Force.com Tooling API

Reference and Developer Guide

Version 36.0, Spring '16

@salesforcedocs

Last updated: February 17, 2016

CONTENTS

CHAPTER 1 Introducing Tooling API

Use Tooling API to build custom development tools or apps for Force.com applications. Tooling API's SOQL capabilities for many metadata types allow you to retrieve smaller pieces of metadata. Smaller retrieves improve performance, which makes Tooling API a better fit for developing interactive applications.Tooling API provides SOAP and REST interfaces.

For example, you can:

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

Tooling API exposes metadata used in developer tooling that you can access through REST or SOAP.

For detailed descriptions of Tooling API objects and the REST resources and SOAP calls that each object supports, see Tooling API Objects.

When to Use Tooling API

Use Tooling API when you need fine-grained access to an org's metadata. Tooling API's SOQL capabilities for many metadata types allow you to retrieve smaller pieces of metadata. Smaller retrieves improve performance, which makes Tooling API a better fit for developing interactive applications.

Because Tooling API allows you to change just one element within a complex type, it can be easier to use than Metadata API. Other use cases include:

- Source control integration
- Continuous integration
- Apex classes or trigger deployment

Some of the specific tasks you can accomplish using the Tooling API:

Retrieve metadata about an object's field

Use FieldDefinition.

Retrieve custom or standard object properties

Use FieldDefinition.

Manage working copies of Apex classes and triggers and Visualforce pages and components.
Use ApexClassMember, ApexTriggerMember, ApexPageMember, ApexComponentMember, and MetadataContainer .

Manage working copies of static resource files.
Use StaticResource.

Check for updates and errors in working copies of Apex classes and triggers and Visualforce pages and components.
ContainerAsyncRequest

Commit changes to your organization.
Use ContainerAsyncRequest.

Set heap dump markers.
Use ApexExecutionOverlayAction

Overlay Apex code or SOQL statements on an Apex execution.
Use ApexExecutionOverlayAction.

Execute anonymous Apex.
For sample code, see SOAP Calls and REST Overview.

Generate log files for yourself or for other users.
Set checkpoints with TraceFlag

Access debug log and heap dump files.
Use ApexLog and ApexExecutionOverlayResult.

Manage custom fields on custom objects.
Use CustomField.

Access code coverage results.
Use ApexCodeCoverage, ApexOrgWideCoverage, and ApexCodeCoverageAggregate.

Execute tests, and manage test results.
Use ApexTestQueueItem and ApexTestResult.

Manage validation rules and workflow rules.
Use ValidationRule and WorkflowRule.

REST Overview

Use REST if you're using a language that isn't strongly typed, like JavaScript.

For details on usage, syntax, and authentication, see the *Force.com REST API Developer Guide*.

IN THIS SECTION:

REST Resources

REST Resources give you access to Tooling objects.

For details on usage, syntax, and authentication, see the *Force.com REST API Developer Guide*.

For examples, see REST Resource Examples.

REST Resources

This section lists supported REST resources in Tooling API.

The base URI for each Tooling REST API resource is `http://`***`domain`***`/services/data/`***`vXX.X`***`/tooling/` where *`domain`* is a Salesforce instance or a custom domain and *`vXX.X`* is the API version number. For example: `http://na1.salesforce.com/services/data/v35.0/tooling/`

Like the Force.com REST API, Tooling API uses the following resources.

`/completions?type=`

Supported methods: GET

Retrieves available code completions of the referenced type for Apex system method symbols (`type=`**`apex`**). Available from API version 28.0 or later.

`/executeAnonymous/?anonymousBody= <url encoded body>`

Supported methods: GET

Executes Apex code anonymously. Available from API version 29.0 or later.

`/query/?q=`*`SOQL_Query_Statement`*

Supported methods: GET

Executes a query against an object and returns data that matches the specified criteria. Tooling API exposes objects like EntityDefinition and FieldDefinition that use the external object framework--that is, they don’t exist in the database but are constructed dynamically. Special query rules apply to virtual entities.

If the query result is too large, it’s broken up into batches. The response contains the first batch of results and a query identifier. The identifier can be used in a request to retrieve the next batch.

`/runTestsAsynchronous/?classids= <comma separated list of class IDs>`

Supported methods: GET

Executes the tests in the specified classes. Running tests asynchronously allows methods to process in parallel, cutting down your test run times.

`/runTestsAsynchronous/ Body: {"classids":"<comma-separated list of class IDs>", "suiteids":"<comma-separated list of test suite IDs>", "maxFailedTests":"<integer value>"}`

Supported methods: POST

Runs one or more methods within one or more Apex classes, using the asynchronous test execution mechanism.

You can POST both a `suiteids` list and a `classids` list to `runTestsAsynchronous`. However, if you send a `tests` array, you can’t send `suiteids` or `classids`.

You can also POST an optional `maxFailedTests` parameter. To allow all tests in your org to run, regardless of how many tests fail, omit `maxFailedTests` or set it to `-1`. To stop the test run from executing new tests after a given number of tests fail, set `maxFailedTests` to an integer value from `0` to `1,000,000`.This integer value sets the maximum allowable test failures. A value of `0` causes the test run to stop if any failure occurs. A value of `1` causes the test run to stop on the second failure, and so on.

Keep in mind that high values can cause slow performance. Each 1,000 tests that you add to your maxFailedTests value add about 3 seconds to your test run, not including the time that the tests take to execute.

/runTestsAsynchronous/ Body: {"tests":<tests array>}

Supported methods: POST

Runs one or more methods within one or more Apex classes, using the asynchronous test execution mechanism.

<tests array> is an array of objects that represent Apex test classes—each of which has a classId and a testMethods parameter—and an optional maxFailedTests parameter.

Multiple occurrences of a test method name in a testMethods array are ignored. Test methods that don't exist are skipped. A null or missing testMethods array specifies that all test methods in the test class are run.

To allow all tests in your org to run, regardless of how many tests fail, omit maxFailedTests or set it to -1. To stop the test run from executing new tests after a given number of tests fail, set maxFailedTests to an integer value from 0 to 1,000,000.This integer value sets the maximum allowable test failures. A value of 0 causes the test run to stop if any failure occurs. A value of 1 causes the test run to stop on the second failure, and so on. Keep in mind that high values can cause slow performance. Each 1,000 tests that you add to your maxFailedTests value add about 3 seconds to your test run, not including the time that the tests take to execute.

Example <tests array>:

```
[{
"classId" : "01pD0000000Fhy9IAC",
"testMethods" : ["testMethod1","testMethod2","testMethod3"]
},{
"classId" : "01pD0000000FhyEIAS",
"testMethods" : ["testMethod1","testMethod2"]
},
"maxFailedTests":"2"]
```

/runTestsSynchronous/?classnames= <comma-separated list of class names>

Supported methods: GET

Executes the tests in the specified classes using the synchronous test execution mechanism.

/runTestsSynchronous/ Body: {"tests":<tests array>}

Supported methods: POST

Runs one or more methods within an Apex class, using the synchronous test execution mechanism. All test methods in a synchronous test run must be in the same class.

<tests array> is an array of an object that represents an Apex test class—which has a classId and a testMethods parameter—and an optional maxFailedTests parameter.

Multiple occurrences of a test method name in a testMethods array are ignored. Test methods that don't exist are skipped. A null or missing testMethods array specifies that all test methods in the test class are run.

To allow all tests in your org to run, regardless of how many tests fail, omit maxFailedTests or set it to -1. To stop the test run from executing new tests after a given number of tests fail, set maxFailedTests to an integer value from 0 to 1,000,000.This integer value sets the maximum allowable test failures. A value of 0 causes the test run to stop if any failure occurs. A value of 1 causes the test run to stop on the second failure, and so on. Keep in mind that high values can cause slow performance. Each 1,000 tests that you add to your maxFailedTests value add about 3 seconds to your test run, not including the time that the tests take to execute.

Example `<tests array>`:

```
[{
"classId" : "01pD0000000Fhy9IAC",
"testMethods" : ["testMethod1","testMethod2","testMethod3"]
},
"maxFailedTests":"2"]
```

/search/?q=*SOSL_Search_Statement*

Supported methods: GET

Search for records containing specified values.

/sobjects/

Supported methods: GET

Lists the available Tooling API objects and their metadata.

/sobjects/*SObjectName*/

Supported methods: GET, POST

Describes the individual metadata for the specified object or creates a record for a given object.

- To retrieve the metadata for the ApexExecutionOverlayAction object, use the GET method.
- To create a ApexExecutionOverlayAction object, use the POST method.

/sobjects/*SObjectName*/describe/

Supported methods: GET

Completely describes the individual metadata at all levels for the specified object.

For example, use this resource to retrieve the fields, URLs, and child relationships for a Tooling API object.

/sobjects/*SObjectName*/*id*/

Supported methods: GET, PATCH, DELETE

Accesses records based on the specified object ID.

Use the GET method to retrieve records or fields, the DELETE method to delete records, and the PATCH method to update records.

/sobjects/ApexLog/*id*/Body/

Supported methods: GET

Retrieves a raw debug log by ID. Available from API version 28.0 or later.

REST Resource Examples

Robust examples using REST resources in the Tooling API.

Example Setup

The following examples use Apex to execute REST requests, but you can use any standard REST tool to access the Tooling API.

Note: Salesforce runs on multiple server instances. The examples in this guide use `*instance*.salesforce.com`. Be sure to use your org's instance name.

First, set up the connection to your org and the HTTP request type:

```
HttpRequest req = new HttpRequest();
req.setHeader('Authorization', 'Bearer ' + UserInfo.getSessionID());
req.setHeader('Content-Type', 'application/json');
```

At the end of each request, add the following code to send the request and retrieve the body of the response:

```
Http h = new Http();
HttpResponse res = h.send(req);
system.debug(res.getBody());
```

Retrieve a Description

To get a description of all available objects in Tooling API:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/');
req.setMethod('GET');
```

To get a description of a specific Tooling API object, for example, TraceFlag:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
TraceFlag/');
req.setMethod('GET');
```

To get a description of all metadata for a specific Tooling API object, for example, TraceFlag:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
TraceFlag/describe/');
req.setMethod('GET');
```

Manipulate Objects by ID

To create a new Tooling API object, for example, MetadataContainer:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
MetadataContainer/');
req.setBody('{"Name":"TestContainer"}');
req.setMethod('POST');
```

Tip: Use the ID from this call in the rest of the examples.

To retrieve a Tooling API object by ID, for example, MetadataContainer:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
MetadataContainer/ + containerID + '/');
req.setMethod('GET');
```

To update a Tooling API object by ID, for example, MetadataContainer:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
MetadataContainer/ + containerID + '/');
req.setBody('{"Name":"NewlyNamedContainer"}');
req.setMethod('PATCH');
```

To query a Tooling API object by ID, for example, MetadataContainer:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/query/?q=
Select+id,Name+from+MetadataContainer+Where+ID=\'' + containerID +  '\'');
req.setMethod('GET');
```

Query Within MetadataContainer

To query an object within a MetadataContainer:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/query/?q=
Select+id,Body,LastSyncDate,Metadata+from+ApexClassMember+Where+MetadataContainerID=\'
+ containerID +  '\'');
req.setMethod('GET');
```

Check Deployment Status

To check on the status of a deployment, using ContainerAsyncRequest:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/sobjects/
ContainerAsyncRequest/' + requestID + '/');
req.setMethod('GET');
```

Execute Anonymous Apex

To execute anonymous Apex:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/executeAnonymous/?
anonymousBody=System.debug('Test')%3B');
req.setMethod('GET');
```

Retrieve Apex

To retrieve your Apex classes and triggers, and the global Apex classes and triggers from your installed managed packages:

```
req.setEndpoint('http://instance.salesforce.com/services/data/v36.0/tooling/apexManifest');
req.setMethod('GET');
```

Execute Apex Unit Tests

To execute Apex unit tests, use the `runTestsSynchronous` or `runTestsAsynchronous` resource. This example illustrates how to `POST` to the `runTestsSynchronous` resource using JavaScript. The comment blocks show which objects these calls return.

```
var xhttp = new XMLHttpRequest();
xhttp.open("POST",
"http://instance.salesforce.com/services/data/v36.0/tooling/runTestsSynchronous/", true)

// SESSION_ID is the session ID
xhttp.setRequestHeader("Authorization", "OAuth <SESSION_ID>")
xhttp.setRequestHeader('Accept', "application/json");
```

```
// testObject should include a list of object(s) with the classId and list of
//     desired test methods for the desired classes to be tested
testObject = {tests: [{classId: "N0tARealClassId", testMethods: ["testMethod1",
"testMethod2"]}]}
requestObject = json.stringify(testObject);
response = xhttp.send(requestObject)
response = JSON.parse(response)

/*
{
  "successes": [
    {
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod1",
      "id": "N0tARealTestId1",
      "time": 1167,
      "seeAllData": false
    },
    {
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod2",
      "id": "N0tARealTestId2",
      "time": 47,
      "seeAllData": false
    }
  ],
  "failures": [
    {
      "type": "Class",
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod3",
      "message": "System.AssertException: Assertion Failed",
      "stackTrace": "Class.MyTestClass.testMethod3: line 13, column 1",
      "id": "01pxx0000000JTpAAM",
      "seeAllData": false,
      "time": 27,
      "packageName": "MyTestClass"
    },
    {
      "type": "Class",
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod4",
      "message": "System.AssertException: Assertion Failed",
      "stackTrace": "Class.MyTestClass.testMethod4: line 17, column 1",
      "id": "01pxx0000000JTpAAM",
      "seeAllData": false,
      "time": 32,
      "packageName": "MyTestClass"
    }
  ],
  "totalTime": 143,
  "apexLogId": "07Lxx0000000A9NEAU",
  "numFailures": 2,
  "codeCoverage": [

  ],
  "codeCoverageWarnings": [

  ],
  "numTestsRun": 4
}
*/

// Check how many tests ran
response["numTestRun"] === 4
// Check how many tests passed
response["successes"].length === 2

// Return a list of objects that correspond to the tests that passed
response["successes"]
/*
[
    {
      "id": "N0tARealTestId1",
      "methodName": "testMethod1",
      "name": "MyTestClass",
      "namespace": null,
      "seeAllData": false,
      "time": 1167
    }
  ]
*/

// Access the first object in the list
response["successes"][0]["name"] === "MyTestClass"
response["successes"][0]["methodName"] === "testMethod1"
// This ID refers to the classId
response["successes"][0]["id"] === "MyTestClass"
response["successes"][0]["time"] === 1167 // milliseconds

response["failures"]
/*
 {
      "type": "Class",
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod3",
      "message": "System.AssertException: Assertion Failed",
      "stackTrace": "Class.MyTestClass.testMethod3: line 13, column 1",
      "id": "01pxx0000000JTpAAM",
      "seeAllData": false,
      "time": 27,
      "packageName": "MyTestClass"
    },
    {
      "type": "Class",
      "namespace": null,
      "name": "MyTestClass",
      "methodName": "testMethod4",
      "message": "System.AssertException: Assertion Failed",
      "stackTrace": "Class.MyTestClass.testMethod4: line 17, column 1",
      "id": "01pxx0000000JTpAAM",
      "seeAllData": false,
      "time": 32,
      "packageName": "MyTestClass"
    }
*/

response["failures"][0]["name"] === "MyTestClass"
response["failures"][0]["methodName"] === "testMethod3"
response["failures"][0]["message"] === "System.AssertException: Assertion Failed"
response["failures"][0]["stackTrace"] === "Class.MyTestClass.testMethod3: line 13, column 1"
response["failures"][0]["time"] === 27
```

REST Headers

Use REST if you're using a language that isn't strongly typed, like JavaScript.

For details on usage, syntax, and authentication, see the *Force.com REST API Developer Guide*.

REST headers available in the Tooling API WSDL are described in REST Headers for Tooling API on page 290.

For examples of REST headers being used, see REST Header Examples.

REST Header Examples

Use these examples to understand REST headers.

REST headers in the Tooling API WSDL are described in REST Headers for Tooling API on page 290. For more details about REST Resources, see the *Force.com REST API Developer Guide*.

Examples

The following examples use Apex to execute REST requests with headers. You can use any standard REST tool to access Tooling REST API.

Note: Salesforce runs on multiple server instances. The examples in this guide use the *na1* instance. Be sure to use your organization's instance name.

First, set up the connection to your org and the HTTP request type:

```
HttpRequest req = new HttpRequest();
req.setHeader('Authorization', 'Bearer ' + UserInfo.getSessionID());
req.setHeader('Content-Type', 'application/json');
```

At the end of each request, add the following code to send the request and retrieve the body of the response:

```
Http h = new Http();
HttpResponse res = h.send(req);
system.debug(res.getBody());
```

SOAP Calls

Use SOAP if you're using a strongly typed language like Java that generates Web service client code. For details about usage, syntax, and authentication, see the *SOAP API Developer's Guide*.

To access the Tooling API WSDL, from Setup, enter `API` in the `Quick Find` box, then select **API** and click **Generate Tooling WSDL**.

Like the Salesforce SOAP API, Tooling API uses the following calls.

`create()`
Adds one or more records to your organization's data.

`delete()`
Deletes one or more records from your organization's data.

`describeLayout()`
Retrieve metadata about page layouts for a specified SOjbect.

`describeGlobal()`
Lists the available Tooling API objects and their metadata.

`describeSObjects()`
Describes the metadata (field list and object properties) for the specified object or array of objects.

Call `describeGlobal()` to retrieve a list of all Tooling API objects for your organization, then iterate through the list and use `describeSObjects()` to obtain metadata about individual objects.

`describeValueType()`
Describes the metadata for a specified namespace and value type. For information about `describeValueType`, see the *Metadata API Developer Guide*.

`describeWorkitemActions()`
Describes which actions are available for a specified work item.

`executeanonymous(string apexcode)`
Executes the specified block of Apex anonymously and returns the result.

`query()`
Executes a query against a Tooling API object and returns data that matches the specified criteria.

`queryMore()`
Retrieves the next batch of objects from a `query()`.

`retrieve()`
Retrieves one or more records based on the specified IDs.

`runTests()` and `runTestsAsynchronous()`
Executes test methods in the specified classes. Running tests asynchronously allows methods to process in parallel, cutting down your test run times.

The synchronous `runTests()` call accepts a RunTestsRequest object.

This example shows the structure of a call to a class that calls the `runTestsAsynchronous` endpoint.

```
conn.runTestsAsynchronous(classids, suiteids, maxFailedTests)
```

For more `runTestsAsynchronous()` example code, see ApexTestQueueItem.

Either a `classids` or a `suiteids` parameter is mandatory for `runTestsAsynchronous`, but only one of the two is required. To provide only one, specify the other as `null`.

A value for `maxFailedTests` is mandatory. To allow all tests in your org to run, regardless of how many tests fail, set `maxFailedTests` to `-1`. To stop the test run from executing new tests after a given number of tests fail, set `maxFailedTests` to an integer value from `0` to `1,000,000`. This integer value sets the maximum allowable test failures. A value of `0` causes the test run to stop if any failure occurs. A value of `1` causes the test run to stop on the second failure, and so on. Keep in mind that high values can cause slow performance. Each 1,000 tests that you add to your `maxFailedTests` value add about 3 seconds to your test run, not including the time that the tests take to execute.

`search()`
Search for records that match a specified text string.

`update()`
Updates one or more existing records in your organization’s data.

`upsert()`
Creates records and updates existing records; uses a custom field to determine the presence of existing records.

SOAP Headers

The SOAP headers available in the Tooling API WSDL are described in SOAP Headers for Tooling API.

Examples

These examples use C#, but you can use any language that supports Web services.

To compile Apex classes or triggers in Developer Edition or sandbox organizations, use `create()`. The next sample uses ApexClass to compile a simple class with a single method called `SayHello`.

```
String classBody = "public class Messages {\n"
   + "public string SayHello() {\n"
   + " return 'Hello';\n" + "}\n"
   + "}";

// create an ApexClass object and set the body
ApexClass apexClass = new ApexClass();
apexClass.Body = classBody;
ApexClass[] classes = { apexClass };

// call create() to add the class
SaveResult[] saveResults = sforce.create(classes);
for (int i = 0; i < saveResults.Length; i++)
   {
   if (saveResults[i].success)
      {
        Console.WriteLine("Successfully created Class: " +
         saveResults[i].id);
      }
```

```
     else
        {
           Console.WriteLine("Error: could not create Class ");
           Console.WriteLine("   The error reported was: " +
           saveResults[i].errors[0].message + "\n");
        }
     }
```

The `IsCheckOnly` parameter on ContainerAsyncRequest indicates whether an asynchronous request compiles code but doesn't execute or save it (`true`), or compiles and save the code (`false`).

The next example expands upon the first by modifying the `SayHello()` method to accept a person's first and last name. This example uses MetadataContainer with ApexClassMember to retrieve and update the class, and ContainerAsyncRequest to compile and deploy the changes to the server. You can use the same method with ApexTriggerMember, ApexComponentMember, and ApexPageMember.

Note: To test your code, modify the `IsCheckOnly` parameter in the next sample, and log in to your organization after a successful execution to verify the results.

- When `IsCheckOnly = true`, the `SayHello()` method remains the same. ApexClassMember contains the compiled results, but the class on the server remains the same.
- When `IsCheckOnly = false`, the `SayHello()` method shows the change to accept a person's first and last name.

```
String updatedClassBody = "public class Messages {\n"
   + "public string SayHello(string fName, string lName) {\n"
   + " return 'Hello ' + fName + ' ' + lName;\n" + "}\n"
   + "}";

//create the metadata container object
MetadataContainer Container = new MetadataContainer();
Container.Name = "SampleContainer";

MetadataContainer[] Containers = { Container };
SaveResult[] containerResults = sforce.create(Containers);
if (containerResults[0].success)
{
   String containerId = containerResults[0].id;

   //create the ApexClassMember object
   ApexClassMember classMember = new ApexClassMember();
   //pass in the class ID from the first example
   classMember.ContentEntityId = classId;
   classMember.Body = updatedClassBody;
   //pass the ID of the container created in the first step
   classMember.MetadataContainerId = containerId;
   ApexClassMember[] classMembers = { classMember };

   SaveResult[] MembersResults = sforce.create(classMembers);
   if (MembersResults[0].success)
   {
      //create the ContainerAsyncRequest object
      ContainerAsyncRequest request = new ContainerAsyncRequest();
      //if the code compiled successfully, save the updated class to the server
      //change to IsCheckOnly = true to compile without saving
      request.IsCheckOnly = false;
      request.MetadataContainerId = containerId;
```

```
ContainerAsyncRequest[] requests = { request };
SaveResult[] RequestResults = sforce.create(requests);
if (RequestResults[0].success)
{
   string requestId = RequestResults[0].id;

   //poll the server until the process completes
   QueryResult queryResult = null;
   String soql = "SELECT Id, State, ErrorMsg
                  FROM ContainerAsyncRequest
                  Where id = '" + requestId + "'";
   queryResult = sforce.query(soql);
   if (queryResult.size > 0)
   {
    ContainerAsyncRequest _request = (ContainerAsyncRequest)queryResult.records[0];

      while (_request.State.ToLower() == "queued")
      {
         //pause the process for 2 seconds
         Thread.Sleep(2000);

         //poll the server again for completion
         queryResult = sforce.query(soql);
         _request = (ContainerAsyncRequest)queryResult.records[0];
      }

      //now process the result
      switch (_request.State)
      {
         case "Invalidated":
            break;

         case "Completed":
         //class compiled successfully
         //see the next example on how to process the SymbolTable
            break;

         case "Failed":
      . .   break;

         case "Error":
            break;

         case "Aborted":
            break;

         }
      }
      else
      {
         //no rows returned
      }
   }
   else
```

```
         {
            Console.WriteLine("Error: could not create ContainerAsyncRequest object");
            Console.WriteLine("   The error reported was: " +
            RequestResults[0].errors[0].message + "\n");
         }
      }
      else
      {
         Console.WriteLine("Error: could not create Class Member ");
         Console.WriteLine("   The error reported was: " +
         MembersResults[0].errors[0].message + "\n");
      }
   }
   else
   {
   .. Console.WriteLine("Error: could not create MetadataContainer ");
      Console.WriteLine("   The error reported was: " +
      containerResults[0].errors[0].message + "\n");
   }
}
```

To access Apex class and trigger data in a structured format, use a SymbolTable.

The next sample queries the ApexClassMember object created in the previous example to obtain the SymbolTable of the modified class.

Note: The SOQL statement used depends on when the data is retrieved.

- To execute the query from within the previous sample, use the ID of the ContainerAsyncRequest. For example, `SELECT Body, ContentEntityId, SymbolTable FROM ApexClassMember where MetadataContainerId = '" + requestId + "'"`
- Otherwise, use the ID of the modified class as shown in the next sample. For example, `SELECT ContentEntityId, SymbolTable FROM ApexClassMember where ContentEntityId = '" + classId + "'"`

```
   //use the ID of the class from the previous step
   string classId = "01pA00000036itIIAQ";
   QueryResult queryResult = null;
   String soql = "SELECT ContentEntityId, SymbolTable FROM ApexClassMember where
ContentEntityId = '" + classId + "'";

   queryResult = sforce.query(soql);
   if (queryResult.size > 0)
   {
      ApexClassMember apexClass = (ApexClassMember)queryResult.records[0];
      SymbolTable symbolTable = apexClass.SymbolTable;

      foreach (Method _method in symbolTable.methods)
      {
         //here's the SayHello method
         String _methodName = _method.name;

         //report the modifiers on the method such as global, public, private, or static
         String _methodVisibility = _method.modifiers;

         //get the method's return type
         string _methodReturnType = _method.returnType;
         //get the fName & lName parameters
         foreach (Parameter _parameter in _method.parameters)
         {
            string _paramName = _parameter.name;
            string _parmType = _parameter.type;
         }
      }
   }
   else
   {
      //unable to locate class
   }
```

To add checkpoints to your code for debugging, use ApexExecutionOverlayAction.

This sample adds a checkpoint to the class from the previous samples:

```
//use the ID of the class from the first sample.
string classId = "01pA00000036itIIAQ";

ApexExecutionOverlayAction action = new ApexExecutionOverlayAction();
action.ExecutableEntityId = classId;
action.Line = 3;
action.LineSpecified = true;
action.Iteration = 1;
action.IterationSpecified = true;
ApexExecutionOverlayAction[] actions = { action };

SaveResult[] actionResults = sforce.create(actions);
if (actionResults[0].success)
{
   // checkpoint created successfully
}
else
{
   Console.WriteLine("Error: could not create Checkpoint ");
   Console.WriteLine("   The error reported was: " +
   actionResults[0].errors[0].message + "\n");
}
```

Objects, Namespaces, and Tooling API

Tooling API objects provide programmatic access to data and metadata.

The Tooling API WSDL includes the tooling namespace `tns` (`urn:tooling.soap.sforce.com`) and the metadata namespace `mns` (`urn:metadata.tooling.soap.sforce.com`). Some objects or types in the `mns` namespace occur in both the Metadata API WSDL and the Tooling API WSDL.

- If objects or types are identical in the Tooling API WSDL and Metadata API WSDL, they are documented in the Metadata API Developer Guide.
- If objects or types are different in the Tooling API WSDL, or only occur in the Tooling API WSDL, they are documented here.

To verify the complete list of fields for an object, access the Tooling API WSDL:

1. From Setup, enter `API` in the `Quick Find` box, then select **API**.
2. Select either **Generate Tooling WSDL** or **Generate Tooling WSDL With Strongly Typed Enums**.

Note: Frequently occurring system fields are described in System Fields on page 19.

Object Quick-Reference Tables, System Fields, and SOQL Limitations

Learn about objects, system fields that occur on most objects, and SOQL limitations that apply to some objects in Tooling API. An alphabetical list of objects is also available.

IN THIS SECTION:

SOQL Operation Limitations
Some Tooling API objects have SOQL limitations.

SOSL Operation Limitations
Two Tooling API objects, EntityDefinition and FieldDefinition, have SOSL limitations.

System Fields
Some fields are system-generated. They are on most Tooling API objects, and are read-only.

Programming Objects
Use programming objects to interact with programmatic artifacts: Apex, Visualforce, and Lightning.

Setup Objects
Use setup objects to interact with metadata for declarative development. For example, you may wish to create your own version of Setup, or restrict the amount of data you need to push to an app on a mobile phone.

Tooling Objects
Use these objects to build tools around test results, debugging, code coverage, and more.

Operational Objects
Use the following objects for Tooling API operations.

SOQL Operation Limitations

Some Tooling API objects have SOQL limitations.

The following objects in Tooling API don't support SOQL operations `COUNT()`, `GROUP BY`, `LIMIT`, `LIMIT OFFSET`, `OR`, and `NOT`.

- CompactLayoutInfo
- CompactLayoutItemInfo
- DataType
- EntityDefinition
- EntityLimit
- EntityParticle
- FieldDefinition
- Publisher
- RelationshipDomain
- RelationshipInfo

- SearchLayout
- ServiceFieldDataType
- StandardAction
- UserEntityAccess
- UserFieldAccess

The unsupported operations for these objects return errors or incorrect results as these examples illustrate.

GROUP BY

Example Query: `SELECT COUNT(qualifiedapiname), isfeedenabled FROM EntityDefinition GROUP BY isfeedenabled`

Error Returned: `The requested operation is not yet supported by this SObject storage type, contact salesforce.com support for more information.`

LIMIT, LIMIT OFFSET

Example Queries:

```
SELECT qualifiedapiname FROM EntityDefinition LIMIT 5
```

```
SELECT qualifiedapiname FROM EntityDefinition LIMIT 5 OFFSET 10
```

An incorrect result is returned because LIMIT and LIMIT OFFSET are ignored.

NOT

Example Query: `SELECT qualifiedapiname FROM EntityDefinition WHERE qualifiedapiname!='Account'`

Error Returned: `Only equals comparisons permitted`

OR

Example Query: `SELECT qualifiedapiname, keyprefix FROM EntityDefinition WHERE isdeletable=true OR (isfeedenabled=false AND keyprefix='01j')`

Error Returned: `Disjunctions not supported`

SOSL Operation Limitations

Two Tooling API objects, EntityDefinition and FieldDefinition, have SOSL limitations.

EntityDefinition and FieldDefinition support the following SOSL operations:

`FIND`

- Literal text search, for example:

  ```
  FIND {account}
  ```

- Text search with a single wildcard, for example:

  ```
  FIND {account*} RETURNING EntityDefinition
  FIND {account?} RETURNING FieldDefinition
  FIND {account*fax} RETURNING EntityDefintion
  FIND {account?fax} RETURNING FieldDefinition
  ```

  The wildcard can’t be the first character in a search term, consistent with search behavior for all objects.

- Quotation marks are supported.

- The escape character `\` (slash) is supported. For example, to search for the character `*` (asterisk), include the escape character:

  ```
  FIND {account\*}
  RETURNING EntityDefinition
  ```

  .

- RETURNING is required.

  ```
  FIND {MyString}
  RETURNING FieldDefinition
  ```

  - Multiple object type names are supported.

    ```
    ...RETURNING EntityDefinition, FieldDefinition
    ```

  - A field list is supported.

    ```
     ... RETURNING EntityDefinition (MasterLabel, QualifiedApiName)
    ```

  - WHERE is supported, though logical operators are not.
  - LIMIT is supported.

Example

```
FIND {account*}
RETURNING FieldDefinition (MasterLabel, NamespacePrefix
WHERE EntityDefinitionId='Account')
```

All other SOSL operations are unsupported. If you include an unsupported expression in a search term, the expression is ignored, except for the following, which cause an error:

- Multiple wildcards in a search term
- Unsupported operators OR or NOT
- Parentheses for grouping operators
- Morphological tokenization
- An asterisk wildcard isn't added at the end of single-character searches.

System Fields

Some fields are system-generated. They are on most Tooling API objects, and are read-only.

These fields are automatically updated during API operations. For example, the `Id` field is automatically generated when a record is created, and the `LastModifiedDate` is automatically updated during any operation on an object.

Field	Field Type	Description
`Id`	ID	Globally unique string that identifies a record. `Id` fields have Defaulted on create and Filter access.
`IsDeleted`	boolean	Indicates whether the record has been moved to the Recycle Bin (`true`) or not (`false`). Because this field does not appear in all objects, it is listed in the field table for each object.
`CreatedBy`	User	The user who created the record. `CreatedBy` fields have Defaulted on create, Filter, Group, and Sort access.

Field	Field Type	Description
`CreatedById`	reference	ID of the User who created this record. `CreatedById` fields have Defaulted on create, Filter, Group, and Sortaccess.
`CreatedDate`	dateTime	Date and time when this record was created. `CreatedDate` fields have Defaulted on create, Filter, and Sort access.
`LastModifiedBy`	User	The user who last modified this record. `LastModifiedBy` fields have Defaulted on create, Filter, Group, and Sort access.
`LastModifiedById`	reference	ID of the User who last updated this record. `LastModifiedById` fields have Defaulted on create, Filter, Group, and Sortaccess.
`LastModifiedDate`	dateTime	Date and time when a user last modified this record. `LastModifiedDate` fields have Defaulted on create, Filter, and Sort access.
`SystemModstamp`	dateTime	Date and time when this record was last modified by a user or by an automated process (such as a trigger). `SystemModstamp` fields have Defaulted on create and Filter access.

To verify which fields are available for an object, check the Tooling API WSDL.

Programming Objects

Use programming objects to interact with programmatic artifacts: Apex, Visualforce, and Lightning.

ApexClass
Represents the saved copy of an Apex class. ApexClass uses the cached version of the class unless one is unavailable.

ApexClassMember
Represents the working copy of an Apex class for editing, saving or compiling in a MetadataContainer.

ApexComponent
Represents the saved copy of a Visualforce component. ApexComponent uses the cached version of the class unless one is unavailable.

ApexComponentMember
Represents the working copy of a Visualforce component for editing, saving, or compiling in a MetadataContainer.

ApexPage
Represents the saved copy of an Apex page. ApexPage uses the cached version of the class unless one is unavailable.

ApexPageMember
Represents the working copy of a Visualforce page for editing, saving, or compiling in a MetadataContainer.

ApexTrigger
Represents the saved copy of an Apex trigger. ApexTrigger uses the cached version of the class unless one is unavailable.

ApexTriggerMember
Represents the working copy of an Apex trigger for editing, saving, or compiling in a MetadataContainer.

AuraDefinition
Represents a Lightning definition, such as component markup, a client-side controller, or an event.

AuraDefinitionBundle
Represents a Lightning definition bundle, such as a component or application bundle. A bundle contains a Lightning definition and all its related resources. Lightning components is a beta feature.

StaticResource

Represents the working copy of a static resource file for editing or saving. Static resources allow you to upload content that you can reference in a Visualforce page, including images, stylesheets, JavaScript, and other files.

Setup Objects

Use setup objects to interact with metadata for declarative development. For example, you may wish to create your own version of Setup, or restrict the amount of data you need to push to an app on a mobile phone.

BusinessProcess

Represents a business process.

CompactLayout

Represents the values that define a compact page layout.

CompactLayoutItemInfo

Represents a field selected for a compact layout, and the order of that field in the compact layout.

CompactLayoutInfo

Represents the metadata for a custom or standard compact layout.

CustomField

Represents a custom field on a custom object that stores data unique to your organization.

CustomFieldMember

Represents the working copy of a field for editing or saving in a MetadataContainer.

CustomObject

Represents a custom object that stores data unique to your organization. Includes access to the associated CustomObject object and related fields in Salesforce Metadata API.

CustomTab

Represents a custom tab.

EmailTemplate

Represents an email template.

EntityDefinition

Provides row-based access to metadata about standard and custom objects.

EntityLimit

Represents the limits for an object as displayed in the Setup UI.

FieldDefinition

Represents a standard or custom field, providing row-based access to field metadata. Contrast FieldDefinition with EntityParticle, which represents each element of a field that can be presented in a user interface. FieldDefinition has parity with metadata type Field.

FieldSet

Represents the metadata for a group of fields.

FlexiPage

Represents a Lightning Page. A Lightning Page is the home page for an app that appears as a menu item in the Salesforce1 navigation menu. Includes access to the associated FlexiPage object in the Salesforce Metadata API.

Flow

Use the Flow object to retrieve and update specific flow versions.

FlowDefinition
The parent of a set of flow versions.

HistoryRetentionJob
Represents the body of retained data from the archive, and the status of the archived data.

KeywordList
Represents a list of keywords used in community moderation.

Layout
Represents a page layout.

LookupFilter
Represents a lookup filter, which restricts the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

MenuItem
Represents a menu item.

ModerationRule
Represents a rule used in your community to moderate user-generated content.

Profile
Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce.

ProfileLayout
Represents a profile layout.

QuickActionDefinition
Represents the definition of a quick action.

QuickActionList
Represents a list of quick actions.

QuickActionListItem
Represents an item in a quick action list.

RecentlyViewed
Represents metadata entities typically found in Setup such as page layout definitions, workflow rule definitions, and email templates that the current user has recently viewed.

RecordType
Represents a custom record type.

SearchLayout
Represents a search layout defined for an object.

Scontrol
Represents a custom s-control, which is custom content that our system hosts, but client applications execute. An s-control can contain any type of content that you can display or run in a Web browser.

User
Represents a user. You can retrieve standard fields on User with the Tooling API, but custom fields can’t be retrieved.

WebLink
Represents a custom link to a URL or S-control.

ValidationRule
Represents a validation rule or workflow rule which specifies the formula for when a condition is met.

WorkflowAlert

Represents a workflow alert. A workflow alert is an email generated by a workflow rule or approval process and sent to designated recipients.

WorkflowFieldUpdate

Represents a workflow field update.

WorkflowOutboundMessage

Represents an outbound message. An outbound message is a workflow, approval, or milestone action that sends the information you specify to an endpoint you designate, such as an external service. Outbound messaging is configured in the Salesforce setup menu. Then you must configure the external endpoint. You can create a listener for the messages using the SOAP API.

WorkflowRule

Represents a workflow rule that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

WorkflowTask

Represents a workflow task that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

Tooling Objects

Use these objects to build tools around test results, debugging, code coverage, and more.

ApexCodeCoverage

Represents code coverage test results for an Apex class or trigger.

ApexCodeCoverageAggregate

Represents aggregate code coverage test results for an Apex class or trigger. Available in Tooling API version 29.0 and later.

ApexExecutionOverlayAction

Specifies an Apex code snippet or SOQL query to execute at a specific line of code in an Apex class or trigger and optionally generate a heap dump.

ApexExecutionOverlayResult

Represents the result from the Apex code snippet or SOQL query defined in the associated ApexExecutionOverlayAction, and the resulting heap dump if one was returned.

ApexLog

Represents a debug log.

ApexOrgWideCoverage

Represents code coverage test results for an entire organization.

ApexResult

A complex type that represents the result of Apex code executed as part of an ApexExecutionOverlayAction, returned in an ApexExecutionOverlayResult.

ApexTestQueueItem

Represents a single Apex class in the Apex job queue.

HeapDump

A complex type that represents a heap dump in an ApexExecutionOverlayResult object.

SOQLResult

A complex type that represents the result of a SOQL query in an ApexExecutionOverlayResult object.

SymbolTable

A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`.

TraceFlag

Represents a trace flag that triggers an Apex debug log at the specified logging level.

Operational Objects

Use the following objects for Tooling API operations.

ContainerAsyncRequest

Allows you to compile and asynchronously deploy a MetadataContainer object to your organization.

DeployDetails

A complex type that contains detailed XML for any compile errors reported in the asynchronous request defined by a ContainerAsyncRequest object.

MetadataContainer

Manages working copies of ApexClassMember, ApexTriggerMember, ApexPageMember, and ApexComponentMember objects, including collections of objects to be deployed together.

The following Tooling API objects are used internally by the Developer Console.

- IDEPerspective
- IDEWorkspace
- User.WorkspaceId

CHAPTER 2 Tooling API Objects

Tooling API includes the following objects:

IN THIS SECTION:

ApexClass

Represents the saved copy of an Apex class. ApexClass uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

ApexClassMember

Represents the working copy of an Apex class for editing, saving or compiling in a MetadataContainer.

ApexCodeCoverage

Represents code coverage test results for an Apex class or trigger. Available in Tooling API version 29.0 and later.

ApexCodeCoverageAggregate

Represents aggregate code coverage test results for an Apex class or trigger. Available in Tooling API version 29.0 and later.

ApexComponent

Represents the saved copy of a Visualforce component. ApexComponent uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

ApexComponentMember

Represents the working copy of a Visualforce component for editing, saving, or compiling in a MetadataContainer.

ApexEmailNotification

Stores Salesforce users and external email addresses to be notified when unhandled Apex exceptions occur. Available in API version 35.0 and later.

ApexExecutionOverlayAction

Specifies an Apex code snippet or SOQL query to execute at a specific line of code in an Apex class or trigger and optionally generate a heap dump.

ApexExecutionOverlayResult

Represents the result from the Apex code snippet or SOQL query defined in the associated ApexExecutionOverlayAction, and the resulting heap dump if one was returned. Available from API version 28.0 or later.

ApexLog

Represents a debug log.

ApexOrgWideCoverage

Represents code coverage test results for an entire organization. Available in Tooling API version 29.0 and later.

ApexPage

Represents the saved copy of an Apex page. ApexPage uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

ApexPageMember

Represents the working copy of a Visualforce page for editing, saving, or compiling in a MetadataContainer.

ApexResult

A complex type that represents the result of Apex code executed as part of an ApexExecutionOverlayAction, returned in an ApexExecutionOverlayResult. Available from API version 28.0 or later.

ApexTestQueueItem

Represents a single Apex class in the Apex job queue. Available in API version 30.0 and later.

ApexTestResult

Represents the result of an Apex test method execution. Available from API version 30.0 or later.

ApexTrigger

Represents the saved copy of an Apex trigger. ApexTrigger uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

ApexTriggerMember

Represents the working copy of an Apex trigger for editing, saving, or compiling in a MetadataContainer.

AssignmentRule

Don't use this object.

AuraDefinition

Represents a Lightning definition, such as component markup, a client-side controller, or an event. Available in API version 32.0 and later.

AuraDefinitionBundle

Represents a Lightning definition bundle, such as a component or application bundle. A bundle contains a Lightning definition and all its related resources. Lightning components is a beta feature. Available in API version 32.0 and later.

AutoResponseRule

Specifies whether the autoresponse rule is active (`true`).

BusinessProcess

Represents a business process.

CompactLayout

Represents the values that define a compact page layout.

CompactLayoutItemInfo

Represents a field selected for a compact layout, and the order of that field in the compact layout.

CompactLayoutInfo

Represents the metadata for a custom or standard compact layout.

ContainerAsyncRequest

Allows you to compile and asynchronously deploy a MetadataContainer object to your organization.

CustomField

Represents a custom field on a custom object that stores data unique to your organization. Includes access to the associated CustomField object and related fields in Salesforce Metadata API. Available from API version 28.0 or later.

CustomFieldMember

Represents the working copy of a field for editing or saving in a MetadataContainer. This object is available in API version 33.0 and later.

CustomObject

Represents a custom object that stores data unique to your organization. Includes access to the associated CustomObject object and related fields in Salesforce Metadata API. Available from API version 31.0 or later.

CustomTab

Represents a custom tab.

DataType

Represents the datatype of a field. Use this object with EntityDefinition, EntityParticle, or FieldDefinition to simplify queries. Available in Tooling API version 34.0 and later.

DebugLevel

Represents a set of log category levels to assign to a `TraceFlag` object. Multiple trace flags can use a debug level.

DeployDetails

A complex type that contains detailed XML for any compile errors reported in the asynchronous request defined by a ContainerAsyncRequest object. Replaces the JSON field `CompilerErrors` in Tooling API version 31.0 and later.

EmailTemplate

Represents an email template.

EntityDefinition

Provides row-based access to metadata about standard and custom objects.

EntityLimit

Represents the limits for an object as displayed in the Setup UI.

EntityParticle

Represents each element of a field that can be presented in a user interface. Contrast EntityParticle with FieldDefinition, which represents each element of a field defined in the Metadata API. EntityParticle has parity with `describe`. Available in Tooling API version 34.0 and later.

FieldDefinition

Represents a standard or custom field, providing row-based access to field metadata. Contrast FieldDefinition with EntityParticle, which represents each element of a field that can be presented in a user interface. FieldDefinition has parity with metadata type Field.

FieldSet

Represents the metadata for a group of fields. Available from API version 33.0 or later.

FlexiPage

Represents a Lightning Page. A Lightning Page is the home page for an app that appears as a menu item in the Salesforce1 navigation menu. Includes access to the associated FlexiPage object in the Salesforce Metadata API. Available from API version 31.0 or later.

Flow

Use the Flow object to retrieve and update specific flow versions.

FlowDefinition

The parent of a set of flow versions.

HeapDump

A complex type that represents a heap dump in an ApexExecutionOverlayResult object. Available from API version 28.0 or later.

HistoryRetentionJob

Represents the body of retained data from the archive, and the status of the archived data. Available in API version 29.0 or later.

HomePageComponent

Represents a home page component.

HomePageLayout

Represents a home page layout.

KeywordList

Represents a list of keywords used in community moderation. Available in Tooling API version 36.0 and later.

Layout

Represents a page layout.

LookupFilter

Represents a lookup filter, which restricts the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

MenuItem

Represents a menu item.

MetadataContainer

Manages working copies of ApexClassMember, ApexTriggerMember, ApexPageMember, and ApexComponentMember objects, including collections of objects to be deployed together.

ModerationRule

Represents a rule used in your community to moderate user-generated content. Available in Tooling API version 36.0 and later.

OwnerChangeOptionInfo

Represents default and optional actions that can be performed when a record’s owner is changed. Available in Tooling API version 35.0 and later.

PathAssistant

Represents a Sales Path. Available in Tooling API version 36.0 and later.

PathAssistantStepInfo

Represents guidance for a step on a Sales Path. Available in Tooling API version 36.0 and later.

PathAssistantStepItem

Represents layout or guidance details for a step on a Sales Path. Available in Tooling API version 36.0 and later.

PostTemplate

Represents an approval post tempalte for Approvals in Chatter.

Profile

Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce.

ProfileLayout

Represents a profile layout.

Publisher

Represents the publisher of objects and fields. For example, Salesforce is the publisher for standard objects, the organization is the publisher for custom objects, and the package is the publisher for installed packages. Available in Tooling API version 34.0 and later.

QueryResult

Represents the results of a query. For example, if you query on the object EntityDefinition, all the layouts for that entity are returned as an array of QueryResult objects in the `Layouts` field. Available in Tooling API version 34.0 and later.

QuickActionDefinition

Represents the definition of a quick action.

QuickActionList

Represents a list of quick actions.

QuickActionListItem

Represents an item in a quick action list.

RecentlyViewed

Represents metadata entities typically found in Setup such as page layout definitions, workflow rule definitions, and email templates that the current user has recently viewed.

RecordType

Represents a custom record type.

RelationshipDomain

Represents the relationship an object has with other objects. RelationshipDomain allows you to write simpler queries. For example, "which objects are the child objects for the object defined in `ParentSobject`" is easier using RelationshipDomain. Available in Tooling API version 34.0 and later.

RelationshipInfo

Represents the properties of a relationship between objects. Simplify queries with RelationshipInfo, such as answering the question "which objects are parent objects for the object defined in `ChildSobject`. Available in Tooling API version 34.0 and later.

SandboxInfo

Represents a sandbox.

SandboxProcess

Represents the sandbox copy process for a SandboxInfo record.

SearchLayout

Represents a search layout defined for an object.

ServiceFieldDataType

Don't use this object.

Scontrol

Represents a custom s-control, which is custom content that our system hosts, but client applications execute. An s-control can contain any type of content that you can display or run in a Web browser.

SOQLResult

A complex type that represents the result of a SOQL query in an ApexExecutionOverlayResult object. Available from API version 28.0 or later.

StandardAction

Represents the buttons, links, and actions (standard actions) for a standard or custom object. This object is available in API version 34.0 and later.

StaticResource

Represents the working copy of a static resource file for editing or saving. Static resources allow you to upload content that you can reference in a Visualforce page, including images, stylesheets, JavaScript, and other files. Available in Tooling API version 29.0 and later.

SymbolTable

A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`.

TraceFlag

Represents a trace flag that triggers an Apex debug log at the specified logging level.

TransactionSecurityPolicy

Represents a transaction security policy definition (policy).

User

Represents a user. You can retrieve standard fields on User with the Tooling API, but custom fields can't be retrieved.

UserEntityAccess

Represents the access that the current user has to an object. Available in Tooling API version 34.0 and later.

UserFieldAccess

Represents the access that the current user has to a field. Available in Tooling API version 34.0 and later.

ValidationRule

Represents a validation rule or workflow rule which specifies the formula for when a condition is met. Available from API version 34.0 or later.

WebLink

Represents a custom link to a URL or S-control. Available in the Tooling APIfrom API version 34.0 or later.

WorkflowAlert

Represents a workflow alert. A workflow alert is an email generated by a workflow rule or approval process and sent to designated recipients.

WorkflowFieldUpdate

Represents a workflow field update.

WorkflowOutboundMessage

Represents an outbound message. An outbound message is a workflow, approval, or milestone action that sends the information you specify to an endpoint you designate, such as an external service. Outbound messaging is configured in the Salesforce setup menu. Then you must configure the external endpoint. You can create a listener for the messages using the SOAP API.

WorkflowRule

Represents a workflow rule that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

WorkflowTask

Represents a workflow task that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

ApexClass

Represents the saved copy of an Apex class. ApexClass uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

To edit, save, or compile Apex classes, use ApexClassMember.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query, GET, POST, PATCH, DELETE`

Fields

Field Name	Details
`SymbolTable`	**Type** SymbolTable **Properties** Nillable **Description** A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`. This field is null if the symbol table cannot be created.

Usage

To retrieve information about an Apex class, create an ApexClass object that references it. For example code, see SOAP Calls.

To edit, save, or compile Apex classes, use ApexClassMember.

Note: If there is not a cached version of SymbolTable, it will be compiled in the background and the query might take longer than expected. The SymbolTable returned from ApexClass does not contain references; to retrieve a SymbolTable with references, use ApexClassMember.

ApexClassMember

Represents the working copy of an Apex class for editing, saving or compiling in a MetadataContainer.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Body`	**Type** string **Properties** Create, Update

Field Name	Details
	Description The data for the Apex class. The `Body` field is the only field you can `update()` or `PATCH`.
`Content`	**Type** string **Properties** None **Description** A string representation of ApexClassMetadata that lists the version, status, and packaged versions of the corresponding Apex class.
`ContentEntityId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to an Apex class. There can be only one `ContentEntityId` per ApexClassMember, otherwise, an error is reported. This field is required if `FullName` is not specified.
`FullName`	**Type** string **Properties** Group, Nillable **Description** The full name of the associated object in the Metadata API. Use to avoid race conditions on create, before you have IDs. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance. This field is required if `ContentEntityId` is not specified.
`LastSyncDate`	**Type** dateTime **Properties** Filter, Sort **Description** The date and time that this ApexClassMember `Body` was replicated from the underlying Apex class.

Field Name	Details
	When you deploy a MetadataContainer, this value is compared with the `LastModifiedDate` of the underlying Apex class. If `LastSyncDate` is older than `LastModifiedDate`, the deployment fails with an error.
`Metadata`	**Type** ApexClassMetadata **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding Apex class. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`MetadataContainerId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to a MetadataContainer or ContainerAsyncRequest object. As part of a successful deployment, this field is reset from the ID of the deployed MetadataContainer to the ID of the corresponding ContainerAsyncRequest object. This field is required.
`SymbolTable`	**Type** SymbolTable **Properties** Nillable **Description** A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`. This field is null if the symbol table can't be created. A symbol table can't be created if the content referenced by the `ContentEntityId` field doesn't use a symbol table. Compiler errors for the last deployment of the MetadataContainer in the `MetadataContainerId` field also prevent a symbol table from being created.

Usage

To edit, save, or compile an Apex class, create an ApexClassMember object that references it.

Note: Once an ApexClassMember is successfully deployed in a MetadataContainer, the `MetadataContainerId` is changed to the ID of the ContainerAsyncRequest, and the ApexClassMember can't be modified or reused.

Apex classes are often dependent on each other for functionality. For example, a method in one class can call a method in another class. If source file A is dependent on modified source file B and you try to save and compile source file A before you've saved the changes to source file B, the compiler will throw an error. To successfully save and compile a group of related source files, put the corresponding ApexClassMember and ApexTriggerMember objects in a single MetadataContainer object.

Each ApexClassMember object can only refer to a single MetadataContainer object. Multiple ApexClassMember objects can refer to the same MetadataContainer object.

ApexCodeCoverage

Represents code coverage test results for an Apex class or trigger. Available in Tooling API version 29.0 and later.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API HTTP Methods

`Query`, `GET`

Fields

Field	Details
`ApexTestClassId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the test class.
`TestMethodName`	**Type** string **Properties** Filter, Group, Sort **Description** The name of the test method.
`ApexClassorTriggerId`	**Type** string **Properties** Filter, Group, Sort

Field	Details
	Description The ID of the class or trigger under test.
`NumLinesCovered`	**Type** int **Properties** Filter, Group, Sort **Description** The number of covered lines.
`NumLinesUncovered`	**Type** int **Properties** Filter, Group, Sort **Description** The number of uncovered lines.
`Coverage`	**Type** complexvalue **Properties** None **Description** Two lists of integers. The first is the covered lines, and the second is the list of uncovered lines. If a lines is missing from both lists, the line is not executable and does not require coverage. Coverage includes the following fields: • `coveredLines` • `namespace` • `uncoveredLines`

Usage

To query for code coverage, specify an Apex class, test class, or both. The returned JSON or XML object will contain two lists of integers: one for covered and one for uncovered lines.

The following example SOQL query retrieves code coverage results for a specific class or trigger covered by a specific test class:

```
SELECT Coverage
FROM ApexCodeCoverage
WHERE ApexClassOrTrigger = ‘01pD000000066GR’
AND ApexTestClass = ‘01pD000000064pu’
```

For per-class code coverage, the query would be:

```
SELECT Coverage
FROM ApexCodeCoverage
WHERE ApexClassOrTrigger = ‘01pD000000066GR’
```

Note: In this case, multiple rows may be returned, since there may be multiple test classes that cover the same test class.

As noted above, `Coverage` is returned as two lists of integers. The first is the covered lines, and the second is the list of uncovered lines. If a line is missing from both lists, the line is not executable and does not require coverage. For example, if the covered lines are 2, 9, and 11, and uncovered lines are 3, 4, 5, and 6; the result would be: `{2,9,11},{3,4,5,6}`. The missing lines (1, 7, 8 and 10) are not executable.

Code coverage percentage is a simple calculation of the number of covered lines divided by the sum of the number of covered lines and the number of uncovered lines. For example, to calculate code coverage percentage in SOAP:

```
ApexCodeCoverage acc = null; //Query for an ApexCodeCoverage object
Coverage coverage = acc.coverage;
int[] covered = coverage.coveredLines;
int[] uncovered = coverage.uncoveredLines;
int percent = covered.length / (covered.length + uncovered.length);
System.out.println("Total class coverage is " + percent + "%.");
```

ApexCodeCoverageAggregate

Represents aggregate code coverage test results for an Apex class or trigger. Available in Tooling API version 29.0 and later.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API HTTP Methods

`Query`, `GET`, `DELETE`

Fields

Field	Details
`ApexClassorTriggerId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the class or trigger under test.
`NumLinesCovered`	**Type** int

Field	Details
	Properties Filter, Group, Sort **Description** The number of covered lines.
`NumLinesUncovered`	**Type** int **Properties** Filter, Group, Sort **Description** The number of uncovered lines.
`Coverage`	**Type** complexvalue **Properties** None **Description** Two lists of integers. The first is the covered lines, and the second is the list of uncovered lines. If a lines is missing from both lists, the line is not executable and does not require coverage. Coverage includes the following fields: • `coveredLines` • `namespace` • `uncoveredLines`

Usage

To query for aggregate code coverage, specify an Apex test class. The returned JSON or XML object will contain two lists of integers: one for covered and one for uncovered lines. For examples, see ApexCodeCoverage.

ApexComponent

Represents the saved copy of a Visualforce component. ApexComponent uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

To edit, save, or compile Visualforce components, use ApexComponentMember.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

Query, GET, POST, PATCH, DELETE

Usage

To retrieve information about a Visualforce component, create an ApexComponent object that references it. For example code, see SOAP Calls.

To edit, save, or compile Visualforce components, use ApexComponentMember.

ApexComponentMember

Represents the working copy of a Visualforce component for editing, saving, or compiling in a MetadataContainer.

Supported SOAP API Calls

create(), delete(), describeSObjects(), query(), retrieve(), update(), upsert()

Supported REST API HTTP Methods

Query, GET, POST, PATCH, DELETE

Fields

Field Name	Details
Body	**Type** string **Properties** Create, Update **Description** The data for the Visualforce component. The Body field is the only field you can update() or PATCH.
Content	**Type** string **Properties** None **Description** A string representation of ApexComponentMetadata that lists the version, status, and packaged versions of the corresponding Visualforce component.
ContentEntityId	**Type** reference

Field Name	Details
	Properties Create, Filter, Group, Sort **Description** A reference to a Visualforce component. There can be only one `ContentEntityId` per ApexComponentMember, otherwise, an error is reported. This field is required if `FullName` is not specified.
`FullName`	**Type** string **Properties** Group, Nillable **Description** The full name of the associated object in the Metadata API. Use to avoid race conditions on create, before you have IDs. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance. This field is required if `ContentEntityId` is not specified.
`LastSyncDate`	**Type** dateTime **Properties** Filter, Sort **Description** The date that this ApexComponentMember `Body` was replicated from the underlying entity. When you deploy a MetadataContainer, this value is compared with the `LastModifiedDate` of the underlying Visualforce component. If `LastSyncDate` is older than `LastModifiedDate`, the deployment fails with an error.
`Metadata`	**Type** ApexComponentMetadata **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding Visualforce component. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field Name	Details
MetadataContainerId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to a MetadataContainer or ContainerAsyncRequest object. As part of a successful deployment, this field is reset from the ID of the deployed MetadataContainer to the ID of the corresponding ContainerAsyncRequest object. This field is required.

Usage

To edit, save, or compile a Visualforce component, create an ApexComponentMember object that references it. To create a Visualforce component, use the Force.com REST API or the Metadata API.

Note: Once an ApexComponentMember is successfully deployed in a MetadataContainer, the `MetadataContainerId` is changed to the ID of the ContainerAsyncRequest, and the ApexComponentMember can't be modified or reused.

Visualforce pages and components are often dependent on each other for functionality. To successfully save and compile a group of related source files, put the corresponding ApexComponentMember and ApexPageMember objects in a single MetadataContainer object.

Each ApexComponentMember object can only refer to a single MetadataContainer object. Multiple ApexComponentMember objects can refer to the same MetadataContainer object.

ApexEmailNotification

Stores Salesforce users and external email addresses to be notified when unhandled Apex exceptions occur. Available in API version 35.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Email`	**Type** email **Properties** Create, Filter, Group, idLookup, Nillable, Sort, Update **Description** A semicolon-delimited list of email addresses to notify when unhandled Apex exceptions occur.
`UserId`	**Type** ID **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** Users of your org to notify when unhandled Apex exceptions occur.

Usage

To notify users of your org at the email addresses they have on record, use `UserId`. To notify external users or alternate email addresses, use `Email`.

ApexExecutionOverlayAction

Specifies an Apex code snippet or SOQL query to execute at a specific line of code in an Apex class or trigger and optionally generate a heap dump.

Supported SOAP Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`ActionScript`	**Type** string

Field Name	Details
	Properties Create, Nillable, Update **Description** The Apex code or SOQL query to run when execution reaches the line in the Apex class or trigger at the specified iteration. Results will be included in the heap dump file.
`ActionScriptType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Indicates whether the `ActionScript` is written in Apex or SOQL. Valid values are: • `None` • `Apex` • `SOQL` This field is required.
`ExecutableEntityId`	**Type** reference **Properties** Create, Filter, Group, Sort, **Description** The ID of the Apex class or trigger being executed. This field is required.
`ExpirationDate`	**Type** dateTime **Properties** Create, Filter, Sort, Update **Description** The expiration date of the overlay action. This field is required.
`IsDumpingHeap`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether a heap dump is generated (`true`) or not (`false`). To execute the `ActionScript` without generating a heap dump, set this field to `false`. This field is required.

Field Name	Details
Iteration	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The number of times the specified line should execute before the heap dump is generated. This field is required.
Line	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The line number of the heap dump marker. This field is required.
ScopeId	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** The user who executed the action. This field is required.

Usage

When you are troubleshooting a runtime issue, you often want to find out more about the state of a variable or the state of the database, or create a specific condition to test your code. Use ApexExecutionOverlayAction to overlay a diagnostic output on an Apex class or trigger without compromising production code.

ApexExecutionOverlayResult

Represents the result from the Apex code snippet or SOQL query defined in the associated ApexExecutionOverlayAction, and the resulting heap dump if one was returned. Available from API version 28.0 or later.

Supported SOAP Calls

query(), retrieve(), delete()

Supported REST HTTP Methods

Query, GET, DELETE

Fields

Field Name	Details
`ActionScript`	**Type** string **Properties** Nillable **Description** The Apex code or SOQL query that was run.
`ActionScriptType`	**Type** picklist **Properties** Filter, Group, Sort, Nillable **Description** Indicates whether the `ActionScript` is written in Apex or SOQL. Valid values are: - `None` - `Apex` - `SOQL`
`ApexResult`	**Type** ApexResult **Properties** Nillable **Description** A complex type that represents the result of Apex code executed as part of an ApexExecutionOverlayAction, returned in an ApexExecutionOverlayResult.
`ExpirationDate`	**Type** dateTime **Properties** Filter, Sort **Description** The expiration date of the overlay action.
`HeapDump`	**Type** HeapDump **Properties** Nillable

Field Name	Details
	Description A complex type that represents a heap dump in an ApexExecutionOverlayResult object. You can only have a single row when using HeapDump in SOQL. To select only one row, you can use a LIMIT=1 clause in your SOQL query, or you can list multiple rows for the user and have them select the row to inspect.
`IsDumpingHeap`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether a heap dump was generated (`true`) or not (`false`).
`Iteration`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The number of times the specified line should execute before the heap dump is generated. This field is required.
`Line`	**Type** int **Properties** Filter, Group, Sort, Nillable **Description** The line number of the checkpoint.
`SOQLResult`	**Type** SOQLResult **Properties** Nillable **Description** A complex type that represents the result of a SOQL query in an ApexExecutionOverlayResult object.
`UserId`	**Type** reference **Properties** Filter, Group, Sort, **Description** The user who executed the action.

Usage

When you are troubleshooting a runtime issue, you often want to find out more about the state of a variable or the state of the database, or create a specific condition to test your code. Use ApexExecutionOverlayAction to overlay a diagnostic output on an Apex class or trigger without compromising production code, and use ApexExecutionOverlayResult to navigate the results.

ApexLog

Represents a debug log.

To retrieve a raw log by ID, use the REST resource: `/sobjects/`**`ApexLog/`*`id`***`/Body/`. (Available from API version 28.0 or later.)

Supported SOAP Calls

`delete()`, `describeSObjects()`, `query()`, `retrieve()`

Supported REST HTTP Methods

`Query`, `GET`, `DELETE`

Fields

Field	Details
`Application`	**Type** textarea **Properties** Filter, Group, Sort **Description** This value depends on the client type that triggered the log or heap dump. • For API clients, this value is the client ID. • For browser clients, this value is `Browser`. This field is required.
`DurationMilliseconds`	**Type** int **Properties** Filter, Group, Sort **Description** The duration of the transaction in milliseconds. This field is required.
`Location`	**Type** picklist

Field	Details
	Properties Filter, Group, Sort, Nillable, Restricted picklist **Description** Specifies the location of the origin of the log or heap dump. Values are: - `Monitoring` — Generated as part of debug log monitoring and visible to all administrators. These types of logs are maintained until the user or the system overwrites them. - `SystemLog` — Generated as part of system log monitoring and visible only to you. These types of logs are only maintained for 60 minutes or until the user clears them. - `Preserved` — A system log that is maintained longer than 60 minutes. Used for internal support.
`LogLength`	**Type** int **Properties** Filter, Group, Sort **Description** Length of the log or heap dump in bytes. This field is required.
`LogUserId`	**Type** reference **Properties** Filter, Group, Sort, Nillable **Description** ID of the user whose actions triggered the debug log or heap dump.
`Operation`	**Type** string **Properties** Filter, Group, Sort **Description** Name of the operation that triggered the debug log or heap dump, such as `APEXSOAP`, `Apex Sharing Recalculation`, and so on. This field is required.
`Request`	**Type** string **Properties** Filter, Group, Sort **Description** Request type. Values are: - `API` — Request came from an API.

Field	Details
	- `Application` — Request came from the Salesforce user interface. This field is required.
`StartTime`	**Type** dateTime **Properties** Filter, Sort **Description** Start time of the transaction. This field is required.
`Status`	**Type** string **Properties** Filter, Group, Sort **Description** Status of the transaction. This value is either `Success`, or the text of an unhandled Apex exception. This field is required.

ApexOrgWideCoverage

Represents code coverage test results for an entire organization. Available in Tooling API version 29.0 and later.

Supported SOAP API Calls

`describeSObjects()`, `delete()`, `query()`, `retrieve()`

Supported REST API HTTP Methods

`Query`, `GET`, `DELETE`

Fields

Field	Details
`PercentCovered`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The percentage of the code in the organization that is covered by tests.

ApexPage

Represents the saved copy of an Apex page. ApexPage uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

To edit, save, or compile Apex pages, use ApexPageMember.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Usage

To retrieve information about a Visualforce page, create an ApexPage object that references it. For example code, see SOAP Calls.

To edit, save, or compile Visualforce pages, use ApexPageMember.

ApexPageMember

Represents the working copy of a Visualforce page for editing, saving, or compiling in a MetadataContainer.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Body`	**Type** string **Properties** Create, Update **Description** The data for the Visualforce page. The `Body` field is the only field you can `update()` or `PATCH`.

Field Name	Details
`Content`	**Type** string **Properties** None **Description** A string representation of ApexPageMetadata that lists the version, status, and packaged versions of the corresponding Visualforce page.
`ContentEntityId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to a Visualforce page. There can be only one `ContentEntityId` per ApexPageMember, otherwise, an error is reported. This field is required if `FullName` is not specified.
`FullName`	**Type** string **Properties** Group, Nillable **Description** The full name of the associated object in the Metadata API. Use to avoid race conditions on create, before you have IDs. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance. This field is required if `ContentEntityId` is not specified.
`LastSyncDate`	**Type** dateTime **Properties** Filter, Sort **Description** The date that this ApexPageMember `Body` was replicated from the underlying entity. When you deploy a MetadataContainer, this value is compared with the `LastModifiedDate` of the underlying Visualforce page. If `LastSyncDate` is older than `LastModifiedDate`, the deployment fails with an error.

Field Name	Details
`Metadata`	**Type** ApexPageMetadata **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding Visualforce page. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`MetadataContainerId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to a MetadataContainer or ContainerAsyncRequest object. As part of a successful deployment, this field is reset from the ID of the deployed MetadataContainer to the ID of the corresponding ContainerAsyncRequest object. This field is required.

Usage

To edit, save, or compile a Visualforce page, create an ApexPageMember object that references it. To create a Visualforce page, use the Force.com REST API or the Metadata API.

Note: Once an ApexPageMember is successfully deployed in a MetadataContainer, the `MetadataContainerId` is changed to the ID of the ContainerAsyncRequest, and the ApexPageMember can't be modified or reused.

Visualforce pages and components are often dependent on each other for functionality. To successfully save and compile a group of related source files, put the corresponding ApexPageMember and ApexComponentMember objects in a single MetadataContainer object. Use ContainerAsyncRequest to send the MetadataContainer to the application server.

Each ApexPageMember object can only refer to a single MetadataContainer object. Multiple ApexPageMember objects can refer to the same MetadataContainer object.

ApexResult

A complex type that represents the result of Apex code executed as part of an ApexExecutionOverlayAction, returned in an ApexExecutionOverlayResult. Available from API version 28.0 or later.

Fields

Field	Details
`apexError`	**Type** string **Description** The error text returned if the execution was unsuccessful.
`apexExecutionResult`	**Type** ExecuteAnonymousResult **Description** The structured result returned from a successful execution. ExecuteAnonymousResult includes the following fields: • `column` • `compileProblem` • `compiled` • `exceptionMessage` • `exceptionStackTrace` • `line` • `success` Note: `ExecuteAnonymousResult` is outside the current execution context and does not provide access to variables in the heap.

Usage

Overlay Apex on checkpoints to capture structured debugging information. If your SOQL query may return more than one record when dealing with complex types, select only one row. For example, you can use a LIMIT=1 clause in your SOQL query, or you can list rows for the user and have them select the row to inspect.

ApexTestQueueItem

Represents a single Apex class in the Apex job queue. Available in API version 30.0 and later.

Supported SOAP API Calls

`create()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`

Fields

Field Name	Details
`ApexClassId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The Apex class whose tests are to be executed. This field can't be updated.
`Status`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort, Update **Description** The status of the test. Valid values are: - `Queued` - `Processing` - `Aborted` - `Completed` - `Failed` - `Preparing` - `Holding` To abort a class that is in the Apex job queue, perform an update operation on the `ApexTestQueueItem` object and set its `Status` field to `Aborted`.
`ExtendedStatus`	**Type** string **Properties** Filter, Sort, Nillable **Description** The pass rate of the test run. For example: "(4/6)". This means that four out of a total of six tests passed. If the class fails to execute, this field contains the cause of the failure.
`ParentJobId`	**Type** reference **Properties** Filter, Group, Sort, Nillable,

Field Name	Details
	Description Read-only. Points to the `AsyncApexJob` that represents the entire test run. If you insert multiple Apex test queue items in a single bulk operation, the queue items will share the same parent job. This means that a test run can consist of the execution of the tests of several classes if all the test queue items are inserted in the same bulk operation.

Usage

Insert an ApexTestQueueItem object to place its corresponding Apex class in the Apex job queue for execution. The Apex job executes the test methods in the class.

The example `RunTestListener.java` class below subscribes to the TestResult system topic and prints out the test results using ApexTestQueueItem and ApexTestResult. The example assumes the following:

- You have already set up a Java client application for Streaming API. This example uses the `org.cometd.client.BayeuxClient` created in the Java Client code example in the *Streaming API Developer Guide*.
- You have a logged in `com.sforce.soap.tooling.SoapConnection`. For examples, see the *SOAP API Developer Guide*.

Note: The `RunTestListener.java` class must be instantiated after the Streaming API handshake. For example:

```
SoapConnection toolingConn; //Already set and logged in
BayeuxClient client; //Already set and logged in

//Listen on the handshake event
boolean handshaken = client.waitFor(10 * 1000, BayeuxClient.State.CONNECTED);
if (!handshaken) {
   System.out.println("Failed to handshake: " + client);
   System.exit(1);
}
final RunTestListener = null;
client.getChannel(Channel.META_SUBSCRIBE).addListener(
   new ClientSessionChannel.MessageListener() {
      public void onMessage(ClientSessionChannel channel, Message message) {
         boolean success = message.isSuccessful();
         if (success) {
            //Replace with your own ApexClass ids
            String apexTestClassId1 = "01pD00000007M0CIAU";
            String apexTestClassId2 = "01pD00000007NqtIAE";
            String apexTestSuiteId1 = "05FD00000004CDBMA2";
            listener.runTests(new String[]{apexTestClassId1, apexTestClassId2}, new
String[]{apexTestSuiteId1}, 1);
         }
      }
   };
);
//This will subscribe to the TestRun system topic
listener = new RunTestListener(client, toolingConn);
```

```
import java.util.HashMap;
import org.cometd.bayeux.Message;
import org.cometd.bayeux.client.ClientSessionChannel;
import org.cometd.bayeux.client.ClientSessionChannel.MessageListener;
import org.cometd.client.BayeuxClient;

import com.sforce.soap.tooling.ApexTestQueueItem;
import com.sforce.soap.tooling.ApexTestResult;
import com.sforce.soap.tooling.QueryResult;
import com.sforce.soap.tooling.SObject;
import com.sforce.soap.tooling.SoapConnection;
import com.sforce.ws.ConnectionException;

public class RunTestListener {
   private static final String CHANNEL = "/systemTopic/TestResult";
   private SoapConnection conn;

   public RunTestListener(BayeuxClient client, SoapConnection conn) {
      this.conn = conn;
      System.out.println("Subscribing for channel: " + CHANNEL);
      client.getChannel(CHANNEL).subscribe(new MessageListener() {
         @Override
         public void onMessage(ClientSessionChannel channel, Message message) {
            HashMap data = (HashMap) message.getData();
            HashMap sobject = (HashMap) data.get("sobject");
            String id = (String) sobject.get("Id");
            System.out.println("\nAysncApexJob " + id);
            getTestQueueItems(id);
         }
     });
   }

   public void runTests(String[] apexTestClassIds, String[] apexTestSuiteIds, Integer
maxFailedTests) {
      // apexTestClassIds or apexTestSuiteIds is required; maxFailedTests is optional
      if (apexTestClassIds.length == 0 && apexTestSuiteIds.length == 0) {
         System.out.println("No test to run");
         return;
      }
      String classIds = null;
      if (apexTestClassIds.length > 0) {
         classIds = apexTestClassIds[0];
         for (int i = 1; i < apexTestClassIds.length; i++) {
            classIds += "," + apexTestClassIds[i];
            }
      }
      String suiteIds = null;
      if (apexTestSuiteIds.length > 0) {
         suiteIds = apexTestSuiteIds[0];
         for (int i = 1; i < apexTestSuiteIds.length; i++) {
            suiteIds += "," + apexTestSuiteIds[i];
         }
      }
      try {
         System.out.println("Running async test run");
         conn.runTestsAsynchronous(classIds, suiteIds, maxFailedTests);
      } catch (ConnectionException e) {
         e.printStackTrace();
      }
   }

   private void getTestQueueItems(String asyncApexJobId) {
      try {
         QueryResult res = conn
           .query("SELECT Id, Status, ApexClassId FROM ApexTestQueueItem WHERE ParentJobId
 = '"
               + asyncApexJobId + "'");
         if (res.getSize() > 0) {
            for (SObject o : res.getRecords()) {
               ApexTestQueueItem atqi = (ApexTestQueueItem) o;
               System.out.println("\tApexTestQueueItem - "+atqi.getStatus());
               if (atqi.getStatus().equals("Completed")) {
                  getApexTestResults(atqi.getId());
               }
            }
         } else {
            System.out.println("No queued items for " + asyncApexJobId);
         }
      } catch (ConnectionException e) {
         e.printStackTrace();
      }
   }

   private void getApexTestResults(String apexTestQueueItemId) {
      try {
         QueryResult res = conn
           .query("SELECT StackTrace,Message, AsyncApexJobId,MethodName, Outcome,ApexClassId
 FROM ApexTestResult WHERE QueueItemId = '"
               + apexTestQueueItemId + "'");
         if (res.getSize() > 0) {
            for (SObject o : res.getRecords()) {
               ApexTestResult atr = (ApexTestResult) o;
               System.out.println("\tTest result for "
                  + atr.getApexClassId() + "." + atr.getMethodName());
               String msg = atr.getOutcome().equals("Fail") ? " - "
                  + atr.getMessage() + " " + atr.getStackTrace() : "";
               System.out.println("\t\tTest " + atr.getOutcome() + msg);
            }
         } else {
            System.out.println("No Test Results for " + apexTestQueueItemId);
         }
      } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

ApexTestResult

Represents the result of an Apex test method execution. Available from API version 30.0 or later.

Supported SOAP API Calls

`describeSObjects()`, `query()`, `retrieve()`

Supported REST API HTTP Methods

`Query`, `GET`

Fields

Field Name	Details
`ApexClassId`	**Type** reference **Properties** Filter, Group, Sort **Description** The Apex class whose test methods were executed.
`ApexLogId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** Points to the `ApexLog` for this test method execution if debug logging is enabled; otherwise, `null`.
`AsyncApexJobId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** Read-only. Points to the `AsyncApexJob` that represents the entire test run.

Field Name	Details
	This field points to the same object as `ApexTestQueueItem.ParentJobId`.
`Message`	**Type** string **Properties** Filter, Nillable, Sort **Description** The exception error message if a test failure occurs; otherwise, `null`.
`MethodName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the test method.
`Outcome`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The result of the test. Valid values are: • `Pass` • `Fail` • `CompileFail` • `Skip`
`QueueItemId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** Points to the ApexTestQueueItem which is the class that this test method is part of.
`StackTrace`	**Type** string **Properties** Filter, Nillable, Sort

Field Name	Details
	Description The Apex stack trace if the test failed; otherwise, `null`.
`TestTimestamp`	**Type** dateTime **Properties** Filter, Sort **Description** The start time of the test method.

Usage

You can query the fields of the ApexTestResult record that corresponds to a test method executed as part of an Apex class execution.

Each test method execution is represented by a single ApexTestResult record. For example, if an Apex test class contains six test methods, six ApexTestResult records are created. These records are in addition to the ApexTestQueueItem record that represents the Apex class.

For example code, see ApexTestQueueItem.

ApexTrigger

Represents the saved copy of an Apex trigger. ApexTrigger uses the cached version of the class unless one is unavailable. Available from API version 28.0 or later.

To edit, save, or compile Apex triggers, use ApexTriggerMember.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`ApiVersion`	**Type** double **Properties** Create, Filter, Sort, Update

Field Name	Details
	Description The API version for this trigger. Every trigger has an API version specified at creation.
`Body`	**Type** string **Properties** Create, Nillable, Update **Description** The Apex trigger definition. Limit: 1 million characters.
`BodyCrc`	**Type** double **Properties** Create, Defaulted on create, Filter, Nillable, Sort, Update **Description** The CRC (cyclic redundancy check) of the class or trigger file.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The Id of the EntityDefinition object associated with this object.
`IsValid`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether any dependent metadata has changed since the trigger was last compiled (`true`) or not (`false`).
`LengthWithoutComments`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** Length of the trigger without comments.
`ManageableState`	**Type** ManageableState enumerated list

Field Name	Details
	Properties Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** ApexTriggerMetadata **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding Apex trigger. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Status`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The current status of the Apex trigger. The following string values are valid: • `Active`—The trigger is active. • `Inactive`—The trigger is inactive, but not deleted. • `Deleted`—The trigger is marked for deletion. This is useful for managed packages, because it allows a class to be deleted when a managed package is updated. Note: `Inactive` is not valid for ApexClass. For more information, see the *Metadata API Developer Guide*.
`UsageAfterDelete`	**Type** boolean **Properties** Create, Filter, Update

Field Name	Details
	Description Specifies whether the trigger is an after delete trigger (`true`) or not (`false`).
`UsageAfterInsert`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is an after insert trigger (`true`) or not (`false`).
`UsageAfterUndelete`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is an after undelete trigger (`true`) or not (`false`).
`UsageAfterUpdate`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is an after update trigger (`true`) or not (`false`).
`UsageBeforeDelete`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is an before delete trigger (`true`) or not (`false`).
`UsageBeforeInsert`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is an before insert trigger (`true`) or not (`false`).
`UsageBeforeUpdate`	**Type** boolean

Field Name	Details
	Properties Create, Filter, Update **Description** Specifies whether the trigger is an before update trigger (`true`) or not (`false`).
`UsageIsBulk`	**Type** boolean **Properties** Create, Filter, Update **Description** Specifies whether the trigger is defined as a bulk trigger (`true`) or not (`false`).

Usage

To retrieve information about an Apex trigger, create an ApexTrigger object that references it. For example code, see SOAP Calls.

To edit, save, or compile Apex triggers, use ApexTriggerMember.

ApexTriggerMember

Represents the working copy of an Apex trigger for editing, saving, or compiling in a MetadataContainer.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Body`	**Type** string **Properties** Create, Update **Description** The data for the Apex trigger. The `Body` field is the only field you can `update()` or `PATCH`.

Field Name	Details
Content	**Type** string **Properties** None **Description** A string representation of ApexTriggerMetadata that lists the version, status, and packaged versions of the corresponding Apex trigger.
ContentEntityId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to an Apex trigger. There can be only one `ContentEntityId` per ApexTriggerMember, otherwise, an error is reported. This field is required if `FullName` is not specified.
FullName	**Type** string **Properties** Group, Nillable **Description** The full name of the associated object in the Metadata API. Use to avoid race conditions on create, before you have IDs. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance. This field is required if `ContentEntityId` is not specified.
LastSyncDate	**Type** dateTime **Properties** Filter, Sort **Description** The date that this ApexTriggerMember `Body` was replicated from the underlying entity. When you deploy a MetadataContainer, this value is compared with the `LastModifiedDate` of the underlying Apex trigger. If `LastSyncDate` is older than `LastModifiedDate`, the deployment fails with an error.

Field Name	Details
`Metadata`	**Type** ApexTriggerMetadata **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding Apex trigger. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`MetadataContainerId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** A reference to a MetadataContainer or ContainerAsyncRequest object. As part of a successful deployment, this field is reset from the ID of the deployed MetadataContainer to the ID of the corresponding ContainerAsyncRequest object. This field is required.
`SymbolTable`	**Type** SymbolTable **Properties** Nillable **Description** A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`. This field is null if the symbol table cannot be created. A symbol table can't be created if the content referenced by the `ContentEntityId` field doesn't use a symbol table. Compiler errors for the last deployment of the MetadataContainer in the `MetadataContainerId` field also prevent a symbol table from being created.

Usage

To edit, save, or compile an Apex trigger, create an ApexTriggerMember object that references it. To create a trigger, use the Force.com REST API or the Metadata API.

Note: Once an ApexTriggerMember is successfully deployed in a MetadataContainer, the `MetadataContainerId` is changed to the ID of the ContainerAsyncRequest, and the ApexTriggerMember can't be modified or reused.

Apex triggers and classes are often dependent on each other for functionality. For example, a method in one class can call a method in another class. If source file A is dependent on modified source file B and you try to save and compile source file A before you've saved the changes to source file B, the compiler will throw an error. To successfully save and compile a group of related source files, put the corresponding ApexTriggerMember and ApexClassMember objects in a single MetadataContainer object. Use ContainerAsyncRequest to send the MetadataContainer to the application server.

Each ApexTriggerMember object can only refer to a single MetadataContainer object. Multiple ApexTriggerMember objects can refer to the same MetadataContainer object.

AssignmentRule

Don't use this object.

This object is exposed in API version 35.0, however AssignmentRule is reserved for future use.

AuraDefinition

Represents a Lightning definition, such as component markup, a client-side controller, or an event. Available in API version 32.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`AuraDefinitionBundleId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the bundle containing the definition. A bundle contains a Lightning definition and all its related resources.
`DefType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field Name	Details
	Description The definition type. Valid values are: • `APPLICATION` — Lightning Components app • `CONTROLLER` — client-side controller • `COMPONENT` — component markup • `EVENT` — event definition • `HELPER` — client-side helper • `INTERFACE` — interface definition • `RENDERER` — client-side renderer • `STYLE` — style (CSS) resource • `PROVIDER` — reserved for future use • `MODEL` — deprecated, do not use • `TESTSUITE` — reserved for future use • `DOCUMENTATION` — documentation markup • `TOKENS` — tokens collection • `DESIGN` — design definition • `SVG` — SVG graphic resource
`Format`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The format of the definition. Valid values are: • `XML` for component markup • `JS` for JavaScript code • `CSS` for styles
`Source`	**Type** textarea **Properties** Create, Update **Description** The contents of the Lightning definition. This is all the markup or code for the definition.

Usage

For more information, see the Lightning Components Developer Guide.

AuraDefinitionBundle

Represents a Lightning definition bundle, such as a component or application bundle. A bundle contains a Lightning definition and all its related resources. Lightning components is a beta feature. Available in API version 32.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`ApiVersion`	**Type** double **Properties** Create, Filter, Sort, Update **Description** The API version for this bundle. Every bundle has an API version specified at creation.
`Description`	**Type** textarea **Properties** Create, Filter, Group, Sort, Update **Description** The text description of the bundle. Maximum size of 255 characters.
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The unique name of the record in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is automatically generated but you can supply your own value if you create the record using the API.

Field Name	Details
	Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, performance may slow while Salesforce generates one for each record.
`Language`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The language of the `MasterLabel`.
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** Master label for the Lightning bundle. This internal label doesn’t get translated.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field’s value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.

Usage

For more information, see the Lightning Components Developer Guide.

AutoResponseRule

Specifies whether the autoresponse rule is active (`true`).

Available in API version 35.0 and later.

Supported SOAP API Calls

`query()`

Supported REST API HTTP Methods

`Query`, `GET`

Fields

Field Name	Details
`Active`	**Type** boolean **Properties** Defaulted on create Filter, Group, Sort **Description** If `true`, the autoresponse rule is active.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** Represents the object associated with this autoresponse rule.
`Name`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Represents the name of the autoresponse rule.

Usage

Use this object to query whether an autoresponse rule is active.

```
SELECT Name, Active
FROM AutoResponseRule
```

More information about the autoresponse rule is available by querying the metadata type AutoResponseRules or AutoResponse in the metadata namespace (`mns`).

BusinessProcess

Represents a business process.

This object is available in API version 33.0 and later.

Supported SOAP Calls

`create()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

`GET`, `PATCH`, `POST`

Fields

Field	Details
`Description`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The business process description, limited to 255 characters.
`IsActive`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether this business process is active (`true`) or not (`false`).
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update

Field	Details
	Description Indicates the manageable state of the specified component that is contained in a package: - `beta` - `deleted` - `deprecated` - `installed` - `released` - `unmanaged`
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The process name.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** A unique string to distinguish this type from any others.

CompactLayout

Represents the values that define a compact page layout.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
DeveloperName	**Type** string **Properties** Filter, Group, Sort **Description** The developer's internal name for the compact layout (for example, "CL_c") used in the API.
FullName	**Type** string **Properties** Create, Group, Nillable **Description** The unique name used as the compact layout identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
MasterLabel	**Type** string **Properties** Filter, Group, Sort **Description** The name of the compact layout in Setup.
Metadata	**Type** `mns:`CompactLayout **Properties** Create, Nillable, Update **Description** The compact layout metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	**Type** string **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The namespace of the package of which the compact layout is a part.
`SobjectType`	**Type** Restricted picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The type of object used in the layout, such as an Account or Lead.

CompactLayoutItemInfo

Represents a field selected for a compact layout, and the order of that field in the compact layout.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`CompactLayoutInfo`	**Type** CompactLayoutInfo **Properties** Filter, Group, Nillable, Sort **Description** The compact layout associated with this CompactLayoutItemInfo.

Field	Details
`CompactLayoutInfoId`	**Type** `Id` **Properties** Filter, Group, Nillable, Sort **Description** ID of the compact layout associated with this field.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** This field reserved for future use. Do not use.
`FieldDefinition`	**Type** FieldDefinition on page 131 **Properties** Filter, Group, Nillable, Sort **Description** Required. The definition of this field.
`FieldDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Required. ID of this field.
`SortOrder`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The order of the field in the compact layout. `1` is first.

CompactLayoutInfo

Represents the metadata for a custom or standard compact layout.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The developer's internal name for the compact layout (for example, `CL_c`) used in the API.
`DurableId`	**Type** string **Properties** Filter, Group, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** Required. Available starting with version 32.0. The entity definition for the object associated with this CompactLayoutInfo.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description Required. ID of the record associated with this CompactLayoutInfo. The record's object type is in `EntityDefinition`.
`FullName`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name used as the compact layout identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`IsDefault`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this compact layout is the default for the associated object.
`Items`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** A foreign key field pointing to CompactLayoutItemsInfo. Because this field represents a relationship, use only in subqueries.
`Label`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The compact layout's label.
`Metadata`	**Type** `mns`: CompactLayout on page 72 **Properties** Create, Nillable, Update

Field	Details
	Description Metadata that defines compact layouts. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.

ContainerAsyncRequest

Allows you to compile and asynchronously deploy a MetadataContainer object to your organization.

Supported SOAP API Calls

`create()`, `describeSObjects()`, `query()`, `retrieve()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`

Fields

Field Name	Details
`DeployDetails`	**Type** DeployDetails **Properties** Nillable **Description** Provides detailed XML for any compile errors reported during an asynchronous request. Includes `componentFailures`. Replaces the JSON field `CompilerErrors` in Tooling API version 31.0 and later.
`ErrorMsg`	**Type** textarea **Properties** Nillable **Description** Errors reported during an asynchronous request.
`IsCheckOnly`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort **Description** Indicates whether the asynchronous request compiles the code without making any changes to the organization (`true`) or compiles and saves the code (`false`). This field is required. Note: You can compile without saving but you can’t save without compiling.
`IsRunTests`	**Type** boolean **Properties** None **Description** Reserved for future use.
`MetadataContainerId`	**Type** reference **Properties** Create, Filter, Group, Sort

Field Name	Details
	Description The ID of a MetadataContainer object. Specify a `MetadataContainerId` or a `MetadataContainerMemberId`, but not both.
`MetadataContainerMemberId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort **Description** The ID of an ApexClassMember, ApexTriggerMember, ApexPageMember or ApexComponentMember object. Specify a `MetadataContainerId` or a `MetadataContainerMemberId`, but not both.
`State`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The state of the request. Valid values are: • `Queued`—the job is in the queue. • `Invalidated`—Salesforce cancelled the job because the results might not be valid. This state occurs if someone changes the container members while `IsCheckOnly=true`, or if a newer compile request is added to the queue. • `Completed`—the compilation or deployment finished. The `SymbolTable` fields for the specified object(s) were successfully updated. If `IsCheckOnly` is `false`, the `Body` for each object was saved and the `MetadataContainerId` field for each object was reset from the ID of the deployed MetadataContainer to the ID of the corresponding ContainerAsyncRequest object. • `Failed`—the compilation or deployment failed for the reasons stated in the `CompilerError` field. • `Error`—an unexpected error occurred. The messages in the `ErrorMsg` field can be provided to Salesforce support if the issue persists. • `Aborted`—use this value to delete a queued deployment. This field is required.

Usage

When you deploy a ContainerAsyncRequest, you must specify whether to save the compiled entities:

- To compile entities without saving, set the request to `IsCheckOnly=true`. This option is only supported if a `MetadataContainerMember` is specified. A single `MetadataContainerMemberId` can't be compiled without saving.
- To compile and save entities to your organization, set the request to `IsCheckOnly=false`.

If the compile succeeds, the `SymbolTable` field is updated on each object in the specified MetadataContainer. If the save or compile fails and a `SymbolTable` field cannot be updated, the field is cleared. If there is an outstanding save request, all updates, inserts, and deployments fail.

To terminate a queued deployment, set the `State` field to `Aborted`.

CustomField

Represents a custom field on a custom object that stores data unique to your organization. Includes access to the associated CustomField object and related fields in Salesforce Metadata API. Available from API version 28.0 or later.

Supported SOAP Calls

`create()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`Query`, `GET`, `POST`, `PATCH`

Fields

Field Name	Details
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer's internal name for the custom field (for example "CF_c").
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta`

Field Name	Details
	- `deleted` - `deprecated` - `installed` - `released` - `unmanaged`
`Metadata`	**Type** CustomFieldMetadata **Properties** Create, Nillable, Update **Description** CustomFieldMetadata includes the following fields: - `caseSensitive` - `customDataType` - `defaultValue` - `deleteConstraint` - `deprecated` - `description` - `displayFormat` - `displayLocationInDecimal` - `escapeMarkup` - `externalDeveloperName` - `externalId` - `formula` - `formulaTreatBlanksAs` - `inlineHelpText` - `isFilteringDisabled` - `isNameField` - `isSortingDisabled` - `label` - `length` - `maskChar` - `maskType` - `picklist` - `populateExistingRows` - `precision` - `readOnlyProxy` - `referenceTo` - `relationshipLabel`

Field Name	Details
	• relationshipName • relationshipOrder • reparentableMasterDetai • required • restrictedAdminField • scale • startingNumber • stripMarkup • summarizedField • summaryFilterItems • summaryForeignKey • summaryOperation • trackFeedHistory • trackHistory • type • unique • visibleLines • writeRequiresMasterRead Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the custom field. A custom field can be in an extension namespace different than the object.
TableEnumOrId	**Type** Restricted picklist **Properties** Filter, Group, Sort **Description** The enum (for example, Account) or ID of the object this field is on.

CustomFieldMember

Represents the working copy of a field for editing or saving in a MetadataContainer. This object is available in API version 33.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Content`	**Type** string **Properties** None **Description** A string representation of CustomField that contains the field's metadata.
`ContentEntityId`	**Type** ID **Properties** Create, Filter, Group, Sort **Description** A reference to a custom field. There can be only one `ContentEntityId` per CustomField, otherwise, an error is reported.
`FullName`	**Type** string **Properties** Group, Nillable **Description** The full name of the associated object in the Metadata API. Use to avoid race conditions on create, before you have IDs. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field Name	Details
`IsDeleted`	**Type** boolean **Properties** Group, Nillable **Description** Indicates whether the object is marked as deleted (`true`) or not (`false`).
`LastSyncDate`	**Type** dateTime **Properties** Filter, Sort **Description** The date that this CustomField was replicated from the underlying entity.
`Metadata`	**Type** CustomField **Properties** None **Description** An object that describes the version, status, and packaged versions of the corresponding CustomField. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

CustomObject

Represents a custom object that stores data unique to your organization. Includes access to the associated CustomObject object and related fields in Salesforce Metadata API. Available from API version 31.0 or later.

Supported SOAP Calls

`query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`Query`, `GET`

Fields

Field Name	Details
`CustomHelpId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** The control that contains the help content if this custom object has customized help content.
`Description`	**Type** string **Properties** Filter, Nillable, Sort **Description** The object's description. This can be useful to describe the reason for creating the object or its intended use.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer's internal name for the custom object (for example "CF_c").
`ExternalName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Maps to a table in the external data source. If you created the external object using Validate and Sync for the data source, this name is automatically created.
`ExternalRepository`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Maps to a table in the external data source. If you created the external object using Validate and Sync for the data source, this name is automatically created; do not modify it.

Field Name	Details
`Language`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** The language of the action. Valid values are: - Chinese (Simplified): `zh_CN` - Chinese (Traditional): `zh_TW` - Danish: `da` - Dutch: `nl_NL` - English: `en_US` - Finnish: `fi` - French: `fr` - German: `de` - Italian: `it` - Japanese: `ja` - Korean: `ko` - Norwegian: `no` - Portuguese (Brazil): `pt_BR` - Russian: `ru` - Spanish: `es` - Spanish (Mexico): `es_MX` - Swedish: `sv` - Thai: `th`
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - `beta` - `deleted` - `deprecated` - `installed` - `released` - `unmanaged`

Field Name	Details
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package of which the custom object is a part.
`SharingModel`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The sharing model. Values are: • Edit • ControlledByparent • None • Read

CustomTab

Represents a custom tab.

This object is available in the Tooling API version 33.0 and later.

Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Fields

Field	Details
`ContentId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** Read-only. The ID of the item that the custom tab points to. For Lightning components, this is the ID of the component bundle. For custom object tabs, this field is `null`.

Field	Details
`Description`	**Type** string **Properties** Filter, Nillable, Sort **Description** The tab's description.
`DeveloperName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The developer's internal name for the custom tab.
`EncodingKey`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Read-only. Type of encoding assigned to the URL called by the tab. The default encoding setting is Unicode: `UTF-8`. Change it if you are passing information to a URL that requires data in a different format. This option is available when the value `URL` is selected in the tab type. Valid values are: - `UTF-8`—Unicode (UTF-8) - `ISO-8859-1`—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) - `Shift_JIS`—Japanese (Shift-JIS) - `ISO-2022-JP`—Japanese (JIS) - `EUC-JP`—Japanese (EUC-JP) - `x-SJIS_0213`—Japanese (Shift-JIS_2004) - `ks_c_5601-1987`—Korean (ks_c_5601-1987) - `Big5`—Traditional Chinese (Big5) - `GB2312`—Simplified Chinese (GB2312) - `Big5-HKSCS`—Traditional Chinese Hong Kong (Big5–HKSCS)
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The name of the tab. The value of this field depends on the type of tab, and the API version.

Field	Details
	- For custom object tabs, the `fullName` is the developer-assigned name of the custom object (MyCustomObject__c, for example). - For Web tabs, the `fullName` is the developer-assigned name of the tab (MyWebTab, for example). Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`HasSidebar`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates if the tab displays the sidebar panel.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - `beta` - `deleted` - `deprecated` - `installed` - `released` - `unmanaged`
`MasterLabel`	**Type** string **Properties** Filter, Group, idLookup, Nillable, Sort **Description** Required. The label for the custom tab, which displays in Setup.
`Metadata`	**Type** CustomTabMetadata **Properties** Create, Nillable, Update **Description** Custom tab metadata.

Field	Details
	Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`MotifName`	**Type** string **Properties** Filter, Group, Sort **Description** Read-only. The name of the tab style assigned to the custom tab.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package of which the custom tab is a part.
`Type`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The type of custom tab. Valid values are: • `apexPage` • `aura` • `customObject` • `flexiPage` • `sControl` • `url`
`Url`	**Type** string **Properties** Filter, Nillable, Sort **Description** The URL for the external web-page to embed in this tab.

DataType

Represents the datatype of a field. Use this object with EntityDefinition, EntityParticle, or FieldDefinition to simplify queries. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`ContextServiceDataTypeId`	Don't use this field. It's reserved for future use. Properties and behavior are likely to change.
`ContextWsdlDataTypeId`	Don't use this field. It's reserved for future use. Properties and behavior are likely to change.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.

Field	Details
`IsComplex`	**Type** **boolean** **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the datatype contains other datatypes, in contrast to a simple datatype like string.

Example

From an object, retrieve all the fields of one datatype.

```
SELECT DataType, QualifiedApiName
FROM EntityParticle
WHERE DataType = 'phone' AND
      EntityDefinition.QualifiedApiName = 'Account'
```

SOQL Limitations

This object doesn't support some SOQL operations.

GROUP BY

Example Query: `SELECT COUNT(qualifiedapiname), isfeedenabled FROM EntityDefinition GROUP BY isfeedenabled`

Error Returned: `The requested operation is not yet supported by this SObject storage type, contact salesforce.com support for more information.`

LIMIT, LIMIT OFFSET

Example Queries:

`SELECT qualifiedapiname FROM EntityDefinition LIMIT 5`

`SELECT qualifiedapiname FROM EntityDefinition LIMIT 5 OFFSET 10`

An incorrect result is returned because LIMIT and LIMIT OFFSET are ignored.

NOT

Example Query: `SELECT qualifiedapiname FROM EntityDefinition WHERE qualifiedapiname!='Account'`

Error Returned: `Only equals comparisons permitted`

OR

Example Query: `SELECT qualifiedapiname, keyprefix FROM EntityDefinition WHERE isdeletable=true OR (isfeedenabled=false AND keyprefix='01j')`

Error Returned: `Disjunctions not supported`

DebugLevel

Represents a set of log category levels to assign to a `TraceFlag` object. Multiple trace flags can use a debug level.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`ApexCode`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for Apex code. Includes information about Apex code and can include information such as log messages generated by data manipulation language (DML) statements, inline SOQL or SOSL queries, the start and completion of any triggers, the start and completion of any test method, and so on. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`ApexProfiling`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field Name	Details
	Description The log category level for profiling information. Includes cumulative profiling information, such as the limits for your namespace, the number of emails sent, and so on. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`Callout`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for callouts. Includes the request-response XML that the server is sending and receiving from an external Web service. The request-response XML is useful when debugging issues related to SOAP API calls. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`Database`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field Name	Details
	Description The log category for database activity. Includes information about database activity, including every DML statement or inline SOQL or SOSL query. The following are valid values. - `NONE` - `ERROR` - `WARN` - `INFO` - `DEBUG` - `FINE` - `FINER` - `FINEST` This field is required.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The developer's internal name for the debug level. Also displays in the Developer Console and in Setup.
`Language`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The language of the `MasterLabel`. Valid values are: - Chinese (Simplified): `zh_CN` - Chinese (Traditional): `zh_TW` - Danish: `da` - Dutch: `nl_NL` - English: `en_US` - Finnish: `fi` - French: `fr` - German: `de` - Italian: `it` - Japanese: `ja` - Korean: `ko`

Field Name	Details
	- Norwegian: `no` - Portuguese (Brazil): `pt_BR` - Russian: `ru` - Spanish: `es` - Spanish (Mexico): `es_MX` - Swedish: `sv` - Thai: `th`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** Reserved for future use. However, this field is required and must contain a value. We suggest that you use the same value used for `DeveloperName`.
`System`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for calls to all system methods, such as the `System.debug` method. The following are valid values. - `NONE` - `ERROR` - `WARN` - `INFO` - `DEBUG` - `FINE` - `FINER` - `FINEST` This field is required.
`Validation`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field Name	Details
	Description The log category level for validation rules. Includes information about validation rules, such as the name of the rule, or whether the rule evaluated true or false. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`Visualforce`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for Visualforce. Includes information about Visualforce events, including serialization and deserialization of the view state or the evaluation of a formula field in a Visualforce page. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`Workflow`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for workflow rules. Includes information for workflow rules, such as the rule name and the actions taken. This field is required. The following are valid values.

Field Name	Details
	• `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST`

Usage

If you delete a debug level, all the trace flags that use it are deleted.

DeployDetails

A complex type that contains detailed XML for any compile errors reported in the asynchronous request defined by a ContainerAsyncRequest object. Replaces the JSON field `CompilerErrors` in Tooling API version 31.0 and later.

Fields

Field	Details
`componentFailures`	**Type** string **Description** The line number, component name and a short description for any compile errors. For example:

```
<DeployDetails>
    <componentFailures>
        <lineNumber>5</lineNumber>
        <fullName>myApex</fileName>
        <problem>invalid name 'abc'</problem>
    </componentFailures>
    <componentFailures>
        <lineNumber>10</lineNumber>
        <fullName>myApex2</fileName>
        <problem>invalid type 'hello'</problem>
    </componentFailures>
</DeployDetails>
```

EmailTemplate

Represents an email template.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`ApiVersion`	**Type** double **Properties** Filter, Nillable, Sort **Description** The API version if this is a Visualforce email template. Every Visualforce email template has an API version specified at creation.
`Description`	**Type** string **Properties** Filter, Nillable, Sort **Description** The email template description. This can be useful to describe the reason for creating the template or its intended use.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The unique name used as the template identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field	Details
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** EmailTemplateMetadata **Properties** Create, Nillable, Update **Description** Email template metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The email template name.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** A unique string to distinguish this template from any others. For example, if this template is being using by a flow, use the `NamespacePrefix` to uniquely identify the templates in multiple flow instances.
`Subject`	**Type** string

Field	Details
	Properties Group, Nillable, Sort **Description** The email subject.

EntityDefinition

Provides row-based access to metadata about standard and custom objects.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`, `search()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`ApexTriggers`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the Apex triggers associated with this object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`AssignmentRules`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description Represents assignment rules that allow you to automatically route cases to the appropriate users or queues. Available in Tooling API version 34.0 and later. Because this field represents a relationship, use only in subqueries.
`AutoResponseRules`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the auto-response rules defined for the object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`BusinessProcesses`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the business processes defined for the object. Business processes display different picklist values for users based on their profile and associated record type. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`ChildRelationships`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the child relationships defined for the object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`CompactLayouts`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the compact layouts defined for the object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`CustomFields`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description Represents the custom fields defined for the object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`DefaultCompactLayout`	**Type** CompactLayoutInfo **Properties** Create, Nillable, Update **Description** Metadata about the compact layout defined as the default for this object, if any.
`DefaultCompactLayoutId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the default compact layout, if any.
`DetailUrl`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** URL to the read-only detail page for this object. Corresponds to the `urlDetail` field in DescribeSobjectResult. Available in Tooling API starting version 34.0.
`DeveloperName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The developer's internal name for the custom object (for example `CF_c`).
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. Simplify queries by using this field instead of making multiple queries.

Field	Details
`EditDefinitionUrl`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Available in Tooling API starting version 34.0.
`EditUrl`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The URL used when editing the custom entity definition. Corresponds to the `urlEdit` field on DescribeSobjectResult. Available in Tooling API starting version 34.0.
`FieldSets`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the field sets defined for the object. Because this field represents a relationship, use only in subqueries.
`Fields`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the standard and custom fields defined for this object. Because this field represents a relationship, use only in subqueries.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The name of the entity. If a field, the name must specify the parent object, for example `Account.FirstName`. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field	Details
HelpSettingPageName	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the custom help setting page. Available in Tooling API starting version 34.0.
HelpSettingPageUrl	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The URL pointing to the custom help setting page for custom objects. Available in Tooling API starting version 34.0.
IsApexTriggerable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, Apex triggers can be defined for the entity.
IsCompactLayoutable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object supports compact layouts. That is, compact layouts can be defined, a system compact layout can be synthesized, or both.
IsCreatable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, records based on the object (`true`) can be created. This field is unavailable starting with version 35.0. Use `IsCreatable` on UserEntityAccess instead.
IsCustomSetting	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** If `true`, the object is a custom setting. Available in Tooling API starting version 35.0.
`IsCustomizable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, custom fields can be defined for the entity.
`IsDeletable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be deleted. This field is unavailable starting with version 35.0. Use `IsDeletable` on UserEntityAccess instead.
`IsDeprecatedAndHidden`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this object is unavailable for the current version. Available in Tooling API starting version 35.0.
`IsEverCreatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be created using an API, assuming the current user has the appropriate permissions. If false, the application server manages the object and no user can create it. Available in Tooling API starting version 35.0.
`IsEverDeletable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, the object can be created using an API, assuming the current user has the appropriate permissions. If false, the application server manages the object and no user can delete it. Available in Tooling API starting version 35.0.
`IsEverUpdatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be created using an API, assuming the current user has the appropriate permissions. If false, the application server manages the object and no user can modify it. Available in Tooling API starting version 35.0.
`IsFeedEnabled`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the Chatter feed is enabled for this object. Available in Tooling API starting version 34.0.
`IsFlsEnabled`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can set field-level security on applicable fields. Available in Tooling API starting version 35.0.
`IsIdEnabled`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can include `Id` in the SELECT clause of a query on this object. Available in Tooling API starting version 35.0. For example, assume that there is an object backed by an OData data source with the High Data Volume option selected. That object's `IsIdEnabled` field is `false`.
`IsLayoutable`	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** If `true`, layouts can be defined for the object. Available in Tooling API starting version 35.0.
`IsQueryable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be queried.
`IsReplicateable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be replicated. Available in Tooling API starting version 35.0.
`IsRetrieveable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the object can be retrieved. Available in Tooling API starting version 35.0.
`IsSearchable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, records of this object are indexed for search. Available in Tooling API starting version 35.0.
`IsSearchLayoutable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can customize search layouts for this object. Available in Tooling API starting version 35.0.

Field	Details
IsTriggerable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can use triggers with this object. Available in Tooling API starting version 35.0.
IsWorkflowEnabled	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, workflow rules can be defined for the entity.
KeyPrefix	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The first three digits of the entity's ID, which identify the object type, such as Account or Opportunity.
Label	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The label for this object, used in the compact layout and in the user's language locale.
Layouts	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the layouts defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
Limits	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The limits defined for this object. Corresponds to the Limits page for each standard object in Setup, or the Limits related list for each custom object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`LookupFilters`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the lookup filters defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** The label for this object, which displays in Setup. The master label is in the default language locale for the organization. If there is no default language locale, the label is in `en_US`.
`Metadata`	**Type** `mns:` CustomObject **Properties** Create, Nillable, Update **Description** Metadata about the standard or custom object. Details are available in the CustomObject entry in the metadata namespace in the Tooling API WSDL. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values:

Field	Details
	• In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`NewUrl`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The URL used when editing a new standard or custom record. Corresponds to the `urlNew` field on DescribeSobjectResult. Available in Tooling API starting version 34.0.
`OwnerChangeOptions`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Use only in subqueries. Available in Tooling API starting version 35.0. Because this field represents a relationship, use only in subqueries.
`Particles`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** The particles defined for this object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`PluralLabel`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The plural version of the object's `Label`.
`Publisher`	**Type** Publisher

Field	Details
	Properties Create, Nillable, Update **Description** The publisher of this object, for example Salesforce, a user, or a package name. Available in Tooling API starting version 34.0.
`PublisherId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the publisher associated with this object. Available in Tooling API starting version 34.0.
`QualifiedApiName`	**Type** string **Properties** Filter, Group, Sort **Description** A unique external ID for the entity of the form `NamespacePrefix__DeveloperName` for standard objects and `NamespacePrefix__DeveloperName__c` for custom objects.
`QuickActionDefinitions`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the quick actions defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`RecordTypes`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the record types defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`RecordTypesSupported`	**Type** RecordTypesSupported on page 116 **Properties** Nillable

Field	Details
	Description Represents the record types defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0.
`RelationshipDomains`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Metadata about the relationships with other objects that this object has. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`RunningUserEntityAccess`	**Type** UserEntityAccess **Properties** Create, Nillable, Update **Description** Represents the running user's access to this object. Available in Tooling API starting version 34.0.
`RunningUserEntityAccessId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the UserEntityAccess record associated with this object. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`SearchLayouts`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the search layouts associated with this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`StandardActions`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description Represents the standard actions defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`ValidationRules`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the validation rules defined for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`WebLinks`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the Weblinks associated with this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`WorkflowAlerts`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the workflow alerts associated with this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`WorkflowFieldUpdates`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the workflow field updates for this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`WorkflowOutboundMessages`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description Represents the workflow outbound messages associated with this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`WorkflowTasks`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Represents the workflow tasks associated with this object. Use only in subqueries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.

RecordTypesSupported Metadata

`RecordTypesSupported` is in the `tns` namespace. Represents the record types associated with this object.

Field	Details
`recordTypeInfos`	**Type** RecordTypeInfo **Description** Represents the RecordTypeInfo records for the object. Use only in subqueries. Available in Tooling API starting version 35.0.

RecordTypeInfo Metadata

`RecordTypeInfo` is in the `tns` namespace. Represents a record type associated with the object.

Field	Details
`available`	**Type** boolean **Description** If `true`, this record type is available for use. Available in Tooling API starting version 35.0.
`defaultRecordTypeMapping`	**Type** boolean **Description** Available in Tooling API starting version 35.0.

Field	Details
`master`	**Type** boolean **Description** Available in Tooling API starting version 35.0.
`name`	**Type** string **Description** Name of the record type. Available in Tooling API starting version 35.0.
`recordTypeId`	**Type** Id **Description** ID of the record type. Available in Tooling API starting version 35.0.

EntityLimit

Represents the limits for an object as displayed in the Setup UI.

This object is available in API version 34.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`DurableId`	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn’t guaranteed to stay the same from one release to the next. Use this field to simplify queries.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** The object to which these limits apply.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** ID of the object to which these limits apply.
`Label`	**Type** string **Properties** Filter, Group, Sort **Description** The label of the object to which these limits apply.
`Max`	**Type** int **Properties** Filter, Group, Sort **Description** The maximum number of objects that the organization is allowed to have.
`Remaining`	**Type** int **Properties** Filter, Group, Sort **Description** The number of objects still available. For example, if the limit on custom objects is 100, and you create 75, this value is `25`.

Field	Details
Type	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** What type of component the limit applies to: • ActiveLookupFilters • ActiveRules • ActiveValidationRules • ApprovalProcesses • CbsSharingRules • CustomFields • CustomRelationship • RollupSummary • SharingRules • TotalRules • VLookup

EntityParticle

Represents each element of a field that can be presented in a user interface. Contrast EntityParticle with FieldDefinition, which represents each element of a field defined in the Metadata API. EntityParticle has parity with `describe`. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`ByteLength`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The maximum length of the field represented by this EntityParticle, in bytes.

`DataType`	**Type**
	string
	Properties
	Filter, Group, Nillable, Sort
	Description
	Data type of the field, for example `Text(40)` or `Date/Time`. The values are defined as they are in the user interface, not the corresponding API data type names.
	For example, from an object, retrieve all the fields of one datatype.
	`SELECT DataType, QualifiedApiName`
	`FROM EntityParticle`
	`WHERE DataType = 'phone' AND`
	`      EntityDefinition.QualifiedApiName = 'Account'`

`DefaultValueFormula`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The default value specified for the field when a formula isn't specified. If no default value has been specified, this field is not returned.
`DeveloperName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.

Field	Details
`Digits`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Maximum number of digits for a field of type int. If an integer value exceeds the number of digits, the API returns an error.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The ID for the object defined in the `DurableId` field.
`ExtraTypeInfo`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Represents further definition of a type. • For type textarea: – `plaintextarea` – `richtextarea` • For type URL: – `image` • For type reference: – `externallookup` – `indirectlookup` • For Account: – `switchablepersonname`

Field	Details
	- `personname`
`FieldDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the field definition associated with this EntityParticle.
`InlineHelpText`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Represents the content of field-level help. **Restrictions** Available in Tooling API starting version 35.0.
`IsApiFilterable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field represented by this EntityParticle can be specified in the `WHERE` clause of a query string. **Restrictions** You can't sort or filter compound fields. This field's value is always `false` for compound fields.
`IsApiGroupable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field represented by this EntityParticle can be included in the GROUP BY clause of a SOQL query.
`IsApiSortable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, a query can sort on the field represented by this EntityParticle. **Restrictions** You can't sort or filter compound fields. This field's value is always `false` for compound fields.
`IsAutoNumber`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field's value is automatically assigned when the record is created. **Restrictions** Available in Tooling API starting version 35.0.
`IsCalculated`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field represented by this EntityParticle is calculated.
`IsCaseSensitive`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field represented by this EntityParticle is case sensitive. **Restrictions** Available in Tooling API starting version 35.0.
`IsCompactLayoutable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be included in a compact layout.
`IsCreatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, a value for the field represented by this EntityParticle can be created. **Restrictions** Available in Tooling API starting version 35.0.
`IsDefaultedOnCreate`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, and if no other value is supplied, a default value is applied when the record is created. **Restrictions** Available in Tooling API starting version 35.0.
`IsDependentPicklist`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is a dependent picklist. **Restrictions** Available in Tooling API starting version 35.0.
`IsDeprecatedAndHidden`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Reserved for future use.
`IsDisplayLocationInDecimal`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, and if the field represented by this EntityParticle is a Geolocation custom field, the value appears in decimal notation. If `false`, the value appears as degrees, minutes, and seconds. **Restrictions** This field has no affect on custom fields that aren’t Geolocation fields. Available in Tooling API starting version 35.0.

Field	Details
IsEncrypted	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is marked for Classic Encryption. **Restrictions** Available in Tooling API starting version 35.0.
IsFieldHistoryTracked	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field's history can be tracked.
IsHighScaleNumber	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the field stores numbers to 8 decimal places regardless of what's specified in the field details (`true`) or not (`false`). Used to handle currencies for products that cost fractions of a cent, in large quantities. If high-scale unit pricing isn't enabled in your organization, this field isn't returned.
IsHTMLFormatted	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field contains HTML.
IsIdLookup	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can use the field to specify a record for upsert. **Restrictions** Available in Tooling API starting version 35.0.

Field	Details
IsLayoutable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be assigned to a layout. **Restrictions** Available in Tooling API starting version 35.0.
IsListVisible	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be included in a related list.
IsNameField	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is a name field.
IsNamePointing	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field represents a polymorphic relationship. Determine the object type at runtime for dynamic queries when this value is `true`. **Restrictions** Available in Tooling API starting version 35.0.
IsNillable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be left out of queries on the object.

Field	Details
`IsPermissionable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can specify field permissions for the field. **Restrictions** Available in Tooling API starting version 35.0.
`IsUnique`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is unique. **Restrictions** Available in Tooling API starting version 35.0.
`IsUpdatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can update the field. **Restrictions** Available in Tooling API starting version 35.0.
`IsWorkflowFilterable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be filtered for a workflow.
`IsWriteRequiresMasterRead`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, writing to the detail object requires read sharing instead of read-write sharing of the parent.

Field	Details
	Restrictions Available in Tooling API starting version 35.0.
`Label`	**Type** string **Properties** Filter, Group, Sort **Description** The label that corresponds to the field in the user interface. If the label has translations, the label returned is in the user's language.
`Length`	**Type** int **Properties** Filter, Group, Sort **Description** The maximum number of bytes available to store the value in the field represented by this EntityParticle.
`Mask`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Reserved for future use.
`MaskType`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Reserved for future use.
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** Master label for this object. This display value is the internal label that is not translated. Limit: 40 characters.

Field	Details
`Name`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Name of the object for the field represented by this EntityParticle. **Restrictions** This field is available in API version 35.0 and later.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`Precision`	**Type** int **Properties** Filter, Group, Sort **Description** The maximum number of digits allowed for the field represented by this EntityParticle.
`QualifiedApiName`	**Type** string **Properties** Filter, Group, Sort **Description** A unique external name for the field.

Field	Details
`ReferenceTargetField`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Specifies the custom field on the parent object to match against this indirect lookup relationship field, whose values come from an external data source. The specified custom field on the parent object must have both `IsIdLookup` and `IsUnique` set to `true`. **Restrictions** Available only if the field represented by this EntityParticle is an indirect lookup relationship field on an external object. This field is available in API version 35.0 and later.
`ReferenceTo`	**Type** RelationshipReferenceTo on page 131 **Properties** Nillable **Description** The array of values in this field represents the possible object types of the referenced objects. For example, if EntityParticle represents a field on Events, the values are `Contact`, `Lead`, and custom objects with a relationship to Events.
`RelationshipName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** If the field represented by this EntityParticle is a master-detail relationship field, this field's value is the relationship name.
`RelationshipOrder`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** A value of `0` represents a primary relationship, and a value of `1` represents a secondary relationship. This field is relevant only for relationship fields.
`Scale`	**Type** int **Properties** Filter, Group, Sort

Field	Details
	Description The number of digits to the right of the decimal in an integer. For example, 3.00 has a scale of 2.
`ValueTypeId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the value type, if any, for the field represented by this EntityParticle.

RelationshipReferenceTo Type

Represents the set of objects that this EntityParticle can have a relationship with.

Field	Type	Details
`referenceTo`	string[]	The name of an object that can be referenced. For example, if EntityParticle represents Event.WhoId (the `WhoId` field on Event), the value of this field would be at least [`Contact, Lead`].

Example Query: Retrieve All Account Fields

```
SELECT DataType, FieldDefinition.QualifiedApiName
FROM EntityParticle
WHERE EntityDefinition.QualifiedApiName ='Account'
```

Example Query: Find Parent Object Types

Some fields can have more than one type of object in a relationship (polymorphic). For example, Task and Event can have relationships with Contact or Lead.

Use this query to retrieve a list of objects that can have a relationship with Event.WhoId (represented by EntityParticle).

```
SELECT QualifiedApiName, RelationshipName, ReferenceTo, ReferenceTargetField
FROM EntityParticle
WHERE EntityDefinition.QualifiedApiName = 'Event' AND QualifiedApiName = 'WhoId'
```

FieldDefinition

Represents a standard or custom field, providing row-based access to field metadata. Contrast FieldDefinition with EntityParticle, which represents each element of a field that can be presented in a user interface. FieldDefinition has parity with metadata type Field.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details

`CompactLayoutItems`	**Type**
	QueryResult
	Properties
	Filter, Group, Sort
	Description
	The CompactLayoutItemInfo records associated with this field definition. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
	Example subquery:
	`SELECT Id, QualifiedApiName, (SELECT DurableId, SortOrder FROM`
	`CompactLayoutItems) FROM`
	`FieldDefinition WHERE EntityDefinition.QualifiedApiName`
	`= 'Account' AND QualifiedApiName = 'Name'`

`ControlledFields`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** The controlled fields in a dependent picklist. Use in subqueries to reduce the number of queries. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`ControllingFieldDefinition`	**Type** FieldDefinition

Field	Details
	Properties Filter, Group, Sort **Description** The field definition of the controlling field if this field is a dependent picklist. A dependent picklist works with a controlling picklist or checkbox to filter the available options. The value chosen in the controlling field affects the values available in the dependent field. This field is available in API version 14.0 and later. Available in Tooling API starting version 34.0.
`ControllingFieldDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the `ControllingFieldDefinition` for this field. Available in Tooling API starting version 34.0.
`DataType`	**Type** string **Properties** Filter, Group, Sort **Description** Data type of the field, for example `Text(40)` or `Date/Time`. The values are defined as they are in the user interface, not the corresponding API data type names. Available in Tooling API starting version 34.0.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`DurableId`	**Type** string **Properties** Filter, Group, Sort

Field	Details
	Description Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.

`EntityDefinition`	**Type**
	EntityDefinition
	Properties
	Filter, Group, Sort
	Description
	A relationship lookup to the object type that contains this field. For example, if the field is defined on an account, the lookup is to `Account`. You can't interact directly with this field. Instead, use it in queries.
	` SELECT EntityDefinition.Label FROM`
	`      FieldDefinition WHERE EntityDefinition.QualifiedApi`
	`Name='Lead'`

`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The durable ID for the object defined in the `EntityDefinition` field.
`ExtraTypeInfo`	**Type** string **Properties** Filter, Group, Sort **Description** Represents further definition of a type. Available in Tooling API starting version 34.0. • For type textarea: – `plaintextarea` – `richtextarea` • For type URL, `image` • For type reference, – `externallookup` – `indirectlookup` – `externallookup` • For Account, – `switchablepersonname` – `personname`

Field	Details
FullName	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
IsApiFilterable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be specified in the `WHERE` clause of a query string. Available in Tooling API starting version 34.0. You can't sort or filter compound fields. This field's value is always `false` for compound fields.
IsApiGroupable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be included in the GROUP BY clause of a SOQL query. Available in Tooling API starting version 34.0.
IsApiSortable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, a query can sort on this field. Available in Tooling API starting version 34.0. You can't sort or filter compound fields. This field's value is always `false` for compound fields.
IsCalculated	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, the field's value is calculated. Available in Tooling API starting version 34.0.
`IsCompactLayoutable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be included in a compact layout. Available in Tooling API starting version 34.0.
`IsFieldHistoryTracked`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field's history can be tracked. Available in Tooling API starting version 34.0.
`IsFlsEnabled`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, you can set field-level security on this field. Available in Tooling API starting version 35.0.
`IsHighScaleNumber`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the field stores numbers to 8 decimal places regardless of what's specified in the field details (`true`) or not (`false`). Used to handle currencies for products that cost fractions of a cent, in large quantities. If high-scale unit pricing isn't enabled in your organization, this field isn't returned. Available in Tooling API starting version 34.0.
`IsHtmlFormatted`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field contains HTML. Available in Tooling API starting version 34.0.

Field	Details
IsIndexed	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is indexed in the database. Available in Tooling API version 35.0 and later. Internal (database) indexing is different from indexing for search. We recommend targeting indexed fields for better response times in SOQL queries, reports, and list views.
IsListFilterable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be filtered for a related list. Available in Tooling API starting version 34.0.
IsListSortable	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be sorted for a related list. Available in Tooling API starting version 34.0.
IsListVisible	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be included in a related list. Available in Tooling API starting version 34.0.
IsNameField	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field is a name field. Available in Tooling API starting version 34.0.
IsNillable	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be left out of queries on the object. Available in Tooling API starting version 34.0.
`IsWorkflowFilterable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the field can be filtered for a workflow. Available in Tooling API starting version 34.0.
`Label`	**Type** string **Properties** Filter, Group, Sort **Description** The label that corresponds to the field in the user interface. If the label has been translated, the value returned is in the user's language.
`Length`	**Type** int **Properties** Filter, Group, Sort **Description** The maximum number of bytes available to store the value in this field. Available in Tooling API starting version 34.0.

`LookupFilters`	**Type**
	QueryResult
	Properties
	Filter, Group, Nillable, Sort
	Description
	The lookup filters associated with the field. Because this field represents a relationship, use only in subqueries.
	Example subquery:
	`SELECT DurableId, QualifiedApiName, (SELECT Id, SourceObject,`
	`     SourceFieldDefinition.Label, IsOptional, Active, Developer`
	`Name, LastModifiedBy.Name,`
	`     LastModifiedDate FROM LookupFilters) FROM EntityDefinition`

Field	Details

	`WHERE QualifiedApiName =`
	`'User'"`
	Note: LookupFilter is not supported on the article type object.

`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** Master label for this object. This display value is the internal label that is not translated. Limit: 40 characters.
`Metadata`	**Type** CustomField **Properties** Create, Nillable, Update **Description** Compact layout metadata, from the `mns` namespace. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.

Field	Details
Particles	**Type** QueryResult **Properties** Filter, Group, Sort **Description** The EntityParticles associated with this field. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
Precision	**Type** int **Properties** Filter, Group, Sort **Description** The maximum number of digits allowed for this field. Available in Tooling API starting version 34.0.
Publisher	**Type** Publisher **Properties** Filter, Group, Sort **Description** The publisher of this field, for example Salesforce, a user, or a package name. Available in Tooling API starting version 34.0.
PublisherId	**Type** string **Properties** Filter, Group, Sort **Description** ID of the publisher associated with this field. Available in Tooling API starting version 34.0.
QualifiedApiName	**Type** string **Properties** Filter, Group, Sort **Description** A unique external name for the field.
ReferenceTargetField	**Type** string **Properties** Filter, Group, Sort

Field	Details
	Description Available only for indirect lookup relationship fields on external objects. Specifies the field on the parent object to match against this indirect lookup relationship field, whose values come from an external data source. The specified custom field on the parent object must have both `externalId` and `unique` set to `true`.Available in Tooling API starting version 34.0.
`ReferenceTo`	**Type** RelationshipReferenceTo on page 151 **Properties** Filter, Group, Sort **Description** The array of values in this field represents the possible object types of the referenced objects. For example, if a FieldDefinition represents a field on Event.WhoId, the values in this field are `Contact`, `Lead`, and custom objects with a relationship to Events. Available in Tooling API starting version 34.0.
`RelationshipDomains`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** Metadata about the relationships with other objects that this field has. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`RelationshipName`	**Type** string **Properties** Filter, Group, Sort **Description** The value for one-to-many relationships. For example, in the object MyObject with a relationship to YourObject, the relationship name is typically YourObjects. Available in Tooling API starting version 34.0.
`RunningUserFieldAccessId`	**Type** string **Properties** **Description** Don’t use this field. Available in Tooling API starting version 34.0.
`Scale`	**Type** int

Field	Details
	Properties Filter, Group, Sort **Description** The number of digits to the right of the decimal in an integer. For example, 3.00 has a scale of 2. Available in Tooling API starting version 34.0.
`ServiceDataType`	**Type** DataType **Properties** Filter, Group, Sort **Description** The service datatype for this field. Available in Tooling API starting version 34.0.
`ServiceDataTypeId`	**Type** string **Properties** Filter, Group, Sort **Description** ID of the ServiceDataType. Available in Tooling API version 34.0. Do not use, provided for backward compatibility only.
`ServiceDataTypes`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** The ServiceDataTypes associated with this field. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.
`ValueType`	**Type** DataType on page 92 **Properties** Filter, Group, Sort **Description** The datatype of the field. Available in Tooling API version 35.0.
`ValueTypeId`	**Type** string **Properties** Filter, Group, Sort **Description** ID of the ValueType. Available in Tooling API version 35.0.

Field	Details
`WorkflowFieldUpdates`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** The workflow field updates for this field. Field updates allow you to automatically update a field value to one that you specify when a workflow rule is triggered. Available in Tooling API starting version 34.0. Because this field represents a relationship, use only in subqueries.

Example Query: Find Parent Object Types

Some fields can have more than one type of object in a relationship (polymorphic). For example, Task and Event can have relationships with Contact or Lead.

Use this query to retrieve a list of objects that can have a relationship with the FieldDefinition.

```
SELECT QualifiedApiName, RelationshipName, ReferenceTo, ReferenceTargetField
FROM FieldDefinition
WHERE EntityDefinition.QualifiedApiName = 'Event' AND QualifiedApiName = 'WhoId'
```

Example Query: Find the Datatypes for Account Fields

```
SELECT QualifiedApiName, (Select DataType From Particles)
FROM FieldDefinition
WHERE EntityDefinition.QualifiedApiName ='Account'
```

CustomField Metadata

Field Name	Field Type	Description
`caseSensitive`	boolean	Indicates whether the field is case sensitive (`true`) or not (`false`). For indirect lookup relationship fields on external objects, this attribute affects how this custom field's values are matched against the values of `referenceTargetField`.
`defaultValue`	string	If specified, represents the default value of the field.
`deleteConstraint`	DeleteConstraint is an enumeration of strings	Provides deletion options for lookup relationships. Valid values are: **`SetNull`** Default value. If the lookup record is deleted, the lookup field is cleared. **`Restrict`** If the record is in a lookup relationship, prevents it from being deleted.

Field Name	Field Type	Description
		Cascade Deletes the lookup record and associated lookup fields. For more information on lookup relationships, see "Object Relationships" in the Salesforce Help.
description	string	Description of the field.
displayFormat	string	The display format.
displayLocationInDecimal	boolean	Indicates how the value of a Geolocation custom field appears in the user interface. If true, the geolocation appears in decimal notation. If false, the geolocation appears as degrees, minutes, and seconds.
externalDeveloperName	string	Available only for external objects. Name of the table column on the external data source that maps to this custom field in Salesforce. Corresponds to External Column Name in the user interface. This field is available in API version 32.0 and later.
externalId	boolean	Indicates whether the field is an external ID field (true) or not (false).
formula	string	If specified, represents a formula on the field.
formulaTreatBlankAs	TreatBlanksAs	Indicates how to treat blanks in a formula. Valid values are BlankAsBlank or BlankAsZero.
fullName	string	Required. The internal name of the object. White spaces and special characters are escaped for validity. The name must: • Contain characters, letters, or the underscore (_) character • Must start with a letter • Can't end with an underscore • Can't contain two consecutive underscore characters.
indexed	boolean	Indicates if the field is indexed. If this field is unique or the externalId is set true, the isIndexed value is set to true. This field has been deprecated as of version 14.0 and is only provided for backward compatibility.
inlineHelpText	string	Represents the content of field-level help. For more information, see "Define Field-Level Help" in the Salesforce Help.
isFilteringDisabled	boolean	Available only for external objects. Indicates whether the custom field is available in filters. This field is available in API version 32.0 and later.
isNameField	boolean	Available only for external object fields of type text. For each external object, you can specify one field as the name field. If you set this field's value to true, make sure that the external table column identified by externalDeveloperName contains name values. This field is available in API version 32.0 and later.
isSortingDisabled	boolean	Available only for external objects. Indicates whether the custom field is sortable. This field is available in API version 32.0 and later.

Field Name	Field Type	Description
reparentableMasterDetail	boolean	Indicates whether the child records in a master-detail relationship on a custom object can be reparented to different parent records. The default value is false. This field is available in API version 25.0 and later.
label	string	Label for the field. You cannot update the label for standard picklist fields, such as the Industry field for accounts.
length	int	Length of the field.
lookupFilter	LookupFilter	The Lookup filter definition for the custom field. Available in API version 30.0 and later.
maskChar	EncryptedFieldMaskChar	For encrypted fields, specifies the character to be used as a mask. Valid values are: • asterisk • X
maskType	EncryptedFieldMaskType	For encrypted text fields, specifies the format of the masked and unmasked characters in the field. Valid values: **all** All characters in the field are hidden. This option is equivalent to the Mask All Characters. **creditCard** The first 12 characters are hidden, the last four are displayed. This option is equivalent to the Credit Card Number option. **ssn** The first five characters are hidden, the last four are displayed. This option is equivalent to the Social Security Number option. **lastFour** All characters are hidden, except the last four are displayed. This option is equivalent to the Last Four Characters Clear option. **sin** All characters are hidden, except the last four are displayed. This option is equivalent to the Social Insurance Number option. **nino** All characters are hidden. If the field contains nine characters, Salesforce automatically inserts spaces after each pair of characters. This option is equivalent to the National Insurance Number option.
picklist	Picklist	If specified, the field is a picklist, and this field enumerates the picklist values and labels.
populateExistingRows	boolean	Indicates whether existing rows are populated (true) or not (false).
precision	int	The precision for number values. Precision is the number of digits in a number. For example, the precision value for the number 256.99 is 5.

Field Name	Field Type	Description
`referenceTargetField`	string	Available only for indirect lookup relationship fields on external objects. Specifies the custom field on the parent object to match against this indirect lookup relationship field, whose values come from an external data source. The specified custom field on the parent object must have both `externalId` and `unique` set to `true`. This field is available in API version 32.0 and later.
`referenceTo`	string	If specified, indicates a reference this field has to another object.
`relationshipLabel`	string	Label for the relationship.
`relationshipName`	string	If specified, indicates the value for one-to-many relationships. For example, in the object MyObject that had a relationship to YourObject, the relationship name YourObjects makes the relationship type obvious.
`relationshipOrder`	int	This field is valid for all master-detail relationships, but the value is only non-zero for junction objects. A junction object has two master-detail relationships, and is analogous to an association table in a many-to-many relationship. Junction objects must define one parent object as primary (0), the other as secondary (1). The definition of primary or secondary affects delete behavior and inheritance of look and feel, and record ownership for junction objects. For more information, see the Salesforce Help. 0 or 1 are the only valid values, and 0 is always the value for objects that are not junction objects.
`required`	boolean	Indicates whether the field requires a value on creation (`true`) or not (`false`).
`scale`	int	The scale for the field. Scale is the number of digits to the right of the decimal point in a number. For example, the number 256.99 has a scale of 2.
`startingNumber`	int	If specified, indicates the starting number for the field.
`stripMarkup`	boolean	Set to `true` to remove markup, or `false` to preserve markup. Used when converting a rich text area to a long text area.
`summarizedField`	string	Represents the field on the detail row that is being summarized. This field cannot be null unless the value of `SummaryOperation` is `count`.
`summaryForeignKey`	string	Represents the master-detail field on the child that defines the relationship between the parent and the child.
`summaryOperation`	SummaryOperations is an enumeration of strings	Represents the sum operation to be performed. Valid values are: • `Count` • `Min` • `Max` • `Sum`
`trackFeedHistory`	boolean	Indicates whether the field is enabled for feed tracking (`true`) or not (`false`). This field is available in API version 18.0 and later.

Field Name	Field Type	Description
`trackHistory`	boolean	Indicates whether history tracking is enabled for the field (`true`) or not (`false`). Also available for standard object fields (picklist and lookup fields only) in API version 30.0 and later.
`trackTrending`	boolean	Indicates whether historical trending data is captured for the field (`true`) or not (`false`). If this attribute is `true` for at least one field, the object is enabled for historical trending. Available in API version 29.0 and later.
`trueValueIndexed`	boolean	This field is only relevant for a checkbox field. If set, true values are built into the index. This field has been deprecated as of API version 14.0 and is only provided for backward compatibility.
`type`	FieldType is an enumeration of strings	Indicates the field type for the field. Valid values are: • `AutoNumber` • `Lookup` • `MasterDetail` • `Checkbox` • `Currency` • `Date` • `DateTime` • `Email` • `EncryptedText` • `Number`[1] • `Percent` • `Phone` • `Picklist` • `MultiselectPicklist` • `Summary` • `Text` • `TextArea` • `LongTextArea` • `Summary` • `Url` • `Hierarchy` • `File` • `CustomDataType` • `Html` • `Geolocation` For standard fields on standard objects, the `type` field is optional. This field is included for some standard field types, such as Picklist or Lookup, but not for others. The `type` field is included for custom fields.

Field Name	Field Type	Description
`unique`	boolean	Indicates whether the field is unique (`true`) or not (`false`).
`visibleLines`	int	Indicates the number of lines displayed for the field.
`writeRequiresMasterRead`	boolean	Sets the minimum sharing access level required on the master record to create, edit, or delete child records. This field applies only to master-detail or junction object custom field types. • `true`—Allows users with “Read” access to the master record permission to create, edit, or delete child records. This setting makes sharing less restrictive. • `false`—Allows users with “Read/Write” access to the master record permission to create, edit, or delete child records. This setting is more restrictive than `true`, and is the default value. For junction objects, the most restrictive access from the two parents is enforced. For example, you set to `true` on both master-detail fields, but users have “Read” access to one master record and “Read/Write” access to the other master record. In this example, users are unable to create, edit, or delete child records.

Picklist Metadata

Field Name	Field Type	Description
`controllingField`	string	The `fullName` of the controlling field if this field is a dependent picklist. A dependent picklist works with a controlling picklist or checkbox to filter the available options. The value chosen in the controlling field affects the values available in the dependent field. This field is available in API version 14.0 and later.
`picklistValues`	PicklistValue[]	Required. Represents a set of values for a picklist.
`sorted`	boolean	Required. Indicates whether values are sorted (`true`), or not (`false`).

PicklistValue Metadata

This metadata type defines a value in the picklist and specifies whether this value is the default value. This type extends `Metadata` and inherits its `fullName` field.

Note the following when working with picklist values:

- When you retrieve a standard object, you all picklist values are retrieved, not just the customized picklist values.
- When you deploy changes to standard picklist fields, picklist values are added as needed.
- You can’t set a picklist value as inactive, but if the picklist value is missing and you invoke an `update()` call, the missing value becomes inactive.

Field Name	Field Type	Description
`allowEmail`	boolean	Indicates whether this value lets users email a quote PDF (`true`), or not (`false`). This field is only relevant for the `Status` field in quotes. This field is available in API version 18.0 and later.
`closed`	boolean	Indicates whether this value is associated with a closed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in cases and tasks. This field is available in API version 16.0 and later.
`color`	string	Indicates the color assigned to the picklist value when used in charts on reports and dashboards. The color is in hexadecimal format; for example, #FF6600. If a color is not specified, it's assigned dynamically during chart generation. This field is available in API version 17.0 and later.
`controllingFieldValues`	string[]	A list of values in the controlling field that are linked to this picklist value. The controlling field can be a checkbox or a picklist. This field is available in API version 14.0 and later. The values in the list depend on the field type: • `Checkbox`: `checked` or `unchecked`. • `Picklist`: The `fullName` of the picklist value in the controlling field.
`converted`	boolean	Indicates whether this value is associated with a converted status (`true`), or not (`false`). This field is relevant for only the standard `Lead Status` field in leads. Your organization can set its own guidelines for determining when a lead is qualified. The best practice is to convert a lead when it becomes a real opportunity that you want to forecast. For more information, see "Convert Qualified Leads" in the Salesforce online help. This field is available in API version 16.0 and later.
`cssExposed`	boolean	Indicates whether this value is available in your Self-Service Portal (`true`), or not (`false`). This field is only relevant for the standard `Case Reason` field in cases. Self-Service provides an online support channel for your customers - allowing them to resolve their inquiries without contacting a customer service representative. For more information about Self-Service, see "Setting Up Self-Service" in the Salesforce online help. Note: Starting with Spring '12, the Self-Service portal isn't available for new organizations. Existing organizations continue to have access to the Self-Service portal. This field is available in API version 16.0 and later.
`default`	boolean	Required. Indicates whether this value is the default picklist value in the specified picklist (`true`), or not (`false`).
`description`	string	Description of a custom picklist value. This field is only relevant for the standard `Stage` field in opportunities. It is useful to include a description for a customized picklist value so that the historical reason

Field Name	Field Type	Description
		for creating it can be tracked. This field is available in API version 16.0 and later.
`forecastCategory`	ForecastCategories, an enumeration of type string	Indicates whether this value is associated with a forecast category (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. Valid values: • `Omitted` • `Pipeline` • `BestCase` • `Forecast` • `Closed` This field is available in API version 16.0 and later.
`fullName`	string	The name used as a unique identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. This field is inherited from Metadata.
`highPriority`	boolean	Indicates whether this value is a high priority item (`true`), or not (`false`). This field is only relevant for the standard `Priority` field in tasks. For more information about tasks, see “Considerations for Using Tasks” in the Salesforce online help. This field is available in API version 16.0 and later.
`probability`	int	Indicates whether this value is a probability percentage (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.
`reverseRole`	string	A picklist value corresponding to a reverse role name for a partner. For example, for the role “subcontractor”, a reverse role is “general contractor”. Assigning a partner role to an account in Salesforce creates a reverse partner relationship so that both accounts list the other as a partner. This field is only relevant for partner roles. For more information, see “Partner Fields” in the Salesforce online help. This field is available in API version 18.0 and later.
`reviewed`	boolean	Indicates whether this value is associated with a reviewed status (`true`), or not (`false`). This field is only relevant for the standard `Status` field in solutions. For more information about opportunities, see “Creating Solutions” in the Salesforce online help. This field is available in API version 16.0 and later.
`won`	boolean	Indicates whether this value is associated with a closed or won status (`true`), or not (`false`). This field is only relevant for the standard `Stage` field in opportunities. This field is available in API version 16.0 and later.

RelationshipReferenceTo Type

Field	Type	Details
`referenceTo`	string[]	The objects that can have a relationship to the field represented by this FieldDefinition.

WorkflowFieldUpdate Metadata

For more information about WorkflowFieldUpdate, see the *Metadata API Developer's Guide*.

FieldSet

Represents the metadata for a group of fields. Available from API version 33.0 or later.

Supported SOAP Calls

`create()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

`GET`, `HEAD`

Fields

Field	Details
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The field set description. This can be useful to describe the reason for creating the set or its intended use.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The API name of the field set.

Field	Details
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - `beta` - `deleted` - `deprecated` - `installed` - `released` - `unmanaged`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** The set's label.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package of which the field set is a part.

FlexiPage

Represents a Lightning Page. A Lightning Page is the home page for an app that appears as a menu item in the Salesforce1 navigation menu. Includes access to the associated FlexiPage object in the Salesforce Metadata API. Available from API version 31.0 or later.

Note: These app pages are known as FlexiPages in the API, but are referred to as Lightning Pages in the rest of the Salesforce documentation and UI.

Supported SOAP Calls

`create()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

GET, HEAD

Fields

Field	Details
Description	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The email page description. This field can be useful to describe the reason for creating the page or its intended use.
DeveloperName	**Type** string **Properties** Filter, Group, Sort **Description** The API name of the Lightning Page.
FullName	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated FlexiPage object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
MasterLabel	**Type** string **Properties** Filter, Group, Sort **Description** The page’s label.
Metadata	**Type** FlexiPageMetadata **Properties** Create, Nillable, Update

Field	Details
	Description Lightning Page metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package of which the flexipage is a part.
`ParentFlexiPage`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** This field is reserved for future use. Available in API version 35.0 or later.
`SobjectType`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** This field is reserved for future use. Once the value of this field is set, it can't be changed. Available in API version 33.0 or later.
`Type`	**Type** picklist **Properties** Filter, Group, Restricted picklistSort **Description** Required. The type of the Lightning Page. Available in API version 32.0 or later. In API version 32.0 and later, this field can only have a value of AppPage.

Sample Code

This code sample creates a Lightning Page with a single Recent Items component, that shows recently used Accounts and `MyCustomObject__cs`

```
 ComponentInstance recentItems = new ComponentInstance();
 recentItems.setComponentName("flexipage:recentItems");
 ComponentInstanceProperty cip = new ComponentInstanceProperty();
 cip.setName("entityNames");
 cip.setValue("Account,MyCustomObject__c");
 recentItems.setComponentInstanceProperties(new ComponentInstanceProperty[]{cip});

FlexiPageRegion mainRegion = createRegion("main");
mainRegion.setComponentInstances(new ComponentInstance[] { recentItems });

FlexiPageMetadata fpMetadata = new FlexiPageMetadata();
fpMetadata.setFlexiPageRegions(new FlexiPageRegion[]{mainRegion});
fpMetadata.setMasterLabel("My FlexiPage");
fpMetadata.setDescription("A FlexiPage with a recent items component");

FlexiPage flexiPage = new FlexiPage();
flexiPage.setFullName("MyFlexiPageDevName");
flexiPage.setMetadata(fp);

// Create
SaveResult saveResult = soapConnection.create(new SObject[] { flexiPage });
```

Flow

Use the Flow object to retrieve and update specific flow versions.

With Flow, you can create an application that navigates users through a series of screens 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. For information about the corresponding UI-based flow building tool, see Cloud Flow Designer in the Salesforce Help.

When using the Tooling API to work with flows, consider that:

- You can describe information for a flow installed from a managed package but not its metadata.
- You can't overwrite an active flow or one that was once active.
- Every time you update a flow, you're actually deleting the existing flow and creating a new flow from it, with a new ID.

Note: Legacy flows created with the Desktop Flow Designer can't be modified with the API. Update your flow by recreating it with the Cloud Flow Designer.

You can delete a flow version, as long as it:

- Isn't active
- Doesn't have any paused or waiting interviews

If the flow version has paused or waiting interviews, wait for those interviews to finish, or delete them.

This object is available in API version 34.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`DELETE`, `GET`, `HEAD`, `PATCH`, `POST`

Fields

Field	Details
`Definition`	**Type** FlowDefinition **Properties** Filter, Group, Nillable, Sort **Description** This flow's definition object.
`DefinitionId`	**Type** ID **Properties** Filter, Group, Sort **Description** The ID of this flow's FlowDefinition.
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** A description of the flow, such as what it's meant to do or how it works.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the flow in the Metadata API. The `fullName` consists of two parts, separated by a hyphen: • Unique name for the flow that contains only underscores and alphanumeric characters. It must be unique across the organization, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field	Details
	• Version number for the flow. For example, “sampleFlow-3” specifies version 3 of the flow whose unique name is sampleFlow. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`MasterLabel`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** Label for the flow.
`Metadata`	**Type** `mns:` Flow **Properties** Create, Nillable, Update **Description** The flow’s metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance. If the flow is part of a managed package this field will be Null. Metadata isn’t returned for flows in managed packages.
`ProcessType`	**Type** Restricted picklist **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The type of the flow: • AutoLaunchedFlow • Flow • Workflow • LoginFlow • ActionPlan • JourneyBuilderIntegration • UserProvisioningFlow Note: This value has significant impact on validation when saving the flow and on the flow's runtime behavior. Don't change this value unless you understand the flow properties of the specified type.
Status	**Type** Restricted picklist **Properties** Filter, Group **Description** The flow's status: • Active • Draft • Obsolete • InvalidDraft
VersionNumber	**Type** int **Properties** Filter, Group, Sort **Description** The flow's version number.

FlowDefinition

The parent of a set of flow versions.

When using the Tooling API to work with flow definitions, consider that:

- You can activate and deactivate flows with the `Metadata` field.
- You can update `masterlabel` and `description` of a FlowDefinition.
- FlowDefinition are implicitly created when the Flow object is created. This means FlowDefinition objects can only be updated.

Note: Legacy flows created with the Desktop Flow Designer can’t be modified with the API. Update your flow by recreating it with the Cloud Flow Designer.

This object is available in API version 34.0 and later.

Supported SOAP API Calls

`query()`, `retrieve()`, `update()`

Supported REST API HTTP Methods

`GET`, `HEAD`, `PATCH`, `POST`

Fields

Field Name	Details
`ActiveVersion`	**Type** Flow **Properties** Filter, Group, Nillable, Sort **Description** The active flow version object.
`ActiveVersionId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** The ID of the active flow version.
`Description`	**Type** string **Properties** Nillable **Description** Flow definition information, specified by the organization’s administrator.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** Developer name of this flow definition.

Field Name	Details
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the flow definition in the Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`LatestVersion`	**Type** Flow **Properties** Filter, Group, Nillable, Sort **Description** The latest flow version object, regardless of the flow's state.
`LatestVersionId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** ID of the latest flow version, regardless of the flow's state.
`MasterLabel`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Label for the flow definition.
`Metadata`	**Type** `mns:`FlowDefinition **Properties** Create, Nillable, Update **Description** The flow definition's metadata object, containing information about which flow version is active and the flow definition's description. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field Name	Details
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace associated with this flow definition.

HeapDump

A complex type that represents a heap dump in an ApexExecutionOverlayResult object. Available from API version 28.0 or later.

Fields

Field	Details
`className`	**Type** string **Description** The name of the Apex class or trigger.
`extents`	**Type** array of TypeExtent **Description** TypeExtent includes the following fields: • `collectionType` • `count` • `definition` (array of AttributeDefinition) • `extent` (array of HeapAddress) • `totalSize` • `typeName`
`heapDumpDate`	**Type** dateTime **Description** The date and time that the heap dump was captured.
`namespace`	**Type** string **Description** The namespace of the Apex class or trigger. Null if there is no namespace.

Usage

Use heap dumps to capture structured debugging information.

HistoryRetentionJob

Represents the body of retained data from the archive, and the status of the archived data. Available in API version 29.0 or later.

Supported SOAP API Calls

`describeSObjects()`, `query()`

Supported REST API HTTP Methods

`GET`

Fields

Field Name	Details
`DurationSeconds`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** How many seconds the field history retention job took to complete (whether successful or not).
`HistoryType`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort **Description** The object type that contains the field history that you retained. Valid values for standard objects are: • `Account` • `Case` • `Contact` • `Leads` • `Opportunity` For custom objects, use the object name.

Field Name	Details
`NumberOfRowsRetained`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** The number of field history rows that a field history retention job has retained.
`RetainOlderThanDate`	**Type** dateTime **Properties** Filter, Sort **Description** The date and time before which all field history data was retained.
`StartDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** The start date of the field history retention job.
`Status`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** Provides the status of the field history retention job. By default, the pilot feature copies data to the archive, leaving a duplicate of the archived data in Salesforce. Deletion of data from Salesforce after archiving is available upon request. Status can include: • `CopyScheduled` • `CopyRunning` • `CopySucceeded` • `CopyFailed` • `CopyKilled` • `NothingToArchive` • `DeleteScheduled` • `DeleteRunning` • `DeleteSucceeded` • `DeleteFailed`

Field Name	Details
	- `DeleteKilled`

HomePageComponent

Represents a home page component.

This object is available in API version 35.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`Body`	**Type** string **Properties** Nillable **Description** If this component is an HTML page component, this field is the body of the HTML.
`Height`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Required for Visualforce Area components. Indicates the height (in pixels) of the component.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Filter, Group, Nillable, Sort **Description** Indicates the manageable state of the specified component that is contained in a package: - `beta`

Field	Details
	• `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Name`	**Type** string **Properties** Filter, Group, idLookup, Namefield, Sort **Description** The name of the home page component.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** A unique string to distinguish this type from any others.
`ShowLabel`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** ID of the home page layout.
`ShowScrollbars`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** ID of the home page layout.

HomePageLayout

Represents a home page layout.

This object is available in API version 35.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`Id`	**Type** string **Properties** Defaulted on create, Filter, Group, idLookup, Sort **Description** ID of the home page layout.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Filter, Group, Nillable, Sort **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Name`	**Type** string **Properties** Filter, Group, idLookup, Namefield, Sort **Description** The home page layout name.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description A unique string to distinguish this type from any others.

KeywordList

Represents a list of keywords used in community moderation. Available in Tooling API version 36.0 and later.

This keyword list is a type of moderation criteria that defines offensive language or inappropriate content that you don’t want in your community.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`Description`	**Type** textarea **Properties** Filter, Nillable, Sort. **Description** A description of the keyword list.
`DeveloperName`	**Type** string **Properties** Filter, Group, Namefield, Sort **Description** The developer’s internal name for the keyword list used in the API.
`FullName`	**Type** string **Properties** Create, Group, Nillable.

Field	Details
	Description The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Language`	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort. **Description** The language of the keyword list. Valid values are: • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW` • Danish: `da` • Dutch: `nl_NL` • English: `en_US` • Finnish: `fi` • French: `fr` • German: `de` • Italian: `it` • Japanese: `ja` • Korean: `ko` • Norwegian: `no` • Portuguese (Brazil): `pt_BR` • Russian: `ru` • Spanish: `es` • Spanish (Mexico): `es_MX` • Swedish: `sv` • Thai: `th`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort. **Description** Label for the keyword list.
`Metadata`	**Type** mns:KeywordList

Field	Details
	Properties Create, Nillable, Update. **Description** KeywordList metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Layout

Represents a page layout.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The Id of the EntityDefinition object associated with this object.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The unique name of the layout used as the identifier for API access. The `fullName` can contain only underscores and alphanumeric characters. It must be unique, begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.

Field	Details
	Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`LayoutType`	**Type** LayoutType enumerated list **Properties** Filter, Group, Restricted picklist, Sort **Description** Indicates the type of the layout. Valid values are: • `GlobalQuickActionList` • `ProcessDefinition` • `Standard`
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** `mns:`Layout **Properties** Create, Nillable, Update **Description** Layout metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Nillable, Sort

Field	Details
	Description The layout name.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** A unique string to distinguish this layout from any others. For example, if this layout is being using by a flow, use the `NamespacePrefix` to uniquely identify the layouts in multiple flow instances.
`ShowSubmitAndAttachButton`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Only allowed on Case layout. If true, **Submit & Add Attachment** displays on case edit pages to portal users in the Customer Portal.
`TableEnumOrId`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** The enum (for example, Account) or ID of the object this layout is on.

LookupFilter

Represents a lookup filter, which restricts the valid values and lookup dialog results for lookup, master-detail, and hierarchical relationship fields.

Available from Tooling API version 34.0 or later.

Note: LookupFilter is not supported on the article type object.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Fields

Field Name	Details
`Active`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the lookup filter is active.
`DeveloperName`	**Type** string **Properties** Filter, Group, Namefield, Sort **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **Record Type Name**.
`FullName`	**Type** string **Properties** Filter, Group, Sort **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`IsOptional`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Required. If `true`, the lookup filter is optional.

Field Name	Details
ManageableState	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - beta - deleted - deprecated - installed - released - unmanaged
Metadata	**Type** LookupFilter **Properties** Create, Nillable, Update **Description** The metadata for this lookup filter. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the custom field, which is sometimes different from the object's namespace.
SourceFieldDefinition	**Type** string **Properties** Filter, Group, Sort **Description** The field that this filter applies to.
SourceFieldDefinitionId	**Type** string

Field Name	Details
	Properties Filter, Group, Sort **Description** Durable ID of the object specified in `SourceFieldDefinition`.
`SourceObject`	**Type** string **Properties** Filter, Group, Sort **Description** The object that contains the lookup field that uses this lookup filter. Null if the lookup filter doesn't reference fields on the source object.
`TargetEntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** The entity definition for the source lookup field.
`TargetEntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** ID of the TargetEntityDefinition.

Example: **The `Owner` field on Account represents a user with certain characteristics.**

In this example, SourceFieldDefinition is `Account.Owner` and TargetEntityDeifnition is `User`, because Owner is a lookup field to User.

LookupFilter Metadata

Metadata about the lookup filter is returned in the `Metadata` field:

Field	Type	Description
`active`	boolean	Required. If `true`, the lookup filter is active.
`booleanFilter`	string	The filter logic, if any, applied to this filter using Boolean operators AND, OR, or NOT.

Field	Type	Description
description	string	A description of the filter does.
errorMessage	string	If the lookup filter fails, the error m.
filterItems	FilterItem	Required. The set of filter conditions. Each lookup filter can have up to 10 FilterItems.
infoMessage	string	Information displayed on the page to help the user. For example, explaining why some items are excluded in the lookup filter.
isOptional	boolean	Required. If true, the lookup filter is optional.

FilterItem Metadata

Represents one entry in a set of filter criteria.

Field	Type	Description
field	string	Represents the field specified in the filter.
operation	FilterOperation, an enumeration of strings	Valid values: • equals • notEqual • lessThan • greaterThan • lessOrEqual • greaterOrEqual • contains • notContain • startsWith • includes • excludes • within (DISTANCE criteria only)
value	string	Represents the value of the filter item being operated upon. For example, if the filter is my_number_field__c > 1, the value of this field is 1.
valueField	string	Specifies if the final column in the filter contains a field or a field value. Approval processes don’t support this field in filter criteria.

MenuItem

Represents a menu item.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`, `update()`

Supported REST HTTP Methods

`GET`, `POST`

Fields

Field	Details
`Active`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether the item in the menu is active (`true`) or not (`false`).
`AppId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the app that this menu item is associated with. Can be an enum (such as Feed or People) or an alphanumeric ID. Use `AppId` as the unique ID for the menu item, not `Id`.
`Color`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The color of the menu item that appears in the user interface. This field is described in Web color RGB format, such as `00FF00`.
`IconURL`	**Type** url **Properties** Filter, Group, Nillable, Sort **Description** The URL of an icon in the menu item.

Field	Details
`Label`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The menu item label that appears in the user interface.
`MenuType`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The type of menu that this menu item belongs to. Valid values are: • `AppSwitcher`: the Force.com app menu, a drop-down menu that's displayed at the top of every app page • `Salesforce1`: the Salesforce1 navigation menu • `NetworkTabs`: the Salesforce Communities tab set This field is required for `query()`.
`SortOrder`	**Type** int **Properties** Filter, Group, Nillable, Sort, Update **Description** The `SortOrder` value determines the order in which a menu item is displayed in the user interface. This field must be an ordinal number greater than 0, and must be unique in the list. Inactive menu items have a value of -1.
`Theme`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The associated theme, which must be one of the following values. • `theme2`: the Salesforce theme that was used prior to Spring '10 • `theme3`: the Salesforce theme that was introduced in Spring '10 • `theme4`: the theme that was introduced in Winter '14 for the mobile touchscreen version of Salesforce • `custom`: the theme that's associated with a custom icon This field is required for `query()` for `Color` and `IconURL`.

Usage

MenuItem can be queried and manipulated to change how menu items appear in Salesforce. The following example modifies the Salesforce1 left navigation menu.

```
String query = "SELECT AppId, Label, Active, SortOrder FROM MenuItem "
+
     "WHERE MenuType = 'Salesforce1'";
SObject[] records = sforce.query(query).getRecords();

//Activate all menu items
for (int i = 0; i < records.length; i++) {
   MenuItem item = (MenuItem)records[i];
   item.setOrder(i + 1);
   item.setActive(true);
}

sforce.update(records);
```

MetadataContainer

Manages working copies of ApexClassMember, ApexTriggerMember, ApexPageMember, and ApexComponentMember objects, including collections of objects to be deployed together.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Name`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The name of the MetadataContainer. If a container with the same name already exists, an error is reported on `create()` or `POST`. This field is required.

Usage

Use a MetadataContainer as a package for your tool's workspace. As a user works in the tool, update the ApexClassMember, ApexTriggerMember, ApexPageMember and ApexComponentMember objects in the MetadataContainer and use a ContainerAsyncRequest object to save and deploy changes to the current organization.

A MetadataContainer can be reused, but container members can't.

- When a ContainerAsyncRequest completes successfully, the `MetadataContainerId` field on all container members is changed from the ID of the MetadataContainer to the ID of the ContainerAsyncRequest. At this point, container members can no longer be modified or deployed, and can't be queried via the MetadataContainer; you have to query the ContainerAsyncRequest to see what was deployed.
- If the deployment fails, container members remain on the MetadataContainer and can still be modified until they are successfully deployed on another ContainerAsyncRequest. The `MetadataContainerId` field on the completed (failed deployment) ContainerAsyncRequest is set to the ID of the MetadataContainer, so you can have multiple completed ContainerAsyncRequests on a single MetadataContainer.

Note: Deleting a MetadataContainer deletes all objects that reference it.

ModerationRule

Represents a rule used in your community to moderate user-generated content. Available in Tooling API version 36.0 and later.

Each rule specifies the user-generated content the rule applies to, the criteria to enforce the rule on, and the moderation action to take. You can create rules that block, flag, or replace user-generated content that contains offensive language or inappropriate content.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
Action	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist. **Description** Required. Indicates the moderation action to take when your criteria is matched. The valid values are: - `Block` - `Replace`

Field	Details
	• `Flag`
Active	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort. **Description** Required. Indicates whether the moderation rule is active (`true`) or inactive (`false`).
Description	**Type** textarea **Properties** Filter, Nillable, Sort. **Description** A description of the moderation rule.
DeveloperName	**Type** string **Properties** Filter, Group, Namefield, Sort **Description** The developer's internal name for the moderation rule used in the API.
FullName	**Type** string **Properties** Create, Group, Nillable. **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
Language	**Type** picklist **Properties** Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort. **Description** The language of the moderation rule. Valid values are: • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW`

Field	Details
	• Danish: `da` • Dutch: `nl_NL` • English: `en_US` • Finnish: `fi` • French: `fr` • German: `de` • Italian: `it` • Japanese: `ja` • Korean: `ko` • Norwegian: `no` • Portuguese (Brazil): `pt_BR` • Russian: `ru` • Spanish: `es` • Spanish (Mexico): `es_MX` • Swedish: `sv` • Thai: `th`
MasterLabel	**Type** string **Properties** Filter, Group, Sort. **Description** Label for the moderation rule.
Metadata	**Type** mns:ModerationRule **Properties** Create, Nillable, Update. **Description** Moderation rule metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
UserMessage	**Type** textarea **Properties** Filter, Nillable, Sort. **Description** The message you want your community members to see when their content is blocked. Use the *`%BLOCKED_KEYWORD%`* variable to display up to five blocked words in the user message. If you don’t specify a message, the user sees the standard message: “You can't

Field	Details
	use *%BLOCKED_KEYWORD%* or other inappropriate words in this community. Review your content and try again.”

OwnerChangeOptionInfo

Represents default and optional actions that can be performed when a record’s owner is changed. Available in Tooling API version 35.0 and later.

Supported SOAP Calls

`describeSObject()`, `query()`, `retrieve()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`DefaultValue`	**Type** boolean **Properties** Filter, Group, Sort **Description** Default value of the checkbox for this option in the user interface.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** The object to which this change applies.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the record.

Field	Details
`IsEditable`	**Type** boolean **Properties** Filter, Group, Sort **Description** Indicates whether this option is editable by the user when updating the owner using the `OwnerChangeOptions` SOAP header.
`Label`	**Type** string **Properties** Filter, Group, Sort **Description** The label that corresponds to the option in the user interface.
`Name`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name for the option.

Example

Retrieve all the change options for contacts.

```
SELECT Id, Name, Label, IsEditable, DefaultValue, EntityDefinition.QualifiedApiName
FROM OwnerChangeOptionInfo
WHERE EntityDefinition.QualifiedName='Contact’
```

PathAssistant

Represents a Sales Path. Available in Tooling API version 36.0 and later.

Supported SOAP Calls

`retrieve()`, `query()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
DeveloperName	**Type** string **Properties** Filter, Group, Sort **Description** The unique name of the sales path in the API.
FullName	**Type** string **Properties** Group, Nillable **Description** The name of the sales path in the Metadata API. Query this field only if the query result contains no more than 1 record. Otherwise, an error is returned. If more than 1 record exists, use multiple queries to retrieve the records. This limit protects performance.
IsActive	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether the sales path is active (true) or inactive (false).
IsDeleted	**Type** boolean **Properties** Sort **Description** Indicates whether the record has been moved to the Recycle Bin (true) or not (false).
IsMasterRecordType	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether this sales path is for the master record type (true) or not (false).
Language	**Type** string

Field	Details
	Properties Filter, Group, Restricted picklist, Sort **Description** The language of the sales path. Valid values are: • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW` • Danish: `da` • Dutch: `nl_NL` • English: `en_US` • Finnish: `fi` • French: `fr` • German: `de` • Italian: `it` • Japanese: `ja` • Korean: `ko` • Norwegian: `no` • Portuguese (Brazil): `pt_BR` • Russian: `ru` • Spanish: `es` • Spanish (Mexico): `es_MX` • Swedish: `sv` • Thai: `th`
`MasterLabel`	**Type** string **Properties** Filter, Group, Sort **Description** Label for this sales path.
`Metadata`	**Type** msn:PathAssistant **Properties** Create, Nillable, Update **Description** Sales path metadata from the msn namespace. Query this field only if the query result contains no more than 1 record. Otherwise, an error is returned. If more than 1 record exists, use multiple queries to retrieve the records. This limit protects performance.

Field	Details
RecordTypeId	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** The unique identifier for a record type.
SobjectProcessField	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** Name of the picklist field which determines the steps you can use in the sales path. For example, OpportunityStage in the case of opportunities or LeadStatus in the case of leads.
SobjectType	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** The object this path relates to. Valid values are: `Opportunity` or `Lead`.

PathAssistantStepInfo

Represents guidance for a step on a Sales Path. Available in Tooling API version 36.0 and later.

Supported SOAP Calls

`update()`, `query()`

Supported REST HTTP Methods

`GET`, `PATCH`

Fields

Field	Details
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name of the sales path guidance information.
`Info`	**Type** string **Properties** Filter, Nillable, Sort **Description** The text of the guidance displayed to the user in the user interface.
`IsDeleted`	**Type** boolean **Properties** Sort **Description** Indicates whether the record has been moved to the Recycle Bin (`true`) or not (`false`).
`Language`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** The language of the sales path. Valid values are: • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW` • Danish: `da` • Dutch: `nl_NL` • English: `en_US` • Finnish: `fi` • French: `fr` • German: `de` • Italian: `it` • Japanese: `ja` • Korean: `ko`

Field	Details
	- Norwegian: no - Portuguese (Brazil): pt_BR - Russian: ru - Spanish: es - Spanish (Mexico): es_MX - Swedish: sv - Thai: th
MasterLabel	**Type** string **Properties** Filter, Group, Sort **Description** Label for this sales path guidance information record.

PathAssistantStepItem

Represents layout or guidance details for a step on a Sales Path. Available in Tooling API version 36.0 and later.

Supported SOAP Calls

query()

Supported REST HTTP Methods

GET

Fields

Field	Details
IsDeleted	**Type** boolean **Properties** Sort **Description** Indicates whether the record has been moved to the Recycle Bin (true) or not (false).
ItemId	**Type** ID

Field	Details
	Properties Filter, Group, Sort **Description** A foreign key field pointing to the `Type` field that represents either the layout (if `Type` is set to `Layout`) or the PathAssistantStepInfo (if `Type` is set to `Information`) of this guidance detail.
`PathAssistantId`	**Type** ID **Properties** Filter, Group, Sort **Description** ID of the PathAssistant related to this step.
`RecordTypeId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** ID of the record type associated with this sales path.
`Type`	**Type** string **Properties** Filter, Group, Sort **Description** The type of data that `ItemId` refers to. Valid values are: • `Information` • `Layout`

PostTemplate

Represents an approval post tempalte for Approvals in Chatter.

This object is available in API version 35.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`, `search()`, `update()`

Supported REST HTTP Methods

`GET`, `PATCH`

Fields

Field	Details
`Description`	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** A description of the feed post template, limited to 255 characters.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** A relationship lookup to the object type associated with this PostTemplate. You can’t interact directly with this field. Instead, use it in queries.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The durable ID for the object defined in the `EntityDefinition` field.
`Name`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The template name.

Profile

Represents a user profile. A profile defines a user's permission to perform different functions within Salesforce.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`GET`, `PATCH`

Fields

Field	Details
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The profile description, limited to 255 characters.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The unique profile name. Use this name when creating the profile, before you have an ID. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Metadata`	**Type** ProfileMetadata **Properties** Create, Nillable, Update **Description** The profile metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Name`	**Type** string

Field	Details
	Properties Filter, Group, idLookup, Sort **Description** The profile name.

ProfileLayout

Represents a profile layout.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`LayoutId`	**Type** ID **Properties** Filter, Group, Sort **Description** The unique identifier for this layout.
`ProfileId`	**Type** ID **Properties** Filter, Group, Sort **Description** The unique identifier for this profile.
`RecordTypeId`	**Type** ID **Properties** Filter, Group, Sort

Field	Details
	Description The unique identifier for the record.
`TableEnumOrId`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** The enum (for example, Account) or ID of the object this field is on.

Publisher

Represents the publisher of objects and fields. For example, Salesforce is the publisher for standard objects, the organization is the publisher for custom objects, and the package is the publisher for installed packages. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()>`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn’t guaranteed to stay the same from one release to the next. To simplify queries, use this field.

Field	Details
`InstalledEntityDefinitions`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Metadata for the objects installed by this publisher. Because this field represents a relationship, use only in subqueries.
`InstalledFieldDefinitions`	**Type** QueryResult **Properties** Filter, Group, Nillable, Sort **Description** Metadata for the fields installed by this publisher. Because this field represents a relationship, use only in subqueries.
`IsSalesforce`	**Type** boolean **Properties** Filter, Group, Nillable, Sort **Description** Indicates whether Salesforce provided the associated objects or fields (`true`).
`Name`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The presentation-friendly name of the publisher.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in

Field	Details
	an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. - In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects. `NamespacePrefix` is null if the publisher is Salesforce.

QueryResult

Represents the results of a query. For example, if you query on the object EntityDefinition, all the layouts for that entity are returned as an array of QueryResult objects in the `Layouts` field. Available in Tooling API version 34.0 and later.

QueryResult is not an extension of sObject.

Fields

Field	Details
`done`	**Type** boolean **Description** If `true`, no additional rows can be retrieved from the query result. If `false`, one or more rows remain to be retrieved. Use this value as a loop condition while iterating through query results.
`entityTypeName`	**Type** string **Description** The object or entity type, such as `ApexClass` or `CompactLayoutInfo`.
`nextRecordsUrl`	**Type** string **Description** If the results exceed the current batch size, this field contains the URL of the next record in the query result set. This field is populated for the REST resource `queryAll`, and is analogous to `queryLocator` for SOAP calls.
`queryLocator`	**Type** QueryLocator

Field	Details
	Description If the results exceed the current batch size, this field contains a unique identifier used to retrieve the next batch of records. This field is populated for SOAP `queryMore()` and is analogous to the REST resource `queryAll`. Each new batch returns a new `queryLocator` value.
`records`	**Type** sObject **Description** Array of sObjects matching the data specified in the query.
`size`	**Type** int **Description** Total number of rows returned. If no rows were returned, the value is (`0`). This field is the same as the `size` field in QueryResult in the Enterprise and Partner WSDLs.
`totalSize`	**Type** int **Description** Total number of rows returned. Indicates whether the query retrieved any rows (any value greater than `0`) or not (`0`). This field is the same as the `totalSize` field in QueryResult using the REST resource `query` or `queryAll`.

QueryLocator Metadata

Field	Details
`queryLocator`	**Type** string **Description** If the results exceed the current batch size, this field contains an identifier. Use with the SOAP `queryMore()` call to retrieve the next batch of records. Each new batch returns a new `queryLocator` value.

QuickActionDefinition

Represents the definition of a quick action.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`Description`	**Type** textarea **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The description of the action.
`DeveloperName`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The unique name of the action in the API. This field corresponds to the **Name** field in the user interface.
`Height`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The height of the action, in pixels. This field is set only when the quick action has a custom icon.
`IconId`	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the action icon. This field is set only when the quick action has a custom icon.
`Label`	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** The action label that corresponds to the **Label** field in the user interface.
`Language`	**Type** picklist **Properties** Create, Defaulted on create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The language of the action. Valid values are: • Chinese (Simplified): `zh_CN` • Chinese (Traditional): `zh_TW` • Danish: `da` • Dutch: `nl_NL` • English: `en_US` • Finnish: `fi` • French: `fr` • German: `de` • Italian: `it` • Japanese: `ja` • Korean: `ko` • Norwegian: `no` • Portuguese (Brazil): `pt_BR` • Russian: `ru` • Spanish: `es` • Spanish (Mexico): `es_MX` • Swedish: `sv` • Thai: `th`
`MasterLabel`	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The action label.
`NamespacePrefix`	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** The namespace of the action.
`OptionsCreateFeedItem`	**Type** boolean **Properties** Create, Filter, Update **Description** Indicates whether successful completion of the action creates a feed item (`true`) or not (`false`). Applies only to Create Record, Update Record, and Log a Call quick action types. Available in API version 36.0 and later.
`SobjectType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort **Description** The associated object's API name. For example, `FeedItem`.
`StandardLabel`	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The standard label for the action. Valid values are: • `ChangeDueDate` • `ChangePriority` • `ChangeStatus` • `CreateNew` • `CreateNewRecordType` • `Defer` • `EditDescription` • `LogACall` • `LogANote` • `New` • `NewChild` • `NewChildRecordType` • `NewRecordType`

Field	Details
	- Quick - QuickRecordType - SendEmail - SocialPost - Update
SuccessMessage	**Type** textarea **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The message that displays to the user upon successful completion of the action. Available in API version 36.0 and later.
TargetField	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The API name of the parent object for the record created by this quick action. For example, CollaborationGroup.
TargetRecordTypeId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the target record type.
TargetSobjectType	**Type** picklist **Properties** Create, Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** The API name of the type of object record this action will create. For example, OpportunityLineItem.
Type	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update

Field	Details
	Description The type of action. Valid values are: • `Canvas` • `Create` • `LogACall` • `Post` • `SendEmail` • `SocialPost` • `Update` • `VisualforcePage`
`Width`	**Type** int **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The width of the action, in pixels. This field is set only when the quick action has a custom icon.

Usage

A QuickActionDefinition represents information about a quick action. The following example creates a global quick action that lets users quickly create a task.

```
QuickActionDefinition qad = new QuickActionDefinition();
qad.setDeveloperName("MyQuickCreateTaskAction");
qad.setSobjectType("Global");
qad.setTargetSobjectType("Task");
qad.setMasterLabel("Quick create a task");
qad.setType(QuickActionType.Create);
qad.setDescription("Quickly creates a Task");

sforce.create(new SObject[]{qad});
```

QuickActionList

Represents a list of quick actions.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

DELETE, GET, PATCH, POST

Fields

Field	Details
LayoutId	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the associated layout.

Usage

A QuickActionList is a junction between QuickActionListItem objects and a layout. If a layout doesn't have an associated QuickActionList, it inherits the actions from the global page layout.

The following example retrieves all quick action lists in an organization and their associated layout ID.

```
String query = "SELECT Id,LayoutId FROM QuickActionList";
SObject[] records = sforce.query(query).getRecords();

for (int i = 0; i < records.length; i++) {
   QuickActionList list = (QuickActionList)records[i];
   String relatedLayoutId = list.get("LayoutId");
}
```

QuickActionListItem

Represents an item in a quick action list.

This object is available in API version 32.0 and later.

Supported SOAP Calls

create(), delete(), query(), retrieve(), update(), upsert()

Supported REST HTTP Methods

DELETE, GET, PATCH, POST

Fields

Field	Details
`QuickActionDefinition`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The enum name or ID of the QuickActionDefinition that's associated with this list item. Valid values are: • `Case.ChangeStatus` • `Case.LogACall` • `FeedItem.ContentPost` • `FeedItem.LinkPost` • `FeedItem.MobileSmartActions` • `FeedItem.PollPost` • `FeedItem.QuestionPost` • `FeedItem.TextPost`
`QuickActionListId`	**Type** reference **Properties** Create, Filter, Group, Sort **Description** The ID of the QuickActionList associated with this list item.
`SortOrder`	**Type** int **Properties** Create, Filter, Group, Sort, Update **Description** The order in which this list item appears in the picklist. This field must be an ordinal number greater than 0, and must be unique in the list.

Usage

A QuickActionListItem associates a QuickActionDefinition with a QuickActionList. You can query to find out which quick actions are in a list, insert or delete to add or remove quick actions from a list, and update to change the order of quick actions in the list.

The following example reverses the order in the list of the actions, and then removes the first action from the list.

```
String query = "SELECT Id,SortOrder FROM QuickActionListItem Where QuickActionListId='" +
 listId + "'"
```

```
SObject[] records = sforce.query(query).getRecords();

for(int i=0;i<records.length;i++) {
   QuickActionListItem item = (QuickActionListItem)records[i];
   item.setSortOrder(records.length-i);
}

sforce.update(records);

// Last record in array is first record in reordered list
sforce.delete(records[records.length-1].getId());
```

RecentlyViewed

Represents metadata entities typically found in Setup such as page layout definitions, workflow rule definitions, and email templates that the current user has recently viewed.

This object is available in the Tooling API version 33.0 and later.

Supported SOAP Calls

`query()`, `update()`

Supported REST HTTP Methods

`GET`

Special Usage Rules

The RecentlyViewed object supports the following metadata entities:

- Apex classes
- Apex triggers
- Approval processes
- Apps
- Custom report types
- Email templates
- Fields
- Objects
- Page layouts
- Permission sets
- Profiles
- Static resources
- Tabs
- Users
- Validation rules

- Visualforce pages
- Visualforce components
- Workflow email alerts
- Workflow field updates
- Workflow outbound messages
- Workflow rules
- Workflow tasks

Fields

Field	Details
`Alias`	**Type** string **Properties** Filter, Group,Nillable, Sort **Description** The alias on the item.
`Email`	**Type** email **Properties** Filter, Group, Nillable, Sort **Description** The email address on the item.
`FirstName`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The first name on the item.
`Id`	**Type** ID **Properties** Defaulted on create, Filter, Group, Sort **Description** The ID of the recently viewed item.
`IsActive`	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** Indicates whether the recently viewed item is an active user (true) or not (false). This field contains a value only if the recently viewed item is a user.
`LastName`	**Type** string **Properties** Filter, Group,Nillable, Sort **Description** The last name on the item.
`LastReferencedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort, Update **Description** The timestamp for when the current user last viewed an item related to this item.
`LastViewedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort, Update **Description** The timestamp for when the current user last viewed this item. If this value is null, this item might only have been referenced (see `LastReferencedDate`) and not viewed.
`Name`	**Type** string **Properties** Filter, Group,Nillable, Sort **Description** If the recently viewed item is a user, this is the user's name. Specifically, it's the concatenation of the `FirstName` and `LastName` field values.
`NetworkId`	**Type** reference **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The ID of the community that this group is part of. This field is available only if Salesforce Communities is enabled in your organization.
`Phone`	**Type** phone **Properties** Filter, Group,Nillable, Sort **Description** The phone number on the item.
`ProfileId`	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** If the recently viewed item is a user, this is the user's profile ID.
`RelatedObject`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** The object that the recently viewed item is related to. For example, if the recently viewed item is an Account Custom Field, then the related object will be Account. Not all recently viewed items will have a related object.
`Title`	**Type** string **Properties** Filter, Group,Nillable, Sort **Description** If the recently viewed item is a user, this is the user's title. For example, CFO or CEO.
`Type`	**Type** picklist **Properties** Filter, Group, Nillable,Restricted picklist, Sort **Description** The sObject type for this recently viewed item.

Field	Details
UserRoleId	**Type** reference **Properties** Filter, Group, Nillable, Sort **Description** The ID of the user role associated with this object.

Usage

This object provides a heterogeneous list of different metadata types and consists of recently viewed records. A record is considered viewed when the user sees the details associated with it, but not when the user sees it in a list with other records. Use this object to programmatically construct a list of recently viewed items specific to the current user, for example, on a custom user interface or for search auto-complete options. You can also retrieve a filtered list of records by object type (`Type`). The RecentlyViewed data is periodically truncated down to 200 records per object.

Use this query in your code to retrieve a list of all the records that were recently viewed. The results are ordered from most to least recent.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE LastViewedDate !=null
ORDER BY LastViewedDate DESC
```

Use this query to retrieve data that was either viewed or referenced, but only for a limited set of objects.

```
SELECT Id, Name
FROM RecentlyViewed
WHERE Type IN ('CustomEntityDefinition', 'CustomFieldDefinition')
ORDER BY LastViewedDate DESC
```

RecordType

Represents a custom record type.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

`GET`, `PATCH`, `POST`

Fields

Field	Details
`BusinessProcessId`	**Type** ID **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** ID of an associated BusinessProcess.
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The record type description, limited to 255 characters.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the record.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`IsActive`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether this record is active (`true`) or not (`false`). Only active record types can be applied to records.

Field	Details
ManageableState	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - beta - deleted - deprecated - installed - released - unmanaged
Metadata	**Type** RecordTypeMetadata **Properties** Create, Nillable, Update **Description** Record metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
Name	**Type** string **Properties** Nillable **Description** The record type name.
NamespacePrefix	**Type** string **Properties** Nillable **Description** A unique string to distinguish this type from any others.
SobjectType	**Type** string **Properties** Filter, Group, Nillable, Sort

Field	Details
	Description The type of standard object that this record type is derived from.

RelationshipDomain

Represents the relationship an object has with other objects. RelationshipDomain allows you to write simpler queries. For example, "which objects are the child objects for the object defined in `ParentSobject`" is easier using RelationshipDomain. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`ChildSobject`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** Metadata for the child object, if any.
`ChildSobjectId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the `ChildSobject`.

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`Field`	**Type** FieldDefinition **Properties** Filter, Group, Sort **Description** The relationship field on this object that defines the relationship to `ChildSobject` or `ParentSobject`.
`FieldId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of `Field`.
`IsCascadeDelete`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this object's parent can't be deleted until all records for this object are deleted. Corresponds to `Cascade` value for DeleteConstraint in the Metadata API.
`IsDeprecatedAndHidden`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this object is unavailable for the current version.
`IsRestrictedDelete`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, this object can't be deleted. Corresponds to `Restrict` value for DeleteConstraint in the Metadata API.
`JunctionIdListName`	**Type** string **Properties** Defaulted on create, Filter, Group, Sort **Description** The name of the list of junction IDs associated with an object. Each ID represents an object that has a relationship with the associated object.
`ParentSobject`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** Metadata for the parent object, if any.
`ParentSobjectId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the `ParentSobject`.
`RelationshipInfo`	**Type** RelationshipInfo **Properties** Filter, Group, Sort **Description** Properties about the relationship.
`RelationshipInfoId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of `RelationshipInfo` for this relationship domain.
`RelationshipName`	**Type** string

Field	Details
	Properties Filter, Group, Nillable, Sort **Description** Name of this relationship.

RelationshipInfo

Represents the properties of a relationship between objects. Simplify queries with RelationshipInfo, such as answering the question "which objects are parent objects for the object defined in `ChildSobject`. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`, `search()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`ChildSobject`	**Type** EntityDefinition **Properties** Filter, Group, Sort **Description** Metadata for the child object, if any.
`ChildSobjectId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the `ChildSobject`.

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`Field`	**Type** FieldDefinition **Properties** Filter, Group, Sort **Description** The relationship field that defines the relationship to `ChildSobject` or `ParentSobject`.
`FieldId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of `Field`.
`IsCascadeDelete`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this object's parent can't be deleted until all records for this object are deleted. Corresponds to `Cascade` value for DeleteConstraint in the Metadata API.
`IsDeprecatedAndHidden`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, this object is unavailable for the current version.
`IsRestrictedDelete`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort

Field	Details
	Description If `true`, this object can't be deleted. Corresponds to `Restrict` value for DeleteConstraint in the Metadata API.
`JunctionIdListName`	**Type** string **Properties** Defaulted on create, Filter, Group, Sort **Description** The name of the list of junction IDs associated with an object. Each ID represents an object that has a relationship with the associated object.
`RelationshipDomains`	**Type** QueryResult **Properties** Filter, Group, Sort **Description** The RelationshipDomain records associated with this object. Because this field represents a relationship, use only in subqueries.

SandboxInfo

Represents a sandbox.

SandboxInfo enqueues a sandbox for creation or refresh. A create operation on SandboxInfo represents creation of a new sandbox, and an update represents refresh of an existing sandbox. For every create or update, a SandboxProcess is automatically created and is used for monitoring the sandbox copy process.

This object is available in API version 35.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`

Supported REST HTTP Methods

`GET`, `PATCH`, `POST`, `DELETE`

Fields

Field	Details
`ApexClassId`	**Type** ID **Properties** Create, Filter, Group **Description** An Apex class that runs after each copy of the sandbox, allowing you to perform business logic on the sandbox to prepare it for use. **Restrictions** • Must be specified during sandbox creation. • The class must extend the `System.SandboxPostCopy` interface. • Available in API version 36.0 and later.
`AutoActivate`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If `true`, you can activate a sandbox refresh immediately. **Restrictions** This field only affects behavior for update operations (Sandbox refresh).
`CopyArchivedActivities`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If `true`, archived activity data is copied to the sandbox. **Restrictions** This field is visible only if your organization has purchased an option to copy archived activities for sandbox. To obtain this option, contact Salesforce Customer Support. You can only set the value to `true` for a Full sandbox.
`CopyChatter`	**Type** boolean **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** If `true`, archived Chatter data is copied to the sandbox.

Field	Details
	Restrictions You can only set the value to `true` for a Full sandbox.
`Description`	**Type** string **Properties** Create, Filter, Nillable, Sort, Update **Description** A description of the sandbox, which is useful if you have more than one sandbox. **Restrictions** Description can't exceed 1,000 characters.
`HistoryDays`	**Type** int **Properties** Create, Defaulted on create, Filter, Group, Sort, Update **Description** Represents the number of days of object history to be copied in the sandbox. Valid values: • `-1`, which means all available days • `0` (default) • `10` • `20` • `30` • `60` • `90` • `120` • `150` • `180` **Restrictions** This field only affects behavior for Full sandboxes.
`LicenseType`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** Represents the sandbox license type. Valid values: • `DEVELOPER` • `DEVELOPER_PRO`

Field	Details
	• PARTIAL • FULL
SandboxName	**Type** string **Properties** Create, Filter, Group, idLookup, Unique, Update **Description** Name of the sandbox. **Restrictions** • Must be a unique sandbox name. • Must be alpha-numeric characters. • Must be ten or fewer characters. • Can’t be the same as a pending deleted sandbox.
TemplateId	**Type** ID **Properties** Create, Filter, Nillable, Sort, Update **Description** ID of the sandbox template associated with this sandbox. A sandbox template lets you select which objects to copy in a sandbox. **Restrictions** • Setting a value for a Partial Copy sandbox is required. • Setting a value for a Full sandbox is optional. • Setting a value for other sandbox types is prohibited, because other sandbox types don’t support sandbox templates.

Usage

SandboxInfo and SandboxProcess work together to manage the creation or refresh of a sandbox.

SandboxProcess

Represents the sandbox copy process for a SandboxInfo record.

When you create a SandboxInfo record, a corresponding SandboxProcess record is created. The latest SandboxProcess record for a SandboxInfo record represents the current state of the sandbox.

This object is available in API version 35.0 and later.

Supported SOAP Calls

`query()`, `retrieve()`, `update()`

Supported REST HTTP Methods

`GET`, `PATCH`

Fields

Except for `RefreshAction`, all fields are read only. The read-only fields represent the attributes chosen on SandboxInfo when a copy process was enqueued, or represent the state of the process for monitoring purposes.

Field	Details
`ActivatedById`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** Represents the user who requested sandbox activation.
`ActivatedDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Represents when the sandbox was activated during a refresh.
`AutoActivate`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Represents whether the sandbox refresh configured to activate immediately upon completion.
`CopyArchivedActivities`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Represents whether archived activity data is copied to the sandbox.
`CopyChatter`	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** Represents whether archived Chatter data is copied to the sandbox.
`CopyProgress`	**Type** int **Properties** Filter, Nillable, Sort **Description** Represents how much of a copy has been completed. Available for Developer, Developer Pro, and Full sandboxes. Not available for Full or Partial sandboxes using sandbox templates.
`Description`	**Type** string **Properties** Filter, Nillable, Sort **Description** A description of the sandbox, which is useful if you have more than one sandbox.
`EndDate`	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Represents when the sandbox copy process finished.
`HistoryDays`	**Type** int **Properties** Defaulted on create, Filter, Group, Sort **Description** Represents the number of days of object history to be copied in the sandbox. Valid values: • `-1`, which means all available days • `0` • `10` • `20` • `30`

Field	Details
	• `60` • `90` • `120` • `150` • `180`
`IsDeleted`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Do not use.
`LicenseType`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The sandbox license type. Valid values: • `DEVELOPER` • `DEVELOPER_PRO` • `PARTIAL` • `FULL`
`RefreshAction`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Editing this field activates or discards a sandbox refresh. Valid values: • `ACTIVATE` • `DISCARD` **Restrictions** If all the following are true, you can activate or discard a sandbox refresh by editing the value in this field: • This record is the latest SandboxProcess record. • The associated sandbox has been refreshed. • This record's `Status` is `Pending Activation`.

Field	Details
SandboxInfoId	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** ID of the SandboxInfo being processed (create or refresh).
SandboxName	**Type** string **Properties** Create, Filter, Group, idLookup, Unique, Update **Description** Name of the sandbox.
StartDate	**Type** dateTime **Properties** Filter, Nillable, Sort **Description** Represents when the sandbox copy process started.
Status	**Type** string **Properties** Group, Nillable, Sort **Description** Current state of the sandbox copy process.
TemplateId	**Type** ID **Properties** Create, Filter, Nillable, Sort, Update **Description** ID of the sandbox template associated with the sandbox for this process. A sandbox template selects which objects to copy in a sandbox.

Usage

SandboxInfo represents a sandbox, and SandboxProcess represents the sandbox copy process, which occurs when you create a sandbox or refresh it. You can also delete a sandbox.

Creating a Sandbox

To enqueue a new sandbox:

1. Create a SandboxInfo record.
2. To find the status of a sandbox after it is enqueued, query SandboxProcess for a given `SandboxInfoId` field to find the latest SandboxProcess record. The value of `Completed` in `Status` indicates that the creation process is finished.

Refreshing a Sandbox

To refresh a sandbox:

1. To start a sandbox refresh, Edit the SandboxInfo record.
2. To find the status of a sandbox after it is enqueued, query SandboxProcess for a given `SandboxInfoId` field to find the latest SandboxProcess record. The value of `Status` indicates the current state of the process.
3. When the `Status` field value is `Pending Activation`, either change the value of the `RefreshAction` field to `ACTIVATE` or `DISCARD`.

Deleting a Sandbox

To delete a sandbox, delete the SandboxInfo record that represents the sandbox, which deletes the sandbox and frees up a license.

SearchLayout

Represents a search layout defined for an object.

This object is available in the Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`, `search()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Supported REST Methods

`GET`

Fields

Field	Details
`ButtonsDisplayed`	**Type** SearchLayoutButtonsDisplayed **Properties** Nillable **Description** The list of buttons available in list views for an object. This field is equivalent to the `Buttons Displayed` value in Object Name List View in the Search Layouts related list on the object detail page. It's also equivalent to the `listViewButtons` field on SearchLayouts in the Metadata API.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. `DurableId` in queries allows you to find the right record without having to retrieve the entire record.
`EntityDefinition`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The name of the object associated with this search layout. Use in subqueries.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the record in `EntityDefinition`. Use in subqueries.
`FieldsDisplayed`	**Type** SearchLayoutFieldsDisplayed **Properties** Nillable **Description** The list of fields displayed in a search result for the object. The name field is required. It's always displayed as the first column header, so it is not included in this list; all additional

Field	Details
	fields are included. The field name relative to the object name, for example MyCustomField__c, is specified for each custom field. This field is equivalent to the Search Results in the Search Layouts related list on the object detail page in the application user interface. It's also equivalent to `searchResultsAdditionalFields` in the Metadata API.
`Label`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The label for this search layout.
`LayoutType`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The type of search layout.

SearchLayoutButton Metadata

Type	Details
`apiName`	**Type** string **Description** The API name of the button.
`label`	**Type** string **Description** The button's label text.

SearchLayoutButtonsDisplayed Metadata

Type	Details
`applicable`	**Type** boolean

Type	Details
	Description If `true`, the buttons listed in `buttons` apply to the object associated with this search layout.
`buttons`	**Type** string **Description** The list of buttons on the object associated with this search layout.

SearchLayoutField Metadata

Type	Details
`apiName`	**Type** string **Description** The API name of the field.
`label`	**Type** string **Description** The field's label text.
`sortable`	**Type** boolean **Description** If `true`, the fields can be sorted.

SearchLayoutFieldsDisplayed Metadata

Type	Details
`applicable`	**Type** boolean **Description** If `true`, the fields listed in `fields` are available in the object associated with this search layout.
`fields`	**Type** string

Type	Details
	Description The list of fields on the object associated with this search layout.

ServiceFieldDataType

Don't use this object.

This object is visible in version 34.0 for some organizations, but we plan to remove it in a later release.

Scontrol

Represents a custom s-control, which is custom content that our system hosts, but client applications execute. An s-control can contain any type of content that you can display or run in a Web browser.

Important: Visualforce pages supersede s-controls. Organizations that haven't previously used s-controls can't create them. Existing s-controls are unaffected, and can still be edited. We recommend that you move your s-controls to Visualforce. We continue to support this object.

Supported SOAP Calls

`query()`

Supported REST Methods

`GET`

Special Access Rules

- Your organization must be using Enterprise, Developer, or Unlimited Edition and be enabled for custom s-controls.
- Customer Portal users can't access this object.

Fields

Field	Details
`ContentSource`	**Type** picklist **Properties** Filter, Group, Nillable, Restricted picklist, Sort, Update **Description** Specify the source of the s-control content, either custom HTML, a snippet (s-controls that are included in other s-controls), or a URL.

Field	Details
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort, Update **Description** Description of the custom s-control.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort, Update **Description** The unique name of the object in the API. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores. In managed packages, this field prevents naming conflicts on package installations. With this field, a developer can change the object's name in a managed package and the changes are reflected in a subscriber's organization. Label is **S-Control Name**. Note: When creating large sets of data, always specify a unique `DeveloperName` for each record. If no `DeveloperName` is specified, Salesforce generates one for each record, which slows performance.
`EncodingKey`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort, Update **Description** Picklist of character set encodings, including ISO-08859-1, UTF-8, EUC, JIS, Shift-JIS, Korean (ks_c_5601-1987), Simplified Chinese (GB2312), and Traditional Chinese (Big5).
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released`

Field	Details
	- `unmanaged`
`Name`	**Type** string **Properties** Filter, Group, Sort, Update **Description** Required. Name of this custom s-control. Label is **Label**.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: - In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field is the namespace prefix of the Developer Edition organization of the package developer. - In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`SupportsCaching`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort, Update **Description** Indicates whether the s-control supports caching (`true`) or not (`false`).

Usage

Use custom s-controls to manage custom content that extends application functionality. All users can view custom s-controls, but the “Customize Application” permission is required to create or update custom s-controls.

SOQLResult

A complex type that represents the result of a SOQL query in an ApexExecutionOverlayResult object. Available from API version 28.0 or later.

Fields

Field	Details
`queryError`	**Type** string **Description** The error text returned if the execution was unsuccessful.
`queryMetadata`	**Type** QueryResultMetadata **Description** The structured result returned from a successful execution. QueryResultMetadata includes the following fields: • `columnMetadata` • `entityName` • `groupBy` • `idSelected` • `keyPrefix`
`queryResult`	**Type** array of MapValue **Description** MapValue contains an array of MapEntry, which contains the following fields: • `keyDisplayValue` • `value` (reference to StateValue)

Usage

Overlay SOQL on checkpoints to capture structured debugging information. If your SOQL query may return more than one record when dealing with complex types, select only one row. For example, you can use a LIMIT=1 clause in your SOQL query, or you can list rows for the user and have them select the row to inspect.

StandardAction

Represents the buttons, links, and actions (standard actions) for a standard or custom object. This object is available in API version 34.0 and later.

You can view the standard actions from an object's management settings by going to Buttons, Links, and Actions.

Supported SOAP Calls

`query()`

Supported REST HTTP Methods

`GET`

Fields

Field	Details
`ContentType`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** Indicates whether the button or link is standard, URL, s-control, JavaScript action, or Visualforce page. This value maps to the `Content Source` field in the user interface.
`Description`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Text displayed to an administrator in the standard action setup page. `Description` can be different from the `Label`, which is the label displayed in the user interface to end users. It can also be different from `Name`, which is a unique string used in merge fields.
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. `DurableId` in queries allows you to find the right record without having to retrieve the entire record.

Field	Details

`EntityDefinition`	**Type**
	EntityDefinition
	Properties
	Filter, Group, Nillable, Sort
	Description
	The entity definition of the standard or custom object for which these standard actions are defined.
	For example, to find all the standard actions for Account, use a query similar to the following:
	`SELECT Label, EntityDefinition.Label`
	`FROM StandardAction`
	`WHERE EntityDefinition.QualifiedApiName = 'Account'`

`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the standard or custom object for which this standard action is defined.
`IsOverridden`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** Indicates whether this standard action has been overridden (`true`, or not.
`Label`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The text that displays in a user interface for the standard action.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Filter, Group, Restricted picklist, Sort **Description** Indicates the action’s manageable state. Valid values: • `beta`

<table>
<thead>
<tr><th>Field</th><th>Details</th></tr>
</thead>
<tbody>
<tr><td></td><td>
<ul>
<li><code>deleted</code></li>
<li><code>deprecated</code></li>
<li><code>installed</code></li>
<li><code>released</code></li>
<li><code>unmanaged</code></li>
</ul>
</td></tr>
<tr><td><code>Name</code></td><td>
<b>Type</b>
string
<b>Properties</b>
Filter, Group,Sort
<b>Description</b>
The unique name for the button or link when referenced from a merge field. This name can contain only underscores and alphanumeric characters, and must be unique in your organization. It must begin with a letter, not include spaces, not end with an underscore, and not contain two consecutive underscores.
</td></tr>
<tr><td><code>OverrideContent</code></td><td>
<b>Type</b>
<code>tns:</code>Name
<b>Properties</b>
Filter, Group,Sort
<b>Description</b>
Reference to the actions that override standard actions for an object.
To retrieve information, use this field with the fields in Name, because you can’t query the field directly.
For example, assume that you have overridden a standard action for Account. The following query returns the label of the standard action, and the name and ID of the action that overrides the standard action.
<pre>
SELECT Label, OverrideContent.Id, OverrideContent.Name
FROM StandardAction
WHERE EntityDefinition.QualifiedAPiName='Account'
      AND IsOverriden=true
</pre>
Because OverrideContent is a Name object, you have access to all the fields in Name, in this case Name.<code>Name</code> and Name.<code>Id</code>.
</td></tr>
<tr><td><code>OverrideContentId</code></td><td>
<b>Type</b>
string
<b>Properties</b>
Filter, Group, Nillable, Sort
<b>Description</b>
ID of an <code>OverrideContent</code> record. Returns the same value as <code>OverrideContent.Id</code> in the sample SOQL query for <code>OverrideContent</code>.
</td></tr>
</tbody>
</table>

StaticResource

Represents the working copy of a static resource file for editing or saving. Static resources allow you to upload content that you can reference in a Visualforce page, including images, stylesheets, JavaScript, and other files. Available in Tooling API version 29.0 and later.

Supported SOAP API Calls

`create()`, `delete()`, `describeSObjects()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST API HTTP Methods

`Query`, `GET`, `POST`, `PATCH`, `DELETE`

Fields

Field Name	Details
`Name`	**Type** string **Properties** Create, Update **Description** The static resource name. The name can only contain characters, letters, and the underscore (_) character, must start with a letter, and cannot end with an underscore or contain two consecutive underscore characters
`Body`	**Type** string **Properties** Create, Update **Description** The data for the static resource file.
`ContentType`	**Type** string **Properties** Create, Update **Description** Required. The content type of the file, for example text/plain.
`CacheControl`	**Type** string **Properties** Create, Update

Field Name	Details
	Description Required. Indicates whether the static resource is marked with a public caching tag so that a third-party delivery client can cache the content. The valid values are: - Private - Public

Usage

To create, edit, or save a static resource file, create a StaticResource object that references it.

SymbolTable

A complex type that represents all user-defined tokens in the `Body` of an ApexClass, ApexClassMember, or ApexTriggerMemeber and their associated line and column locations within the `Body`.

Fields

Field	Details
`constructors`	**Type** array of Constructor **Description** Contains the position, scope, and signature of constructors for the Apex class. Apex triggers don't have constructors. Constructor includes the following fields: - `annotations` - `location` - `modifiers` - `name` - `references` - `visibility` (available only in API versions 33.0 and earlier; scope: Global, Public or Private) - `parameters`
`externalReferences`	**Type** array of ExternalReference

Field	Details
	Description Contains the name, namespace, external class, method, and variable references for the Apex class or trigger. These references can be used for symbol highlighting or code navigation. ExternalReference includes the following fields: - `methods` - `name` - `namespace` - `references` - `variables`
`innerClasses`	**Type** array of SymbolTable **Description** Contains a symbol table for each inner class of the Apex class or trigger.
`interfaces`	**Type** array of String **Description** Contains a set of strings for each interface with the namespace and name, for example: `['System.Batchable', 'MyNamespace.MyInterface']`.
`methods`	**Type** array of Method **Description** Contains the position, name, scope, signature, and return type of available Apex methods. Method includes the following fields: - `annotations` - `location` - `modifiers` - `name` - `references` - `visibility` (available only in API versions 33.0 and earlier; scope: Global, Public or Private) - `parameters` - `returnType`
`name`	**Type** string **Description** The name of the Apex class or trigger.

Field	Details
`namespace`	**Type** string **Description** The namespace of the Apex class or trigger. Null if there is no namespace.
`parentClass`	**Type** string **Description** Returns parents of inner classes and extending classes.
`properties`	**Type** array of VisibilitySymbol **Description** Contains the position, name, scope, and references of properties for the Apex class or trigger. VisibilitySymbol includes the following fields: - `annotations` - `location` - `modifiers` - `name` - `references` - `visibility` (available only in API versions 33.0 and earlier; scope: Global, Public or Private)
`tableDeclaration`	**Type** array of Symbol **Description** Contains the position, name, and references of the Apex class or trigger. Symbol includes the following fields: - `annotations` - `location` - `modifiers` - `name` - `references`
`variables`	**Type** array of Symbol **Description** Contains the position, name, and references of related variables. Symbol includes the following fields: - `annotations`

Field	Details
	• location • modifiers • name • references

Annotations

Available values for `annotations` fields include:

- `Deprecated`
- `Future`
- `HttpDelete`
- `HttpGet`
- `HttpPatch`
- `HttpPost`
- `HttpPut`
- `InvocableMethod`
- `InvocableVariable`
- `IsTest`
- `ReadOnly`
- `RemoteAction`
- `TestSetup`
- `TestVisible`
- `RestResource`

Modifiers

Modifiers can include more values than those values explicitly specified in classes and methods. All relevant modifiers, including implicit ones, are now returned. For example, all `webservice` methods have an implicit `global` modifier. Also, because fields and methods are private unless otherwise specified, the `private` modifier is returned by default.

The `testMethod` modifier is returned when either the `testMethod` modifier or the `IsTest` annotation is used.

Available values for `modifiers` fields include:

- `abstract`
- `final`
- `global`
- `override`
- `private`
- `protected`
- `public`

- static
- testMethod
- transient
- virtual
- webService
- with sharing
- without sharing

Usage

Use symbol tables instead of building a parser or compiler. Symbol tables allow you to do symbol highlighting, code navigation, code completion, symbol searches, and more.

A symbol table can't be created if the content referenced by the ContentEntityId field doesn't use a symbol table. Compiler errors for the last deployment of the MetadataContainer in the MetadataContainerId field also prevent a symbol table from being created.

TraceFlag

Represents a trace flag that triggers an Apex debug log at the specified logging level.

Supported SOAP API Calls

create(), delete(), describeSObjects(), query(), retrieve(), update(), upsert()

Supported REST API HTTP Methods

Query, GET, POST, PATCH, DELETE

Fields

Field Name	Details
ApexCode	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for Apex code. Includes information about Apex code and can include information such as log messages generated by data manipulation language (DML) statements, inline SOQL or SOSL queries, the start and completion of any triggers, the start and completion of any test method, and so on. The following are valid values. - NONE

Field Name	Details
	• `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`ApexProfiling`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for profiling information. Includes cumulative profiling information, such as the limits for your namespace, the number of emails sent, and so on. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO` • `DEBUG` • `FINE` • `FINER` • `FINEST` This field is required.
`Callout`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for callouts. Includes the request-response XML that the server is sending and receiving from an external Web service. The request-response XML is useful when debugging issues related to SOAP API calls. The following are valid values. • `NONE` • `ERROR` • `WARN` • `INFO`

Field Name	Details
	• DEBUG • FINE • FINER • FINEST This field is required.
Database	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category for database activity. Includes information about database activity, including every DML statement or inline SOQL or SOSL query. The following are valid values. • NONE • ERROR • WARN • INFO • DEBUG • FINE • FINER • FINEST This field is required.
DebugLevelId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the debug level assigned to this trace flag. A debug level, which is a set of log category levels, can be assigned to multiple trace flags.
ExpirationDate	**Type** dateTime **Properties** Create, Filter, Sort, Update **Description** The date and time that the trace flag expires. `ExpirationDate` must be less than 24 hours after `StartDate`. If `StartDate` is null, `ExpirationDate` must be less than 24 hours from the current time. This field is required.

Field Name	Details
LogType	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort **Description** The type of log to generate. The following are valid values. - CLASS_TRACING - DEVELOPER_LOG - PROFILING (reserved for future use) - USER_DEBUG When you open the Developer Console, it sets a `DEVELOPER_LOG` trace flag to log your activities. `USER_DEBUG` trace flags cause logging of an individual user's activities. `CLASS_TRACING` trace flags override logging levels for Apex classes and triggers, but don't generate logs. This field is required.
ScopeId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** **Deprecated.** This field is available in API version 34.0 and earlier. A reference to a user. This field is used with the `TracedEntityID` field. - When the value is `user`, the actions of the user or entity specified by `TracedEntityID` are traced to the system log at the described level. System logs are visible only to you. Use this scope for class-level filtering. If there are both user and entity-level flags, the user flags take precedence until a method from a class with an entity trace flag is entered. When the method returns, the user trace flags are restored. - When the value is ***emptyid***, the user's actions are traced to the organization's debug log at the described level. Debug logs are visible to all administrators. This option is only available if `TracedEntityID` references a user (not an Apex class or Apex trigger). The variable *emptyid* can be the value 000000000000000 or null. The scope defined here is reflected in the ApexLog `Location` field.
StartDate	**Type** dateTime **Properties** Create, Filter, Group, Nillable, Sort, Update

Field Name	Details
	Description The date and time when the trace flag takes effect.
`System`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for calls to all system methods, such as the `System.debug` method. The following are valid values. - `NONE` - `ERROR` - `WARN` - `INFO` - `DEBUG` - `FINE` - `FINER` - `FINEST` This field is required.
`TracedEntityId`	**Type** reference **Properties** Create, Filter, Group, Sort, Update **Description** A reference to the following: - Apex class - Apex trigger - User This field is used with the `LogType` field. This field is required.
`Validation`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for validation rules. Includes information about validation rules, such as the name of the rule, or whether the rule evaluated true or false. The following are valid values. - `NONE`

Field Name	Details
	- `ERROR` - `WARN` - `INFO` - `DEBUG` - `FINE` - `FINER` - `FINEST` This field is required.
`Visualforce`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for Visualforce. Includes information about Visualforce events, including serialization and deserialization of the view state or the evaluation of a formula field in a Visualforce page. The following are valid values. - `NONE` - `ERROR` - `WARN` - `INFO` - `DEBUG` - `FINE` - `FINER` - `FINEST` This field is required.
`Workflow`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The log category level for workflow rules. Includes information for workflow rules, such as the rule name and the actions taken. This field is required. The following are valid values. - `NONE` - `ERROR` - `WARN` - `INFO` - `DEBUG`

Field Name	Details
	- `FINE` - `FINER` - `FINEST`

Usage

To diagnose a functional issue or a performance problem, use the TraceFlag object to set up logging for yourself or for another user. The following options are available.

- To set up logging for a specific user, set `LogType` to `USER_DEBUG` and `TracedEntityId` to the ID of the user. This option can be configured only for a user, not for an Apex class or Apex trigger.
- To set up logging level overrides for an Apex class or trigger, set `LogType` to `CLASS_TRACING` and `TracedEntityId` to the ID of the Apex class or trigger. `CLASS_TRACING` trace flags override other logging levels, but don't cause logs to be generated or persisted.

TransactionSecurityPolicy

Represents a transaction security policy definition (policy).

This object is available in Tooling API version 35.0 and later.

Supported Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST Methods

`Query`, `DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`ActionConfig`	**Type** string **Properties** Create, Update **Description** Describes the action to take when the matching Transaction Security policy is triggered. Multiple actions can be taken. The actions available depend on the `Event Type` field.

<table>
<tr><th>Field</th><th>Details</th></tr>
<tr><td></td><td>
<table>
<tr><th>Event Type</th><th>Available Actions</th></tr>
<tr><td><code>AccessResource</code></td><td>• Block
• Two-factor authentication</td></tr>
<tr><td><code>DataExport</code></td><td>None; only notifications are available.</td></tr>
<tr><td><code>Entity</code></td><td>None; only notifications are available.</td></tr>
<tr><td><code>Login</code></td><td>• Block
• Two-factor authentication
• End an existing session</td></tr>
</table>
</td></tr>
<tr><td><code>ApexPolicyId</code></td><td><b>Type</b>
reference
<b>Properties</b>
Create, Filter, Group, Nillable, Sort, Update
<b>Description</b>
Represents the Apex <code>TxnSecurity.PolicyCondition</code> interface for this policy.</td></tr>
<tr><td><code>DeveloperName</code></td><td><b>Type</b>
string
<b>Properties</b>
Create, Filter, Group, Sort, Update
<b>Description</b>
The API, or program name, for this policy.</td></tr>
<tr><td><code>EventType</code></td><td><b>Type</b>
picklist
<b>Properties</b>
Create, Filter, Group, Nillable, Restricted picklist, Sort, Update
<b>Description</b>
Indicates the type of event the policy monitors. Valid values are:
• <code>AccessResource</code>—Notifies you when the selected resource has been accessed.
• <code>AuditTrail</code>—Reserved for future use.
• <code>DataExport</code>—Notifies you when the selected object type has been exported using the Data Loader API client.
• <code>Entity</code>—Notifies you on use of an object type such as an authentication provider or client browser.
• <code>Login</code>—Notifies you when a user logs in.</td></tr>
</table>

Field	Details
ExecutionUserId	**Type** reference **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** The ID of the user to notify when the policy is triggered. This user must be active and assigned the System Administrator profile.
MasterLabel	**Type** string **Properties** Create, Filter, Group, Sort, Update **Description** The policy's name.
NamespacePrefix	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***namespacePrefix__componentName*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, NamespacePrefix is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
ResourceName	**Type** string **Properties** Create, Filter, Group, Nillable, Sort, Update **Description** A resource used to narrow down the conditions under which the policy triggers. For example, with a Login event, you can add a resource to specify which login URL triggers the policy. The resources available depend on the Event Type field.

<table>
<tr><th>Field</th><th>Details</th></tr>
<tr><td></td><td>
<table>
<tr><th>Event Type</th><th>Available Actions</th></tr>
<tr><td><code>AccessResource</code></td><td><ul><li>EventTimestamp</li><li>SessionLevel</li><li>SourceIp</li></ul></td></tr>
<tr><td><code>DataExport</code></td><td><ul><li>EventTimestamp</li><li>SessionLevel</li><li>SourceIp</li></ul></td></tr>
<tr><td><code>Entity</code></td><td><ul><li>AuthorizeUrl</li><li>ConsumerKey</li><li>ConsumerSecret</li><li>DefaultScopes</li><li>DeveloperName</li><li>ErrorUrl</li><li>FriendlyName</li><li>IconUrl</li><li>IdTokenIssuer</li><li>LogoutUrl</li><li>TokenUrl</li><li>UserInfoUrl</li></ul></td></tr>
<tr><td><code>Login</code></td><td><ul><li>ApiType</li><li>ApiVersion</li><li>Application</li><li>Browser</li><li>ClientVersion</li><li>LoginUrl</li><li>Platform</li><li>Status</li></ul></td></tr>
</table>
</td></tr>
<tr><td><code>State</code></td><td><b>Type</b>
picklist
<b>Properties</b>
Create, Filter, Group, Restricted picklist, Sort, Update
<b>Description</b>
Indicates whether the policy is active. Valid values are:</td></tr>
</table>

Field	Details
	- `Disabled` - `Enabled`
`Type`	**Type** picklist **Properties** Create, Filter, Group, Restricted picklist, Sort, Update **Description** The type of validation that the policy uses. The only valid value is `CustomApexPolicy`.

User

Represents a user. You can retrieve standard fields on User with the Tooling API, but custom fields can't be retrieved.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`describeLayout()`, `getDeleted()`, `getUpdated()`, `query()`, `retrieve()`, `search()`, `update()`

Supported REST HTTP Methods

`GET`, `PATCH`

Fields

Field	Details
`FirstName`	**Type** string **Properties** Create, Filter, Group, NillableSort, Update **Description** The user's first name.
`LastName`	**Type** string **Properties** Create, Filter, Group, NillableSort, Update **Description** The user's last name.

Field	Details
`Name`	**Type** string **Properties** Filter, Group, Sort **Description** Concatenation of `FirstName` and `LastName`. Limited to 121 characters.
`Username`	**Type** string **Properties** Create, Filter, Group, idLookup, Sort, Update **Description** The name of the user in your organization.
`WorkspaceId`	**Type** ID **Properties** Filter, Group, Nillable, Sort, Update **Description** The ID of the last open Developer Console workspace.

UserEntityAccess

Represents the access that the current user has to an object. Available in Tooling API version 34.0 and later.

Supported SOAP Calls

`query()`, `search()`

Supported REST HTTP Methods

`GET`

Limitations

SOQL Limitations on page 17

SOSL Limitations on page 18

Fields

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn't guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Nillable, Sort **Description** The entity definition for the object associated with this user entity access record. Because this field represents a relationship, use only in subqueries.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the EntityDefinition.
`IsActivateable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to activate records of the associated object type if the User owns them. For example, a user owns an Apex trigger or workflow rule, and can activate them if this field is `true` for ApexTrigger or WorkflowRule.
`IsCreatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to create records of the associated object type.

Field	Details
`IsDeletable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to delete records of the associated object type.
`IsEditable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to edit records of the associated object type.
`IsFlsUpdatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to change field-level security settings on appropriate fields of the associated object type. For example, an administrator could deny a group of users access to the `Type` field on Account.
`IsMergeable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to merge records of the associated object type.
`IsReadable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to view records of the associated object type.

Field	Details
`IsUndeletable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to undelete records of the associated object type.
`IsUpdatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to edit records of the associated object type.
`User`	**Type** User **Properties** Filter, Group, Nillable, Sort **Description** The user who has the access defined in this user entity access record, for the entity specified in the `EntityDefinition` field. Because this field represents a relationship, use only in subqueries.
`UserId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** ID of the user specified in the `User` field.

Usage

Queries on UserEntityAcces need filters on both the entity side and the user side.

Example: Entity Side

```
SELECT EntityDefinition.QualifiedApiName, EntityDefinition.MasterLabel
FROM UserEntityAccess WHERE UserId={current_user_id}
AND IsCreateable=true AND EntityDefinition.IsCustomizable=true
```

Example: User Side

```
UserId={current_user_id}
```

UserFieldAccess

Represents the access that the current user has to a field. Available in Tooling API version 34.0 and later.

Fields

Field	Details
`DurableId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Unique identifier for the field. Always retrieve this value before using it, as the value isn’t guaranteed to stay the same from one release to the next. To simplify queries, use this field.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Nillable, Sort **Description** The entity definition for the object associated with this user entity access record.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** ID of the EntityDefinition.
`IsAccessible`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to view the associated field.
`IsCreatable`	**Type** boolean

Field	Details
	Properties Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to create records of the associated field.
`IsUpdatable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If `true`, the user specified in the `User` field has access to edit the associated field.
`User`	**Type** User on page 250 **Properties** Filter, Group, Nillable, Sort **Description** The user who has access defined in this user field access record, for the entity specified in the `EntityDefinition` field. Because this field represents a relationship, use only in subqueries.
`UserId`	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** ID of the user specified in the `User` field.

SOQL Limitations

This object doesn't support some SOQL operations.

GROUP BY

Example Query: `SELECT COUNT(qualifiedapiname), isfeedenabled FROM EntityDefinition GROUP BY isfeedenabled`

Error Returned: `The requested operation is not yet supported by this SObject storage type, contact salesforce.com support for more information.`

LIMIT, LIMIT OFFSET

Example Queries:

```
SELECT qualifiedapiname FROM EntityDefinition LIMIT 5
```

```
SELECT qualifiedapiname FROM EntityDefinition LIMIT 5 OFFSET 10
```

An incorrect result is returned because LIMIT and LIMIT OFFSET are ignored.

NOT

Example Query: `SELECT qualifiedapiname FROM EntityDefinition WHERE qualifiedapiname!='Account'`

Error Returned: `Only equals comparisons permitted`

OR

Example Query: `SELECT qualifiedapiname, keyprefix FROM EntityDefinition WHERE isdeletable=true OR (isfeedenabled=false AND keyprefix='01j')`

Error Returned: `Disjunctions not supported`

ValidationRule

Represents a validation rule or workflow rule which specifies the formula for when a condition is met. Available from API version 34.0 or later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `update()`, `upsert()`

Supported REST HTTP Methods

`Query`, `GET`, `POST`, `PATCH`

Fields

Field Name	Details
`Active`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort. **Description** Required. Indicates whether this validation rule is active, (`true`), or not active (`false`).
`Description`	**Type** string **Properties** Filter, Nillable, Sort. **Description** A description of the validation rule.

Field Name	Details
EntityDefinition	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. The entity definition for the object associated with the validation rule.
EntityDefinitionId	**Type** string **Properties** Filter, Group, Sort. **Description** Required. ID of the record in `EntityDefinition`.
ErrorDisplayField	**Type** string **Properties** Filter, Group, Nillable, Sort. **Description** The fully specified name of a field in the application. If a value is supplied, the error message appears next to the specified field. If you do not specify a value or the field isn’t visible on the page layout, the value changes automatically to `Top of Page`.
ErrorMessage	**Type** string **Properties** Filter, Group, Nillable, Sort . **Description** Required. The message that appears if the validation rule fails. The message must be 255 characters or less.
FullName	**Type** string **Properties** Create, Group, Nillable. **Description** The internal name of the object. White spaces and special characters are escaped for validity. The name must: • Contain characters, letters, or the underscore (_) character • Must start with a letter • Can’t end with an underscore

Field Name	Details
	- Can't contain two consecutive underscore characters. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Id`	**Type** `Id` **Properties** Defaulted on create, Filter, Group, idLookup, Sort. **Description** The unique system ID for this record.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - `beta` - `deleted` - `deprecated` - `installed` - `released` - `unmanaged`
`Metadata`	**Type** ValidationRule Metadata **Properties** Create, Nillable, Update. **Description** Validation rule metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort.

Field Name	Details
	Description The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***`namespacePrefix__componentName`*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`ValidationName`	**Type** string **Properties** Filter, Group, Namefield, Sort. **Description** The name or ID of the object that this rule is associated with.

ValidationRule Metadata

`active`, `description`, `errorDisplayField`, and `errorMessage` are described in the previous table.

Field Name	Details
`errorConditionFormula`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** Required. The formula defined in the validation rule. If the formula returns a value of `true`, an error message is displayed.

WebLink

Represents a custom link to a URL or S-control. Available in the Tooling APIfrom API version 34.0 or later.

Supported SOAP Calls

`getUpdated()`, `query()`, `retrieve()`, `search()`

Supported REST HTTP Methods

`GET`

Fields

Field Name	Details
`Description`	**Type** string **Properties** Filter, Nillable, Sort **Description** A description of the WebLink.
`DisplayType`	**Type** string **Properties** Filter, Group, Restricted picklist, Sort **Description** Represents how this WebLink is rendered. Valid values: • `link` for a hyperlink • `button` for a button • `massAction` for a button attached to a related list
`EncodingKey`	**Type** string **Properties** Filter, Sort **Description** Valid values include: • `UTF-8`—Unicode (UTF-8) • `ISO-8859-1`—General US & Western Europe (ISO-8859–1, ISO-LATIN-1) • `Shift_JIS`—Japanese (Shift-JIS) • `ISO-2022-JP`—Japanese (JIS) • `EUC-JP`—Japanese (EUC-JP) • `x-SJIS_0213`—Japanese (Shift-JIS_2004) • `ks_c_5601-1987`—Korean (ks_c_5601-1987)

Field Name	Details
	- `Big5`—Traditional Chinese (Big5) - `GB2312`—Simplified Chinese (GB2312) - `Big5-HKSCS`—Traditional Chinese Hong Kong (Big5–HKSCS)
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. Available in version 34.0. The entity definition for the object associated with this WebLink.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort. **Description** Required. ID of the record associated with this WebLink. The record’s object type is in `EntityDefinition`.
`FullName`	**Type** string **Properties** Filter, Group, Sort. **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`HasMenubar`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort. **Description** If `OpenType` is `newWindow`, whether to show the browser menu bar for the popup window (`true`, or not (`false`). For other values of `OpenType`, don't specify a value here.
`HasScrollbars`	**Type** boolean

Field Name	Details
	Properties Defaulted on create, Filter, Group, Sort. **Description** If the value of `OpenType` is `newWindow`, whether to show the scroll bars for the window (`true`) or not (`false`). For other values of `OpenType`, don't specify a value here.
`HasToolbar`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort. **Description** If the value of `OpenType` is `newWindow`, whether to show the browser toolbar for the window (`true`) or not (`false`). For other values of `OpenType`, don't specify a value here.
`Height`	**Type** int **Properties** Filter, Group, Nillable, Sort. **Description** Required if the value of `OpenType` is `newWindow`. Height in pixels of the window opened by this WebLink. For other values of `OpenType`, don't specify a value here.
`IsResizable`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort. **Description** If the value of `OpenType` is `newWindow`, whether to allow resizing of the window (`true`) or not (`false`). For other values of `OpenType`, don't specify a value here.
`LinkType`	**Type** WebLinkType enumerated list **Properties** Filter, Group, Restricted picklist, Sort **Description** Required. Represents whether the content of this WebLink is specified by a URL, an sControl, a JavaScript code block, or a Visualforce page. • `url`

Field Name	Details
	• sControl • javascript • page • flow—Reserved for future use.
ManageableState	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • beta • deleted • deprecated • installed • released • unmanaged
MasterLabel	**Type** string **Properties** Filter, Group, Nillable, Sort, Update **Description** Master label for this object. This display value is the internal label that is not translated. Limit: 240 characters.
Metadata	**Type** mns:WebLink **Properties** Filter, Group, idLookup, Sort **Description** The metadata for this object as defined in the Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
Name	**Type** string **Properties** Filter, Group, idLookup, Sort

Field Name	Details
	Description Required. Name to display on the page.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Sort. **Description** The namespace prefix associated with this object. Each Developer Edition organization that creates a managed package has a unique namespace prefix. Limit: 15 characters. You can refer to a component in a managed package by using the ***namespacePrefix__componentName*** notation. The namespace prefix can have one of the following values: • In Developer Edition organizations, the namespace prefix is set to the namespace prefix of the organization for all objects that support it. There is an exception if an object is in an installed managed package. In that case, the object has the namespace prefix of the installed managed package. This field's value is the namespace prefix of the Developer Edition organization of the package developer. • In organizations that are not Developer Edition organizations, `NamespacePrefix` is only set for objects that are part of an installed managed package. There is no namespace prefix for all other objects.
`OpenType`	**Type** WebLinkWindowType enumerated list **Properties** Filter, Group, Sort **Description** Valid values: • `newWindow` • `sidebar` • `noSidebar` • `replace` • `onClickJavaScript`
`Position`	**Type** WebLinkPosition enumerated list **Properties** Filter, Group, Nillable, Restricted picklist, Sort **Description** If the value of `OpenType` is `newWindow`, how the new window should be displayed. Otherwise, don't specify a value. Valid values:

Field Name	Details
	- `fullScreen` - `none` - `topLeft`
`RequireRowSelection`	**Type** boolean **Properties** Defaulted on create, Filter, Group, Sort **Description** If the value of `OpenType` is `massAction`, indicates whether to require individual row selection to execute the action for this button (`true`) or not (`false`). Otherwise, leave this field empty.
`Scontrol`	**Type** string **Properties** Filter, Group, Sort **Description** If the value of `LinkType` is `sControl`, this field represents the name of the sControl. Otherwise, leave this field empty.
`ShowsLocation`	**Type** boolean **Properties** Filter, Group, Sort **Description** If the value of `OpenType` is `newWindow`, indicates whether to show the browser location bar for the window (`true`) or not (`false`). Otherwise, leave this field empty.
`ShowsStatus`	**Type** boolean **Properties** Filter, Group, Sort **Description** If the value of `OpenType` is `newWindow`, show the browser status bar for the window (`true`. Otherwise, don't specify a value.
`Url`	**Type** string **Properties** Filter, Group, Nillable, Sort

Field Name	Details
	Description If the value of `LinkType` is `url`, `Url` represents the URL value. If the value of `LinkType` is `javascript`, `Url`represents the JavaScript content. For other values of `LinkType`, leave this field empty. Content must be escaped in a manner consistent with XML parsing rules. Required. URL of the page to link to. Can include fields as tokens within the URL. Limit: 1,024 KB.
`Width`	**Type** int **Properties** Filter, Group, Nillable, Sort **Description** Width in pixels of the window opened by this WebLink. Required if the value of `OpenType` is `newWindow`. Otherwise, leave this field empty.

WorkflowAlert

Represents a workflow alert. A workflow alert is an email generated by a workflow rule or approval process and sent to designated recipients.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`CcEmails`	**Type** string **Properties** Filter, Nillable, Sort

Field	Details
	Description Additional CC email addresses.
`Description`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** A description of the workflow alert.
`DeveloperName`	**Type** string **Properties** Filter, Group, Sort **Description** The unique name of the workflow alert in the API.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. Available in version 34.0. The entity definition for the object associated with this WebLink.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the alert.
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field	Details
ManageableState	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - beta - deleted - deprecated - installed - released - unmanaged
Metadata	**Type** mns:WorkflowAlert **Properties** Create, Nillable, Update **Description** Alert definition metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
NamespacePrefix	**Type** string **Properties** Filter, Group, Sort **Description** The namespace of the package to uniquely identify the workflow alert.
SenderType	**Type** ActionEmailSenderType enumerated list **Properties** Defaulted on create, Filter, Group, Restricted picklist, Sort **Description** The type of sender. Values are: - CurrentUser - OrgWideEmailAddress - DefaultWorkflowUser

Field	Details
`TemplateId`	**Type** ID **Properties** Filter, Group, Sort **Description** A reference to an email template.

WorkflowFieldUpdate

Represents a workflow field update.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. Available in version 34.0. The entity definition for the object associated with this WebLink.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the workflow field update.

Field	Details
FieldDefinition	**Type** FieldDefinition on page 131 **Properties** Filter, Group, Sort **Description** Required. The definition of this field.
FieldDefinitionId	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the field for the workflow field update.
FullName	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
LiteralValue	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** If the update uses a literal value, this is that value.
LookupValueId	**Type** ID **Properties** Filter, Group, Nillable, Sort **Description** If the update looks up a value, this is the lookup value referenced.
ManageableState	**Type** ManageableState enumerated list

Field	Details
	Properties Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** `mns`:WorkflowFieldUpdate **Properties** Create, Nillable, Update **Description** The workflow field update metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Name`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The name of the workflow field update.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package containing the workflow field update object.
`SourceTableEnumOrId`	**Type** picklist **Properties** Filter, Group, Restricted picklist, Sort **Description** The enum (for example, Account) or ID of the object this workflow field update is on.

WorkflowOutboundMessage

Represents an outbound message. An outbound message is a workflow, approval, or milestone action that sends the information you specify to an endpoint you designate, such as an external service. Outbound messaging is configured in the Salesforce setup menu. Then you must configure the external endpoint. You can create a listener for the messages using the SOAP API.

This object is available in API version 32.0 and later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`DELETE`, `GET`, `PATCH`, `POST`

Fields

Field	Details
`ApiVersion`	**Type** double **Properties** Filter, Sort **Description** The API version is automatically generated and set to the current API version when the outbound message was created.
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. Available in version 34.0. The entity definition for the object associated with this WebLink.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the outbound message.
`FullName`	**Type** string

Field	Details
	Properties Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`IntegrationUserId`	**Type** ID **Properties** Filter, Group, Sort **Description** The ID of the user under which this message is sent.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** `mns`:WorkflowOutboundMessage **Properties** Create, Nillable, Update **Description** Outbound message definition metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`Name`	**Type** string

Field	Details
	Properties Filter, Group, idLookup, Sort **Description** The name of the outbound message.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package containing the outbound message.

WorkflowRule

Represents a workflow rule that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

Available from API version 30.0 or later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`Query`, `DELETE`, `GET`, `PATCH`, `POST`

Fields

Field Name	Details
`FullName`	**Type** string **Properties** Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.

Field Name	Details
ManageableState	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: - beta - deleted - deprecated - installed - released - unmanaged
Metadata	**Type** mns:WorkflowRule **Properties** Create, Nillable, Update **Description** Workflow rule metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
Name	**Type** string **Properties** Filter, Group, Sort **Description** The enum name or ID of entity this rule is associated with.
NamespacePrefix	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** The namespace of the package containing the workflow rule object.
TableEnumOrId	**Type** picklist

Field Name	Details
	Properties Filter, Group, Restricted picklist, Sort **Description** The enum (for example, Account) or ID of the object for this workflow rule.

WorkflowTask

Represents a workflow task that is used to fire off a specific workflow action when the specified criteria is met. Includes access to the associated WorkflowRule object in Salesforce Metadata API.

Available from API version 32.0 or later.

Supported SOAP Calls

`create()`, `delete()`, `query()`, `retrieve()`, `search()`, `update()`, `upsert()`

Supported REST HTTP Methods

`Query`, `DELETE`, `GET`, `PATCH`, `POST`

Fields

Field Name	Details
`EntityDefinition`	**Type** EntityDefinition **Properties** Filter, Group, Sort. **Description** Required. The entity definition for the object associated with the validation rule.
`EntityDefinitionId`	**Type** string **Properties** Filter, Group, Sort **Description** The ID of the entity containing the workflow task.
`FullName`	**Type** string

Field Name	Details
	Properties Create, Group, Nillable **Description** The full name of the associated metadata object in Metadata API. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`ManageableState`	**Type** ManageableState enumerated list **Properties** Create, Filter, Update **Description** Indicates the manageable state of the specified component that is contained in a package: • `beta` • `deleted` • `deprecated` • `installed` • `released` • `unmanaged`
`Metadata`	**Type** `mns`:WorkflowTask **Properties** Create, Nillable, Update **Description** Workflow task metadata. Query this field only if the query result contains no more than one record. Otherwise, an error is returned. If more than one record exists, use multiple queries to retrieve the records. This limit protects performance.
`NamespacePrefix`	**Type** string **Properties** Filter, Group, Nillable, Sort **Description** The namespace of the package containing the workflow task object.
`Priority`	**Type** picklist

Field Name	Details
	Properties Filter, Group, Sort **Description** The task’s priority. Values are: • High • Normal • Low
`Status`	**Type** picklist **Properties** Filter, Group, Sort **Description** The task’s status. Values are: • Not Started • In Progress • Completed • Waiting on someone else • Deferred
`Subject`	**Type** string **Properties** Filter, Group, idLookup, Sort **Description** A subject for the workflow task. It is used if an email notification is sent when the task is assigned.

CHAPTER 3 SOAP Headers for Tooling API

In this chapter ...

Tooling API provides SOAP headers similar to the SOAP API headers.

Control the behavior of SOAP calls:

AllOrNoneHeader

Allows a call to roll back all changes unless all records are processed successfully.

Without the AllOrNoneHeader header, records without errors are committed, while records with errors are marked as failed in the call results. This header is available in API version 20.0 and later.

Even if the header is enabled, it's still necessary to inspect the `success` field in the call result for each record to identify records with errors. Each `success` field contains `true` or `false` indicating whether the call was processed successfully.

If there is an error associated with at least one record, the `errors` field in the call result for the record gives more information on the error. If other records in the same call have no errors, their `errors` fields indicate that they were rolled back due to other errors.

API Calls

`create()`, `delete()`, `undelete()`, `update()`, `upsert()`

Fields

Element Name	Type	Description
`allOrNone`	boolean	If `true`, any failed records in a call cause all changes for the call to be rolled back. Record changes aren't committed unless all records are processed successfully. The default is `false`. Some records can be processed successfully while others are marked as failed in the call results.

Sample Code—Java

This sample shows how to use the `AllOrNoneHeader`. It attempts to create two contacts. The second contact doesn’t have all required fields set and causes a failure on creation. Next, the sample sets the `allOrNone` field to `true`, and then attempts to create the contacts. Creating one of the contacts results in an error, so the entire transaction is rolled back and no contacts are created.

```
public void allOrNoneHeaderSample() {
  try {
    // Create the first contact.
    SObject[] sObjects = new SObject[2];
    Contact contact1 = new Contact();
    contact1.setFirstName("Robin");
    contact1.setLastName("Van Persie");

    // Create the second contact. This contact doesn't
    // have a value for the required
    // LastName field so the create will fail.
    Contact contact2 = new Contact();
    contact2.setFirstName("Ashley");
    sObjects[0] = contact1;
    sObjects[1] = contact2;

    // Set the SOAP header to roll back the create unless
```

```
      // all contacts are successfully created.
      connection.setAllOrNoneHeader(true);
      // Attempt to create the two contacts.
      SaveResult[] sr = connection.create(sObjects);
      for (int i = 0; i < sr.length; i++) {
        if (sr[i].isSuccess()) {
          System.out.println("Successfully created contact with id: " +
            sr[i].getId() + ".");
        }
        else {
          // Note the error messages as the operation was rolled back
          // due to the all or none header.
          System.out.println("Error creating contact: " +
            sr[i].getErrors()[0].getMessage());
          System.out.println("Error status code: " +
            sr[i].getErrors()[0].getStatusCode());
        }
      }
    } catch (ConnectionException ce) {
      ce.printStackTrace();
    }
  }
}
```

AllowFieldTruncationHeader

Specifies that for some fields, when a string is too large, the operation fails. Without the header, strings for these fields are truncated.

The `AllowFieldTruncationHeader` header affects the following datatypes:

- anyType, if it represents one of the other datatypes in this list
- email
- encryptedstring
- multipicklist
- phone
- picklist
- string
- textarea

In API versions previous to 15.0, if a value for one of the listed fields is too large, the value is truncated.

For API version 15.0 and later, if a value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned. `AllowFieldTruncationHeader` allows you to specify that the previous behavior, truncation, be used instead of the new behavior in API versions 15.0 and later.

This header has no effect in versions 14.0 and earlier.

API Calls

`convertLead()`, `create()`, `merge()`, `process()`, `undelete()`, `update()`, and `upsert()`

Apex: `executeanonymous()`

Fields

Element Name	Type	Description
`allowFieldTruncation`	boolean	If `true`, truncate field values that are too long, which is the behavior in API versions 14.0 and earlier. Default is `false`: no change in behavior. If a `string` or `textarea` value is too large, the operation fails and the fault code `STRING_TOO_LONG` is returned. The following list shows the field types affected by truncation and this header: • anyType, if it represents one of the other datatypes in this list • email • encryptedstring • multipicklist • phone • picklist • string • textarea

Sample Code—Java

To create an account with a name that is too long for the `Name` field, use the AllowFieldTruncation header.

This sample:

1. Creates an Account object with a name that exceeds the field limit of 255 characters.
2. Sends the create call, which fails because of the name field length.
3. Sets the `AllowFieldTruncationHeader` to `true` and retries the account creation, which succeeds.

```
public void allowFieldTruncationSample() {
  try {
    Account account = new Account();
    // Construct a string that is 256 characters long.
    // Account.Name's limit is 255 characters.
    String accName = "";
    for (int i = 0; i < 256; i++) {
      accName += "a";
    }
    account.setName(accName);
    // Construct an array of SObjects to hold the accounts.
    SObject[] sObjects = new SObject[1];
    sObjects[0] = account;
    // Attempt to create the account. It will fail in API version 15.0
    // and above because the account name is too long.
    SaveResult[] results = connection.create(sObjects);
    System.out.println("The call failed because: "
      + results[0].getErrors()[0].getMessage());
    // Now set the SOAP header to allow field truncation.
    connection.setAllowFieldTruncationHeader(true);
    // Attempt to create the account now.
    results = connection.create(sObjects);
    System.out.println("The call: " + results[0].isSuccess());
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

CallOptions

Specifies the API client identifier.

Version

This call is available in all API versions.

Supported Calls

All Metadata API calls.

Fields

Field Name	Type	Description
`client`	string	A value that identifies an API client.

Sample Code—Java

To change the API client ID, add the `CallOptions` header to the metadata connection before you perform a call as follows:

```
metadataConnection.setCallOptions("client ID");
```

DebuggingHeader

Return the debug log in the output header, `DebuggingInfo`, and specify the level of detail in the debug log.

API Calls

`compileAndTest()executeanonymous()runTests()`

Fields

Element Name	Type	Description
debugLevel	logtype	Note: Don't use this field, because it's been deprecated and is only provided for backwards compatibility with older versions. Specifies the type of information returned in the debug log. The values are listed from the least amount of information returned to the most information returned. Valid values: • NONE • DEBUGONLY • DB • PROFILING • CALLOUT • DETAIL
categories	LogInfo[]	Specifies the type and amount of information to be returned in the debug log.

LogInfo

Specifies the type and amount of information to be returned in the debug log. The categories field takes a list of these objects.

Fields

Element Name	Type	Description
LogCategory	string	Specify the type of information returned in the debug log. Valid values are: • Db • Workflow • Validation • Callout • Apex_code • Apex_profiling • All
LogCategoryLevel	string	Specifies the level of detail returned in the debug log. Only the Apex_code LogCategory uses the log category levels. Valid log levels are (listed from lowest to highest): • NONE • ERROR • WARN

Element Name	Type	Description
		• INFO • DEBUG • FINE • FINER • FINEST

DisableFeedTrackingHeader

Specifies that changes made in the current call are tracked in feeds.

Use this header if you want to process many records without tracking the changes in various feeds related to the records. This header is available if the Chatter feature is enabled for your organization.

API Calls

`convertLead()`, `create()`, `delete()`, `merge()`, `process()`, `undelete()`, `update()`, `upsert()`

Fields

Element Name	Type	Description
`disableFeedTracking`	boolean	If `true`, the changes made in the current call are not tracked in feeds. The default is `false`.

Sample Code—Java

This sample shows how to use the `DisableFeedTrackingHeader`. It sets this header to `true` to disable feed tracking and then creates many account records in bulk.

```
public void disableFeedTrackingHeaderSample() {
  try {
    // Insert a large number of accounts.
    SObject[] sObjects = new SObject[500];
    for (int i = 0; i < 500; i++)  {
       Account a = new Account();
       a.setName("my-account-" + i);
       sObjects[i] = a;
    }
    // Set the SOAP header to disable feed tracking to avoid generating a
    // large number of feed items because of this bulk operation.
    connection.setDisableFeedTrackingHeader(true);
    // Perform the bulk create. This won't result in 500 feed items, which
    // would otherwise be generated without the DisableFeedTrackingHeader.
    SaveResult[] sr = connection.create(sObjects);
    for (int i = 0; i < sr.length; i++) {
      if (sr[i].isSuccess()) {
        System.out.println("Successfully created account with id: " +
          sr[i].getId() + ".");
      } else {
        System.out.println("Error creating account: " +
          sr[i].getErrors()[0].getMessage());
      }
    }
  } catch (ConnectionException ce) {
    ce.printStackTrace();
  }
}
```

MetadataWarningsHeader

Allows you to save metadata even if warnings are returned.

Version

This header is available in API version 35.0 and later.

Supported Calls

`delete()`, `update()`, `upsert()`

Field

Field Name	Type	Description
`ignoreSaveWarnings`	boolean	If `true`, you can save metadata such as a flow even if there are warnings, but not if there are errors.

PackageVersionHeader

Specifies the package version for each installed managed package.

A managed package can have several versions with different content and behavior. This header allows you to specify the version used for each package referenced by your API client.

If a package version is not specified, the API client uses the version of the package specified in Setup (enter *`API`* in the `Quick Find` box, then select **API**).

This header is available in API version 16.0 and later.

Associated API Calls

convertLead(), create(), delete(), describeGlobal(), describeLayout(), describeSObject(), describeSObjects(), describeSoftphoneLayout(), describeTabs(), merge(), process(), query(), retrieve(), search(), undelete(), update(), upsert()

Fields

Element Name	Type	Description
packageVersions	PackageVersion[]	A list of package versions for installed managed packages referenced by your API client.

PackageVersion

Specifies a version of an installed managed package. A package version is *majorNumber.minorNumber*, for example 2.1.

Fields

Field	Type	Description
majorNumber	int	The major version number of a package version.
minorNumber	int	The minor version number of a package version.
namespace	string	The unique namespace of the managed package.

Sample Code—Java

This sample sets the package version for one installed package in the PackageVersionHeader. Next, it executes the code passed into this method via the executeAnonymous Apex method.

```
public void PackageVersionHeaderSample(String code) throws Exception
{
    _PackageVersionHeader pvh = new _PackageVersionHeader();
    PackageVersion pv = new PackageVersion();
    pv.setNamespace("installedPackageNamespaceHere");
    pv.setMajorNumber(1);
    pv.setMinorNumber(0);
    // In this case, we are only referencing one installed package.
    PackageVersion[] pvs = new PackageVersion[]{pv};
    pvh.setPackageVersions(pvs);

    apexBinding.setHeader(new SforceServiceLocator().getServiceName().getNamespaceURI(),
        "PackageVersionHeader", pvh);
    // Execute the code passed into the method.
    ExecuteAnonymousResult r = apexBinding.executeAnonymous(code);
    if (r.isSuccess()) {
        System.out.println("Code executed successfully");
    }
    else {
        System.out.println("Exception message: " + r.getExceptionMessage());
        System.out.println("Exception stack trace: " + r.getExceptionStackTrace());
    }
}
```

SessionHeader

Specifies the session ID returned from the login server after a successful `login()`. This session ID is used in all subsequent calls.

In version 12.0 and later, include the API namespace in the SOAP message associated with this header. The namespace is defined in the enterprise or partner WSDL.

API Calls

All calls, including utility calls.

Fields

Element Name	Type	Description
`sessionId`	string	Session ID returned by the `login()` call to be used for subsequent call authentication.

Sample Code

See the examples provided for `login()`.

CHAPTER 4 REST Headers for Tooling API

In this chapter ...

Tooling API provides a subset of the REST headers available in REST API.

Control the behavior of REST requests:

Call Options Header

Specifies the client-specific options when accessing REST API resources. For example, you can write client code that ignores namespace prefixes by specifying the prefix in the call options header.

The Call Options header can be used with SObject Basic Information, SObject Rows, Query, QueryAll, Search, and SObject Rows by External ID.

Header Field Name and Values

Field name

`Sforce-Call-Options`

Field values

- `client`—A string that identifies a client.
- `defaultNamespace`—A string that identifies a developer namespace prefix. Resolve field names in managed packages without having to specify the namespace everywhere.

Example

If the developer namespace prefix is `battle`, and you have a custom field called `botId` in a package, set the default namespace with the call options header:

```
Sforce-Call-Options: client=SampleCaseSensitiveToken/100, defaultNamespace=battle
```

Then queries such as the following succeed:

```
/vXX.X/query/?q=SELECT+Id+botID__c+FROM+Account
```

In this case the actual field queried is the `battle__botId__c` field.

Using this header allows you to write client code without having to specify the namespace prefix. In the previous example, without the header you must write `battle__botId__c`.

If this field is set, and the query also specifies the namespace, the response doesn't include the prefix. For example, if you set this header to `battle`, and issue a query like `SELECT+Id+battle__botID__c+FROM+Account`, the response uses a `botId__c` element, not a `battle_botId__c` element.

The `defaultNamespace` field is ignored when retrieving describe information, which avoids ambiguity between namespace prefixes and customer fields of the same name.

Limit Info Header

This response header is returned in each request to the REST API. You can use the information to monitor API limits.

Header Field Name and Values

Field name

`Sforce-Limit-Info`

Field values

- `api-usage`—Specifies the API usage for the organization against which the call was made in the format `nn/nnnn`. The first number is the number of API calls used, and the second number is the API limit for the organization.

- `per-app-api-usage`—Specifies the limit quota information for the currently connected app. API limit app quotas are currently available through a pilot program. For information on enabling this feature for your organization, contact Salesforce. This example includes the limit quota for a `sample-connected-app` connected app. If there is no limit quota information, this field isn't returned.

  ```
  Sforce-Limit-Info: api-usage=25/5000;
  per-app-api-usage=17/250(appName=sample-connected-app)
  ```

Example

Response to a REST request for a Merchandise record, including the limit information in line three:

```
HTTP/1.1 200 OK
Date: Mon, 20 May 2013 22:21:46 GMT
Sforce-Limit-Info: api-usage=18/5000
Last-Modified: Mon, 20 May 2013 20:49:32 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked

{
  "attributes" : {
    "type" : "Merchandise__c",
    "url" : "/services/data/v36.0/sobjects/Merchandise__c/a00D0000008pQSNIA2"
  },
  "Id" : "a00D0000008pQSNIA2",
  "OwnerId" : "005D0000001QX8WIAW",
  "IsDeleted" : false,
  "Name" : "Phone Case - iPhone 4/4S",
  "CreatedDate" : "2013-05-20T20:49:32.000+0000",
  "CreatedById" : "005D0000001QX8WIAW",
  "LastModifiedDate" : "2013-05-20T20:49:32.000+0000",
  "LastModifiedById" : "005D0000001QX8WIAW",
  "SystemModstamp" : "2013-05-20T20:49:32.000+0000",
  "LastActivityDate" : null,
  "LastViewedDate" : "2013-05-20T22:19:56.000+0000",
  "LastReferencedDate" : "2013-05-20T22:19:56.000+0000",
  "Description__c" : "Phone Case for iPhone 4/4S",
  "Price__c" : 16.99,
  "Stock_Price__c" : 12.99,
  "Total_Inventory__c" : 108.0
}
```

Package Version Header

Specifies the version of each package referenced by a client. A package version is a number that identifies the set of components and behavior contained in a package. This header can also be used to specify a package version when making calls to an Apex REST web service.

The Package Version header can be used with the following resources: Describe Global, SObject Describe, SObject Basic Information, SObject Rows, Describe Layouts, Query, QueryAll, Search, and SObject Rows by External ID.

Header Field Name and Values

Field name and value

`x-sfdc-packageversion-[namespace]: xx.x`, where `[namespace]` is the unique namespace of the managed package and `xx.x` is the package version.

Example

```
x-sfdc-packageversion-clientPackage: 1.0
```

Query Options Header

Specifies options used in a query, such as the query results batch size. Use this request header with the Query resource.

Header Field Name and Values

Field name

```
Sforce-Query-Options
```

Field values

- `batchSize`—A numeric value that specifies the number of records returned for a query request. Child objects count toward the number of records for the batch size. For example, in relationship queries, multiple child objects are returned per parent row returned.

  The default is 2,000; the minimum is 200, and the maximum is 2,000. There is no guarantee that the requested batch size is the actual batch size. Changes are made as necessary to maximize performance.

Example

```
Sforce-Query-Options: batchSize=1000
```

INDEX