionTypeId', false);

        // transactionType is optional
        setInvocationParameter(action, getRatingInputPayload, 'transactionType',
false);

        // effectiveDate is optional
        setInvocationParameter(action, getRatingInputPayload, 'effectiveDate',
false);

        // ratingDate is optional
        setInvocationParameter(action, getRatingInputPayload, 'ratingDate', false);

        // rootProductCodes is optional
        setInvocationParameter(action, getRatingInputPayload, 'rootProductCodes',
true);

        // additionalFields is optional, but to be serialized
        setInvocationParameter(action, getRatingInputPayload, 'additionalFields',
true);

        // Execute the action
        List<Invocable.Action.Result> results = action.invoke();

        if (results != null && !results.isEmpty()) {
            Invocable.Action.Result result = results[0];
            System.debug('IA result: ' + JSON.serialize(result));

            if (result.isSuccess()) {
                System.debug('Get Rating Input created successfully from Invocable
Action');
                output.put('output', result.getOutputParameters());

                // Extract key information from the result
                Map<String, Object> outputParams = result.getOutputParameters();
                if (outputParams != null) {
                    if (outputParams.containsKey('getRatingInputOutputRep')) {
                        output.put('getRatingInputOutputRep',
outputParams.get('getRatingInputOutputRep'));
                    }
                    System.debug('Rating Input data extracted: ' + outputParams);
                }
                List<Invocable.Action.Error> actionErrors = result.getErrors();

```

```

        if (actionErrors != null && !actionErrors.isEmpty()) {
            output.put('success', false);
            output.put('error', result.getErrors());
            return false;
        }
        output.put('success', true);
        return true;
    } else {
        // Action failed - get error details
        List<Invocable.Action.Error> actionErrors = result.getErrors();
        String errorMessage = 'Get Rating Input Invocable Action failed';

        if (actionErrors != null && !actionErrors.isEmpty()) {
            List<String> errorMessages = new List<String>();
            List<Map<String, Object>> errorDetailsList = new List<Map<String,
Object>>>();

            for (Invocable.Action.Error actionError : actionErrors) {
                errorMessages.add(actionError.getMessage());
                errorDetailsList.add(new Map<String, Object>{
                    'message' => actionError.getMessage(),
                    'code' => actionError.getCode()
                });
            }
            errorMessage = String.join(errorMessages, '; ');
            output.put('actionErrors', errorDetailsList);
        }
        output.put('success', false);
        output.put('errorMessage', errorMessage);

        System.debug(LoggingLevel.ERROR, 'Get Rating Input Invocable Action
failed: ' + errorMessage);
        return false;
    }

    } else {
        output.put('success', false);
        output.put('errorMessage', 'Get Rating Input Invocable Action returned
null or empty results');
        System.debug(LoggingLevel.WARN, 'Get Rating Input Invocable Action
returned null/empty results');
    }
    return false;
} catch (Exception e) {
    output.put('success', false);
    output.put('error', e.getMessage());
    return false;
}
}

private void setInvocationParameter(Invocable.Action action, Map<String, Object>
quotePayload, String inputAttribute, boolean serializeJSON) {

```

```

        if (quotePayload.containsKey(inputAttribute) && quotePayload.get(inputAttribute)
!= null) {
            System.debug('InsuranceGetRatingInputApexService::' + inputAttribute + ':'
+ JSON.serialize(quotePayload.get(inputAttribute)));
            if (serializeJSON) {
                action.setInvocationParameter(inputAttribute,
JSON.serialize(quotePayload.get(inputAttribute)));
            } else {
                action.setInvocationParameter(inputAttribute,
quotePayload.get(inputAttribute));
            }
        } else {
            System.debug('Payload does not contain attribute: ' + inputAttribute);
        }
    }
}

```

This sample response is for the Get Insurance Rating Input action.

```

[
  {
    "actionName": "getInsuranceRatingInput",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outputValues": {
      "isSuccess": true,
      "getRatingInputOutputRep": {
        "ratingInputs": [
          {
            "additionalFields": {},
            "attributes": {},
            "instanceKeys": [
              "Key1"
            ],
            "productCode": "ProductCode1"
          }
        ],
        "success": true,
        "message": "successful"
      },
      "errorsList": []
    },
    "sortOrder": -1,
    "version": 1
  }
]

```

Reprice Insurance Product Action

Recalculate the price of insurance products based on user inputs.

This action is available in API version 63.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/repriceInsuranceProduct

Formats

JSON

HTTP Methods

GET, POST

Authentication

Bearer Token

Inputs

Input	Details
addedNodes	Type String Description Product instances details that are added.
additionalFields	Type String Description Map of additional fields that must be updated in the insurance quote context.
contextId	Type String Description Required. Context ID of the Insurance quote.
deletedNodes	Type String Description Set of deleted product instances details.
effectiveDate	Type Date Description Date that the quote becomes effective.
executeRating	Type String

Input	Details
	Description Indicates whether rating must be executed (<code>true</code>) or not (<code>false</code>). The default value is <code>true</code> .
<code>executeConfigurationRules</code>	Type Boolean Description Indicates whether configuration rules must be executed (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
<code>pricingProcedureApiName</code>	Type String Description API name of the pricing procedure that's used to price the product. This property is deprecated from version 66.0 and later.
<code>ratingDate</code>	Type Date Description Date that's used to find the active pricing procedure.
<code>returnContextJson</code>	Type Date Description Indicates whether context JSON must be returned (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
<code>returnProductDetails</code>	Type Date Description Indicates whether product details must be returned (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
<code>returnRatingResults</code>	Type Date Description Indicates whether rating results must be returned as an Apex class (<code>true</code>) or not (<code>false</code>). The default value is <code>false</code> .
<code>updatedNodes</code>	Type String

Input	Details
	Description Set of product instances details that are updated.

Outputs

Output	Details
contextId	Type String Description Context ID of the insurance quote.
pricingResults	Type Map<String, Object> Description Pricing details of the insurance product that are recalculated.
productRatingOutput	Type Apex-defined Description An Apex ConnectApi__ProductRatingOutputRepresentation record that contains the product rating details.

Example

Sample Request

```
{
  "inputs": [
    {
      "contextId": "273e01fd6b6f9f21428869eb784ee756976afb0e9bdc49f2a360be397a10ca12",
      "effectiveDate": "2024-10-18",
      "addedNodes":
      "[{"instances":["Auto","Auto","No Drive"],"product":"Auto","attributes":{"DriveActions":4},"tagRecord":{"03x0004m7x"}]",
      "returnRatingResults": true,
      "returnContextJson": true,
      "returnProductDetails": true
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for repriceInsuranceProduct invocable action
public class RepriceInsuranceProductInvoke {
    private static final String ACTION_NAME = 'repriceInsuranceProduct';

    public static Boolean invokeMethod(
        Map<String, Object> inputs,
        Map<String, Object> output,
        Map<String, Object> options
    ) {
        Map<String, Object> patchPayload = (Map<String, Object>)
getValueFromMap('patchPayload', inputs, options);
        System.debug('Rating patch Input: ' + patchPayload);

        if (patchPayload == null) {
            output.put('success', false);
            output.put('errorMessage', 'patchPayload is required for rating patch');
            output.put('errorType', 'VALIDATION_ERROR');
            return false;
        }

        try {
            System.debug('Calling Invocable Action: repriceInsuranceProduct');

            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            System.debug('Context Id:::::::::' + patchPayload.get('contextId'));
            action.setInvocationParameter('contextId', patchPayload.get('contextId'));

            System.debug('Updated Nodes:::::::::' + patchPayload.get('updatedNodes'));
            setInvocationParameter(action, patchPayload, 'updatedNodes', true);

            System.debug('Added Nodes:::::::::' + patchPayload.get('addedNodes'));
            setInvocationParameter(action, patchPayload, 'addedNodes', true);

            System.debug('Deleted Nodes:::::::::' + patchPayload.get('deletedNodes'));
            setInvocationParameter(action, patchPayload, 'deletedNodes', true);

            if (patchPayload.containsKey('executeRating')) {
                System.debug('Options:::::::::' + patchPayload.get('executeRating'));
                action.setInvocationParameter('executeRating',
patchPayload.get('executeRating'));
            }

            if (patchPayload.containsKey('executeConfigurationRules')) {
                System.debug('Options:::::::::' +
patchPayload.get('executeConfigurationRules'));
                action.setInvocationParameter('executeConfigurationRules',
patchPayload.get('executeConfigurationRules'));
            }

            if (patchPayload.containsKey('returnContextJSON')) {
                System.debug('Options:::::::::' + patchPayload.get('returnContextJSON'));
            }
        }
    }
}
```



```

        action.setInvocationParameter('returnContextJSON',
patchPayload.get('returnContextJSON'));
    }

    if (patchPayload.containsKey('returnProductDetails')) {
        System.debug('Options::::::::::' + patchPayload.get('returnProductDetails'));

        action.setInvocationParameter('returnProductDetails',
patchPayload.get('returnProductDetails'));
    }

    if (patchPayload.containsKey('returnRatingResults')) {
        System.debug('Options::::::::::' + patchPayload.get('returnRatingResults'));

        action.setInvocationParameter('returnRatingResults',
patchPayload.get('returnRatingResults'));
    }

    List<Invocable.Action.Result> results = action.invoke();

    if (results != null && !results.isEmpty()) {
        Invocable.Action.Result result = results[0];

        if (result.isSuccess()) {
            System.debug('Rating patch successfully via Invocable Action');
            output.put('ratingData', result.getOutputParameters());

            Map<String, Object> outputParams = result.getOutputParameters();
            if (outputParams != null) {
                if (outputParams.containsKey('contextId')) {
                    output.put('contextId', outputParams.get('contextId'));
                }
                if (outputParams.containsKey('contextJSON')) {
                    output.put('contextJSON', outputParams.get('contextJSON'));
                }
                if (outputParams.containsKey('pricingResult')) {
                    output.put('pricingResult', outputParams.get('pricingResult'));
                }
                if (outputParams.containsKey('productDetails')) {
                    output.put('productDetails', outputParams.get('productDetails'));
                }
            }

            System.debug('Rating Patch data extracted: ' + outputParams);
        }
        List<Invocable.Action.Error> actionErrors = result.getErrors();
        if (actionErrors != null && !actionErrors.isEmpty()) {
            output.put('success', false);
            output.put('error', result.getErrors());
            return false;
        }
        output.put('success', true);
        return true;
    }

```

```

        } else {
            List<Invocable.Action.Error> actionErrors = result.getErrors();
            String errorMessage = 'Rating Patch Invocable Action failed';

            if (actionErrors != null && !actionErrors.isEmpty()) {
                List<String> errorMessages = new List<String>();
                List<Map<String, Object>> errorDetailsList = new List<Map<String,
Object>>();

                for (Invocable.Action.Error actionError : actionErrors) {
                    errorMessages.add(actionError.getMessage());
                    errorDetailsList.add(new Map<String, Object>{
                        'message' => actionError.getMessage(),
                        'code' => actionError.getCode()
                    });
                }

                errorMessage = String.join(errorMessages, ' ');
                output.put('actionErrors', errorDetailsList);
            }

            output.put('success', false);
            output.put('errorMessage', errorMessage);

            System.debug(LoggingLevel.ERROR, 'Rating Patch Invocable Action failed:
' + errorMessage);
            return false;
        }

    } else {
        output.put('success', false);
        output.put('errorMessage', 'Rating Patch Invocable Action returned null
or empty results');
        System.debug(LoggingLevel.WARN, 'Rating Patch Invocable Action returned
null/empty results');
        return false;
    }

} catch (Exception e) {
    String errorMsg = 'Exception: ' + e.getMessage();
    output.put('success', false);
    output.put('errorMessage', errorMsg);
    output.put('errorType', 'EXCEPTION');
    System.debug(LoggingLevel.ERROR, 'Exception while rating patch: ' + errorMsg);

    return false;
}

}

private static Object getValueFromMap(String key, Map<String, Object> inputs, Map<String,
Object> options) {
    if (inputs != null && inputs.get(key) != null &&
String.isNotBlank(String.valueOf(inputs.get(key)))) {

```

```

        return inputs.get(key);
    }
    else if (options != null && options.get(key) != null &&
String.isNotBlank(String.valueOf(options.get(key)))) {
        return options.get(key);
    }

    return null;
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String inputAttribute,
    Boolean serializeJSON
) {
    if (payload.containsKey(inputAttribute) && payload.get(inputAttribute) != null) {

        System.debug(inputAttribute + ': ' +
JSON.serialize(payload.get(inputAttribute)));
        if (serializeJSON) {
            action.setInvocationParameter(inputAttribute,
JSON.serialize(payload.get(inputAttribute)));
        } else {
            action.setInvocationParameter(inputAttribute, payload.get(inputAttribute));
        }
    } else {
        System.debug('payload does not contain attribute: ' + inputAttribute);
    }
}
}

```

Sample Response

```

[
  {
    "actionName": "repriceInsuranceProduct",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "contextId": "273e01fd6b6f9f21428869eb784ee756976afb0e9bdc49f2a360be397a10ca12",
      "productRatingOutput": {
        "ratingResult": {
          "childNodes": [
            {
              "childNodes": [
                {
                  "childNodes": [
                    {
                      "childNodes": [],
                      "details": {
                        "NetUnitPrice": null,

```

```

        "InstanceKey": "Driver",
        "ProductCode": "driver",
        "NetTotalPrice": 0,
        "Driver_Accident_Points": 15,
        "ProratedQLITaxAmount": 0,
        "Driver_MVR_Points": 8,
        "Age_First_Licensed": 16
    }
},
{
    "childNodes": [],
    "details": {
        "NetUnitPrice": 501.84,
        "InstanceKey": "Comprehensive",
        "ProductCode": "comprehensive",
        "NetTotalPrice": 501.84,
        "ProratedQLITaxAmount": 0
    }
}
],
"details": {
    "NetUnitPrice": 501.84,
    "InstanceKey": "Auto1",
    "ProductCode": "auto",
    "NetTotalPrice": 501.84,
    "ProratedQLITaxAmount": 0
}
}
],
"details": {
    "NetUnitPrice": 1485.84,
    "InstanceKey": "AutoSilver",
    "ProductCode": "autoSilver",
    "NetTotalPrice": 1485.84,
    "ProratedQLITaxAmount": 0
}
},
"details": {}
},
"productDetails": [
{
    "productSpecificationType": {
        "productSpecificationRecordType": {},
        "name": "InsuredParty"
    },
    "childProducts": [],
    "additionalFields": {
        "ProductCode": "driver",
        "SpecificationType": "InsuredParty",
        "BasedOnId": "11Bxx000002C1kZEAS",
        "Name": "Driver"
    },
    "availabilityDate": null,

```

```

"description": null,
"productInformation": null,
"childVariationIds": [],
"isActive": true,
"qualificationContext": null,
"productClassification": {
  "id": "11Bxx000002C1kZEAS"
},
"productComponentGroups": [],
"configureDuringSale": "Allowed",
"productRelatedComponent": null,
"id": "01ttx0000006i3iAAA",
"prices": [],
"isComponentRequired": null,
"productType": null,
"isAssetizable": true,
"isSoldOnlyWithOtherProds": false,
"attributeCategories": [
  {
    "code": "Driving History",
    "records": [
      {
        "isRequired": false,
        "code": "AgeFirstLicensed",
        "additionalFields": {},
        "attributeCategoryId": "0v3xx0000000001AAA",
        "hidden": false,
        "isConfigurable": false,
        "dataType": "NUMBER",
        "label": "Age First Licensed",
        "isReadOnly": false,
        "isCloneable": false,
        "name": "Age First Licensed",
        "id": "0tjxx0000000001FAAQ",
        "attributeNameOverride": "Age First Licensed",
        "isPriceImpacting": true,
        "developerName": "Age_First_Licensed",
        "status": "Active"
      }
    ],
    "name": "Driving History"
  }
],
"nodeType": "simpleProduct",
"displayUrl": null,
"productSellingModelOptions": [
  {
    "productSellingModel": {
      "sellingModelType": "OneTime",
      "name": "One Time",
      "id": "0jPxx0000000001EAA",
      "doesAutoRenewByDefault": false,
      "status": "Active"
    }
  }
],

```

```

        "productSellingModelId": "0jPxx0000000001EAA",
        "isDefault": true,
        "productId": "01txx0000006i3iAAA",
        "id": "0iOxx000000000zEAA"
    }
],
"productQuantity": null,
"productCode": "driver",
"name": "Driver"
}
],
"contextJSON": {
    "salesTransactions": [
        {
            "salesTransactionItems": [
                {
                    "salesTransactionItemClauses": [],
                    "childNodes": [
                        {
                            "salesTransactionItemClauses": [],
                            "childNodes": [],
                            "salesTransactionItemAttributes": [
                                {
                                    "fields": {
                                        "AttributeKey": "0tjxx000000000PAAQ",
                                        "AttributeValue": "50",
                                        "ParentReference": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",

                                        "hasObjectRef": null,
                                        "AttributePicklistValue": "0v6xx000000000ZAAQ",
                                        "IsPriceImpacting": false,
                                        "businessObjectType": "QuoteLineItemAttribute",
                                        "AttributeName": "Deductible",
                                        "AttributeDefinitionCode": "Deductible",
                                        "id": "ref_fdd78e30_1258_4ea3_9f56_c74245ae58d6",
                                        "SalesTransactionItemAttrParent":
"ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
                                        "AttributeDeveloperName": "Deductible"
                                    }
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    ],
    "salesTransactionItemRelatedObjects": [],
    "salesTransactionItemTaxes": [],
    "fields": {
        "LevelTwoBasePremium": 120,
        "IntermediateVariableTwo": null,
        "Product": "01txx0000006i3UAAQ",
        "businessObjectType": "QuoteLineItem",
        "DYN__License_State": null,
        "RevenueCloudPackagingFlag": "C",
        "ValidationResult": null,
        "DYN__AssetRecordName": null,

```

```

    "PolicyPremium": 1485.84,
    "PeriodBoundaryStartMonth": null,
    "SalesTransactionSourceAsset": null,
    "ProratedQLITaxAmount": 0,
    "id": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
    "TaxTreatment": null,
    "Subtotal": null,
    "LineItemPath":
"ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc/ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
    "DiscountAmount": null,
    "PricebookEntry": null,
    "NetUnitPrice": 590.4,
    "ParentSalesTransactionItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",

    "ObligatedAmount": null,
    "DYN__Age_First_Licensed": null,
    "LevelThreePremium": null,
    "ProductName": "Bodily Injury & Property Damage",
    "ProformaBillingPeriodAmount": null,
    "CommissionAmount": null,
    "ProductSpec": "Coverage",
    "DYN__Last_Name": null,
    "PricingTermCount": null,
    "DYN__ContactRecordName": null,
    "RoundedLineAmount": null,
    "SalesTransactionAction": null,
    "TransactionType": null,
    "PricingTermUnit": null,
    "BasisTransactionItem": null,
    "PricingSource": null,
    "StockKeepingUnit": null,
    "DYN__Year": null,
    "PartnerUnitPrice": null,
    "ItemTotalAdjustmentAmount": null,
    "ArePartialPeriodsAllowed": false,
    "DYN__Bodily_Injury_Per_Person_Limit": 1000,
    "DYN__Colour": null,
    "FactorP": null,
    "FactorO": null,
    "CustomProductName": null,
    "SalesTransactionItemParent": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",

    "Quantity": 1,
    "ContractId": null,
    "ItemGroupSummarySubtotal": null,
    "DYN__AccountRecordName": null,
    "DYN__E_Mail": null,
    "DYN__First_Name": null,
    "DYN__Bodily_Injury_Per_Accident_Limit": 1000,
    "DYN__Driver_Accident_Points": null,
    "SalesItemType": null,
    "ItemPath": "01txx0000006i3DAAQ/01txx0000006i3UAAQ"
  },
  "salesTransactionItemRelationships": [

```

```

    {
      "fields": {
        "ParentReference": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
        "AssociatedItem": "ref_681c203e_5491_4d5f_a296_02a0b3dfa181",
        "MainItemRole": null,
        "ProductRelatedComponent": "0dSxx000000000XEAQ",
        "businessObjectType": "QuoteLineRelationship",
        "AssociatedItemPricing": "IncludedInBundlePrice",
        "ProductRelationshipType": "0yoxx0000000001AAA",
        "RootItemProductCode": "autoSilver",
        "IsPriceInclusive": true,
        "RootItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
        "RootItemProductSellingModel": null,
        "AssociatedQuantScaleMethod": null,
        "MainItem": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
        "AssociatedItemRole": null,
        "SalesTrnItemRelationshipParent": null,
        "id": "ref_80623cb4_3abd_4ee0_876a_c5ca9ecc07fc",
        "MainItemProduct": "01txx0000006i3DAAQ",
        "MainItemProductSellingModel": null,
        "RootItemProduct": "01txx0000006i3DAAQ"
      }
    }
  ]
},
"salesTransactionItemAttributes": [],
"salesTransactionItemRelatedObjects": [],
"salesTransactionItemTaxes": [],
"fields": {
  "LevelTwoBasePremium": null,
  "IntermediateVariableTwo": null,
  "Product": "01txx0000006i3DAAQ",
  "businessObjectType": "QuoteLineItem",
  "DYN__License_State": null,
  "RevenueCloudPackagingFlag": "C",
  "ValidationResult": null,
  "DYN__AssetRecordName": null,
  "PolicyPremium": 1485.84,
  "PeriodBoundaryStartMonth": null,
  "SalesTransactionSourceAsset": null,
  "ProratedQLITaxAmount": 0,
  "id": "ref_e112c127_367c_41c8_a5ce_1d7e69ac58bc",
  "DYN__Age": null,
  "DYN__Property_Damage_Per_Accident_Limit": null,
  "DYN__Deductible": null,
  "STICurrencyIsoCode__std": null,
  "IntermediateVariableOne": null,
  "BillingFrequency": null,
  "DYN__State": null,
  "DYN__DOB": null,
  "ProductCode": "autoSilver",
  "DerivedPricingAttribute": false,
  "ItemPath": "01txx0000006i3DAAQ"
}

```



```

        },
        "salesTransactionItemRelationships": []
    }
},
"fields": {
    "Account": null,
    "BillingCity": null,
    "businessObjectType": "Quote",
    "SalesTransactionName": null,
    "StatusCode": null,
    "ProratedAmount": 1485.84,
    "StartDate": "2025-11-04",
    "CurrencyIsoCode__std": null,
    "BillingPostalCode": null,
    "id": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",
    "BillingState": null,
    "CensusRatingType__std": null,
    "TransactionType": "AutoTransactionType",
    "UserProfile": "System Administrator",
    "QuoteAccount": null,
    "TotalProratedAmount": 1485.84,
    "CensusType__std": null,
    "GroupCensus": null,
    "SalesTransactionType": null,
    "Pricebook": null,
    "Opportunity": null,
    "ShippingCountry": null,
    "ShippingCity": null,
    "StandardAmount": 1485.84,
    "ProratedTaxAmount": 0,
    "RatingDate": null,
    "SalesTransactionSource": "ref_ed3d9643_12e4_4709_9f74_69c8292d65b4",
    "EffectiveDate": "2025-11-04T00:00:00.000Z",
    "TotalStandardAmount": 1485.84
},
"appUsageAssignments": []
}
},
"sortOrder": -1,
"version": 1
}
]

```

Update Insurance Quote Action

Update an existing insurance quote by adding, updating, or deleting nodes and optionally executing pricing and rules.

This action is available in API version 66.0 and later.

Supported REST HTTP Methods

URI

/services/data/v/actions/standard/updateInsuranceQuote

Formats

JSON, XML

HTTP Methods

POST

Authentication

Authorization: Bearer **token**

Inputs

Input	Details
addedNodes	Type string Description JSON string that contains the set of nodes that must be added to the quote.
additionalFields	Type string Description JSON string that contains additional fields that must be updated in the quote context.
contextId	Type string Description ID of the context that's used to issue the quote update.
deletedNodes	Type string Description JSON string that contains the set of nodes that must be deleted from the quote.
pricingProcedureApiName	Type string Description API name of the pricing procedure that's used to price the product.
quoteId	Type string

Input	Details
	Description Required. ID of the insurance quote that must be updated.
quoteOptions	Type Apex-defined Description An Apex runtime_industries_insurance.UpdateInsuranceQuoteOptions record that contains the configuration and execution options to update the quote.
updatedNodes	Type string Description JSON string that contains the set of nodes that must be updated in the quote.

Outputs

Output	Details
contextId	Type string Description ID of the context that's used to issue the quote update.
errorsList	Type string Description List of errors that occurred during the update operation.
isSuccess	Type boolean Description Indicates whether the update operation is successful (<code>true</code>) or not (<code>false</code>).
quoteId	Type string Description Required. ID of the insurance quote that's updated.

Example

POST

This sample request is for the Update Insurance Quote action.

```
{
  "inputs": [
    {
      "quoteId": "0Q0xx000000001xAAA",
      "contextId": "110xx000001G01UAAS",
      "pricingProcedureApiName": "StandardPricing",
      "addedNodes":
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "22", "autoBundle": "1", "driveAge": "35"}]},
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "22", "autoBundle": "1", "driveAge": "35"}]},
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "22", "autoBundle": "1", "driveAge": "35"}]},
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "22", "autoBundle": "1", "driveAge": "35"}]}],

      "updatedNodes":
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "1", "autoBundle": "1", "driveAge": "35"}]},
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "1", "autoBundle": "1", "driveAge": "35"}]}],

      "deletedNodes":
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "1", "autoBundle": "1", "driveAge": "35"}]},
      [{"productCode": "Drive", "instanceKeys": [{"multiAuto": "1", "autoBundle": "1", "driveAge": "35"}]}],

      "additionalFields": "{\"customField1\":\"value1\",\"customField2\":\"value2\"}",
      "quoteOptions": {
        "executePricing": true,
        "executeConfigurationRules": true,
        "executeQualificationRules": false,
        "validateProductCatalog": true,
        "saveQuote": false
      }
    }
  ]
}
```

This is a sample request to call this invocable action from Apex code.

```
// Sample Apex call for updateInsuranceQuote invocable action
public class UpdateInsuranceQuoteInvoke {
    private static final String ACTION_NAME = 'updateInsuranceQuote';

    public static Boolean invokeMethod(Map<String, Object> inputs, Map<String, Object>
output) {
        if (output == null) {
            output = new Map<String, Object>();
        }

        try {
            Invocable.Action action = Invocable.Action.createStandardAction(ACTION_NAME);

            // Required input
            setInvocationParameter(action, inputs, 'quoteId', false);

            // Optional inputs
            setInvocationParameter(action, inputs, 'contextId', false);
            setInvocationParameter(action, inputs, 'addedNodes', true);
            setInvocationParameter(action, inputs, 'updatedNodes', true);
        }
    }
}
```

```

setInvocationParameter(action, inputs, 'deletedNodes', true);
setInvocationParameter(action, inputs, 'additionalFields', true);

// quoteOptions is required by the IA contract
if (inputs != null && inputs.containsKey('quoteOptions')) {
    Map<String, Object> optionsFromInput =
        (Map<String, Object>) inputs.get('quoteOptions');
    if (optionsFromInput != null) {
        runtime_industries_insurance.UpdateInsuranceQuoteOptions quoteOptions
=
        new runtime_industries_insurance.UpdateInsuranceQuoteOptions();

        if (optionsFromInput.containsKey('executePricing')) {
            quoteOptions.executePricing =
                (Boolean) optionsFromInput.get('executePricing');
        }
        if (optionsFromInput.containsKey('validateProductCatalog')) {
            quoteOptions.validateProductCatalog =
                (Boolean) optionsFromInput.get('validateProductCatalog');
        }
        if (optionsFromInput.containsKey('executeConfigurationRules')) {
            quoteOptions.executeConfigurationRules =
                (Boolean) optionsFromInput.get('executeConfigurationRules');
        }
        if (optionsFromInput.containsKey('executeQualificationRules')) {
            quoteOptions.executeQualificationRules =
                (Boolean) optionsFromInput.get('executeQualificationRules');
        }
        if (optionsFromInput.containsKey('saveQuote')) {
            quoteOptions.saveQuote =
                (Boolean) optionsFromInput.get('saveQuote');
        }
        action.setInvocationParameter('quoteOptions', quoteOptions);
    }
}

List<Invocable.Action.Result> results = action.invoke();
if (results != null && !results.isEmpty() && results[0].isSuccess()) {
    output.put('output', results[0].getOutputParameters());
    output.put('success', true);
    return true;
}

output.put('success', false);
if (results != null && !results.isEmpty()) {
    output.put('errors', results[0].getErrors());
}
return false;
} catch (Exception e) {
    output.put('success', false);
    output.put('errorMessage', e.getMessage());
    return false;
}

```

```

    }
}

private static void setInvocationParameter(
    Invocable.Action action,
    Map<String, Object> payload,
    String attribute,
    Boolean serializeJson
) {
    if (payload != null && payload.containsKey(attribute) && payload.get(attribute)
    != null) {
        if (serializeJson) {
            action.setInvocationParameter(attribute,
            JSON.serialize(payload.get(attribute)));
        } else {
            action.setInvocationParameter(attribute, payload.get(attribute));
        }
    }
}
}
}

```

This sample response is for the Update Insurance Quote action.

```

[
  {
    "actionName": "repriceInsuranceProduct",
    "errors": null,
    "invocationId": null,
    "isSuccess": true,
    "outcome": null,
    "outputValues": {
      "isSuccess": true,
      "quoteId": "0Q0xx000000001xAAA",
      "contextId": "110xx000001G01UAAS",
      "errorsList": "[]"
    },
    "sortOrder": -1,
    "version": 1
  }
]

```

Insurance Quoting Metadata API Types

Metadata API enables you to access some types and feature settings that you can customize in the user interface.

[Flow for Insurance Quoting](#)

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

Flow for Insurance Quoting

Represents the metadata associated with a flow. Use Flow to create an application that takes users through a series of pages to query and update records in the database. You can also execute logic and provide branching capability based on user input to build dynamic applications.

FlowActionCall

Insurance Quoting exposes additional `actionType` values for the `FlowActionCall` Metadata type.

Field Name	Field Type	Description
<code>actionType</code>	<code>InvokableActionType</code> (enumeration of type string)	Required. The action type. Additional valid values only for Insurance Quoting include: • <code>addEligibleInsuranceClauses</code>—Add eligible insurance clauses to a quote or context. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID along with insurance product clause details. • <code>createInsuranceQuote</code>—Create an Insurance quote by using a context ID or a set of user inputs that represent quote details. • <code>createInsuranceRating</code>—Create a runtime context that contains the details to create a quote. • <code>generateInsuranceClauses</code>—Generate insurance clauses based on the specified quote and context information. You must specify either quote ID or context ID to generate insurance clauses. • <code>getEligibleInsuranceClauses</code>—Get eligible insurance clauses based on quote ID and context ID. You must specify either a quote ID or a context ID, and either an instance key or a quote line item ID to get the eligible insurance clauses. • <code>getInsuranceQuoteDetails</code>—Get the details of an insurance quote. • <code>getInsuranceRatingInput</code>—Get the rating input data for the insurance products to support the pricing calculations and coverage analysis. • <code>repriceInsuranceProduct</code>—Recalculate the price of Insurance products based on user inputs. • <code>repriceInsuranceProduct</code>—Recalculate the price of insurance products based on user inputs. • <code>updateInsuranceQuote</code>—Update an existing insurance quote by adding, updating, or deleting nodes and optionally executing pricing and rules.

Insurance Quoting Developer Journey

Learn how to use the Insurance Quoting and Rating APIs to build user journeys for creating and updating quotes with Salesforce Flows and Omniscripts. These APIs help you build user journeys for users, such as insurance representatives who use the Salesforce console as well as the policyholders and partners who use the services through community portals.

Refer to these use cases to deepen your knowledge about these user journeys.

[Build a Self-service Quote and Policy Creation Functionality for Community Portal Users](#)

This user journey applies when a user, such as a policyholder or partner, logs into a customer community portal to select a product, view its price, and configure its attributes. The user then creates a quote, and proceeds to issue a policy.

[Calculate Quote Price Without Creating a Formal Quote](#)

This user journey applies when a customer wants to see an estimated quote price based on their requirements without having to save a formal quote.

[Implement Quote Creation Functionality for Insurance Representatives](#)

This user journey applies when a policyholder contacts an insurance company and an insurance representative guides them through identifying coverage needs, configuring and pricing a quote, and sharing it for review.

[Implement Quote Update Functionality for Insurance Representatives](#)

This user journey applies when a policyholder wants to update an existing quote, such as adding, deleting, or editing quote line items.

Build a Self-service Quote and Policy Creation Functionality for Community Portal Users

This user journey applies when a user, such as a policyholder or partner, logs into a customer community portal to select a product, view its price, and configure its attributes. The user then creates a quote, and proceeds to issue a policy.

Assumption

It's assumed that the user logs into the community portal, selects a product to generate a quote, and then gets the policy issued.

Use these steps to build the user journey.

1. **Generate Priced Context**—Make a POST request on the [Insurance Product Rating](#) Connect API with the required inputs to generate the price adjusted context.
This request returns the context ID in the response.
2. **Create Quote**—Use the context ID to create a quote either by:
 - Making a POST request on the [Insurance Quote](#) Connect API, or
 - Using the [Create Insurance Quote](#) invocable action.
 This request returns a quote ID in the response.
3. **Get Quote Context**—Use the quote ID to retrieve the quote details either by:
 - Making a GET request on the [Insurance Quote Details](#) Connect API, or
 - Using the [Get Insurance Quote Details](#) invocable action.
 This request returns the context ID in the response that contains the context of the quote.
4. **Issue Policy**—Use the retrieved quote context to create a policy record either by:
 - Making a POST request on the [Issue Insurance Policy](#) Connect API, or

- Using the [Issue Insurance Policy](#) invocable action.

Calculate Quote Price Without Creating a Formal Quote

This user journey applies when a customer wants to see an estimated quote price based on their requirements without having to save a formal quote.

Use the Quoting API if Quoting is managed within Salesforce. Use the Product Rating API when Quoting is not managed by Salesforce or when a third-party system handles the quoting process.

1. Calculate the quote price by using one of these methods.
 - **Quoting API**—Make a POST request on the [Insurance Quote](#) Connect API. Set the `saveQuote` parameter to `false`.
 - **Product Rating API**—Make a POST request on the [Insurance Product Rating](#) Connect API to generate a price-adjusted context based on user inputs without creating a formal quote.

Implement Quote Creation Functionality for Insurance Representatives

This user journey applies when a policyholder contacts an insurance company and an insurance representative guides them through identifying coverage needs, configuring and pricing a quote, and sharing it for review.

Assumption

It's assumed that a valid quote ID is available to update the existing quote.

The insurance representative collects the policyholder's requirements and adds any other additional rules during the API call to create a quote.

1. **Create Quote**—Create a quote either by:
 - Making a POST request on the [Insurance Quote](#) Connect API, or
 - Using the [Create Insurance Quote](#) invocable action.

This request returns a quote ID in the response.
2. **(Optional) Invoke Underwriting Review**—To invoke an underwriting review after the initial quote is created, make a POST request on the [Insurance Invoke Underwriting Rule](#) Connect API with the necessary inputs.
3. Use the quote ID that's generated during the quote creation process for further processing, such as policy issuance.

Implement Quote Update Functionality for Insurance Representatives

This user journey applies when a policyholder wants to update an existing quote, such as adding, deleting, or editing quote line items.

Assumption

It's assumed that the insurance representative has collected the policyholder's requirements to update a quote.

Use the quote ID to update the quote.

1. Make a PATCH request on the [Insurance Quote Details](#) Connect API to update the quote.

This request returns a new price-adjusted quote ID in the response.

CHAPTER 8 Insurance Apex Reference

In this chapter ...

- [runtime_industries_insurance](#)
Namespace

Use built-in Apex classes grouped by namespace.

runtime_industries_insurance Namespace

The runtime_industries_insurance namespace provides Options classes to control optional input parameters for managing insurance operations, such as creating and updating insurance quotes, generating insurance clauses, and running insurance ratings. Use these classes when calling Insurance APIs from Apex to control pricing, configuration rules, qualification rules, and product catalog validation.

The runtime_industries_insurance namespace includes these classes.

Usage

You can access the runtime_industries_insurance classes if your org has the Insurance feature. These Options classes are used with group insurance quoting, product configuration, and policy administration workflows.

[AddEligibleInsuranceClausesOptions Class](#)

Contains options for adding eligible product clauses to an insurance product. Use this class when configuring which clauses, such as exclusions or coverage terms, are available for selection on an insurance product.

[CreateInsuranceQuoteOptions Class](#)

Contains options for creating an insurance quote. Use this class to control pricing execution, product catalog validation, configuration and qualification rules, quote persistence, and the product selling model when creating group or individual insurance quotes.

[CreateInsuranceRatingOptions Class](#)

Contains options for creating an insurance rating. Use this class to control pricing execution, configuration rules, and the response data, such as context JSON, product details, rating results, when running a rating operation.

[GenerateInsuranceClausesOptions Class](#)

Contains options for generating insurance clauses. Use this class to control context saving, manual clause generation, and asynchronous clause generation when generating product clauses for insurance products.

[UpdateInsuranceQuoteOptions Class](#)

Contains options for updating an existing insurance quote. Use this class to control pricing execution, configuration and qualification rules, product catalog validation, and quote persistence when updating insurance quotes.

AddEligibleInsuranceClausesOptions Class

Contains options for adding eligible product clauses to an insurance product. Use this class when configuring which clauses, such as exclusions or coverage terms, are available for selection on an insurance product.

Namespace

[runtime_industries_insurance](#) on page 717

Usage

Use the AddEligibleInsuranceClausesOptions class with the [Add Eligible Insurance Clauses](#) invocable action to add eligible insurance clauses.

Example

When calling the [Add Eligible Insurance Clauses](#) invocable action from Apex, create an `AddEligibleInsuranceClausesOptions` instance and set the properties, then use it as `addClauseOptions`.

```
// Optional optionFlags for addEligibleInsuranceClauses
if (inputs != null && inputs.containsKey('optionFlags')) {
    Map<String, Object> addClauseOptions =
        (Map<String, Object>) inputs.get('optionFlags');
    if (addClauseOptions != null) {
        runtime_industries_insurance.AddEligibleInsuranceClausesOptions addClauseApexOptions
        =
            new runtime_industries_insurance.AddEligibleInsuranceClausesOptions();
        if (addClauseOptions.containsKey('saveContext')) {
            addClauseApexOptions.saveContext =
                (Boolean) addClauseOptions.get('saveContext');
        }
        action.setInvocationParameter('addClauseOptions', addClauseApexOptions);
    }
}
```

[AddEligibleInsuranceClausesOptions Constructors](#)

Learn more about the constructors that are available with the `AddEligibleInsuranceClausesOptions` class.

[AddEligibleInsuranceClausesOptions Properties](#)

Set the `AddEligibleInsuranceClausesOptions` class property to save the context of a quote.

AddEligibleInsuranceClausesOptions Constructors

Learn more about the constructors that are available with the `AddEligibleInsuranceClausesOptions` class.

The `AddEligibleInsuranceClausesOptions` class includes these constructors.

[AddEligibleInsuranceClausesOptions\(saveContext\)](#)

Creates an options instance with the specified context-saving behavior for adding eligible insurance clauses.

[AddEligibleInsuranceClausesOptions\(\)](#)

Creates an options instance with default settings for adding eligible insurance clauses. Context is not saved by default.

AddEligibleInsuranceClausesOptions (saveContext)

Creates an options instance with the specified context-saving behavior for adding eligible insurance clauses.

Signature

```
public AddEligibleInsuranceClausesOptions(Boolean saveContext)
```

Parameters

saveContext

Type: Boolean

Indicates whether to save the operation context when adding eligible insurance clauses (`true`) or not (`false`).

The default value is `true`.

AddEligibleInsuranceClausesOptions()

Creates an options instance with default settings for adding eligible insurance clauses. Context is not saved by default.

Signature

```
public AddEligibleInsuranceClausesOptions()
```

AddEligibleInsuranceClausesOptions Properties

Set the AddEligibleInsuranceClausesOptions class property to save the context of a quote.

The AddEligibleInsuranceClausesOptions class includes this property.

[saveContext](#)

Indicates whether to save the operation context when adding eligible insurance clauses (true) or not (false).

saveContext

Indicates whether to save the operation context when adding eligible insurance clauses (true) or not (false).

Signature

```
public Boolean saveContext {get; set;}
```

Property Value

Type: Boolean

CreateInsuranceQuoteOptions Class

Contains options for creating an insurance quote. Use this class to control pricing execution, product catalog validation, configuration and qualification rules, quote persistence, and the product selling model when creating group or individual insurance quotes.

Namespace

[runtime_industries_insurance](#) on page 717

Usage

Use the CreateInsuranceQuoteOptions class with the [Insurance Quote \(POST\)](#) API and the [Create Insurance Quote](#) invocable action to create an insurance quote so that you can specify which operations to run and whether to save the quote or not.

Example

When calling the [Create Insurance Quote](#) invocable action from Apex, create a `CreateInsuranceQuoteOptions` instance, set the desired properties, and then use it as `quoteOptions`.

```
runtime_industries_insurance.CreateInsuranceQuoteOptions createQuoteOptions
    = new runtime_industries_insurance.CreateInsuranceQuoteOptions();
    Map<String, Object> optionsFromInput = (Map<String,
Object>)quotePayload.get('quoteOptions');
    if (optionsFromInput.containsKey('executePricing')) {
        createQuoteOptions.executePricing =
(Boolean)optionsFromInput.get('executePricing');
    }
    if (optionsFromInput.containsKey('validateProductCatalog')) {
        createQuoteOptions.validateProductCatalog =
(Boolean)optionsFromInput.get('validateProductCatalog');
    }
    if (optionsFromInput.containsKey('executeConfigurationRules')) {
        createQuoteOptions.executeConfigurationRules =
(Boolean)optionsFromInput.get('executeConfigurationRules');
    }
    if (optionsFromInput.containsKey('executeQualificationRules')) {
        createQuoteOptions.executeQualificationRules =
(Boolean)optionsFromInput.get('executeQualificationRules');
    }
    if (optionsFromInput.containsKey('saveQuote')) {
        createQuoteOptions.saveQuote = (Boolean)optionsFromInput.get('saveQuote');
    }
    if (optionsFromInput.containsKey('productSellingModel')) {
        createQuoteOptions.productSellingModel =
(String)optionsFromInput.get('productSellingModel');
    }
    action.setInvocationParameter('quoteOptions', createQuoteOptions);
```

[CreateInsuranceQuoteOptions Constructors](#)

Learn more about the constructors that are available with the `CreateInsuranceQuoteOptions` class.

[CreateInsuranceQuoteOptions Properties](#)

Set the `CreateInsuranceQuoteOptions` class properties to create an insurance quote.

CreateInsuranceQuoteOptions Constructors

Learn more about the constructors that are available with the `CreateInsuranceQuoteOptions` class.

The `CreateInsuranceQuoteOptions` class includes these constructors.

[CreateInsuranceQuoteOptions\(executePricing, validateProductCatalog, executeConfigurationRules, executeQualificationRules, saveQuote, productSellingModel\)](#)

Creates an options instance for quote creation with explicit control over pricing, validation, rules execution, quote saving, and product selling model.

[CreateInsuranceQuoteOptions\(\)](#)

Creates an options instance with the default settings for creating an insurance quote.

CreateInsuranceQuoteOptions(executePricing, validateProductCatalog, executeConfigurationRules, executeQualificationRules, saveQuote, productSellingModel)

Creates an options instance for quote creation with explicit control over pricing, validation, rules execution, quote saving, and product selling model.

Signature

```
public CreateInsuranceQuoteOptions(Boolean executePricing, Boolean
validateProductCatalog, Boolean executeConfigurationRules, Boolean
executeQualificationRules, Boolean saveQuote, String productSellingModel)
```

Parameters*executePricing*

Type: Boolean

Indicates whether to execute pricing when creating the quote (`true`) or not (`false`). The default value is `false`.

validateProductCatalog

Type: Boolean

Indicates whether to validate against product catalog (`true`) or not (`false`). The default value is `false`.

executeConfigurationRules

Type: Boolean

Indicates whether to run product configuration rules (`true`) or not (`false`). The default value is `false`.

executeQualificationRules

Type: Boolean

Indicates whether to run qualification rules (`true`) or not (`false`). The default value is `false`.

saveQuote

Type: Boolean

Indicates whether to save the details after the quote is created (`true`) or not (`false`). The default value is `false`.

productSellingModel

Type: String

ID of the product selling model for the quote.

CreateInsuranceQuoteOptions()

Creates an options instance with the default settings for creating an insurance quote.

Signature

```
public CreateInsuranceQuoteOptions()
```

CreateInsuranceQuoteOptions Properties

Set the CreateInsuranceQuoteOptions class properties to create an insurance quote.

The CreateInsuranceQuoteOptions class includes these properties.

[executeConfigurationRules](#)

Indicates whether product configuration rules are executed when creating the insurance quote (true) or not (false).

[executePricing](#)

Indicates whether pricing is executed when creating the insurance quote (true) or not (false).

[executeQualificationRules](#)

Indicates whether qualification rules are executed when creating the insurance quote (true) or not (false).

[productSellingModel](#)

Identifies the product selling model that's used for the insurance quote, for example, group or individual.

[saveQuote](#)

Indicates whether the created quote is saved in the database (true) or not (false).

[validateProductCatalog](#)

Indicates whether the product catalog is validated when creating the insurance quote (true) or not (false).

executeConfigurationRules

Indicates whether product configuration rules are executed when creating the insurance quote (true) or not (false).

Signature

```
public Boolean executeConfigurationRules {get; set;}
```

Property Value

Type: Boolean

executePricing

Indicates whether pricing is executed when creating the insurance quote (true) or not (false).

Signature

```
public Boolean executePricing {get; set;}
```

Property Value

Type: Boolean

executeQualificationRules

Indicates whether qualification rules are executed when creating the insurance quote (true) or not (false).

Signature

```
public Boolean executeQualificationRules {get; set;}
```

Property Value

Type: Boolean

productSellingModel

Identifies the product selling model that's used for the insurance quote, for example, group or individual.

Signature

```
public String productSellingModel {get; set;}
```

Property Value

Type: String

saveQuote

Indicates whether the created quote is saved in the database (true) or not (false).

Signature

```
public Boolean saveQuote {get; set;}
```

Property Value

Type: Boolean

validateProductCatalog

Indicates whether the product catalog is validated when creating the insurance quote (true) or not (false).

Signature

```
public Boolean validateProductCatalog {get; set;}
```

Property Value

Type: Boolean

CreateInsuranceRatingOptions Class

Contains options for creating an insurance rating. Use this class to control pricing execution, configuration rules, and the response data, such as context JSON, product details, rating results, when running a rating operation.

Namespace

[runtime_industries_insurance](#) on page 717

Usage

Use the `CreateInsuranceRatingOptions` class when using the [Create Insurance Rating](#) invocable action to create an insurance rating so that you can specify which operations to run and what response data to include.

Example

Create a `CreateInsuranceRatingOptions` instance and set properties, then use it (or a serialized map) as `ratingOptions` when calling the [Create Insurance Rating](#) invocable action from Apex.

```
// Build rating options for createInsuranceRating
runtime_industries_insurance.CreateInsuranceRatingOptions ratingOptions =
    new runtime_industries_insurance.CreateInsuranceRatingOptions(
        true,    // executePricing
        true,    // executeConfigurationRules
        false,   // returnContextJSON
        true,    // returnProductDetails
        true     // returnRatingResults
    );
// Pass ratingOptions when invoking the createInsuranceRating action
action.setInvocationParameter('ratingOptions', ratingOptions);
```

[CreateInsuranceRatingOptions Constructors](#)

Learn more about the constructors that are available with the `CreateInsuranceRatingOptions` class.

[CreateInsuranceRatingOptions Properties](#)

Set the properties in the `CreateInsuranceRatingOptions` class to create insurance ratings.

CreateInsuranceRatingOptions Constructors

Learn more about the constructors that are available with the `CreateInsuranceRatingOptions` class.

The `CreateInsuranceRatingOptions` class includes these constructors.

[CreateInsuranceRatingOptions\(executePricing, executeConfigurationRules, returnContextJSON, returnProductDetails, returnRatingResults\)](#)

Creates an options instance for rating with explicit control over pricing, configuration rules, and which data is returned in the response.

[CreateInsuranceRatingOptions\(\)](#)

Creates an options instance with the default settings for creating an insurance rating.

CreateInsuranceRatingOptions(executePricing, executeConfigurationRules, returnContextJSON, returnProductDetails, returnRatingResults)

Creates an options instance for rating with explicit control over pricing, configuration rules, and which data is returned in the response.

Signature

```
public CreateInsuranceRatingOptions(Boolean executePricing, Boolean
executeConfigurationRules, Boolean returnContextJSON, Boolean returnProductDetails,
Boolean returnRatingResults)
```

Parameters

executePricing

Type: Boolean

Indicates whether to execute pricing for the rating (`true`) or not (`false`). The default value is `true`.

executeConfigurationRules

Type: Boolean

Indicates whether to run product configuration rules (`true`) or not (`false`). The default value is `false`.

returnContextJSON

Type: Boolean

Indicates whether to include context as JSON in the response (`true`) or not (`false`). The default value is `false`.

returnProductDetails

Type: Boolean

Indicates whether to include product details in the response (`true`) or not (`false`). The default value is `false`.

returnRatingResults

Type: Boolean

Indicates whether to include rating results in the response (`true`) or not (`false`). The default value is `false`.

CreateInsuranceRatingOptions()

Creates an options instance with the default settings for creating an insurance rating.

Signature

```
public CreateInsuranceRatingOptions()
```

CreateInsuranceRatingOptions Properties

Set the properties in the CreateInsuranceRatingOptions class to create insurance ratings.

The following are properties for CreateInsuranceRatingOptions.

[executeConfigurationRules](#)

Indicates whether to run product configuration rules (`true`) or not (`false`).

[executePricing](#)

Indicates whether to execute pricing for the rating (`true`) or not (`false`).

[returnContextJSON](#)

Indicates whether to include context as JSON in the response (`true`) or not (`false`).

[returnProductDetails](#)

Indicates whether to include product details in the response (`true`) or not (`false`).

returnRatingResults

Indicates whether to include rating results in the response (true) or not (false).

executeConfigurationRules

Indicates whether to run product configuration rules (true) or not (false).

Signature

```
public Boolean executeConfigurationRules {get; set;}
```

Property Value

Type: Boolean

executePricing

Indicates whether to execute pricing for the rating (true) or not (false).

Signature

```
public Boolean executePricing {get; set;}
```

Property Value

Type: Boolean

returnContextJSON

Indicates whether to include context as JSON in the response (true) or not (false).

Signature

```
public Boolean returnContextJSON {get; set;}
```

Property Value

Type: Boolean

returnProductDetails

Indicates whether to include product details in the response (true) or not (false).

Signature

```
public Boolean returnProductDetails {get; set;}
```

Property Value

Type: Boolean

returnRatingResults

Indicates whether to include rating results in the response (true) or not (false).

Signature

```
public Boolean returnRatingResults {get; set;}
```

Property Value

Type: Boolean

GenerateInsuranceClausesOptions Class

Contains options for generating insurance clauses. Use this class to control context saving, manual clause generation, and asynchronous clause generation when generating product clauses for insurance products.

Namespace

[runtime_industries_insurance](#) on page 717

Usage

Use `GenerateInsuranceClausesOptions` class when using the [Insurance Product Clause Generation](#) API or the [Generate Insurance Clauses](#) invocable action to generate insurance clauses so that you can control context saving and synchronous or asynchronous execution.

Example

When calling the [Generate Insurance Clauses](#) invocable action from Apex, create a `GenerateInsuranceClausesOptions` instance and set the properties, then use it as `optionFlags`.

```
// Optional optionFlags for generateInsuranceClauses
if (inputs != null && inputs.containsKey('optionFlags')) {
    Map<String, Object> generateClauseOptions =
        (Map<String, Object>) inputs.get('optionFlags');
    if (generateClauseOptions != null) {
        runtime_industries_insurance.GenerateInsuranceClausesOptions
generateClauseApexOptions =
        new runtime_industries_insurance.GenerateInsuranceClausesOptions();
        if (generateClauseOptions.containsKey('saveContext')) {
            apexOptions.saveContext =
                (Boolean) generateClauseOptions.get('saveContext');
        }
        if (generateClauseOptions.containsKey('generateManualClauses')) {
            apexOptions.generateManualClauses =
                (Boolean) generateClauseOptions.get('generateManualClauses');
        }
        if (generateClauseOptions.containsKey('generateClausesAsync')) {
            generateClauseApexOptions.generateClausesAsync =
                (Boolean) generateClauseOptions.get('generateClausesAsync');
        }
    }
}
```

```
    }  
    action.setInvocationParameter('optionFlags', generateClauseApexOptions);  
  }  
}
```

[GenerateInsuranceClausesOptions Constructors](#)

Learn more about the constructors that are available with the GenerateInsuranceClausesOptions class.

[GenerateInsuranceClausesOptions Properties](#)

Set the properties in the GenerateInsuranceClausesOptions class to generate insurance product clauses.

GenerateInsuranceClausesOptions Constructors

Learn more about the constructors that are available with the GenerateInsuranceClausesOptions class.

The GenerateInsuranceClausesOptions class includes these constructors.

[GenerateInsuranceClausesOptions\(saveContext, generateManualClauses, generateClausesAsync\)](#)

Creates an options instance with full control over context saving, manual clause generation, and asynchronous execution.

[GenerateInsuranceClausesOptions\(saveContext, generateManualClauses\)](#)

Creates an options instance for synchronously generating insurance clauses with context and manual clause settings.

[GenerateInsuranceClausesOptions\(\)](#)

Creates an options instance with the default settings for generating insurance clauses.

GenerateInsuranceClausesOptions(saveContext, generateManualClauses, generateClausesAsync)

Creates an options instance with full control over context saving, manual clause generation, and asynchronous execution.

Signature

```
public GenerateInsuranceClausesOptions(Boolean saveContext, Boolean  
generateManualClauses, Boolean generateClausesAsync)
```

Parameters

saveContext

Type: Boolean

Indicates whether to save the operation context (`true`) or not (`false`). The default value is `true`.

generateManualClauses

Type: Boolean

Indicates whether to generate manual clauses along with automated clauses (`true`) or not (`false`). The default value is `false`.

generateClausesAsync

Type: Boolean

Indicates whether to generate clauses asynchronously (`true`) or not (`false`). The default value is `false`.

GenerateInsuranceClausesOptions(saveContext, generateManualClauses)

Creates an options instance for synchronously generating insurance clauses with context and manual clause settings.

Signature

```
public GenerateInsuranceClausesOptions(Boolean saveContext, Boolean
generateManualClauses)
```

Parameters

saveContext

Type: Boolean

Indicates whether to save the operation context (`true`) or not (`false`).

generateManualClauses

Type: Boolean

Indicates whether to generate manual clauses along with automated clauses (`true`) or not (`false`).

GenerateInsuranceClausesOptions()

Creates an options instance with the default settings for generating insurance clauses.

Signature

```
public GenerateInsuranceClausesOptions()
```

GenerateInsuranceClausesOptions Properties

Set the properties in the GenerateInsuranceClausesOptions class to generate insurance product clauses.

The GenerateInsuranceClausesOptions class includes these properties.

[generateClausesAsync](#)

Indicates whether clauses are generated asynchronously (`true`) or synchronously (`false`).

[generateManualClauses](#)

Indicates whether to generate manual clauses along with system-generated clauses (`true`) or not (`false`).

[saveContext](#)

Indicates whether to save the operation context (`true`) or not (`false`).

generateClausesAsync

Indicates whether clauses are generated asynchronously (`true`) or synchronously (`false`).

Signature

```
public Boolean generateClausesAsync {get; set;}
```

Property Value

Type: Boolean

generateManualClauses

Indicates whether to generate manual clauses along with system-generated clauses (true) or not (false).

Signature

```
public Boolean generateManualClauses {get; set;}
```

Property Value

Type: Boolean

saveContext

Indicates whether to save the operation context (true) or not (false).

Signature

```
public Boolean saveContext {get; set;}
```

Property Value

Type: Boolean

UpdateInsuranceQuoteOptions Class

Contains options for updating an existing insurance quote. Use this class to control pricing execution, configuration and qualification rules, product catalog validation, and quote persistence when updating insurance quotes.

Namespace

[runtime_industries_insurance](#) on page 717

Usage

Use the `UpdateInsuranceQuoteOptions` class when calling the [Update Insurance Quote](#) invocable action to update an insurance quote so that you can specify which operations to run and whether to save the updated quote.

Example

When calling the [Update Insurance Quote](#) invocable action from Apex, create an `UpdateInsuranceQuoteOptions` instance, set the desired properties, and use it as `quoteOptions`.

```
// quoteOptions for updateInsuranceQuote
if (inputs != null && inputs.containsKey('quoteOptions')) {
    Map<String, Object> optionsFromInput =
```



```

        (Map<String, Object>) inputs.get('quoteOptions');
        if (optionsFromInput != null) {
            runtime_industries_insurance.UpdateInsuranceQuoteOptions updateQuoteOptions
                = new runtime_industries_insurance.UpdateInsuranceQuoteOptions();
            Map<String, Object> optionsFromInput = (Map<String,
Object>) quotePayload.get('quoteOptions');
            if (optionsFromInput.containsKey('executePricing')) {
                updateQuoteOptions.executePricing =
(Boolean) optionsFromInput.get('executePricing');
            }
            if (optionsFromInput.containsKey('validateProductCatalog')) {
                updateQuoteOptions.validateProductCatalog =
(Boolean) optionsFromInput.get('validateProductCatalog');
            }
            if (optionsFromInput.containsKey('executeConfigurationRules')) {
                updateQuoteOptions.executeConfigurationRules =
(Boolean) optionsFromInput.get('executeConfigurationRules');
            }
            if (optionsFromInput.containsKey('executeQualificationRules')) {
                updateQuoteOptions.executeQualificationRules =
(Boolean) optionsFromInput.get('executeQualificationRules');
            }
            if (optionsFromInput.containsKey('saveQuote')) {
                updateQuoteOptions.saveQuote = (Boolean) optionsFromInput.get('saveQuote');
            }
            action.setInvocationParameter('quoteOptions', updateQuoteOptions);
        }
    }
}

```

[UpdateInsuranceQuoteOptions Constructors](#)

Learn more about the constructors that are available with the UpdateInsuranceQuoteOptions class.

[UpdateInsuranceQuoteOptions Properties](#)

Set the properties in the UpdateInsuranceQuoteOptions class to update an insurance quote.

UpdateInsuranceQuoteOptions Constructors

Learn more about the constructors that are available with the UpdateInsuranceQuoteOptions class.

The UpdateInsuranceQuoteOptions class includes these constructors.

[UpdateInsuranceQuoteOptions\(executePricing, executeConfigurationRules, executeQualificationRules, validateProductCatalog, saveQuote\)](#)

Creates an options instance for quote updates with explicit control over pricing, configuration and qualification rules, catalog validation, and quote saving.

[UpdateInsuranceQuoteOptions\(\)](#)

Creates an options instance with the default settings for updating an insurance quote.

UpdateInsuranceQuoteOptions(*executePricing*, *executeConfigurationRules*, *executeQualificationRules*, *validateProductCatalog*, *saveQuote*)

Creates an options instance for quote updates with explicit control over pricing, configuration and qualification rules, catalog validation, and quote saving.

Signature

```
public UpdateInsuranceQuoteOptions(Boolean executePricing, Boolean
executeConfigurationRules, Boolean executeQualificationRules, Boolean
validateProductCatalog, Boolean saveQuote)
```

Parameters

executePricing

Type: Boolean

Indicates whether to execute pricing when updating the quote (*true*) or not (*false*). The default value is *false*.

executeConfigurationRules

Type: Boolean

Indicates whether to run product configuration rules when updating the quote (*true*) or not (*false*). The default value is *false*.

executeQualificationRules

Type: Boolean

Indicates whether to run qualification rules when updating the quote (*true*) or not (*false*). The default value is *false*.

validateProductCatalog

Type: Boolean

Indicates whether to validate against the product catalog when updating the quote (*true*) or not (*false*). The default value is *false*.

saveQuote

Type: Boolean

Indicates whether to save the updated quote (*true*) or not (*false*). The default value is *false*.

UpdateInsuranceQuoteOptions()

Creates an options instance with the default settings for updating an insurance quote.

Signature

```
public UpdateInsuranceQuoteOptions()
```

UpdateInsuranceQuoteOptions Properties

Set the properties in the UpdateInsuranceQuoteOptions class to update an insurance quote.

The UpdateInsuranceQuoteOptions class includes these properties.

[executeConfigurationRules](#)

Indicates whether to run product configuration rules when updating the quote (*true*) or not (*false*).

[`executePricing`](#)

Indicates whether to execute pricing when updating the quote (true) or not (false).

[`executeQualificationRules`](#)

Indicates whether to run qualification rules when updating the quote (true) or not (false).

[`saveQuote`](#)

Indicates whether to save the updated quote (true) or not (false).

[`validateProductCatalog`](#)

Indicates whether to validate against the product catalog when updating the quote (true) or not (false).

`executeConfigurationRules`

Indicates whether to run product configuration rules when updating the quote (true) or not (false).

Signature

```
public Boolean executeConfigurationRules {get; set;}
```

Property Value

Type: Boolean

`executePricing`

Indicates whether to execute pricing when updating the quote (true) or not (false).

Signature

```
public Boolean executePricing {get; set;}
```

Property Value

Type: Boolean

`executeQualificationRules`

Indicates whether to run qualification rules when updating the quote (true) or not (false).

Signature

```
public Boolean executeQualificationRules {get; set;}
```

Property Value

Type: Boolean

`saveQuote`

Indicates whether to save the updated quote (true) or not (false).

Signature

```
public Boolean saveQuote {get; set;}
```

Property Value

Type: Boolean

validateProductCatalog

Indicates whether to validate against the product catalog when updating the quote (true) or not (false).

Signature

```
public Boolean validateProductCatalog {get; set;}
```

Property Value

Type: Boolean

CHAPTER 9 Error Codes and Responses

In this chapter ...

- [Common Error Codes](#)
- [Insurance Policy Administration Error Codes](#)
- [Insurance Quoting Errors](#)

When using the Insurance APIs, you may encounter error codes and messages under certain conditions. For each family in Insurance, such as Insurance Policy Administration and Quoting, it lists the potential error codes that are related to a specific family and the associated error messages.

Error codes are categorized by ranges to indicate their scope and applicability. Each error code follows the format `SF-xxxx-yyyy`, where:

- `SF` identifies Salesforce error codes.
- `xxxx` specifies the Insurance API family or scope.
- `yyyy` is a unique four-digit identifier for the individual error.

Code ranges and their applicability:

- `SF-0001-xxxx`—Common error codes applicable to all Insurance APIs across all Insurance families.
- `SF-0002-xxxx`—Error codes specific to the Insurance Quoting APIs.
- `SF-0003-xxxx`—Error codes specific to the Insurance Policy Administration APIs.

The following is a list of error code details for the Insurance APIs.

Common Error Codes

Contains the error codes and error code messages that are applicable to all Insurance APIs across all Insurance families.

Error Code	HTTP Response Code	Message
SF-0001-0001	400	Specify a context ID of type {expectedContextType}.
SF-0001-0002	400	The {dateField1} must be after the {dateField2}.
SF-0001-0003	400	Specify a date in yyyy-MM-dd format.
SF-0001-0004	400	Specify an input for {inputField}.
SF-0001-0005	400	Specify a valid input for {inputField}.
SF-0001-0006	400	Specify a valid context ID.
SF-0001-0007	400	Your input is invalid. Specify a valid payload and try again.
SF-0001-0008	400	Specify the default picklist value for an attribute in {entity} and try again.
SF-0001-0009	400	Something went wrong. Check the following error and try again: {errorMessage}. If the issue persists, contact your Salesforce admin for help.
SF-0001-0010	400	An internal error has occurred. Contact your Salesforce admin for help.
SF-0001-0011	400	Looks like you don't have access to {operation} this {entity}. Your Salesforce admin can help with that.
SF-0001-0012	400	Field Excluded From Update. Error {errorMessage}
SF-0001-0013	400	{errorMessage}. Check field names are correct and user has access to the specified fields

Insurance Policy Administration Error Codes

Contains the error codes and error code messages that are applicable to the Insurance Policy Administration APIs.

Error Code	HTTP Response Code	Message
SF-0003-0001	400	The cancellation effective date can't be on or after {cancellationEffectiveDate}.
SF-0003-0002	400	Specify a cancellation effective date that is after {originalEffectiveFromDate} and before {originalEffectiveToDate}.
SF-0003-0003	400	The policy ID {insurancePolicyId} does not exist. Specify a valid policy ID and try again.

Error Code	HTTP Response Code	Message
SF-0003-0004	400	We couldn't find the <code>accountId</code> in the context. Specify an <code>accountId</code> .
SF-0003-0005	400	Specify an endorsement date that's between { <code>effectiveFromDate</code> } and { <code>effectiveToDate</code> }.
SF-0003-0006	400	The field set IDs { <code>invalidIds</code> } aren't associated with the entities { <code>entitiesList</code> }. Specify valid field set IDs and try again.
SF-0003-0007	400	Specify only one field set ID for { <code>entitiesList</code> } and try again.

Insurance Quoting Errors

Contains the error codes and error code messages that are applicable to the Insurance Quoting APIs.

Error Code	HTTP Response Code	Message
SF-0002-0001	400	For the <code>contextId</code> and <code>quoteInputs</code> parameters, specify a value and try again.
SF-0002-0002	400	Specify the <code>relatedRecordId</code> and <code>relatedRecordObjName</code> parameters in the <code>relatedRecords</code> array and try again.
SF-0002-0003	400	More than one related record mapping found for { <code>entityName</code> }. Ensure that you specify only one related record mapping for { <code>entityName</code> } and try again.
SF-0002-0004	400	{ <code>attributeList</code> } not found for product { <code>productCode</code> }. Specify supported values for the <code>attributes</code> object and try again.
SF-0002-0005	400	Values of the instance key arrays are not unique { <code>instanceKey</code> }. Specify unique instance key arrays and try again.
SF-0002-0006	400	More than one instance key array found { <code>instanceKey</code> }. Specify unique instance key arrays and try again.
SF-0002-0007	400	Value { <code>inputValue</code> } is invalid for { <code>inputField</code> }. Specify unique values for the instance key array and try again.
SF-0002-0008	400	The instance { <code>instanceKey</code> } you are trying to delete doesn't exist.
SF-0002-0009	400	The instance { <code>instanceKey</code> } you are trying to update doesn't exist.
SF-0002-0010	400	Specify the configuration parameters in the <code>configOptions</code> object and try again.
SF-0002-0011	400	Product with product code { <code>productCode</code> } failed the qualification criteria. Specify a valid product and try again.
SF-0002-0012	400	The { <code>instanceKey</code> } instance key has conflicting values for <code>attributes</code> or <code>relatedRecords</code> . Specify identical values and try again.