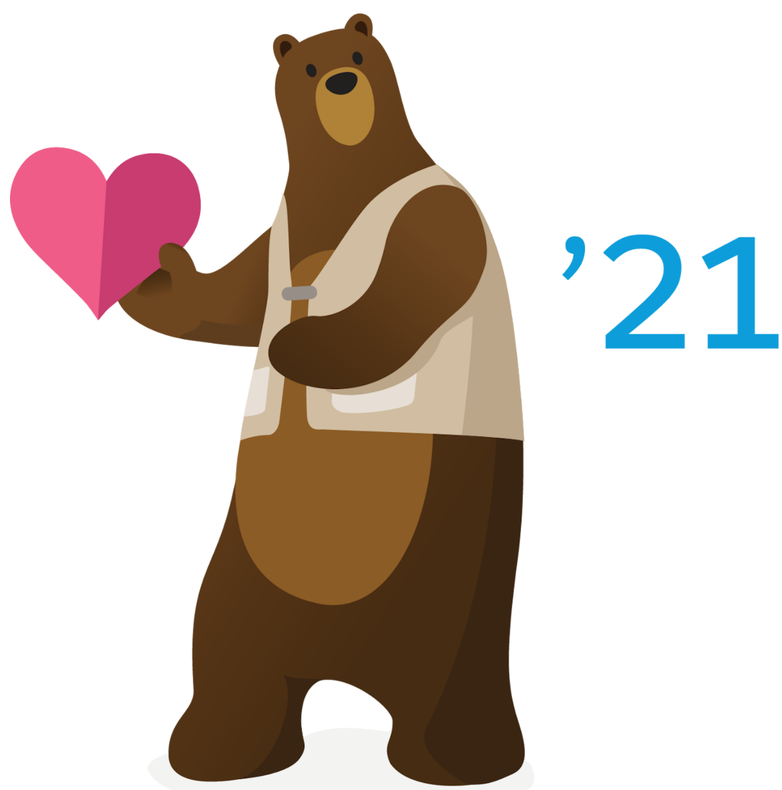
Actions Developer Guide

Developer Guide

Version 51.0, Spring '21



@salesforcedocs

Last updated: December 17, 2020

CONTENTS

CHAPTER 1 Introducing Actions

Use actions to add more functionality to your applications. Choose from standard actions, such as posting to Chatter or sending email, or create actions based on your company's needs.

For example, you can:

- Add features and functionality to your existing Lightning Platform tools.
- Build dynamic modules for Lightning Platform development into your enterprise integration tools.
- Build specialized development tools for a specific application or service.

You can batch actions to improve performance in API version 35.0 and later.

Overview

Actions allow you to build custom development tools for Lightning Platform applications.

Actions are about "getting things done" in Salesforce. They encapsulate a piece of logic that allows a user to perform some work, such as sending email. When an action runs, it saves changes in your organization by updating the database.

Actions are easy to discover and use, and also easy to understand and implement. Every button and link in Salesforce can be considered an action. A consistent Actions API and framework support the creation and distributed use of actions throughout Salesforce. Actions are available in the REST API.

The types of actions are:

Type	Description
InvocableAction	Invocable actions, also known as dynamic actions, can be invoked from a common endpoint in the REST API. They provide "describe" support – a programmatic mechanism to learn about all invocable actions on the platform. There are two types of invocable actions. **Standard action** A standard action is ready to use right away. The work it performs is predefined, along with its inputs and outputs, and they're available in every organization. **Custom action** You create custom actions because these actions require a definition. For example, to use an Apex action, create the Apex class method for the action.
QuickAction	Quick Actions, formerly known as Publisher Actions, use page layouts to make it easy for administrators to configure an action to create or update a record. The API always works with an sObject.
StandardButton	Standard buttons are URLs allowing users to either go to another page (for example, the Edit page) or accomplish some task (for example, lead conversion).
CustomButton	Custom buttons are URLs that an administrator can specify and when included on a page and clicked, will redirect a user to that URL.

To call an action from a flow, use `FlowActionCall`, as described in the *Metadata API Developer's Guide*.

The `If-Modified-Since` header can be used with actions, with a date format of `EEE, dd MMM yyyy HH:mm:ss z`. When this header is used, if the action metadata has not changed since the provided date, a `304 Not Modified` status code is returned, with no response body.

Invoking Actions

Most actions are invoked using the same JSON body format. The top-level JSON key name must be `inputs`.

Note: Invoke Salesforce Order Management actions with the corresponding Connect REST API resources or Apex ConnectApi methods, not the standard endpoints.

The following example request shows two Chatter posts made with a single Post to Chatter action.

```
POST /vXX.X/actions/standard/chatterPost

{ "inputs" :
  [
  {
    "subjectNameOrId" : "jsmith@salesforce.com",
    "type" : "user",
    "text" : "first chatter post!"
  },
  {
    "subjectNameOrId" : "hsmith@salesforce.com",
    "type" : "user",
    "text" : "second chatter post!"
  }
  ]
}
```

Here is the response.

```
[ {
  "actionName" : "chatterPost",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "feedItemId" : "0D5D0000000kynqKBA"
  }
}, {
  "actionName" : "chatterPost",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "feedItemId" : "0D5D0000000kynrKBz"
  }
} ]
```

Standard actions return their name in `actionName`. The value of `actionName` varies for custom actions.

Action	`actionName` value
Flow	The flow name

Action	`actionName` **value**
Apex	The class's invocable method name
Quick action	`<object name>.<quick action name>` For a global quick action, there's no `<object name>.` prefix.
Email alert	`<object name>.<email alert name>`

Available Actions

The following actions are available.

Action	Description
Apex Actions	Invoke Apex methods annotated with `@InvocableMethod` and include custom parameters with `@InvocableVariable`.
Batch Management Actions	Runs an active Batch Management job definition.
Create Service Report Actions	Creates a service report for a service appointment, work order, or work order line item.
Custom Notification Actions	Send custom notifications to recipients via desktop or mobile channels.
Decision Table Actions	Invokes a decision table or refreshes business rules for an active decision table.
Data Processing Engine Actions	Runs an active Data Processing Engine definition.
Email Alert Actions	Send emails from flows by reusing already-configured workflow email alerts.
Financial Services Cloud Actions	Create person accounts, financial accounts, and related records from a residential loan application for Financial Services Cloud.
Flow Actions	Invoke an active autolaunched flow or active invocable process that exists in the current org.
Generate Work Orders Actions	Generates work orders from a maintenance plan.
Knowledge Actions	Manage your Knowledge articles using invocable actions.
Loyalty Management Actions	Create and manage loyalty programs for your organization using invocable actions.
Manufacturing Cloud Actions	Recalculate and mass update account forecast and sales agreement fields for Manufacturing Cloud.
Omni-Channel Action	Create a `PendingServiceRouting` record used for Omni-Channel skills-based routing.
PlatformAction	PlatformAction is a virtual read-only object. It enables you to query for actions displayed in the UI, given a user, a context, device format, and a record ID. Examples include standard and custom buttons, quick actions, and productivity actions.

Action	Description
Post to Chatter Actions	Post a message to a specified feed, such as to a Chatter group or a case record. The message can contain mentions and topics, but only text posts are supported.
Quick Actions	Use a quick action to create a task or a case. Invoke existing quick actions, both global and object-specific, to create records, update records, or log calls.
Rebate Management Actions	Create and manage rebate programs and manage payouts and transactions using the Rebate Management invocable actions.
Refresh Metric Actions	Update a metric's Current Value field if it's linked to a summary field in a Salesforce report. The refresh runs as the metric owner.
Salesforce Order Management Actions	Manage, fulfill, and service orders in flows with Salesforce Order Management.
Session-Based Permission Set Actions	Activate or deactivate a session-based permission set for the current user's API session.
Simple Email Actions	Send an email where you specify the subject, body, and recipients.
Submit for Approval Actions	Submit a Salesforce record for approval if an approval process is defined for the current entity.

CHAPTER 2 Action Objects

This is the reference for quick actions and dynamic actions. Invocable actions are also known as dynamic actions.

Apex Actions

Invoke Apex methods annotated with `@InvocableMethod` and include custom parameters with `@InvocableVariable`.

This object is available in API version 33.0 and later.

Supported REST HTTP Methods

URI

Get a list of available Apex actions:

`/vXX.X/actions/custom/apex`

Get information about a specific Apex action:

`/vXX.X/actions/custom/apex/action_name`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Parameters

None

Example

This example invokes the Apex action called `ActionTest`, which takes a list of IDs and returns a list of associated account names. The top-level key name in the JSON request body must be `inputs`.

```
POST /vXX.X/actions/custom/apex/ActionTest

{
   "inputs":[
      {
         "ids":"001d0000023BBAoAAO"
      },
      {
         "ids":"001d0000026ChP1AAK"
      },
      {
         "ids":"001d000001tYHulAAG"
      }
   ]
}
```

Here is the Apex code:

```
public class ActionTest {

  @InvocableMethod(label='Action Test' description='Given a list of Account IDs, return
 a list of Account names.' category='Account')

  public static List<String> getAccountNames(List<ID> ids) {

     List<String> accountNames = new List<String>();
     Account a = new Account();

     for (ID id : ids) {
         a = [SELECT Name FROM Account WHERE Id = :id];
         accountNames.add(a.Name);
     }

    return accountNames;
  }
}
```

Note: The resource is the name of the Apex class, not the Apex method. In this example, the resource is `/ActionTest`, not `/getAccountNames`.

Notes

- Describe and invoke for an Apex action respect the profile access for the Apex class. If you don't have access, an error is issued.
- If you add an Apex action to a flow, and then remove the `@InvocableMethod` annotation from the Apex class, you get a runtime error in the flow.
- If an Apex action is used in a flow, packageable components that reference these elements aren't automatically included in the package. For example, if you use an email alert, you must manually add the email template that is used by that email alert. To deploy the package successfully, manually add those referenced components to the package.
- An Apex invocable action can be declared `public` or `global` in a managed package. However, that action doesn't appear in Flow Builder's list of available Apex actions. These invocable actions can still be referred to by flows within the same managed package. Global Apex invocable actions in a managed package can be used in flows outside the managed package, anywhere in the organization, and appear in Flow Builder's list of available Apex actions.

Inputs

Supply input values that correspond to the Apex action.

- A POST request body must use the JSON format specified in Invoking Actions.
- Apex methods annotated with `@InvocableMethod` must take a List as an input and return a List or `Null`. For more information, see `@InvocableMethod` Annotation in the *Apex Developer Guide*.
- Only the following primitive types are supported as inputs in a POST request body:
  - `Blob`
  - `Boolean`
  - `Date`
  - `Datetime`
  - `Decimal`

  - `Double`
  - `ID`
  - `Integer`
  - `Long`
  - `String`
  - `Time`
- A specific sObject type– the generic `sObject` type is not supported.
- A user-defined type, containing variables of the supported types and with the `InvocableVariable` annotation. Create a custom global or public Apex class to implement your data type, and make sure your class contains at least one member variable with the invocable variable annotation.

For more information, see the `@InvocableMethod` and `@InvocableVariable` annotations in the *Apex Developer Guide*.

Outputs

The Apex InvocableMethod determines the output values.

Create Service Report Actions

Creates a service report for a service appointment, work order, or work order line item.

This object is available in API version 39.0 and later.

Supported REST HTTP Methods

URI
`/v`***`XX.X`***`/actions/standard/createServiceReport`

Formats
JSON

HTTP Methods
POST

Authentication
`Authorization: Bearer` ***`token`***

Inputs

Input	Details
`entityId`	**Type** reference **Description** Required. The ID of the service appointment, work order, or work order line item that the service report is created for.

Input	Details
`signatures`	**Type** string **Description** Optional. A list of JSON definitions for a digital signature. • data—(Required) The base64 code for an image. • contentType—(Required)The file type of the signature. • signatureType—(Required) The role of the person signing; for example, "Customer." Signature Type picklist values are defined by the Salesforce admin ahead of time. Each signature block must use a different signature type, and the signature types you define in your call must match the service report template's signature types. • name—The signature block title. This value appears on the generated service report. • place—The place of signing. This value appears on the generated service report. • signedBy—The name of the person signing. This value appears on the generated service report. • signedDate—The date of signing. This value appears on the generated service report.
`templateId`	**Type** reference **Description** Required. The ID of the standard or custom service report template that is used to create the service report.

Usage

Sample Input

The following code sample creates a service report with two signatures by making an Apex callout to the *createServiceReport* action REST API resource.

```
{
   "inputs" : [ {
      "entityId" : "0WOxx000000001E",
         "signatures" : [
                  {"data":"Base64 code for the captured signature image",
                  "contentType":"image/png",
                  "name":"Customer Signature",
                  "signatureType":"Customer",
                  "place":"San Francisco",
                  "signedBy":"John Doe",
                  "signedDate":"Thu Jul 13 22:34:43 GMT 2017"
                  },
                  {"data":"Base64 code for the captured signature image",
                  "contentType":"image/png",
                  "name":"Technician Signature",
                  "signatureType":"Technician"
```

```
                }],
            "templateId" : "0SLR00000004DBFOA2"
    } ]
}
```

Custom Notification Actions

Send custom notifications to recipients via desktop or mobile channels.

Before you send a custom notification, you must first create a notification type.

Important: In orgs created in Winter ’21 or later, the Send Custom Notifications user permission is required to trigger the Send Custom Notification action in flows that run in user context, REST API calls, and Apex callouts.

The Send Custom Notifications user permission isn’t required to trigger the Send Custom Notification action in processes or flows that run in system context.

This object is available in API version 46.0 and later.

Supported REST HTTP Methods

URI
`/vXX.X/actions/standard/customNotificationAction`

Formats
JSON, XML

HTTP Methods
GET, HEAD, POST

Authentication
`Authorization: Bearer token`

Inputs

Input	Details
`customNotifTypeId`	**Type** reference **Description** Required. The ID of the Custom Notification Type being used for the notification.
`recipientIds`	**Type** reference **Description** Required. The ID of the recipient or recipient type of the notification. Valid recipient or recipient type values are: • `UserId` — The notification will be sent to this user, if this user is active.

Input	Details
	- `AccountId` — The notification will be sent to all active users who are members of this account's Account Team. Note: This recipient type is valid if account teams are enabled for your org. - `OpportunityId` — The notification will be sent to all active users who are members of this opportunity's Opportunity Team. Note: This recipient type is valid if team selling is enabled for your org. - `GroupId` — The notification will be sent to all active users who are members of this group. - `QueueId` — The notification will be sent to all active users who are members of this queue. Values can be combined in a list up to the maximum of 500 values.
`senderId`	**Type** reference **Description** Optional. The User ID of the sender of the notification.
`title`	**Type** string **Description** Required. The title of the notification, as it will be seen by recipients. Maximum characters: 250. Note: The content of mobile push notifications depends on the Display full content push notifications setting.
`body`	**Type** string **Description** Required. The body of the notification, as it will be seen by recipients. Maximum characters: 750. Note: The content of mobile push notifications depends on the Display full content push notifications setting.
`targetId`	**Type** reference **Description** Optional. The Record ID for the target record of the notification. You must specify either a `targetID` or a `targetPageRef`.
`targetPageRef`	**Type** string **Description** Optional. The `PageReference` for the navigation target of the notification.

Input	Details
	You must specify either a `targetID` or a `targetPageRef`.

Usage

GET

The following example shows how to get information about the custom notification action type:

```
curl --include --request GET \
--header "Authorization: Authorization: Bearer 00DR...xyz" \
--header "Content-Type: application/json" \
"https://instance.salesforce.com/services/data/v46.0/actions/standard/customNotificationAction"
```

POST

The following example shows how to create a custom notification action:

```
curl --include --request POST \
--header "Authorization: Authorization: Bearer 00DR...xyz" \
--header "Content-Type: application/json" \
--data '{ "inputs" :
  [
  {
    "customNotifTypeId" : "0MLR0000000008eOAA",
    "recipientIds" : ["005R0000000LSqtIAG"],
    "title" : "Custom Notification",
    "body" : "This is a custom notification.",
    "targetId" : "001R0000003fSUDIA2"
  }
  ]
}' \
"https://instance.salesforce.com/services/data/v46.0/actions/standard/customNotificationAction"
```

The response is:

```
[
   {
      "actionName" : "customNotificationAction",
      "errors" : null,
      "isSuccess" : true,
      "outputValues" : {
         "SuccessMessage" : "Your custom notification is processed successfully."
      }
   }
]
```

Email Alert Actions

Send emails from flows by reusing already-configured workflow email alerts.

The email alert is already configured with the email's contents, recipients, and sender, so the flow only needs to provide the record ID. Email alerts are entity-specific. For more information about creating email alerts, see Creating Email Alerts for Workflow, Approvals, or Milestones in the Salesforce Help. Make sure to review the daily limits for emails sent from email alerts.

This object is available in API version 32.0 and later.

Supported REST HTTP Methods

URI

Get a list of available email alert actions:

`/vXX.X/actions/custom/emailAlert`

Get information about a specific email alert action:

`/vXX.X/actions/custom/emailAlert/entity_name/action_name`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

The email alert action uses the record specified by `SObjectRowId` to get the information it needs. For example, if a Case was specified for the action, the action could retrieve the email address and recipient's name from the Case object's *`SuppliedEmail`* and *`SuppliedName`* fields, respectively.

Input	Details
SObjectRowId	**Type** ID **Description** Required. The ID of a record such as an Account.

Outputs

None.

Flow Actions

Invoke an active autolaunched flow or active invocable process that exists in the current org.

For more information about creating flows and processes, see Build a Flow and Create a Process in the Salesforce Help.

This object is available for autolaunched flows in API version 32.0 and later. This object is available for invocable processes in API version 38.0 and later.

Supported REST HTTP Methods

URI

Get a list of available flow actions:

`/vXX.X/actions/custom/flow`

Invokes the `LargeOrder` flow:

`/vXX.X/actions/custom/flow/LargeOrder`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input values vary according to the input variables specified for each flow. For autolaunched flows, the input values vary according to the input variables in that flow.

Invocable processes always require either one of the following input parameters:

- sObjectId: The Id of the sObject record that you want the process to execute on. The record must be of the same object type as the one on which the process is defined.
- sObject: The sObject itself that you want the process to execute on. The sObject must be of the same object type as the one on which the process is defined.

Outputs

Output values vary according to the output variables specified. `Flow__InterviewStatus` is available for flows. Invocable processes do not have outputs.

Output	Details
`Flow__InterviewStatus`	**Type** picklist **Description** The current status of the flow interview. Valid values are: • Created • Started • Finished • Error • Waiting

Generate Work Orders Actions

Generates work orders from a maintenance plan.

This object is available in API version 40.0 and later.

Supported REST HTTP Methods

URI

`/vXX.X/actions/standard/generateWorkOrders`

Formats

JSON, XML

HTTP Methods

POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`recordId`	**Type** reference **Description** The ID of the maintenance plan from which you want to generate work orders.

Knowledge Actions

Manage your Knowledge articles using invocable actions.

The Assign and Publish actions are available in API version 44.0 and later. All the other actions are available in API version 45.0 and later.

Lightning Knowledge must be set up in your org. The user must have permissions to manage articles.

Tip: You can use multiple inputs to an invocable action. This technique is useful for actions that don't take lists, such as `restoreKnowledgeArticleVersion`.

Supported REST HTTP Methods

URIs

Archive Knowledge articles:

`/vXX.X/actions/standard/archiveKnowledgeArticles`

Assign Knowledge articles:

`/vXX.X/actions/standard/assignKnowledgeArticles`

Create draft from online Knowledge articles:

`/vXX.X/actions/standard/createDraftFromOnlineKnowledgeArticle`

Delete Knowledge articles:

`/vXX.X/actions/standard/deleteKnowledgeArticles`

Publish Knowledge articles:

`/vXX.X/actions/standard/publishKnowledgeArticles`

Restore Knowledge article version:

`/vXX.X/actions/standard/restoreKnowledgeArticleVersion`

Submit Knowledge article for translation:

`/vXX.X/actions/standard/submitKnowledgeArticleForTranslation`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Other Information

Error Response Types on page 24

Archive Knowledge Articles

URI: `/vXX.X/actions/standard/archiveKnowledgeArticles`

Table 1: Inputs

Input	Details
articleVersionIdList	**Type** string **Description** Required. Comma-separated article version ID list.

Sample Input

The following code sample archives two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "archiveKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

The following code sample illustrates a response with one success and one failure:

```
[ {
  "actionName" : "archiveKnowledgeArticles",
  "errors" : null,
  "isSuccess" : false,
  "outputValues" : {
    "ka0RM00000004Ve" : "You can't perform this action. Be sure the action is valid for
 the current state of the article, and that you have permission to perform it.",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Assign Knowledge Articles

URI: `/vXX.X/actions/standard/assignKnowledgeArticles`

Table 2: Inputs

Input	Details
`articleVersionIdList`	**Type** string **Description** Required. Comma-separated article version ID list.
`assigneeId`	**Type** ID **Description** Required. ID of the assigned user.
`assignAction`	**Type** string **Description** Required. Assign action. Valid actions are: • ASSIGN_DRAFT_MASTER • ASSIGN_DRAFT_TRANSLATION

Input	Details
dueDate	**Type** string **Description** Optional. Assigned due date.
instruction	**Type** string **Description** Optional. Instructions for the assignee.
sendEmailNotification	**Type** boolean **Description** Optional. Indicates whether to send an email notification. Defaults to `false`.

Sample Input

The following code sample assigns two articles for translation:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
      "assigneeId" : "005RM00000AAAAAYA4",
      "assignAction" : "ASSIGN_DRAFT_TRANSLATION"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "assignKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Create Draft from Online Knowledge Article

URI: `/vXX.X/actions/standard/createDraftFromOnlineKnowledgeArticle`

Table 3: Inputs

Input	Details
`action`	**Type** string **Description** Required. Edit action for master language or translation articles. Valid actions are: • EDIT_AS_DRAFT_ARTICLE • EDIT_AS_DRAFT_TRANSLATION
`unpublish`	**Type** boolean **Description** Required. Indicates whether to keep the article published (`false`) or archive the published article (`true`). Use `false` to keep the current article version online and create a new draft. Use `true` to archive the current online version, which removes it from the knowledge base, and creates a new draft.
`articleVersionId`	**Type** string **Description** Article version ID. Required to create a draft from an online (published) translation. Optional to create a draft from the online master article if the Article ID is provided.
`articleId`	**Type** string **Description** Article ID. Required when creating a draft from the online (published) master article if the Article Version ID is not provided.

Sample Input

The following code sample creates a draft from a master article and archives the original article:

```
{
  "inputs" : [
    {
      "action" : "EDIT_AS_DRAFT_ARTICLE",
      "unpublish" : true,
      "articleId" : "kA0RM00000004pP0AQ"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "createDraftFromOnlineKnowledgeArticle",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "kA0RM00000004pP0AQ" : "Success"
  }
} ]
```

Delete Knowledge Articles

URI: `/vXX.X/actions/standard/deleteKnowledgeArticles`

Table 4: Inputs

Input	Details
`articleVersionIdList`	**Type** string **Description** Required. Comma-separated article version ID list.

Sample Input

The following code sample deletes two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ]
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "deleteKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
    "ka0RM00000003do" : "Success"
  }
} ]
```

Publish Knowledge Articles

URI: `/vXX.X/actions/standard/publishKnowledgeArticles`

Table 5: Inputs

Input	Details
`articleVersionIdList`	**Type** string **Description** Required. Comma-separated article version ID list.
`pubAction`	**Type** string **Description** Required. Publish action. Valid actions are: • PUBLISH_ARTICLE (which replaces the latest version) • PUBLISH_ARTICLE_NEW_VERSION (which creates a new version) • SCHEDULE_ARTICLE_FOR_PUBLICATION • PUBLISH_TRANSLATION
`pubDate`	**Type** string **Description** Optional. Scheduled publish date.

Sample Input

The following code sample publishes two articles:

```
{
  "inputs" : [
    {
      "articleVersionIdList" : [ "ka0RM00000004VeYAI", "ka0RM00000003doYAA" ],
      "pubAction" : "PUBLISH_ARTICLE"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "publishKnowledgeArticles",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "ka0RM00000004Ve" : "Success",
```

```
    "ka0RM00000003do" : "Success"
  }
} ]
```

Restore Knowledge Article Version

URI: `/vXX.X/actions/standard/restoreKnowledgeArticleVersion`

Table 6: Inputs

Input	Details
`action`	**Type** string **Description** Required. The only valid action is: RESTORE_KNOWLEDGE_ARTICLE_VERSION
`articleId`	**Type** string **Description** Required. Article ID.
`versionNumber`	**Type** integer **Description** Optional. Version number of the archived article version to restore. Default is the latest archived version.

Sample Input

The following code restores the latest archived version:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "articleId" : "kA0RM00000004pP0AQ"
    }
  ]
}
```

The following code restores a past archived version of a published article:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "versionNumber":3,
      "articleId" : "kA0RM00000004pP0AQ"
    }
```

```
  ]
}
```

The following code restores two archived articles:

```
{
  "inputs" : [
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "articleId" : "kA0RM00000004pP0AQ"
    },
    {
      "action" : "RESTORE_KNOWLEDGE_ARTICLE_VERSION",
      "articleId" : "kA0RM00000004pP0AB"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "restoreKnowledgeArticleVersion",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "kA0RM00000004pP0AQ" : "Success"
  }
} ]
```

Submit Knowledge Article for Translation

URI: `/vXX.X/actions/standard/submitKnowledgeArticleForTranslation`

Table 7: Inputs

Input	Details
`articleId`	**Type** string **Description** Required. Article ID.
`language`	**Type** string **Description** Required. Language code for the translation.
`assigneeId`	**Type** ID

Input	Details
	Description Required. ID of the assigned user.
`dueDate`	**Type** string **Description** Optional. Assigned due date.
`sendEmailNotification`	**Type** boolean **Description** Optional. Indicates whether to send an email notification. Defaults to `false`.

Table 8: Outputs

Output	Details
`articleId`	**Type** ID **Description** Article ID.
`language`	**Type** string **Description** Language code for the translation.

Sample Input

The following code sample submits one article for translation into Spanish:

```
{
  "inputs" : [
    {
      "articleId" : "kA0RM00000004pP0AQ",
      "language" : "es",
      "assigneeId" : "005RM00000AAAAAYA4"
    }
  ]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[ {
  "actionName" : "submitKnowledgeArticleForTranslation",
  "errors" : null,
  "isSuccess" : true,
  "outputValues" : {
    "articleId" : "kA0RM00000004pP0AQ",
    "language" : "es"
  }
} ]
```

Error Response Types

Knowledge actions can respond with two types of error responses: action-scoped errors and item-scoped errors.

Action-scoped errors describe an error about the overall action that you're trying to invoke. Action-scoped errors have a `statusCode` in addition to a `message`. This example illustrates an action-scoped error caused by sending invalid input values.

```
[ {
         "actionName" : "restoreKnowledgeArticleVersion",
         "errors" : [ {
         "statusCode" : "INVALID_API_INPUT",
         "message" : "You can't perform this action. Be sure the action is valid for the
current state of the article, and that you have permission to perform it.",
         "fields" : [ ]
         } ],
         "isSuccess" : false,
         "outputValues" : null
         } ]
```

Item-scoped errors describe a problem with a specific article or article version within the action. For example, this code illustrates an archiveKnowledgeArticles action response with one failed item and one successful item.

```
[ {
         "actionName" : "archiveKnowledgeArticles",
         "errors" : null,
         "isSuccess" : false,
         "outputValues" : {
         "ka0RM00000004Ve" : "You can't perform this action. Be sure the action is valid
for the current state of the article, and that you have permission to perform it.",
         "ka0RM00000003do" : "Success"
         }
         } ]
```

If any type of error occurs with an action, the `isSuccess` field is `false`.

Omni-Channel Action

Create a `PendingServiceRouting` record used for Omni-Channel skills-based routing.

For more information about skills-based routing, see Skills-Based Routing for Omni-Channel in Salesforce Help.

This object is available in API version 44.0 and later.

Supported REST HTTP Methods

URI
/v***XX.X***/actions/standard/skillsBasedRouting

Formats
JSON, XML

HTTP Methods
POST

Authentication
Authorization: Bearer ***token***

Inputs

Input	Details
recordId	**Type** ID **Description** Required. ID of the Salesforce record to be routed by Omni-Channel.
routingConfigId	**Type** ID **Description** Required. ID of the QueueRoutingConfig record to be used by Omni-Channel.
skillIdList	**Type** string **Description** Optional. Comma-separated list of Skill IDs. Maximum number of skills is 25.

Outputs

Output	Details
pendingServiceRoutingId	**Type** ID **Description** ID of the new PendingServiceRouting record, if the request was successful.

Usage

Sample Input

The following code sample attempts to create a `PendingServiceRouting` record using a work record (`recordId`), the routing configuration (`routingConfigId`), and two skills (`skillIdList`).

```
{
  "inputs": [{
      "recordId":"400B0000004GGUUIA4",
      "routingConfigId":"0K8B0000000CbgZKAS",
      "skillIdList":"0C4B00000008QImKAM,0C4B0000000CcR1KAK"
  }]
}
```

Sample Output

The following code sample illustrates a response after a successful request.

```
[
  {
    "actionName":"skillsBasedRouting",
    "errors":null,
    "isSuccess":true,
    "outputValues":{
      "pendingServiceRoutingId":"0JRB0000000TWA2"
    }
  }
]
```

PlatformAction

PlatformAction is a virtual read-only object. It enables you to query for actions displayed in the UI, given a user, a context, device format, and a record ID. Examples include standard and custom buttons, quick actions, and productivity actions.

Supported Calls

`describeSObjects()`, `query()`

Fields

Field	Details
`ActionListContext`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Required. The list context this action applies to. Valid values are: • `ActionDefinition`

Field	Details
	• `Assistant` • `BannerPhoto` • `Chatter` • `Dockable` • `FeedElement` • `Flexipage` • `Global` • `ListView` • `ListViewDefinition` • `ListViewRecord` • `Lookup` • `MruList` • `MruRow` • `ObjectHomeChart` • `Photo` • `Record` • `RecordEdit` • `RelatedList` • `RelatedListRecord`
`ActionTarget`	**Type** textarea **Properties** Nillable **Description** The URL to invoke or describe the action when the action is invoked. If the action is a standard button overridden by a Visualforce page, the ActionTarget returns the URL of the Visualforce page, such as `/apex/`*`pagename`*. This field is available in API version 35.0 and later.
`ActionTargetType`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of the target when this action is triggered. Valid values are: • `Describe`—applies to actions with a user interface, such as quick actions • `Invoke`—applies to actions with no user interface, such as action links or invocable actions • `Visualforce`—applies to standard buttons overridden by a Visualforce page

Field	Details
ActionTargetUrl	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** URL to invoke or describe the action when the action is invoked. This field is deprecated in API version 35.0 and later. Use `ActionTarget` instead.
Category	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Applies only to action links. Denotes whether the action link shows up in the feed item list of actions or the overflow list of actions. Valid values are: • `Primary` • `Overflow`
ConfirmationMessage	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Applies only to action links. The message to display before the action is invoked. Field is null if no confirmation is required before invoking the action.
DeviceFormat	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort **Description** Specifies which action icon the PlatformAction query returns. If this field isn't specified, it defaults to Phone. Valid values are: • `Aloha` • `Desktop` • `Phone` • `Tablet`
ExternalId	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** The unique ID for the PlatformAction. If the action doesn’t have an ID, its API name is used.
`GroupId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The unique ID of a group of action links.
`IconContentType`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The content type—such as .jpg, .gif, or .png—of the icon for this action. Applies to both custom and standard icons assigned to actions.
`IconHeight`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The height of the icon for this action. Applies only to standard icons.
`IconUrl`	**Type** url **Properties** Filter, Group, Nillable, Sort **Description** The URL of the icon for this action.
`IconWidth`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The width of the icon for this action. Applies only to standard icons.

Field	Details
InvocationStatus	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The status of the action within the feed item. Applies to action links only. Valid values are: • Failed • New • Pending • Successful
InvokedByUserId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user who most recently invoked this action within the current feed item. Applies to action links only.
IsGroupDefault	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Denotes whether this action is the default in an action link group. False for other action types. Applies to action links only.
IsMassAction	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the action can be performed on multiple records. This field is available in API version 38.0 and later.
Label	**Type** string **Properties** Filter, Group, Sort **Description** The label to display for this action.

Field	Details
`PrimaryColor`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The primary color of the icon for this action.
`RelatedListRecordId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Represents the ID of a record in an object's related list. This field is available in API version 38.0 and later.
`RelatedSourceEntity`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** When the `ActionListContext` is RelatedList or RelatedListRecord, this field represents the API name of the related list to which the action belongs.
`Section`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The section of the user interface the action resides in. Applicable only to Lightning Experience. Valid values are: • ActivityComposer • CollaborateComposer • NotesComposer • Page • SingleActionLinks This field is available in API version 35.0 and later.
`SourceEntity`	**Type** string **Properties** Filter, Group, Sort

Field	Details
	Description Required. The object or record with which this action is associated.
`Subtype`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The subtype of the action. For quick actions, the subtype is `QuickActionType`. For custom buttons, the subtype is `WebLinkTypeEnum`. For action links, subtypes are `Api`, `ApiAsync`, `Download`, and `Ui`. Standard buttons and productivity actions have no subtype.
`TargetObject`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The type of object record the action creates, such as a contact or opportunity. This field is available in API version 41.0 and later.
`TargetUrl`	**Type** string **Properties** Filter, Nillable, Sort **Description** The URL that a custom button or link points to. This field is available in API version 41.0 and later.
`Type`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The type of the action. Valid values are: • `ActionLink`—An indicator on a feed element that targets an API, a web page, or a file, represented by a button in the Salesforce Chatter feed UI. • `CustomButton`—When clicked, opens a URL or a Visualforce page in a window or executes JavaScript. • `InvocableAction`

Field	Details
	• `ProductivityAction`—Productivity actions are predefined and attached to a limited set of objects. Productivity actions include Send Email, Call, Map, View Website, and Read News. Except for the Call action, you can't edit productivity actions. • `QuickAction`—A global or object-specific action. • `StandardButton`—A predefined Salesforce button such as New, Edit, and Delete.

Usage

PlatformAction can be described using describeSObject().

You can directly query for PlatformAction. For example, this query returns all fields for actions associated with each of the records of the listed objects:

```
SELECT ExternalId, ActionTargetType, ActionTargetUrl, ApiName, Category,
      ConfirmationMessage, ExternalId, GroupId, UiTheme, IconUrl, IconContentType,
      IconHeight, IconWidth, PrimaryColor, InvocationStatus, InvokedByUserId,
      IsGroupDefault, Label, LastModifiedDate, Subtype, SourceEntity, Type
FROM PlatformAction
WHERE SourceEntity IN ('001xx000003DGsH', '001xx000003DHBq', 'Task') AND
      ActionListContext = 'Record';
```

Note: To query PlatformAction, provide the `ActionListContext` and `SourceEntity`. If you query for `ActionListContext` with a value of `RelatedList`, and don't specify a `RelatedSourceEntity`, the query returns the API name of the related list. In API v43.0 and before, `SourceEntity = '`***`Object API Name`***`' and ActionListContext = 'ListView'` is an invalid combination to fetch quick actions in a SOQL query. Use `SourceEntity = '`***`Object ID`***`' and ActionListContext = 'ListView'` instead.

This query uses multiple `ActionListContext` values in its `WHERE` clause to return all actions in the Lightning Experience user interface (`DeviceFormat = 'Desktop'`) for the specified object:

```
SELECT ActionListContext, Label, Type, Subtype, Section, SourceEntity,
     RelatedSourceEntity, ActionTarget, ActionTargetType, ApiName, Category,
     ConfirmationMessage, DeviceFormat, ExternalId, GroupId, IconContentType,
     IconHeight, IconUrl, IconWidth, Id, InvocationStatus, InvokedByUserId,
     IsGroupDefault, LastModifiedDate, PrimaryColor
FROM PlatformAction
WHERE ActionListContext IN ('Record','Chatter','RelatedList') AND
      SourceEntity = '001xx000003DlvX' AND
      DeviceFormat = 'Desktop'
```

Post to Chatter Actions

Post to the feed for a specific record, user, or Chatter group.

Use a Post to Chatter action to post a message at run time to a specified feed. Post to Chatter supports @mentions and topics, but only text posts are supported.

This object is available in API version 32.0 and later.

Supported REST HTTP Methods

URI

Get a list of available post to Chatter actions:

`/vXX.X/actions/standard/chatterPost`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`communityId`	**Type** reference **Description** Optional. Specifies the ID of an Experience Cloud site to post to. Valid only if Digital Experiences is enabled. Required if posting to a user or Chatter group that belongs to an Experience Cloud site. This value is available in API version 35.0 and later.
`subjectNameOrId`	**Type** string **Description** Required. Reference to the user, Chatter group, or record whose feed you want to post to. • To post to a user's feed, enter the user's `ID` or `Username`. For example: *`jsmith@salesforce.com`* • To post to a Chatter group, enter the group's `Name` or `ID`. For example: *`Entire Organization`* • To post to a record, enter the record's `ID`. For example: *`001D000000JWBDx`*
`text`	**Type** string **Description** Required. The text that you want to post. Must be a string of no more than 10,000 characters. To mention a user or group, enter @[`reference`], where `reference` is the ID for the user or group that you want to mention. The reference can be a literal value, a merge field, or a flow resource. To add a topic, enter #[`string`], where `string` is the topic that you want to add.

Input	Details
	For example, the string `Hi @[005000000000001] check this out #[some_topic].` is stored appropriately as `Hi @Joe, check this out #some_topic.` where "@Joe" and "#some_topic" are links to the user and topic, respectively.
`type`	**Type** picklist **Description** Required only if `subjectNameOrId` is set to a user name or a Chatter group name. The type of feed that you want to post to. • *`User`*—Enter this value if `subjectNameOrId` is set to a user's `Username`. • *`Group`*—Enter this value if `subjectNameOrId` is set to a Chatter group's `Name`.
`visibility`	**Type** picklist **Description** Optional. Valid only if Digital Experiences is enabled. Specifies whether this feed item is available to all users or internal users only. Valid values are: • *`allUsers`* • *`internalUsers`* This value is available in API version 35.0 and later.

Outputs

Output	Details
`feedItemId`	**Type** reference **Description** The ID of the new Chatter feed item.

Quick Actions

Use a quick action to create a task or a case. Invoke existing quick actions, both global and object-specific, to create records, update records, or log calls.

For more information about creating global quick actions, see Create Global Quick Actions, and for more information on object-specific quick actions, see Create Object-Specific Quick Actions, in the Salesforce online help.

This object is available in API version 32.0 and later.

Supported REST HTTP Methods

URI

Get a list of quick actions:

`/vXX.X/actions/custom/quickAction`

Get a specific quick action:

`/vXX.X/actions/custom/quickAction/quick_action_name`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Notes

The only type of quick actions that can be invoked are create, update, and logACall.

Inputs

All quick actions have the `contextId` input parameter. It's a reference to the related record for the quick action. Other inputs vary according to the layout of the quick action. To determine inputs for a specific quick action, use the describe feature. For example, perform a GET with `/services/data/vXX.X/actions/custom/quickAction/Task/deferTask` to see the inputs for the quick action `deferTask`.

Refresh Metric Actions

Update a metric's Current Value field if it's linked to a summary field in a Salesforce report. The refresh runs as the metric owner.

This object is available in API version 34.0 and later.

Supported REST HTTP Methods

URI

Get a list of metric refresh actions:

`/vXX.X/actions/standard/metricRefresh`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`metricId`	**Type** string **Description** Required. The metric linked to a Salesforce report.

Outputs

Output	Details
`resultingMessage`	**Type** string **Description** The message that indicates the results of the metric refresh.

Salesforce Order Management Actions

Manage, fulfill, and service orders in flows with Salesforce Order Management.

For more information about using Order Management actions in flows, see Salesforce Order Management Flow Core Actions in Salesforce Help.

These actions are available in API version 48.0 and later.

Your org must have a Salesforce Order Management license.

Supported REST HTTP Methods

URI

Get a specific Order Management action:

`/vXX.X/actions/standard/om_action_name`

Formats

JSON, XML

HTTP Methods

GET

Authentication

`Authorization: Bearer token`

Notes

Order Management actions are intended for use in flows. The standard actions URI can only be used to get information about them. To use these actions in code, call the corresponding Connect REST API endpoints or Apex ConnectApi methods. For more information, see Order Management Resources in the *Connect REST API Developer Guide* and ConnectApi Namespace in the *Apex Developer Guide*.

In flows, Order Management action inputs and outputs use Apex-defined variables that map to input and output classes in the ConnectApi namespace.

Session-Based Permission Set Actions

Activate or deactivate a session-based permission set for the current user's API session.

This object is available in API version 40.0 and later.

This action activates or deactivates the provided permission set for the current user's API session. The activation or deactivation doesn't affect other sessions. The permission set must already be assigned to the current user.

Supported REST HTTP Methods

URI

Activate session-based permission set:

`/vXX.X/actions/standard/activateSessionPermSet`

Deactivate session-based permission set:

`/vXX.X/actions/standard/deactivateSessionPermSet`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`PermSetName`	**Type** string **Description** Required. Specifies the developer name of the permission set.
`PermSetNamespace`	**Type** string **Description** Optional. Specifies the namespace of the permission set.

Outputs

None.

Simple Email Actions

Send an email where you specify the subject, body, and recipients.

This object is available in API version 32.0 and later.

Each licensed org can send single emails to a maximum of 5,000 external email addresses per day based on Greenwich Mean Time (GMT). For orgs created before Spring '19, the daily limit is enforced only for emails sent via Apex and Salesforce APIs except for the REST API. For orgs created in Spring '19 and later, the daily limit is also enforced for email alerts, simple email actions, Send Email actions in flows, and REST API. If one of the newly counted emails can't be sent because your org has reached the limit, we notify you by email and add an entry to the debug logs. Single emails sent using the email author or composer in Salesforce don't count toward this limit. There's no limit on sending single emails to contacts, leads, person accounts, and users in your org directly from account, contact, lead, opportunity, case, campaign, or custom object pages. In Developer Edition orgs and orgs evaluating Salesforce during a trial period, you can send to a maximum of 50 recipients per day, and each single email can have up to 15 recipients.

Supported REST HTTP Methods

URI

Get a list of available simple email actions:

`/vXX.X/actions/standard/emailSimple`

Formats

JSON, XML

HTTP Methods

GET, HEAD, POST

Authentication

`Authorization: Bearer token`

Inputs

Input	Details
`emailAddresses`	**Type** string **Description** Required if `emailAddressArray` isn't used. The email recipients specified as a comma-separated list. The number of email recipients must be five or less. If `emailAddressArray` is also used, the combined number of recipients must be five or less.
`emailAddressesArray`	**Type** string **Description** Required if `emailAddress` isn't used. Up to five email recipients, specified as a collection of strings. If `emailAddress` is also used, the combined number of recipients must be five or less.

Input	Details
`emailBody`	**Type** textarea **Description** Required. The body of the email in plain text.
`emailSubject`	**Type** string **Description** Required. The email's subject.
`senderAddress`	**Type** string **Description** Optional. The organization-wide email address to be used as the sender. Required only if `senderType` is set to `OrgWideEmailAddress`.
`senderType`	**Type** string **Description** Optional. Email address used as the email's From and Reply-To addresses. Valid values are: • `CurrentUser`—Email address of the user running the flow. This is the default setting. • `DefaultWorkflowUser`—Email address of the default workflow user. • `OrgWideEmailAddress`—The organization-wide email address that is specified in `senderAddress`.

Outputs

None.

Submit for Approval Actions

Submit a Salesforce record for approval if an approval process is defined for the current entity.

For more information about creating submit for approval actions, see Creating Approval Actions, and to learn more about approval processes, see Approval Process Overview, in the Salesforce online help.

This object is available in API version 32.0 and later.

Supported REST HTTP Methods

URI

Get a list of actions:

`/vXX.X/actions/standard/submit`

Formats
JSON, XML

HTTP Methods
GET, HEAD, POST

Authentication
`Authorization: Bearer token`

Inputs

Input	Details
`comment`	**Type** string **Description** Optional. Comments that accompany the Approval submission.
`nextApproverIds`	**Type** reference **Description** Optional. An array of one ID of the next approver, which can be a user or a group. This input is optional because some approval processes have the next approver specified in the approval process definition. You can't specify more than 2,000 approvers.
`objectId`	**Type** reference **Description** Required. The ID of the record being submitted for approval.
`processDefinitionNameOrId`	**Type** string **Description** Optional. The ID or name of the approval process to run against the record. If none is specified, the first approval process whose entry criteria the record satisfies is used. Names can't be longer than 120 characters.
`skipEntryCriteria`	**Type** boolean **Description** Optional. A Boolean value controlling whether the entry criteria for the specified approval process should be evaluated for the record (true) or not (false). If set to true, also specify `processDefinitionNameOrId`.

Input	Details
`submitterId`	**Type** string **Description** Optional. The ID of the user submitting the record for approval. If none is specified, the submitter is the current user.

Outputs

Output	Details
`actorIds`	**Type** reference **Description** An array of the IDs of the next approvers.
`entityId`	**Type** reference **Description** The ID of the record submitted for approval.
`instanceId`	**Type** reference **Description** The ID of the approval process instance.
`instanceStatus`	**Type** string **Description** The status of the approval. The valid values are • Approved • Pending • Rejected • Removed
`newWorkItemIds`	**Type** reference **Description** An array of the IDs of the work items created for the next step in this approval process.