

# REST API Developer Guide

Version 47.0, Winter '20





# CONTENTS

Chapter 1: Introducing Lightning Platform REST API
Lightning Platform REST Resources
Using Compression
Using Conditional Requests
Using cURL in the REST Examples
Understanding Authentication
Defining Connected Apps
Understanding OAuth Endpoints
How Are Apps Authenticated with the Web Server OAuth Authentication Flow?
Understanding the User-Agent OAuth Authentication Flow
Understanding the Username-Password OAuth Authentication Flow
How Does the OAuth Refresh Token Fit Into the Authentication Flow?
Finding Additional Resources
Use CORS to Access Salesforce Resources from Web Browsers
API End-of-Life
Chapter 2: Quick Start
Prerequisites
Step One: Obtain a Salesforce Developer Edition Organization
Step Two: Set Up Authorization
Step Three: Send HTTP Requests with cURL
Step Four: Walk Through the Sample Code
Using Workbench
ŭ
<b>Chapter 3:</b> Examples
Getting Information About My Organization
List Available REST API Versions
List Organization Limits
List Available REST Resources
Get a List of Objects
Get an Image from a Rich Text Area Field
Get a List of Objects If Metadata Has Changed
Working with Object Metadata
Retrieve Metadata for an Object
Get Field and Other Metadata for an Object
Get Object Metadata Changes
Working with Records
Create a Record
Undate a Record

Delete a Record	47
Get Field Values from a Standard Object Record	47
Get Field Values from an External Object Record by Using the Salesforce ID	47
Get Field Values from an External Object Record by Using the External ID Standard	
Field	48
Retrieve a Record Using an External ID	48
Insert or Update (Upsert) a Record Using an External ID	49
Traverse Relationships with Friendly URLs	53
Get Attachment Content from a Record	58
Get a List of Deleted Records Within a Given Timeframe	58
Get a List of Updated Records Within a Given Timeframe	59
Delete Lightning Experience Event Series	60
Working with Searches and Queries	6
Execute a SOQL Query	62
Execute a SOQL Query that Includes Deleted Items	63
Get Feedback on Query Performance	64
Search for a String	66
Get the Default Search Scope and Order	68
Get Search Result Layouts for Objects	69
View Relevant Items	7
Insert or Update Blob Data	73
Working with Recently Viewed Information	79
View Recently Viewed Records	79
Mark Records as Recently Viewed	80
Managing User Passwords	80
Manage User Passwords	8
Working with Approval Processes and Process Rules	82
Get a List of All Approval Processes	83
Submit a Record for Approval	83
Approve a Record	84
Reject a Record	84
Bulk Approvals	85
Get a List of Process Rules	86
Get a Particular Process Rule	87
Trigger Process Rules	87
Using Event Monitoring	88
Describe Event Monitoring Using REST	89
Query Event Monitoring Data with REST	90
Get Event Monitoring Content from a Record	9
Download Large Event Log Files Using cURL with REST	9
Delete Event Monitoring Data	92
Query or View Hourly Event Log Files	93
Using Composite Resources	95
Execute Dependent Requests in a Single API Call	95

Update an Account, Create a Contact, and Link Them with a Junction Object	
Update a Record and Get Its Field Values in a Single Request	
Create Nested Records	
Create Multiple Records	. 102
Chapter 4: Reference	. 104
Versions	. 109
Resources by Version	. 109
Limits	. 109
Describe Global	. 112
SObject Basic Information	. 112
SObject Describe	
SObject Get Deleted	
SObject Get Updated	
SObject Named Layouts	
SObject Rows	
SObject Rows by External ID	
SObject Blob Retrieve	
SObject ApprovalLayouts	
SObject CompactLayouts	
Describe Layouts	
SObject PlatformAction	
SObject Quick Actions	
SObject Rich Text Image Retrieve	
SObject Relationships	
SObject Suggested Articles	
SObject User Password	
Platform Event Schema by Event Name	
Platform Event Schema by Schema ID	
AppMenu	
Compact Layouts	
Consent	
Embedded Service Configuration Describe	
Invocable Actions	
Standard Invocable Actions	
Custom Invocable Actions	
List View Describe	
List View Results	
List Views	
Support Knowledge with REST API	
Data Category Groups	
Data Category Detail	
Articles List	
Articles Details	

Parameterized Search	. 192
Process Approvals	200
Process Rules	202
Product Schedules	203
Query	204
QueryAll	207
Quick Actions	207
Recent List Views	208
Recently Viewed Items	209
Record Count	. 210
Record Count Response Body	. 211
Relevant Items	. 212
Retrieve Knowledge Language Settings	. 214
Search	. 215
Search Scope and Order	. 215
Search Result Layouts	
Lightning Toggle Metrics	
Lightning Usage by App Type	
Lightning Usage by Browser	
Lightning Usage by Page	
Lightning Usage by FlexiPage	
Lightning Exit by Page Metrics	
Lightning Scheduler Resources	
Scheduling	
Get Appointment Slots	
Get Appointment Candidates	
Request Bodies	
Response Bodies	
Search for Records Suggested by Autocomplete and Instant Results	
Search Suggested Article Title Matches	
Search Suggested Queries	
Tabs	
Themes	
Composite Resources	
Composite	
Batch	
SObject Tree	
SObject Collections	
Headers	
Assignment Rule Header	
Call Options Header	
Limit Info Header	
Package Version Header	
Query Options Header	
	A. A. E.

			077
Status Codes and Error Res	sponses	 	2//

## **CHAPTER 1** Introducing Lightning Platform REST API

### In this chapter ...

- Lightning Platform REST Resources
- Using Compression
- Using Conditional Requests
- Using cURL in the REST Examples
- Understanding Authentication
- Use CORS to Access Salesforce Resources from Web Browsers
- API End-of-Life

REST API provides a powerful, convenient, and simple Web services API for interacting with Lightning Platform. Its advantages include ease of integration and development, and it's an excellent choice of technology for use with mobile applications and Web 2.0 projects. If you have many records to process, consider using Bulk API, which is based on REST principles and optimized for large sets of data.

REST API uses the same underlying data model and standard objects as those in SOAP API. See the SOAP API Developer Guide for details. REST API also follows the same limits as SOAP API. See the Limits section in the SOAP API Developer Guide.

To use the API requires basic familiarity with software development, web services, and the Salesforce user interface.

Use this introduction to understand:

- The key characteristics and architecture of REST API. This will help you understand how your applications can best use the Lightning Platform REST resources.
- How to set up your development environment so you can begin working with REST API immediately.
- How to use REST API by following a quick start that leads you step by step through a typical use case.

## **Lightning Platform REST Resources**

A REST resource is an abstraction of a piece of information or an action, such as a single data record, a collection of records, or a query. Each resource in REST API is identified by a named Uniform Resource Identifier (URI) and is accessed using standard HTTP methods (HEAD, GET, POST, PATCH, DELETE). REST API is based on the usage of resources, their URIs, and the links between them.

You use a resource to interact with your Salesforce org. For example, you can:

- Retrieve summary information about the API versions available to you.
- Obtain detailed information about a Salesforce object, such as Account, User, or a custom object.
- Perform a query or search.
- Update or delete records.

Suppose you want to retrieve information about the Salesforce version. Submit a request for the Versions resource.

```
curl https://yourInstance.salesforce.com/services/data/
```

The output from this request is as follows.

```
[
    "version":"20.0",
    "url":"/services/data/v20.0",
    "label":"Winter '11"
}
...
]
```

Ø

**Note**: Salesforce runs on multiple server instances. The examples in this guide use *yourInstance* in place of a specific instance. Replace that text with the instance for your org.

Important characteristics of the Lightning Platform REST API resources and architecture:

### **Stateless**

Each request from client to server must contain all the information necessary to understand the request, and not use any stored context on the server. However, the representations of the resources are interconnected using URLs, which allow the client to progress between states.

### **Caching behavior**

Responses are labeled as cacheable or non-cacheable.

### **Uniform interface**

All resources are accessed with a generic interface over HTTP.

### Named resources

All resources are named using a base URI that follows your Lightning Platform URI.

### Layered components

The Lightning Platform REST API architecture allows for the existence of such intermediaries as proxy servers and gateways to exist between the client and the resources.

### Authentication

The Lightning Platform REST API supports OAuth 2.0 (an open protocol to allow secure API authorization). See Understanding Authentication for more details.

### Support for JSON and XML

JSON is the default. You can use the HTTP ACCEPT header to select either JSON or XML, or append .json or .xml to the URI (for example, /Account/001D000000INjVe.json).

The JavaScript Object Notation (JSON) format is supported with UTF-8. Date-time information is in ISO8601 format.

XML serialization is similar to SOAP API. XML requests are supported in UTF-8 and UTF-16, and XML responses are provided in UTF-8.

### Friendly URLs

Why make two API calls when you can make just one? A friendly URL provides an intuitive way to construct REST API requests and minimizes the number of round-trips between your app and Salesforce org. Friendly URLs are available in API version 36.0 and later.

Accessing a contact's parent account without a friendly URL involves requesting the contact record using the SObject Rows resource. Then you examine the account relationship field to obtain the account ID and request the account record with another call to SObject Rows. Using a friendly URL, you can access the account in a single call directly from the contact's path:

/services/data/v36.0/sobjects/contact/id/account.

This functionality is exposed via the SObject Relationships on page 130 resource. For more examples of using friendly URLs to access relationship fields, see Traverse Relationships with Friendly URLs on page 53.

## **Using Compression**

The REST API allows the use of compression on the request and the response, using the standards defined by the HTTP 1.1 specification. Compression is automatically supported by some clients, and can be manually added to others. Visit Salesforce Developers for more information on particular clients.



Tip: For better performance, we suggest that clients accept and support compression as defined by the HTTP 1.1 specification.

To use compression, include the HTTP header Accept-Encoding: gzip or Accept-Encoding: deflate in a request. The REST API compresses the response if the client properly specifies this header. The response includes the header Content-Encoding: gzip or Accept-Encoding: deflate. You can also compress any request by including a Content-Encoding: gzip or Content-Encoding: deflate header.

## Response Compression

The REST API can optionally compress responses. Responses are compressed only if the client sends an Accept-Encoding header. The REST API is not required to compress the response even if you have specified Accept-Encoding, but it normally does. If the REST API compresses the response, it also specifies a Content-Encoding header.

## **Request Compression**

Clients can also compress requests. The REST API decompresses any requests before processing. The client must send a Content-Encoding HTTP header in the request with the name of the appropriate compression algorithm. For more information, see:

- Content-Encoding at: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11
- Accept-Encoding at: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.3
- Content Codings at: www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.5

## **Using Conditional Requests**

To support response caching, REST API allows conditional request headers that follow the standards defined by the HTTP 1.1 specification.

For strong validation, include either the If-Match or If-None-Match header in a request, and reference the entity tags (ETag) of the records you want to match against. For weak validation, include either the If-Modified-Since or

If-Unmodified-Since header in a request along with the date and time you want to check against. The REST API conditional headers follow the HTTP 1.1 specification with the following exceptions.

- When you include an invalid header value for If-Match, If-None-Match, or If-Unmodified-Since on a PATCH or POST request, a 400 Bad Request status code is returned.
- The If-Range header isn't supported.
- DELETE requests are not supported with these headers.

#### **ETag**

```
HTTP 1.1 specification: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.19
```

The ETag header is a response header that's returned when you access the SObject Rows resource. It's a hash of the content that's used by the If-Match and If-None-Match request headers in subsequent requests to determine if the content has changed.

Supported resources: SObject Rows (account records only)

```
Example: ETag: "U5iWijwWbQD18jeiXwsqxeGpZQk=-gzip"
```

### If-Match

```
HTTP 1.1 specification: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.24
```

The If-Match header is a request header for SObject Rows that includes a list of ETags. If the ETag of the record you're requesting matches an ETag specified in the header, the request is processed. Otherwise, a 412 Precondition Failed status code is returned, and the request isn't processed.

Supported resources: SObject Rows (account records only)

```
Example: If-Match: "Jbjuzw7dbhaEG3fd90kJbx6A0ow=-gzip",
"U5iWijwWbQD18jeiXwsqxeGpZQk=-gzip"
```

### If-None-Match

```
HTTP 1.1 specification: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.26
```

The If-None-Match header is a request header for SObject Rows that's the inverse of If-Match. If the ETag of the record you're requesting matches an ETag specified in the header, the request isn't processed. A 304 Not Modified status code is returned for GET or HEAD requests, and a 412 Precondition Failed status code is returned for PATCH requests.

Supported resources: SObject Rows (account records only)

```
Example: If-None-Match: "Jbjuzw7dbhaEG3fd90kJbx6A0ow=-gzip",
"U5iWijwWbQD18jeiXwsqxeGpZQk=-gzip"
```

### If-Modified-Since

```
HTTP 1.1 specification: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.25
```

The If-Modified-Since header is a time-based request header. The request is processed only if the data has changed since the date and time specified in the header. Otherwise, a 304 Not Modified status code is returned, and the request isn't processed.

Supported resources: SObject Rows, SObject Describe, Describe Global, and Invocable Actions

```
Example: If-Modified-Since: Tue, 10 Aug 2015 00:00:00 GMT
```

### If-Unmodified-Since

```
HTTP 1.1 specification: www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.28
```

The If-Unmodified-Since header is a request header that's the inverse of If-Modified-Since. If you make a request and include the If-Unmodified-Since header, the request is processed only if the data hasn't changed since the specified date. Otherwise, a 412 Precondition Failed status code is returned, and the request isn't processed.

Supported resources: SObject Rows, SObject Describe, Describe Global, and Invocable Actions

Example: If-Unmodified-Since: Tue, 10 Aug 2015 00:00:00 GMT

## Using cURL in the REST Examples

The examples in this guide use the cURL tool to send HTTP requests to access, create, and manipulate REST resources on the Lightning Platform. cURL is pre-installed on many Linux and Mac systems. Windows users can download a version at curl.haxx.se/. When using HTTPS on Windows, ensure that your system meets the cURL requirements for SSL.



Note: cURL is an open source tool and is not supported by Salesforce.

### Escaping the Session ID or Using Single Quotes on Mac and Linux Systems

When running the cURL examples for the REST resources, you may get an error on Mac and Linux systems due to the presence of the exclamation mark special character in the session ID argument. To avoid getting this error, do one of the following:

• Escape the exclamation mark (!) special character in the session ID by inserting a backslash before it (\!) when the session ID is enclosed within double quotes. For example, the session ID string in this cURL command has the exclamation mark (!) escaped:

```
curl https://instance_name.salesforce.com/services/data/v47.0/
-H "Authorization: Bearer
00D500000001ehZ\!AQcAQH0dMHZfz972Szmpkb58urFRkgeBGsxL_QJWwYMfAbUeeG7c1E6
LYUfiDUkWe6H34r1AAwOR8B8fLEz6n04NPGRrq0FM"
```

• Enclose the session ID within single quotes. For example:

```
curl https://instance_name.salesforce.com/services/data/v47.0/
-H 'Authorization: Bearer sessionID'
```

## **Understanding Authentication**

Salesforce uses the OAuth protocol to allow users of applications to securely access data without having to reveal username and password credentials.

Before making REST API calls, you must authenticate the application user using OAuth 2.0. To do so, you'll need to:

- Set up your application as a connected app in the Salesforce organization.
- Determine the correct Salesforce OAuth endpoint for your connected app to use.
- Authenticate the connected app user via one of several different OAuth 2.0 authentication flows. An OAuth authentication flow defines a series of steps used to coordinate the authentication process between your application and Salesforce. Supported OAuth flows include:
  - Web server flow, where the server can securely protect the consumer secret.
  - User-agent flow, used by applications that cannot securely store the consumer secret.
  - Username-password flow, where the application has direct access to user credentials.

After successfully authenticating the connected app user with Salesforce, you'll receive an access token which can be used to make authenticated REST API calls.

## **Defining Connected Apps**

To authenticate using OAuth, you must create a connected app that defines your application's OAuth settings for the Salesforce organization.

When you develop an external application that needs to authenticate with Salesforce, you need to define it as a new connected app within the Salesforce organization that informs Salesforce of this new authentication entry point.

Use the following steps to create a new connected app.

- 1. Use Salesforce to create a new connected app.
  - In Lightning Experience, use the App Manager to create connected apps. From Setup, enter App in the Quick Find box, then select **App Manager**. Click **New Connected App**.
  - In Salesforce Classic, from Setup, enter Apps in the Quick Find box, then select Apps (under Build > Create). Under Connected Apps, click New.
- 2. Enter the name of your application.
- 3. Enter the contact email information, as well as any other information appropriate for your application.
- 4. Select Enable OAuth Settings.
- 5. Enter a Callback URL. Depending on which OAuth flow you use, this is typically the URL that a user's browser is redirected to after successful authentication. As this URL is used for some OAuth flows to pass an access token, the URL must use secure HTTP (HTTPS) or a custom URI scheme. Separate multiple callback URLs with line breaks.
  - The callback URL field has a limit of 2000 characters, cumulatively. If you enter several URLs and they exceed this limit, configure another connected app to manage more callback URLs.
- **6.** Add all supported OAuth scopes to **Selected OAuth Scopes**. These scopes refer to permissions given by the user running the connected app.
- 7. Enter a URL for Info URL. This is where the user can go for more information about your application.
- 8. Click Save. The Consumer Key is created and displayed, and the Consumer Secret is created (click the link to reveal it).

Once you define a connected app, you use the consumer key and consumer secret to authenticate your application. See Creating a Connected App in the Salesforce online help for specific steps to create a connected app for the type of authentication you need.

## **Understanding OAuth Endpoints**

OAuth endpoints are the URLs you use to make OAuth authentication requests to Salesforce.

You need to use the correct Salesforce OAuth endpoint when issuing authentication requests in your application. The primary OAuth endpoints are:

- For authorization: https://login.salesforce.com/services/oauth2/authorize
- For token requests: https://login.salesforce.com/services/oauth2/token
- For revoking OAuth tokens: https://login.salesforce.com/services/oauth2/revoke

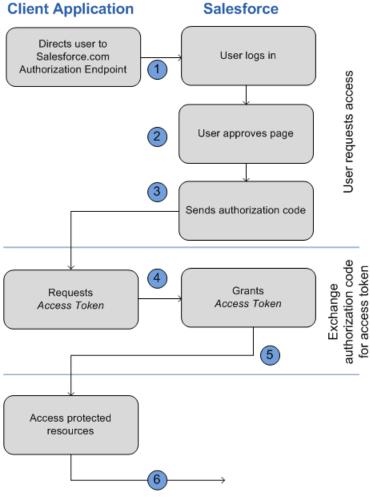
All endpoints require secure HTTP (HTTPS). Each OAuth flow defines which endpoints you need to use and what request data you need to provide.

If you're verifying authentication on a sandbox organization, use "test.salesforce.com" instead of "login.salesforce.com" in all the OAuth endpoints listed above.

# How Are Apps Authenticated with the Web Server OAuth Authentication Flow?

Apps that are hosted on a secure server use the web server authentication flow. A critical aspect of the web server flow is that the server must be able to protect the consumer secret. You can use code challenges and verifier values in the flow to prevent authorization code interception.

In this flow, the client application requests the authorization server to redirect the user to another web server or resource. The web server or resource authorizes the user, and sends the application an authorization code. The application uses the authorization code to request an access token. The following shows the steps for this flow.



1. The application redirects the user to the appropriate Salesforce authorization endpoint, such as https://login.salesforce.com/services/oauth2/authorize. The following parameters are required:

Parameter	Description
response_type	Must be code for this authentication flow.

Parameter	Description
client_id	The Consumer Key from the connected app definition.
redirect_uri	The Callback URL from the connected app definition.

The following parameters are optional:

Parameter	Description
code_challenge	Specifies the SHA256 hash value of the code_verifier value in the token request to help prevent authorization code interception attacks. The hash value must be base64url encoded as defined here:
	https://tools.ietf.org/html/rfc4648#section-5.
	<ul> <li>If the code_challenge value is provided in the authorization request and a code_verifier value is provided in the token request, Salesforce compares the code_challenge to the code_verifier. If the code_challenge is invalid or doesn't match, the login fails with the invalid_request error code.</li> </ul>
	<ul> <li>If the code_challenge value is provided in the authorization request, but a code_verifier value is not provided in the token request, the login fails with the invalid_grant error code.</li> </ul>
	Note: Encode this value with a base64urlonly once.
display	Changes the login page's display type. Valid values are:
	<ul> <li>page—Full-page authorization screen. This value is the default if none is specified.</li> </ul>
	<ul> <li>popup—Compact dialog optimized for modern web browser popup windows.</li> </ul>
	<ul> <li>touch—Mobile-optimized dialog designed for modern mobile devices such as Android and iPhone.</li> </ul>
	<ul> <li>mobile—Mobile optimized dialog designed for mobile devices such as BlackBerry OS 5 that don't support touch screens.</li> </ul>
immediate	Determines whether the user is prompted for login and approval.  Values are either true or false. Default is false.
	<ul> <li>If set to true, and if the user is currently logged in and has previously approved the application, the approval step is skipped.</li> </ul>
	<ul> <li>If set to true and the user is not logged in or has not previously approved the application, the session is</li> </ul>

Parameter	Description
	immediately terminated with the immediate_unsuccessful error code.
login_hint	Provides a valid username value to pre-populate the login page with the username. For example:login_hint=username@company.com. If a user already has an active session in the browser, then the login_hint parameter does nothing; the active user session continues.
nonce	Specifies a value to be returned in the response; this parameter is useful for detecting "replay" attacks. Optional with the openid scope for getting a user ID token.
prompt	Specifies how the authorization server prompts the user for reauthentication and reapproval. This parameter is optional. The only values Salesforce supports are:
	<ul> <li>login—The authorization server must prompt the user for reauthentication, forcing the user to log in again.</li> </ul>
	<ul> <li>consent—The authorization server must prompt the user for reapproval before returning information to the client.</li> </ul>
	It is valid to pass both values, separated by a space, to require the user to both log in and reauthorize. For example:
	?prompt=login%20consent
scope	Specifies what data your application can access. See "Scope Parameter Values" in Salesforce Help for more information.
state	Specifies any additional URL-encoded state data to be returned in the callback URL after approval.

This example shows an authorization URL:

https://login.salesforce.com/services/oauth2/authorize?response\_type=code &client\_id=3MVG9lKcPoNINVBIPJjdw1J9LLM82HnFVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCscA9GE&redirect\_uri=https%3A%2F%2Fwww.mysite.com%2Fcode\_callback.jsp&state=mystate

- 2. The user logs into Salesforce with their credentials. The user is interacting with the authorization endpoint directly, so the application never sees the user's credentials. After successfully logging in, the user is asked to authorize the application. If the user has already authorized the application, this step is skipped.
- 3. After Salesforce confirms that the client application is authorized, the end user's web browser is redirected to the callback URL specified by the redirect\_uri parameter. Salesforce appends authorization information to the redirect URL with the following values:

Parameters	Description
code	Authorization code the consumer must use to obtain the access and refresh tokens. The authorization code expires after 15 minutes.
state	The state value that was passed in as part of the initial request, if applicable.

This example shows a callback URL with authorization information:

https://www.mysite.com/authcode\_callback?code=aWekysIEeqM9PiT hEfm0Cnr6MoLIfwWyRJcqOqHdF8f9INokharASO9ia7UNP6RiVScerfhc4w%3D%3D

**4.** The application extracts the authorization code and passes it in a request to Salesforce for an access token. This request is a POST request sent to the appropriate Salesforce token request endpoint, such as <a href="https://login.salesforce.com/services/oauth2/token">https://login.salesforce.com/services/oauth2/token</a>. The following parameters are required:

Parameter	Description
grant_type	Value must be authorization_code for this flow.
client_secret	The Consumer Secret from the connected app definition. Required unless the Require Secret for Web Server Flow setting is not enabled in the connected app definition. If a client_secret isn't required, and the connected app sends it in the authorization request, Salesforce attempts to validate it, anyway.
client_id	The Consumer Key from the connected app definition.
redirect_uri	The Callback URL from the connected app definition.
code	Authorization code the consumer must use to obtain the access and refresh tokens. The authorization code expires after 15 minutes.

The following parameters are optional:

Parameter	Description
client_assertion	Instead of passing in client_secret you can choose to provide a client_assertion and client_assertion_type. If a client_secret parameter is not provided, Salesforce checks for the client_assertion and client_assertion_type automatically. The value of client_assertion must be a typical JWT bearer token, signed with the private key associated with the OAuth consumer's uploaded certificate. Only the RS256 algorithm is supported. For more information on using

Parameter	Description
	<pre>client_assertion, see the OpenID Connect specifications for the private_key_jwt client authentication method.</pre>
client_assertion_type	Provide this value when using the client_assertion parameter. The value of client_assertion_type must be urn:ietf:params:oauth:client-assertion-type:jwt-bearer.
code_verifier	Specifies 128 bytes of random data with high enough entropy to make it difficult to guess the value to help prevent authorization code interception attacks. The value also must be base64url encoded as defined here:  https://tools.ietf.org/html/rfc4648#section-5.
	<ul> <li>If the code_verifier value is provided in the token request and a code_challenge value is in the authorization request, Salesforce compares the code_verifier to the code_challenge. If the code_verifier is invalid or doesn't match, the login fails with the invalid_grant error code.</li> <li>If the code_verifier value is provided in the token request, but a code_challenge value was not provided</li> </ul>
	in the authorization request, the login fails with the invalid_grant error code.
	Note: Encode the value with a base64url only once.
format	Expected return format. The default is json. Values are:
	<ul><li>urlencoded</li></ul>
	• json
	• xml
	The return format can also be specified in the header of the request using one of the following:
	• Accept:
	application/x-www-form-urlencoded
	• Accept: application/json
	<ul> <li>Accept: application/xml</li> </ul>

This example shows an access token POST request that sends the client id and client secret in the body.

POST /services/oauth2/token HTTP/1.1

Host: login.salesforce.com

grant\_type=authorization\_code&code=aPrxsmIEeqM9PiQroGEWx1UiMQd95\_5JUZ
VEhsOFhS8EVvbfYBBJli2W5fn3zbo.8hojaNW\_1g%3D%3D&client\_id=3MVG91KcPoNI
NVBIPJjdw1J9LLM82HnFVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCs
cA9GE&client\_secret=1955279925675241571&

redirect\_uri=https%3A%2F%2Fwww.mysite.com%2Fcode\_callback.jsp

Instead of sending client credentials as parameters in the body of the POST request, Salesforce supports the HTTP Basic authentication scheme. The scheme requires the client id and client secret in the authentication header of the request as follows:

Authorization: Basic64Encode(client\_id:secret)

The client\_id and client\_secret are separated with a colon (:). For more information, see The OAuth 2.0 Authorization Framework.

This example shows an access token POST request that uses HTTP Basic authentication scheme (rather than sending client credentials in the POST request's body).

POST /services/oath2/token HTTP/1.1

Host: login.salesforce.com

Authorization: Basic client\_id=3MVG9lKcPoNINVBIPJjdw1J9LLM82HnFVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCscA9GE&client secret=1955279925675241571

grant\_type=authorization\_code&code=aPrxsmIEeqM9PiQroGEWx1UiMQd95\_5JUZ
VEhsOFhS8EVvbfYBBJli2W5fn3zbo.8hojaNW\_1g%3D%3D&
redirect uri=https%3A%2F%2Fwww.mysite.com%2Fcode callback.jsp

 ${}_{\hspace{-0.1cm}\text{\tiny MOTE}}$  Note: If the <code>client\_id</code> and <code>client\_secret</code> are sent in the POST's body, the authentication header is ignored.

5. If this request is successful, the server returns a response body that contains the following:

Parameters	Description
access_token	Access token that acts as a session ID that the application uses for making requests. This token should be protected as though it were user credentials.
token_type	Value is Bearer for all responses that include an access token.
refresh_token	Token that can be used in the future to obtain new access tokens.
	Warning: This value is a secret. Treat it like the user's password and use appropriate measures to protect it.
instance_url	Identifies the Salesforce instance to which API calls are sent.
id	Identity URL that can be used to both identify the user and query for more information about the user. Can be used in an HTTP request to get more information about the end user.
issued_at	When the signature was created, represented as the number of seconds since the Unix epoch (00:00:00 UTC on 1 January 1970).
signature	Base64-encoded HMAC-SHA256 signature signed with the client_secret (private key) containing the concatenated ID and issued_at value. Use the signature to verify that the identity URL wasn't modified when the server sent it.

This example shows a JSON response body:

{"id":"https://login.salesforce.com/id/00Dx000000BV7z/005x00000012Q9P",
"issued\_at":"1278448101416",

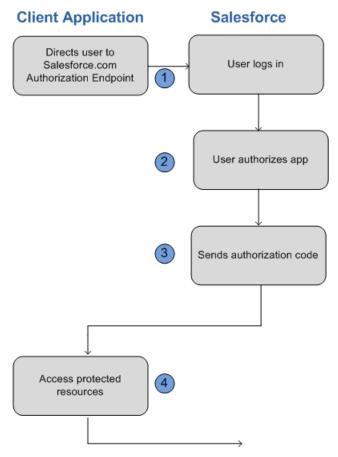
```
"refresh_token":"5Aep8614iLM.Dq661ePDmPEgaAW90h_L3JKkDpB4xReb54_
pZebnUG0h6Sb4KUVDpNtWEofWM39yg==",
"instance_url":"https://yourInstance.salesforce.com/",
"signature":"CMJ41+CCaPQiKjoOEwEig9H4wqhpuLSk4J2urAe+fVg=",
"access_token":"00Dx000000BV7z!AR8AQP0jITN80ESEsj5EbaZTFG0R
NBaT1cyWk7TrqoDjoNIWQ2ME_sTZzBjfmOE6zMHq6y8PIW4eWze9JksNEkWUl.Cju7m4"}
```

**6.** The application uses the provided access token and refresh token to access protected user data.

## Understanding the User-Agent OAuth Authentication Flow

The user-agent authentication flow is used by client apps (consumers) that reside on the user's device or computer. It's also used by client apps running in a browser using a scripting language such as JavaScript. These apps can protect per-user secrets. But, because the apps are widely distributed, the client secret can't be confidential.

In this flow, the client application requests the authorization server to redirect the user to another Web server or resource which is capable of extracting the access token and passing it back to the application. The following shows the steps for this flow.



1. The application redirects the user to the appropriate Salesforce authorization endpoint, such as https://login.salesforce.com/services/oauth2/authorize.The following parameters are required:

Parameter	Description
response_type	Must be token for this authentication flow

Parameter	Description
client_id	The Consumer Key from the connected app definition.
redirect_uri	The Callback URL from the connected app definition.

The following parameters are optional:

Parameter	Description
display	Changes the login page's display type. Valid values are:
	<ul> <li>page—Full-page authorization screen. This value is the default if none is specified.</li> </ul>
	<ul> <li>popup—Compact dialog optimized for modern web browser popup windows.</li> </ul>
	<ul> <li>touch—Mobile-optimized dialog designed for modern mobile devices such as Android and iPhone.</li> </ul>
	<ul> <li>mobile—Mobile optimized dialog designed for mobile devices such as BlackBerry OS 5 that don't support touch screens.</li> </ul>
scope	Specifies what data your application can access. See "Scope Parameter Values" in the online help for more information.
state	Specifies any additional URL-encoded state data to be returned in the callback URL after approval.

An example authorization URL might look something like the following:

https://login.salesforce.com/services/oauth2/authorize?response\_type=token&client\_id=3MVG9lKcPoNINVBIPJjdw1J9LLJbP\_pqwoJYyuisjQhr\_LLurNDv7AgQvDTZwCoZuDZrXcPCmBv4o.8ds.5iE&redirect\_uri=https%3A%2F%2Fwww.mysite.com%2Fuser\_callback.jsp&state=mystate

- 2. The user logs into Salesforce with their credentials. The user interacts with the authorization endpoint directly, so the application never sees the user's credentials.
- **3.** Once authorization is granted, the authorization endpoint redirects the user to the redirect URL. This URL is defined in the remote access application created for the application. Salesforce appends access token information to the redirect URL with the following values:

Parameters	Description
access_token	Access token that acts as a session ID that the application uses for making requests. This token should be protected as though it were user credentials.
token_type	Value is Bearer for all responses that include an access token.

Parameters	Description
refresh_token	Token that can be used in the future to obtain new access tokens.
	Warning: This value is a secret. Treat it like the user's password and use appropriate measures to protect it.
	The refresh token is only returned if the redirect URI is https://login.salesforce.com/services/oauth2/success or used with a custom protocol that is not HTTPS.
scope	A space-separated list of scope values.
state	The state value that was passed in as part of the initial request, if applicable.
instance_url	Identifies the Salesforce instance to which API calls are sent.
id	Identity URL that can be used to both identify the user and query for more information about the user. Can be used in an HTTP request to get more information about the end user.
issued_at	When the signature was created, represented as the number of seconds since the Unix epoch (00:00:00 UTC on 1 January 1970).
signature	Base64-encoded HMAC-SHA256 signature signed with the client_secret (private key) containing the concatenated ID and issued_at value. Use the signature to verify that the identity URL wasn't modified when the server sent it.

An example callback URL with access information appended after the hash sign (#) might look something like:

https://www.mysite.com/user\_callback.jsp#access\_token=00Dx0000000BV7z%21AR8
AQBM8J\_xr9kLqmZIRyQxZgLcM4HVi41aGtW0qW3JCzf5xdTGGGSoVim8FfJkZEqxbjaFbberKGk
8v8AnYrvChG4qJbQo8&refresh\_token=5Aep8614iLM.Dq661ePDmPEgaAW9Oh\_L3JKkDpB4xR
eb54\_pZfVtildPEk8aimw4Hr9ne7VXXVSIQ%3D%3D&expires\_in=7200&state=mystate

**4.** The application uses the provided access token and refresh token to access protected user data.

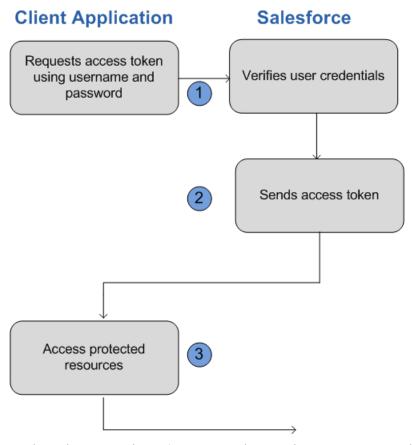
Keep the following considerations in mind when using the user-agent OAuth flow:

Because the access token is encoded into the redirection URI, it might be exposed to the end-user and other applications residing
on the computer or device. If you're authenticating using JavaScript, call window.location.replace(); to remove the
callback from the browser's history.

## Understanding the Username-Password OAuth Authentication Flow

Use the username-password authentication flow to authenticate when the consumer already has the user's credentials. In this flow, the user's credentials are used by the application to request an access token as shown in the following steps.

Warning: This OAuth authentication flow passes the user's credentials back and forth. Use this authentication flow only when necessary. No refresh token is issued.



1. The application uses the user's username and password to request an access token. This is done via an out-of-band POST request to the appropriate Salesforce token request endpoint, such as https://login.salesforce.com/services/oauth2/token. These request fields are required:

Parameter	Description
grant_type	Must be password for this authentication flow.
client_id	The Consumer Key from the connected app definition.
client_secret	The Consumer Secret from the connected app definition. Required unless the Require Secret for Web Server Flow setting is not enabled in the connected app definition.
username	End-user's username.
password	End-user's password.

Note: You must append the user's security token to their password A security token is an automatically-generated key from Salesforce. For example, if a user's password is mypassword, and their security token is XXXXXXXXXX, then the value provided for this parmeter must be mypasswordXXXXXXXXXXXX. For more information on

Parameter	Description
	security tokens see "Reset Your Security Token" in the online help.

An example request body might look something like the following:

```
grant_type=password&client_id=3MVG91KcPoNINVBIPJjdw1J9LLM82Hn
FVVX19KY1uA5mu0QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCscA9GE&client_secret=
1955279925675241571&username=testuser%40salesforce.com&password=mypassword123456
```

2. Salesforce verifies the user credentials, and if successful, sends a response to the application with the access token. This response contains the following values:

Parameters	Description
access_token	Access token that acts as a session ID that the application uses for making requests. This token should be protected as though it were user credentials.
instance_url	Identifies the Salesforce instance to which API calls are sent.
id	Identity URL that can be used to both identify the user and query for more information about the user. Can be used in an HTTP request to get more information about the end user.
issued_at	When the signature was created, represented as the number of seconds since the Unix epoch (00:00:00 UTC on 1 January 1970).
signature	Base64-encoded HMAC-SHA256 signature signed with the client_secret (private key) containing the concatenated ID and issued_at value. Use the signature to verify that the identity URL wasn't modified when the server sent it.

An example response body might look something like:

```
{"id":"https://login.salesforce.com/id/00Dx000000BV7z/005x00000012Q9P",
"issued_at":"1278448832702","instance_url":"https://yourInstance.salesforce.com/",
"signature":"0CmxinZir53Yex7nE0TD+zMpvIWYGb/bdJh6XfOH6EQ=","access_token":
"00Dx000000BV7z!AR8AQAxo9UfVkh8AlV0Gomt9Czx9LjHnSSpwBMmbRcgKFmxOtvxjTrKW1
9ye6PE3Ds1eQz3z8jr3W7_VbWmEu4Q8TVGSTHxs"}
```

**3.** The application uses the provided access token to access protected user data.

Keep the following considerations in mind when using the username-password OAuth flow:

• Since the user is never redirected to login at Salesforce in this flow, the user can't directly authorize the application, so no refresh tokens can be used. If your application requires refresh tokens, you should consider using the Web server or user-agent OAuth flow.

## How Does the OAuth Refresh Token Fit Into the Authentication Flow?

Both the web server OAuth authentication flow and user-agent flow provide a refresh token that can be used to get a new access token.

Access tokens have a limited lifetime specified by the session timeout in Salesforce. If an application uses an expired access token, a "Session expired or invalid" error is returned. If the application is using the web server or user-agent OAuth authentication flow, a refresh token is provided during authorization. Use this refresh token to get a new access token.

### **Refresh Token Request**

The client application obtains a new access token by sending a POST request to the token request endpoint with the following request parameters:

Parameters	Description
grant_type	Value must be refresh_token.
refresh_token	The refresh token the client application already received.
client_id	The Consumer Key from the connected app definition.
client_secret	The Consumer Secret from the connected app definition. Required unless the Require Secret for Web Server Flow setting is not enabled in the connected app definition. This parameter is optional.
format	Expected return format. The default is json. Values are:
	• urlencoded
	• json
	• xml
	The return format can also be specified in the header of the request using one of the following:
	• Accept:
	application/x-www-form-urlencoded
	• Accept: application/json
	• Accept: application/xml
	This parameter is optional.

This example shows a refresh token POST request that sends the client\_id and client\_secret in the body.

```
POST /services/oauth2/token HTTP/1.1
Host: login.salesforce.com/
grant_type=refresh_token&client_id=3MVG9lKcPoNINVBIPJjdw1J9LLM82HnFVVX19KY1uA5mu0
QqEWhqKpoW3svG3XHrXDiCQjK1mdgAvhCscA9GE&client_secret=1955279925675241571
&refresh_token=your token here
```

Instead of sending client credentials as parameters in the body of the POST request, Salesforce supports the HTTP Basic authentication scheme. The scheme requires the client id and client secret in the authentication header of the request as follows:

Authorization: Basic64Encode(client id:secret)

The client\_id and client\_secret are separated with a colon (:). For more information, see The OAuth 2.0 Authorization Framework.

This example shows a refresh token POST request that uses HTTP Basic authentication scheme (rather than sending client credentials in the POST request's body).

```
POST /services/oath2/token HTTP/1.1
Host: login.salesforce.com
Authorization: Basic client id=3MVG91KcPoNINVBIPJjdw1J9LLM82HnFVVX19KY1uA5mu0
QqEWhqKpoW3svG3XHrXDiCQjK1mdqAvhCscA9GE&client secret=1955279925675241571
grant type=refresh token&refresh token=your token here
```



🕜 Note: If the client id and client secret are sent in the POST request's body, the authentication header is ignored.

### Salesforce Response

After Salesforce verifies the refresh token request, it sends a response to the application with the following response body parameters:

Parameters	Description
access_token	Access token that acts as a session ID that the application uses for making requests. This token should be protected as though it were user credentials.
instance_url	Identifies the Salesforce instance to which API calls are sent.
id	Identity URL that can be used to both identify the user and query for more information about the user. Can be used in an HTTP request to get more information about the end user.
issued_at	When the signature was created, represented as the number of seconds since the Unix epoch (00:00:00 UTC on 1 January 1970).
signature	Base64-encoded HMAC-SHA256 signature signed with the client_secret (private key) containing the concatenated ID and issued_at value. Use the signature to verify that the identity URL wasn't modified when the server sent it.

This example shows a JSON response body:

```
{ "id": "https://login.salesforce.com/id/00Dx000000BV7z/005x00000012Q9P",
"issued at":"1278448384422","instance url":"https://yourInstance.salesforce.com/",
"signature": "SSSbLO/gBhmmyNUvN18ODBDFYHzakxOMggYtu+hDPsc=",
"access token": "00Dx0000000BV7z! AR8AQP0jITN80ESEsj5EbaZTFG0RNBaT1cyWk7T
rqoDjoNIWQ2ME sTZzBjfmOE6zMHq6y8PIW4eWze9JksNEkWUl.Cju7m4"}
```

### Considerations

Keep in mind the following considerations when using the refresh token OAuth process:

- The session timeout for an access token can be configured in Salesforce from Setup by entering Session Settings in the Quick Find box, then selecting **Session Settings**.
- If the application uses the username-password OAuth authentication flow, no refresh token is issued, as the user cannot authorize the application in this flow. If the access token expires, the application using username-password OAuth flow must reauthenticate the user.

## **Finding Additional Resources**

The following resources provide additional information about using OAuth with Salesforce:

- Authenticate Apps with OAuth
- Digging Deeper into OAuth 2.0 in Salesforce
- Using OAuth to Authorize External Applications

The following resources are examples of third-party client libraries that implement OAuth that you might find useful:

- For Ruby on Rails: OmniAuth
- For Java: Apache Amber
- Additional OAuth client libraries: OAuth.net

## Use CORS to Access Salesforce Resources from Web Browsers

Cross-Origin Resource Sharing (CORS) enables web browsers to request resources from origins other than their own (cross-origin). For example, using CORS, JavaScript code at

https://www.example.com could request a resource from

https://www.salesforce.com.To access supported Salesforce APIs, Apex REST resources, and Lightning Out from JavaScript code in a web browser, add the origin serving the code to a Salesforce CORS whitelist.

These Salesforce technologies support CORS.

- Analytics REST API
- Bulk API
- Chatter REST API
- Salesforce IoT REST API
- Lightning Out
- REST API
- User Interface API
- Apex REST

### EDITIONS

Available in: Salesforce Classic (not available in all orgs) and Lightning Experience

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

### **USER PERMISSIONS**

To create, read, update, and delete:

Modify All Data

In Salesforce, add the origin serving the code to a CORS whitelist. If a browser that supports CORS makes a request to an origin in the whitelist, Salesforce returns the origin in the Access-Control-Allow-Origin HTTP header, along with any additional CORS HTTP headers. If the origin is not included in the whitelist, Salesforce returns HTTP status code 403.

- 1. From Setup, enter CORS in the Quick Find box, then select CORS.
- 2. Select New.
- 3. Enter an origin URL pattern.

The origin URL pattern must include the HTTPS protocol (unless you're using your localhost) and a domain name and can include a port. The wildcard character (\*) is supported and must be in front of a second-level domain name. For example,

https://\*.example.com adds all subdomains of example.com to the whitelist.

The origin URL pattern can be an IP address. However, an IP address and a domain that resolve to the same address are not the same origin, and you must add them to the CORS whitelist as separate entries.

The origin URL pattern might not match the URL that appears in the address bar in your browser. Make sure that you're whitelisting the origin in the request header.



(1) Important: CORS does not support requests for unauthenticated resources, including OAuth endpoints. You must pass an OAuth token with requests that require it.

## API End-of-Life

Salesforce is committed to supporting each API version for a minimum of three years from the date of first release. In order to mature and improve the quality and performance of the API, versions that are more than three years old might cease to be supported.

When an API version is to be deprecated, advance notice is given at least one year before support ends. Salesforce will directly notify customers using API versions planned for deprecation.

## **CHAPTER 2** Quick Start

### In this chapter ...

- Prerequisites
- Step One: Obtain a Salesforce Developer Edition Organization
- Step Two: Set Up Authorization
- Step Three: Send HTTP Requests with cURL
- Step Four: Walk Through the Sample Code
- Using Workbench

Create a sample REST application in your development environment to see the power and flexibility of the REST API.

Quick Start Prerequisites

## **Prerequisites**

Completing the prerequisites makes it easier to build and use the guick-start sample.

If you're unfamiliar with cURL and JavaScript Object Notation (JSON), you can also use Workbench to obtain data.

- Install your development platform according to its product documentation.
- Become familiar with cURL, the tool used to execute REST requests in this quick start. If you use another tool, you should be familiar enough with it to translate the example code.
- Become familiar with JSON which is used in this quick start, or be able to translate samples from JSON to the standard you use.
- Enable an SSL endpoint in your application server.
- Become familiar with OAuth 2.0, which requires some setup. We provide the steps, but it will help if you are familiar with the basic concepts and workflow.
- Read through all the steps before beginning this quick start. You may also wish to review the rest of this document to familiarize yourself with terms and concepts.

## Step One: Obtain a Salesforce Developer Edition Organization

Set up a Developer Edition organization for testing your code.

If you are not already a member of the Lightning Platform developer community, go to developer.salesforce.com/signup and follow the instructions for signing up for a Developer Edition organization. Even if you already have Enterprise Edition, Unlimited Edition, or Performance Edition, use Developer Edition for developing, staging, and testing your solutions against sample data to protect your organization's live data. This is especially true for applications that insert, update, or delete data (as opposed to simply reading data).

If you already have a Developer Edition organization, verify that you have the API Enabled permission. This permission is enabled by default, but may have been changed by an administrator. For more information, see the help in the Salesforce user interface.

## Step Two: Set Up Authorization

You can set up authorization using OAuth 2.0 or by passing a session ID.



**Important**: If you're handling someone else's password, don't use session ID.

## Set Up OAuth 2.0

Setting up OAuth 2.0 requires that you take some steps within Salesforce and in other locations. If any of the steps are unfamiliar, see Understanding Authentication. The following example uses the Web server OAuth flow.

1. Create a connected app if you haven't already done so. See Create a Connected App in Salesforce Help.

Click **Enable OAuth Settings** and specify your callback URL and OAuth scopes. The Callback URL you supply here is the same as your application's callback URL. Usually it is a servlet if you work with Java. It must be secure: http:// does not work, only https://. For development environments, the callback URL is similar to

https://localhost:8443/RestTest/oauth/\_callback. When you click Save, the Consumer Key is created and displayed, and a Consumer Secret is created (click the link to reveal it).



Note: The OAuth 2.0 specification uses "client" instead of "consumer." Salesforce supports OAuth 2.0.

The values here correspond to the following values in the sample code in the rest of this procedure:

- client id is the Consumer Key
- client secret is the Consumer Secret
- redirect uri is the Callback URL.

In your client application, redirect the user to the appropriate Salesforce authorization endpoint. On successful user login, Salesforce calls your redirect URI with an authorization code. You use the authorization code in the next step to get the access token. The authorization code expires after 15 minutes.

2. From your Java or other client application, make a request to the appropriate Salesforce token request endpoint that passes in grant\_type, client\_id, client\_secret, and redirect\_uri. The redirect\_uri is the URI that Salesforce sends a callback to.

```
init.Params = {
    @WebInitParam(name = "clientId", value =
            "3MVG91KcPoNINVBJSoQsNCD.HHDdbugPsNXwwyFbgb47KWa PTv"),
    @WebInitParam(name = "clientSecret", value = "5678471853609579508"),
    @WebInitParam(name = "redirectUri", value =
            "https://localhost:8443/RestTest/oauth/ callback"),
    @WebInitParam(name = "environment", value =
            "https://login.salesforce.com/services/oauth2/token") }
HttpClient httpclient = new HttpClient();
PostMethod post = new PostMethod(environment);
post.addParameter("code", code);
post.addParameter("grant type", "authorization code");
  /** For session ID instead of OAuth 2.0, use "grant type", "password" **/
post.addParameter("client_id",clientId);
post.addParameter("client secret", clientSecret);
post.addParameter("redirect uri", redirectUri);
```

If the value of client\_id (or consumer key) and client\_secret (or consumer secret) are valid, Salesforce sends a callback to the URI specified in redirect\_uri that contains a value for access\_token.

**3.** Store the access token value as a cookie to use in all subsequent requests. For example:

```
//exception handling removed for brevity...
//this is the post from step 2
httpclient.executeMethod(post);
   String responseBody = post.getResponseBodyAsString();

String accessToken = null;
   JSONObject json = null;
   try {
      json = new JSONObject(responseBody);
      accessToken = json.getString("access_token");
      issuedAt = json.getString("issued_at");
      /** Use this to validate session
      * instead of expiring on browser close.
      */
   } catch (JSONException e) {
      e.printStackTrace();
   }
}
```

```
HttpServletResponse httpResponse = (HttpServletResponse)response;
Cookie session = new Cookie(ACCESS_TOKEN, accessToken);
session.setMaxAge(-1); //cookie not persistent, destroyed on browser exit
httpResponse.addCookie(session);
```

This step completes the authentication.

**4.** Once authenticated, every request must pass in the access\_token value in the header. It cannot be passed as a request parameter.

```
HttpClient httpclient = new HttpClient();
  GetMethod gm = new GetMethod(serviceUrl);
  //set the token in the header
  qm.setRequestHeader("Authorization", "Bearer "+accessToken);
  //set the SOQL as a query param
  NameValuePair[] params = new NameValuePair[1];
   * other option instead of query string, pass just the fields you want back:
    * https://instance name.salesforce.com/services/data/v20.0/sobjects/Account/
            001D000000INjVe?fields=AccountNumber,BillingPostalCode
   * /
  params[0] = new NameValuePair("q","SELECT name, title FROM Contact LIMIT 100");
  gm.setQueryString(params);
  httpclient.executeMethod(gm);
  String responseBody = gm.getResponseBodyAsString();
       //exception handling removed for brevity
  JSONObject json = new JSONObject(responseBody);
  JSONArray results = json.getJSONArray("records");
  for(int i = 0; i < results.length(); i++)</pre>
       response.getWriter().write(results.getJSONObject(i).getString("Name")+
         "+results.getJSONObject(i).getString("Title")+"\n");
```

The syntax to provide the access token in your REST requests:

```
Authorization: Bearer access_token
```

For example:

```
\verb|curl https://instance_name.salesforce.com/services/data/v20.0/ -H 'Authorization: Bearer | access\_token'|
```

## Session ID Authorization

You can use a session ID instead of an OAuth 2.0 access token if you aren't handling someone else's password:

1. Obtain a session ID, for example, a SOAP API login() call returns the session ID. You can also have the session ID, for example as part of the Apex current context. If you need a session ID just for testing purposes during development, use the username-password OAuth flow in a cURL command similar to the following:

```
curl https://login.salesforce.com/services/oauth2/token -d "grant_type=password" -d
"client_id=myclientid" -d "client_secret=myclientsecret"
    -d "username=mylogin@salesforce.com" -d "password=mypassword123456"
```

You must provide your client id, client secret, username, and password with user security token appended.

2. Use the session ID when you send a request to the resource. Substitute the ID for the token value. The syntax is the same:

```
Authorization: Bearer access_token

For example:

curl https://instance_name.salesforce.com/services/data/v20.0/ -H 'Authorization: Bearer access_token'
```

## Step Three: Send HTTP Requests with cURL

To interact with the Lightning Platform REST API, you need to set up your client application (we use cURL) to construct HTTP requests.

## **Setting Up Your Client Application**

The REST API uses HTTP GET and HTTP POST methods to send and receive JSON and XML content, so it is very simple to build client applications using the tool or the language of your choice. We use a command-line tool called cURL to simplify sending and receiving HTTP requests and responses.

cURL is pre-installed on many Linux and Mac systems. Windows users can download a version at curl.haxx.se/. When using HTTPS on Windows, ensure that your system meets the cURL requirements for SSL.

## Sending HTTP Requests Using REST API Resources

Your HTTP requests to a REST API resource should contain the following information:

- An HTTP method (HEAD, GET, POST, PATCH, or DELETE).
- An OAuth 2.0 access token used to authenticate the request. For information on how to retrieve the token, see Quick Start on page 22.
- An HTTP ACCEPT header used to indicate the resource format (XML or JSON), or a .json or .xml extension to the URI. The default is JSON.
- The Lightning Platform REST resource.
- Any JSON or XML files containing information needed for requests, such as updating a record with new information.

The HTTP methods are used to indicate the desired action, such as retrieving information, as well as creating, updating, and deleting records.

- HEAD is used to retrieve resource metadata.
- GET is used to retrieve information, such as basic resource summary information.
- POST is used to create a new object.
- PATCH is used to update a record.

DELETE is used to delete a record.

To access a resource, submit an HTTP request containing a header, method, and resource name.

For example, assume you want to create an Account record using a JSON-formatted file called newaccount.json. It contains the information to be stored in the new account:

```
{
    "Name" : "test"
}
```

Using cURL on your instance, the request would appear as follows:

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/ -H "Authorization: Bearer token -H "Content-Type: application/json" -d "@newaccount.json"
```

The request HTTP header:

### The response:

```
Date: Thu, 21 Oct 2010 22:16:22 GMT
Content-Length: 71
Location: /services/data/v20.0/sobjects/Account/001T000000NU96UIAT
Content-Type: application/json; charset=UTF-8 Server:
{ "id" : "001T000000NU96UIAT",
    "errors" : [ ],
    "success" : true }
```

For a list of the resources and their corresponding URIs, see Reference on page 104.

SEE ALSO:

Using cURL in the REST Examples

## Step Four: Walk Through the Sample Code

In this section you will create a series of REST requests. cURL will be used to construct the requests, and JSON will be used as the format for all requests and responses.

In each request, a base URI will be used in conjunction with the REST resource. The base URI for these examples is https://yourInstance.salesforce.com/services/data. For more information, see Lightning Platform REST Resources on page 2.

In this example, a series of REST requests will be used in the following scenario:

- 1. Get the Salesforce version.
- **2.** Use the Salesforce version to get a list of the resources available.

- 3. Use one of the resources to get a list of the available objects.
- 4. Select one of the objects and get a description of its metadata.
- **5.** Get a list of fields on that same object.
- **6.** Execute a SOQL query to retrieve values from all name fields on Account records.
- 7. Update the Billing City for one of the Account objects.

### Get the Salesforce Version

Begin by retrieving information about each available Salesforce version. To do this, submit a request for the Versions resource. In this case the request does not require authentication:

```
curl https://yourInstance.salesforce.com/services/data/
```

The output from this request, including the response header:

The output specifies the resources available for all valid versions (your result may include more than one value). Next, use one of these versions to discover the resources it contains.

## Get a List of Resources

The next step is to retrieve a list of the resources available for Salesforce, in this example for version 20.0. To do this, submit a request for the Resources by Version:

```
curl https://yourInstance.salesforce.com/services/data/v20.0/ -H "Authorization: Bearer
access_token" -H "X-PrettyPrint:1"
```

The output from this request is as follows:

```
"sobjects" : "/services/data/v20.0/sobjects",
"search" : "/services/data/v20.0/search",
"query" : "/services/data/v20.0/query",
"recent" : "/services/data/v20.0/recent"
}
```

From this output you can see that sobjects is one of the available resources in Salesforce version 20.0. You will be able to use this resource in the next request to retrieve the available objects.

### Get a List of Available Objects

Now that you have the list of available resources, you can request a list of the available objects. To do this, submit a request for the Describe Global:

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/ -H "Authorization: Bearer access_token" -H "X-PrettyPrint:1"
```

The output from this request is as follows:

```
Transfer-Encoding: chunked
Content-Type: application/json;
charset=UTF-8 Server:
 "encoding" : "UTF-8",
 "maxBatchSize" : 200,
 "sobjects" : [ {
    "name" : "Account",
    "label" : "Account",
    "custom" : false,
    "keyPrefix" : "001",
    "updateable" : true,
    "searchable" : true,
    "labelPlural" : "Accounts",
    "layoutable" : true,
    "activateable" : false,
    "urls" : { "sobject" : "/services/data/v20.0/sobjects/Account",
    "describe" : "/services/data/v20.0/sobjects/Account/describe",
    "rowTemplate" : "/services/data/v20.0/sobjects/Account/{ID}" },
    "createable" : true,
    "customSetting" : false,
    "deletable" : true,
    "deprecatedAndHidden" : false,
    "feedEnabled" : false,
    "mergeable" : true,
    "queryable" : true,
    "replicateable" : true,
    "retrieveable" : true,
    "undeletable" : true,
    "triggerable" : true },
   },
```

From this output you can see that the Account object is available. You will be able to get more information about the Account object in the next steps.

### Get Basic Object Information

Now that you have identified the Account object as an available resource, you can retrieve some basic information about its metadata. To do this, submit a request for the SObject Basic Information:

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/ -H "Authorization: Bearer access_token" -H "X-PrettyPrint:1"
```

The output from this request is as follows:

```
"objectDescribe" :
        "name" : "Account",
        "updateable" : true,
        "label" : "Account",
        "keyPrefix" : "001",
        "replicateable" : true,
        "retrieveable" : true,
        "undeletable" : true,
        "triggerable" : true
    } ,
    "recentItems" :
    [
        {
            "attributes" :
                "type" : "Account",
                "url" : "/services/data/v20.0/sobjects/Account/001D000000INjVeIAL"
            "Id" : "001D000000INjVeIAL",
            "Name" : "asdasdasd"
        },
        . . .
    ]
}
```

From this output you can see some basic attributes of the Account object, such as its name and label, as well as a list of the most recently used Accounts. Since you may need more information about its fields, such as length and default values, in the next step you will retrieve more detailed information about the Account object.

### Get a List of Fields

Now that you have some basic information about the Account object's metadata, you may be interested in retrieving more detailed information. To do this, submit a request for the SObject Describe:

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/describe/-H "Authorization: Bearer access\_token" -H "X-PrettyPrint:1"
```

The output from this request is as follows:

```
"type" : "id",
        "defaultValue" : { "value" : null },
        "updateable" : false,
        "label" : "Account ID",
    },
    . . .
"updateable" : true,
"label" : "Account",
"urls" :
    "uiEditTemplate" : "https://yourInstance.salesforce.com/{ID}/e",
    "sobject": "/services/data/v20.0/sobjects/Account",
    "uiDetailTemplate" : "https://yourInstance.salesforce.com/{ID}",
    "describe" : "/services/data/v20.0/sobjects/Account/describe",
    "rowTemplate" : "/services/data/v20.0/sobjects/Account/{ID}",
    "uiNewRecord" : "https://yourInstance.salesforce.com/001/e"
"childRelationships" :
[
    {
        "field" : "ParentId",
        "deprecatedAndHidden" : false,
        . . .
    },
],
"createable" : true,
"customSetting" : false,
```

From this output you can see much more detailed information about the Account object, such as its field attributes and child relationships. Now you have enough information to construct useful queries and updates for the Account objects in your organization, which you will do in the next steps.

### **Execute a SOQL Query**

Now that you know the field names on the Account object, you can execute a SOQL query, for example, to retrieve a list of all the Account name values. To do this, submit a Query request:

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/query?q=SELECT+name+from+Account
-H "Authorization: Bearer access_token" -H "X-PrettyPrint:1"
```

The output from this request is as follows:

```
"done" : true,
"totalSize" : 14,
"records" :
```

```
[
            "attributes" :
                 "type" : "Account",
                 "url": "/services/data/v20.0/sobjects/Account/001D000000IRFmaIAH"
            },
            "Name" : "Test 1"
        },
        {
            "attributes" :
                 "type" : "Account",
                 "url" : "/services/data/v20.0/sobjects/Account/001D000000IomazIAB"
            "Name" : "Test 2"
        },
        . . .
    ]
}
```

From this output you have a listing of the available Account names, and each name's preceding attributes include the Account IDs. In the next step you will use this information to update one of the accounts.



Note: You can find more information about SOQL in the Salesforce SOQL and SOSL Reference Guide.

### Update a Field on a Record

Now that you have the Account names and IDs, you can retrieve one of the accounts and update its Billing City. To do this, you will need to submit an SObject Rows request. To update the object, supply the new information about the Billing City. Create a text file called patchaccount.json containing the following information:

```
{
    "BillingCity" : "Fremont"
}
```

Specify this JSON file in the REST request. The cURL notation requires the —d option when specifying data. For more information, see <a href="http://curl.haxx.se/docs/manpage.html">http://curl.haxx.se/docs/manpage.html</a>.

Also, specify the PATCH method, which is used for updating a REST resource. The following cURL command retrieves the specified Account object using its ID field, and updates its Billing City.

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/001D000000IroHJ
-H "Authorization: Bearer access_token" -H "X-PrettyPrint:1" -H "Content-Type:
application/json" --data-binary @patchaccount.json -X PATCH
```

No response body is returned, just the headers:

```
HTTP/1.1 204 No Content
Server:
Content-Length: 0
```

Refresh the page on the account and you will see that the Billing Address has changed to Fremont.

Quick Start Using Workbench

### Other Resources

- Search for Ruby on *developer.salesforce.com*
- Lightning Platform Cookbook recipe for getting started in Ruby
- Lightning Platform REST API Board

# **Using Workbench**

Use the Workbench tool to obtain data about your organization.

If you don't want to use CURL, you can use the Workbench REST explorer to obtain response data.

- 1. Log in to your organization.
- 2. Open a new browser tab and navigate to https://developer.salesforce.com/page/Workbench.
- 3. Log in to Workbench and allow access to your organization. Workbench is a public site and won't retain your data.
- **4.** Click **Utilities** > **REST Explorer**.
- **5.** Ensure that **Get** is selected. The Execute text box is prepopulated with a portion of a resource path. Add the remaining information for your resource. For example, if your cURL syntax is

https://**yourInstance**.salesforce.com/services/data/v32.0/sobjects/EventLogFile/describe -H "Authorization: Bearer token"

type

/services/data/v32.0/sobjects/EventLogFile/describe.

- **6.** Click **Execute**.
- 7. Click **Expand All** or **Show Raw Response** to view your data.
- 1 Tip: If you receive a "Service not found" message, verify your resource path.

# **CHAPTER 3** Examples

### In this chapter ...

- Getting Information About My Organization
- Working with Object Metadata
- Working with Records
- Delete Lightning Experience Event Series
- Working with Searches and Queries
- Insert or Update Blob Data
- Working with Recently Viewed Information
- Managing User Passwords
- Working with Approval Processes and Process Rules
- Using Event Monitoring
- Using Composite Resources

This section provides examples of using REST API resources to do a variety of different tasks, including working with objects, organization information, and queries.

For complete reference information on REST API resources, see Reference on page 104.

### Getting Information About My Organization

The examples in this section use REST API resources to retrieve organization-level information, such as a list of all objects available in your organization.

#### IN THIS SECTION:

#### List Available REST API Versions

Use the Versions resource to list summary information about each REST API version currently available, including the version, label, and a link to each version's root. You do not need authentication to retrieve the list of versions.

#### List Organization Limits

Use the Limits resource to list your org limits.

#### List Available REST Resources

Use the Resources by Version resource to list the resources available for the specified API version. This provides the name and URI of each additional resource.

### Get a List of Objects

Use the Describe Global resource to list the objects available in your org and available to the logged-in user. This resource also returns the org encoding, as well as maximum batch size permitted in queries.

### Get an Image from a Rich Text Area Field

Use the SObject Rich Text Image Retrieve to retrieve an image that has been uploaded to a rich text area field.

### Get a List of Objects If Metadata Has Changed

Use the Describe Global resource and the If-Modified-Since HTTP header to determine if an object's metadata has changed.

### List Available REST API Versions

Use the Versions resource to list summary information about each REST API version currently available, including the version, label, and a link to each version's root. You do not need authentication to retrieve the list of versions.

### Example usage

```
curl https://yourInstance.salesforce.com/services/data/
```

#### **Example request body**

none required

#### **Example JSON response body**

```
{
    "version" : "20.0",
    "label" : "Winter '11",
    "url" : "/services/data/v20.0"
},
{
    "version" : "21.0",
    "label" : "Spring '11",
    "url" : "/services/data/v21.0"
},
...
{
```

Examples List Organization Limits

```
"version": "26.0",

"label": "Winter '13",

"url": "/services/data/v26.0"

}
```

### **Example XML response body**

```
<?xml version="1.0" encoding="UTF-8"?>
<Versions>
   <Version>
       <label>Winter &apos;11</label>
       <url>/services/data/v20.0</url>
       <version>20.0
    </Version>
   <Version>
       <label>Spring &apos;11</label>
       <url>/services/data/v21.0</url>
       <version>21.0</version>
   </Version>
    <Version>
       <label>Winter &apos;13</label>
       <url>/services/data/v26.0</url>
       <version>26.0
   </Version>
</Versions>
```

### List Organization Limits

Use the Limits resource to list your org limits.

- Max is the limit for the org.
- Remaining is the number of calls or events left for the org.

### **Example usage**

```
curl https://instance.salesforce.com/services/data/v45.0/limits/ -H "Authorization:
Bearer token "X-PrettyPrint:1"
```

### **Example request body**

none required

#### **Example response body**

```
"ConcurrentAsyncGetReportInstances" : {
    "Max" : 200,
    "Remaining" : 200
},

"ConcurrentSyncReportRuns" : {
    "Max" : 20,
    "Remaining" : 20
},

"DailyApiRequests" : {
```

Examples List Organization Limits

```
"Max" : 15000,
  "Remaining": 14998
},
"DailyAsyncApexExecutions" : {
  "Max" : 250000,
  "Remaining" : 250000
},
"DailyBulkApiRequests" : {
  "Max" : 5000,
  "Remaining" : 5000
},
"DailyBulkV2QueryFileStorageMB": {
 "Max": 976562,
  "Remaining": 976562
"DailyBulkV2QueryJobs": {
  "Max": 10000,
  "Remaining": 10000
},
"DailyDurableGenericStreamingApiEvents" : {
  "Max" : 10000,
 "Remaining" : 10000
"DailyDurableStreamingApiEvents" : {
 "Max" : 10000,
  "Remaining" : 10000
"DailyGenericStreamingApiEvents" : {
  "Max" : 10000,
  "Remaining" : 10000
"DailyStandardVolumePlatformEvents" : {
  "Max" : 10000,
 "Remaining" : 10000
"DailyStreamingApiEvents" : {
  "Max" : 10000,
  "Remaining" : 10000
},
"DailyWorkflowEmails" : {
 "Max" : 390,
  "Remaining" : 390
},
"DataStorageMB" : {
  "Max" : 5,
  "Remaining" : 5
},
"DurableStreamingApiConcurrentClients" : {
 "Max" : 20,
  "Remaining" : 20
},
"FileStorageMB" : {
  "Max" : 20,
  "Remaining" : 20
```

Examples List Organization Limits

```
"HourlyAsyncReportRuns" : {
 "Max" : 1200,
 "Remaining" : 1200
},
"HourlyDashboardRefreshes" : {
 "Max" : 200,
 "Remaining" : 200
},
"HourlyDashboardResults" : {
 "Max" : 5000,
 "Remaining" : 5000
"HourlyDashboardStatuses" : {
 "Max" : 999999999,
 "Remaining": 999999999
},
"HourlyLongTermIdMapping" : {
 "Max" : 100000,
 "Remaining" : 100000
"HourlyODataCallout" : {
 "Remaining" : 9999,
 "Max" : 10000
},
"HourlyPublishedPlatformEvents" : {
 "Max" : 50000,
 "Remaining" : 50000
},
"HourlyPublishedStandardVolumePlatformEvents" : {
 "Max" : 1000,
 "Remaining": 1000
},
"HourlyShortTermIdMapping" : {
 "Max" : 100000,
 "Remaining" : 100000
},
"HourlySyncReportRuns" : {
 "Max" : 500,
 "Remaining" : 500
"HourlyTimeBasedWorkflow" : {
 "Max" : 50,
 "Remaining": 50
},
"MassEmail" : {
 "Max" : 10,
 "Remaining": 10
"MonthlyPlatformEvents" : {
 "Max" : 300000,
 "Remaining" : 300000
"SingleEmail" : {
```

Examples List Available REST Resources

```
"Max" : 15,
"Remaining" : 15
}}
```

### List Available REST Resources

Use the Resources by Version resource to list the resources available for the specified API version. This provides the name and URI of each additional resource.

### Example

```
curl https://yourInstance.salesforce.com/services/data/v26.0/ -H "Authorization: Bearer
token"
```

### **Example request body**

none required

### **Example JSON response body**

```
"sobjects" : "/services/data/v26.0/sobjects",
   "licensing" : "/services/data/v26.0/licensing",
   "connect" : "/services/data/v26.0/connect",
   "search" : "/services/data/v26.0/search",
   "query" : "/services/data/v26.0/query",
   "tooling" : "/services/data/v26.0/tooling",
   "chatter" : "/services/data/v26.0/chatter",
   "recent" : "/services/data/v26.0/recent"
}
```

### **Example XML response body**

### Get a List of Objects

Use the Describe Global resource to list the objects available in your org and available to the logged-in user. This resource also returns the org encoding, as well as maximum batch size permitted in queries.

### Example usage

```
\verb| curl https://yourInstance.salesforce.com/services/data/v37.0/sobjects/ -H "Authorization: Bearer token" \\
```

### **Example request body**

none required

### **Example response body**

```
"encoding" : "UTF-8",
"maxBatchSize" : 200,
"sobjects" : [ {
  "activateable" : false,
  "custom" : false,
  "customSetting" : false,
  "createable" : true,
  "deletable" : true,
  "deprecatedAndHidden" : false,
  "feedEnabled" : true,
  "keyPrefix" : "001",
  "label" : "Account",
  "labelPlural" : "Accounts",
  "layoutable" : true,
  "mergeable" : true,
  "mruEnabled" : true,
  "name" : "Account",
  "queryable" : true,
  "replicateable" : true,
  "retrieveable" : true,
  "searchable" : true,
  "triggerable" : true
  "undeletable" : true,
  "updateable" : true,
  "urls" : {
    "sobject": "/services/data/v37.0/sobjects/Account",
    "describe": "/services/data/v37.0/sobjects/Account/describe",
    "rowTemplate" : "/services/data/v37.0/sobjects/Account/{ID}"
 },
},
. . .
]
```

### Get an Image from a Rich Text Area Field

Use the SObject Rich Text Image Retrieve to retrieve an image that has been uploaded to a rich text area field.

The following example retrieves an image that has been uploaded to a rich text area field for a Lead record.

#### Example for retrieving an image from a rich text field for a Lead record

```
curl
https://yourInstance.salesforce.com/services/data/v43.0/sabjects/leed/00gl122223333/rid4TextImageFields/oustonRTA_c/OEMR0000000A8V/body
-H "Authorization: Bearer token"
```

#### **Example request body**

None required.

#### **Example response body**

Attachment body content is returned in binary form. The response content type is not JSON or XML since the returned data is binary.

### Get a List of Objects If Metadata Has Changed

Use the Describe Global resource and the If-Modified-Since HTTP header to determine if an object's metadata has changed.

You can include the If-Modified-Since header with a date in EEE, dd MMM yyyy HH:mm:ss z format when you use the Describe Global resource. If you do, response metadata is returned only if an available object's metadata has changed since the provided date. If no metadata has been modified since the provided date, a 304 Not Modified status code is returned with no response body.

The following example assumes that no changes have been made to objects after March 23, 2015.

### **Example Describe Global request**

/services/data/v34.0/sobjects

#### Example If-Modified-Since header used with request

```
If-Modified-Since: Tue, 23 Mar 2015 00:00:00 GMT
```

### **Example response body**

No response body returned

### Example response status code

```
HTTP/1.1 304 Not Modified
Date: Wed, 25 Jul 2015 00:05:46 GMT
```

If changes to an object were made after March 23, 2015, the response body contains metadata for all available objects. For an example, see Get a List of Objects.

### Working with Object Metadata

The examples in this section use REST API resources to retrieve object metadata information. For modifying or creating object metadata information, see the *Metadata API Developer Guide*.

#### IN THIS SECTION:

Retrieve Metadata for an Object

Use the SObject Basic Information resource to retrieve metadata for an object.

Get Field and Other Metadata for an Object

Use the SObject Describe resource to retrieve all the metadata for an object, including information about each field, URLs, and child relationships.

Get Object Metadata Changes

Use the SObject Describe resource and the If-Modified-Since HTTP header to determine if object metadata has changed.

### Retrieve Metadata for an Object

Use the SObject Basic Information resource to retrieve metadata for an object.

### **Example for retrieving Account metadata**

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/ -H
"Authorization: Bearer token"
```

### **Example request body for retrieving Account metadata**

none required

### Example response body for retrieving Account metadata

```
"objectDescribe" :
  "name" : "Account",
  "updateable" : true,
  "label" : "Account",
  "keyPrefix" : "001",
  "replicateable" : true,
  "retrieveable" : true,
  "undeletable" : true,
  "triggerable" : true
},
"recentItems" :
    "attributes" :
      "type" : "Account",
     "url" : "/services/data/v20.0/sobjects/Account/001D000000INjVeIAL"
    "Id" : "001D000000INjVeIAL",
    "Name" : "asdasdasd"
  },
```

To get a complete description of an object, including field names and their metadata, see Get a List of Objects.

# Get Field and Other Metadata for an Object

Use the SObject Describe resource to retrieve all the metadata for an object, including information about each field, URLs, and child relationships.

### **Example**

```
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/describe/ -H "Authorization: Bearer token"
```

### **Example request body**

none required

### **Example response body**

```
"name" : "Account",
 "fields" :
   {
     "length" : 18,
      "name" : "Id",
      "type" : "id",
      "defaultValue" : {     "value" : null },
      "updateable" : false,
      "label" : "Account ID",
   },
   . . .
 ],
 "updateable" : true,
 "label" : "Account",
 "keyPrefix" : "001",
  "custom" : false,
  . . .
  "urls" :
   "uiEditTemplate" : "https://yourInstance.salesforce.com/{ID}/e",
   "sobject" : "/services/data/v20.0/sobjects/Account",
   "uiDetailTemplate" : "https://yourInstance.salesforce.com/{ID}",
   . . .
  },
  "childRelationships" :
  [
     "field" : "ParentId",
     "deprecatedAndHidden" : false,
   },
    . . . .
 ],
 "createable" : true,
 "customSetting" : false,
}
```

### Get Object Metadata Changes

Use the SObject Describe resource and the If-Modified-Since HTTP header to determine if object metadata has changed.

You can include the If-Modified-Since header with a date in EEE, dd MMM yyyy HH:mm:ss z format when you use the SObject Describe resource. If you do, response metadata will only be returned if the object metadata has changed since the provided date. If the metadata has not been modified since the provided date, a 304 Not Modified status code is returned, with no response body.

The following example assumes that no changes, such as new custom fields, have been made to the Merchandise\_\_c object after July 3rd, 2013.

### **Example SObject Describe request**

/services/data/v29.0/sobjects/Merchandise c/describe

#### Example If-Modified-Since header used with request

If-Modified-Since: Wed, 3 Jul 2013 19:43:31 GMT

### **Example response body**

No response body returned

#### Example response status code

```
HTTP/1.1 304 Not Modified
Date: Fri, 12 Jul 2013 05:03:24 GMT
```

If there were changes to Merchandise\_\_c made after July 3rd, 2013, the response body would contain the metadata for Merchandise\_\_c. See Get Field and Other Metadata for an Object for an example.

## Working with Records

The examples in this section use REST API resources to create, retrieve, update, and delete records, along with other record-related operations.

#### IN THIS SECTION:

### Create a Record

Use the SObject Basic Information resource to create new records. You supply the required field values in the request data, and then use the POST method of the resource. The response body will contain the ID of the created record if the call is successful.

#### Update a Record

You use the SObject Rows resource to update records. Provide the updated record information in your request data and use the PATCH method of the resource with a specific record ID to update that record. Records in a single file must be of the same object type.

### Delete a Record

Use the SObject Rows resource to delete records. Specify the record ID and use the DELETE method of the resource to delete a record.

#### Get Field Values from a Standard Object Record

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

### Get Field Values from an External Object Record by Using the Salesforce ID

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

Examples Create a Record

### Get Field Values from an External Object Record by Using the External ID Standard Field

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

### Retrieve a Record Using an External ID

You can use the GET method of the SObject Rows by External ID resource to retrieve records with a specific external ID.

#### Insert or Update (Upsert) a Record Using an External ID

You can use the SObject Rows by External ID resource to create records or update existing records (upsert) based on the value of a specified external ID field.

### Traverse Relationships with Friendly URLs

You can traverse relationship fields in objects by constructing friendly URLs via the SObject Relationship resource. This approach allows you to directly access records associated with relationships. Using friendly URLs is easier than accessing records by obtaining object IDs from relationship fields and then inspecting the associated object ID record.

### Get Attachment Content from a Record

Use the SObject Blob Retrieve resource to retrieve blob data for a given record.

#### Get a List of Deleted Records Within a Given Timeframe

Use the SObject Get Deleted resource to get a list of deleted records for the specified object. Specify the date and time range within which the records for the given object were deleted. Deleted records are written to a delete log (that is periodically purged), and will be filtered out of most operations, such as SObject Rows or Query (although QueryAll will include deleted records in results).

### Get a List of Updated Records Within a Given Timeframe

Use the SObject Get Updated resource to get a list of updated (modified or added) records for the specified object. Specify the date and time range within which the records for the given object were updated.

### Create a Record

Use the SObject Basic Information resource to create new records. You supply the required field values in the request data, and then use the POST method of the resource. The response body will contain the ID of the created record if the call is successful.

The following example creates a new Account record, with the field values provided in newaccount.json.

### **Example for creating a new Account**

```
curl https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/ -H "Authorization: Bearer token -H "Content-Type: application/json" -d "@newaccount.json"
```

### Example request body newaccount. json file for creating a new Account

```
{
  "Name" : "Express Logistics and Transport"
}
```

### Example response body after successfully creating a new Account

```
{
  "id" : "001D000000IqhSLIAZ",
  "errors" : [],
  "success" : true
}
```

Examples Update a Record

### Update a Record

You use the SObject Rows resource to update records. Provide the updated record information in your request data and use the PATCH method of the resource with a specific record ID to update that record. Records in a single file must be of the same object type.

In the following example, the Billing City within an Account is updated. The updated record information is provided in patchaccount.json.

### **Example for updating an Account object**

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/001D000000INjVe
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d
@patchaccount.json -X PATCH
```

### Example request body patchaccount. json file for updating fields in an Account object

```
{
    "BillingCity" : "San Francisco"
}
```

### Example response body for updating fields in an Account object

none returned

#### **Error response**

See Status Codes and Error Responses on page 277.

The following example uses Java and HttpClient to update a record using REST API. Note that there is no PatchMethod in HttpClient, so PostMethod is overridden to return "PATCH" as its method name. This example assumes the resource URL has been passed in and contains the object name and record ID.

```
public static void patch(String url, String sid) throws IOException {
 PostMethod m = new PostMethod(url) {
    @Override public String getName() { return "PATCH"; }
 };
 m.setRequestHeader("Authorization", "OAuth " + sid);
 Map<String, Object> accUpdate = new HashMap<String, Object>();
 accUpdate.put("Name", "Patch test");
 ObjectMapper mapper = new ObjectMapper();
 m.setRequestEntity(new StringRequestEntity(mapper.writeValueAsString(accUpdate),
"application/json", "UTF-8"));
 HttpClient c = new HttpClient();
 int sc = c.executeMethod(m);
 System.out.println("PATCH call returned a status code of " + sc);
 if (sc > 299) {
    // deserialize the returned error message
   List<ApiError> errors = mapper.readValue(m.getResponseBodyAsStream(), new
TypeReference<List<ApiError>>() {} );
    for (ApiError e : errors)
      System.out.println(e.errorCode + " " + e.message);
  }
private static class ApiError {
```

Examples Delete a Record

```
public String errorCode;
public String message;
public String [] fields;
}
```

If you use an HTTP library that doesn't allow overriding or setting an arbitrary HTTP method name, you can send a POST request and provide an override to the HTTP method via the query string parameter \_HttpMethod. In the PATCH example, you can replace the PostMethod line with one that doesn't use override:

```
PostMethod m = new PostMethod(url + "?_HttpMethod=PATCH");
```

### Delete a Record

Use the SObject Rows resource to delete records. Specify the record ID and use the DELETE method of the resource to delete a record.

### **Example for deleting an Account record**

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/001D000000INjVe
-H "Authorization: Bearer token" -X DELETE
```

### **Example request body**

None needed

### **Example response body**

None returned

### Get Field Values from a Standard Object Record

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

In the following example, the Account Number and Billing Postal Code are retrieved from an Account.

### Example for retrieving values from fields on an Account object

```
/services/data/v20.0/sobjects/Account/001D000000INjVe ?fields=AccountNumber,BillingPostalCode
```

### **Example request body**

None required

### **Example response body**

```
"AccountNumber" : "CD656092",
    "BillingPostalCode" : "27215",
}
```

# Get Field Values from an External Object Record by Using the Salesforce ID

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

In the following example, the Country\_c custom field is retrieved from an external object that's associated with a non-high-data-volume external data source.

### Example for retrieving values from fields on the Customer external object

```
/services/data/v32.0/sobjects/Customer__x/x01D0000000002RIAQ?fields=Country__c
```

### **Example request body**

None required

#### **Example response body**

```
{
  "attributes" : {
    "type" : "Customer__x",
    "url" : "/services/data/v32.0/sobjects/Customer__x/x01D0000000002RIAQ"
},
  "Country__c" : "Argentina",
  "Id" : "x01D0000000002RIAQ"
}
```

# Get Field Values from an External Object Record by Using the External ID Standard Field

You use the SObject Rows resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

In the following example, the Country\_c custom field is retrieved from an external object. Notice that the id (CACTU) isn't a Salesforce ID. Instead, it's the External ID standard field of the external object.

### Example for retrieving values from fields on the Customer external object

```
/services/data/v32.0/sobjects/Customer__x/CACTU?fields=Country__c
```

### **Example request body**

None required

### **Example response body**

```
"attributes" : {
    "type" : "Customer__x",
    "url" : "/services/data/v32.0/sobjects/Customer__x/CACTU"
},
"Country__c" : "Argentina",
"ExternalId" : "CACTU"
}
```

### Retrieve a Record Using an External ID

You can use the GET method of the SObject Rows by External ID resource to retrieve records with a specific external ID.

The following example assumes there is a Merchandise c custom object with a MerchandiseExtID c external ID field.

### Example usage for retrieving a Merchandise\_\_c record using an external ID

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Merchandise__c/MerchandiseExtID__c/123
-H "Authorization: Bearer token"
```

### **Example request body**

none required

### **Example response body**

```
"attributes" : {
   "type" : "Merchandise c",
   "url": "/services/data/v20.0/sobjects/Merchandise c/a00D0000008oWP8IAM"
"Id" : "a00D0000008oWP8IAM",
"OwnerId": "005D0000001KyEIIA0",
"IsDeleted" : false,
"Name" : "Example Merchandise",
"CreatedDate": "2012-07-12T17:49:01.000+0000",
"CreatedById" : "005D000001KyEIIA0",
"LastModifiedDate": "2012-07-12T17:49:01.000+0000",
"LastModifiedById" : "005D000001KyEIIA0",
"SystemModstamp" : "2012-07-12T17:49:01.000+0000",
"Description c" : "Merch with external ID",
"Price c" : 10.0,
"Total_Inventory__c" : 100.0,
"Distributor c" : null,
"MerchandiseExtID c": 123.0
```

### Insert or Update (Upsert) a Record Using an External ID

You can use the SObject Rows by External ID resource to create records or update existing records (upsert) based on the value of a specified external ID field.

- If the specified value doesn't exist, a new record is created.
- If a record does exist with that value, the field values specified in the request body are updated.
- If the value is not unique, REST API returns a 300 response with the list of matching records.

The following sections show you how to work with the external ID resource to retrieve records by external ID and upsert records.

### **Upserting New Records**

This example uses the PATCH method to insert a new record. It assumes that an external ID field, "customExtIdField\_\_c," has been added to Account. It also assumes that an Account record with a customExtIdField value of 11999 does not already exist.

### Example for upserting a record that does not yet exist

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/customExtIdField_c/11999
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newrecord.json
-X PATCH
```

### Example JSON request body newrecord. json file

```
"Name" : "California Wheat Corporation",
"Type" : "New Customer"
}
```

### **Example JSON response**

The successful response is:

```
"id" : "0019000001pPvHAAU",
    "errors" : [ ],
    "success" : true,
    "created": true
}
```

The HTTP status code is 201 (Created).



Note: The created parameter is present in the response in API version 46.0 and later. It doesn't appear in earlier versions.

### **Error responses**

Incorrect external ID field:

```
"message" : "The requested resource does not exist",
   "errorCode" : "NOT_FOUND"
}
```

For more information, see Status Codes and Error Responses on page 277.

### Inserting New Records Using ia as the External ID

This example uses the POST method as a special case to insert a record where the Id field is treated as the external ID. Because the value of Id is null, it's omitted from the request. This pattern is useful when you're writing code to upsert multiple records by different external IDs and you don't want to request a separate resource. POST using Id is available in API version 37.0 and later.

### Example of inserting a record that does not yet exist

```
curl https://yourInstance.salesforce.com/services/data/v37.0/sobjects/Account/Id -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newrecord.json -X POST
```

### Example JSON request body newrecord. json file

```
"Name" : "California Wheat Corporation",
"Type" : "New Customer"
}
```

#### **Example JSON response**

The successful response is:

```
"id" : "001D000000Kv3g5IAB",
    "success" : true,
    "errors" : [ ],
```

```
"created": true
}
```

The HTTP status code is 201 (Created).



Note: The created parameter is present in the response in API version 46.0 and later. It doesn't appear in earlier versions.

### **Upserting Existing Records**

This example uses the PATCH method to update an existing record. It assumes that an external ID field, "customExtldField\_\_c," has been added to Account and an Account record with a customExtldField value of 11999 exists. The request uses updates.json to specify the updated field values.

### Example of upserting an existing record

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/customExtIdField__c/11999
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @updates.json
-X PATCH
```

### Example JSON request body updates.json file

```
{
    "BillingCity" : "San Francisco"
}
```

### **Example JSON response**

In API version 46.0 and later, the HTTP status code is 200 (OK) and the successful response is:

```
"id" : "001D000000Kv3g5IAB",
    "success" : true,
    "errors" : [],
    "created": false
}
```

In API version 45.0 and earlier, the HTTP status code is 204 (No Content) and there is no response body.

### **Error responses**

If the external ID value isn't unique, an HTTP status code 300 is returned, plus a list of the records that matched the query. For more information about errors, see Status Codes and Error Responses on page 277.

If the external ID field doesn't exist, an error message and code is returned:

```
"message" : "The requested resource does not exist",
   "errorCode" : "NOT_FOUND"
}
```

### Upserting Records and Associating with an External ID

If you have an object that references another object using a relationship, you can use REST API to both insert or update a record and also reference another object using an external ID.

The following example creates a record and associates it with a parent record via external ID. It assumes the following:

• A Merchandise\_c custom object that has an external ID field named MerchandiseExtID\_c.

- A Line\_Item\_\_c custom object that has an external ID field named LineItemExtID\_\_c, and a relationship to Merchandise\_\_c.
- A Merchandise\_c record exists that has a MerchandiseExtID\_c value of 123.
- A Line\_Item\_\_c record with a LineItemExtID\_\_c value of 456 does **not** exist. This is the record that gets created and associated to the Merchandise c record.

### Example of upserting a record and referencing a related object

```
curl
https://yourInstance.salesforce.com/services/data/v25.0/sobjects/Line_Item__c/LineItemExtID__c/456
  -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @new.json -X
PATCH
```

### Example JSON request body new.json file

Notice that the related Merchandise\_\_c record is referenced using the Merchandise\_\_c's external ID field.

```
"Name" : "LineItemCreatedViaExtID",
"Merchandise__r" :
{
    "MerchandiseExtID__c" : 123
}
}
```

### **Example JSON response**

The successful response is:

```
"id" : "a02D0000006YUHrIAO",
    "errors" : [ ],
    "success" : true,
    "created": true
}
```

The HTTP status code is 201 (Created).



Note: The created parameter is present in the response in API version 46.0 and later. It doesn't appear in earlier versions.

If the external ID value isn't unique, an HTTP status code 300 is returned, plus a list of the records that matched the query. For more information about errors, see Status Codes and Error Responses on page 277.

If the external ID field doesn't exist, an error message and code is returned:

```
"message" : "The requested resource does not exist",
   "errorCode" : "NOT_FOUND"
}
```

You can also use this approach to update existing records. For example, if you created the Line\_Item\_\_c shown in the example above, you can try to update the related Merchandise\_\_c using another request.

### Example for updating a record

```
curl
https://yourInstance.salesforce.com/services/data/v25.0/sobjects/Line_Item__c/LineItemExtID__c/456
  -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @updates.json
  -X PATCH
```

### Example JSON request body updates.json file

This assumes another Merchandise\_\_c record exists with a MerchandiseExtID\_\_c value of 333.

```
{
   "Merchandise__r" :
   {
      "MerchandiseExtID__c" : 333
   }
}
```

#### **Example JSON response**

In API version 46.0 and later, the HTTP status code is 200 (OK) and the successful response is:

```
"id" : "001D000000Kv3g5IAB",
    "success" : true,
    "errors" : [ ],
    "created": false
}
```

In API version 45.0 and earlier, the HTTP status code is 204 (No Content) and there is no response body.

If the relationship type is master-detail and the relationship is set to not allow reparenting, and you try to update the parent external ID, you get an HTTP status code 400 error with an error code of INVALID\_FIELD\_FOR\_INSERT\_UPDATE.

### Traverse Relationships with Friendly URLs

You can traverse relationship fields in objects by constructing friendly URLs via the SObject Relationship resource. This approach allows you to directly access records associated with relationships. Using friendly URLs is easier than accessing records by obtaining object IDs from relationship fields and then inspecting the associated object ID record.

Relationship names follow certain conventions that depend on the direction (parent to child, or child to parent) of the relationship and the name of the related object. The conventions are described in Understanding Relationship Names in the SOQL and SOSL Reference.

There are limits to the number of relationship traversals you can make in a single REST API call. These limits are the same as the limits for SOQL, as described in Understanding Relationship Query Limitations in the SOQL and SOSL Reference. Keep the following limitations in mind when traversing relationships.

• When specifying child-to-parent relationships, no more than five levels can be traversed. The following traverses two child-to-parent relationships.

```
\label{local_prop}  \mbox{https://instance name}. sales force.com/services/data/v47.0/sobjects/ChildOfChild\_c/recording/Child\_r/ParentOfChild\_r
```

 When specifying parent-to-child relationships, no more than one level can be traversed. The following traverses one parent-to-child relationship.

### Example of traversing a simple relationship

This custom object named Merchandise\_\_c contains a lookup relationship field to a child Distributor\_\_c custom object. The following example retrieves the Distributor\_\_c record related to a Merchandise\_\_c record.

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Merchandise_c/a01D0000001NjVe/Distributor_r
-H "Authorization: Bearer token"
```

#### Example request body for traversing a simple relationship

none required

### Example response body for traversing a simple relationship

If no related record is associated with the relationship name, the REST API call fails, because the relationship can't be traversed. Using the previous example, if the Distributor\_\_c field in the Merchandise\_\_c record was set to null, the GET call would return a 404 error response.

You can traverse multiple relationships within the same relationship hierarchy in a single REST API call as long as you don't exceed the relationship query limits. If a Line\_ltem\_\_c custom object is the child in a relationship to a Merchandise\_\_c custom object, and Merchandise\_\_c also has a child Distributor\_\_c custom object, you can access the Distributor\_\_c record starting from the Line\_ltem\_\_c record using something like the following.

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Line_Item_c/a02D0000006YL7XIAW/Merchandise_r/Distributor_r
-H "Authorization: Bearer token"
```

Relationship traversal also supports PATCH and DELETE methods for relationships that resolve to a single record. Using the PATCH method, you can update the related record.

### Example of using PATCH to update a relationship record

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Merchandise_c/a01D000000INjVe/Distributor_r
-H "Authorization: Bearer token" -d @update_info.json -X PATCH
```

### Example JSON request body for updating a relationship record contained in update info.json

```
{
    "Location__c" : "New York"
}
```

### Example response body for updating relationship record

none returned

Finally, using the DELETE method, you can delete the related record.

#### Example using DELETE to delete a relationship record

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Merchandise_c/a01D000000INjVe/Distributor_r
-H "Authorization: Bearer token" -X DELETE
```

### Example request body for deleting a relationship record

none required

### Example response body for update relationship record

none returned

### Traversing Relationships with Multiple Records

You can traverse relationships with multiple records, and get a response that contains the set of records. For relationships that resolve to multiple records, only GET methods are supported.

### Example traversing a relationship with multiple records

If we have a custom object named Merchandise\_\_c that contains a master—detail relationship field to a Line\_Item\_\_c custom object, the following example retrieves the set of Line\_Item\_\_c records related to a Merchandise\_\_c record.

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Merchandise__c/a01D000000INjVe/Line_Items__r
-H "Authorization: Bearer token"
```

#### Example request body for traversing a relationship with multiple records

none required

### Example response body for traversing a relationship with multiple records

For this example, two Line\_Item\_\_c records were retrieved.

```
"CreatedDate": "2011-12-16T17:44:07.000+0000",
        "CreatedById": "005D000001KyEIIA0",
        "LastModifiedDate": "2011-12-16T17:44:07.000+0000",
        "LastModifiedById" : "005D0000001KyEIIA0",
        "SystemModstamp" : "2011-12-16T17:44:07.000+0000",
        "Unit Price c" : 9.75,
        "Units Sold c" : 10.0,
        "Merchandise c": "a00D0000008oLnXIAU",
        "Invoice Statement c": "a01D00000D85hkIAB"
    },
        "attributes" :
            "type" : "Line Item c",
           "url": "/services/data/v36.0/sobjects/Line Item c/a02D0000006YL7YIAW"
        "Id" : "a02D0000006YL7YIAW",
        "IsDeleted" : false,
        "Name" : "LineItem2",
        "CreatedDate": "2011-12-16T18:53:59.000+0000",
        "CreatedById" : "005D000001KyEIIA0",
        "LastModifiedDate": "2011-12-16T18:53:59.000+0000",
        "LastModifiedById" : "005D000001KyEIIA0",
        "SystemModstamp": "2011-12-16T18:54:00.000+0000",
        "Unit Price c" : 8.5,
        "Units Sold c" : 8.0,
        "Merchandise__c" : "a00D0000008oLnXIAU",
        "Invoice Statement c": "a01D00000D85hkIAB"
    }
]
```

The serialized structure for the result data is the same format as result data from executing a SOQL query via REST API. See Query on page 204 for more details on executing SOQL queries via REST API

If no related records are associated with the relationship name, the REST API call returns a 200 response with no record data in the response body. This result is in contrast to the results when traversing an empty relationship to a single record, which returns a 404 error response. This behavior is because the single record case resolves to a REST resource that can be used with PATCH or DELETE methods. In contrast, the multiple record case can only be gueried.

If an initial GET request for a relationship with multiple records returns only part of the results, the end of the response contains the field nextRecordsUrl. For example, you could get a field like the following at the end of your response.

```
"nextRecordsUrl" : "/services/data/v47.0/query/01gD0000002HU6KIAW-2000"
```

You can request the next batch of records using the provided URL with your instance and session information, and repeat until all records have been retrieved. These requests use nextRecordsUrl and don't include any parameters. The final batch of records doesn't have a nextRecordsUrl field.

### Example usage for retrieving the remaining results

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/query/01gD0000002HU6KIAW-2000
-H "Authorization: Bearer token"
```

### Example request body for retrieving the remaining results

none required

### Example response body for retrieving the remaining results

```
"done" : true,
  "totalSize" : 3200,
  "records" : [...]
}
```

### Filtering Result Fields

When retrieving records via relationship traversals, you can optionally specify only a subset of the record fields be returned by using the fields parameter. Multiple fields are separated by commas. The following example retrieves just the Location\_\_c field from the Distributor\_\_c record associated with a Merchandise\_\_c record:

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sdbjects/Merchandise_c/a01D000000INjVe/Distributor_r?fields=Location_c
-H "Authorization: Bearer token"
```

The JSON response data would look like the following:

```
"attributes" :
{
    "type" : "Distributor__c",
    "url" : "/services/data/v36.0/sobjects/Distributor__c/a03D0000003DUhhIAG"
},
    "Location__c" : "Chicago",
}
```

Similarly, for requests that result in multiple records, you can use a list of fields to specify the fields returned in the record set. For example, assume you have a relationship that was associated with two Line\_Item\_\_c records. You want just the Name and Units\_Sold\_\_c fields from those records. You could use the following call.

```
curl
https://yourInstance.salesforce.com/services/data/v36.0/sobjects/Merchandise_c/a01D000000INjve/Distributor_r?fields=Name,Units_Sold_c
-H "Authorization: Bearer token"
```

The response data would look like the following.

If any field listed in the fields parameter set isn't visible to the active user, the REST API call fails. In the previous example, if the Units\_Sold\_c field was hidden from the active user by field-level security, the call would return a 400 error response.

### Get Attachment Content from a Record

Use the SObject Blob Retrieve resource to retrieve blob data for a given record.

The following example retrieves the blob data for an Attachment record. The Attachment can be associated with a Case, Campaign, or other object that allows attachments.

### Example for retrieving blob body for an Attachment record

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Attachment/001D000000INjVe/body
-H "Authorization: Bearer token"
```

### **Example request body**

none required

#### Example response body

Attachment body content is returned in binary form. Note that the response content type will not be JSON or XML since the returned data is binary.

The following example retrieves the blob data for a Document record.

#### Example for retrieving blob body for a Document record

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Document/015D0000000NdJOIA0/body
-H "Authorization: Bearer token"
```

#### **Example request body**

none required

#### **Example response body**

Document body content is returned in binary form. Note that the response content type will not be JSON or XML since the returned data is binary.

### Get a List of Deleted Records Within a Given Timeframe

Use the SObject Get Deleted resource to get a list of deleted records for the specified object. Specify the date and time range within which the records for the given object were deleted. Deleted records are written to a delete log (that is periodically purged), and will be filtered out of most operations, such as SObject Rows or Query (although QueryAll will include deleted records in results).

### Example usage for getting a list of Merchandise\_\_c records that were deleted between May 5th, 2013 and May 10th, 2013

```
/services/data/v29.0/sobjects/Merchandise__c/deleted/
?start=2013-05-05T00%3A00%3A00%2B00%3A00&end=2013-05-10T00%3A00%3A00%2B00%3A00
```

### **Example request body**

None required

#### JSON example response body

### XML example response body

### Get a List of Updated Records Within a Given Timeframe

Use the SObject Get Updated resource to get a list of updated (modified or added) records for the specified object. Specify the date and time range within which the records for the given object were updated.

### Example usage for getting a list of Merchandise\_c records that were updated between May 6th, 2013 and May 10th, 2013

```
/services/data/v29.0/sobjects/Merchandise__c/updated/
?start=2013-05-06T00%3A00%3A00%2B00%3A00&end=2013-05-10T00%3A00%3A00%2B00%3A00
```

#### **Example request body**

None required

### JSON example response body

```
"ids" :
[
    "a00D0000008pQR5IAM",
    "a00D0000008pQRGIA2",
    "a00D0000008pQRFIA2"
],
```

```
"latestDateCovered" : "2013-05-08T21:20:00.000+0000" }
```

### XML example response body

### Delete Lightning Experience Event Series

Use the HTTP DELETE method to remove one or more IsRecurrence2 events in a series. You can remove a single event, all events following and including a specific event, or an entire event series.

### Deleting a Single Event in a Series

Use the SObject Rows on page 116 resource to delete event records. To delete a single occurrence in a series, specify the event ID and use the DELETE method on page 47 of the resource to delete a record.

### Delete from an Event Onwards

To delete all events in a series from a specific occurrence and onwards, specify the event ID and use the DELETE method of the resource to delete a record. When calling this method, IsRecurrence2 must be true and IsRecurrence2Exclusion must be false.

### Example for deleting from an event onwards

```
\label{lem:curlhttps://www.nstance.salesforce.com/services/data/v45.0/sobjects/Event/00Uxx0000000000from \cite{thisEventOnwards-H}.
```

#### **Example request body**

```
None needed
```

#### Example response body after successfully deleting events from the series

```
{
success: We're deleting the selected events from the series. Wait for all events to be
removed.
}
```

### Delete All Events in a Series

To delete an entire event series, specify the event ID of the first occurrence in the series and use the DELETE method of the resource to delete a record.

#### Example for deleting an entire series

```
 \hbox{curl https://} {\it your Instance}. \hbox{sales force.com/services/data/v45.0/sobjects/Event/00Uxx00000000000from II his Event Onwards \\ - \hbox{H "Authorization: Bearer } {\it token} - \hbox{X DELETE}
```

### **Example request body**

```
None needed
```

### Example response body after successfully deleting events from the series

```
{
success: We're deleting the selected events from the series. Wait for all events to be
removed.
}
```

### Considerations

Delete from an Event Onwards does not support calls from events that:

- Occurred before the original value of Recurrence2PatternStartDate.
- Are defined as child events.

If the event series originated outside of Salesforce and the event ID of the first occurrence is unavailable, you can't delete all events in a series. Instead, delete events from a specific occurrence onwards.

# Working with Searches and Queries

The examples in this section use REST API resources to search and query records using Salesforce Object Search Language (SOSL) and Salesforce Object Query Language (SOQL), and other search APIs. For more information on SOSL and SOQL see the SOQL and SOSL Reference.

#### IN THIS SECTION:

### Execute a SOQL Query

Use the Query resource to execute a SOQL query that returns all the results in a single response, or if needed, returns part of the results and an identifier used to retrieve the remaining results.

#### Execute a SOQL Query that Includes Deleted Items

Use the QueryAll resource to execute a SOQL query that includes information about records that have been deleted because of a merge or delete. Use QueryAll rather than Query, because the Query resource will automatically filter out items that have been deleted.

#### Get Feedback on Query Performance

Use the Query resource along with the explain parameter to get feedback on how Salesforce executes your query, report, or list view. Salesforce analyzes each query to find the optimal approach to obtain the query results. Depending on the query and query filters, Salesforce uses an index or internal optimization. Use the explain parameter to return details on how Salesforce optimizes your query, without actually running the query. Based on the response, you can decide whether to fine-tune the performance of your query by making changes like adding filters to make the query more selective.

#### Search for a String

Use the Search resource to execute a SOSL search or use the Parameterized Search resource to execute a simple RESTful search without SOSL.

Examples Execute a SOQL Query

### Get the Default Search Scope and Order

Use the Search Scope and Order resource to retrieve the default global search scope and order for the logged-in user, including any pinned objects in the user's search results page.

### Get Search Result Layouts for Objects

Use the Search Result Layouts resource to retrieve the search result layout configuration for each object specified in the query string.

### View Relevant Items

Use the Relevant Items resource to get a list of relevant records.

### **Execute a SOQL Query**

Use the Query resource to execute a SOQL query that returns all the results in a single response, or if needed, returns part of the results and an identifier used to retrieve the remaining results.

The following query requests the value from name fields from all Account records.

### Example usage for executing a query

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/query/?q=SELECT+name+from+Account
-H "Authorization: Bearer token"
```

### Example request body for executing a query

none required

### Example response body for executing a query

```
"done" : true,
    "totalSize" : 14,
    "records" :
        {
            "attributes" :
                "type" : "Account",
                "url" : "/services/data/v20.0/sobjects/Account/001D000000IRFmaIAH"
            },
            "Name" : "Test 1"
        },
            "attributes" :
                "type" : "Account",
                 "url": "/services/data/v20.0/sobjects/Account/001D000000IomazIAB"
            "Name" : "Test 2"
        },
        . . .
    ]
}
```

### Retrieving the Remaining SOQL Query Results

If the initial query returns only part of the results, the end of the response will contain a field called nextRecordsUrl. For example, you might find this attribute at the end of your query:

```
"nextRecordsUrl" : "/services/data/v20.0/query/01gD0000002HU6KIAW-2000"
```

In such cases, request the next batch of records and repeat until all records have been retrieved. These requests use nextRecordsUrl, and do not include any parameters.

### Example usage for retrieving the remaining query results

```
curl
https://yourInstance.salesforce.com/services/data/v20.0/query/01gD0000002HU6KIAW-2000
-H "Authorization: Bearer token"
```

### Example request body for retrieving the remaining query results

none required

### Example response body for retrieving the remaining query results

```
"done" : true,
  "totalSize" : 3214,
  "records" : [...]
}
```

### Execute a SOQL Query that Includes Deleted Items

Use the QueryAll resource to execute a SOQL query that includes information about records that have been deleted because of a merge or delete. Use QueryAll rather than Query, because the Query resource will automatically filter out items that have been deleted.

The following query requests the value from the Name field from all deleted Merchandise\_\_c records, in an organization that has one deleted Merchandise\_\_c record. The same query using Query instead of QueryAll would return no records, because Query automatically filters out any deleted record from the result set.

### Example usage for executing a query for deleted Merchandise\_\_c records

```
/services/data/v29.0/queryAll/?q=SELECT+Name+from+Merchandise c+WHERE+isDeleted+=+TRUE
```

### Example request body for executing a query

none required

### Example response body for executing a query

```
"Name" : "Merchandise 1"
},
]
```

### Retrieving the Remaining SOQL Query Results

If the initial query returns only part of the results, the end of the response will contain a field called nextRecordsUrl. For example, you might find this attribute at the end of your query:

```
"nextRecordsUrl" : "/services/data/v29.0/query/01gD0000002HU6KIAW-2000"
```

In such cases, request the next batch of records and repeat until all records have been retrieved. These requests use nextRecordsUrl, and do not include any parameters.

Note that even though nextRecordsUrl has query in the URL, it will still provide remaining results from the initial QueryAll request. The remaining results will include deleted records that matched the initial query.

### Example usage for retrieving the remaining results

```
/services/data/v29.0/query/01gD0000002HU6KIAW-2000
```

### Example request body for retrieving the remaining results

none required

### Example response body for retrieving the remaining results

```
"done" : true,
  "totalSize" : 3214,
  "records" : [...]
}
```

# Get Feedback on Query Performance

Use the Query resource along with the explain parameter to get feedback on how Salesforce executes your query, report, or list view. Salesforce analyzes each query to find the optimal approach to obtain the query results. Depending on the query and query filters, Salesforce uses an index or internal optimization. Use the explain parameter to return details on how Salesforce optimizes your query, without actually running the query. Based on the response, you can decide whether to fine-tune the performance of your query by making changes like adding filters to make the query more selective.



**Note:** Using explain with the REST API query resource is a beta feature. There is no support associated with this beta feature. For more information, contact Salesforce.

The response contains one or more query execution plans that, sorted from most optimal to least optimal. The most optimal plan is the plan that's used when the query, report, or list view is executed.

See the explain parameter in Query for more details on the fields returned when using explain. See Working with Very Large SOQL Queries in the Apex Developer Guide for more information on making your queries more selective.



Example:

### Example usage for getting performance feedback on a query that uses Merchandise\_c

```
/services/data/v47.0/query/?explain=
SELECT+Name+FROM+Merchandise__c+WHERE+CreatedDate+=+TODAY+AND+Price__c+>+10.0
```

# Example response body for executing a performance feedback query

```
"plans" : [ {
   "cardinality" : 1,
   "fields" : [ "CreatedDate" ],
   "leadingOperationType" : "Index",
   "notes" : [ {
      "description": "Not considering filter for optimization because unindexed",
     "fields" : [ "IsDeleted" ],
     "tableEnumOrId" : "Merchandise c"
   } ],
   "relativeCost" : 0.0,
   "sobjectCardinality" : 3,
   "sobjectType" : "Merchandise c"
 }, {
   "cardinality" : 1,
   "fields" : [ ],
   "leadingOperationType" : "TableScan",
   "notes" : [ {
     "description": "Not considering filter for optimization because unindexed",
     "fields" : [ "IsDeleted" ],
     "tableEnumOrId" : "Merchandise c"
   } ],
   "relativeCost" : 0.65,
   "sobjectCardinality" : 3,
   "sobjectType" : "Merchandise c"
 } ]
}
```

This response indicates that Salesforce found two possible execution plans for this query. The first plan uses the CreatedDate index field to improve the performance of this query. The second plan scans all records without using an index. The first plan is used if the query is executed. Both plans note that a secondary optimization used when filtering out records marked as deleted was not used because the IsDeleted field is not indexed.

# Example:

#### Example usage for getting performance feedback on a report

```
/services/data/v47.0/query/?explain=000D000001hCzMMCU
```

# Example response body for getting performance feedback on a report

```
"plans" : [ {
    "cardinality" : 1,
    "fields" : [ ],
    "leadingOperationType" : "TableScan",
    "notes" : [ {
        "description" : "Not considering filter for optimization because unindexed",
        "fields" : [ "IsDeleted" ],
        "tableEnumOrId" : "Merchandise__c"
        } ],
        "relativeCost" : 0.65,
        "sobjectCardinality" : 3,
```

Examples Search for a String

```
"sobjectType" : "Merchandise__c"
} ]
}
```

This response indicates that Salesforce found one possible execution plan for the query behind this report. The plan scans all records without using an index. It can't apply a secondary optimization when filtering out records marked as deleted, because the IsDeleted field is not indexed.

# Search for a String

Use the Search resource to execute a SOSL search or use the Parameterized Search resource to execute a simple RESTful search without SOSL.

# Example SOSL Search Using the GET Method

The following example executes a SOSL search for Acme. The search string in this example must be URL-encoded.

#### Example usage

```
curl https://yourInstance.salesforce.com/services/data/v37.0/search/?q=FIND+%7BAcme%7D
-H "Authorization: Bearer token"
```

## **Example request body**

None required

#### **Example response body**

```
"searchRecords" : [ {
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v35.0/sobjects/Account/001D000000IqhSLIAZ"
},
    "Id" : "001D000000IqhSLIAZ",
}, {
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v35.0/sobjects/Account/001D000000IomazIAB"
    },
    "Id" : "001D000000IomazIAB",
} ]
}
```

# Example Parameterized Search Using the GET Method

The following example executes a parameterized search for Acme. The search string in this example must be URL-encoded.

#### **Example usages**

#### Global search for all results containing Acme

curl https://yourInstance.salesforce.com/services/data/v37.0/parameterizedSearch/?q=Acme

Examples Search for a String

## Account search for results containing Acme, returning the id and name fields

curl
https://**yar/instance**.salesforce.com/services/data/v37.0/parameterizedSearch/?q=Acress/object=Accants/Accant.fields=id\_names/Accant.limit=10

# **Example request body**

None required

#### **Example response body**

```
"searchRecords" : [ {
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v35.0/sobjects/Account/001D000000IqhSLIAZ"
    },
    "Id" : "001D000000IqhSLIAZ"
}, {
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v35.0/sobjects/Account/001D000000IomazIAB"
    },
    "Id" : "001D000000IomazIAB"
    }
}
```

## Example response body with metadata parameter

Mote: The metadata parameter is only returned if the request specified metadata=LABELS.

```
"searchRecords" : [ {
   "attributes" : {
     "type" : "Account",
      "url" : "/services/data/v35.0/sobjects/Account/001D000000IqhSLIAZ"
    },
    "Id" : "001D000000IqhSLIAZ",
  }, {
    "attributes" : {
      "type" : "Account",
      "url" : "/services/data/v35.0/sobjects/Account/001D000000IomazIAB"
    "Id" : "001D000000IomazIAB",
  } ],
  "metadata" : {
    "entityetadata" : [ {
      "entityName" : "Account",
      "fieldMetadata" : [ {
        "name" : "Name",
        "label" : "Account Name"
       } ]
   } ]
  }
}
```

# Example Parameterized Search Using the POST Method

Execute a parameterized search using the POST method to access all search features available.

#### Example usage

```
curl https://yourInstance.salesforce.com/services/data/v36.0/parameterizedSearch "Authorization: Bearer token-H "Content-Type: application/json" -d "@search.json"
```

# **Example request body**

None required

#### **Example JSON file**

#### **Example response body**

```
"searchRecords" : [ {
    "attributes" : {
        "type" : "Contact",
        "url" : "/services/data/v36.0/sobjects/Contact/003xx000004TraiAAC"
    },
    "Id" : "003xx000004TraiAAC",
    "FirstName" : "Smith",
    "LastName" : "Johnson"
    }, {
        "attributes" : {
            "type" : "Account",
            "url" : "/services/data/v36.0/sobjects/Account/001xx000003DHXnAAO"
    },
    "Id" : "001xx000003DHXnAAO",
    "NumberOfEmployees" : 100
    } ]
}
```

# Get the Default Search Scope and Order

Use the Search Scope and Order resource to retrieve the default global search scope and order for the logged-in user, including any pinned objects in the user's search results page.

In the following example, the default global search scope of the logged-in user consists of the site, idea, case, opportunity, account, and user objects, in the order in which they are returned in the response body.

## **Example usage**

```
curl https://yourInstance.salesforce.com/services/data/v26.0/search/scopeOrder -H "Authorization: Bearer token"
```

## **Example request body**

none required

#### **Example response body**

```
"type": "Site",
  "url": "/services/data/v26.0/sobjects/Site/describe"
},
  "type": "Idea",
  "url":"/services/data/v26.0/sobjects/Idea/describe"
},
  "type": "Case",
  "url":"/services/data/v26.0/sobjects/Case/describe"
},
  "type": "Opportunity",
  "url":"/services/data/v26.0/sobjects/Opportunity/describe"
},
  "type": "Account",
 "url": "/services/data/v26.0/sobjects/Account/describe"
},
  "type": "User",
  "url":"/services/data/v26.0/sobjects/User/describe"
```

# Get Search Result Layouts for Objects

Use the Search Result Layouts resource to retrieve the search result layout configuration for each object specified in the query string.

# **Example usage**

```
curl
https://yourInstance.salesforce.com/services/data/v28.0/search/layout/?q=Account,Contact,Lead,Asset
"Authorization: Bearer token"
```

## **Example request body**

None required

#### **Example response body**

```
"label" : "Account Name",
        "name" : "Name"
      },
      { "field" : "Account.Site",
        "format" : null,
        "label" : "Account Site",
        "name" : "Site"
      { "field" : "Account.Phone",
        "format" : null,
        "label" : "Phone",
        "name" : "Phone"
      } ,
      { "field" : "User.Alias",
       "format" : null,
        "label" : "Account Owner Alias",
        "name" : "Owner.Alias"
    1
},
{ "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Contact.Name",
       "format" : null,
        "label" : "Name",
        "name" : "Name"
      },
      { "field" : "Account.Name",
        "format" : null,
        "label" : "Account Name",
        "name" : "Account.Name"
      },
      { "field" : "Account.Site",
        "format" : null,
        "label" : "Account Site",
        "name" : "Account.Site"
      },
      { "field" : "Contact.Phone",
       "format" : null,
        "label" : "Phone",
        "name" : "Phone"
      },
      { "field" : "Contact.Email",
        "format" : null,
        "label" : "Email",
        "name" : "Email"
      } ,
      { "field" : "User.Alias",
       "format" : null,
        "label" : "Contact Owner Alias",
        "name" : "Owner.Alias"
      }
   ]
},
```

Examples View Relevant Items

```
{ "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Lead.Name",
        "format" : null,
        "label" : "Name",
        "name" : "Name"
      },
      { "field" : "Lead.Title",
        "format" : null,
        "label" : "Title",
        "name" : "Title"
      },
      { "field" : "Lead.Phone",
        "format" : null,
        "label" : "Phone",
        "name" : "Phone"
      { "field" : "Lead.Company",
        "format" : null,
        "label" : "Company",
        "name" : "Company"
      },
      { "field" : "Lead.Email",
        "format" : null,
        "label" : "Email",
        "name" : "Email"
      },
      { "field" : "Lead.Status",
        "format" : null,
        "label" : "Lead Status",
        "name" : "toLabel(Status)"
      },
      { "field" : "Name.Alias",
        "format" : null,
        "label" : "Owner Alias",
        "name" : "Owner.Alias"
      }
    ]
},
```

# View Relevant Items

Use the Relevant Items resource to get a list of relevant records.

## Example usage for getting a list of the current user's most relevant records

/vXX.X/sobjects/relevantItems

## **Example request body**

None required

#### **Example response body**

```
[ { "apiName" : "Contact",
```

Examples View Relevant Items

```
"key" : "003",
    "label" : "Contacts",
    "lastUpdatedId" : "135866748",
   "recordIds" : [ "003xx000004TxBA" ]
}, { "apiName" : "Account",
   "key" : "001",
    "label" : "Accounts",
    "lastUpdatedId" : "193640553",
    "recordIds" : [ "001xx000003DWsT" ]
}, {
    "apiName" : "User",
   "key" : "005",
    "label" : "Users",
    "lastUpdatedId" : "-199920321",
   "recordIds" : [ "005xx000001Svqw", "005xx000001SvwK", "005xx000001SvwA" ]
}, { "apiName" : "Case",
   "key" : "069",
    "label" : "Cases",
    "lastUpdatedId" : "1033471693",
   "recordIds": [ "069xx0000000006", "069xx000000001", "069xx000000002" ]
} ]
```

# Example usage for filtering the response to certain objects

/v37.0/sobjects/relevantItems?sobjects=Account,User

# **Example request body**

None required

#### **Example response body**

```
[ {
    "apiName" : "Account",
    "key" : "001",
    "label" : "Accounts",
    "lastUpdatedId" : "193640553",
    "recordIds" : [ "001xx000003DWsT" ]
}, {
    "apiName" : "User",
    "key" : "005",
    "label" : "Users",
    "lastUpdatedId" : "102959935",
    "recordIds" : [ "005xx000001Svqw", "005xx000001SvwK", "005xx000001SvwA" ]
} ]
```

## Example usage for comparing the user's current list of relevant records to a previous version

/v37.0/sobjects/relevantItems?lastUpdatedId=102959935

# **Example request body**

None required

# Example response header

```
lastUpdatedId: 102959935
newResultSetSinceLastQuery: true
```

## **Example response body**

```
"apiName" : "User",
    "key" : "003",
    "label" : "Users",
    "lastUpdatedId" : "102959935",
    "recordIds" : [ "003xx000004TxBA" ]
   "apiName" : "Account",
    "key" : "001",
    "label" : "Accounts",
    "lastUpdatedId" : "193640553",
    "recordIds" : [ "001xx000003DWsT" ]
}, {
    "apiName" : "Case",
   "key" : "005",
   "label" : "Cases",
  "lastUpdatedId" : "1740766611",
    "recordIds" : [ "005xx000001Svqw", "005xx000001SvwA" ]
} ]
```

### Example usage for comparing the user's current list of relevant records to a previous version for a particular object

/v37.0/sobjects/relevantItems?mode=MRU&sobjects=Account,Contact&Account.lastUpdatedId=102959935

#### **Example request body**

None required

#### **Example response body**

```
[ {
    "apiName" : "Account",
    "key" : "001",
    "label" : "Accounts",
    "lastUpdatedId" : "193640553",
    "recordIds" : [ "001xx000003DWsT" ]
} ]
```

# Insert or Update Blob Data

You can use SObject Basic Information, SObject Rows, or SObject Collections REST resources to insert or update blob data in Salesforce standard objects. You can upload files of any type, and you must use a multipart message that conforms to the MIME multipart content-type standard. For more information, see the WC3 Standards. You can insert or update files on any standard object that contains a blob field.

Using the SObject Basic Information or SObject Rows APIs, the maximum file size for uploads is 2 GB for ContentVersion objects and 500 MB for all other eligible standard objects. Using the SObject Collections API, the maximum total size of all files in a single request is 500 MB



Note: You can insert or update blob data using a non-multipart message, but you are limited to 50 MB of text data or 37.5 MB of base64–encoded data.

The first part of the request message body contains non-binary field data, such as the Description or Name. The second part of the message contains the binary data of the file that you're uploading.

The following sections provide JSON examples of how to insert or update blob data using a multipart content-type.

- Inserting a New Document
- Updating a Document
- Inserting a ContentVersion
- Using SObject Collections to Insert a Collection of Blob Records
- Multipart Message Considerations

# Inserting a New Document

This syntax and code creates a new Document. In addition to the binary data of the file itself, this code also specifies other field data such as the Description, Keywords, and Name.



 $\bigcap$  Tip: After you add a new Document, you can view the results of your changes on the Documents tab.

## **Example for creating a new Document**

```
curl https://yourInstance.salesforce.com/services/data/v23.0/sobjects/Document/ -H
"Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary string\"" --data-binary @newdocument.json
```

#### Example request body for creating a new Document

This code is the contents of newdocument. json. The binary data for the PDF content has been omitted for brevity and replaced with "Binary data goes here." An actual request contains the full binary content.

```
--boundary string
Content-Disposition: form-data; name="entity document";
Content-Type: application/json
    "Description" : "Marketing brochure for Q1 2011",
    "Keywords" : "marketing, sales, update",
    "FolderId" : "005D0000001GiU7",
    "Name" : "Marketing Brochure Q1",
    "Type" : "pdf"
--boundary_string
Content-Type: application/pdf
Content-Disposition: form-data; name="Body"; filename="2011Q1MktqBrochure.pdf"
Binary data goes here.
--boundary string--
```

#### Example response body for creating a new Document

On success, the ID of the new Document is returned.

```
"id" : "015D0000000N3ZZIA0",
    "errors" : [ ],
    "success" : true
}
```

#### **Example error response**

```
"fields" : [ "FolderId" ],
    "message" : "Folder ID: id value of incorrect type: 005D0000001GiU7",
    "errorCode" : "MALFORMED_ID"
}
```

# **Updating a Document**

This syntax and code updates an existing Document. In addition to the binary data of the file itself, this code also updates other field data, such as the Name and Keywords.

# Example usage for updating fields in a Document object

```
curl https://yourInstance.salesforce.com/services/data/v23.0/Document/015D0000000N3ZZIA0
  -H "Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary_string\"" --data-binary @UpdateDocument.json -X PATCH
```

## Example request body for updating fields in a Document object

This code is the contents of the file UpdateDocument.json. The binary data for the PDF content has been omitted for brevity and replaced with "Updated document binary goes here." An actual request contains the full binary content.

```
--boundary_string
Content-Disposition: form-data; name="entity_content";
Content-Type: application/json

{
    "Name" : "Marketing Brochure Q1 - Sales",
    "Keywords" : "sales, marketing, first quarter"
}

--boundary_string
Content-Type: application/pdf
Content-Disposition: form-data; name="Body"; filename="2011Q1MktgBrochure.pdf"

Updated document binary data goes here.

--boundary_string--
```

#### Example response body for updating fields in a Document object

None returned

#### **Error responses**

See Status Codes and Error Responses on page 277.

# Inserting a ContentVersion

This syntax and code inserts a new ContentVersion. In addition to the binary data of the file itself, this code also updates other fields, such as the ReasonForChange and PathOnClient. This message contains the ContentDocumentId because a ContentVersion is always associated with a ContentDocument.



Tip: The ContentVersion object doesn't support updates. Therefore, you cannot update a ContentVersion. You can only insert a new ContentVersion. You can see the results of your changes on the Content tab.

## Example usage for inserting a ContentVersion

```
curl https://yourInstance.salesforce.com/services/data/v23.0/sobjects/ContentVersion
-H "Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary_string\"" --data-binary @NewContentVersion.json
```

### Example request body for inserting a ContentVersion

This code is the contents of the file NewContentVersion.json. The binary data for the PDF content has been omitted for brevity and replaced with "Binary data goes here." An actual request contains the full binary content.

```
--boundary_string
Content-Disposition: form-data; name="entity_content";
Content-Type: application/json

{
    "ContentDocumentId" : "069D00000000002",
    "ReasonForChange" : "Marketing materials updated",
    "PathOnClient" : "Q1 Sales Brochure.pdf"
}

--boundary_string
Content-Type: application/octet-stream
Content-Disposition: form-data; name="VersionData"; filename="Q1 Sales Brochure.pdf"

Binary data goes here.

--boundary_string--
```

## Example response body for inserting a ContentVersion

```
"id" : "068D0000000pg0IAQ",
    "errors" : [ ],
    "success" : true
}
```

# Error responses for inserting a ContentVersion

See Status Codes and Error Responses on page 277.

# Using SObject Collections to Insert a Collection of Blob Records

This syntax and code inserts a collection of new Documents. In addition to the binary data of the files themselves, this code also specifies other field data, such as the Description and Name for each record in the collection.



Tip: After you add new Documents, you can view the results of your changes on the Documents tab.

#### **Attributes**

If you're using sObject Collections with blob data, you must specify certain attribute values in addition to type in the request body's attributes map.

Parameter	Description
binaryPartName	Required for blob data. A unique identifier for the binary part.

Parameter	Description
binaryPartNameAlias	Required for blob data. The name of the field in which the binary data is inserted or updated.

#### **Example for creating new Documents**

```
curl https://yourInstance.salesforce.com/services/data/v42.0/composite/sobjects/ -H
"Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary_string\"" --data-binary @newdocuments.json
```

## **Example request body for creating new Documents**

This code is the contents of newdocuments.json. The binary data for the PDF content has been omitted for brevity and replaced with "Binary data goes here." An actual request contains the full binary content.

```
--boundary string
Content-Disposition: form-data; name="collection"
Content-Type: application/json
    "allOrNone" : false,
    "records" :
        {
            "attributes" :
                "type" : "Document",
                "binaryPartName": "binaryPart1",
                "binaryPartNameAlias": "Body"
            },
            "Description" : "Marketing Brochure",
            "FolderId" : "005xx000001Svs4AAC",
            "Name" : "Brochure",
            "Type" : "pdf"
        },
            "attributes" :
                "type" : "Document",
                "binaryPartName": "binaryPart2",
                "binaryPartNameAlias": "Body"
            },
            "Description" : "Pricing Overview",
            "FolderId" : "005xx000001Svs4AAC",
            "Name" : "Pricing",
            "Type" : "pdf"
        }
    ]
--boundary string
Content-Disposition: form-data; name="binaryPart1"; filename="Brochure.pdf"
Content-Type: application/pdf
```

```
Binary data goes here.

--boundary_string
Content-Disposition: form-data; name="binaryPart2"; filename="Pricing.pdf"
Content-Type: application/pdf

Binary data goes here.

--boundary_string--
```

### **Example response body for creating new Documents**

On success, the IDs of the new Documents are returned.

```
[
    "id": "015xx00000013QjAAI",
    "errors": [],
    "success": true
},
{
    "id": "015xx00000013QkAAI",
    "errors": [],
    "success": true
}
```

For more information, see SObject Collections.

# **Multipart Message Considerations**

Following are some considerations for the format of a multipart message when you insert or update blob data.

#### **Boundary String**

- Separates the various parts of a multipart message.
- Required in a multipart content-type.
- Can be up to 70 characters.
- Cannot be a string value that appears anywhere in any of the message parts.
- The first boundary string must be prefixed by two hyphens (--).
- The last boundary string must be postfixed by two hyphens (--).

## **Content-Disposition Header**

- Required in each message part.
- Must be the value form-data and have a name attribute.
  - In the non-binary part of the message, the name attribute can be any value.
  - For single documents, in the binary part of the message, use the name attribute to contain the name of the object field
    that contains the binary data. In the previous example of adding a new Document, the name of the binary field that contains
    the file is Body.

- For documents inserted or updated using sObject Collections, use the name attribute to contain a unique identifier for the part. This identifier is referenced by the non-binary part of the message.
- The binary part of the message must have a filename attribute that represents the name of the local file.

#### **Content-Type Header**

- Required in each message part.
- The content types supported by the non-binary message part are application/json and application/xml.
- The Content-Type header for the binary part of the message can be any value.

#### **New Line**

A new line must be between the message part header and the data of the part. As shown in the code examples, a new line must be between the Content-Type and Content-Disposition headers and the JSON or XML. In the binary part, a new line must be between the Content-Type and Content-Disposition headers and the binary data.

# Working with Recently Viewed Information

The examples in this section use REST API Query and Recently Viewed resources to programmatically retrieve and update recently viewed record information.

#### IN THIS SECTION:

### View Recently Viewed Records

Use the Recently Viewed Items resource to get a list of recently viewed records.

#### Mark Records as Recently Viewed

To mark a record as recently viewed using REST API, use the Query resource with a FOR VIEW or FOR REFERENCE clause. Use SOQL to mark records as recently viewed to ensure that information such as the date and time the record was viewed is correctly set.

# View Recently Viewed Records

Use the Recently Viewed Items resource to get a list of recently viewed records.

# Example usage for getting the last two most recently viewed records

```
/services/data/v28.0/recent/?limit=2
```

# **Example request body**

none required

#### Example response body

```
{
    "attributes" :
    {
        "type" : "Account",
        "url" : "/services/data/v28.0/sobjects/Account/a06U000000CelH0IAJ"
    },
    "Id" : "a06U000000CelH0IAJ",
    "Name" : "Acme"
},
{
```

```
"attributes" :
{
    "type" : "Opportunity",
    "url" : "/services/data/v28.0/sobjects/Opportunity/a06U000000CelGvIAJ"
},
"Id" : "a06U000000CelGvIAJ",
"Name" : "Acme - 600 Widgets"
}
```

# Mark Records as Recently Viewed

To mark a record as recently viewed using REST API, use the Query resource with a FOR VIEW or FOR REFERENCE clause. Use SOQL to mark records as recently viewed to ensure that information such as the date and time the record was viewed is correctly set.

Use FOR VIEW to notify Salesforce when a record is viewed from a custom interface, such as a mobile application or from a custom page. Use FOR REFERENCE when a record is referenced from a custom interface. A record is referenced every time a related record is viewed. For more information, see "FOR VIEW" and "FOR REFERENCE" in the SOQL and SOSL Reference.

## Example usage for executing a query that marks one Account record as recently viewed

```
/services/data/v28.0/query/?q=SELECT+Name+FROM+Account+LIMIT+1+FOR+VIEW
```

#### Example request body for executing a query

none required

## Example response body for executing a query

# Managing User Passwords

The examples in this section use REST API resources to manage user passwords, such as setting or resetting passwords.

#### IN THIS SECTION:

#### Manage User Passwords

Use the SObject User Password resource to set, reset, or get information about a user password. Use the HTTP GET method to get password expiration status, the HTTP POST method to set the password, and the HTTP DELETE method to reset the password.

Examples Manage User Passwords

# Manage User Passwords

Use the SObject User Password resource to set, reset, or get information about a user password. Use the HTTP GET method to get password expiration status, the HTTP POST method to set the password, and the HTTP DELETE method to reset the password.

The associated session must have permission to access the given user password information. If the session does not have proper permissions, an HTTP error 403 response is returned from these methods.

These methods are available for both users and self-service users. For managing self-service user passwords, use SelfServiceUser instead of User in the REST API URL.

Here is an example of retrieving the current password expiration status for a user:

#### Example usage for getting current password expiration status

```
curl
https://yourInstance.salesforce.com/services/data/v25.0/sobjects/User/005D0000001KyEIIA0/password
-H "Authorization: Bearer token"
```

# Example request body for getting current password expiration status

None required

#### JSON example response body for getting current password expiration status

```
{
    "isExpired" : false
}
```

### XML example response body for getting current password expiration status

#### Example error response if session has insufficient privileges

```
{
   "message" : "You do not have permission to view this record.",
   "errorCode" : "INSUFFICIENT_ACCESS"
}
```

Here is an example of changing the password for a given user:

#### Example usage for changing a user password

```
curl
https://yourInstance.salesforce.com/services/data/v25.0/sobjects/User/005D0000001KyEIIA0/password
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newpwd.json
-X POST
```

#### Contents for file newpwd.json

```
{
    "NewPassword" : "myNewPassword1234"
}
```

## Example response for changing a user password

No response body on successful password change, HTTP status code 204 returned.

## Example error response if new password does not meet organization password requirements

```
"message" : "Your password must have a mix of letters and numbers.",
"errorCode" : "INVALID_NEW_PASSWORD"
}
```

And finally, here is an example of resetting a user password:

# Example usage for resetting a user password

# Example request body for resetting a user password

None required

#### JSON example response body for resetting a user password

```
{
    "NewPassword" : "2sv0xHAuM"
}
```

# XML example response body for resetting a user password

# Working with Approval Processes and Process Rules

The examples in this section use REST API resources to work with approval processes and process rules.

#### IN THIS SECTION:

## Get a List of All Approval Processes

Use the Process Approvals resource to get information about approvals.

#### Submit a Record for Approval

Use the Process Approvals resource to submit a record or a collection of records for approval. Each call takes an array of requests.

# Approve a Record

Use the Process Approvals resource to approve a record or a collection of records. Each call takes an array of requests. The current user must be an assigned approver.

#### Reject a Record

Use the Process Approvals resource to reject a record or a collection of records. Each call takes an array of requests. The current user must be an assigned approver.

# **Bulk Approvals**

Use the Process Approvals resource to do bulk approvals. You can specify a collection of different Process Approvals requests to have them all executed in bulk.

#### Get a List of Process Rules

Use the Process Rules resource to get information about process rules.

#### Get a Particular Process Rule

Use the Process Rules resource and specify the SObjectName and workflowRuleId of the rule you want to get the metadata for

### **Trigger Process Rules**

Use the Process Rules resource to trigger process rules. All rules associated with the specified ID will be evaluated, regardless of the evaluation criteria. All IDs must be for records on the same object.

# Get a List of All Approval Processes

Use the Process Approvals resource to get information about approvals.

# **Example usage**

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/approvals/ -H "Authorization: Bearer token"
```

# **Example request body**

none required

# **Example JSON response body**

```
{
  "approvals" : {
    "Account" : [ {
        "description" : null,
        "id" : "04aD00000008Py9",
        "name" : "Account Approval Process",
        "object" : "Account",
        "sortOrder" : 1
    } ]
}
```

# Submit a Record for Approval

Use the Process Approvals resource to submit a record or a collection of records for approval. Each call takes an array of requests.

#### Example usage

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/approvals/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @submit.json"
```

# Example request body submit.json file

In the following example, the record "001D00000018mlm" is submitted for approval process "PTO\_Request\_Process" by skipping its entry criteria on behalf of submitter "005D00000015rZy."

```
"requests" : [{
  "actionType": "Submit",
  "contextId": "001D00000018mIm",
  "nextApproverIds": ["005D00000015rY9"],
  "comments": "this is a test",
  "contextActorId": "005D000000015rZy",
```

Examples Approve a Record

```
"processDefinitionNameOrId" : "PTO_Request_Process",
"skipEntryCriteria": "true"}]
}
```

#### **Example JSON response body**

```
[ {
  "actorIds" : [ "005D00000015rY9IAI" ],
  "entityId" : "001D00000018mImIAJ",
  "errors" : null,
  "instanceId" : "04gD0000000cvm5IAC",
  "instanceStatus" : "Pending",
  "newWorkitemIds" : [ "04iD0000000Cw6SIAS" ],
  "success" : true } ]
```

# Approve a Record

Use the Process Approvals resource to approve a record or a collection of records. Each call takes an array of requests. The current user must be an assigned approver.

#### Example usage

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/approvals/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @approve.json"
```

# Example request body approve.json file

```
"requests" : [{
   "actionType" : "Approve",
   "contextId" : "04iD0000000cw6SIAS",
   "nextApproverIds" : ["005D00000015rY9"],
   "comments" : "this record is approved"}]
```

## **Example JSON response body**

```
[ {
  "actorIds" : null,
  "entityId" : "001D00000018mImIAJ",
  "errors" : null,
  "instanceId" : "04gD0000000cvmAIAS",
  "instanceStatus" : "Approved",
  "newWorkitemIds" : [],
  "success" : true
} ]
```

# Reject a Record

Use the Process Approvals resource to reject a record or a collection of records. Each call takes an array of requests. The current user must be an assigned approver.

Examples Bulk Approvals

## **Example usage**

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/approvals/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @reject.json"
```

### Example request body reject. json file

```
"requests" : [{
   "actionType" : "Reject",
   "contextId" : "04iD000000Cw6cIAC",
   "comments" : "This record is rejected."}]
}
```

#### **Example JSON response body**

```
[ {
  "actorIds" : null,
  "entityId" : "001D00000018mImIAJ",
  "errors" : null,
  "instanceId" : "04gD0000000CvmFIAS",
  "instanceStatus" : "Rejected",
  "newWorkitemIds" : [],
  "success" : true
} ]
```

# **Bulk Approvals**

Use the Process Approvals resource to do bulk approvals. You can specify a collection of different Process Approvals requests to have them all executed in bulk.

# **Example usage**

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/approvals/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @bulk.json"
```

#### Example request body bulk. json file

```
"requests" :
[{
    "actionType" : "Approve",
    "contextId" : "04iD0000000cw6r",
    "comments" : "approving an account"
    },{
    "actionType" : "Submit",
    "contextId" : "001D000000JRWBd",
    "nextApproverIds" : ["005D00000015rY9"],
    "comments" : "submitting an account"
    },{
    "actionType" : "Submit",
    "contextId" : "003D000000QBZ08",
    "comments" : "submitting a contact"
```

Examples Get a List of Process Rules

```
}]
}
```

#### **Example JSON response body**

```
]
 "actorIds" : null,
 "entityId" : "001D00000018mImIAJ",
 "errors" : null,
 "instanceId" : "04gD000000CvmZIAS",
 "instanceStatus" : "Approved",
 "newWorkitemIds" : [ ],
 "success" : true
 }, {
 "actorIds" : null,
 "entityId" : "003D000000QBZ08IAH",
 "errors" : null,
 "instanceId" : "04gD000000CvmeIAC",
 "instanceStatus" : "Approved",
 "newWorkitemIds" : [ ],
 "success" : true
 }, {
 "actorIds" : [ "005D00000015rY9IAI" ],
 "entityId" : "001D00000JRWBdIAP",
 "errors" : null,
 "instanceId" : "04gD000000CvmfIAC",
 "instanceStatus" : "Pending",
 "newWorkitemIds" : [ "04iD000000Cw6wIAC" ],
 "success" : true
} ]
```

# Get a List of Process Rules

Use the Process Rules resource to get information about process rules.

#### Example usage

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/rules/ -H
"Authorization: Bearer token"
```

# **Example request body**

none required

# **Example JSON response body**

```
{
  "rules" : {
    "Account" : [ {
        "actions" : [ {
            "id" : "01VD0000000D2w7",
            "name" : "ApprovalProcessTask",
            "type" : "Task"
        } ],
```

Examples Get a Particular Process Rule

```
"description" : null,
    "id" : "01QD000000APli",
    "name" : "My Rule",
    "namespacePrefix" : null,
    "object" : "Account"
    } ]
}
```

# Get a Particular Process Rule

Use the Process Rules resource and specify the SObjectName and workflowRuleId of the rule you want to get the metadata for.

## **Example usage**

```
curl
https://yourInstance.salesforce.com/services/data/v30.0/process/rules/Account/01QD0000000APli
-H "Authorization: Bearer token"
```

# **Example request body**

none required

#### **Example JSON response body**

```
"actions" : [ {
    "id" : "01VD0000000D2w7",
    "name" : "ApprovalProcessTask",
    "type" : "Task"
    } ],
    "description" : null,
    "id" : "01QD0000000APli",
    "name" : "My Rule",
    "namespacePrefix" : null,
    "object" : "Account"
}
```

# **Trigger Process Rules**

Use the Process Rules resource to trigger process rules. All rules associated with the specified ID will be evaluated, regardless of the evaluation criteria. All IDs must be for records on the same object.

# **Example usage**

```
curl https://yourInstance.salesforce.com/services/data/v30.0/process/rules/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @rules.json"
```

## Example request body rules.json file

```
{
    "contextIds" : [
    "001D00000JRWBd",
    "001D000000I8mIm",
```

Examples Using Event Monitoring

```
"001D00000018aaf"]
}
```

# **Example JSON response body**

```
{
  "errors" : null,
  "success" : true
}
```

# **Using Event Monitoring**

These examples use REST API event monitoring data that contains information useful for assessing org usage trends and user behavior. Event monitoring is accessed through the Lightning Platform SOAP API and REST API by way of the EventLogFile object. Therefore, you can integrate log data with your own back-end storage and data marts to correlate data from multiple orgs and across disparate systems.



**Note**: For the supported event types that you can use with event monitoring, see Object Reference for Salesforce and Lightning Platform: EventLogFile Object.

When using event monitoring, keep the following in mind.

- In the unlikely case in which no log files are generated for 24 hours, contact Salesforce Customer Support.
- Log data is read only. You can't insert, update, or delete log data.
- Use the EventType field to determine which files were generated for your org.
- An event generates log data in real time. However, daily log files are generated during nonpeak hours the day after an event takes
  place. Therefore, daily log file data is unavailable for at least one day after an event. For hourly log files, depending on event delivery
  and final processing time, an event is expected to take three to six hours from the time of the event to be available in the log file.
  However, it can take longer.
- Log files are generated only when at least one event of a type (represented by the EventType field) occurs for the day or hour.

  If no events took place, the file isn't generated.
- Log files are available based on CreatedDate for the last 30 days when orgs purchase Event Monitoring or one day for Developer Edition orgs.
- All event monitoring logs are exposed to the API through the EventLogFile object. However, there is no access through the user interface, except through the Event Monitoring Analytics app.
- Event Monitoring log files aren't a system of record for user activity. They are a source of truth, but aren't durable. During Salesforce site switches, instance refreshes, or unplanned system outages, data loss can occur. For example, if Salesforce moves your production org instance, your event log files might have a gap in data. Salesforce makes commercially reasonable efforts to preserve event log file data integrity and avoid data loss. When Salesforce performs a site switch or instance refresh, it uses an automated process to replicate event logs.
- Hourly event log files are provided for you to review events in your orgs on an accelerated basis. However, it's possible that you don't get all event log data in hourly event log files, especially during site switches, instance refreshes, or unplanned system outages. For complete data, use the daily log files.
- If event transmission failures take too long to recover from, log files are retransmitted to ensure that they are delivered at least once. As a result, latent log files can sometimes contain duplicate event data. When your application consumes latent log files, make sure that your application handles duplicate event delivery.
- We recommend that you always query the EventLogFile object for new log files to ensure that you also include latent ones. Use the EventLogFile CreatedDate field to identify newly created log files. Hourly and daily incremental logs are delivered differently. See Differences Between Hourly and 24-Hour Event Logs for details.

All queries and examples in this section require the View Event Log Files and API Enabled user permissions. Users with the View All Data permission can also view event monitoring data.

#### IN THIS SECTION:

#### Describe Event Monitoring Using REST

Use the SObject Describe resource to retrieve all metadata for an object, including information about fields, URLs, and child relationships.

#### Query Event Monitoring Data with REST

Use the Query resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

## Get Event Monitoring Content from a Record

Use the SObject Blob Retrieve resource to retrieve BLOB data for a given record.

#### Download Large Event Log Files Using cURL with REST

You might have some event log files that are larger than your tool can handle. A command line tool such as cURL is one method to download files larger than 100 MB using the SObject Blob Retrieve object

#### Delete Event Monitoring Data

You can delete event log files that contain a user's log data. Deleting log files helps you comply with data protection and privacy regulations and controls the information that others can access. You can't delete individual rows from event logs. Instead, you must delete the entire log file that contains the user activity.

#### Query or View Hourly Event Log Files

To review events in your org on an accelerated basis, get event log files in hourly increments for recent activity. Hourly event log files can give you quicker visibility into security anomalies and custom code performance issues.

# **Describe Event Monitoring Using REST**

Use the SObject Describe resource to retrieve all metadata for an object, including information about fields, URLs, and child relationships.

# **Example**

You can use Workbench to describe event log files. In the Execute text box, type /services/data/v32.0/sobjects/EventLogFile/describe.

#### Example raw response

```
"actionOverrides" : [ ],
"activateable" : false,
"childRelationships" : [ ],
"compactLayoutable" : false,
"createable" : false,
"custom" : false,
"customSetting" : false,
"deletable" : false,
"deprecatedAndHidden" : false,
"feedEnabled" : false,
"fields" : [ {
  "autoNumber" : false,
  "byteLength" : 18,
  "calculated" : false,
  "calculatedFormula" : null,
  "cascadeDelete" : false,
```

```
"caseSensitive" : false,
  "controllerName" : null,
  "createable" : false,
   ...
}
```

# **Query Event Monitoring Data with REST**

Use the Query resource to retrieve field values from a record. Specify the fields you want to retrieve in the fields parameter and use the GET method of the resource.

You can use Workbench to query event monitoring data. To retrieve event monitoring records based on LogDate and EventType, in the Execute text box, type:

```
/services/data/v32.0/query?q=SELECT+Id+,+EventType+,+LogFile+,+LogFileLength+FROM+EventLogFile+WHERE+
LogDate+>+Yesterday+AND+EventType+=+'API'
```

#### Example raw response

```
"totalSize" : 4,
  "done" : true,
  "records" : [ {
    "attributes" : {
      "type" : "EventLogFile",
      "url": "/services/data/v32.0/sobjects/EventLogFile/0ATD00000001bROAQ"
    "Id" : "0ATD00000001bR0AQ",
    "EventType" : "API",
    "LogFile": "/services/data/v32.0/sobjects/EventLogFile/0ATD00000001bROAQ/LogFile",
    "LogDate": "2014-03-14T00:00:00.000+0000",
    "LogFileLength" : 2692.0
   }, {
    "attributes" : {
      "type" : "EventLogFile",
       "url" : "/services/data/v32.0/sobjects/EventLogFile/0ATD00000001SdOAI"
                                                                                   },
      "Id" : "0ATD00000001Sd0AI",
      "EventType" : "API",
      "LogFile" :
"/services/data/v32.0/sobjects/EventLogFile/0ATD00000001SdOAI/LogFile",
       "LogDate": "2014-03-13T00:00:00.000+0000",
      "LogFileLength" : 1345.0
       "attributes" : {
       "type" : "EventLogFile",
       "url" : "/services/data/v32.0/sobjects/EventLogFile/0ATD00000003p10AA"
                                                                                    },
       "Id": "OATD00000003p10AA",
       "EventType" : "API",
       "LogFile" :
"/services/data/v32.0/sobjects/EventLogFile/0ATD000000003p10AA/LogFile",
       "LogDate" : "2014-06-21T00:00:00.000+0000",
       "LogFileLength": 605.0 },
      "attributes" : {
```

# Get Event Monitoring Content from a Record

Use the SObject Blob Retrieve resource to retrieve BLOB data for a given record.

### **Example**

You can use Workbench to retrieve BLOB data for event monitoring. In the Execute text box, use a GET request similar to this: /services/data/v32.0/sobjects/EventLogFile/OATD00000000pyOAA/LogFile.

#### Example response body

Event monitoring content is returned in binary form. Note that the response content type won't be JSON or XML because the returned data is binary.

```
HTTP/1.1 200 OK
Date: Tue, 06 Aug 2013 16:46:10 GMT
Sforce-Limit-Info: api-usage=135/5000
Content-Type: application/octetstream
Transfer-Encoding: chunked
"EVENT_TYPE", "ORGANIZATION_ID", "TIMESTAMP","USER_ID", "CLIENT_IP",
"URI", "REFERRER_URI", "RUN_TIME"
"URI", "00DD00000000K5xD", "20130728185606.020", "005D0000001REDy",
"10.0.62.141", "/secur/contentDoor", "https-//login-salesforce-com/",
"11"
"URI", "00DD0000000K5xD", "20130728185556.930", "005D0000001REIO",
"10.0.62.141", "/secur/logout.jsp", "https-//yourInstance-salesforce-com/000/o",
"54"
"URI", "00DD0000000K5xD", "20130728185536.725", "005D0000001REIO",
"10.0.62.141", "/000D0000001ckx3",
"https-//yourInstance-salesforce-com/000D000001ckx3", "93"
```

# Download Large Event Log Files Using cURL with REST

You might have some event log files that are larger than your tool can handle. A command line tool such as cURL is one method to download files larger than 100 MB using the SObject Blob Retrieve object

# Example: Use the "X-PrettyPrint" header and the "-o" flag to output large files to .csv formats

This command downloads a file onto your machine into your downloads folder.

```
curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/EventLogFile/OAT30000000000GAA/LogFile
-H "Authorization: Bearer token" -H "X-PrettyPrint:1" -o ~/downloads/outputLogFile.csv
```

We recommend using compression when downloading large event log files. See Using Compression.

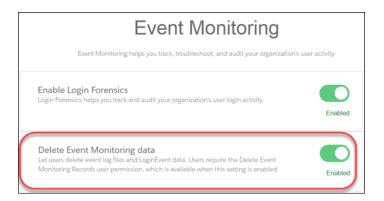
# **Delete Event Monitoring Data**

You can delete event log files that contain a user's log data. Deleting log files helps you comply with data protection and privacy regulations and controls the information that others can access. You can't delete individual rows from event logs. Instead, you must delete the entire log file that contains the user activity.

To delete an event log file, enable deletion in Setup, create a permission set that includes the Delete Event Monitoring Records user permission, and assign this permission set to your users. (Alternatively, you can assign the user permission to a custom profile.) Then those users can query and delete the EventLogFile record by using Query and Delete resources in REST or delete() in SOAP.



- 1. In Setup, in the Quick Find box, enter *Event*, and then select **Event Monitoring Settings**.
- 2. Enable deletion of event monitoring data. This action is recorded in Setup Audit Trail.



The Delete Event Monitoring Records user permission is now available to assign to a permission set. (Alternatively, you can assign the user permission to a custom profile.)

- 3. In Setup, in the Quick Find box, enter Permission, and then select Permission Sets.
- 4. Create a permission set that includes the Delete Event Monitoring Records user permission, and save the permission set.
- **5.** In Setup, in the Quick Find box, enter *users*, and then select **Users**.
- **6.** Select the user you want to grant permission to delete event monitoring data.
- 7. In the Permission Set Assignment section for this user, assign the permission set, and click **Save**. This action is recorded in Setup Audit Trail.
  - Users assigned this permission set (or any custom profile that includes the Delete Event Monitoring Records user permission) can now delete event monitoring data. The next steps show you how to use the API to delete the data.
- **8.** To locate the logs containing the user activity that you want to delete, query the EventLogFile object. For details, see Query Event Monitoring Data with REST on page 90.
- 9. Note the IDs of the returned logs.
- 10. Use the SObject Rows resource to delete records. Specify the record ID, and use the DELETE method. For more information, see Delete a Record on page 47. Or, you can use Workbench to delete blob format data for event monitoring. In the Execute text box, use a DELETE request similar to /services/data/v41.0/sobjects/EventLogFile/0ATD00000000pyOAA.

# Query or View Hourly Event Log Files

To review events in your org on an accelerated basis, get event log files in hourly increments for recent activity. Hourly event log files can give you quicker visibility into security anomalies and custom code performance issues.

#### Examples

Suppose you're a security analyst monitoring for anomalous user behavior. By pulling more frequent updates into your security system, you can be alerted that a suspicious event has taken place within hours, rather than one or two days later.

In another example, let's say you're a developer. You've identified a series of Apex failures in your org, and you want to proactively refactor your Apex code to improve performance. You review hourly log files to pinpoint the issues and fix your code in hours, before your end users start complaining about poor performance.

#### Considerations

- Hourly event log file integration with the Event Monitoring Analytics app is unavailable.
- Depending on event delivery and final processing time, an event is expected to take three to

# six hours from the time of the event to be available in the log file. However, it can take longer. When delays in processing occur and event logs for a particular hour arrive later, a new log file is created for the event/date/hour

and lists only the new events. Use the creation date and an incremental sequence number to identify a new log file. Always use the most recently processed event log file for a particular date. However, if event log files have already been pulled into a third-party application, they could require deduplication in that application.

If both hourly and daily logs are enabled, daily logs always have a sequence number of 0 because there is only one file per daily interval. CreatedDate indicates when the file was generated. If CreatedDate is after the last event log file download, there are new events to be processed.

For best practices, use CreatedDate in the WHERE clause to select logs created after the last downloaded event log file. For example, if the last downloaded file was at 12PM 2/1/2018, to find the next log file, use +CreatedDate+>+"2018-02-01T12:00:00Z" in the WHERE clause.

- During Salesforce site switches, instance refreshes, or unplanned system outages, data loss can occur. For example, if Salesforce moves your production org instance, your event log files could have a gap in data. Salesforce makes commercially reasonable efforts to preserve event log file data integrity and avoid data loss. When Salesforce performs a site switch or instance refresh, it uses an automated process to replicate event logs.
- In the unlikely case in which no log files are generated for 24 hours, contact Salesforce Support.

# IN THIS SECTION:

#### Query Hourly Event Log Files

You query hourly event log files in the same way you query 24-hour log files.

#### Differences Between Hourly and 24-Hour Event Logs

You receive event log files approximately every hour in addition to 24-hour log files. Review the differences between the two logs so that you can filter your files to analyze the event data you want.

# Query Hourly Event Log Files

You guery hourly event log files in the same way you guery 24-hour log files.

# **EDITIONS**

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

# **USER PERMISSIONS**

To access the API and query log files:

API Enabled AND View **Event Log Files** 

To view event log files:

View All Data

Suppose you're an administrator. Your Chief Security Officer asks you to identify who modified specific accounts and opportunities in the past two hours. You query the hourly URI event log files using the EventLogFile object to review the page requests and request status. Because EventLogFile also returns 24-hour log files, use this SOQL syntax to filter out the 24-hour log files.

- 1. In Workbench, select utilities > REST Explorer.
- 2. Replace the existing text with:/services/data/v API\_version
  .0/query?q=SELECT+Id+,+EventType+,+Interval+,+LogDate+,+LogFile+
- 3. Append the following to the query to make it complete:

  FROM+EventLogFile+WHERE+EventType+=+'URI',+Interval+=+'Hourly'

  In the query, Interval=Hourly makes sure that only hourly event log file data is returned. Alternatively, you can use Sequence to filter out 24-hour event log files (Sequence!=0). To get both hourly and 24-hour files, use Sequence>=0.

#### 4. Click Execute.

If your sandbox org has URI events, you see log file records in your query results. You can also download the event log files to review the data in a CSV file. For more information, see Trailhead: Download and Visualize Event Log Files.

# Differences Between Hourly and 24-Hour Event Logs

You receive event log files approximately every hour in addition to 24-hour log files. Review the differences between the two logs so that you can filter your files to analyze the event data you want.

Hourly Log Files	24-Hour Log Files	
One or more files generated for every hour of activity.	One file generated for every 24 hours of activity.	
Available in the API. You can manually import data into third-party visualization apps.	Available in the API and integrated with the Event Monitoring Analytics app and third-party visualization apps.	
Key values in the EventLogFile object are:	Key values in the EventLogFile object are:	
• Interval—Hourly	• Interval—Daily	
<ul> <li>CreatedDate—Timestamp of when the log file was created. Use this field to identify new files.</li> </ul>	<ul> <li>CreatedDate—Timestamp of when the log file was created. Use this field to identify new files.</li> </ul>	
• LogDate—Timestamp that marks the beginning of the interval when the events occurred. For example, for events that occurred between 11:00 AM and 12:00 PM on 3/7/2016, this field's value is 2016-03-07T11:00:00.000Z.	• LogDate—Timestamp that marks the beginning of the interval when the events occurred. For example, for events that occurred on 3/7/2016, this field's value is 2016-03-07T00:00:00.000+0000.	
<ul> <li>Sequence—1+. This value increases by 1 when events are added in the same hour after the latest event log file is created. The value resets to 1 in the subsequent hour.</li> <li>See also these Considerations regarding hourly event logs.</li> </ul>	• Sequence—0	
When an hourly incremental log file is delivered, a file with new logs for that hour is created. The Sequence field is incremented for each new file.	When a daily incremental log file is delivered, a new file replaces the original file with the full set of available logs for that date. The CreatedDate field is updated.	



Note: Like with 24-hour event monitoring, hourly event log data is available for the past 30 days.

# **Using Composite Resources**

The examples in this section use composite resources to improve your application's performance by minimizing the number of round-trips between the client and server.

#### IN THIS SECTION:

#### Execute Dependent Requests in a Single API Call

The following example uses the Composite resource to execute several dependent requests all in a single call. First, it creates an account and retrieves its information. Next it uses the account data and the Composite resource's reference ID functionality to create a contact and populate its fields based on the account data. Then it retrieves specific information about the account's owner by using query parameters in the request string. Finally, if the metadata has been modified since a certain date, it retrieves account metadata. The composite.json file contains the composite request and subrequest data.

#### Update an Account, Create a Contact, and Link Them with a Junction Object

The following example uses the Composite resource to update some fields on an account, create a contact, and link the two records with a junction object called AccountContactJunction. All these requests are executed in a single call. The composite.json file contains the composite request and subrequest data.

## Update a Record and Get Its Field Values in a Single Request

Use the Batch resource to execute multiple requests in a single API call.

#### Create Nested Records

Use the SObject Tree resource to create nested records that share a root record type. For example, in a single request, you can create an account along with its child contacts, and a second account along with its child accounts and contacts. Once the request is processed, the records are created and parents and children are automatically linked by ID. In the request data, you supply the record hierarchies, required and optional field values, each record's type, and a reference ID for each record, and then use the POST method of the resource. The response body will contain the IDs of the created records if the request is successful. Otherwise, the response contains only the reference ID of the record that caused the error and the error information.

#### Create Multiple Records

While the SObject Tree resource can be used to create nested records, you can also create multiple, unrelated records of the same type. In a single request, you can create up to two hundred records. In the request data, you supply the required and optional field values for each record, each record's type, and a reference ID for each record, and then use the POST method of the resource. The response body will contain the IDs of the created records if the request is successful. Otherwise, the response contains only the reference ID of the record that caused the error and the error information.

# Execute Dependent Requests in a Single API Call

The following example uses the Composite resource to execute several dependent requests all in a single call. First, it creates an account and retrieves its information. Next it uses the account data and the Composite resource's reference ID functionality to create a contact and populate its fields based on the account data. Then it retrieves specific information about the account's owner by using query parameters in the request string. Finally, if the metadata has been modified since a certain date, it retrieves account metadata. The composite.json file contains the composite request and subrequest data.

# Execute dependent requests in a single API call

```
curl https://yourInstance.salesforce.com/services/data/v38.0/composite/ -H
"Authorization: Bearer token -H "Content-Type: application/json" -d "@composite.json"
```

## Request body composite.json file

```
"allOrNone" : true,
    "compositeRequest" : [{
       "method" : "POST",
       "url" : "/services/data/v38.0/sobjects/Account",
       "referenceId" : "NewAccount",
        "body" : {
            "Name" : "Salesforce",
            "BillingStreet" : "Landmark @ 1 Market Street",
            "BillingCity" : "San Francisco",
            "BillingState" : "California",
            "Industry" : "Technology"
        }
   },{
       "method" : "GET",
        "referenceId" : "NewAccountInfo",
        "url" : "/services/data/v38.0/sobjects/Account/@{NewAccount.id}"
    },{
        "method" : "POST",
        "referenceId" : "NewContact",
        "url" : "/services/data/v38.0/sobjects/Contact",
        "body" : {
            "lastname" : "John Doe",
            "Title" : "CTO of @{NewAccountInfo.Name}",
            "MailingStreet" : "@{NewAccountInfo.BillingStreet}",
            "MailingCity" : "@{NewAccountInfo.BillingAddress.city}",
            "MailingState" : "@{NewAccountInfo.BillingState}",
            "AccountId" : "@{NewAccountInfo.Id}",
            "Email" : "jdoe@salesforce.com",
            "Phone": "1234567890"
        }
   },{
       "method" : "GET",
        "referenceId" : "NewAccountOwner",
        "url" :
"/services/data/v38.0/sobjects/User/@{NewAccountInfo.OwnerId}?fields=Name,companyName,Title,City,State"
   },{
        "method" : "GET",
        "referenceId" : "AccountMetadata",
        "url" : "/services/data/v38.0/sobjects/Account/describe",
        "httpHeaders" : {
            "If-Modified-Since": "Tue, 31 May 2016 18:13:37 GMT"
   } ]
```

#### Response body after successfully executing the composite request

```
{
    "compositeResponse" : [{
        "body" : {
            "id" : "001R00000033JNuIAM",
```

```
"success" : true,
        "errors" : [ ]
    "httpHeaders" : {
     "Location" : "/services/data/v38.0/sobjects/Account/001R00000033JNuIAM"
    "httpStatusCode" : 201,
    "referenceId" : "NewAccount"
},{
    "body" : {
       all the account data
    "httpHeaders" : {
        "ETag" : "\"Jbjuzw7dbhaEG3fd90kJbx6A0ow=\"",
        "Last-Modified": "Fri, 22 Jul 2016 20:19:37 GMT"
    "httpStatusCode" : 200,
    "referenceId" : "NewAccountInfo"
},{
   "body" : {
        "id" : "003R00000025REHIA2",
        "success" : true,
        "errors" : [ ]
    "httpHeaders" : {
        "Location" : "/services/data/v38.0/sobjects/Contact/003R00000025REHIA2"
   },
    "httpStatusCode" : 201,
    "referenceId" : "NewContact"
},{
    "body" : {
        "attributes" : {
        "type" : "User",
        "url": "/services/data/v38.0/sobjects/User/005R0000000190CIAS"
        "Name" : "Jane Doe",
        "CompanyName" : "Salesforce",
        "Title" : Director,
        "City" : "San Francisco",
        "State" : "CA",
        "Id" : "005R0000000190CIAS"
    },
    "httpHeaders" : { },
    "httpStatusCode" : 200,
    "referenceId" : "NewAccountOwner"
},{
   "body" : null,
    "httpHeaders" : {
        "ETag" : "\"f2293620\"",
        "Last-Modified" : "Fri, 22 Jul 2016 18:45:56 GMT"
    },
    "httpStatusCode" : 304,
    "referenceId" : "AccountMetadata"
```

```
} ]
}
```

# Update an Account, Create a Contact, and Link Them with a Junction Object

The following example uses the Composite resource to update some fields on an account, create a contact, and link the two records with a junction object called AccountContactJunction. All these requests are executed in a single call. The composite.json file contains the composite request and subrequest data.

### Update an account, create a contact, and link them with a junction object

```
curl https://yourInstance.salesforce.com/services/data/v38.0/composite/ -H "Authorization: Bearer token -H "Content-Type: application/json" -d "@composite.json"
```

# Request body composite.json file

```
"allOrNone" : true,
"compositeRequest" : [{
   "method" : "PATCH",
    "url": "/services/data/v38.0/sobjects/Account/001xx000003DIpcAAG",
    "referenceId" : "UpdatedAccount",
    "body" : {
        "Name" : "Salesforce",
        "BillingStreet" : "Landmark @ 1 Market Street",
        "BillingCity" : "San Francisco",
        "BillingState" : "California",
        "Industry" : "Technology"
   }
   "method" : "POST",
   "referenceId" : "NewContact",
   "url" : "/services/data/v38.0/sobjects/Contact/",
    "body" : {
        "lastname" : "John Doe",
        "Phone": "1234567890"
    }
},{
   "method" : "POST",
   "referenceId" : "JunctionRecord",
   "url" : "/services/data/v38.0/sobjects/AccountContactJunction c",
    "body" : {
        "accountId c": "001xx000003DIpcAAG",
        "contactId c" : "@{NewContact.id}"
} ]
```

#### Response body after successfully executing the composite request

```
"compositeResponse" : [{
   "body" : null,
   "httpHeaders" : { },
   "httpStatusCode" : 204,
```

```
"referenceId" : "UpdatedAccount"
  }, {
    "body" : {
      "id" : "003R00000025R22IAE",
      "success" : true,
      "errors" : [ ]
   },
    "httpHeaders" : {
      "Location": "/services/data/v38.0/sobjects/Contact/003R00000025R22IAE"
    "httpStatusCode" : 201,
    "referenceId" : "NewContact"
    "body" : {
      "id" : "a00R0000000iN4gIAE",
      "success" : true,
      "errors" : [ ]
    },
    "httpHeaders" : {
      "Location" :
"/services/data/v38.0/sobjects/AccountContactJunction c/a00R0000000iN4gIAE"
    "httpStatusCode" : 201,
    "referenceId" : "JunctionRecord"
  } ]
```

# Update a Record and Get Its Field Values in a Single Request

Use the Batch resource to execute multiple requests in a single API call.

The following example updates the name on an account and gets some of the account's field values in a single request. The batch.json file contains the subrequest data.

#### Update a record and query its name and billing postal code in a single request

```
curl https://yourInstance.salesforce.com/services/data/v34.0/composite/batch/ -H "Authorization: Bearer token -H "Content-Type: application/json" -d "@batch.json"
```

# Request body batch. json file

Examples Create Nested Records

## Response body after successfully executing the subrequests

```
"hasErrors" : false,
"results" : [{
    "statusCode" : 204,
    "result" : null
    },{
    "statusCode" : 200,
    "result": {
        "attributes" : {
            "type" : "Account",
            "url" : "/services/data/v34.0/sobjects/Account/001D000000K0fXOIAZ"
        },
        "Name" : "NewName",
        "BillingPostalCode" : "94105",
        "Id" : "001D000000K0fXOIAZ"
     }
}
}
```

SEE ALSO:

Batch

# Create Nested Records

Use the SObject Tree resource to create nested records that share a root record type. For example, in a single request, you can create an account along with its child contacts, and a second account along with its child accounts and contacts. Once the request is processed, the records are created and parents and children are automatically linked by ID. In the request data, you supply the record hierarchies, required and optional field values, each record's type, and a reference ID for each record, and then use the POST method of the resource. The response body will contain the IDs of the created records if the request is successful. Otherwise, the response contains only the reference ID of the record that caused the error and the error information.

The following example creates two sets of nested records. The first set includes an account and two child contact records. The second set includes an account, one child account record, and one child contact record. The record data is provided in newrecords. i son.

#### Example for creating two new accounts and their child records

```
curl https://yourInstance.salesforce.com/services/data/v34.0/composite/tree/Account/
-H "Authorization: Bearer token -H "Content-Type: application/json" -d "@newrecords.json"
```

## Example request body newrecords.json file for creating two new Accounts and their child records

```
{
"records" :[{
    "attributes" : {"type" : "Account", "referenceId" : "ref1"},
    "name" : "SampleAccount1",
    "phone" : "1234567890",
    "website" : "www.salesforce.com",
    "numberOfEmployees" : "100",
    "industry" : "Banking",
    "Contacts" : {
        "records" : [{
```

Examples Create Nested Records

```
"attributes" : {"type" : "Contact", "referenceId" : "ref2"},
         "lastname" : "Smith",
         "Title" : "President",
         "email" : "sample@salesforce.com"
         "attributes" : {"type" : "Contact", "referenceId" : "ref3"},
         "lastname" : "Evans",
         "title" : "Vice President",
         "email" : "sample@salesforce.com"
         } ]
     }
   },{
    "attributes" : {"type" : "Account", "referenceId" : "ref4"},
    "name" : "SampleAccount2",
    "phone": "1234567890",
    "website" : "www.salesforce.com",
    "numberOfEmployees" : "52000",
    "industry" : "Banking",
    "childAccounts" : {
     "records" : [{
       "attributes" : {"type" : "Account", "referenceId" : "ref5"},
       "name" : "SampleChildAccount1",
       "phone": "1234567890",
       "website" : "www.salesforce.com",
       "numberOfEmployees": "100",
       "industry" : "Banking"
       } ]
     },
    "Contacts" : {
     "records" : [{
       "attributes" : {"type" : "Contact", "referenceId" : "ref6"},
       "lastname" : "Jones",
       "title" : "President",
       "email" : "sample@salesforce.com"
       } ]
 } ]
}
```

## Example response body after successfully creating records and relationships

```
"hasErrors" : false,
"results" : [{
    "referenceId" : "ref1",
    "id" : "001D0000000K0fX0IAZ"
    },{
    "referenceId" : "ref4",
    "id" : "001D0000000K0fXPIAZ"
    },{
    "referenceId" : "ref2",
    "id" : "003D000000QV9n2IAD"
    },{
    "referenceId" : "ref3",
    "id" : "003D000000QV9n3IAD"
```

Examples Create Multiple Records

```
},{
    "referenceId" : "ref5",
    "id" : "001D000000K0fXQIAZ"
},{
    "referenceId" : "ref6",
    "id" : "003D000000QV9n4IAD"
}]
}
```

Once the request is processed, all six records are created with the parent-child relationships specified in the request.

SEE ALSO:

**SObject Tree** 

## Create Multiple Records

While the SObject Tree resource can be used to create nested records, you can also create multiple, unrelated records of the same type. In a single request, you can create up to two hundred records. In the request data, you supply the required and optional field values for each record, each record's type, and a reference ID for each record, and then use the POST method of the resource. The response body will contain the IDs of the created records if the request is successful. Otherwise, the response contains only the reference ID of the record that caused the error and the error information.

The following example creates four new accounts. The record data is provided in newrecords.json.

## **Example for creating four new accounts**

```
curl https://yourInstance.salesforce.com/services/data/v34.0/composite/tree/Account/
-H "Authorization: Bearer token -H "Content-Type: application/json" -d "@newrecords.json"
```

#### Example request body newrecords. json file for creating four new accounts

```
"records" :[{
   "attributes" : {"type" : "Account", "referenceId" : "ref1"},
   "name" : "SampleAccount1",
   "phone" : "1111111111",
   "website" : "www.salesforce.com",
   "numberOfEmployees" : "100",
   "industry" : "Banking"
   "attributes" : {"type" : "Account", "referenceId" : "ref2"},
   "name" : "SampleAccount2",
   "phone" : "222222222",
   "website" : "www.salesforce2.com",
   "numberOfEmployees" : "250",
   "industry" : "Banking"
   },{
   "attributes" : {"type" : "Account", "referenceId" : "ref3"},
   "name" : "SampleAccount3",
   "phone": "3333333333",
   "website" : "www.salesforce3.com",
   "numberOfEmployees" : "52000",
   "industry" : "Banking"
   },{
```

Examples Create Multiple Records

```
"attributes" : {"type" : "Account", "referenceId" : "ref4"},
   "name" : "SampleAccount4",
   "phone" : "44444444444",
   "website" : "www.salesforce4.com",
   "numberOfEmployees" : "2500",
   "industry" : "Banking"
   }]
}
```

## Example response body after successfully creating records

```
"hasErrors" : false,
   "results" : [{
        "referenceId" : "ref1",
        "id" : "001D000000K1YFjIAN"
        },{
        "referenceId" : "ref2",
        "id" : "001D000000K1YFKIAN"
        },{
        "referenceId" : "ref3",
        "id" : "001D000000K1YFIIAN"
        },{
        "referenceId" : "ref4",
        "id" : "001D000000K1YFmIAN"
        }]
}
```

SEE ALSO:

SObject Tree

# **CHAPTER 4** Reference

The following table lists supported REST resources in the API and provides a brief description for each. In each case, the URI for the resource follows the base URI, which you retrieve from the authentication service: http://domain/services/data.domain might be the Salesforce instance you are using, or a custom domain. For example, to retrieve basic information about an Account object in version 20.0: https://yourInstance.salesforce.com/services/data/v20.0/sobjects/Account/.

Click a call name to see syntax, usage, and more information for that call.

Resource Name URI and Description	
Versions	/
	Lists summary information about each Salesforce version currently available, including the version, label, and a link to each version's root.
Resources by Version	/vXX.X/
	Lists available resources for the specified API version, including resource name and URI.
Limits	/vXX.X/limits/
	Lists information about limits in your org.
Describe Global	/vXX.X/sobjects/
	Lists the available objects and their metadata for your organization's data.
SObject Basic Information	/vXX.X/sobjects/ <b>SObject</b> /
	Describes the individual metadata for the specified object. Can also be used to create a new record for a given object.
SObject Describe	/vXX.X/sobjects/ <i>SObject</i> /describe/
	Completely describes the individual metadata at all levels for the specified object.
SObject Get Deleted	/vXX.X/sobjects/ <b>SObject</b> /deleted/ ?start= <b>startDateAndTime</b> &end= <b>endDateAndTime</b>
	Retrieves the list of individual records that have been deleted within the given timespan for the specified object.
SObject Get Updated	/vXX.X/sobjects/ <b>SObject</b> /updated/ ?start= <b>startDateAndTime</b> &end= <b>endDateAndTime</b>
	Retrieves the list of individual records that have been updated (added or changed) within the given timespan for the specified object.
SObject Named Layouts	/vXX.X/sobjects/ <b>SObject</b> /describe/namedLayouts/ <b>layoutName</b>
	Retrieves information about alternate named layouts for a given object.

## Reference

Resource Name	URI and Description
SObject Rich Text Image	/vXX.X/sobjects/ <b>SObject/id/</b> /richTextImageFields/ <b>fieldName</b> /contentReferenceId
Retrieve	Retrieves the specified image data from a specific rich text area field in a given record.
SObject Rows	/vXX.X/sobjects/ <b>SObject/id</b> /
	Accesses records based on the specified object ID. Retrieves, updates, or deletes records. This resource can also be used to retrieve field values.
SObject Rows by External ID	/vXX.X/sobjects/ <b>SObject/fieldName/fieldValue</b>
	Creates new records or updates existing records (upserts records) based on the value of a specified external ID field.
SObject ApprovalLayouts	/vXX.X/sobjects/ <b>SObjectName</b> /describe/approvalLayouts/
	Returns a list of approval layouts for a specified object.
SObject CompactLayouts	/vXX.X/sobjects/ <i>Object</i> /describe/compactLayouts/
	Returns a list of compact layouts for a specific object.
Describe Layouts	/vXX.X/sobjects/global/describe/layouts/
	/vXX.X/sobjects/object/describe/layouts/
	Returns a list of layouts and descriptions.
SObject PlatformAction	/services/data/vXX.X/sobjects/PlatformAction
	PlatformAction is a virtual read-only object. It enables you to query for actions displayed in the UI, given a user, a context, device format, and a record ID. Examples include standard and custom buttons, quick actions, and productivity actions.
Lightning Toggle Metrics	/services/data/vXX.X/sobjects/LightningToggleMetricsReturn details about users who switched between Salesforce Classic and Lightning Experience.
Lightning Usage by App Type	/services/data/vXX.0/sobjects/LightningUsageByAppTypeMetrics Return the total number of Lightning Experience and Salesforce Mobile users.
Lightning Usage by Browser	
Lightning Usage by blowser	/services/data/vXX.0/sobjects/LightningUsageByBrowserMetrics
	Return Lightning Experience usage results grouped by browser instance.
Lightning Usage by Page	/services/data/vXX.0/sobjects/LightningUsageByPageMetrics
	Represents standard pages users viewed most frequently in Lightning Experience.
Lightning Usage by FlexiPage	/services/data/vXX.0/sobjects/LightningUsageByFlexiPageMetrics
	Return details about the custom pages viewed most frequently in Lightning Experience.
Lightning Exit by Page Metrics	/services/data/vXX.0/sobjects/LightningUsageByFlexiPageMetrics

Resource Name	URI and Description	
	Return frequency metrics about the standard pages within which users switched from Lightning Experience to Salesforce Classic.	
SObject Relationships	/vXX.X/sobjects/ <b>SObject/id/relationship name</b>	
	Accesses records by traversing object relationships via friendly URLs. You can retrieve, update, or delete the record associated with the traversed relationship field. If there are multiple related records, you can retrieve the complete set of associated records.	
SObject Blob Retrieve	/vXX.X/sobjects/ <b>SObject</b> /id/blobField	
	Retrieves the specified blob field from an individual record.	
SObject Quick Actions	/vXX.X/sobjects/ <b>object</b> /quickActions/	
	/vXX.X/sobjects/object/quickActions/{action name}	
	/vXX.X/sobjects/object/quickActions/{action name}/describe/	
	<pre>services/data/vXX.X/sobjects/object/quickActions/{action name}/defaultValues/</pre>	
	<pre>vXX.X/sobjects/object/quickActions/{action name}/defaultValues/{parent id}</pre>	
	Returns a list of actions and their details.	
SObject Suggested Articles	vXX.X/sobjects/ <b>SObject</b> /suggestedArticles?language= <b>article</b> <pre>language&amp;subject=subject&amp;description=description</pre>	
	vXX.X/sobjects/ <b>SObject</b> / <b>ID</b> /suggestedArticles?language= <b>article</b> language	
	Returns a list of suggested Salesforce Knowledge articles for a case, work order, or work order line item.	
SObject User Password	/vXX.X/sobjects/User/ <i>user id</i> /password	
	/vXX.X/sobjects/SelfServiceUser/ <b>self service user id</b> /password	
	Set, reset, or get information about a user password.	
Platform Event Schema by	/v <b>XX.X</b> /sobjects/ <b>Event_Name</b> /eventSchema	
Event Name	Gets the definition of a platform event in JSON format for an event name.	
Platform Event Schema by	/v <b>XX.X</b> /event/eventSchema/ <b>Schema_ID</b>	
Schema ID	Gets the definition of a platform event in JSON format for a schema ID.	
AppMenu	/vXX.X/appMenu/AppSwitcher/	
	/vXX.X/appMenu/Salesforce1/	
	Returns a list of items in either the Salesforce app drop-down menu or the Salesforce for Android, iOS, and mobile web navigation menu.	

Resource Name	URI and Description	
Compact Layouts	/vXX.X/compactLayouts?q= <b>object list</b>	
	Returns a list of compact layouts for multiple objects.	
Invocable Actions	/vXX.X/actions/standard	
	/vXX.X/actions/custom	
	Use actions to add more functionality to your applications. Choose from standard actions, such as posting to Chatter or sending email, or create actions based on your company's needs.	
Parameterized Search	/vXX.X/parameterizedSearch/?q= <b>search string</b>	
	Executes a simple RESTful search using parameters instead of a SOSL clause. Indicate parameters in a URL in the GET method. Or, use POST for more complex JSON searches.	
Process Approvals	/vXX.X/process/approvals/	
	Returns a list of all approval processes. Can also be used to submit a particular record if that entity supports an approval process and one has already been defined. Records can be approved and rejected if the current user is an assigned approver.	
Process Rules	/vXX.X/process/rules/	
	Returns a list of all active workflow rules. If a rule has actions, the actions will be listed under the rule. Can also be used to trigger all workflow rules that are associated with a specified record. The actions for a rule are only fired if the rule's criteria is met.	
Query	/vXX.X/query/?q= <b>soq1</b>	
	Executes the specified SOQL query.	
QueryAll	/vXX.X/queryAll/?q= <b>soql</b>	
	Executes the specified SOQL query. Results can include deleted, merged and archived records.	
Quick Actions	/vXX.X/quickActions/	
	Return a list of global quick actions and their types, as well as custom fields and objects that appear in the Chatter feed.	
Recently Viewed Items	/vXX.X/recent	
	Gets the most recently accessed items that were viewed or referenced by the current user.	
Record Count	/vXX.X/limit/recordCount	
	Lists information about object record counts in your organization.	
Relevant Items	/vXX.X/sobjects/relevantItems	
	Gets the current user's most relevant items. Relevant items include records for objects in the user's global search scope and also most recently used (MRU) objects.	
Search	/vXX.X/search/?q= <b>sos1</b>	

Resource Name	URI and Description	
	Executes the specified SOSL search. The search string must be URL-encoded.	
Search Scope and Order	/vXX.X/search/scopeOrder	
	Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often and arranges the search results accordingly. Objects used most frequently appear at the top of the list.	
Search Result Layouts	/vXX.X/searchlayout/?q=Comma delimited object list	
	Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.	
Search Suggested Article Title	/vXX.X/search/suggestTitleMatches?q= <b>search</b> string&language= <b>article</b>	
Matches	<pre>language&amp;publishStatus=article publication status</pre>	
	Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search.	
Search Suggested Queries	vXX.X/search/suggestSearchQueries?q= <b>search string</b> &language= <b>language of query</b>	
	Returns a list of suggested searches based on the user's query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search.	
Tabs	/vXX.X/tabs	
	Returns a list of all tabs—including Lightning page tabs—available to the current user, regardless of whether the user has chosen to hide tabs via the All Tabs (+) tab customization feature.	
Themes	/vXX.X/theme	
	Gets the list of icons and colors used by themes in the Salesforce application.	

# Composite Resources

Resource Name	URI	Description
Batch	/vXX.X/composite/batch	Executes up to 25 subrequests in a single request.
SObject Tree	/vXX.X/composite/tree	Creates one or more sObject trees with root records of the specified type. An sObject tree is a collection of nested, parent-child records with a single root record.

Reference Versions

# **Versions**

Lists summary information about each Salesforce version currently available, including the version, label, and a link to each version's root.

URI

/

## **Formats**

JSON, XML

## **HTTP Method**

**GET** 

#### Authentication

none

## **Parameters**

none

## Example

See List Available REST API Versions on page 35.

# Resources by Version

Lists available resources for the specified API version, including resource name and URI.

URI

/vxx.x/

## **Formats**

JSON, XML

## **HTTP Method**

**GET** 

## Authentication

Authorization: Bearer token

## **Parameters**

none

#### Example

See List Available REST Resources. on page 39

## Limits

Lists information about limits in your org. This resource is available in REST API version 29.0 and later for API users with the View Setup and Configuration permission. The resource returns these limits:

Limit Label	Description
Analytics	
ConcurrentAsyncGetReportInstances	Concurrent REST API requests for results of asynchronous report runs

Reference Limits

Limit Label	Description
ConcurrentSyncReportRuns	Concurrent synchronous report runs via REST API
HourlyAsyncReportRuns	Hourly asynchronous report runs via REST API
HourlySyncReportRuns	Hourly synchronous report runs via REST API
HourlyDashboardRefreshes	Hourly dashboard refreshes via REST API
HourlyDashboardResults	Hourly REST API requests for dashboard results
HourlyDashboardStatuses	Hourly dashboard status requests via REST API
Email	
MassEmail	Daily number of mass emails that are sent to external email addresses via Apex or APIs
SingleEmail	Daily number of single emails that are sent to external email addresses
	Note: For orgs created before Spring '19, the daily limit is enforced only for emails sent via Apex and Salesforce APIs except for the REST API. For orgs created in Spring '19 and later, the daily limit is also enforced for email alerts, simple email actions, Send Email actions in flows, and REST API. If one of the newly counted emails can't be sent because your org has reached the limit, we notify you by email and add an entry to the debug logs.
Lightning Platform REST and Bulk APIs	
DailyApiRequests	Daily API calls
DailyBulkApiRequests	Daily Bulk API calls
DailyBulkV2QueryFileStorageMB	Daily storage for queries in Bulk API 2.0 (measured in MB). This limit is available in API version 47.0 and later.
DailyBulkV2QueryJobs	Daily number of query jobs in Bulk API 2.0. This limit is available in API version 47.0 and later.
Platform Events	
The following values apply only to platform events. They don't apply to Change Data Capture events.	
HourlyPublishedPlatformEvents	High-volume platform event notifications published per hour
HourlyPublishedStandardVolumePlatform Events	Standard-volume platform event notifications published per hour
DailyStandardVolumePlatformEvents	Daily standard-volume platform event notifications delivered to CometD clients
Platform Events and Change Data Capture	
MonthlyPlatformEvents	Monthly entitlement or default allocation of high-volume platform event and Change Data Capture event delivery to CometD clients. Both types of events

Reference Limits

Limit Label	Description
	share this value. If you don't have an add-on license, your daily default allocation is the monthly entitlement returned divided by 30.
Salesforce Connect	
HourlyLongTermIdMapping	Hourly new long-term external record ID mappings
HourlyODataCallout	Hourly OData callouts
HourlyShortTermIdMapping	Hourly new short-term external record ID mappings
Salesforce Developer Experience	
ActiveScratchOrgs	Daily and active scratch org counts
Storage	
DataStorageMB	Data storage (MB)
	(The API user must have the Manage Users permission)
FileStorageMB	File storage (MB)
	(The API user must have the Manage Users permission)
Streaming API—Durable (API version 37.0 and lat	er)
DailyDurableGenericStreamingApiEvents	Generic events notifications delivered in the past 24 hours to all CometD clients
DailyDurableStreamingApiEvents	PushTopic event notifications delivered in the past 24 hours to all CometD clients
DurableStreamingApiConcurrentClients	Concurrent CometD clients (subscribers) across all channels and for all event types
Streaming API (API version 36.0 and earlier)	
DailyGenericStreamingApiEvents	Generic events notifications delivered in the past 24 hours to all CometD clients
DailyStreamingApiEvents	PushTopic event notifications delivered in the past 24 hours to all CometD clients
StreamingApiConcurrentClients	Concurrent CometD clients (subscribers) across all channels and for all event types
Workflows	
DailyWorkflowEmails	Daily workflow emails
HourlyTimeBasedWorkflow	Hourly workflow time triggers

## URI

/vXX.X/limits/

Reference Describe Global

#### **Formats**

JSON, XML

## **HTTP Method**

**GET** 

## **Authentication**

Authorization: Bearer token

#### Example

See List Organization Limits.

## **Describe Global**

Lists the available objects and their metadata for your organization's data. In addition, it provides the organization encoding, as well as the maximum batch size permitted in queries. For more information on encoding, see Internationalization and Character Sets.

You can use the If-Modified-Since header with this resource, with the date format EEE, dd MMM yyyy HH:mm:ss z. When using this header, if no available object's metadata has changed since the provided date, a 304 Not Modified status code is returned with no response body.

### URI

/vXX.X/sobjects/

#### **Formats**

JSON, XML

## **HTTP Method**

**GET** 

## Authentication

Authorization: Bearer token

### **Parameters**

none required

## Example

See Get a List of Objects on page 39.

## **Error responses**

See Status Codes and Error Responses on page 277.

# **SObject Basic Information**

Describes the individual metadata for the specified object. Can also be used to create a new record for a given object. For example, this can be used to retrieve the metadata for the Account object using the GET method, or create a new Account object using the POST method.

#### URI

/vXX.X/sobjects/**SObjectName**/

#### **Formats**

JSON, XML

## **HTTP Method**

GET, POST

Reference SObject Describe

#### Authentication

Authorization: Bearer token

#### **Parameters**

none required

## **Examples**

For an example of retrieving metadata for an object, see Retrieve Metadata for an Object on page 41.

- For an example of creating a new record using POST, see Create a Record on page 45.
- For an example of create a new record along with providing blob data for the record, see Insert or Update Blob Data on page

# **SObject Describe**

Completely describes the individual metadata at all levels for the specified object. For example, this can be used to retrieve the fields, URLs, and child relationships for the Account object.

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

#### URI

/vXX.X/sobjects/SObjectName/describe/

#### **Formats**

JSON, XML

#### **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

#### **Parameters**

none required

#### Example

See Get Field and Other Metadata for an Object on page 42. For an example that uses the If-Modified-Since HTTP header, see Get Object Metadata Changes on page 44.

# SObject Get Deleted

Retrieves the list of individual records that have been deleted within the given timespan for the specified object. SObject Get Deleted is available in API version 29.0 and later.

This resource is commonly used in data replication applications. Note the following considerations:

- Deleted records are written to a delete log which this resource accesses. A background process that runs every two hours purges
  records that have been in an organization's delete log for more than two hours if the number of records is above a certain limit.
   Starting with the oldest records, the process purges delete log entries until the delete log is back below the limit. This is done to
  protect Salesforce from performance issues related to massive delete logs
- Information on deleted records are returned only if the current session user has access to them.

Reference SObject Get Deleted

• Results are returned for no more than 15 days previous to the day the call is executed (or earlier if an administrator has purged the Recycle Bin).

See "Data Replication" in the SOAP API Developer Guide for additional details on data replication and data replication limits.

## URI

/vXX.X/sobjects/SObjectName/deleted/?start=startDateAndTime&end=endDateAndTime

## **Formats**

JSON, XML

## **HTTP Method**

GET

## Authentication

Authorization: Bearer token

## **Parameters**

Parameter	Description
start	Starting date/time (Coordinated Universal Time (UTC)—not local—timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:30:15 is interpreted as 12:30:00 UTC). The date and time should be provided in ISO 8601 format:  YYYY-MM-DDThh:mm:ss+hh:mm. The date/time value for start must chronologically precede end. This parameter should be URL-encoded.
end	Ending date/time (Coordinated Universal Time (UTC)—not local—timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:35:15 is interpreted as 12:35:00 UTC). The date and time should be provided in ISO 8601 format:  YYYY—MM—DDT hh: mm: ss+hh: mm. This parameter should be URL-encoded

## **Response format**

Property	Туре	Description
deletedRecords	array	Array of deleted records that satisfy the start and end dates specified in the request. Each entry contains the record ID and the date and time the record was deleted in ISO 8601 format, using Coordinated Universal Time (UTC) timezone.
earliestDateAvailable	String	ISO 8601 format timestamp (Coordinated Universal Time (UTC)—not local—timezone) of the last physically deleted object.
latestDateCovered	String	ISO 8601 format timestamp (Coordinated Universal Time (UTC)—not local—time zone) of the last date covered in the request.

## Example

For an example of getting a list of deleted items, see Get a List of Deleted Records Within a Given Timeframe on page 58.

Reference SObject Get Updated

# SObject Get Updated

Retrieves the list of individual records that have been updated (added or changed) within the given timespan for the specified object. SObject Get Updated is available in API version 29.0 and later.

This resource is commonly used in data replication applications. Note the following considerations:

- Results are returned for no more than 30 days previous to the day the call is executed.
- Your client application can replicate any objects to which it has sufficient permissions. For example, to replicate all data for your organization, your client application must be logged in with "View All Data" access rights to the specified object. Similarly, the objects must be within your sharing rules.
- There is a limit of 600,000 IDs returned from this resource. If more than 600,000 IDs would be returned, EXCEEDED\_ID\_LIMIT is returned. You can correct the error by choosing start and end dates that are closer together.

See "Data Replication" in the SOAP API Developer Guide for additional details on data replication and data replication limits.

#### URI

/vXX.X/sobjects/SObjectName/updated/?start=startDateAndTime&end=endDateAndTime

## **Formats**

JSON, XML

#### **HTTP Method**

**GET** 

## Authentication

Authorization: Bearer token

## **Parameters**

Parameter	Description
start	Starting date/time (Coordinated Universal Time (UTC) time zone—not local—timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:30:15 is interpreted as 12:30:00 UTC). The date and time should be provided in ISO 8601 format:  YYYY-MM-DDThh:mm:ss+hh:mm. The date/time value for start must chronologically precede end. This parameter should be URL-encoded
end	Ending date/time (Coordinated Universal Time (UTC) time zone—not local—timezone) of the timespan for which to retrieve the data. The API ignores the seconds portion of the specified dateTime value (for example, 12:35:15 is interpreted as 12:35:00 UTC). The date and time should be provided in ISO 8601 format:  YYYY-MM-DDThh:mm:ss+hh:mm. This parameter should be URL-encoded

## Response format

Property	Туре	Description
ids	array	Array of updated records that satisfy the start and end dates specified in the request. Each entry contains the record ID.
latestDateCovered	String	ISO 8601 format timestamp (Coordinated Universal Time (UTC)—not local—time zone) of the last date covered in the request.

Reference SObject Named Layouts

## **Example**

For an example of getting a list of updated deleted items, see Get a List of Updated Records Within a Given Timeframe on page 59.

# **SObject Named Layouts**

Retrieves information about alternate named layouts for a given object.

## **Syntax**

## URI

/vXX.X/sobjects/Object/describe/namedLayouts/layoutName

## Available since release

31.0

## **Formats**

JSON, XML

## **HTTP** methods

**GET** 

#### Authentication

Authorization: Bearer token

## Request body

None

# Example

/services/data/v31.0/sobjects/User/describe/namedLayouts/UserAlt

This example retrieves information on the "UserAlt" named layout for User.

# Usage

Use this resource to get information on a named layout for a given object. You must provide a valid named layout name as part of the resource LIRI

To get a list of named layouts for a given object, use the SObject Describe resource and look for the "namedLayoutInfos" field in the response body.

# **SObject Rows**

Accesses records based on the specified object ID. Retrieves, updates, or deletes records. This resource can also be used to retrieve field values. Use the GET method to retrieve records or fields, the DELETE method to delete records, and the PATCH method to update records.

To create new records, use the SObject Basic Information resource.

#### URI

/vXX.X/sobjects/SObjectName/id/

## **Formats**

JSON, XML

#### **HTTP Method**

GET, PATCH, DELETE

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
fields	Optional list of fields used to return values for.

## Usage

This resource can be used with external objects in API version 32.0 and later.

• External objects that are associated with non-high-data-volume external data sources use the 18-character Salesforce ID for the *id*. Otherwise, external objects use the External ID standard field of the external object for the *id*.

## **Examples**

- For examples of retrieving field values using GET, see:
  - Get Field Values from a Standard Object Record on page 47
  - Get Field Values from an External Object Record by Using the External ID Standard Field on page 48
  - Get Field Values from an External Object Record by Using the Salesforce ID on page 47
- For an example of updating a record using PATCH, see Update a Record on page 46.
- For an example of deleting a record using DELETE, see Delete a Record on page 47.
- For an example of updating the blob data for an object, see Insert or Update Blob Data on page 73.

# SObject Rows by External ID

Creates new records or updates existing records (upserts records) based on the value of a specified external ID field.

- If the specified value doesn't exist, a new record is created.
- If a record does exist with that value, the field values specified in the request body are updated.
- If the value is not unique, the REST API returns a 300 response with the list of matching records.



Note: Do not specify Id or an external ID field in the request body or an error is generated.

## URI

/vXX.X/sobjects/SObjectName/fieldName/fieldValue

#### **Formats**

JSON, XML

#### **HTTP Method**

HEAD, GET, PATCH, DELETE, POST (see Usage section)

## Authentication

Authorization: Bearer token

#### **Parameters**

None

Reference SObject Blob Retrieve

#### Usage

As a special case, in API version 37.0 and later, you can use this resource to create a record by POSTing to /vXX.X/sobjects/**SObjectName**/Id. This pattern represents the use of Id as the specified external ID field and null as the value. It's useful when you're writing code to upsert multiple records by different external IDs and you don't want to request a separate resource.

## **Examples**

- For an example of retrieving a record based on an external ID, see Retrieve a Record Using an External ID on page 48.
- For examples of creating and updating records based on external IDs, see Insert or Update (Upsert) a Record Using an External ID on page 49.

# **SObject Blob Retrieve**

Retrieves the specified blob field from an individual record.

#### URI

/vXX.X/sobjects/SObjectName/id/blobField

#### **Formats**

Because blob fields contain binary data, you can't use JSON or XML to retrieve this data.

#### **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

#### **Parameters**

none required

## Example

For an example of retrieving the blob data from an Attachment or Document, see Get Attachment Content from a Record on page 58

## **Error responses**

See Status Codes and Error Responses on page 277.

# **SObject ApprovalLayouts**

Returns a list of approval layouts for a specified object. Specify a particular approval process name to limit the return value to one specific approval layout. This resource is available in REST API version 30.0 and later.

# **Syntax**

#### URI

To get an approval layout description for a specified object, use

/vXX.X/sobjects/**SObjectName**/describe/approvalLayouts/

To get an approval layout description for a particular approval process, use

/vXX.X/sobjects/SObjectName/describe/approvalLayouts/approvalProcessName

## **Formats**

JSON, XML

Reference SObject CompactLayouts

#### **HTTP** methods

HEAD, GET

#### Authentication

Authorization: Bearer token

## **Request parameters**

None required

## Example

## Getting all approval layouts for an sObject

```
curl
https://yourInstance.salesforce.com/services/data/v30.0/sobjects/Account/describe/approvalLayouts/
-H "Authorization: Bearer token"
```

## **Example JSON Response body**

```
"approvalLayouts" : [ {
    "id" : "04aD00000008Py9IAE",
    "label" : "MyApprovalProcessName",
    "layoutItems" : [...],
    "name" : "MyApprovalProcessName"
    }, {
    "id" : "04aD00000008Q0KIAU",
    "label" : "Process1",
    "layoutItems" : [...],
    "name" : "Process1"
    }]
}
```

If you haven't defined any approval layouts for an object, the response is { "approvalLayouts": []}.

## Getting the approval layout for a particular approval process

```
\label{local_curl_stance} $$  \text{top://yourInstance.salesforce.com/services/data/v30.0/sobjects/Account/describe/approvalIayouts/MyApprovalProcessName} $$ -H "Authorization: Bearer $token"$
```

## **Example JSON Response body**

```
"approvalLayouts" : [ {
    "id" : "04aD00000008Py9IAE",
    "label" : "MyApprovalProcessName",
    "layoutItems" : [...],
    "name" : "MyApprovalProcessName"
} ]
```

# **SObject CompactLayouts**

Returns a list of compact layouts for a specific object. This resource is available in REST API version 29.0 and later.

Reference SObject CompactLayouts

## **Syntax**

#### URI

For a compact layout description for a specific object, use /vXX.X/sobjects/Object/describe/compactLayouts/

#### **Formats**

JSON, XML

## **HTTP** methods

HEAD, GET

## Authentication

Authorization: Bearer token

## **Request parameters**

None required

## Example

## **Getting compact layouts**

/services/data/v29.0/sobjects/Account/describe/compactLayouts

## **Example JSON Response body**

This sample JSON response is for compact layouts created on the Account object. In this example, only one custom compact layout was created for Account. The custom compact layout is assigned as the primary compact layout for the object, and it contains two fields: Account Name and Phone.

```
"compactLayouts" : [ {
 "actions" : [ {
   "custom" : false,
   "icons" : null,
   "label" : "Call",
   "name" : "CallHighlightAction"
 }, {
    "custom" : false,
   "icons" : null,
   "label" : "Send Email",
   "name" : "EmailHighlightAction"
 }, {
    "custom" : false,
   "icons" : null,
   "label" : "Map",
   "name" : "MapHighlightAction"
    "custom" : false,
   "icons" : null,
   "label" : "Read News",
   "name" : "NewsHighlightAction"
    "custom" : false,
   "icons" : null,
   "label" : "View Website",
```

```
"name" : "WebsiteHighlightAction"
} ],
"fieldItems" : [ {
  "editable" : false,
  "label" : "Account Name",
  "layoutComponents" : [ {
    "components" : [ ],
    "details" : {
      "autoNumber" : false,
      "byteLength" : 765,
      "calculated" : false,
      "calculatedFormula" : null,
      "cascadeDelete" : false,
      "caseSensitive" : false,
      "controllerName" : null,
      "createable" : true,
      "custom" : false,
      "defaultValue" : null,
      "defaultValueFormula" : null,
      "defaultedOnCreate" : false,
      "dependentPicklist" : false,
      "deprecatedAndHidden" : false,
      "digits" : 0,
      "displayLocationInDecimal" : false,
      "externalId" : false,
      "extraTypeInfo" : null,
      "filterable" : true,
      "groupable" : true,
      "htmlFormatted" : false,
      "idLookup" : false,
      "inlineHelpText" : null,
      "label" : "Account Name",
      "length" : 255,
      "mask" : null,
      "maskType" : null,
      "name" : "Name",
      "nameField" : true,
      "namePointing" : false,
      "nillable" : false,
      "permissionable" : false,
      "picklistValues" : [ ],
      "precision" : 0,
      "queryByDistance" : false,
      "referenceTo" : [ ],
      "relationshipName" : null,
      "relationshipOrder" : null,
      "restrictedDelete" : false,
      "restrictedPicklist" : false,
      "scale" : 0,
      "soapType" : "xsd:string",
      "sortable" : true,
      "type" : "string",
      "unique" : false,
      "updateable" : true,
```

```
"writeRequiresMasterRead" : false
    },
    "displayLines" : 1,
    "tabOrder" : 2,
    "type" : "Field",
    "value" : "Name"
  } ],
  "placeholder" : false,
  "required" : false
}, {
  "editable" : false,
  "label" : "Phone",
  "layoutComponents" : [ {
    "components" : [ ],
    "details" : {
      "autoNumber" : false,
      "byteLength" : 120,
      "calculated" : false,
      "calculatedFormula" : null,
      "cascadeDelete" : false,
      "caseSensitive" : false,
      "controllerName" : null,
      "createable" : true,
      "custom" : false,
      "defaultValue" : null,
      "defaultValueFormula" : null,
      "defaultedOnCreate" : false,
      "dependentPicklist" : false,
      "deprecatedAndHidden" : false,
      "digits" : 0,
      "displayLocationInDecimal" : false,
      "externalId" : false,
      "extraTypeInfo" : null,
      "filterable" : true,
      "groupable" : true,
      "htmlFormatted" : false,
      "idLookup" : false,
      "inlineHelpText" : null,
      "label" : "Account Phone",
      "length" : 40,
      "mask" : null,
      "maskType" : null,
      "name" : "Phone",
      "nameField" : false,
      "namePointing" : false,
      "nillable" : true,
      "permissionable" : true,
      "picklistValues" : [],
      "precision" : 0,
      "queryByDistance" : false,
      "referenceTo" : [ ],
      "relationshipName" : null,
      "relationshipOrder" : null,
      "restrictedDelete" : false,
```

```
"restrictedPicklist" : false,
      "scale" : 0,
      "soapType" : "xsd:string",
      "sortable" : true,
      "type" : "phone",
      "unique" : false,
      "updateable" : true,
      "writeRequiresMasterRead" : false
    },
   "displayLines" : 1,
   "tabOrder" : 3,
   "type" : "Field",
   "value" : "Phone"
 "placeholder" : false,
 "required" : false
"id" : "0AHD000000000Ab0AI",
"imageItems" : [ {
 "editable" : false,
 "label" : "Photo URL",
  "layoutComponents" : [ {
    "components" : [ ],
   "details" : {
     "autoNumber" : false,
      "byteLength" : 765,
      "calculated" : false,
      "calculatedFormula" : null,
      "cascadeDelete" : false,
      "caseSensitive" : false,
      "controllerName" : null,
      "createable" : false,
      "custom" : false,
      "defaultValue" : null,
      "defaultValueFormula" : null,
      "defaultedOnCreate" : false,
      "dependentPicklist" : false,
      "deprecatedAndHidden" : false,
      "digits" : 0,
      "displayLocationInDecimal" : false,
      "externalId" : false,
      "extraTypeInfo" : "imageurl",
      "filterable" : true,
      "groupable" : true,
      "htmlFormatted" : false,
      "idLookup" : false,
      "inlineHelpText" : null,
      "label" : "Photo URL",
      "length" : 255,
      "mask" : null,
      "maskType" : null,
      "name" : "PhotoUrl",
      "nameField" : false,
      "namePointing" : false,
```

Reference Describe Layouts

```
"nillable" : true,
          "permissionable" : false,
          "picklistValues" : [ ],
          "precision" : 0,
          "queryByDistance" : false,
          "referenceTo" : [ ],
          "relationshipName" : null,
          "relationshipOrder" : null,
          "restrictedDelete" : false,
          "restrictedPicklist" : false,
          "scale" : 0,
          "soapType" :
          "xsd:string",
          "sortable" : true,
          "type" : "url",
          "unique" : false,
          "updateable" : false,
          "writeRequiresMasterRead" : false
        },
        "displayLines" : 1,
        "tabOrder" : 1,
        "type" : "Field",
        "value" : "PhotoUrl"
      } ],
      "placeholder" : false,
      "required" : false
    "label" : "Custom Account Compact Layout",
    "name" : "Custom Account Compact Layout"
  "defaultCompactLayoutId" : "OAHD000000000AbOAI",
  "recordTypeCompactLayoutMappings" : [ {
    "available" : true,
    "compactLayoutId" : "OAHD000000000AbOAI",
    "compactLayoutName" : "Custom_Account_Compact_Layout",
    "recordTypeId" : "012000000000000AAA",
    "recordTypeName" : "Master",
    "urls" : {
      "compactLayout" :
"/services/data/v31.0/sobjects/Account/describe/compactLayouts/01200000000000AAA"
  } ],
  "urls" : {
   "primary": "/services/data/v31.0/sobjects/Account/describe/compactLayouts/primary"
}
```

If you haven't defined any compact layouts for an object, the compactLayoutId returns as Null.

# **Describe Layouts**

Returns a list of layouts and descriptions. The list of fields and the layout name are returned.

Reference Describe Layouts

#### URI

To return descriptions of global publisher layouts, the URI is: /vXX.X/sobjects/Global/describe/layouts/

For a layout description for a specific object, use /vXX.X/sobjects/**Object**/describe/layouts/

For a layout description for objects that have more than one record type defined, use

/vXX.X/sobjects/ $\it{Object}$ /describe/layouts/<recordTypeId>. For example, the following URI requests the layout for a specific Contact record ID: /vXX.X/sobjects/Contact/describe/layouts/012xx00000024x5AAA. A GET request for objects that have more than one record type defined returns null for the layouts section of the response.

#### **Formats**

JSON, XML

#### **HTTP Method**

HEAD, GET

#### Authentication

Authorization: Bearer token

#### **Parameters**

None required

## **Example for getting global publisher layouts**

```
curl
https://yourInstance.salesforce.com/services/data/v35.0/sobjects/Global/describe/layouts/
-H "Authorization: Bearer token"
```

## Example JSON Response body contactlayout. json file

```
[ { "name" : "contactlayout",
    "searchColumns" : [ { "field" : "Account.Name",
         "format" : null,
          "label" : "Account Name",
         "name" : "Name"
       } ,
        { "field" : "Account.Site",
         "format" : null,
          "label" : "Account Site",
          "name" : "Site"
       },
        { "field" : "Account.Phone",
         "format" : null,
         "label" : "Phone",
          "name" : "Phone"
       },
        { "field" : "User.Alias",
         "format" : null,
          "label" : "Account Owner Alias",
          "name" : "Owner.Alias"
       }
     1
 { "label" : "Search Results",
   "limitRows" : 25,
   "searchColumns" : [ { "field" : "Contact.Name",
         "format" : null,
```

Reference Describe Layouts

```
"label" : "Name",
       "name" : "Name"
     },
      { "field" : "Account.Name",
        "format" : null,
       "label" : "Account Name",
       "name" : "Account.Name"
      { "field" : "Account.Site",
       "format" : null,
       "label" : "Account Site",
       "name" : "Account.Site"
     } ,
      { "field" : "Contact.Phone",
       "format" : null,
       "label" : "Phone",
       "name" : "Phone"
     },
      { "field" : "Contact.Email",
       "format" : null,
       "label" : "Email",
       "name" : "Email"
     } ,
      { "field" : "User.Alias",
       "format" : null,
       "label" : "Contact Owner Alias",
        "name" : "Owner.Alias"
     }
   ]
},
{ "label" : "Search Results",
 "limitRows" : 25,
  "searchColumns" : [ { "field" : "Lead.Name",
       "format" : null,
       "label" : "Name",
        "name" : "Name"
     },
      { "field" : "Lead.Title",
       "format" : null,
       "label" : "Title",
       "name" : "Title"
     },
      { "field" : "Lead.Phone",
       "format" : null,
        "label" : "Phone",
        "name" : "Phone"
     } ,
      { "field" : "Lead.Company",
       "format" : null,
       "label" : "Company",
        "name" : "Company"
     },
      { "field" : "Lead.Email",
       "format" : null,
```

Reference SObject PlatformAction

# **SObject PlatformAction**

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

Returns the description of the PlatformAction.

## **Syntax**

#### UR

Use /services/data/vXX.X/sobjects/PlatformAction

## Available since release

This resource is available in API version 33.0 and later.

#### **Formats**

JSON, XML

## **HTTP** methods

**GET** 

## **Authentication**

```
Authorization: Bearer token
```

#### Request body

None.

# Usage

The only thing you can do with this resource is Query it.

# **SObject Quick Actions**

Returns a list of actions and their details. This resource is available in REST API version 28.0 and later. When working with actions, also refer to Quick Actions.

Reference SObject Quick Actions

#### URI

To return a specific object's actions as well as global actions, use: /vXX.X/sobjects/object/quickActions/

To return a specific action, use /vXX.X/sobjects/object/quickActions/{action name}

To return a specific action's descriptive detail, use /vXX.X/sobjects/object/quickActions/{action name}/describe/

To return a specific action's default values, including default field values, use

services/data/vXX.X/sobjects/object/quickActions/{action name}/defaultValues/

In API version 28.0, to evaluate the default values for an action, use  $vXX.X/sobjects/object/quickActions/{action name}/defaultValues/{parent id}$ 

In API version 29.0 and greater, to evaluate the default values for an action, use

vXX.X/sobjects/object/quickActions/{action name}/defaultValues/{context id}

This returns the default values specific to the {context id} object.

#### **Formats**

JSON, XML

## **HTTP Method**

HEAD, GET, POST

#### Authentication

```
Authorization: Bearer token
```

#### **Parameters**

None required

## **Example for getting Account actions**

```
curl
https://yourInstance.salesforce.com/services/data/v28.0/sobjects/Account/quickActions
-H "Authorization: Bearer token"
```

## Example for creating a contact on Account using an action

```
curl
https://yourInstance.salesforce.com/services/data/v28.0/sobjects/Account/quickActions/CreateContact
   -H 'Authorization: Bearer access_token -H "Content-Type: application/json" -d
@newcontact.json'
```

## Example JSON request body newcontact. json file

```
"contextId" : "001D000000JRSGf",
   "record" : { "LastName" : "Smith" }
}
```

## Considerations

• The resources return all actions—both global and standard—in addition to the ones requested.

# SObject Rich Text Image Retrieve

Retrieves the specified image data from a specific rich text area field in a given record.

## URI

/vXX.X/sobjects/SObjectName/id/richTextImageFields/fieldName/contentReferenceId

#### **Formats**

PNG or JPEG binary image data.

## **HTTP Method**

**GET** 

## Authentication

Authorization: Bearer token

## **Parameters**

Parameter	Description	
SObjectName	Indicates the name of the standard object of the record.	
id	The ID of the object.	
fieldName	The name of the rich text area field.	
contentReferenceId	The reference ID that uniquely identifies an image within a rich text area field.	
	You can obtain the reference by retrieving information for the object. The description will show the contents of the rich text area field. For example:	
	<pre>"attributes" : {     "type" : "Lead",     "url" : "/services/data/v43.0/sobjects/Lead/00QRM000003ZfDb2AK"     },     "Id" : "00QRM000003ZfDb2AK",      "ContactPhotoc" :     "Sarah Loehr and her two dogs.     <img alt='\"Sarah' eid="00QRM000003ZfDb&amp;" feoid="00NRM000001E73j&amp;" loehr.\"="" refid='0EMRM00000002Ip\"' src='\"https://c.na44.content.stmfa.stm.force.com/servlet/rtaImage?'/>" }</pre>	
	The refid parameter of the image (OEMRM00000002Ip in this example) is the contentReferenceId.	

## Example

For an example of retrieving the blob data from a rich text area field, see Get an Image from a Rich Text Area Field on page 40.

Reference SObject Relationships

#### **Error responses**

See Status Codes and Error Responses on page 277.

# **SObject Relationships**

Accesses records by traversing object relationships via friendly URLs. You can retrieve, update, or delete the record associated with the traversed relationship field. If there are multiple related records, you can retrieve the complete set of associated records. This resource is available in REST API version 36.0 and later.

#### URI

/vXX.X/sobjects/SObject/id/relationship field name

#### **Formats**

JSON, XML

#### **HTTP Methods**

GET, PATCH, DELETE

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
fields	Optional for GET. A list of fields in the associated relationship record to return. Fields are separated by commas. For example:
	/services/data/v36.0/sobjects/ <b>SObject</b> /id/relationship field?fields=field1, field2

## **Response Body**

For retrievals via GET, the response body is the contents of the record associated with the relationship field. Here is an example of a request and JSON response body for a simple relationship traversal that returns the Distributor\_\_c record associated with a relationship field on custom object Merchandise\_\_c.

https://yourInstance.salesforce.com/services/data/v47.0/sobjects/Merchandise\_c/a01D000000INjVe/Distributor\_r

```
"attributes" :

{
    "type" : "Distributor__c",
    "url" : "/services/data/v36.0/sobjects/Distributor__c/a03D0000003DUhcIAG"
},

"Id" : "a03D00000003DUhcIAG",
"OwnerId" : "005D0000001KyEIIAO",
"IsDeleted" : false,
"Name" : "Distributor1",
"CreatedDate" : "2011-12-16T17:43:01.000+0000",
"CreatedById" : "005D0000001KyEIIAO",
"LastModifiedDate" : "2011-12-16T17:43:01.000+0000",
"LastModifiedById" : "005D00000001KyEIIAO",
```

```
"SystemModstamp" : "2011-12-16T17:43:01.000+0000",

"Location__c" : "San Francisco"
}
```

A response body isn't returned for updates via PATCH and deletions via DELETE.

## **Error Responses**

If no record is associated with a relationship field, a 404 error response is returned. If the relationship field normally resolves to multiple records and no relationship set exists, a 200 response is returned. If the fields parameter is used with fields that don't exist or aren't visible to the consumer by field-level security, a 400 error response is returned. For other error messages, see Status Codes and Error Responses on page 277.

#### **Example**

For examples of using SObject Relationships to access relationship fields, see Traverse Relationships with Friendly URLs on page 53.

# **SObject Suggested Articles**

Returns a list of suggested Salesforce Knowledge articles for a case, work order, or work order line item.

## **Syntax**

#### URI

To return suggested articles for a case, work order, or work order line item that is being created, use vXX.X/sobjects/**SObject**/suggestedArticles?language=**article** 

language&subject=subject&description=description. The SObject can be Case, WorkOrder, or
WorkOrderLineItem. Suggestions are based on common keywords in the title, description, and other information that's
entered before the record has been saved and assigned an ID.

For example: vXX.X/sobjects/*Case*/suggestedArticles?language=*article* 

language&subject=subject&description=description Or

vXX.X/sobjects/WorkOrder/suggestedArticles?language=article

language & subject = subject & description = description.

To return suggested articles for an existing record with an ID, use

vXX.X/sobjects/SObject/ID/suggestedArticles?language=article language

## Available since release

30.0

### **Formats**

JSON, XML

## **HTTP** methods

**GET** 

#### Authentication

```
Authorization: Bearer token
```

## Request body

None required

## **Request parameters**

Parameter	Description
articleTypes	Optional. Three-character ID prefixes indicating the desired article types. You can specify multiple values for this parameter in a single REST call, by repeating the parameter name for each value. For example, articleTypes=ka0&articleTypes=ka1.
categories	Optional. The name of the data category group and the data category API name (not category title) for desired articles. The syntax is categories={"Group":"Category"}. Characters in the URL might need to be encoded. For example:
	categories=%7B%22Regions%22%3A%22Asia%22%2C%22 Products%22%3A%22Laptops%22%7D
	The same data category group can't be specified more than once. However, you can specify multiple data category group and data category pairs. For example, categories={"Regions":"Asia", "Products":"Laptops"}.
description	Text of the description. Valid only for new records without an existing ID and required if subject is null. Article suggestions are based on common keywords in the subject, description, or both.
language	Required. Language that the article is written in.
limit	Optional. Specifies the maximum number of suggested articles to return.
publishStatus	Optional. The article's publication status. Valid values:  • Draft-Not published  • Online-Published in Salesforce Knowledge  • Archived
subject	Text of the subject. Valid only for new records without an existing ID and required if description is null. Article suggestions are based on common keywords in the subject, description, or both.
topics	Optional. The topic of returned articles. For example: topics=outlook&topics=email.
validationStatus	Optional. The validation status of returned articles.

# Example for getting suggested articles for a case that's being created

curl

-H "Authorization: Bearer token"

Reference SObject User Password

# Example JSON response body

```
[ {
    "attributes" : {
        "type" : "KnowledgeArticleVersion",
        "url" : "/services/data/v30.0/sobjects/KnowledgeArticleVersion/ka0D00000004CcQ"
    "Id" : "ka0D00000004CcQ"
}, {
    "attributes" : {
        "type" : "KnowledgeArticleVersion",
        "url" : "/services/data/v30.0/sobjects/KnowledgeArticleVersion/ka0D00000004CXo"
    },
    "Id" : "ka0D000000004CXo"
} ]
```

## Usage

Salesforce Knowledge must be enabled in your organization. The user must have the "View Articles" permission enabled. The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.

Articles are suggested based on a relevance algorithm. The suggestedArticles resource is designed to get the IDs of articles relevant to a case, work order, or work order line item. It's intended to be used with other services that then use the IDs to get article data for display.

# **SObject User Password**

Set, reset, or get information about a user password. This resource is available in REST API version 24.0 and later.

#### URI

/vXX.X/sobjects/User/user ID/password

For managing passwords for self-service users, the URI is:

/vXX.X/sobjects/SelfServiceUser/self service user ID/password

#### **Formats**

JSON, XML

## **HTTP Method**

HEAD, GET, POST, DELETE

## Authentication

Authorization: Bearer token

## **Parameters**

None required

#### Example

For examples of getting password information, setting a password, and resetting a password, see Manage User Passwords on page 81.

#### Considerations

- If the session does not have permission to access the user information, an INSUFFICIENT\_ACCESS error will be returned.
- When using this resource to set a new password, the new password must conform to the password policies for the organization, otherwise you will get an INVALID\_NEW\_PASSWORD error response.

- You can only set one password per request.
- When you use the DELETE method of this resource, Salesforce will reset the user password to an auto-generated password, which will be returned in the response.

# Platform Event Schema by Event Name

Gets the definition of a platform event in JSON format for an event name.

# Syntax

URI

/v**XX.X**/sobjects/**Event\_Name**/eventSchema

Available since release

40.0

**Formats** 

**JSON** 

**HTTP** methods

GET

Authentication

Authorization: Bearer token

## **Parameters**

Parameter	Description
payloadFormat	(Optional query parameter. Available in API version 43.0 and later.) The format of the returned event schema. This parameter can take one of the following values.
	• EXPANDED—The JSON representation of the event schema, which is the default format when payloadFormat is not specified in API version 43.0 and later.
	• COMPACT—A format that adheres to the open-source Apache Avro specification for the record complex type (see Apache Avro Format). Subscribers use the compact schema format to deserialize compact events received in binary form.

## **Error Codes**

400 Bad Request	Description
In API version 43.0 and later	The request was formatted incorrectly—an invalid value was passed for the payloadFormat parameter in the URI.
In API version 42.0 and earlier	The request was formatted incorrectly—the payloadFormat parameter was passed in the URI but this API version doesn't support this parameter.

## Examples for API Version 43.0 and Later

This URI gets the schema of a platform event named Low\_Ink\_\_e. In API version 43.0 and later, the default response format is the JSON representation of the event schema.

```
/services/data/v43.0/sobjects/Low_Ink__e/eventSchema
```

Or:

```
/services/data/v43.0/sobjects/Low Ink e/eventSchema?payloadFormat=EXPANDED
```

The returned response for the expanded format looks like the following in API version 43.0.

```
{
 "name" : "Low Ink e",
 "namespace" : "com.sforce.eventbus",
 "type" : "expanded-record",
  "fields" : [ {
   "name" : "data",
   "type" : {
     "type" : "record",
     "name" : "Data",
     "namespace" : "",
     "fields" : [ {
        "name" : "schema",
        "type" : "string"
     }, {
        "name" : "payload",
        "type" : {
          "type" : "record",
          "name" : "Payload",
          "doc" : "",
          "fields" : [ {
            "name" : "CreatedDate",
            "type" : "string",
            "doc" : "CreatedDate:DateTime"
            "name" : "CreatedById",
            "type" : "string",
            "doc" : "CreatedBy:EntityId"
          }, {
            "name" : "Printer Model c",
            "type" : [ "null", "string" ],
            "doc" : "Data:Text",
            "default" : null
            "name" : "Serial Number c",
            "type" : [ "null", "string" ],
            "doc" : "Data:Text",
            "default" : null
          }, {
            "name" : "Ink Percentage c",
            "type" : [ "null", "double" ],
            "doc" : "Data:Double",
            "default" : null
          } ]
```

```
}
    }, {
      "name" : "event",
      "type" : {
        "type" : "record",
        "name" : "Event",
        "fields" : [ {
          "name" : "replayId",
          "type" : "long"
        } ]
      }
    } ]
  }
}, {
  "name" : "channel",
  "type" : "string"
} ]
```

To get the compact (Apache Avro) format, use the following URI.

```
/services/data/v43.0/sobjects/Low_Ink__e/eventSchema?payloadFormat=COMPACT
```

The returned response for the compact format looks like the following in API version 43.0.

```
"name" : "Low Ink e",
"namespace" : "com.sforce.eventbus",
"doc" : "43.0",
"type" : "record",
"fields" : [ {
 "name" : "CreatedDate",
 "type" : "long",
 "doc" : "CreatedDate:DateTime"
  "name" : "CreatedById",
  "type" : "string",
  "doc" : "CreatedBy:EntityId"
}, {
  "name" : "Printer Model c",
 "type" : [ "null", "string" ],
 "doc" : "Data:Text",
 "default" : null
}, {
  "name" : "Serial Number c",
  "type" : [ "null", "string" ],
 "doc" : "Data:Text",
  "default" : null
}, {
  "name" : "Ink Percentage c",
 "type" : [ "null", "double" ],
 "doc" : "Data:Double",
 "default" : null
} ],
```

```
"uuid" : "5E5OtZj5_Gm6Vax9XMXH9A" }
```



**Note**: The compact schema doesn't include the replayId or channel fields because these fields are not necessary for deserializing the compact event received.

## Examples for API Version 42.0 and Earlier

In API version 42.0 and earlier, the response format adheres to the open-source Apache Avro specification for the record complex type.



**Note**: This format is what the API returns in API version 43.0 and later when appending the <code>?payloadFormat=COMPACT</code> parameter.

/services/data/v42.0/sobjects/Low Ink e/eventSchema

The returned response looks like the following in API version 42.0.

```
{
 "name" : "Low_Ink__e",
 "namespace" : "com.sforce.eventbus",
 "doc" : "42.0",
  "type" : "record",
 "fields" : [ {
   "name" : "CreatedDate",
   "type" : "long",
    "doc" : "CreatedDate:DateTime"
 }, {
    "name" : "CreatedById",
   "type" : "string",
   "doc" : "CreatedBy:EntityId"
    "name" : "Printer_Model__c",
   "type" : [ "null", "string" ],
   "doc" : "Data:Text",
    "default" : null
  }, {
    "name" : "Serial Number__c",
   "type" : [ "null", "string" ],
   "doc" : "Data:Text",
   "default" : null
 }, {
    "name" : "Ink_Percentage__c",
   "type" : [ "null", "double" ],
    "doc" : "Data:Double",
    "default" : null
  "uuid" : "5E5OtZj5 Gm6Vax9XMXH9A"
```

Note: When you change the definition of a platform event, the schema ID for this platform event changes.

# Apache Avro Format

The fields in the returned response adhere to the open-source Apache Avro specification for the record complex type (see Avro Records in the Apache Avro specification). Note the following:

- name is the name of the platform event.
- namespace corresponds to com.sforce.eventbus.
- type is the Avro complex type.
- fields is a JSON array containing the fields of the platform event. For each field, type indicates that the field can be either null or have a value of the specified type. When the field is not present, the value is default. Also, doc is the field data type. The data type information is used to convert DateTime fields from long to DateTime.

The response also includes the uuid field, which contains the schema's ID. The ID is the MD5 fingerprint of the normalized Avro schema encoded as a base-64 URL variant. You can append this ID to the /vxx.x/event/eventSchema/URI to retrieve the schema.

# Platform Event Schema by Schema ID

Gets the definition of a platform event in JSON format for a schema ID.

# **Syntax**

URI

/vXX.X/event/eventSchema/Schema ID

Available since release

40.0

**Formats** 

**JSON** 

**HTTP** methods

**GET** 

**Authentication** 

Authorization: Bearer token

## **Parameters**

Parameter	Description	
payloadFormat	(Optional query parameter. Available in API version 43.0 and later.) The format of the returned ever schema. This parameter can take one of the following values.	
	• EXPANDED—The JSON representation of the event schema, which is the default format when payloadFormat is not specified in API version 43.0 and later.	
	• COMPACT—A format that adheres to the open-source Apache Avro specification for the record complex type (see Apache Avro Format). Subscribers use the compact schema format to deserialize compact events received in binary form.	

## **Error Codes**

400 Bad Request	Description
In API version 43.0 and later	The request was formatted incorrectly—an invalid value was passed for the payloadFormat parameter in the URI.
In API version 42.0 and earlier	The request was formatted incorrectly—the payloadFormat parameter was passed in the URI but this API version doesn't support this parameter.

# Examples for API Version 43.0 and Later

This URI gets the schema of a platform event whose schema ID is 5E50tZj5\_Gm6Vax9XMXH9A. This schema ID is a sample ID. Replace it with a valid schema ID for your event.

```
/services/data/v43.0/event/eventSchema/5E5OtZj5_Gm6Vax9XMXH9A
```

Or:

```
/services/data/v43.0/event/eventSchema/5E5OtZj5 Gm6Vax9XMXH9A?payloadFormat=EXPANDED
```

In API version 43.0 and later, the response format is the JSON representation of the event schema by default. The returned response looks like the following in API version 43.0.

```
"name" : "Low Ink e",
"namespace": "com.sforce.eventbus",
"type" : "expanded-record",
"fields" : [ {
  "name" : "data",
  "type" : {
    "type" : "record",
    "name" : "Data",
    "namespace" : "",
    "fields" : [ {
      "name" : "schema",
      "type" : "string"
    }, {
      "name" : "payload",
      "type" : {
        "type" : "record",
        "name" : "Payload",
        "doc" : "",
        "fields" : [ {
          "name" : "CreatedDate",
          "type" : "string",
          "doc" : "CreatedDate:DateTime"
          "name" : "CreatedById",
          "type" : "string",
          "doc" : "CreatedBy:EntityId"
          "name" : "Printer_Model__c",
```

```
"type" : [ "null", "string" ],
          "doc" : "Data:Text",
          "default" : null
        }, {
          "name" : "Serial Number c",
          "type" : [ "null", "string" ],
          "doc" : "Data:Text",
          "default" : null
        }, {
          "name" : "Ink Percentage c",
          "type" : [ "null", "double" ],
          "doc" : "Data:Double",
          "default" : null
        } ]
      }
    }, {
      "name" : "event",
      "type" : {
        "type" : "record",
        "name" : "Event",
        "fields" : [ {
          "name" : "replayId",
          "type" : "long"
        } ]
      }
    } ]
  }
}, {
  "name" : "channel",
  "type" : "string"
} ]
```

To get the compact (Apache Avro) format, use the following URI.

/services/data/v43.0/event/eventSchema/5E5OtZj5 Gm6Vax9XMXH9A?payloadFormat=COMPACT

The returned response for the compact format looks like the following in API version 43.0.

```
"name" : "Low_Ink__e",
"namespace" : "com.sforce.eventbus",
"doc" : "43.0",
"type" : "record",
"fields" : [ {
    "name" : "CreatedDate",
    "type" : "long",
    "doc" : "CreatedDate:DateTime"
}, {
    "name" : "CreatedById",
    "type" : "string",
    "doc" : "CreatedBy:EntityId"
}, {
    "name" : "Printer_Model__c",
    "type" : [ "null", "string" ],
```

```
"doc" : "Data:Text",
   "default" : null
}, {
   "name" : "Serial_Number__c",
   "type" : [ "null", "string" ],
   "doc" : "Data:Text",
   "default" : null
}, {
   "name" : "Ink_Percentage__c",
   "type" : [ "null", "double" ],
   "doc" : "Data:Double",
   "default" : null
} ],
   "uuid" : "5E5OtZj5_Gm6Vax9XMXH9A"
}
```

Note: The compact schema doesn't include the replayId or channel fields because these fields are not necessary for deserializing the compact event received.

## Example for API Version 42.0 and Earlier

In API version 42.0 and earlier, the response format adheres to the open-source Apache Avro specification for the record complex type.



**Note:** This format is what the API returns in API version 43.0 and later when appending the <code>?payloadFormat=COMPACT</code> parameter.

This URI gets the schema of a platform event whose schema ID is 5E50tZj5\_Gm6Vax9XMXH9A. This schema ID is a sample ID. Replace it with a valid schema ID for your event.

/services/data/v42.0/event/eventSchema/5E5OtZj5 Gm6Vax9XMXH9A

The returned response looks like the following in API version 42.0.

```
"name" : "Low_Ink__e",
"namespace" : "com.sforce.eventbus",
"doc" : "42.0",
"type" : "record",
"fields" : [ {
  "name" : "CreatedDate",
  "type" : "long",
  "doc" : "CreatedDate:DateTime"
}, {
  "name" : "CreatedById",
  "type" : "string",
  "doc" : "CreatedBy:EntityId"
  "name" : "Printer Model c",
  "type" : [ "null", "string" ],
  "doc" : "Data:Text",
  "default" : null
  "name" : "Serial Number c",
  "type" : [ "null", "string" ],
  "doc" : "Data:Text",
```

```
"default" : null
}, {
    "name" : "Ink_Percentage__c",
    "type" : [ "null", "double" ],
    "doc" : "Data:Double",
    "default" : null
} ],
    "uuid" : "5E5OtZj5_Gm6Vax9XMXH9A"
}
```

**M** 

Note: When you change the definition of a platform event, the schema ID for this platform event changes.

If you don't have the schema ID, you can get the schema by supplying the platform event name. Make a GET request to /v**XX.X**/sobjects/**Event Name**/eventSchema. See Platform Event Schema by Event Name.

## Apache Avro Format

The fields in the returned response adhere to the open-source Apache Avro specification for the record complex type (see Avro Records in the Apache Avro specification). Note the following:

- name is the name of the platform event.
- namespace corresponds to com.sforce.eventbus.
- type is the Avro complex type.
- fields is a JSON array containing the fields of the platform event. For each field, type indicates that the field can be either null or have a value of the specified type. When the field is not present, the value is default. Also, doc is the field data type. The data type information is used to convert DateTime fields from long to DateTime.

The response also includes the uuid field, which contains the schema's ID. The ID is the MD5 fingerprint of the normalized Avro schema encoded as a base-64 URL variant. You can append this ID to the /vxx.x/event/eventSchema/URI to retrieve the schema.

# **AppMenu**

Returns a list of items in either the Salesforce app drop-down menu or the Salesforce for Android, iOS, and mobile web navigation menu.

## **Syntax**

#### URI

To return a list of the Salesforce app drop-down menu items, the URI is: /vXX.X/appMenu/AppSwitcher/

To return a list of the Salesforce for Android, iOS, and mobile web navigation menu items, the URI is: /vXX.X/appMenu/Salesforce1/

#### Available since release

290

#### **Formats**

JSON, XML

#### **HTTP** methods

GET, HEAD

## Authentication

Authorization: Bearer token

### Request body

None

### **Request parameters**

None required

## Example

### Getting appMenu types

```
curl https://yourInstance.salesforce.com/services/data/v29.0/appMenu/ -H "Authorization: Bearer token"
```

## Example response body for /vXX.X/appMenu/AppSwitcher/

```
"appMenuItems" : [ {
 "type" : "Tabset",
  "content" : null,
 "icons" : null,
 "colors" : null,
  "label" : "Sales",
  "url" : "/home/home.jsp?tsid=02uxx00000056Sq"
  "type" : "Tabset",
  "content" : null,
  "icons" : null,
  "colors" : null,
  "label" : "Service",
  "url" : "/home/home.jsp?tsid=02uxx00000056Sr"
  "type" : "Tabset",
 "content" : null,
 "icons" : null,
  "colors" : null,
  "label" : "Marketing",
  "url" : "/home/home.jsp?tsid=02uxx00000056St"
}, {
  "type" : "Tabset",
  "content" : null,
  "icons" : null,
  "colors" : null,
  "label" : "Salesforce Chatter",
  "url" : "/home/home.jsp?tsid=02uxx00000056Su"
  "type" : "Tabset",
 "content" : null,
 "icons" : null,
  "colors" : null,
  "label" : "Community",
  "url" : "/home/home.jsp?tsid=02uxx00000056Sw"
  "type" : "Tabset",
  "content" : null,
```

```
"icons" : null,
  "colors" : null,
  "label" : "App Launcher",
  "url" : "/app/mgmt/applauncher/appLauncher.apexp?tsid=02uxx00000056Sx"
} ]
}
```

## Example response body for /vXX.X/appMenu/Salesforce1/

```
"appMenuItems" : [ {
 "type": "Standard.Search",
 "content" : null,
  "icons" : null,
  "colors" : null,
 "label" : "Smart Search Items",
 "url" : "/search"
 "type" : "Standard.MyDay",
 "content" : null,
 "icons" : null,
 "colors" : null,
  "label" : "Today",
  "url" : "/myDay"
}, {
  "type": "Standard. Tasks",
 "content" : null,
  "icons" : null,
  "colors" : null,
 "label" : "Tasks",
 "url" : "/tasks"
  "type" : "Standard.Dashboards",
 "content" : null,
 "icons" : null,
  "colors" : null,
  "label" : "Dashboards",
  "url" : "/dashboards"
  "type" : "Tab.flexiPage",
  "content" : "MySampleFlexiPage",
  "icons" : [ {
    "contentType" : "image/png",
    "width" : 32,
    "height" : 32,
    "theme" : "theme3",
    "url" : "http://myorg.com/img/icon/custom51 100/bel132.png"
    "contentType" : "image/png",
    "width" : 16,
    "height" : 16,
    "theme" : "theme3",
    "url" : "http://myorg.com/img/icon/custom51 100/bell16.png"
  }, {
    "contentType" : "image/svg+xml",
```

```
"width" : 0,
    "height" : 0,
    "theme" : "theme4",
    "url" : "http://myorg.com/img/icon/t4/custom/custom53.svg"
 }, {
    "contentType" : "image/png",
    "width" : 60,
    "height" : 60,
    "theme" : "theme4",
    "url" : "http://myorg.com/img/icon/t4/custom/custom53 60.png"
  }, {
    "contentType" : "image/png",
    "width" : 120,
    "height" : 120,
    "theme" : "theme4",
    "url" : "http://myorq.com/img/icon/t4/custom/custom53 120.png"
  "colors" : [ {
    "context" : "primary",
    "color" : "FC4F59",
    "theme" : "theme4"
  }, {
    "context" : "primary",
    "color" : "FC4F59",
    "theme" : "theme3"
  "label" : "My App Home Page",
  "url" : "/servlet/servlet.Integration?lid=01rxx0000000Vsd&ic=1"
}, {
  "type" : "Tab.apexPage",
  "content" : "/apex/myapexpage",
  "icons" : [ {
    "contentType" : "image/png",
    "width" : 32,
    "height" : 32,
    "theme" : "theme3",
    "url" : "http://myorg.com/img/icon/cash32.png"
  }, {
    "contentType" : "image/png",
    "width" : 16,
    "height" : 16,
    "theme" : "theme3",
    "url" : "http://myorg.com/img/icon/cash16.png"
    "contentType" : "image/svg+xml",
    "width" : 0,
    "height" : 0,
    "theme": "theme4",
    "url" : "http://myorg.com/img/icon/t4/custom/custom41.svg"
  }, {
    "contentType" : "image/png",
    "width" : 60,
    "height" : 60,
    "theme" : "theme4",
```

Reference Compact Layouts

```
"url" : "http://myorg.com/img/icon/t4/custom/custom41 60.png"
}, {
  "contentType" : "image/png",
  "width" : 120,
  "height" : 120,
  "theme" : "theme4",
  "url" : "http://myorg.com/img/icon/t4/custom/custom41 120.png"
"colors" : [ {
  "context" : "primary",
  "color" : "3D8D8D",
  "theme" : "theme4"
  "context" : "primary",
  "color" : "3D8D8D",
  "theme" : "theme3"
"label" : "label",
"url" : "/servlet/servlet.Integration?lid=01rxx0000000Vyb&ic=1"
```

# **Compact Layouts**

Returns a list of compact layouts for multiple objects. This resource is available in REST API version 31.0 and later.

This resource returns the primary compact layout for a set of objects. The set of objects is specified using a query parameter. Up to 100 objects can be queried at once.



**Note:** PersonAccount isn't supported for bulk queries. If you want to get the primary compact layout for PersonAccount, get it directly from

/services/data/v31.0/sobjects/Account/describe/compactLayouts/primaryPersonAccount.

# **Syntax**

UR

/vXX.X/compactLayouts?q=object list

### Available since release

31.0

#### **Formats**

JSON, XML

#### **HTTP** methods

**GET** 

### **Authentication**

Authorization: Bearer token

Reference Compact Layouts

### **Request parameters**

Parameter	Description	
q	A comma-delimited list of objects. The primary compact layout for each object in this list will be returned in the response of this resource.	

# Example

## Request for getting compact layouts for multiple objects

```
/services/data/v31.0/compactLayouts?q=Account,Contact,CustomObj__c
```

## Response for compact layouts for multiple objects

```
"Account" : {
  "actions" : [ {
    "behavior" : null,
    "content" : null,
    "contentSource" : null,
    "custom" : false,
    "encoding" : null,
    "height" : null,
    "icons" : null,
    "label" : "Call",
    "menubar" : false,
    "name" : "CallHighlightAction",
    "overridden" : false,
    "resizeable" : false,
    "scrollbars" : false,
    "showsLocation" : false,
    "showsStatus" : false,
    "toolbar" : false,
    "url" : null,
    "width" : null,
    "windowPosition" : null
  },
  "id" : "0AHD000000000Ab0AI",
  "label" : "Custom Account Compact Layout",
  "name" : "Custom Account Compact Layout"
},
"Contact" : {
  "actions" : [ {
    "behavior" : null,
    "content" : null,
    "contentSource" : null,
    "custom" : false,
    "encoding" : null,
    "height" : null,
    "icons" : null,
    "label" : "Call",
```

```
"menubar" : false,
    "name" : "CallHighlightAction",
    "overridden" : false,
    "resizeable" : false,
    "scrollbars" : false,
    "showsLocation" : false,
    "showsStatus" : false,
    "toolbar" : false,
    "url" : null,
    "width" : null,
    "windowPosition" : null
  },
  "id" : null,
  "label" : "System Default",
  "name" : "SYSTEM"
"CustomObj__c" : {
  "actions" : [ {
    "behavior" : null,
    "content" : null,
    "contentSource" : null,
    "custom" : false,
    "encoding" : null,
    "height" : null,
    "icons" : null,
    "label" : "Call",
    "menubar" : false,
    "name" : "CallHighlightAction",
    "overridden" : false,
    "resizeable" : false,
    "scrollbars" : false,
    "showsLocation" : false,
    "showsStatus" : false,
    "toolbar" : false,
    "url" : null,
    "width" : null,
    "windowPosition" : null
 },
  "id" : null,
  "imageItems" : null,
  "label" : "System Default",
  "name" : "SYSTEM"
}
```

## Consent

Your users can store consent preferences in different locations and possibly inconsistently. You can locate your customers' preferences for consent across multiple records when using API version 44.0 and later. Tracking consent preferences helps you and your users respect the most restrictive requests.

Consent API aggregates consent settings across the Contact, Contact Point Type Consent, Data Use Purpose, Individual, Lead, Person Account, and User objects when the records have a lookup relationship. The Consent API can't locate records in which the email address field is protected by Platform Encryption.

The API returns consent details based on a single action, like email or track, or—starting with API version 45.0—the multiaction endpoint allows you to request multiple actions in a single API call.

# **Syntax**

#### URI

/services/data/vXX.0/consent/action/action?ids=list of Ids

/services/data/v $\boldsymbol{xx.0}$ /consent/multiaction?actions= $\boldsymbol{list\_of\_actions}$ &ids= $\boldsymbol{list\_of\_Ids}$  (Available in API version 45.0 and later.)

## Available since release

44.0

#### **Formats**

**JSON** 

### **HTTP** methods

**GET** 

#### **Authentication**

Authorization: Bearer token

#### Request body

None

### **Request parameters**

Parameter	Description
actions	Comma-separated list of proposed actions. This required parameter applies only to the multiaction endpoint.
	This parameter is available in API version 45.0 and later.
	If this is used, action cannot be used.
aggregatedConsent	Optional: true or false. aggregatedConsent is the same as aggregatedConsent=true. If true, one result is returned indicating whether to proceed or not, rather than a result for each ID. If any ID in the list returns false, the aggregated result is false.
datetime	Optional. The timestamp for which consent is determined. The value is converted to the UTC timezone and must be specified in ISO 8601 format. If not specified, defaults to the current date and time.
ids	Required. Comma-separated list of IDs. The ID can be the record ID or the email address listed on the record.
purpose	Optional. The reason for contacting a customer.
verbose	Optional: true or false. verbose is the same as verbose=true. Verbose responses are slower than non-verbose responses. See the examples for a verbose response.

#### Action

Allowed values are:

- email
- fax
- geotrack
- mailing
- phone
- portability
- process
- profile
- shouldForget
- social
- solicit
- storePiiElsewhere
- track
- web

If action is used, actions cannot be used.



**Note:** When you select *email* as the action, the API only aggregates consent for records that contain the same email address. If the record ID specified in the URI is associated with a record that contains a different email address, the consent settings of the associated record aren't included in the API response.

# **Examples**

### Simple URI structure

/services/data/v44.0/consent/action/track?ids=003xx000004TxyY,00Qxx00000syy0,003zz000004zzZ

#### **Multiaction URI structure**

/services/data/v44.0/consent/multiaction?actions=track, qeotrack, email&ids=003xx000008TiyY, 00gxx000008kw0, dek65@tf7h.com

## Email addresses as IDs, specified purpose and timespan, and a verbose response

/services/deta/v46.0/carsent/action/eneil?ids=j0t5t5520tkhyp5ia.com/4qxlsvo623vj7pvh.com6detetime=2018-12-12100:00:00Z6purpose-billing&erbose-true

### Response

```
{
  "j0t5t5b2@tkbxp5ia.com" : {
    "result" : "Success",
    "proceed" : {
        "email" : "true"
        "emailResult" : "Success"
    },
    "explanation" : [ {
        "objectConsulted" : "ContactTypePointConsent",
        "status" : "opt_in",
        "purpose" : "billing",
        "purpose" : "billing",
        "status" : "opt_in",
        "purpose" : "billing",
```

```
"recordId" : "003xx000004TxyY",
   "value" : "true"
 },{
    "objectConsulted" : "Contact",
   "field" : "HasOptedOutOfTracking",
   "recordId" : "1",
   "value" : "true"
 } ]
},
"4quxlswo@23wj7pwh.com" : {
 "result" : "Success",
  "proceed" : {
    "email" : "false"
    "emailResult" : "Success"
 },
  "explanation" : [ {
    "objectConsulted" : "Contact",
    "field" : "HasOptedOutOfEmail",
    "recordId" : "00Qxx00000skw0",
   "value" : "true"
 }
   ]
}
```

# Security

To call Consent API, you must have either the View All Data or the Allow User Access to Privacy Data user permission. Requiring a perm ensures that the System Administrator gives explicit permission. This API accesses org-wide consent data, such as links between records and the value of consent flags, not just records to which the user ordinarily has access.

## Usage

The following table shows how the API responses are determined. If the consulted fields find conflicting consent preferences, the response returns the least permissive preference. For example, if Contact. Has Opted Out Of Email is false, but Lead. Has Opted Out Of Email is true, the response indicates that you can't proceed with emailing the user.

When you select *email* as the action, the API only aggregates consent for records that contain the same email address. If the record ID specified in the URI is associated with a record that contains a different email address, the consent settings of the associated record aren't included in the API response.



Note: When the API compares consent settings across records, it doesn't incorporate settings from converted leads.

Action	Fields Consulted	API Response	Response Schema
email	Contact.HasOptedOutOfEmail ContactPointTypeConsentContactPointType ContactPointTypeConsentEffectiveFrom ContactPointTypeConsentEffectiveTo ContactPointTypeConsentPivacyConsentStatus	Within the time range if specified in ContactPointTypeConsent: Returns TRUE if all consulted field values are 0.	{  " <id email="">": {  "result": "<success errormessage="">",  "proceed": { "emailResult": "<success errormessage="">", email: "<true false="">" }</true></success></success></id>

	DataUsePurpose.Name Lead.HasOptedOutOfEmail PersonAccountHasOptedOutOfEmail	Returns FALSE if any consulted field value is 1 or if no related Contact, Contact Point Type Consent, Lead, or Person Account object exists.	<pre>} }</pre>
fax	Contact. Has Opted Out Of Fax  Data Use Purpose. Name  Lead. Has Opted Out Of Fax  Person Account. Has Opted Out Of Fax	Returns TRUE if all consulted field values are 0.  Returns FALSE if any consulted field value is 1 or if no related Contact, Lead, or Person Account object exists.	{   " <id email="">":   {   "result":"<success errormessage="">",   "proceed":{"faxResult":"<success errormessage="">", fax   : "<true false="">"} }</true></success></success></id>
geotrack	DataUsePurpose.Name Individual.HasOptedOutGeoTracking	Returns TRUE if the consulted field value is 0.  Returns FALSE if the consulted field value is 1 or if no related Individual object exists.	{   " <id email="">": {   "result": "<success errormessage="">",   "proceed": { "geotrackResult": "<success errormessage="">",   "geotrack": "<true false="">" } }</true></success></success></id>
mailing	ContactPointTypeConsentContactPointType ContactPointTypeConsentEffectiveFrom ContactPointTypeConsentEffectiveTo ContactPointTypeConsentPixayConsentStatus DataUsePurpose.Name	Within the time range if specified in ContactPointTypeConsent: Returns TRUE if all consulted field values are 0. Returns FALSE if any consulted field value is 1 or if no related Contact, Contact Point Type Consent, Lead, or Person Account object exists.	{  " <id email="">": {  "result": "<success errormessage="">",  "proceed": { "mailingResult": "<success errormessage="">",  "mailing": "<true false="">" } } }</true></success></success></id>
phone	Contact.DoNotCall	Within the time range if specified in ContactPointTypeConsent:	{ " <id email="">" :</id>

```
Returns TRUE if all
                 ContactPointTypeConsentContactPointType
                                                    consulted field values
                 ContactPointTypeConsent.EffectiveFrom
                                                                             "result": "<Success/errormessage>",
                 ContactPointTypeConsent.EffectiveTo
                                                                             "proceed": { "phoneResult": "<Success/errormessage>",
                                                    Returns FALSE if any
                                                                             "phone": "<true/false>"}
                 ContadPointTypeConsentPivacyConsentStatus
                                                   consulted field value is
                                                    1 or if no related
                 DataUsePurpose.Name
                                                    Contact, Contact Point }
                 Lead.DoNotCall
                                                    Type Consent, Lead, or
                 PersonAccount.DoNotCall
                                                    Person Account object
                                                    exists.
portability
                 DataUsePurpose.Name
                                                    Returns TRUE if the
                                                    consulted field value is
                                                                            "<ID/Email>":
                 Individual.SendIndividualData
                                                    1.
                                                    Returns FALSE if the
                                                                             "result": "<Success/errormessage>",
                                                    consulted field value is
                                                   0 or if no related
                                                                             "proceed": { "portabilityResult":
                                                   Individual object exists.
                                                                             "<Success/errormessage>", "portability": "<true/false>"
process
                 DataUsePurpose.Name
                                                    Returns TRUE if the
                                                    consulted field value is
                                                                             "<ID/Email>":
                 Individual.HasOptedOutProcessing
                                                   Returns FALSE if the
                                                                             "result": "<Success/errormessage>",
                                                    consulted field value is
                                                    1 or if no related
                                                                             "proceed": { "processResult": "< Success/errormessage>",
                                                   Individual object exists.
                                                                             "process": "<true/false>"}
profile
                 DataUsePurpose.Name
                                                    Returns TRUE if the
                                                    consulted field value is
                                                                             "<ID/Email>":
                 Individual.HasOptedOutProfiling
                                                    Returns FALSE if the
                                                                             "result": "<Success/errormessage>",
                                                    consulted field value is
                                                    1 or if no related
                                                                             "proceed": { "profileResult": "<Success/errormessage>",
                                                   Individual object exists.
                                                                             "profile": "<true/false>"}
```

shouldForget	Data Use Purpose. Name Individual. Should Forget	Returns TRUE if the consulted field value is 1.  Returns FALSE if the consulted field value is 0 or if no related Individual object exists.	{   " <id email="">":   {   "result": "<success errormessage="">",   "proceed": { "shouldForgetResult":   "<success errormessage="">", "shouldForget":   "<true false="">" } }</true></success></success></id>
social	ContactPointTypeConsentContactPointType ContactPointTypeConsentEffectiveFrom ContactPointTypeConsentEffectiveTo ContactPointTypeConsentPixacyConsentStatus DataUsePurpose.Name	Within the time range if specified in ContactPointTypeConsent: Returns TRUE if all consulted field values are 0. Returns FALSE if any consulted field value is 1 or if no related Contact, Contact Point Type Consent, Lead, or Person Account object exists.	{  " <id email="">": {  "result":"<success errormessage="">",  "proceed":{"socialResult":"<success errormessage="">",  "social":"<true false="">"} } }</true></success></success></id>
solicit	DataUsePurpose.Name Individual.HasOptedOutSolicit	Returns TRUE if the consulted field value is 0.  Returns FALSE if the consulted field value is 1 or if no related Individual object exists.	{   " <id email="">": {   "result":"<success errormessage="">",   "proceed":{"solicitResult":"<success errormessage="">",   "solicit":"<true false="">"} }</true></success></success></id>
storePIIEIsewhere	DataUsePurpose.Name Individual.CanStorePiiElsewhere	Returns TRUE if the consulted field value is 1.  Returns FALSE if the consulted field value is 0 or if no related Individual object exists.	{   " <id email="">":   {   "result": "<success errormessage="">",   "proceed": { "storePIIEIsewhereResult":   "<success errormessage="">", "storePIIEIsewhere":   "<true false="">" } }</true></success></success></id>

```
}
track
                 DataUsePurpose.Name
                                                    Returns TRUE if the
                                                    consulted field value is
                                                                              "<ID/Email>":
                 Individual.HasOptedOutTracking
                                                    Returns FALSE if the
                                                                              "result": "<Success/errormessage>",
                                                    consulted field value is
                                                    1 or if no related
                                                                              "proceed": { "trackResult": "<Success/errormessage>",
                                                    Individual object exists.
                                                                             "track": "<true/false>"}
                                                                              }
web
                 ContactPointTypeConsent.ContactPointType
                                                    Within the time range
                                                    if specified in
                                                                              "<ID/Email>":
                 ContactPointTypeConsent.EffectiveFrom
                                                    ContactPointTypeConsent:
                 ContactPointTypeConsent.EffectiveTo
                                                    Returns TRUE if all
                 ContadPointTypeConsentPivacyConsentStatus
                                                                              "result": "<Success/errormessage>",
                                                    consulted field values
                                                    are 0.
                                                                              "proceed": { "webResult": "<Success/errormessage>",
                 DataUsePurpose.Name
                                                                              "web": "<true/false>"}
                                                    Returns FALSE if any
                                                    consulted field value is
                                                    1 or if no related
                                                    Contact, Contact Point
                                                    Type Consent, Lead, or
                                                    Person Account object
                                                    exists.
```

# **Embedded Service Configuration Describe**

Retrieves the values for your Embedded Service deployment configuration, including the branding colors, font, and site URL.

## **Syntax**

#### HR

/services/data/vXX.X/support/embeddedservice/configuration/EmbeddedServiceConfigDeveloperName

### Available since release

45.0

#### **Formats**

JSON

#### **HTTP** methods

GET, HEAD

## Authentication

You must be logged in to the account that owns the EmbeddedServiceConfigDeveloperName you are querying.

### **Request parameters**

None

## Example

Retrieving the values for the Embedded Service deployment TestOne

```
/services/data/v45.0/support/embeddedservice/configuration/TestOne
```

JSON Response body

This sample JSON response is for an Embedded Chat deployment. The JSON response for another Embedded Service feature, such as an embedded flow, will have different values.

```
"embeddedServiceConfig" : {
 "areGuestUsersAllowed" : false,
 "authMethod" : "CustomLogin",
  "embeddedServiceBranding" : {
   "contrastInvertedColor" : "\#ffffff",
   "contrastPrimaryColor" : "#333333",
   "font" : "Salesforce Sans",
   "height" : 498,
   "navBarColor" : "#222222",
   "primaryColor" : "#222222",
   "secondaryColor" : "#005290",
   "width" : 320
  "embeddedServiceLiveAgent" : {
   "avatarImg" : "",
   "embeddedServiceQuickActions" : [ {
     "order" : 1,
      "quickActionDefinition" : "Snapins Case OfflineCaseQuickAction 08hRM00000000CC",
      "quickActionType" : "OfflineCase"
      "order" : 1,
      "quickActionDefinition": "Snapins Contact PrechatQuickAction 08hRM0000000RC",
      "quickActionType" : "Prechat"
   }, {
      "order" : 2,
     "quickActionDefinition" : "Snapins Case PrechatQuickAction 08hRM00000000RC",
      "quickActionType" : "Prechat"
    } ],
   "enabled" : true,
   "fontSize" : "Medium",
   "headerBackgroundImg" : "https://google.com/img/headerBgImgUrl.png",
   "isOfflineCaseEnabled" : true,
    "isQueuePositionEnabled" : true,
   "liveChatButton": "573RM0000004GGf",
   "liveChatDeployment": "572RM0000004CDV",
   "offlineCaseBackgroundImg": "https://google.com/img/offlineBgImgUrl.png",
   "prechatBackgroundImg" : "https://google.com/img/prechatBgImgUrl.png",
   "prechatEnabled" : true,
   "scenario" : "Service",
```

Reference Invocable Actions

```
"smallCompanyLogoImg" : "https://google.com/img/logoImgUrl.png",
    "waitingStateBackgroundImg" : "https://google.com/img/bgImgUrl.png"

},
    "shouldHideAuthDialog" : false,
    "siteUrl" : "https://snapins-15f082fb956-15fbc261d27.stmfa.stm.force.com/napili2"
}
```

## Invocable Actions

Represents a standard or custom invocable action.

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

This resource is available in REST API version 32.0 and later.

## **Syntax**

#### URI

Get a list of custom actions:

/vXX.X/actions

#### **Formats**

JSON, XML

#### **HTTP Methods**

GET, POST

#### Authentication

```
Authorization: Bearer token
```

#### **Parameters**

None

#### Example

Using GET to retrieve a list of general action types for the current organization:

```
/services/data/v32.0/actions
```

### **JSON Response body**

```
{
  "standard" : "/services/data/v32.0/actions/standard",
  "custom" : "/services/data/v32.0/actions/custom"
}
```

### Example

Using POST to send a simple email message:

```
/services/data/v32.0/actions/standard/emailSimple
```

## **JSON Request body**

```
"inputs" : [ {
    "emailAddresses" : "Fred@Acme.com",
    "emailSubject" : "Note",
    "emailBody" : "Message of the day.",
    "senderAddress" : "Bill@TheCloud.org"
} ]
}
```

## JSON Response body

```
"actionName" : "emailSimple",
"errors" : null,
"isSuccess" : true,
"outputValues" : null
}
```

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

Action	actionName value	
Flow	The flow name	
Apex	The class's invocable method name	
Quick action	<pre><object name="">.<quick action="" name=""> For a global quick action, there's no <object name="">. prefix.</object></quick></object></pre>	
Email alert	<pre><object name="">.<email alert="" name=""></email></object></pre>	

For more information about actions, see the Actions Developer Guide.

## Standard Invocable Actions

Returns the list of actions that can be statically invoked. You can also get basic information for each type of action.

This resource is available in REST API version 32.0 and later.

## **Syntax**

### URI

Get a list of standard actions:

/vXX.X/actions/standard

#### **Formats**

JSON

#### **HTTP Methods**

GET, HEAD, POST

#### Authentication

Authorization: Bearer token

#### **Parameters**

None

#### Notes

The Post to Chatter action supports the following features using a special format in the body post.

- @mentions using @ [<id>]
- Topics using #[<topicString>]

For example, the string Hi @[00500000000000] check this out #[some\_topic]. is stored appropriately as Hi @Joe, check this out #some topic. where "@Joe" and "#some\_topic" are links to the user and topic, respectively.

## **Examples**

## Retrieving a list of standard actions for the current organization

```
/services/data/v46.0/actions/standard
```

#### **JSON Response body**

```
"actions" : [ {
 "label" : "Post to Chatter",
 "name" : "chatterPost",
 "type" : "CHATTERPOST",
  "url" : "/services/data/v46.0/actions/standard/chatterPost"
}, {
  "label" : "Enable Folder Support for a Content Workspace (Library)",
 "name" : "contentWorkspaceEnableFolders",
 "type" : "CONTENTWORKSPACE ENABLE FOLDERS",
 "url" : "/services/data/v46.0/actions/standard/contentWorkspaceEnableFolders"
}, {
 "label" : "Send Email",
 "name" : "emailSimple",
 "type" : "EMAILSIMPLE",
  "url" : "/services/data/v46.0/actions/standard/emailSimple"
}, {
  "label" : "Submit for Approval",
 "name" : "submit",
 "type" : "SUBMITAPPROVAL",
 "url" : "/services/data/v46.0/actions/standard/submit"
 "label" : "Deactivate Session-Based Permission Set",
 "name" : "deactivateSessionPermSet",
  "type" : "DEACTIVATE SESSION PERM SET",
  "url" : "/services/data/v46.0/actions/standard/deactivateSessionPermSet"
}, {
 "label" : "Activate Session-Based Permission Set",
 "name" : "activateSessionPermSet",
 "type" : "ACTIVATE SESSION PERM SET",
  "url" : "/services/data/v46.0/actions/standard/activateSessionPermSet"
}, {
```

```
"label" : "Choose Price Book",
 "name" : "choosePricebook",
  "type" : "CHOOSE PRICEBOOK",
  "url" : "/services/data/v46.0/actions/standard/choosePricebook"
  "label" : "Routing Address Verification",
  "name" : "routingAddressVerification",
  "type" : "ROUTING ADDRESS VERIFICATION",
  "url": "/services/data/v46.0/actions/standard/routingAddressVerification"
}, {
  "label" : "Create Customer Contact Request",
  "name" : "contactRequestAction",
  "type" : "CONTACT REQUEST ACTION",
  "url" : "/services/data/v46.0/actions/standard/contactRequestAction"
  "label" : "Publish Managed Content Release",
  "name" : "managedContentReleasePublish",
  "type" : "MANAGED_CONTENT_RELEASE PUBLISH",
  "url" : "/services/data/v46.0/actions/standard/managedContentReleasePublish"
} ]
```

## Get the attributes of a single standard action, for example, emailSimple

/services/data/v46.0/actions/standard/emailSimple

#### **JSON Response body**

```
{
"category" : "Email",
"description": "Send an email where you specify the subject, body, and recipients.",
"inputs" : [ {
 "apexClass" : null,
 "byteLength" : 0,
 "description": "Optional. The email recipients specified as a comma-separated list.",
 "label" : "Email Addresses (comma-separated)",
 "maxOccurs" : 1,
 "name" : "emailAddresses",
 "picklistValues" : null,
 "required" : false,
 "sobjectType" : null,
 "type" : "STRING"
  "apexClass" : null,
 "byteLength" : 0,
 "description": "Optional. The email recipients specified as a collection of Strings.",
 "label" : "Email Addresses (collection)",
 "maxOccurs" : 5,
 "name" : "emailAddressesArray",
 "picklistValues" : null,
 "required" : false,
 "sobjectType" : null,
 "type" : "STRING"
}, {
```

```
"apexClass" : null,
 "byteLength" : 0,
 "description": "Optional. Who the email is from. Defaults to the current user.",
 "label" : "Sender Type",
 "maxOccurs" : 1,
 "name" : "senderType",
 "picklistValues" : null,
 "required" : false,
 "sobjectType" : null,
 "type" : "STRING"
}, {
 "apexClass" : null,
 "byteLength" : 0,
 "description": "Optional. The org-wide email address to be used as the sender.",
 "label" : "Sender Address",
 "maxOccurs" : 1,
 "name" : "senderAddress",
 "picklistValues" : null,
 "required" : false,
 "sobjectType" : null,
 "type" : "STRING"
}, {
 "apexClass" : null,
 "byteLength" : 0,
 "description": "Required. The email's subject.",
 "label" : "Subject",
 "maxOccurs" : 1,
 "name" : "emailSubject",
 "picklistValues" : null,
 "required" : true,
 "sobjectType" : null,
 "type" : "STRING"
 "apexClass" : null,
 "byteLength" : 0,
 "description" : "Required. The body of the email in plain text.",
 "label" : "Body",
 "maxOccurs" : 1,
 "name" : "emailBody",
 "picklistValues" : null,
 "required" : true,
 "sobjectType" : null,
 "type" : "TEXTAREA"
 "label" : "Send Email",
 "name" : "emailSimple",
 "outputs" : [ ],
 "standard" : true,
 "targetEntityName" : null,
 "type" : "EMAILSIMPLE"
```

Reference Custom Invocable Actions

## **Custom Invocable Actions**

Returns the list of all custom actions. You can also get basic information for each type of action.

This resource is available in REST API version 32.0 and later.

## **Syntax**

#### URI

Get a list of custom actions:

/vXX.X/actions/custom

#### **Formats**

JSON, XML

## **HTTP Methods**

GET, HEAD, POST

#### Authentication

Authorization: Bearer token

#### **Parameters**

None

#### **Notes**

Sending email with the emailAlert action counts against your daily email limit for workflows. For more information, see "Daily Allocations for Email Alerts" in the Salesforce Help.

When invoking an Apex action using the POST method and supplying the inputs in the request, only the following primitive types are supported as inputs:

- Blob
- Boolean
- Date
- Datetime
- Decimal
- Double
- ID
- Integer
- Long
- String
- Time

Describe and invoke for an Apex action respect the profile access for the Apex class. If you don't have access an error is issued.

If you add an Apex action to a flow, and then remove the Invocable Method annotation from the Apex class, a runtime error in the flow occurs.

When a flow user invokes an autolaunched flow, the active flow version runs. If there's no active version, the latest version runs. When a flow admin invokes a flow, the latest version always runs.

If any of the following elements are used in a flow, packageable components that reference these elements aren't automatically included in the package.

Apex action

Reference List View Describe

- Email alerts
- Post to Chatter core action
- Quick Action core action
- Send Email core action
- Submit for Approval core action

For example, if you use an email alert, manually add the email template that is used by that email alert. To deploy the package successfully, manually add those referenced components to the package.

#### Example

Retrieving a list of custom actions for the current organization:

```
/services/data/v33.0/actions/custom
```

#### **JSON Response body**

```
"quickAction" : "/services/data/v33.0/actions/custom/quickAction",
"apex" : "/services/data/v33.0/actions/custom/apex",
"emailAlert": "/services/data/v33.0/actions/custom/emailAlert",
"flow" : "/services/data/v33.0/actions/custom/flow"
```

## List View Describe

Returns detailed information about a list view, including the ID, the columns, and the SOQL query.

This resource is available in REST API version 32.0 and later.

#### URI

/vXX.X/sobjects/{sobjectType}/listviews/{queryLocator}/describe

## **Formats**

JSON, XML

## **HTTP Method**

**GET** 

### Authentication

```
Authorization: Bearer token
```

#### **Parameters**

None



## Example:

## Retrieving information about a list view

```
curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/describe
 -H "Authorization: Bearer token"
```

## **JSON Response body**

```
"columns" : [ {
 "ascendingLabel" : "Z-A",
```

Reference List View Describe

```
"descendingLabel" : "A-Z",
 "fieldNameOrPath" : "Name",
 "hidden" : false,
 "label" : "Account Name",
 "selectListItem" : "Name",
 "sortDirection" : "ascending",
  "sortIndex" : 0,
 "sortable" : true,
 "type" : "string"
 "ascendingLabel" : "Z-A",
 "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "Site",
 "hidden" : false,
 "label" : "Account Site",
  "selectListItem" : "Site",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
 "type" : "string"
}, {
 "ascendingLabel" : "Z-A",
 "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "BillingState",
 "hidden" : false,
 "label" : "Billing State/Province",
 "selectListItem" : "BillingState",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
 "type" : "string"
}, {
  "ascendingLabel" : "9-0",
 "descendingLabel" : "0-9",
 "fieldNameOrPath" : "Phone",
 "hidden" : false,
 "label" : "Phone",
 "selectListItem" : "Phone",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
 "type" : "phone"
  "ascendingLabel" : "Low to High",
 "descendingLabel" : "High to Low",
 "fieldNameOrPath" : "Type",
 "hidden" : false,
 "label" : "Type",
 "selectListItem" : "toLabel(Type)",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
  "type" : "picklist"
}, {
```

Reference List View Describe

```
"ascendingLabel" : "Z-A",
  "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "Owner.Alias",
 "hidden" : false,
 "label" : "Account Owner Alias",
 "selectListItem" : "Owner.Alias",
  "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
 "type" : "string"
 "ascendingLabel" : null,
 "descendingLabel" : null,
 "fieldNameOrPath" : "Id",
 "hidden" : true,
  "label" : "Account ID",
 "selectListItem" : "Id",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : false,
  "type" : "id"
  "ascendingLabel" : null,
 "descendingLabel" : null,
 "fieldNameOrPath" : "CreatedDate",
 "hidden" : true,
 "label" : "Created Date",
 "selectListItem" : "CreatedDate",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : false,
 "type" : "datetime"
  "ascendingLabel" : null,
 "descendingLabel" : null,
 "fieldNameOrPath" : "LastModifiedDate",
 "hidden" : true,
 "label" : "Last Modified Date",
 "selectListItem" : "LastModifiedDate",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : false,
  "type" : "datetime"
}, {
  "ascendingLabel" : null,
 "descendingLabel" : null,
 "fieldNameOrPath" : "SystemModstamp",
 "hidden" : true,
 "label" : "System Modstamp",
 "selectListItem" : "SystemModstamp",
 "sortDirection" : null,
 "sortIndex" : null,
  "sortable" : false,
 "type" : "datetime"
```

```
"id" : "00BD000005WcBe",
 "orderBy" : [ {
   "fieldNameOrPath" : "Name",
   "nullsPosition" : "first",
   "sortDirection" : "ascending"
 }, {
   "fieldNameOrPath" : "Id",
   "nullsPosition" : "first",
   "sortDirection" : "ascending"
 "query" : "SELECT name, site, billingstate, phone, tolabel(type), owner.alias,
id, createddate, lastmodifieddate, systemmodstamp FROM Account WHERE CreatedDate =
THIS WEEK ORDER BY Name ASC NULLS FIRST, Id ASC NULLS FIRST",
 "scope" : null,
 "sobjectType" : "Account",
 "whereCondition" : {
   "field" : "CreatedDate",
   "operator" : "equals",
   "values" : [ "THIS WEEK" ]
}
```

## List View Results

Executes the SOQL query for the list view and returns the resulting data and presentation information.

This resource is available in REST API version 32.0 and later.

## URI

/vXX.X/sobjects/{sobjectType}/listviews/{listViewID}/results

## **Formats**

JSON, XML

#### **HTTP Method**

GET

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
limit	The maximum number of records to return, between 1-2000. The default value is 25.
offset	The first record to return. Use this parameter to paginate the results. The default value is 1.



#### Retrieving results from a specific list view

curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0/results
-H "Authorization: Bearer token"

### **JSON Response body**

```
"columns" : [ {
 "ascendingLabel" : "Z-A",
 "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "Name",
 "hidden" : false,
 "label" : "Account Name",
 "selectListItem" : "Name",
 "sortDirection" : "ascending",
 "sortIndex" : 0,
  "sortable" : true,
 "type" : "string"
}, {
  "ascendingLabel" : "Z-A",
 "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "Site",
 "hidden" : false,
 "label" : "Account Site",
 "selectListItem" : "Site",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
 "type" : "string"
}, {
  "ascendingLabel" : "Z-A",
 "descendingLabel" : "A-Z",
 "fieldNameOrPath" : "BillingState",
 "hidden" : false,
 "label" : "Billing State/Province",
 "selectListItem" : "BillingState",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
  "type" : "string"
  "ascendingLabel" : "9-0",
 "descendingLabel" : "0-9",
 "fieldNameOrPath" : "Phone",
  "hidden" : false,
 "label" : "Phone",
 "selectListItem" : "Phone",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : true,
  "type" : "phone"
```

```
"ascendingLabel" : "Low to High",
 "descendingLabel" : "High to Low",
 "fieldNameOrPath" : "Type",
 "hidden" : false,
 "label" : "Type",
  "selectListItem" : "toLabel(Type)",
  "sortDirection" : null,
 "sortIndex" : null,
  "sortable" : true,
  "type" : "picklist"
  "ascendingLabel" : "Z-A",
  "descendingLabel" : "A-Z",
  "fieldNameOrPath" : "Owner.Alias",
  "hidden" : false,
  "label" : "Account Owner Alias",
 "selectListItem" : "Owner.Alias",
  "sortDirection" : null,
  "sortIndex" : null,
  "sortable" : true,
  "type" : "string"
}, {
  "ascendingLabel" : null,
  "descendingLabel" : null,
  "fieldNameOrPath" : "Id",
 "hidden" : true,
 "label" : "Account ID",
  "selectListItem" : "Id",
  "sortDirection" : null,
 "sortIndex" : null,
  "sortable" : false,
  "type" : "id"
}, {
  "ascendingLabel" : null,
  "descendingLabel" : null,
 "fieldNameOrPath" : "CreatedDate",
 "hidden" : true,
  "label" : "Created Date",
  "selectListItem" : "CreatedDate",
  "sortDirection" : null,
  "sortIndex" : null,
  "sortable" : false,
  "type" : "datetime"
}, {
  "ascendingLabel" : null,
  "descendingLabel" : null,
  "fieldNameOrPath" : "LastModifiedDate",
  "hidden" : true,
 "label" : "Last Modified Date",
  "selectListItem" : "LastModifiedDate",
  "sortDirection" : null,
  "sortIndex" : null,
  "sortable" : false,
```

```
"type" : "datetime"
}, {
 "ascendingLabel" : null,
 "descendingLabel" : null,
 "fieldNameOrPath" : "SystemModstamp",
 "hidden" : true,
 "label" : "System Modstamp",
 "selectListItem" : "SystemModstamp",
 "sortDirection" : null,
 "sortIndex" : null,
 "sortable" : false,
 "type" : "datetime"
"developerName" : "MyAccounts",
"done" : true,
"id" : "00BD000005WcCN",
"label" : "My Accounts",
"records" : [ {
 "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "Burlington Textiles Corp of America"
    "fieldNameOrPath" : "Site",
    "value" : null
 }, {
    "fieldNameOrPath" : "BillingState",
   "value" : "NC"
 }, {
   "fieldNameOrPath" : "Phone",
   "value" : "(336) 222-7000"
    "fieldNameOrPath" : "Type",
    "value" : "Customer - Direct"
    "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
  }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D00000JliSTIAZ"
    "fieldNameOrPath" : "CreatedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
  }, {
    "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
  }, {
    "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
    "value" : "Dickenson plc"
 }, {
```

```
"fieldNameOrPath" : "Site",
   "value" : null
   "fieldNameOrPath" : "BillingState",
   "value" : "KS"
 }, {
    "fieldNameOrPath" : "Phone",
   "value" : "(785) 241-6200"
 }, {
   "fieldNameOrPath" : "Type",
   "value" : "Customer - Channel"
   "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
    "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSVIAZ"
 }, {
   "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
   "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
 "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "Edge Communications"
 }, {
   "fieldNameOrPath" : "Site",
   "value" : null
 }, {
    "fieldNameOrPath" : "BillingState",
   "value" : "TX"
    "fieldNameOrPath" : "Phone",
   "value" : "(512) 757-6000"
 }, {
   "fieldNameOrPath" : "Type",
   "value" : "Customer - Direct"
    "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
 }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSSIAZ"
  }, {
   "fieldNameOrPath" : "CreatedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } , {
    "fieldNameOrPath" : "LastModifiedDate",
```

```
"value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "Express Logistics and Transport"
   "fieldNameOrPath" : "Site",
   "value" : null
    "fieldNameOrPath" : "BillingState",
    "value" : "OR"
 }, {
    "fieldNameOrPath" : "Phone",
   "value": "(503) 421-7800"
  }, {
   "fieldNameOrPath" : "Type",
   "value" : "Customer - Channel"
    "fieldNameOrPath" : "Owner.Alias",
    "value" : "TUser"
 }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSXIAZ"
 }, {
   "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "LastModifiedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
  }, {
    "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "GenePoint"
  }, {
    "fieldNameOrPath" : "Site",
    "value" : null
 }, {
   "fieldNameOrPath" : "BillingState",
   "value" : "CA"
 }, {
   "fieldNameOrPath" : "Phone",
   "value": "(650) 867-3450"
    "fieldNameOrPath" : "Type",
    "value" : "Customer - Channel"
  }, {
```

```
"fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
   "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSPIAZ"
    "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
   "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "Grand Hotels and Resorts Ltd"
 }, {
    "fieldNameOrPath" : "Site",
   "value" : null
  }, {
    "fieldNameOrPath" : "BillingState",
    "value" : "IL"
 }, {
    "fieldNameOrPath" : "Phone",
   "value" : "(312) 596-1000"
    "fieldNameOrPath" : "Type",
   "value" : "Customer - Direct"
 }, {
   "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
 }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSWIAZ"
    "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "LastModifiedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "Pyramid Construction Inc."
    "fieldNameOrPath" : "Site",
    "value" : null
```

Reference List View Results

```
"fieldNameOrPath" : "BillingState",
   "value" : null
 }, {
   "fieldNameOrPath" : "Phone",
    "value" : "(014) 427-4427"
 }, {
    "fieldNameOrPath" : "Type",
   "value" : "Customer - Channel"
   "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
    "fieldNameOrPath" : "Id",
    "value" : "001D000000JliSUIAZ"
 }, {
    "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "SystemModstamp",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "sForce"
 }, {
   "fieldNameOrPath" : "Site",
   "value" : null
  }, {
    "fieldNameOrPath" : "BillingState",
    "value" : "CA"
 }, {
   "fieldNameOrPath" : "Phone",
   "value" : "(415) 901-7000"
 }, {
   "fieldNameOrPath" : "Type",
   "value" : null
 }, {
    "fieldNameOrPath" : "Owner.Alias",
    "value" : "TUser"
  }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D000000JliSaIAJ"
 }, {
   "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "LastModifiedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
  }, {
```

Reference List View Results

```
"fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
    "value" : "United Oil and Gas Corp."
 }, {
    "fieldNameOrPath" : "Site",
   "value" : null
 }, {
   "fieldNameOrPath" : "BillingState",
   "value" : "NY"
 }, {
    "fieldNameOrPath" : "Phone",
    "value" : "(212) 842-5500"
  }, {
   "fieldNameOrPath" : "Type",
   "value" : "Customer - Direct"
    "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
 }, {
    "fieldNameOrPath" : "Id",
    "value" : "001D000000JliSZIAZ"
 }, {
    "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "SystemModstamp",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
  } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "United Oil and Gas, Singapore"
   "fieldNameOrPath" : "Site",
   "value" : null
  }, {
    "fieldNameOrPath" : "BillingState",
   "value" : "Singapore"
  }, {
    "fieldNameOrPath" : "Phone",
   "value" : "(650) 450-8810"
   "fieldNameOrPath" : "Type",
    "value" : "Customer - Direct"
    "fieldNameOrPath" : "Owner.Alias",
    "value" : "TUser"
```

Reference List View Results

```
"fieldNameOrPath" : "Id",
   "value" : "001D000000JliSRIAZ"
   "fieldNameOrPath" : "CreatedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
    "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
   "fieldNameOrPath" : "SystemModstamp",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "United Oil and Gas, UK"
 }, {
   "fieldNameOrPath" : "Site",
   "value" : null
   "fieldNameOrPath" : "BillingState",
   "value" : "UK"
  }, {
    "fieldNameOrPath" : "Phone",
    "value" : "+44 191 4956203"
  }, {
   "fieldNameOrPath" : "Type",
   "value" : "Customer - Direct"
 }, {
   "fieldNameOrPath" : "Owner.Alias",
   "value" : "TUser"
  }, {
    "fieldNameOrPath" : "Id",
    "value" : "001D000000JliSQIAZ"
  }, {
    "fieldNameOrPath" : "CreatedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
 }, {
   "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "SystemModstamp",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
}, {
  "columns" : [ {
   "fieldNameOrPath" : "Name",
   "value" : "University of Arizona"
 }, {
   "fieldNameOrPath" : "Site",
    "value" : null
 }, {
    "fieldNameOrPath" : "BillingState",
```

Reference List Views

```
"value" : "AZ"
 }, {
    "fieldNameOrPath" : "Phone",
    "value" : "(520) 773-9050"
    "fieldNameOrPath" : "Type",
    "value" : "Customer - Direct"
    "fieldNameOrPath" : "Owner.Alias",
    "value" : "TUser"
 }, {
    "fieldNameOrPath" : "Id",
   "value" : "001D00000JliSYIAZ"
    "fieldNameOrPath" : "CreatedDate",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "LastModifiedDate",
   "value" : "Fri Aug 01 21:15:46 GMT 2014"
    "fieldNameOrPath" : "SystemModstamp",
    "value" : "Fri Aug 01 21:15:46 GMT 2014"
 } ]
} ],
"size" : 12
```

# List Views

Returns the list of list views for the specified sObject, including the ID and other basic information about each list view. You can also get basic information for a specific list view by ID.

This resource is available in REST API version 32.0 and later.

# URI

Get a list of list views:

/vXX.X/sobjects/{sobjectType}/listviews

Get basic information about one list view:

/vXX.X/sobjects/{sobjectType}/listviews/{listViewID}

## Available since release

31.0

# **Formats**

JSON, XML

# **HTTP Methods**

**GET** 

# Authentication

Authorization: Bearer token

Reference List Views

#### **Parameters**

None



Example:

## Retrieving a list of list views for the Account object

```
curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/Account/listviews
-H "Authorization: Bearer token"
```

# **JSON Response body**

```
"done" : true,
 "listviews" : [ {
   "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/describe",
   "developerName" : "NewThisWeek",
   "id" : "00BD0000005WcBeMAK",
    "label" : "New This Week",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/results",
   "soqlCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK"
 }, {
    "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBpMAK/describe",
   "developerName" : "NewLastWeek",
   "id" : "00BD0000005WcBpMAK",
   "label" : "New Last Week",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBpMAK/results",
    "soglCompatible" : true,
    "url": "/services/data/v32.0/sobjects/Account/listviews/00BD000005WcBpMAK"
    "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcC6MAK/describe",
   "developerName" : "PlatinumandGoldSLACustomers",
   "id" : "00BD0000005WcC6MAK",
   "label" : "Platinum and Gold SLA Customers",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcC6MAK/results",
    "soglCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcC6MAK"
    "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD000005WcCEMA0/describe",
   "developerName" : "RecentlyViewedAccounts",
   "id" : "00BD0000005WcCEMA0",
   "label" : "Recently Viewed Accounts",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCEMA0/results",
   "soqlCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCEMA0"
```

```
"describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0/describe",
   "developerName" : "AllAccounts",
   "id" : "00BD000005WcCFMA0",
   "label" : "All Accounts",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0/results",
   "soqlCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0"
    "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0/describe",
   "developerName" : "MyAccounts",
   "id" : "00BD000005WcCNMA0",
   "label" : "My Accounts",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0/results",
   "soqlCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0"
 } ],
 "nextRecordsUrl" : null,
 "size" : 6,
 "sobjectType" : "Account"
```

# Retrieving basic information about one list view

Use the ID of a list view to get basic information about a specific list view.

```
curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK
-H "Authorization: Bearer token"
```

# **JSON Response body**

```
"describeUrl":
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/describe",
   "developerName": "NewThisWeek",
   "id": "00BD0000005WcBeMAK",
   "label": "New This Week",
   "resultsUrl":
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/results",
   "soqlCompatible": true,
   "url": "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK"
}
```

# Support Knowledge with REST API

Knowledge Support REST APIs allow both authorized and guest users to retrieve the user's visible data categories and their associated articles.

Authenticated users need the UserProfile.apiEnabled permission, Knowledge enabled in the organization, read rights on the article type, and any other knowledge specific permission or preference that controls visibility to articles.

Guest users need the Guest Access to the Support API preference enabled on the relevant Site, Knowledge enabled in the organization, and read rights on the article type and article channel that controls the visibility for guest users.

# **Syntax**

The root endpoint for all Knowledge support APIs that can be open to guest users.

#### Available since release

38.0

# Method

**GET** 

#### **Formats**

JSON, XML

# **Authentication**

OAuth accesstoken

## **Endpoint**

cprefix>/support

## **HTTP** headers

**Accept:** Optional. Can be either application/json or application/xml.

## Input

None

#### Output

The following resources are related to Knowledge.

```
{
"dataCategoryGroups" : "/services/data/vxx.0/support/dataCategoryGroups",
"knowledgeArticles" : "/services/data/vxx.0/support/knowledgeArticles"
:
}
```

Where vxx.0 is the API version you requested.

# IN THIS SECTION:

## **Data Category Groups**

Get data category groups that are visible to the current user.

# Data Category Detail

Get data category details and the child categories by a given category.

# Articles List

Get a page of online articles for the given language and category through either search or query.

## **Articles Details**

Get all online article fields, accessible to the user.

Reference Data Category Groups

# **Data Category Groups**

Get data category groups that are visible to the current user.

# **Syntax**

#### Available since release

38.0

# Method

GET

#### **Formats**

JSON, XML

# **Authentication**

OAuth accesstoken

# **Endpoint**

[prefix]/support/dataCategoryGroups

#### **HTTP** headers

**Accept:** Optional. Can be either application/json or application/xml.

**Accept-language:** Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

## Input:

string sObjectName: Required. KnowledgeArticleVersion only.

boolean topCategoriesOnly: Optional. Defaults to true

- True returns only the top level categories.
- False returns the entire tree.



# **Output:**

A list of the active data category groups that are visible to the current user in the site context. Returns id, name, label, and their top level categories or the entire data category group tree that are visible to the current user. The labels must be translated to the given language if they are available.

#### Data Category Group List

This payload lists the active root Data Category Groups that can be used in other requests to return the data categories and articles related to it.

```
{
   "categoryGroups": [ Data Category Group, ....],
}
```



# Data Category Group

Reference **Data Category Groups** 

This represents an individual data category group, and its root category.

```
"name": String, // the unique name of the category group
"label": String, // returns the translated version if it is available
"objectUsage" : String, // currently only "KnowledgeArticleVersion" is available.
"topCategories": [ Data Category Summary, ....]
```

# **Data Category Summary**

This provides a summary of data category information. The Summary and Detail responses share common properties, with the goal of providing only as much information as is necessary from associated resources.

```
{
   "name": String, // the unique name of the category
   "label": String, // returns the translated version if it is available
   "url": URL, // the url points to the data category detail API
"childCategories": [ Data Category Summary, ....] // null if topCategoriesOnly is
true
}
```

Note: The URL property is a pre-calculated path to the unique resource representing this data category, in this case it is a Data Category Detail API.

# Example

# Input

/services/data/v38.0/support/dataCategoryGroups?sObjectName=KnowledgeArticleVersion

# Output

```
"categoryGroups" : [ {
"label" : "Doc",
"name" : "Doc",
"objectUsage" : "KnowledgeArticleVersion",
"topCategories" : [ {
  "childCategories" : null,
  "label" : "All",
  "name" : "All",
  "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"
} ]
 }, {
"label" : "Manual",
"name" : "Manual",
"objectUsage" : "KnowledgeArticleVersion",
"topCategories" : [ {
  "childCategories" : null,
  "label" : "All",
  "name" : "All",
```

Reference Data Category Detail

# Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. Use the language code format used in *Which Languages Does Salesforce Support?*.

Only the user's visible data categories are returned. A user might be able to see several sub trees in the category group, therefore, the top categories that are visible to the user in each group are returned.

# **Data Category Detail**

Get data category details and the child categories by a given category.

# **Syntax**

#### Available since release

38.0

## Method

GET

#### **Formats**

JSON, XML

## Authentication

OAuth accesstoken

#### **Endpoint**

[prefix]/support/dataCategoryGroups/[group]/dataCategories/[category]

# **HTTP** headers

**Accept:** Optional. Can be either application/json or application/xml.

**Accept-language:** Optional. Language to translate the categories. Any ISO-639 language abbreviation, and an ISO-3166 country code subtag in the HTTP Accept-Language header. Only one language accepted. If no language specified, the non-translated labels are returned.

#### Input:

string sObjectName: Required. KnowledgeArticleVersion only.

# **Output:**

Details of the category and a list of child categories (name, label, etc.).

### Data Category Detail

Used for situations where the hierarchical representation of data categories is important. The child property contains a list of child data categories.

```
{
  "name": String, // the unique name of the category
```

Reference Articles List

```
"label": String, // returns the translated version if it is available
"url": URL,
   "childCategories": [ Data Category Summary, ....],
}
```

**Note**: If the category isn't visible to the current user the return is empty.

# Example

## Input

/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion

# Output

```
"childCategories" : [ {
    "childCategories" : null,
    "label" : "Help",
    "name" : "Help",
    "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/Help?sObjectName=KnowledgeArticleVersion"
    "childCategories" : null,
    "label" : "QA",
    "name" : "QA",
    "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/QA?sObjectName=KnowledgeArticleVersion"
  } ],
  "label" : "All",
 "name" : "All",
 "url" :
"/services/data/v38.0/support/dataCategoryGroups/Doc/dataCategories/All?sObjectName=KnowledgeArticleVersion"
```

# Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. Use the language code format used in *Which Languages Does Salesforce Support?*.

# **Articles List**

Get a page of online articles for the given language and category through either search or guery.

# **Syntax**

#### Available since release

38.0

Reference Articles List

#### Method

**GET** 

#### **Formats**

JSON, XML

## Authentication

OAuth access token

#### **Endpoint**

[prefix]/support/knowledgeArticles

#### **HTTP** headers

**Accept:** Optional. Can be either application/json or application/xml.

**Accept-language:** Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."
- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

#### Input:

string q: Optional, Performs an SOSL search. If the query string is null, empty, or not given, an SOQL query runs.

The characters ? and \* are used for wildcard searches. The characters (, ), and " are used for complex search terms. See https://developer.salesforce.com/docs/atlas.en-us.sogl\_sosl.meta/sogl\_sosl/sforce\_api\_calls\_sosl\_find.htm.

string channel: Optional, defaults to user's context. For information on channel values, see Valid channel values.

- **App**: Visible in the internal Salesforce Knowledge application
- **Pkb**: Visible in the public knowledge base
- **Csp**: Visible in the Customer Portal
- Prm: Visible in the Partner Portal

string categories in map json format {"group1":"category1", "group2":"category2", ...} )

Optional, defaults to None. Category group must be unique in each group:category pair, otherwise you get ARGUMENT OBJECT PARSE ERROR.

string queryMethod values are: AT, BELOW, ABOVE, ABOVE\_OR\_BELOW. Only valid when categories are specified, defaults to ABOVE OR BELOW.

string sort: Optional, a sortable field name LastPublishedDate, CreatedDate, Title, ViewScore. Defaults to LastPublishedDate for query and relevance for search.



**Note:** When sorting on ViewScore it is only available for query, not search, and no pagination is supported. You only get one page of results.

string order: Optional, either ASC or DESC, defaults to DESC. Valid only when sort is valid.

integer pageSize: Optional, defaults to 20. Valid range 1 to 100.

integer pageNumber : Optional, defaults to 1.

## **Output:**

A page of online articles in the given language and category visible to the current user.

Article Page

Reference Articles List

A page of articles. The individual entries are article summaries so the size can be kept at a minimum.

```
"articles": [ Article Summary, ... ], // list of articles
   "currentPageUrl": URL, // the article list API with current page number
   "nextPageUrl": URL, // the article list API with next page number,
   which can be null if there are no more articles.
   "pageNumber": Int // the current page number, starting at 1.
}
```

- Note: The API supports paging. Each page of responses includes a URL to its page, as well as the URL to the next page of articles.
- Note: if the user input parameter has the default value, the API does not show this parameter in either currentPageUrl or nextPageUrl.

# Article Summary

A summary of an article used in a list of article responses. It shares similar properties to the Article Detail representation, as one is a superset of the other.

```
"id": Id,
                 // articleId
    "articleNumber": String,
    "articleType": String, // apiName of the article type, available in API v44.0
and later
    "title": String,
    "urlName": String, // available in API v44.0 and later
    "summary": String,
    "url": URL, // to the Article Detail API
                           // view count in the interested channel
    "viewCount": Int,
    "viewScore": double (in xxx.xxxx precision), // view score in the interested
channel.
    "upVoteCount": int, // up vote count in the interested channel.
    "downVoteCount": int, // down vote count in the interested channel.
    "lastPublishedDate": Date // last publish date in ISO8601 format
    "categoryGroups": [ Data Category Group, .... ]}
```

The "url" property always points to the Article Details resource endpoint. For information on valid channel values, see the channel parameter description

# Data Category Group

An individual data category group, its root category, and a list of selected data categories in the group.

```
"groupName": String, // the unique name of the category group
"groupLabel": String, // returns the translated version
"selectedCategories": [ Data Category Summary, ... ]
}
```

## Data Category Summary

Provides a summary of data category information. The Summary and Detail responses share common properties.

```
"categoryName": String, // the unique name of the category
"categoryLabel": String, // returns the translated version, per the API
```

Articles List Reference

```
language specified
          "url": String // returns the url for the DataCategory REST API.
```



Note: The outputs of Data Category Group and Data Category Summary in Article List API are different from the Data Category Groups API.

# Example

# Input

```
/services/data/v38.0/support/knowledgeArticles?sort=ViewScore&channel=Pkb&pageSize=3
   HTTP Headers:
       Content-Type: application/json; charset=UTF-8
       Accept: application/json
       Accept-Language: en-US
```

# Output

```
"articles" : [ {
"articleNumber" : "000001002",
"categoryGroups" : [ ],
"downVoteCount" : 0,
"id" : "kA0xx00000000BCAQ",
"lastPublishedDate" : "2015-02-25T02:07:18Z",
"summary": "With this online Chinese input tool, you can type Chinese characters
through your web browser without installing any Chinese input software in your system.
The Chinese online input tool uses the popular Pin Yin input method. It is a fast and
convenient tool to input Chinese on English OS environments.",
"title" : "Long text test",
"upVoteCount" : 0,
"url": "/services/data/v38.0/support/knowledgeArticles/kA0xx00000000BCAQ",
 "viewCount" : 4,
"viewScore" : 100.0
}, {
"articleNumber" : "000001004",
"categoryGroups" : [ ],
"downVoteCount" : 0,
"id" : "kA0xx00000000LCAQ",
"lastPublishedDate" : "2016-06-21T21:11:02Z",
"summary" : "The number of characters required for complete coverage of all these
languages' needs cannot fit in the 256-character code space of 8-bit character encodings,
requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.
\r\nAlthough CJK encodings have common character sets, the encodings often used to
represent them have been developed separately by different East Asian governments and
software companies, and are mutually incompatible. Unicode has attempted, with some
controversy, to unify the character sets in a process known as Han unification.\r\n\r\nCJK
character encodings should consist minimally of Han characters p",
"title" : "Test Images",
"upVoteCount" : 0,
"url": "/services/data/v38.0/support/knowledgeArticles/kA0xx00000000LCAQ",
"viewCount" : 0,
"viewScore" : 0.0
```

```
"articleNumber" : "000001012",
"categoryGroups" : [ ],
"downVoteCount" : 0,
"id" : "kA0xx00000006GCAQ",
"lastPublishedDate" : "2016-06-21T21:10:48Z",
"summary" : null,
"title" : "Test Draft 2",
"upVoteCount" : 0,
"url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx000000006GCAQ",
"viewCount" : 0,
"viewScore" : 0.0
} ],
"currentPageUrl" :
"/services/data/v38.0/support/knowledgeArticles?channel=Pkb&pageSize=3&sort=ViewScore",
"nextPageUrl" : null,
"pageNumber" : 1
```

# Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in *Which Languages Does Salesforce Support?*.

# Valid channel Values

- When using the options string channel, where the matching articles are visible, the following values are valid.
  - App-Visible in the internal Salesforce Knowledge application
  - Pkb-Visible in the public knowledge base
  - Csp-Visible in the Customer Portal
  - Prm-Visible in the Partner Portal
- If channel isn't specified, the default value is determined by the type of user.
  - Pkb for a guest user
  - Csp for a Customer Portal user
  - Prm for a Partner Portal user
  - App for any other type of user
- If channel is specified, the specified value may be used to retrieve articles.
  - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.
  - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

# **Articles Details**

Get all online article fields, accessible to the user.

# **Syntax**

#### Available since release

38.0

# Method

GET

#### **Formats**

JSON, XML

## Authentication

OAuth access token

#### **Endpoint**

[prefix]/support/knowledgeArticles/{articleId}

[prefix]/support/knowledgeArticles/{articleUrlName} Available in API v44.0 and later

#### **HTTP** headers

**Accept:** Optional. Can be either application/json or application/xml.

**Accept-language:** Required. The article must be an active language in the user's organization

- If the language code isn't valid, an error message is returned: "The language code is not valid or not supported by Knowledge."
- If the language code is valid, but not supported by Knowledge, then an error message is returned: "Invalid language code. Check that the language is included in your Knowledge language settings."

## Input:

string channel: Optional, defaults to user's context. For information on channel values, see Valid channel Values.

- **App**: Visible in the internal Salesforce Knowledge application
- **Pkb**: Visible in the public knowledge base
- Csp: Visible in the Customer Portal
- **Prm**: Visible in the Partner Portal

boolean updateViewStat: Optional, defaults to true. If true, API updates the view count in the given channel as well as the total view count.

boolean isUrlName: Optional, defaults to false. If true, indicates that the last portion of the endpoint is a URL name instead of an article ID. Available in API v44.0 and later

# **Output:**

The detailed fields of the article, if the article is online and visible to the current user.

#### Article Detail

Full detail of an article, with complete metadata and layout-driven fields used for display of an article. It includes all the same properties as an Article Summary representation.

```
"versionNumber": Int,
        "createdDate": Date, // in ISO8601 format
        "createdBy": User Summary on page 189,
        "lastModifiedDate": Date, // in ISO8601 format
        "lastModifiedBy": User Summary on page 189,
        "lastPublishedDate": Date, // in ISO8601 format
       "layoutItems": [ Article Field, ... ], // standard and custom fields visible
to the user, sorted based on the layouts of the article type.
        "categories": [ Data Category Groups, ... ],
        "appUpVoteCount": Int,
        "cspUpVoteCount": Int,
        "prmUpVoteCount": Int,
        "pkbUpVoteCount": Int,
        "appDownVoteCount": Int,
        "cspDownVoteCount": Int,
        "prmDownVoteCount": Int,
        "pkbDownVoteCount": Int,
        "allViewCount": Int,
        "appViewCount": Int,
        "cspViewCount": Int,
        "prmViewCount": Int,
        "pkbViewCount": Int,
        "allViewScore": Double,
        "appViewScore": Double,
        "cspViewScore": Double,
        "prmViewScore": Double,
        "pkbViewScore": Double
```

# User Summary

### Article Field

An individual field of article information, which is listed in an Article Detail in the order required by the administrator's layout.

# Example

# Input

```
/services/data/v38.0/support/knowledgeArticles/kA0xx00000000LCAQ
HTTP Headers:
Content-Type: application/json; charset=UTF-8
Accept: application/json
Accept-Language: en-US
```

## Output

```
"allViewCount" : 17,
"allViewScore" : 100.0,
"appDownVoteCount" : 0,
"appUpVoteCount" : 0,
"appViewCount" : 17,
"appViewScore" : 100.0,
"articleNumber" : "000001004",
"categoryGroups" : [ ],
"createdBy" : {
"email" : "user@company.com",
"firstName" : "Test",
"id" : "005xx000001SvoMAAS",
"isActive" : true,
"lastName" : "User",
"url" : "/services/data/v38.0/chatter/users/005xx000001SvoMAAS",
"userName" : "admin@salesforce.org"
},
"createdDate": "2016-06-21T21:10:54Z",
"cspDownVoteCount" : 0,
"cspUpVoteCount" : 0,
"cspViewCount" : 0,
"cspViewScore" : 0.0,
"id" : "kA0xx00000000LCAQ",
"lastModifiedBy" : {
"email" : "user@company.com",
"firstName" : "Test",
"id" : "005xx000001SvoMAAS",
"isActive" : true,
"lastName" : "User",
"url" : "/services/data/v38.0/chatter/users/005xx000001SvoMAAS",
"userName" : "admin@salesforce.org"
"lastModifiedDate" : "2016-06-21T21:11:02Z",
"lastPublishedDate" : "2016-06-21T21:11:02Z",
"layoutItems" : [ {
"label" : "Out of Date",
"name" : "IsOutOfDate",
"type" : "CHECKBOX",
"value" : "false"
}, {
"label" : "sample",
"name" : "sample",
```

```
"type" : "PICK LIST",
       "value" : null
       "label" : "Language",
       "name" : "Language",
       "type" : "PICK LIST",
       "value" : "en US"
       "label" : "MyNumber",
       "name" : "MyNumber",
       "type" : "NUMBER",
       "value" : null
       }, {
       "label" : "My File",
       "name" : "My File",
       "type" : "FILE",
       "value" : null
       } ],
       "pkbDownVoteCount" : 0,
       "pkbUpVoteCount" : 0,
       "pkbViewCount" : 0,
       "pkbViewScore" : 0.0,
       "prmDownVoteCount" : 0,
       "prmUpVoteCount" : 0,
       "prmViewCount" : 0,
       "prmViewScore" : 0.0,
      "summary" : "The number of characters required for complete coverage of all these
languages' needs cannot fit in the 256-character code space of 8-bit character encodings,
requiring at least a 16-bit fixed width encoding or multi-byte variable-length encodings.
\r\n\r\nAlthough CJK encodings have common character sets, the encodings often used to
represent them have been developed separately by different East Asian governments and
software companies, and are mutually incompatible. Unicode has attempted, with some
controversy, to unify the character sets in a process known as Han unification.\r\n\r\nCJK
 character encodings should consist minimally of Han characters p",
       "title" : "Test Images",
       "url" : "/services/data/v38.0/support/knowledgeArticles/kA0xx00000000LCAQ",
       "versionNumber" : 7
```

# Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 38.0 and later. The Custom File Field is not supported because it returns a link to a binary stream. Use the language code format used in *Which Languages Does Salesforce Support?*.

A lookup custom field is visible to guest users depending on the lookup entity type. For example, User is visible, but Case and Account are not visible. Following standard fields are not visible to a guest user, even if they are in the layout:

- archivedBy
- isLatestVersion
- translationCompletedDate
- translationImportedDate
- translationExportedDate

- versionNumber
- visibleInInternalApp
- visibleInPKB
- visibleToCustomer
- visbileToPartner

# Valid channel Values

• When using the options string channel, where the matching articles are visible, the following values are valid.

- App-Visible in the internal Salesforce Knowledge application
- Pkb-Visible in the public knowledge base
- Csp-Visible in the Customer Portal
- Prm-Visible in the Partner Portal
- If channel isn't specified, the default value is determined by the type of user.
  - Pkb for a guest user
  - Csp for a Customer Portal user
  - Prm for a Partner Portal user
  - App for any other type of user
- If channel is specified, the specified value may be used to retrieve articles.
  - For guest, Customer Portal, and Partner Portal users, if the specified channel is other than the channel accessible to the user, an error is returned.
  - For all users other than guest, Customer Portal, and Partner Portal users, the specified channel value is used.

# Parameterized Search

Executes a simple RESTful search using parameters instead of a SOSL clause. Indicate parameters in a URL in the GET method. Or, use POST for more complex JSON searches.

# Syntax

#### URI

/vXX.X/parameterizedSearch/?q=search string

# **Formats**

JSON, XML

### **HTTP Method**

GET, POST

#### Authentication

Authorization: Bearer token

# **Required Global Parameters**

Name	Description
q	A search string that is properly URL-encoded.
	Note: SOSL clauses aren't supported.
	Available in version 36.0 and later.

# **Optional Global Parameters**

Name	Туре	Supported Methods	Description
dataCategory	string	GET	Single value. If an organization uses Salesforce Knowledge articles or answers, dataCategory filters all search results based on one data category.
			For example, dataCategory=GlobalCategoryc below NorthAmericac.
			When using dataCategories, specify a Salesforce Knowledge article or answer type with sobject and all the required parameters.
			For example:
			q=tourism&sobject=KnowledgeArticleVersion&KnowledgeArticleVersion.where= language='en_US'+and+publishStatus='online'&KnowledgeArticleVersion.fields= id,title&dataCategory=Locationc+Below+North_Americac
			If you require multiple dataCategory filters, use dataCategories with the POST method.
dataCategories	dataCategoriesFilter[]	POST	If an organization uses Salesforce Knowledge articles or answers, filters all search results based on one or more data categories.
			When using dataCategories, specify a Salesforce Knowledge article or answer type with sobjects and the required parameters.
			For example:
			<pre>{    "q":"Acme",    "fields":["id", "title"],    "sobjects":[{"name":"KnowledgeArticleVersion",    "where":"language='en_US' and publishstatus='draft'"}],     "dataCategories":[         {"groupName" : "locationc", "operator":"below",    "categories":["North_Americac"]}</pre>

Name	Туре	Supported Methods	Description
defaultLimit	string	GET, POST	Single value. The maximum number of results to return for each sobject (GET) or sobjects (POST) specified.
			The maximum defaultLimit is 2000.
			At least one sobject must be specified.
			GET example:
			defaultLimit=10&sobject=Account&sobject=Contact.
			When an sobject limit is specified using sobject.limit=value, such as Account.limit=10, this parameter is ignored for that object.
division	string	GET,	Single value. Filters search results based on the division field.
		POST	For example in the GET method, division=global.
			Specify a division by its name rather than ID.
			All searches within a specific division also include the global division.
fields	string	GET	Comma-separated list of one or more fields to return in the response for each sobject specified. At least one sobject must be specified at the global level.
			For example: fields=id&sobject=Account&sobject=Contact.
			The global fields parameter is overridden when sobject are specified using sobject.fields=field names. For example,  Contact.fields=id, FirstName, LastName would override the global setting of just returning the id.
			If unspecified, then the search results contain the IDs of records matching all fields for the specified object.
			Functions
			The following optional functions can be used within the fields parameter.
			• toLabel:Translates response field value into the user's language. For example, Lead.fields=id, toLabel (Status). This function requires extra setup.
			• convertCurrency: Converts response currency fields to the user's currency. For example,
			Opportunity.fields=id, convertCurrency (Amount). This function requires extra setup. Multi-currency must be enabled for your org.
			• format: Applies localized formatting to standard and custom number, date, time, and currency fields. For example,  Opportunity.fields=id, format (Amount).
			Aliasing is support within fields for toLabel, convertCurrency, and
			format. In addition, aliasing is required when the query includes the same field multiple times. For example,
			Opportunity.fields=id,format(Amount) AliasAmount

Name	Туре	Supported Methods	Description
fields	string[]	POST	Array of one or more fields to return in the response for each sobjects specified.  At least one sobjects must be specified at the global level.
			For example:
			<pre>{   "q":"Acme",   "fields":["Id", "Name", "Phone"],   "sobjects":[{"name": "Account"},</pre>
			The global fields parameter is overridden when sobjectsFilter[] fields are specified. Such as, in the previous example, Id, FirstName, and LastName is returned for Contact instead of the global fields of Id, Name and Phone.
			If unspecified, then the search results contain the IDs of records matching all fields for the specified object.
			Functions
			The following optional functions can be used within the fields parameter.
			• toLabel: Translates response field value into the user's language. This function requires extra setup. For example:
			<pre>{   "sobjects":[ {"name": "Lead", "fields":["Id",   "toLabel(Status)"]}, }</pre>
			<ul> <li>convertCurrency: Converts response currency fields to the user's currency.</li> <li>This function requires extra setup. Multi-currency must be enabled in the org.</li> <li>For example:</li> </ul>
			<pre>{   "sobjects":[ {"name": "Opportunity",   "fields":["Id", "convertCurrency(Amount)"]}] }</pre>
			• format: Applies localized formatting to standard and custom number, date, time, and currency fields. For example:
			{ "sobjects":[ {"name": "Opportunity",

Name	Туре	Supported Methods	Description
			"fields":["Id", "format(Amount)"]}] }
			Aliasing is supported within fields for toLabel, convertCurrency, and format. In addition, aliasing is required when the query includes the same field multiple times. For example:
			<pre>{   "sobjects":[ {"name": "Opportunity", "fields":["Id",   "format(Amount) AliasAmount"]}] }</pre>
in	string	GET, POST	Scope of fields to search. If you specify one or more scope values, the fields are returned for all found objects.
			Use one of the following values:
			• ALL
			• NAME
			• EMAIL • PHONE
			• SIDEBAR
			This clause doesn't apply to articles, documents, feed comments, feed items, files, products, and solutions. If any of these objects are specified, the search is not limited to specific fields; all fields are searched.
metadata	string	GET, POST	Specifies if metadata should be returned in the response. No metadata is returned by default. To include metadata in the response, use the LABELS value, which returns the display label for the fields returned in search results. For example: ?q=Acme&metadata=LABELS
netWorkIds	string	GET	Filters search results by a comma-separated list.
			A network ID represents the community ID.
netWorkIds	string[]	POST	Filters search results by an array.
			A network ID represents the community ID.
offset	string	GET,	Single value. The starting row offset into the result set returned.
		POST	The maximum offset is 2000.
			Only one sobject can be specified when using this parameter.

Name	Туре	Supported Methods	Description
overallLimit	string	GET, POST	Single value. The maximum number of results to return across all sobject parameters specified.  The maximum overallLimit is 2000.
pricebookId	string	GET, POST	Single value. Filters product search results by a price book ID for only the Product2 object. The price book ID must be associated with the product that you're searching for. For example, ?q=laptop&sobject=product2&pricebookId=01sxx0000002MffAAE
snippet	string	GET, POST	The target length (maximum number of snippet characters) to return in Salesforce Knowledge article, case, case comment, feed, feed comment, idea, and idea comment search results. The snippet parameter displays contextual excerpts and highlights the search term for each article in the search results. Snippet results are used to differentiate matches to the search term in article search results. The target length can be from 50 to 1000 characters.
			Snippet and highlights are generated from email, text, and text area (long and rich) fields. Snippets aren't displayed for partially matching searches or if the user doesn't have access to the field that contains the snippet. Snippets are only displayed when 20 or fewer results are returned on a page.
			At least one of the following sobject values must be specified.
			To search a specific article type, use the article type name with the suffixkav
			To search all article types, use KnowledgeArticleVersion.
			• To search case, case comment, feed, feed comment, idea, and idea comment types, use Case, CaseComment, FeedItem, FeedComment, Idea, and IdeaComment.
			For example, q=tourism&sobject=Case&snippet=500.
sobject	string	GET	Objects to return in the response. Must be a valid object type.
			You can use multiple sobject values, such as
			sobject=Account&sobject=Contact.  If unspecified, then the search results contain the IDs of all objects.
		2007	, , ,
sobjects	sobjectsFilter[]	POST	Objects to return in the response. Must contain valid object types. Use with the required parameters.  For example:
			<pre>{   "q":"Acme",   "fields":["id", "title"],   "sobjects":[{"name":"Solutionkav",   "where":"language='en_US' and publishstatus='draft'"},</pre>

Name	Туре	Supported Methods	Description
			{"name":"FAQkav", "where":"language='en_US' and publishstatus='draft'"}] }
			If unspecified, then the search results contain the IDs of all objects.
spellCorrection	boolean	GET, POST	Specifies whether spell correction is enabled for a user's search. When set to true, spell correction is enabled for searches that support spell correction. The default value is true.  For example:  q=Acme&sobject=Account&Account.fields=id&spellCorrection=true
updateTracking	string	GET, POST	Specifies a value of true to track keywords that are used in Salesforce Knowledge article searches only.  If unspecified, the default value of false is applied.
updateViewStat	string	GET, POST	Specifies a value of true to update an article's view statistics. Valid only for Salesforce Knowledge article searches.  If unspecified, the default value of false is applied.

# dataCategoriesFilter[] Parameters

Parameters must be specified in the order presented in the table (groupName, operator, categories).

Name	Туре	Description		
groupName	string	The name of the data category group to filter by.		
operator	string	Valid values:  • ABOVE		
		• ABOVE_OR_BELOW • AT • BELOW		
categories	string[]	The name of the categories to filter by.		

# sobjectsFilter[] Parameters (POST Method Only)

Name	Туре	Description
fields	string[]	Array of one or more fields to return in the response for the sobject.
limit	string	Specify the maximum number of rows that are returned for the sobject.

Name	Туре	Description
name	string	Name of the sobject to return in the response.
orderBy	string	Controls the field order of the results using the following syntax "orderBy": "field {ASC DESC} [NULLS_{FIRST LAST}]"  For example:
		<pre>{   "sobjects":[ {"name": "Account", "fields":["Id", "name"], "orderBy":   "Name DESC Nulls_last"}] }</pre>
		<ul> <li>ASC: ascending. Default.</li> <li>DESC: descending.</li> <li>NULLS_FIRST: Null records at the beginning of the results. Default.</li> <li>NULLS_LAST: Null records at the end of the results.</li> </ul>
where	string	Filter search results for this object by specific field values.  For example, where : conditionExpression. Here the conditionExpression of the WHERE clause uses the following syntax: fieldExpression [logicalOperator fieldExpression2].  Add multiple field expressions to a condition expression by using logical and comparison operators.

# sobject-level Parameters (GET Method Only)

The following optional parameters can be used with the sobject parameter in a GET method to further refine search results. These settings would override any settings specified at the global level.

The format is sobject.parameter, such as Account.fields. An sobject must be specified to use these parameters, for example, sobject=Account&Account.fields=id, name.

Name	Туре	Description
fields	string	Comma-separated list of one or more fields to return in the response.  For example, KnowledgeArticleVersion.fields=id, title.
limit	string	Specifies the maximum number of rows that are returned for the sobject.  For example, Account.limit=10.
orderBy	string	Controls the field order of the results using the following syntax orderBy = field {ASC DESC} [NULLS_{FIRST LAST}] For example: Account.orderBy=Name  ASC: ascending. Default.  DESC: descending.  NULLS_FIRST: Null records at the beginning of the results. Default.

Reference Process Approvals

Name	Туре	Description
		NULLS_LAST: Null records at the end of the results.
where	string	Filter search results for this object by specific field values.
		For example, Account.where = conditionExpression. Here the conditionExpression of the WHERE clause uses the following syntax: fieldExpression [logicalOperator fieldExpression2].
		Add multiple field expressions to a condition expression by using logical and comparison operators. For example, KnowledgeArticleVersion.where=publishstatus='online' and language='en_US'.

# **Example GET Method**

.../v37.0/parameterizedSearch/?q=Acme&sobject=Account&Account.fields=id,name&Account.limit=10

# **Example POST Method**

# **Process Approvals**

Returns a list of all approval processes. Can also be used to submit a particular record if that entity supports an approval process and one has already been defined. Records can be approved and rejected if the current user is an assigned approver. When using a POST request to do bulk approvals, the requests that succeed are committed and the requests that don't succeed send back an error.

# **Syntax**

URI

To return a list of the approvals, the URI is: /vXX.X/process/approvals/

#### Available since release

30.0

# Formats

JSON, XML

# **HTTP** methods

GET, HEAD, POST

Reference Process Approvals

# Authentication

Authorization: Bearer token

# **Request parameters**

None required

# **Request body**

The request body contains an array of process requests that contain the following information:

Name	Туре	Description
actionType	string	Represents the kind of action to take: Submit, Approve, or Reject.
contextActorId	ı ID	The ID of the submitter who's requesting the approval record.
contextId	ID	The ID of the item that is being acted upon.
comments	string	The comment to add to the history step associated with this request.
nextApproverIds	; ID[]	If the process requires specification of the next approval, the ID of the user to be assigned the next request.
poesefinitioNment	d string	The developer name or ID of the process definition.
skipEntryCriteria	a boolean	Determines whether to evaluate the entry criteria for the process (true) or not (false) if the process definition name or ID isn't null. If the process definition name or ID isn't specified, this argument is ignored, and standard evaluation is followed based on process order. By default, the entry criteria isn't skipped if it's not set by this request.

# **Response body**

The response contains an array of process results that contain the following information:

Name	Туре	Description
actorIds	ID[]	IDs of the users who are currently assigned to this approval step.
entityId	ID	The object being processed.
errors	Error[]	The set of errors returned if the request failed.
instanceId	ID	The ID of the ProcessInstance associated with the object submitted for processing.
instanceStatus	string	The status of the current process instance (not an individual object but the entire process instance). The valid values are "Approved," "Rejected," "Removed," or "Pending."
newWorkItemIds	ID[]	Case-insensitive IDs that point to ProcessInstanceWorkitem items (the set of pending approval requests)
success	boolean	true if processing or approval completed successfully.

# Examples

• See Get a List of All Approval Processes.

Reference Process Rules

- See Submit a Record for Approval.
- See Approve a Record.
- See Reject a Record.
- See Bulk Approvals.

# **Process Rules**

Returns a list of all active workflow rules. If a rule has actions, the actions will be listed under the rule. Can also be used to trigger all workflow rules that are associated with a specified record. The actions for a rule are only fired if the rule's criteria is met. When using a POST request, if anything fails, the whole transaction is rolled back.

Cross-object workflow rules cannot be invoked using the REST API.

# **Syntax**

#### URI

To get a list of the workflow rules or to trigger one or more workflow rules, the URI is: /vXX.X/process/rules/

To get the rules for a particular object: /vXX.X/process/rules/**SObjectName** 

To get the metadata for a particular rule: /vXX.X/process/rules/SObjectName/workflowRuleId

#### Available since release

30.0

### **Formats**

JSON, XML

# **HTTP** methods

HEAD, GET, POST

# Authentication

Authorization: Bearer token

# **Request parameters**

None required

## Request body

The request body contains an array of context IDs:

Name	Туре	Description
contextId	ID	The ID of the item that is being acted upon.

# **Examples**

- See Get a List of Process Rules.
- See Get a Particular Process Rule.
- See Trigger Process Rules.

Reference Product Schedules

# **Product Schedules**

Work with revenue and quantity schedules for opportunity products. Establish or reestablish a product schedule with multiple installments for an opportunity product. Delete all installments in a schedule.

This resource is available in REST API version 43.0 and later.

In API version 46.0 and later, established and re-established schedules support custom fields, validation rules, and Apex triggers. Deleting all schedules now also fires delete triggers.

## URI

/vXX.X/sobjects/OpportunityLineItem/{OpportunityLineItemId}/OpportunityLineItemSchedules

#### **Formats**

JSON, XML

## **HTTP Method**

GET, PUT, DELETE

#### Authentication

Authorization: Bearer token

# **Parameters**

Parameter	Description
type	The type of the schedule. Required when establishing OpportunityLineItemSchedules. Valid values include Quantity, Revenue, or Both.
quantity	The total number of units to be repeated or divided in a quantity schedule. Must be an integer other than 0.
quantityScheduleType	The type of the quantity schedule, if the product has one. Valid values are Divide or Repeat.
quantityScheduleInstallmentPeriod	If the product has a quantity schedule, the amount of time covered by the schedule. Valid values are Daily, Weekly, Monthly, Quarterly, Or Yearly.
quantityScheduleInstallmentsNumber	If the product has a quantity schedule, the number of installments. Can be an integer from 1 to 150.
quantityScheduleStartDate	The date the quantity schedule starts. Format is YYYY-MM-DD.
revenue	The amount of revenue to be repeated or divided.
revenueScheduleType	The type of the revenue schedule, if the product has one. Valid values are Divide or Repeat.
revenueScheduleInstallmentPeriod	If the product has a revenue schedule, the amount of time covered by the schedule. Valid values are Daily, Weekly, Monthly, Quarterly, Or Yearly.
revenueScheduleInstallmentsNumber	If the product has a revenue schedule, the number of installments. Can be an integer from 1 to 150.

Reference Query

Parameter	Description
revenueScheduleStartDate	The date the revenue schedule starts. Format is YYYY-MM-DD.



## Example:

Establish both quantity and revenue schedules for an opportunity product; establish a revenue schedule only; establish a quantity schedule only.

```
curl
https://yarInstance.salesforce.com/services/data/v43.0/sabjects/OpportunityLireItem/OORFO000001WUAIA4/OpportunityLireItem8dredules
-H "Authorization: Bearer token"
```

# **JSON Request body**

```
"type": "Both",
"quantity": 100,
"quantityScheduleType": "Repeat",
"quantityScheduleInstallmentPeriod": "Monthly",
"quantityScheduleInstallmentsNumber": 12,
"quantityScheduleStartDate": "2018-09-15",
"revenue": 100,
"revenueScheduleType": "Repeat",
"revenueScheduleInstallmentPeriod": "Monthly",
"revenueScheduleInstallmentsNumber": 12,
"revenueScheduleStartDate": "2018-09-15"
"type": "Revenue",
"revenue": 100,
"revenueScheduleType": "Divide",
"revenueScheduleInstallmentPeriod": "Quarterly",
"revenueScheduleInstallmentsNumber": 10,
"revenueScheduleStartDate": "2018-09-15"
"type": "Quantity",
"quantity": 10,
"quantityScheduleType": "Repeat",
"quantityScheduleInstallmentPeriod": "Daily",
"quantityScheduleInstallmentsNumber": 150,
"quantityScheduleStartDate": "2020-09-15",
```

# Query

Executes the specified SOQL query.

Reference Query

If the query results are too large, the response contains the first batch of results and a query identifier in the nextRecordsUrl field of the response. The identifier can be used in an additional request to retrieve the next batch.

#### URI

/vXX.X/query/?q=**SOQL query** 

For retrieving query performance feedback without executing the query:

/vXX.X/query/?explain=SOQL query

For retrieving query performance feedback on a report or list view:

/vXX.X/query/?explain=report or list view ID

For retrieving additional guery results if the initial results are too large:

/vXX.X/query/query identifier

#### **Formats**

JSON, XML

# **HTTP Method**

GFT

#### Authentication

Authorization: Bearer token

## **Parameters**

Parameter	Description	
d	A SOQL query. Note that you will need to replace spaces with "+" characters in your query string to create a valid URI. An example query parameter string might look like: "SELECT+Name+FROM+MyObject". If the SOQL query string is invalid, a MALFORMED_QUERY response is returned.	
explain	A SOQL query to get performance feedback on. Use explain instead of q to get a response that details how Salesforce will process your query. You can use this feedback to further optimize your queries. You can also use a report or list view ID in place of the query string to get feedback on how Salesforce will process your report or list view.	
	The explain parameter is available in API version 30.0 and later.	
	Note: Using explain with the REST API query resource is a beta feature. There is no support associated with this beta feature. For more information, contact Salesforce.	
	If the SOQL query string is invalid, a MALFORMED_QUERY response is returned. If the report or list view ID is invalid, an INVALID_ID response is returned.	

# Response body

For a query using the q parameter, the response contains an array of query result records. For a query using the explain parameter, the response contains one or more query plans that can be used to execute the query, report, or list view. The plans are sorted from most optimal to least optimal. Each plan has the following information:

Reference

Name	Туре	Description
cardinality	number	The estimated number of records the query would return, based on index fields, if any.
fields	string[]	The index fields used for the query, if the leading operation type is Index, otherwise null.
leadingOperationType	string	The primary operation type that will be used to optimize the query. This can be one of these values:
		• Index—The query will use an index on the query object.
		Other—The query will use optimizations internal to Salesforce.
		• Sharing—The query will use an index based on the user's sharing rules. If there are sharing rules that limit which records are visible to the current user, those rules can be used to optimize the query.
		• TableScan—The query will scan all records for the query object, and won't use an index.
notes	feedback note[]	An array of one or more feedback notes. Each note contains:
		• description— A detailed description of an aspect of the optimization. This could include information on optimizations that could not be used, with details on why they weren't used.
		• fields— An array of one or more fields used for the optimization.
		<ul> <li>tableEnumOrId— The table name for the fields used for the optimization.</li> </ul>
		This response field is available in API version 33.0 and later.
relativeCost	number	The cost of this query compared to the SOQL selective query threshold. A value greater than 1.0 means the query won't be selective. See "More Efficient SOQL Queries" in the <i>Apex Code Developer's Guide</i> for more information on selective queries.
sobjectCardinality	number	The approximate count of all records in your organization for the query object.
sobjectType	string	The name of the query object, such as Merchandisec.

# **Example**

For an example of making a query and retrieving additional query results using the query identifier, see Execute a SOQL Query on page 62.

For an example using the explain parameter to get feedback on a query and a report, see Get Feedback on Query Performance on page 64.

For more information on SOQL see the *SOQL and SOSL Reference*. For more information on query batch sizes, see Changing the Batch Size in Queries in the *SOAP API Developer Guide*.

Reference QueryAll

# QueryAll

Executes the specified SOQL query. Unlike the Query resource, QueryAll will return records that have been deleted because of a merge or delete. QueryAll will also return information about archived Task and Event records. QueryAll is available in API version 29.0 and later.

If the query results are too large, the response contains the first batch of results and a query identifier in the nextRecordsUrl field of the response. The identifier can be used in an additional request to retrieve the next batch. Note that even though nextRecordsUrl has query in the URL, it will still provide remaining results from the initial QueryAll request. The remaining results will include deleted records that matched the initial query.

### URI

/vXX.X/queryAll/?q=SOQL query

For retrieving additional query results if the initial results are too large:

/vXX.X/queryAll/query identifier

#### **Formats**

JSON, XML

## **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
ď	A SOQL query. Note that you will need to replace spaces with "+" characters in your query string to create a valid URI. An example query parameter string might look like: "SELECT+Name+FROM+MyObject".

### Example

- For an example of making a query that includes deleted items, see Execute a SOQL Query that Includes Deleted Items on page 63
- For an example of a query that retrieves additional results using the query identifier, see Retrieving the Remaining SOQL Query Results on page 64

For more information on SOQL see the SOQL and SOSL Reference. For more information on query batch sizes, see Changing the Batch Size in Queries in the SOAP API Developer Guide.

# **Quick Actions**

Returns a list of global actions and object-specific actions. This resource is available in REST API version 28.0 and later. When working with actions, also refer to SObject Quick Actions.

# URI

/vXX.X/quickActions/

# **Formats**

JSON, XML

Reference Recent List Views

#### **HTTP Method**

HEAD, GET, POST

#### Authentication

Authorization: Bearer token

#### **Parameters**

None required

#### Considerations

Add all required fields to an object before you create a quick action for that object. If you add a required field after creating a quick action, the field won't appear in the quick action's describe results. Then, when the quick action runs, the field won't be available and an error occurs for the missing field. If you don't want the required field to appear in the quick action's layout, set a default value for the field.

# Example usage for getting global quick actions

```
curl https://yourInstance.salesforce.com/services/data/v28.0/quickActions/ -H
"Authorization: Bearer token"
```

# Example for creating a contact using an action

```
curl https://yourInstance.salesforce.com/services/data/v28.0/quickActions/CreateContact
  -H 'Authorization: Bearer access_token -H "Content-Type: application/json" -d
@newcontact.json'
```

# Example JSON request body newcontact.json file

```
{
    "record" : { "LastName" : "Smith" }
}
```

# **Recent List Views**

Returns the list of recently used list views for the given sObject type.

This resource is available in REST API version 32.0 and later.

#### URI

/vXX.X/sobjects/{sobjectType}/listviews/recent

# **Formats**

JSON, XML

### **HTTP Method**

GET

#### Authentication

```
Authorization: Bearer token
```

#### **Parameters**

None

Reference Recently Viewed Items



#### Retrieving recent list views for the Account object

```
curl
https://yourInstance.salesforce.com/services/data/v32.0/sobjects/Account/listviews/recent
-H "Authorization: Bearer token"
```

#### **JSON Response body**

```
"done" : true,
 "listviews" : [ {
   "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0/describe",
   "developerName" : "MyAccounts",
   "id" : "00BD0000005WcCNMA0",
   "label" : "My Accounts",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0/results",
    "soqlCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCNMA0"
 }, {
   "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/describe",
   "developerName" : "NewThisWeek",
   "id" : "00BD0000005WcBeMAK",
   "label" : "New This Week",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK/results",
   "soqlCompatible" : true,
   "url": "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcBeMAK"
 }, {
   "describeUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0/describe",
   "developerName" : "AllAccounts",
   "id" : "00BD0000005WcCFMA0",
   "label" : "All Accounts",
   "resultsUrl" :
"/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0/results",
   "soglCompatible" : true,
   "url" : "/services/data/v32.0/sobjects/Account/listviews/00BD0000005WcCFMA0"
 "nextRecordsUrl" : null,
 "size" : 3,
 "sobjectType" : "Account"
```

## Recently Viewed Items

Gets the most recently accessed items that were viewed or referenced by the current user. Salesforce stores information about record views in the interface and uses it to generate a list of recently viewed and referenced records, such as in the sidebar and for the auto-complete options in search.

Reference Record Count

This resource only accesses most recently used item information. If you want to modify the list of recently viewed items, you'll need to update recently viewed information directly by using a SOQL Query with a FOR VIEW or FOR REFERENCE clause.

#### URI

/vXX.X/recent

#### **Formats**

JSON, XML

#### **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
limit	An optional limit that specifies the maximum number of records to be returned. If this parameter is not specified, the default maximum number of records returned is the maximum number of entries in RecentlyViewed, which is 200 records per object.

#### **Example**

- For an example of retrieving a list of recently viewed items, see View Recently Viewed Records on page 79.
- For an example of setting records as recently viewed, see Mark Records as Recently Viewed on page 80.

SEE ALSO:

RecentlyViewed

FOR VIEW

FOR REFERENCE

## **Record Count**

Lists information about object record counts in your organization.

This resource is available in REST API version 40.0 and later for API users with the "View Setup and Configuration" permission. The returned record count is approximate, and does not include the following types of records:

- Deleted records in the recycle bin.
- Archived records.

#### URI

/vXX.X/limits/recordCount?sObjects=**Object List** 

#### **Formats**

JSON, XML

#### **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

#### **Parameters**

Parameter	Description
s0bjects	A comma-delimited list of object names. If a listed object is not found in the org, it is ignored and not returned in the response.
	This parameter is optional. If this parameter is not provided, the resource returns record counts for all objects in the org.

#### **Response body**

Record Count Response Body

#### Example

Example request to get record counts for Account and Contact:

```
/services/data/v40.0/limits/recordCount?sObjects=Account,Contact
```

Example response for request:

```
{
    "sObjects" : [ {
        "count" : 3,
        "name" : "Account"
}, {
        "count" : 10,
        "name" : "Contact"
} ]
```

## **Record Count Response Body**

Describes the result of a Record Count request.

#### **Record Count Results**

#### **Properties**

Name	Туре	Description
sObjects	Record Count SObject Result[]	Collection of SObject record count results. The order of objects in the collection is not guaranteed to match the order of objects in the request.

#### JSON example

```
"sObjects" : [ {
    "count" : 3,
    "name" : "Account"
}, {
    "count" : 10,
```

Reference Relevant Items

```
"name" : "Contact"
} ]
```

## Record Count SObject Result

#### **Properties**

Name	Туре	Description
count	Integer	The number of records for the object in the org. This is an approximate count and does not include soft-deleted or archived records.
name	String	The name of the object.

#### JSON example

```
"count" : 10,
"name" : "Contact"
```

### Relevant Items

Gets the current user's most relevant items. Relevant items include records for objects in the user's global search scope and also most recently used (MRU) objects.

Relevant items include up to 50 of the most recently viewed or updated records for each object in the user's global search scope.



Note: The user's global search scope includes the objects the user interacted with most in the last 30 days, including objects the user pinned from the search results page in the Salesforce Classic.

Then, the resource finds more recent records for each most recently used (MRU) object until the maximum number of records, which is 2,000, is returned.

This resource only accesses the relevant item information. Modifying the list of relevant items is not currently supported.

This resource is available in API version 35.0 and later.

#### URI

/vXX.X/sobjects/relevantItems

#### **Formats**

**JSON** 

#### **HTTP Method**

**GET** 

#### Authentication

Authorization: Bearer token

Reference Relevant Items

#### **Parameters**

Parameter	Description	
lastUpdatedId	Optional. Compares the entire current list of relevant items to a previous version, if available. Specify the lastUpdatedId value returned in a previous response.	
sobjects	Optional. To scope the results to a particular object or set of objects, specify the name for one or more sObjects.	
	Note: SObject names are case-sensitive.	
sobject.lastUpdatedId	Optional. Compares the current list of relevant items for this particular object to a previous version, if available. Specify the lastUpdatedId value returned in a previous response.	
	Note: You can only specify this parameter for the sObjects specified in the sobjects parameter.	

#### Response header

The response contains headers unique to this resource.

Name	Туре	Description
lastUpdatedId	string	A unique code that can be used in subsequent calls to compare the results for a complete result set with the results in this response list.
newResultSetSinceLastQuery	boolean(true or false)	If a response was previously requested for the current user, indicates whether the current response matches the previous response, or the one specified by a lastUpdatedId.

#### **Response body**

The response contains an array of records for each object returned, including the following information for each record.

Name	Туре	Description
apiName	string	The object's unique name, such as Account
key	ID	The first 3 characters of the sObject's ID that indicates the object type.
label	string	The object's plural label, such as Accounts.
lastUpdatedId	string	A unique code that can be used in subsequent calls to compare the results for the new result set with the current results for this object.
qualifiedApiName	string	A unique external name for the sObject.
recordIds	ID	A comma-separated list of IDs for the matching records.

#### **Example**

See View Relevant Items.

## Retrieve Knowledge Language Settings

Returns the existing Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.

## **Syntax**

#### URI

/services/data/v31.0/knowledge Management/settings

#### Available since release

31.0

#### **Formats**

JSON, XML

#### **HTTP** methods

**GET** 

#### Authentication

Authorization: Bearer token

#### Request body

None required

#### **Request parameters**

None

## **Example for Getting KnowledgeSettings**

```
curl https://yourInstance.salesforce.com/services/data/v31.0/knowledgeManagement/settings -H "Authorization: Bearer token"
```

## **Example JSON Response Body**

```
"defaultLanguage" : "en_US",
    "knowledgeEnabled" : true,
    "languages" : [ {
        "active" : true,
        "name" : "en_US"
        }, {
        "active" : true,
        "name" : "it"
        }, {
        "active" : true,
        "name" : "it"
        }, {
        "active" : true,
        "name" : "zh_CN"
        }, {
```

Reference Search

```
"active" : true,
"name" : "fr"
} ]
```

### Usage

Salesforce Knowledge must be enabled in your organization. This resource can be used in API version 31.0 and later. It retrieves the Knowledge language settings, including the default knowledge language and a list of supported Knowledge language information.

### Search

Executes the specified SOSL search. The search string must be URL-encoded.

For more information on SOSL see the SOQL and SOSL Reference.

## **Syntax**

#### URI

```
/vXX.X/search/?q=SOSL search string
```

#### **Formats**

JSON, XML

#### **HTTP Method**

**GET** 

#### **Authentication**

```
Authorization: Bearer token
```

#### **Parameters**

Parameter	Description	
q	A SOSL statement that is properly URL-encoded.	

#### Example

See Search for a String on page 66.

## Search Scope and Order

Returns an ordered list of objects in the default global search scope of a logged-in user. Global search keeps track of which objects the user interacts with and how often and arranges the search results accordingly. Objects used most frequently appear at the top of the list.

The returned list reflects the object order in the user's default search scope, including any pinned objects on the user's search results page. This call is useful if you want to implement a custom search results page using the optimized global search scope. The search string must be URL-encoded.

Reference Search Result Layouts

## **Syntax**

URI

/vXX.X/search/scopeOrder

**Formats** 

JSON, XML

**HTTP Method** 

**GET** 

Authentication

Authorization: Bearer token

Example

See Get the Default Search Scope and Order.

## Search Result Layouts

Returns search result layout information for the objects in the query string. For each object, this call returns the list of fields displayed on the search results page as columns, the number of rows displayed on the first page, and the label used on the search results page.

This call supports bulk fetch for up to 100 objects in a query.

## **Syntax**

URI

/vXX.X/search/layout/?q=Comma delimited object list

**Formats** 

JSON, XML

**HTTP Method** 

**GET** 

Authentication

Authorization: Bearer token

#### **Response format**

Property	Туре	Description
field	String	Object and field name formatted with a period separating. For example:  Account.Name.
format	String	The type of date field, such as the date only or date and time. Only date related types are specified; otherwise, null.
label	String	Name as it appears to users
name	String	API name

Reference Lightning Toggle Metrics

#### Example

Get Search Result Layouts for Objects

## **Lightning Toggle Metrics**

Return details about users who switched between Salesforce Classic and Lightning Experience.

## **Syntax**

URI

/services/data/vXX.X/sobjects/LightningToggleMetrics

Available since release

44.0

**Formats** 

JSON, XML

**HTTP** methods

**GET** 

Authentication

Authorization: Bearer token

**Request body** 

SOQL query.

#### **Request parameters**

Parameter	Description
UserId	The user ID.
RecordCount	The count of records returned.
MetricsDate	The date the switch was recorded.
Action	Did the user switch to Salesforce Classic or Lightning Experience.

## Example

This query returns the total number of switches to Salesforce Classic:

SELECT sum(RecordCount) Total FROM LightningToggleMetrics WHERE MetricsDate = LAST\_MONTH AND Action = 'switchToAloha'

## Usage

Use this object with the following APIs:

- Platform
- Metadata API

Tooling API

## Lightning Usage by App Type

Return the total number of Lightning Experience and Salesforce Mobile users.

### **Syntax**

URI

/services/data/vXX.X/sobjects/LightningUsageByAppTypeMetrics

Available since release

44.0

**Formats** 

JSON, XML

**HTTP** methods

**GET** 

**Authentication** 

Authorization: Bearer token

**Request body** 

SOQL query.

#### **Request parameters**

Parameter	Description	
AppExperience	The app used:	
	Salesforce Mobile	
	Lightning Experience	
MetricsDate	The date the data was recorded.	
UserID	The user ID.	

## Example

This query returns the daily active users by profile for Mobile:

```
SELECT MetricsDate, user.profile.name, COUNT_DISTINCT(user.id) Total FROM
LightningUsageByAppTypeMetrics WHERE MetricsDate = LAST_N_DAYS:30 AND AppExperience =
'Salesforce Mobile' GROUP BY MetricsDate, user.profile.name
```

## Usage

Use this object with the following APIs:

Platform

- Metadata API
- Tooling API

# Lightning Usage by Browser

Return Lightning Experience usage results grouped by browser instance.

## Syntax

#### URI

/services/data/vXX.X/sobjects/LightningUsageByBrowserMetrics

#### Available since release

44.00

#### **Formats**

JSON, XML

#### **HTTP** methods

**GET** 

#### **Authentication**

Authorization: Bearer token

### Request body

SOQL Query.

#### **Request parameters**

Parameter	Description
Browser	The browser used.
EptBin3To5	Number of times that a page loaded between 3-5 seconds.
EptBin5To8	Number of times that a page loaded between 5-8 seconds.
EptBin8To10	Number of times that a page loaded between 8-10 seconds.
EptBinOver10	Number of times that a page loaded over 10 seconds.
EptBinUnder3	Number of times that a page loaded under 3 seconds.
MetricsDate	The date the metric was recorded.
PageName	The name of the page.
RecordCountEPT	Number of records for a page/browser where the valid EPT was recorded.
SumEPT	Sum of the EPT values for page/browser.
TotalCount	Total records for a page/browser.

Reference Lightning Usage by Page

## Example

This query returns browser distribution details, for the last 3 months.

SELECT CALENDAR\_MONTH(MetricsDate) MetricsDate, Browser Browser, SUM(TotalCount) Total FROM LightningUsageByBrowserMetrics WHERE MetricsDate = Last\_N\_Months:3 AND (NOT Browser like 'OTHER%') GROUP BY CALENDAR MONTH(MetricsDate), Browser

## Usage

Use this object with the following APIs:

- Platform
- Metadata API
- Tooling API

## Lightning Usage by Page

Represents standard pages users viewed most frequently in Lightning Experience.

## **Syntax**

URI

/services/data/vXX.X/sobjects/LightningUsageByPageMetrics

#### Available since release

44.00

#### **Formats**

JSON, XML

#### **HTTP** methods

GET

#### Authentication

Authorization: Bearer token

#### **Request body**

SOQL Query.

Parameter	Description
EptBin3To5	Number of times that a page loaded between 3-5 seconds.
EptBin5To8	Number of times that a page loaded between 5-8 seconds.
EptBin8To10	Number of times that a page loaded between 8-10 seconds.
EptBinOver10	Number of times that a page loaded over 10 seconds.
EptBinUnder3	Number of times that a page loaded under 3 seconds.
PageName	The name of the page.

Parameter	Description		
MetricsDate	The date the metric was recorded.		
RecordCountEPT	Number of records for a page/user where the valid EPT was recorded.		
SumEPT	Sum of the EPT values for a page/user.		
TotalCount	Total records for a page/user.		
UserId	User ID.		

## Example

This example returns the top 10 most visited pages and how many times each page was visited.

SELECT TotalCount FROM LightningUsageByPageMetrics ORDER BY PageName ASC NULLS FIRST LIMIT 10

## Usage

Use this object with the following APIs:

- Platform
- Metadata API
- Tooling API

## Lightning Usage by FlexiPage

Return details about the custom pages viewed most frequently in Lightning Experience.

## **Syntax**

URI

/services/data/vXX.X/sobjects/LightningUsageByFlexiPageMetrics

Available since release

44.00

**Formats** 

JSON, XML

**HTTP** methods

GET

Authentication

Authorization: Bearer token

SOQL query.

#### **Request parameters**

Parameter	Description	
FlexiPageNameOrId	Namespace and file name, or Page ID of FlexiPage files.	
FlexiPageType	The FlexiPage type. For example, record details are displayed using RecordPage" type.	
MetricsDate	The date the metric was recorded.	
RecordCountEPT	Number of records for a FlexiPage type, where the valid EPT was recorded.	
SumEPT	Sum of the EPT values for a record	
TotalCount	Total records for a type.	

## Example

This guery returns the top 10 most viewed custom pages over the past 7 days.

SELECT FlexiPageNameOrId FlexiPageNameOrId, SUM(TotalCount) Total FROM
LightningUsageByFlexiPageMetrics WHERE MetricsDate = Last\_N\_DAYS:7 AND (NOT FlexiPageNameOrId
= 'unknown unknown') AND (NOT FlexiPageNameOrId = 'unknown | unknown') GROUP BY
FlexiPageNameOrId ORDER BY SUM(TotalCount) Desc Limit 10

## Usage

Use this object with the following APIs:

- Platform
- Metadata API
- Tooling API

## Lightning Exit by Page Metrics

Return frequency metrics about the standard pages within which users switched from Lightning Experience to Salesforce Classic.

### **Syntax**

#### URI

/services/data/vXX.X/sobjects/LightningExitByPageMetrics

#### Available since release

44.0

#### **Formats**

JSON, XML

#### **HTTP** methods

GET

**Authentication** 

Authorization: Bearer token

Request body
SOQL query.

#### **Request parameters**

Parameter	Description		
MetricsDate	The date the data was recorded.		
PageName	Current Page from which User Switched from Lightning to Aloha		
RecordCount	The number of records per user and page.		
UserId	The user ID.		

## Example

This query returns the top 10 pages that were switched from Lightning Experience to Salesforce Classic over the past 7 days.

SELECT PageName PageName, SUM(RecordCount) Total FROM LightningExitByPageMetrics WHERE MetricsDate = Last\_N\_DAYS:7 GROUP BY PageName ORDER BY SUM(RecordCount) Desc Limit 10

## Usage

Use this object with the following APIs:

- Platform
- Metadata API
- Tooling API

## **Lightning Scheduler Resources**

Use Lightning Scheduler REST APIs to get appointment time slots or available resources based your work type group and territories.

#### IN THIS SECTION:

#### Scheduling

Returns a list of available lightning scheduler REST resources and corresponding URIs.

#### **Get Appointment Slots**

Returns a list of available appointment time slots for a resource based on given work type group and territories.

#### **Get Appointment Candidates**

Returns a list of available service resources (appointment candidates) based on work type group and service territories.

#### Request Bodies

Reference Scheduling

**Response Bodies** 

SEE ALSO:

Set Up Lightning Scheduler

## Scheduling

Returns a list of available lightning scheduler REST resources and corresponding URIs.

### **Syntax**

#### URI

/services/data/vxx.x/scheduling/

#### Available since release

45.0

#### **Formats**

JSON, XML

#### **HTTP** methods

**GET** 

#### Authentication

```
Authorization: Bearer token
```



Note: The Lightning Platform REST API supports OAuth 2.0 (an open protocol to allow secure API authorization). See Understanding Authentication on page 5 for more details.

#### **Response Body**

Execution of a successful request returns names of resources and their URIs as key-value pairs.

#### **JSON Example**

```
"getAppointmentCandidates": "/services/data/v45.0/scheduling/getAppointmentCandidates",
"getAppointmentSlots" : "/services/data/v45.0/scheduling/getAppointmentSlots"
```

## **Get Appointment Slots**

Returns a list of available appointment time slots for a resource based on given work type group and territories.

### **Syntax**

URI

/services/data/**vXX.X**/scheduling/getAppointmentSlots

#### Available since release

45.0

Reference Get Appointment Slots

#### **Formats**

JSON, XML

#### **HTTP** methods

**POST** 

#### Authentication

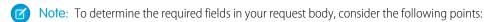
Authorization: Bearer token



**Note:** The Lightning Platform REST API supports OAuth 2.0 (an open protocol to allow secure API authorization). See Understanding Authentication for more details.

#### **Request body**

Parameter	Required	Туре	Description
startTime	No	String	The earliest time that a time slot can begin (inclusive). Defaults to the current time of the request, if empty.
endTime	No	String	The latest time that a time slot can end (inclusive).
workTypeGroupId	Required if workType is not given.	String	The ID of the work type group containing the work types that are being performed.
workType	Required if wakty copid is not given.	Work Type	The type of the work to be performed.
accountId	No	String	The ID of the associated account.
territoryIds	Yes	String[]	List of IDs of service territories, where the work that is being requested is performed.
requiredResourceIds	Yes	String[]	List of resource IDs that must be available during the time slot. At present, only one resource is allowed on this list.
schedulingPolicyId	No	String	The ID of the AppointmentSchedulingPolicy object. If no scheduling policy is passed in the request body, the default configurations are used. The only scheduling policy configuration that is used in determining time slots is the enforcement of account visiting hours.
allowConcurrentScheduling	No	Boolean	If true, allows scheduling of concurrent appointments in a time slot. If false, concurrent appointments are not allowed. The default is false. Available in API version 47.0 and later.



- Provide either workTypeGroupId or workType in your request body, but not both
- If workType is given, then you must provide either id or durationInMinutes, but not both.
- If id of the workType is given, the rest of workType fields are optional.

Reference Get Appointment Slots

#### **Response Body**

Execution of a successful request returns the response body containing a list of available time slots.

Parameter	Required	Туре	Description
timeSlots	Yes	Time Slots on page 232 [ ]	List of time slots included in each territory.

### **Examples**

#### **Request Body Example**

Using workTypeGroupId:

```
{
    "startTime" : "2019-01-23T00:00:00.000Z",
    "endTime" : "2019-02-30T00:00:00.000Z",
    "workTypeGroupId" : "0VSB0000000KyjBOAS",
    "accountId" : "001B000000qAUAWIA4",
    "territoryIds" : ["0HhB0000000T09WKAW"],
    "schedulingPolicyId" : "0VrB0000000KyjB",
    "requiredResourceIds" : ["0HnB0000000T08gKAK"]
}
```

Using workType:

```
{
"startTime" : "2019-01-23T00:00:00.000Z",
"endTime" : "2019-02-30T00:00:00.000Z",
"workType" : {
    "id" : "08qRM00000003fkYAA"
    },
"requiredResourceIds" : ["0HnB0000000T08gKAK"],
"territoryIds" : ["0HnRM00000030Z0AY"]
"accountId" : "001B000000qAUAWIA4",
"schedulingPolicyId" : "0VrB0000000KyjB"
}
```

Using durationInMinutes and without workTypeGroupId or workType idfields:

```
{
"workType" : {
    "durationInMinutes" : 60
},
"territoryIds" : ["0HhRM00000030Z0AY"],
"requiredResourceIds" : ["0HnB000000T08gKAK"]
}
```

#### **Response Body Example**

```
{
"timeSlots" :
[
    {
```

Reference Get Appointment Slots

```
"endTime" : "2019-01-21T19:15:00.000+0000",
    "startTime" : "2019-01-21T16:15:00.000+0000",
    "territoryId" : "0HhB000000T09WKAW"
}, {
    "endTime" : "2019-01-21T19:30:00.000+0000",
    "startTime" : "2019-01-21T16:30:00.000+0000",
    "territoryId" : "0HhB0000000T09WKAW"
}, {
    "endTime" : "2019-01-21T19:45:00.000+0000",
    "startTime" : "2019-01-21T16:45:00.000+0000",
    "territoryId" : "0HhB0000000T09WKAW"
}
```

### **Prerequisites**

The appointment time slots are determined based on your Lightning Scheduler data model configurations. Here are some prerequisites that you can consider while setting up data.

- Set up Lightning Scheduler before making your requests. This setup includes creating or configuring Service Resources, Service
  Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Set Up
  Lightning Scheduler for more information.
- Configure a work type mapped for each territory in the request body via Service Territory Work Type. Map the same work type to the work type group, via work type group member.

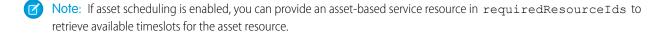
### **Rules and Guidelines**

The following factors affect how time slots are calculated and returned.

- Timezones that differ across operating hours are handled and results are always returned in UTC.
- The resource must be marked as a required resource on the assigned resource object.
- The resource is considered unavailable If the status categories of the resource assigned to service appointments are other than Canceled, Cannot Complete, and Completed.
- Resource Absences of all types are considered unavailable from start to end.
- The following fields of Work Type records, if configured, are used to fine-tune time slot requirements. For more information, see Create Work Types in Lightning Scheduler.

Parameter	Description
Timeframe Start	Time slots sooner than current time + <b>Timeframe Start</b> are not returned.
Timeframe End	Time slots later than current time + <b>Timeframe End</b> are not returned.
Block Time Before Appointment	The time period before the appointment is considered as unavailable.
Block Time After Appointment	The time period after the appointment is considered as unavailable.
Operating Hours	The overlap of all operating hours from the account, work type, service territory, and service territory member are considered while determining time slots. For more information, see Operating Hours Considerations in Lightning Scheduler.

- Only the time slots within the period of 28 days are returned.
- The earliest and latest appointment slots are calculated using the following guidelines:
  - The earliest appointment slot = maximum of
    - Service Territory Member Start Date
    - current time + Time frame Start
    - Start Time passed in the request body
  - The latest appointment slot = minimum of
    - Service Territory Member End Date
    - the earliest appointment slot date + 28 days
    - current time + Timeframe End



## **Get Appointment Candidates**

Returns a list of available service resources (appointment candidates) based on work type group and service territories.

### **Syntax**

#### URI

/services/data/vxx . x/scheduling/getAppointmentCandidates

#### Available since release

45.0

#### **Formats**

JSON, XML

#### **HTTP** methods

POST

#### Authentication

Authorization: Bearer token



**Note:** The Lightning Platform REST API supports OAuth 2.0 (an open protocol to allow secure API authorization). See Understanding Authentication for more details.

#### **Request body**

Parameter	Required Type	Description
workTypeGroupId	Required if String workType is not given.	The ID of the work type group containing the work types that are being performed.
workType	Required if Work Type waktyeeapta	The type of the work to be performed.

Parameter	Required	Type Description	
	is not given.		
territoryIds	Yes	String[]	List of service territory IDs, where the work that is being requested is performed.
schedulingPolicyId	No	String	The ID of the AppointmentSchedulingPolicy object. If no scheduling policy is passed in the request body, the default configurations are used. All Scheduling Policy Configurations are considered when using this API.
startTime	No	String	The earliest time that a time slot can begin (inclusive). Defaults to the current time of the request, if empty.
endTime	No	String	The latest time that a time slot can end (inclusive).
accountId	No	String	The ID of the associated account.
allowConcurrent Scheduling	No	Boolean	If true, allows scheduling of concurrent appointments in a time slot. If false, concurrent appointments are not allowed. The default is false.



Note: To determine the required fields in your request body, consider the following points:

- Provide either workTypeGroupId or workType in your request body, but not both
- If workType is given, then you must provide either id or durationInMinutes, but not both.
- If id of the workType is given, the rest of workType fields are optional.

#### **Response Body**

Execution of a successful request returns the response body containing a list of available appointment resources.

Parameter	Required	Туре	Description
candidates	Yes	Candidates on page 233 [ ]	List of available appointment candidates.

## **Examples**

#### **Request Body Example**

Using workTypeGroupId:

```
"startTime": "2019-01-23T00:00:00.000Z",
"endTime" : "2019-02-30T00:00:00.000Z",
"workTypeGroupId" : "OVSB0000000KyjBOAS",
"accountId" : "001B000000qAUAWIA4",
"territoryIds" : ["0HhB000000TO9WKAW"],
"schedulingPolicyId" : "0VrB0000000KyjB"
```

Using workType:

```
"startTime" : "2019-01-23T00:00:00.000Z",
"endTime" : "2019-02-30T00:00:00.000Z",
"workType" : {
    "id" : "08qRM00000003fkYAA"
    },
"territoryIds" : ["0HhRM00000030Z0AY"],
"accountId" : "001B000000qAUAWIA4",
"schedulingPolicyId" : "0VrB00000000KyjB"
}
```

Using durationInMinutes and without workTypeGroupId or workType id fields:

#### **Response Body Example**

```
"candidates" : [ {
   "endTime": "2019-01-23T19:15:00.000+0000",
   "resources" : [ "0HnB000000D2DsKAK" ],
   "startTime" : "2019-01-23T16:15:00.000+0000",
   "territoryId" : "OHhB000000TO9WKAW"
 }, {
   "endTime": "2019-01-23T19:30:00.000+0000",
   "resources" : [ "0HnB000000D2DsKAK" ],
   "startTime": "2019-01-23T16:30:00.000+0000",
   "territoryId" : "OHhB000000TO9WKAW"
 }, {
   "endTime": "2019-01-23T19:45:00.000+0000",
   "resources" : [ "0HnB000000D2DsKAK" ],
   "startTime": "2019-01-23T16:45:00.000+0000",
   "territoryId" : "OHhB000000TO9WKAW"
 } ]
}
```

### **Prerequisites**

The appointment time slots are determined based on your Lightning Scheduler data model configurations. Here are some prerequisites that you can consider while setting up data.

Reference Request Bodies

• Set up Lightning Scheduler before making requests. This setup includes creating or configuring Service Resources, Service Territory Members, Work Type Groups, Work Types, Work Type Group Members, and Service Territory Work Types. See Set Up Lightning Scheduler for more information.

- The territory type of the Service Territory Member must be either Primary, Secondary, or Relocation. The Primary and Relocation territory types are considered the same.
- The territory type of the Service Territory Member must be either Primary, Secondary, or Relocation. The Primary and Relocation territory types are considered the same.

#### **Rules and Guidelines**

The following factors are considered for returning start time and end time of resources.

#### **Resource Availability**

Determined using service territory member, service territory, work type, and account operating hours fields.

#### **Resource Unavailability**

Determined by resource absences, existing appointments that the resource is assigned to. The resource must be marked as a required resource for the appointment with a status that is not in closed, canceled, or completed.

#### **Appointment Start Time Interval in the Scheduling Policy**

Appointment start time interval field in the Scheduling Policy is used to determine when the appointment can start. This interval can be 5, 10, 15, 20, 30, or 60. By default, it is set to 15.

#### **Work Type Duration**

The end time is calculated as start time + duration of the work type.



**Note**: If asset scheduling is enabled, the response also includes asset-based candidates.

## **Request Bodies**

To perform a POST, PATCH, or PUT request, create a request body formatted in either XML or JSON. This chapter lists the request bodies.

#### IN THIS SECTION:

Work Type

Details about the type of work to be performed.

Skill Requirement

List of skills that are required to complete a particular task for a work type.

## Work Type

Details about the type of work to be performed.

Name	Туре	Required	Description
id	String	Required if durationInMinutes is not given.	ld of the work type.
durationInMinutes	String	Required if id is not given.	Contains the event length, in minutes.

Reference Response Bodies

Name	Туре	Required	Description
timeframeStartInMinutes	String	No	The beginning of the timeframe.
timeframeEndInMinutes	String	No	The end of the timeframe.
blokinesceppointertiMntes	String	No	The time period before the appointment is considered as unavailable.
blokTneAfeApointnertIMntes	String	No	The time period after the appointment is considered as unavailable.
operatingHoursId	String	No	The overlap of all operating hours from the account, work type, service territory, and service territory member are considered while determining time slots.
skillRequirements	Skill Requirement[]	No	List of skills that are required to complete a particular task for a work type.



Note: Provide either Id or durationInMinutes in the request body, but not both.

### Skill Requirement

List of skills that are required to complete a particular task for a work type.

Name	Туре	Required	Description
skillId	String	Yes	The skill that is required.
SkillLevel	String	No	The level of the skill required. Skill levels can range from zero to 99.99. Depending on your business needs, you might want the skill level to reflect years of experience, certification levels, or license classes.

## **Response Bodies**

Successful execution of a request to a lightning scheduler resource can return a response body either in JSON or XML format. For example, the request to get appointment time slots returns a list of available time slots for a selection of work type group and territories.

#### IN THIS SECTION:

Time Slots

Describes the result of Get Appointments Slots request.

Candidates

Describes the result of Get Appointments Candidates request.

#### **Time Slots**

Describes the result of Get Appointments Slots request.

List of time slots available for each territory.

Name	Туре	Description
startTime	String	The start time of the appointment time slot.
endTime	String	The end time of the appointment time slot.
territoryId	String	The service territory associated with this time slot.

SEE ALSO:

**Get Appointment Slots** 

#### **Candidates**

Describes the result of Get Appointments Candidates request.

List of available service resources.

Name	Туре	Description
startTime	String	The start time of the appointment time slot.
endTime	String	The end time of the appointment time slot.
territoryId	String	The service territory associated with this resource.
resources	String[]	List of service resource IDs that are available.
		① Important: At present, only one resource is returned on this list. If there is more than one resource included in a territory, a new child object is added for each resource in the response JSON body.

SEE ALSO:

**Get Appointment Candidates** 

## Search for Records Suggested by Autocomplete and Instant Results

Returns a list of suggested records whose names match the user's search string. The suggestions resource provides autocomplete results and instant results for users to navigate directly to likely relevant records, before performing a full search.

## **Syntax**

URI

 $\verb|vXX.X/search/suggestions?q=|search_string| & \verb|sobject=|object_types||$ 

#### Available since release

32.0

**Formats** 

JSON, XML

**HTTP** methods

GET

Authentication

Authorization: Bearer token

Request body

None required

#### **Request parameters**

Parameter	Description
fields	Optional. Used for creating lookup queries. Specify multiple fields using a comma-separated list. Specifies which lookup fields to be returned in the response.
groupId	Optional. Specifies one or more unique identifiers of one or more groups that the question to return was posted to. Specify multiple groups using a comma-separated list. This parameter is only applicable when the parameter type equals question. Don't use with the userId.
ignoreUnsupportedSObjects	Optional. Specifies what to do if unsupported objects are included in the request. If false and an unsupported object is included, an error is returned. If true and an unsupported object is included, the object is ignored and no error is returned. See the Unsupported Objects section for reference. The default is false.
limit	Optional. Specifies the maximum number of suggested records to return. If a limit isn't specified, 5 records are returned by default. If there are more suggested records than the limit specified, the response body's hasMoreResults property is true.
networkId	Optional. Specifies one or more unique identifiers for the community(ies) that the question to return is associated to. Specify multiple communities using a comma-separated list. This parameter is only applicable when the parameter type equals question or parameter sobject equals user.
ď	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, Korean, and Thai; three characters for all other languages. Query strings that exceed the maximum length of 255 characters (or 200 consecutive characters without a space break) return an error.
sobject	Required. The objects that the search is scoped to, such as Account or offerc.
	If the sobject value is feedItem, it is required to have the type parameter with a value of question.
	Specify up to 10 objects with a comma-separated list. For example: sobject=Account, Contact, Lead. To take advantage of this feature, activate the <b>CrossObjectTypeahead</b> permission.

Parameter	Description
	To specify the specific fields to return by object, use the following syntax with multiple fields in a comma-separated list. The $sobject$ is lowercase.
	sobject= <b>sobject</b> .fields= <b>fields</b>
	For example:
	&sobject=Account,Contact,Lead&account.fields=Website,Phone &contact.fields=Phone
topicId	Optional. Specifies the unique identifier of the single topic that the question to return was tagged as. This parameter is only applicable when the parameter type equals question.
type	Required when the sobject value is feedItem. Including this parameter for all other sobject values doesn't affect the query. Specifies that the type of Feed is questions. Valid value: question.
userId	Optional. Specifies one or more unique identifiers of one or more users who authored the question to return. Specify multiple users using a comma-separated list. This parameter is only applicable when the parameter type equals question. Don't use with the groupId.
useSearchScope	Optional. Available in API version 40.0 and later. The default value is false. If false, the objects specified in the request are used to suggest records. If true, in addition to the objects specified in the request, the user's search scope is used to suggest records. The search scope is the list of objects a user uses most frequently.
	<ul> <li>If the request doesn't specify an object, use useSearchScope=true.</li> </ul>
	<ul> <li>If useSearchScope=true and the user's search scope is empty, the default search scope is used to suggest records.</li> </ul>
	<ul> <li>Only the first 10 objects are used to suggest records.</li> </ul>
	• Objects specified in the sobject parameter are prioritized over objects in the user's search scope.
	<ul> <li>Values for the ignoreUnsupportedSObjects parameter aren't applied to the objects in the search scope.</li> </ul>
	This example uses only the search scope.
	/search/suggestions?q=Acme&useSearchScope=true
	This example uses the search scope and the Account object.
	/search/suggestions?q=Acme&sobject=Account&useSearchScope=true
where	Optional. A filter that follows the same syntax as the SOQL WHERE clause. URLs encode the expression.
	Use the clause for an object, or globally for all compatible objects. An example of an object-specific clause is:
	account.where=name $\$20$ LIKE $\$20\$27$ Smith $\$25\$27$ . An example of a global clause is: where=name $\$20$ LIKE $\$20\$27$ Smith $\$25\$27$ . The parameter

Darameter

Parameter	Description
	must be lower case. Any object-specific where clauses override the global where clause. You can't use this parameter for the Question object.
	To specify multiple entities, see the following example. This feature is available in version 38.0 and later.
	search/suggestions?q=Smith & Schject-Account, Contact, KnowledgeArticleVersion, CollaborationGroup, Topic, FeedItem
	<pre>// Specifies a global where clause (to filter Account and Contact)</pre>
	&where=name%20LIKE%20%27Smith%25%27
	// Overrides the global where clause for Knowledge Article
	(filtering by PublishStatus and Language is required for KnowledgeArticle)
	&knowledgearticleversion.where=PublishStatus='online'+and+language='en US'
	// Overrides the global where clause for Topic &topic.where=networkid=<1234567891>
	// Overrides the global where clause for CollaborationGroup
	&collaborationgroup.where=networkid=<1234567891>
	<pre>// FeedItem-Question doesn't support where clauses, but we can filter</pre>
	the type and networkId&type=question
	&networkId==<1234567891>

Description

## Usage

The suggestions resource returns records when the record's name field includes the exact text in the search string. The last term in the search string can match the beginning of a word. Records that contain the search string within a word aren't considered a match.



Note: If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.



**Solution Example**: The text string national u is treated as national u\* and returns "National Utility", "National Urban Company", and "First National University".

## Suggested Records Response

The suggestions resource returns display-ready data about likely relevant records that the user can access.

A relevance algorithm determines the order of results.

Each suggested record in the results contains these elements:

Element	Description
Attributes	The record's object type and the URL for accessing the record.
	Also includes the requested lookup fields' values. For example, if you requested fields=Id, Name, the result would include the ID and name.

Element	Description
Name (Or Title)	The record's Name field. In the absence of a standard Name field, the Title field is used for these objects:
	<ul> <li>Dashboard</li> </ul>
	• Idea
	<ul> <li>IdeaTheme</li> </ul>
	<ul> <li>Note</li> </ul>
	<ul> <li>Question</li> </ul>
	In the absence of a standard Name or Title field, the main identifying field is used. For example, in cases, the Case Number is used.
Id	The record's unique identifier.

## Example JSON Response Body

```
[ {
 "attributes" : {
   "type" : "Account",
   "url" : "/services/data/v32.0/sobjects/Account/001xx000003DH6WAAW"
 "Id" : "001xx000003DH6WAAW"
 "Name" : "National Utility Service"
}, {
 "attributes" : {
   "type" : "Account",
   "url": "/services/data/v32.0/sobjects/Account/001xx000003DHJ4AAO"
 "Id": "001xx000003DHJ4AAO"
 "Name" : "National Utility Service"
}, {
 "attributes" : {
   "type" : "Account",
   "url" : "/services/data/v32.0/sobjects/Account/001xx000003DHscAAG"
 },
 "Id" : "001xx000003DHscAAG"
 "Name" : "National Urban Technology Center"
} ]
```

## Example JSON Response Body for a Multiple Object Request

```
[ {
  "attributes" : {
    "type" : "Account",
    "url" : "/services/data/v38.0/sobjects/Account/001xx000003DMEKAA4"
},
  "Id" : "001xx000003DMEKAA4"
```

```
"Name" : "Joe Doe Printing"
}, {
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v38.0/sobjects/Account/001xx000003DLjvAAG"
},
    "Id" : "001xx000003DLjvAAGO"
    "Name" : "Joe Doe Plumbing"
}, {
    "attributes" : {
        "type" : "Contact",
        "url" : "/services/data/v38.0/sobjects/Contact/003xx000004U9Y9AAK"
},
    "Id" : "003xx000004U9Y9AAK"
    "Name" : "John Doe"
} ]
```

## Example XML Response Body

```
<?xml version="1.0" encoding="UTF-8"?</pre>
<suggestions>
 <autoSuggestResults type="Account"</pre>
url="/services/data/v32.0/sobjects/Account/001xx000003DH6WAAW">
   <Id>001xx000003DH6WAAW</Id>
   <Name>National Utility Service</Name>
 </autoSuggestResults>
 <autoSuggestResults type="Account"</pre>
url="/services/data/v32.0/sobjects/Account/001xx000003DHJ4AA0">
    <Id>001xx000003DHJ4AAO</Id>
    <Name>National Utility Service</Name>
 </autoSuggestResults>
 <autoSuggestResults type="Account"</pre>
url="/services/data/v32.0/sobjects/Account/001xx000003DHscAAG">
    <Id>001xx000003DHscAAG</Id>
    <Name>National Urban Technology Center</Name>
 </autoSuggestResults>
  <hasMoreResults>true</hasMoreResults>
</suggestions>
```

## **Unsupported Objects**

The suggestions resource supports all searchable objects except the following.

- ContentNote
- Event
- External objects
- FeedComment
- FeedPost
- IdeaComment

- Pricebook2
- Reply
- TagDefinition
- Task

## Search Suggested Article Title Matches

Returns a list of Salesforce Knowledge article titles that match the user's search query string. Provides a shortcut to navigate directly to likely relevant articles before the user performs a search.

## **Syntax**

#### URI

 $/ \texttt{vXX.X/search/suggestTitleMatches?q=} \textbf{\textit{string}} \& \texttt{language=} \textbf{\textit{article publication status}} \\$ 

#### Available since release

30.0

#### **Formats**

JSON, XML

#### **HTTP** methods

**GET** 

#### Authentication

Authorization: Bearer token

#### **Request body**

None required

#### **Request parameters**

Parameter	Description
articleTypes	Optional. Three-character ID prefixes indicating the desired article types. You can specify multiple values for this parameter in a single REST call, by repeating the parameter name for each value. For example, articleTypes=ka0&articleTypes=ka1.
categories	Optional. The name of the data category group and name of the data category for desired articles, expressed as a JSON mapping. You can specify multiple data category group and data category pairs in this parameter. For example, categories={"Regions":"Asia","Products":"Laptops"}. Characters in the URL might need to be encoded. For this example, categories=%7B%22Regions%22%3A%22Asia %22%2C%22Products%22%3A%22Laptops%22%7D.
channel	Optional. The channel where the matching articles are visible. Valid values:  • AllChannels-Visible in all channels the user has access to  • App-Visible in the internal Salesforce Knowledge application  • Pkb-Visible in the public knowledge base

Parameter	Description	
	Csp-Visible in the Customer Portal	
	• Prm—Visible in the Partner Portal	
	If channel isn't specified, the default value is determined by the type of user.	
	Pkb for a guest user	
	Csp for a Customer Portal user	
	<ul> <li>Prm for a Partner Portal user</li> </ul>	
	<ul> <li>App for any other type of user</li> </ul>	
	If channel is specified, the specified value may not be the actual value requested, because of certain requirements.	
	<ul> <li>For guest, Customer Portal, and Partner Portal users, the specified value must match the default value for each user type. If the values don't match or AllChannels is specified, then App replaces the specified value.</li> </ul>	
	<ul> <li>For all users other than guest, Customer Portal, and Partner Portal users:</li> </ul>	
	<ul> <li>If Pkb, Csp, Prm, or App are specified, then the specified value is used.</li> </ul>	
	<ul> <li>If AllChannels is specified, then App replaces the specified value.</li> </ul>	
language	Required. The language of the user's query. Specifies the language that matching articles are written in.	
limit	Optional. Specifies the maximum number of articles to return. If there are more suggested articles than the limit specified, the response body's hasMoreResults property is true.	
publishStatus	Required. The article's publication status. Valid values:	
	<ul> <li>Draft—Articles aren't published in Salesforce Knowledge.</li> </ul>	
	<ul> <li>Online—Articles are published in Salesforce Knowledge.</li> </ul>	
	<ul> <li>Archived-Articles aren't published and are available in Archived Articles view.</li> </ul>	
q	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: one character for queries in Chinese, Japanese, and Korean, and three characters for all other languages. Query strings exceeding the maximum length of 250 characters return an error.	
topics	Optional. The topic of the returned articles. For example: topics=outlook&topics=email.	
validationStatus	Optional. The validation status of returned articles.	

# Example for getting suggested articles with matching titles

 $\label{lem:curl_https://yourInstance.salesforce.com/services/data/v30.0/search/suggestTitleMatches? $$q=orange+banana&language=en_US&publishStatus=Online -H "Authorization: Bearer token"$$$ 

Reference Search Suggested Queries

## Example JSON response body

```
"autoSuggestResults" : [ {
    "attributes" : {
      "type" : "KnowledgeArticleVersion",
      "url" : "/services/data/v30.0/sobjects/KnowledgeArticleVersion/ka0D00000004CcQ"
      },
    "Id" : "ka0D00000004CcQ",
    "UrlName" : "orange-banana",
    "Title" : "orange banana",
    "KnowledgeArticleId" : "kA0D00000004Cfz"
      } ],
      "hasMoreResults" : false
}
```

## Usage

Salesforce Knowledge must be enabled in your organization. The user must have the "View Articles" permission enabled. The articles suggested include only the articles the user can access, based on the data categories and article types the user has permissions to view.

The Suggest Article Title Matches resource is designed to return display-ready data about likely relevant articles. Articles are suggested if their titles contain the entire query string, except stopwords, such as "a," "for," and "the."

For example, a search for Backpacking for desert returns the article, "Backpacking in the desert."



**Note**: Articles with titles that include stopwords from the query string, such as "Backpacking for desert survival" in this example, appear before matching articles with titles that don't include the stopwords.

Stopwords at the end of the query string are treated as search terms.

A wildcard is automatically appended to the last token in the query string.



**Note:** If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI along with any other special characters.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called hasMoreResults. Its value is true if the suggestions returned are only a subset of the suggestions available, and false otherwise.

SEE ALSO:

**SObject Suggested Articles** 

## Search Suggested Queries

Returns a list of suggested searches based on the user's query string text matching searches that other users have performed in Salesforce Knowledge. Provides a way to improve search effectiveness, before the user performs a search.

### **Syntax**

URI

vXX.X/search/suggestSearchQueries?q=search string&language=language of query

Reference Search Suggested Queries

Available since release

30.0

**Formats** 

JSON, XML

**HTTP** methods

GET

Authentication

Authorization: Bearer token

**Request body** 

None required

### **Request parameters**

Parameter	Description
channel	Optional. Specifies the Salesforce Knowledge channel where the article can be viewed Valid values:
	<ul> <li>AllChannels-Visible in all channels the user has access to</li> </ul>
	<ul> <li>App  —Visible in the internal Salesforce Knowledge application</li> </ul>
	<ul> <li>Pkb–Visible in the public knowledge base</li> </ul>
	<ul> <li>Csp-Visible in the Customer Portal</li> </ul>
	• Prm—Visible in the Partner Portal
	If channel isn't specified, the default value is determined by the type of user.
	Pkb for a guest user
	Csp for a Customer Portal user
	• Prm for a Partner Portal user
	<ul> <li>App for any other type of user</li> </ul>
	If channel is specified, the specified value may not be the actual value requested because of certain requirements.
	<ul> <li>For guest, Customer Portal, and Partner Portal users, the specified value must mate the default value for each user type. If the values don't match or AllChannel is specified, then App replaces the specified value.</li> </ul>
	<ul> <li>For all users other than guest, Customer Portal, and Partner Portal users:</li> </ul>
	<ul> <li>If Pkb, Csp, Prm, or App are specified, then the specified value is used.</li> </ul>
	<ul> <li>If AllChannels is specified, then App replaces the specified value.</li> </ul>
language	Required. The language of the user's query.
limit	Optional. Specifies the maximum number of suggested searches to return. If there a more suggested queries than the limit specified, the response body's hasMoreResults property is true.
d	Required. The user's search query string, properly URL-encoded. Suggestions are returned only if the user's query string meets the minimum length requirements: or character for queries in Chinese, Japanese, and Korean, and three characters for all

Reference Tabs

Parameter	Description	
	other languages. Query strings exceeding the maximum length of 250 characters return	
	an error.	

## Example for getting suggested queries

```
\label{lem:curl} $$ \operatorname{ttps://yourInstance.salesforce.com/services/data/v30.0/search/suggestSearchQueries? $$ q=app\&language=en_US -H "Authorization: Bearer $token"$ $$
```

## Example JSON response body

```
"autoSuggestResults" : [ {
    "0" : "apple",
    "1" : "apple banana",
} ],
    "hasMoreResults" : false
}
```

## Usage

Salesforce Knowledge must be enabled in your organization.

Queries are suggested if they exactly match the query string text. The text string must be a prefix within the query; it's not considered a match if it appears within a word. For example, the text string app would return suggested queries apple banana and banana apples but not pineapple.

If the number of suggestions returned exceeds the limit specified in the request, the end of the response contains a field called hasMoreResults. Its value is true if the suggestions returned are only a subset of the suggestions available, and false otherwise.

If the user's search query contains quotation marks or wildcards, those symbols are automatically removed from the query string in the URI.

### **Tabs**

Returns a list of all tabs—including Lightning page tabs—available to the current user, regardless of whether the user has chosen to hide tabs via the All Tabs (+) tab customization feature. This resource is available in REST API version 31.0 and later.

## **Syntax**

#### URI

/vXX.X/tabs/

#### **Formats**

JSON, XML

#### **HTTP** methods

GET, HEAD

Reference Tabs

#### Authentication

Authorization: Bearer token

#### Request body

None

#### **Request parameters**

None

### Example

#### Usage for getting tabs

/services/data/v31.0/tabs

#### Sample JSON Response body for /vXX.X/tabs/

This is a partial code sample, representing the Accounts tab.

```
[...,
 "colors" : [ {
   "color" : "6f7ccb",
   "context" : "primary",
   "theme" : "theme4"
 }, {
   "color" : "236FBD",
   "context" : "primary",
   "theme" : "theme3"
 } ],
 "custom" : false,
 "iconUrl" : "https://yourInstance.salesforce.com/img/icon/accounts32.png",
 "icons" : [ {
   "contentType" : "image/png",
   "height" : 32,
   "theme" : "theme3",
   "url" : "https://yourInstance.salesforce.com/img/icon/accounts32.png",
   "width" : 32
 }, {
   "contentType" : "image/png",
   "height" : 16,
   "theme" : "theme3",
   "url" : "https://yourInstance.salesforce.com/img/icon/accounts16.png",
   "width" : 16
 }, {
   "contentType" : "image/svg+xml",
   "height" : 0,
   "theme" : "theme4",
   "url": "https://yourInstance.salesforce.com/img/icon/t4/standard/account.svg",
   "width" : 0
 }, {
   "contentType" : "image/png",
   "height" : 60,
   "theme" : "theme4",
   "url" : "https://yourInstance.salesforce.com/img/icon/t4/standard/account 60.png",
```

Reference Themes

```
"width" : 60
}, {
    "contentType" : "image/png",
    "height" : 120,
    "theme" : "theme4",
    "url" : "https://yourInstance.salesforce.com/img/icon/t4/standard/account_120.png",
    "width" : 120
} ],
    "label" : "Accounts",
    "miniIconUrl" : "https://yourInstance.salesforce.com/img/icon/accounts16.png",
    "name" : "standard-Account",
    "sobjectName" : "Account",
    "url" : "https://yourInstance.salesforce.com/001/o",
...]
```

## **Themes**

Gets the list of icons and colors used by themes in the Salesforce application. Theme information is provided for objects in your organization that use icons and colors in the Salesforce UI.

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

# **Syntax**

### URI

/vXX.X/theme

## Available since release

29.0

### **Formats**

JSON, XML

### **HTTP** methods

**GET** 

#### Authentication

Authorization: Bearer token

### Request body

None

## **Request parameters**

None

#### Response data

An array of theme items. Each theme item contains the following fields:

Name	Туре	Description
colors	array of theme colors	Array of theme colors.

Reference Themes

Name	Туре	Description	
icons	array of theme icons	Array of theme icons.	
name string		Name of the object that the theme colors and icons are associated with.	

# Each theme color contains the following fields:

Name	Туре	Description
color	string	The color described in Web color RGB format, for example, "00FF00".
context	string	The color context, which determines whether the color is the main color ("primary") for the object, or not.
theme	string	<ul> <li>theme2—Theme used prior to Spring '10, called the "Salesforce Classic 2005 user interface theme"</li> <li>theme3—Theme introduced in Spring '10, called the "Salesforce Classic 2010 user interface theme"</li> <li>theme4—Theme introduced in Winter '14 for the mobile touchscreen version of Salesforce, and in Winter '16 for Lightning Experience</li> <li>custom—Theme associated with a custom icon</li> </ul>

## Each theme icon contains the following fields:

Name	Туре	Description
contentType	string	The icon's content type, for example, "image/png."
height	number	The icon's height in pixels. If the icon content type is an SVG type, height and width values are not used.
theme	string	The associated theme. Possible values include:
		• theme2—Theme used prior to Spring '10, called the "Salesforce Classic 2005 user interface theme"
		• theme3—Theme introduced in Spring '10, called the "Salesforce Classic 2010 user interface theme"
		<ul> <li>theme 4—Theme introduced in Winter '14 for the mobile touchscreen version of Salesforce, and in Winter '16 for Lightning Experience</li> </ul>
		• custom—Theme associated with a custom icon
url	string	The fully qualified URL for this icon.
width	number	The icon's width in pixels. If the icon content type is an SVG type, height and width values are not used.

Reference Composite Resources

# Example

The following is an example JSON response using a request of services/data/v29.0/theme:

```
"themeItems" : [
        "name" : "Merchandise c",
        "icons" : [
            "contentType" : "image/png",
            "width" : 32,
            "url": "https://yourInstance.salesforce.com/img/icon/computer32.png",
            "height" : 32,
            "theme" : "theme3"
        },
        {
            "contentType" : "image/png",
            "width" : 16,
            "url" : "https://yourInstance.salesforce.com/img/icon/computer16.png",
            "height" : 16,
            "theme" : "theme3"
        } ],
        "colors" : [
        {
            "context" : "primary",
            "color" : "6666CC",
            "theme" : "theme3"
        },
            "context" : "primary",
            "color" : "66895F",
            "theme" : "theme4"
        },
    }
}
```

# **Composite Resources**

Use REST API composite resources to improve your application's performance by minimizing the number of round-trips between the client and server.

## IN THIS SECTION:

## Composite

Executes a series of REST API requests in a single call. You can use the output of one request as the input to a subsequent request. The response bodies and HTTP statuses of the requests are returned in a single response body. The entire request counts as a single call toward your API limits.

### Batch

Executes up to 25 subrequests in a single request. The response bodies and HTTP statuses of the subrequests in the batch are returned in a single response body. Each subrequest counts against rate limits.

#### **SObject Tree**

Creates one or more sObject trees with root records of the specified type. An sObject tree is a collection of nested, parent-child records with a single root record.

#### **SObject Collections**

Executes actions on multiple records in one request. Use SObject Collections to reduce the number of round-trips between the client and server. This resource is available in API version 42.0 and later.

# Composite

Executes a series of REST API requests in a single call. You can use the output of one request as the input to a subsequent request. The response bodies and HTTP statuses of the requests are returned in a single response body. The entire request counts as a single call toward your API limits.

The requests in a composite call are called *subrequests*. All subrequests are executed in the context of the same user. In a subrequest's body, you specify a reference ID that maps to the subrequest's response. You can then refer to the ID in the url or body fields of later subrequests by using a JavaScript-like reference notation.

For example, the following composite request body includes two subrequests. The first creates an account and designates the output as refaccount. The second creates a contact parented under the new account by referencing refaccount in the subrequest body.

```
{
"compositeRequest" : [{
    "method" : "POST",
    "url" : "/services/data/v38.0/sobjects/Account",
    "referenceId" : "refAccount",
    "body" : { "Name" : "Sample Account" }
    }, {
    "method" : "POST",
    "url" : "/services/data/v38.0/sobjects/Contact",
    "referenceId" : "refContact",
    "body" : {
        "LastName" : "Sample Contact",
        "AccountId" : "@{refAccount.id}"
     }
    }]
}
```

You can specify whether an error in a subrequest causes the whole composite request to roll back or just the subrequests that depend on it. You can also specify headers for each subrequest.

Composite is supported for the following resources.

- All sObject resources (vxx.x/sobjects/)
- The Query resource (vXX.X/query/?q=soql)
- The QueryAll resource (vXX.X/queryAll/?q=soql)
- The SObject Collections resource (vXX.X/composite/sobjects). Available in API version 43.0 and later.
- Note: You can have up to 25 subrequests in a single call. Up to 5 of these subrequests can be query operations, including Query, QueryAll, and "Query More" requests to obtain the next batch of query results.

### URI

/vXX.X/composite

### **Formats**

JSON

## **HTTP** method

GET (lists other available composite resources), POST

### Authentication

Authorization: Bearer token

### **Parameters**

None required

## Request body

Composite Request Body

## **Response body**

Composite Response Body

## **Examples**

For examples of using the Composite resource, see Execute Dependent Requests in a Single API Call and Update an Account, Create a Contact, and Link Them with a Junction Object.

# Composite Request Body

Describes a collection of subrequests to execute with the Composite resource.

## Composite Collection Input

The request body contains an allorNone flag that specifies how to roll back errors and a compositeRequest collection that includes subrequests to execute.

## **Properties**

Name	Туре	Description	Required or Optional
allOrNone	Boolean	Specifies what to do when an error occurs while processing a subrequest. If the value is true, the entire composite request is rolled back. The top-level request returns HTTP 200 and includes responses for each subrequest.	Optional
		If the value is false, the remaining subrequests that don't depend on the failed subrequest are executed. Dependent subrequests aren't executed.	
		In either case, the top-level request returns HTTP 200 and includes responses for each subrequest.	
compositeRequest	Composite Subrequest[]	Collection of subrequests to execute.	Required

## JSON example

# **Composite Subrequest**

Contains the resource, method, headers, body, and reference ID for the subrequest.

## **Properties**

Name	Туре	Description	Required or Optional
body	Varies	The input body for the subrequest.	Optional
		The type depends on the request specified in the $url$ property.	
httpHeaders	Map <string, String&gt;</string, 	Request headers and their values to include with the subrequest. You can include any header supported by the requested resource except for the following three headers.  Accept Authorization	Optional
		• Content-Type	
		Subrequests inherit these three header values from the top-level request. Don't specify these headers in a subrequest. If you do, the top-level request fails and returns an HTTP 400 response.	
method	String	The method to use with the requested resource. Possible values are POST, PUT, PATCH, GET, and DELETE (case-sensitive). For a list of valid methods, refer to the documentation for the requested resource.	Required
referenceId	String	Reference ID that maps to the subrequest's response and can be used to reference the response in later subrequests. You can reference the referenceId in either the body or URL of a subrequest. Use this syntax to include a reference: @ { referenceId.FieldName }.	Required
		You can use two operators with the reference ID.	
		The operator references a field on a JSON object in the response. For example, let's say you retrieve an account record's data in one subrequest and assign the reference ID Account1Data to the output. You can refer to the account's	

Name	Туре	Description	Required or Optional
		name in later subrequests like this: @ { Account1Data . Name } .	
		The [] operator indexes a JSON collection in the response. For example, let's say you request basic account information with the SObject Basic Information resource in one subrequest and assign the reference ID AccountInfo to the output. Part of the response includes a collection of recently created accounts. You can refer to the ID of the first account in the recent items collection like this:  @{AccountInfo.recentItems[0].Id}.	
		You can use each operator recursively as long as it makes sense in the context of the response. For example, to refer to the billing city component of an account's compound address field:  @ {NewAccount.BillingAddress.city}.	
		referenceId is case-sensitive, so pay close attention to the case of the fields you're referring to.	
url	String	<ul> <li>The resource to request.</li> <li>The URL can include any query string parameters that the subrequest supports. The query string must be URL-encoded.</li> <li>You can use parameters to filter response bodies.</li> <li>The URL must start with /services/data/vxx.x/.</li> </ul>	Required

## **JSON** examples

```
"method" : "GET",
   "url" : "/services/data/v38.0/sobjects/Account/describe",
   "httpHeaders" : { "If-Modified-Since" : "Tue, 31 May 2016 18:00:00 GMT" },
   "referenceId" : "AccountInfo"
}

{
   "method" : "POST",
   "url" : "/services/data/v38.0/sobjects/Account",
   "referenceId" : "refAccount",
   "body" : { "Name" : "Sample Account" }
}

{
   "method" : "GET",
   "url" : "/services/data/v38.0/sobjects/Account/@{refAccount.id}",
   "referenceId" : "NewAccountFields"
}
```

## Usage

Because referenceId is case-sensitive, it's important to note the case of the fields that you're referring to. The same field can use different cases in different contexts. For example, when you create a record, the ID field appears as id in the response. But when you access a record's data with the SObject Rows resource, the ID field appears as Id. In the last example subrequest above, the @{refAccount.id} reference is valid because refAccount refers to the response from the POST in the second subrequest. If you use Id instead (mixed case rather than all lowercase), as in @{refAccount.Id}, you get an error when sending the request because the reference ID uses the wrong case.



**Note:** You can have up to 25 subrequests in a single call. Up to 5 of these subrequests can be query operations, including Query, QueryAll, and "Query More" requests to obtain the next batch of query results.

## Composite Response Body

Describes the result of a Composite request.

## Composite Results

## **Properties**

Name	Туре	Description
compositeResponse	Composite Subrequest Result[]	Collection of subrequest results

## JSON example

```
"compositeResponse" : [{
    Composite Subrequest Result
    }, {
    Composite Subrequest Result
    }, {
    Composite Subrequest Result
    }, {
    Composite Subrequest Result
}]
```

## Composite Subrequest Result

## **Properties**

Name	Туре	Description
body	The type depends on the response type of the subrequest.	The response body of this subrequest. See the documentation for the subrequest resource for more information.  If the subrequest returns an error, the body includes the error code and message. For more details on error responses, see Status Codes and
		Error Responses

Name	Туре	Description
httpHeaders	Map <string, string=""></string,>	Response headers and their values for this subrequest. The Composite resource doesn't support the Content-Length header, so subrequest responses don't include this header and neither does the top-level response.
httpStatusCode	Integer	An HTTP status code for this subrequest. If allOrNone is set to true in the composite request and a subrequest returns an error, all other subrequests return the 400 HTTP status code.
referenceID	String	The reference ID specified in the subrequest. This property lets you easily associate subrequests with their results.

## JSON example

The following example shows the response for a subrequest that successfully created an Account:

```
"body" : {
    "id" : "001R0000003316AIAU",
    "success" : true,
    "errors" : [ ]
},
    "httpHeaders" : {
      "Location" : "/services/data/v38.0/sobjects/Account/001R0000003316AIAU"
},
    "httpStatusCode" : 201,
    "referenceId" : "refAccount"
}
```

The following example shows the response for a subrequest that had an error while trying to create a Contact:

```
"body" : [ {
    "message" : "Email: invalid email address: Not a real email address",
    "errorCode" : "INVALID_EMAIL_ADDRESS",
    "fields" : [ "Email" ]
} ],
"httpHeaders" : { },
"httpStatusCode" : 400,
"referenceId" : "badContact"
}
```

## **Batch**

Executes up to 25 subrequests in a single request. The response bodies and HTTP statuses of the subrequests in the batch are returned in a single response body. Each subrequest counts against rate limits.

The requests in a batch are called subrequests. All subrequests are executed in the context of the same user. Subrequests are independent, and you can't pass information between them. Subrequests execute serially in their order in the request body. When a subrequest executes successfully, it commits its data. Commits are reflected in the output of later subrequests. If a subrequest fails, commits made

by previous subrequests are not rolled back. If a batch request doesn't complete within 10 minutes, the batch times out and the remaining subrequests aren't executed.

Batching for the following resources and resource groups is available in API version 34.0 and later.

- Limits—vXX.X/limits
- SObject resources—vXX.X/sobjects/
- Query—vXX.X/query/?q=soql
- QueryAll—vXX.X/queryAll/?q=soql
- Search—vXX.X/search/?q=sos1
- Connect resources—vXX.X/connect/
- Chatter resources—vXX.X/chatter/

Batching for the following resources and resource groups is available in API version 35.0 and later.

Actions—vXX.X/actions

The API version of the resource accessed in each subrequest must be no earlier than 34.0 and no later than the Batch version in the top-level request. For example, if you post a Batch request to /services/data/v35.0/composite/batch, you can include subrequests that access version 34.0 or 35.0 resources. But if you post a Batch request to

/services/data/v34.0/composite/batch, you can only include subrequests that access version 34.0 resources.

#### URI

/vXX.X/composite/batch

#### **Formats**

JSON, XML

#### HTTP method

POST

#### Authentication

Authorization: Bearer token

#### **Parameters**

None required

#### Request body

Batch Request Body on page 254

## **Response body**

Batch Response Body on page 256

## Examples

For an example of using the Batch resource, see Update a Record and Get Its Field Values in a Single Request on page 99.

# Batch Request Body

Describes a collection of subrequests to execute with the Batch resource.

## **Batch Collection Input**

The request body contains a batchRequests collection that includes subrequests to execute.

## **Properties**

Name	Туре	Description	Required or Optional
batchRequests	Subrequest[]	Collection of subrequests to execute.	Required
haltOnError	Boolean	Controls whether Salesforce should stop processing subrequests if a subrequest fails. The default is false.	Optional
		If the value is false and a subrequest in the batch doesn't complete, Salesforce attempts to execute the subsequent subrequests in the batch.	
		If the value is true and a subrequest in the batch doesn't complete due to an HTTP response in the 400 or 500 range, Salesforce halts execution. It returns an HTTP 412 status code and a BATCH_PROCESSING_HALTED error message for each subsequent subrequest. The top-level request to /composite/batch returns HTTP 200, and the hasErrors property in the response is set to true.	
		This setting is only applied during subrequest processing, and not during initial request deserialization. If an error is detected during deserialization, such as a syntax error in the Subrequest request data, Salesforce returns an HTTP 400 Bad Request error without processing any subrequests, regardless of the value of haltOnError. An example where this could occur is if a subrequest has an invalid method or url field.	

## **Root XML Tag**

<batch>

## JSON example

# Subrequest

Contains the resource, method, and accompanying information for the subrequest.

## **Properties**

Name	Туре	Description	Required or Optional
binaryPartName	String	The name of the binary part in the multipart request.	Optional
		When multiple binary parts are uploaded in one batch request, this value is used to map a request to its binary part. To prevent name collisions, use a unique value for each binaryPartName property in a batch request.	
		If this value exists, a binaryPartNameAlias value must also exist.	
binaryPartNameAlias	String	The name parameter in the Content-Disposition header of the binary body part. Different resources expect different values. See Insert or Update Blob Data.	Optional
		If this value exists, a binaryPartName value must also exist.	
method	String	The method to use with the requested resource. For a list of valid methods, refer to the documentation for the requested resource.	Required
richInput		The input body for the request.	Optional
		The type depends on the request specified in the ${\tt url}$ property.	
url	String	The resource to request.	Required
		• The URL can include any query string parameters that the subrequest supports. The query string must be URL-encoded.	
		You can use parameters to filter response bodies.	
		You cannot apply headers at the subrequest level.	

## **Root XML Tag**

<request>

## JSON example

```
{
   "method" : "GET",
   "url" : "v34.0/sobjects/account/001D000000K0fXOIAZ?fields=Name,BillingPostalCode"
}
```

SEE ALSO:

Batch

Update a Record and Get Its Field Values in a Single Request

# Batch Response Body

Describes the result of a Batch request.

## **Batch Results**

## **Properties**

Name	Туре	Description
hasErrors	Boolean	true if at least one of the results in the result set is an HTTP status code in the 400 or 500 range; false otherwise.
results	Subrequest Result[]	Collection of subrequest results.

## JSON example

```
{
  "hasErrors" : false,
  "results" : [{
     "statusCode" : 204,
     "result" : null
     },{
     "statusCode" : 200,
     "result": {
         "attributes" : {
            "type" : "Account",
            "url" : "/services/data/v34.0/sobjects/Account/001D000000K0fXOIAZ"
         },
         "Name" : "NewName",
        "BillingPostalCode" : "94105",
         "Id" : "001D000000K0fX0IAZ"
  } ]
```

## Subrequest Result

## **Properties**

Name	Туре	Description
result	The type depends on the response type of the subrequest.	The response body of this subrequest.
	(1) Important: If the result is an error, the type is a collection containing the error message and error code.	
statusCode	Integer	An HTTP status code indicating the status of this subrequest.

## JSON example

```
"attributes" : {
    "type" : "Account",
    "url" : "/services/data/v34.0/sobjects/Account/001D0000000K0fXOIAZ"
},
    "Name" : "NewName",
    "BillingPostalCode" : "94015",
    "Id" : "001D000000K0fXOIAZ"
}
```

SEE ALSO:

Batch

Update a Record and Get Its Field Values in a Single Request

# **SObject Tree**

Creates one or more sObject trees with root records of the specified type. An sObject tree is a collection of nested, parent-child records with a single root record.

In the request data, you supply the record hierarchies, required and optional field values, each record's type, and a reference ID for each record. Upon success, the response contains the IDs of the created records. If an error occurs while creating a record, the entire request fails. In this case, the response contains only the reference ID of the record that caused the error and the error information.

The request can contain the following:

- Up to a total of 200 records across all trees
- Up to five records of different types
- SObject trees up to five levels deep

Because an sObject tree can contain a single record, you can use this resource to create up to 200 unrelated records of the same type.

When the request is processed and records are created, triggers, processes, and workflow rules fire separately for each of the following groups of records.

- Root records across all sObject trees in the request
- All second-level records of the same type—for example, second-level Contacts across all sObject trees in the request
- All third-level records of the same type
- All fourth-level records of the same type
- All fifth-level records of the same type

#### URI

/vXX.X/composite/tree/SObjectName

#### **Formats**

JSON, XML

### **HTTP** method

POST

#### **Authentication**

Authorization: Bearer token

#### **Parameters**

None required

### Request body

SObject Tree Request Body on page 259

## **Response body**

SObject Tree Response Body on page 262

### **Examples**

- For an example of creating unrelated records of the same type, see Create Multiple Records on page 102.
- For an example of creating nested records, see Create Nested Records on page 100.

## **SObject Tree Request Body**

Describes a collection of sObject trees to create with the SObject Tree resource.

## **SObject Tree Collection Input**

The request body contains a records collection that includes sObject trees.

## **Properties**

Name	Туре	Description	Required or Optional
records	SObject Tree Input[]	Collection of record trees to create. Each tree's root record type must correspond to the sObject specified in the SObject Tree URI.	•

## **Root XML Tag**

<SObjectTreeRequest>

## JSON example

```
"records" :[{
   "attributes" : {"type" : "Account", "referenceId" : "ref1"},
   "name" : "SampleAccount",
   "phone": "1234567890",
    "website" : "www.salesforce.com",
    "numberOfEmployees" : "100",
    "industry" : "Banking",
    "Contacts" : {
      "records" : [{
         "attributes" : {"type" : "Contact", "referenceId" : "ref2"},
         "lastname" : "Smith",
         "title" : "President",
         "email" : "sample@salesforce.com"
         "attributes" : {"type" : "Contact", "referenceId" : "ref3"},
         "lastname" : "Evans",
         "title" : "Vice President",
```

### XML example

```
<SObjectTreeRequest>
    <records type="Account" referenceId="ref1">
        <name>SampleAccount</name>
        <phone>1234567890</phone>
        <website>www.salesforce.com</website>
        <numberOfEmployees>100</numberOfEmployees>
       <industry>Banking</industry>
        <Contacts>
            <records type="Contact" referenceId="ref2">
               <lastname>Smith
               <title>President</title>
               <email>sample@salesforce.com</email>
            </records>
            <records type="Contact" referenceId="ref3">
               <lastname>Evans
               <title>Vice President</title>
               <email>sample@salesforce.com</email>
            </records>
        </Contacts>
    </records>
    <records type="Account" referenceId="ref4">
        <name>SampleAccount2</name>
        <phone>1234567890</phone>
        <website>www.salesforce2.com</website>
        <numberOfEmployees>100</numberOfEmployees>
        <industry>Banking</industry>
    </records>
</SObjectTreeRequest>
```

## **SObject Tree Input**

An sObject tree is a recursive data structure that contains a root record, its data, and its child records represented as other sObject trees.

## **Properties**

Name	Туре	Description	Required or Optional
attributes	Collection	Attributes for this record. The attributes property contains two subproperties:	Required
		• type (required)—This record's type.	
		<ul> <li>referenceId (required)—Reference ID for this record.</li> <li>Must be unique in the context of the request and start with an alphanumeric character.</li> </ul>	
		In XML, include attributes inside the records element.	
Required object fields	Depends on field	Required fields and field values for this record.	Required
Optional object fields	Depends on field	Optional fields and field values for this record.	Optional
Child relationships	SObject Tree Collection Input	This record's child relationships, such as an account's child contacts. Child relationships are either master-detail or lookup relationships. To view an object's valid child relationships, use the SObject Describe resource or Schema Builder. The value of this property is an SObject Tree Collection Input that contains child sObject trees.	Optional

## **Root XML Tag**

<records>

### JSON example

```
"attributes" : {"type" : "Account", "referenceId" : "ref1"},
"name" : "SampleAccount",
"phone": "1234567890",
"website" : "www.salesforce.com",
"NumberOfEmployees" : "100",
"industry" : "Banking",
"Contacts" : {
  "records" : [{
    "attributes" : {"type" : "Contact", "referenceId" : "ref2"},
    "lastname" : "Smith",
    "title" : "President",
    "email" : "sample@salesforce.com"
     "attributes" : {"type" : "Contact", "referenceId" : "ref3"},
     "lastname" : "Evans",
    "title" : "Vice President",
    "email" : "sample@salesforce.com"
  }
```

## XML example

```
<records type="Account" referenceId="ref1">
   <name>SampleAccount</name>
   <phone>1234567890</phone>
   <website>www.salesforce.com</website>
   <numberOfEmployees>100</numberOfEmployees>
   <industry>Banking</industry>
   <Contacts>
       <records type="Contact" referenceId="ref2">
           <lastname>Smith
           <title>President</title>
           <email>sample@salesforce.com</email>
       </records>
       <records type="Contact" referenceId="ref3">
           <lastname>Evans
           <title>Vice President</title>
           <email>sample@salesforce.com</email>
       </records>
   </Contacts>
</records>
```

SEE ALSO:

**SObject Tree** 

Create Multiple Records

**Create Nested Records** 

# **SObject Tree Response Body**

Describes the result of an SObject Tree request.

## **Properties**

Name	Туре	Description
hasErrors	Boolean	true if an error occurred while creating a record; false otherwise.
results	Collection	Upon success, results contains the reference ID of each requested record and its new record ID. Upon failure, it contains only the reference ID of the record that caused the error, error status code, error message, and fields related to the error. In the case of duplicate reference IDs, results contains one item for each instance of the duplicate ID.

### JSON example upon success

```
"hasErrors" : false,
"results" : [{
    "referenceId" : "ref1",
    "id" : "001D000000K0fXOIAZ"
    },{
```

```
"referenceId" : "ref4",

"id" : "001D000000K0fXPIAZ"
},{
    "referenceId" : "ref2",
    "id" : "003D000000QV9n2IAD"
},{
    "referenceId" : "ref3",
    "id" : "003D00000QV9n3IAD"
}]
}
```

#### XML example upon success

```
<?xml version="1.0" encoding="UTF-8"?>
<SObjectTreeResponse>
    <hasErrors>false</hasErrors>
    <results>
        <id>001D000000K0fXOIAZ</id>
        <referenceId>ref1</referenceId>
    </results>
    <results>
        <id>001D000000K0fXPIAZ</id>
        <referenceId>ref4</referenceId>
    </results>
    <results>
        <id>003D000000QV9n2IAD</id>
        <referenceId>ref2</referenceId>
    </results>
    <results>
        <id>003D000000QV9n3IAD</id>
        <referenceId>ref3</referenceId>
    </results>
</SObjectTreeResponse>
```

## JSON example upon failure

```
"hasErrors" : true,
"results" : [{
    "referenceId" : "ref2",
    "errors" : [{
        "statusCode" : "INVALID_EMAIL_ADDRESS",
        "message" : "Email: invalid email address: 123",
        "fields" : [ "Email" ]
        }]
    }]
}
```

## XML example upon failure

## SEE ALSO:

**SObject Tree** 

Create Multiple Records

Create Nested Records

# **SObject Collections**

Executes actions on multiple records in one request. Use SObject Collections to reduce the number of round-trips between the client and server. This resource is available in API version 42.0 and later.

#### URI

The URI to use depends on the operation.

#### Create

POST /vXX.X/composite/sobjects

#### Retrieve

GET

/vXX.X/composite/sobjects/SobjectName?ids=recordId, recordId&fields=fieldname, fieldname

## Update

PATCH /vXX.X/composite/sobjects

### Delete

DELETE /vXX.X/composite/sobjects?ids=recordId, recordId

### **Formats**

JSON, XML

#### HTTP method

GET, DELETE, PATCH, POST

#### Authentication

```
Authorization: Bearer token
```

The parameters, request body, and response body you use depend on the operation. For details, see the specific operation.

### IN THIS SECTION:

### Create Multiple Records with Fewer Round-Trips

Use a POST request with sObject Collections to add up to 200 records, returning a list of SaveResult objects. You can choose whether to roll back the entire request when an error occurs.

## Retrieve Multiple Records with Fewer Round-Trips

Use a GET or POST request with sObject Collections to retrieve one or more records of the same object type. A list of sObjects that represents the individual records of the specified type is returned. The number of sObjects returned matches the number of IDs passed in the request.

## Update Multiple Records with Fewer Round-Trips

Use a PATCH request with sObject Collections to update up to 200 records, returning a list of SaveResult objects. You can choose whether to roll back the entire request when an error occurs.

### Delete Multiple Records with Fewer Round-Trips

Use a DELETE request with sObject Collections to delete up to 200 records, returning a list of DeleteResult objects. You can choose to roll back the entire request when an error occurs.

## Create Multiple Records with Fewer Round-Trips

Use a POST request with sObject Collections to add up to 200 records, returning a list of SaveResult objects. You can choose whether to roll back the entire request when an error occurs.

## Request Syntax

POST /vXX.X/composite/sobjects

### **Parameters**

Parameter	Description
allOrNone	Optional. Indicates whether to roll back the entire request when the creation of any object fails ( $true$ ) or to continue with the independent creation of other objects in the request. The default is false.
records	Required. A list of sObjects. In a POST request using sObject Collections, set the type attribute for each object, but don't set the id field for any object.

## **Usage Guidelines**

- The list can contain up to 200 objects.
- The list can contain objects of different types, including custom objects.
- Each object must contain an attributes map. The map must contain a value for type.



- Objects are created in the order they're listed. The SaveResult objects are returned in the order in which the create requests were specified.
- If the request body includes objects of more than one type, they are processed as chunks. For example, if the incoming objects are {account1, account2, contact1, account3}, the request is processed in three chunks: {{account1, account2}, {contact1}, {account3}}. A single request can process up to 10 chunks.
- A single request can process up to 10 chunks.
- You can't create records for multiple object types in one call when one of the types is related to a feature in the Salesforce Setup area.

## **Checking for Errors**

If the request isn't well formed, the API returns a 400 Bad Request HTTP Status. Fix the syntax of the request and try again.

• If the request is well formed, the API returns a 200 OK HTTP Status. If an item was processed successfully, the success flag shows for that item. Error information is returned in the errors array.

## Request Body Example

```
POST /composite/sobjects
{
    "allOrNone" : false,
    "records" : [{
        "attributes" : {"type" : "Account"},
        "Name" : "example.com",
        "BillingCity" : "San Francisco"
}, {
        "attributes" : {"type" : "Contact"},
        "LastName" : "Johnson",
        "FirstName" : "Erica"
}]
```

## Response Body Examples

This example shows a response when all the items were processed successfully.

This example shows a response when some items caused errors and allorNone is false.

```
"id" : "003RM0000068xVCYAY",
    "success" : true,
    "errors" : [ ]
}
```

This example shows a response when some items caused errors and allOrNone is true.

```
HTTP/1.1 200 OK
      "success" : false,
      "errors" : [
         {
            "statusCode" : "DUPLICATES DETECTED",
            "message" : "Use one of these records?",
            "fields" : [ ]
      ]
   },
   {
      "success" : false,
      "errors" : [
            "statusCode" : "ALL OR NONE OPERATION ROLLED BACK",
            "message" : "Record rolled back because not all records were valid and the
request was using AllOrNone header",
            "fields" : [ ]
         }
      ]
   }
1
```

# Retrieve Multiple Records with Fewer Round-Trips

Use a GET or POST request with sObject Collections to retrieve one or more records of the same object type. A list of sObjects that represents the individual records of the specified type is returned. The number of sObjects returned matches the number of IDs passed in the request.

You can specify approximately 800 IDs before the URL length causes the HTTP 414 error URI too long. To retrieve more records than the URL length can accommodate, use a POST request to retrieve up to 2,000 records of the same object type. If you use POST, the IDs and fields of the records to retrieve are specified in the request body.

## Request Syntax

If you're using a GET request, use the following syntax, where SObjectName is the object type of the records from which you're retrieving data.

GET

 $/ \texttt{vXX.X/composite/sobjects}/ \textit{SObjectName}? \texttt{ids=} \textit{recordId}, \textit{recordId} \& \texttt{fields=} \textit{fieldname}, \textit{fie$ 

If you're using a POST request, use the following syntax, where SObjectName (required) is the object type of the records from which you're retrieving data.

```
POST /composite/sobjects/SObjectName
{
    "ids" : ["recordId", "recordId"],
    "fields" : ["fieldname", "fieldname"]
}
```

## **Parameters**

Parameter	Description
ids	Required. A list of one or more IDs of the objects to return. All IDs must belong to the same object type.
fields	Required. A list of fields to include in the response. The field names you specify must be valid, and you must have read-level permissions to each field.

## **Usage Guidelines**

- If you specify an invalid field name or a field name that you don't have permission to read, HTTP 400 Bad Request is returned.
- If you don't have access to an object, or if a passed ID is invalid, the array returns null for that object.

## Request Example

If you're using a GET request, use the syntax shown in the following example.

```
GET /composite/sobjects/Account?ids=001xx000003DGb1AAG,001xx000003DGb0AAG,001xx000003DGb9AAG&fields=id,name
```

If you're using a POST request, use a request body as shown in the following example.

```
POST /composite/sobjects/Account
{
    "ids" : ["001xx000003DGb1AAG", "001xx000003DGb0AAG", "001xx000003DGb9AAG"],
    "fields" : ["id", "name"]
}
```

## Response Body Example

```
{
    "attributes" : {
        "type" : "Account",
        "url" : "/services/data/v42.0/sobjects/Account/001xx000003DGb1AAG"
    },
    "Id" : "001xx000003DGb1AAG",
    "Name" : "Acme"
},
    {
        "attributes" : {
```

## Update Multiple Records with Fewer Round-Trips

Use a PATCH request with sObject Collections to update up to 200 records, returning a list of SaveResult objects. You can choose whether to roll back the entire request when an error occurs.

## Request Syntax

PATCH /vXX.X/composite/sobjects

### **Parameters**

Parameter	Description
allOrNone	Optional. Indicates whether to roll back the entire request when the update of any object fails ( $true$ ) or to continue with the independent update of other objects in the request. The default is false.
records	Required. A list of sObjects. In a POST request using sObject Collections, set the type attribute for each object, but don't set the id field for any object.

## **Usage Guidelines**

- The list can contain up to 200 objects.
- The list can contain objects of different types, including custom objects.
- Each object must contain an attributes map. The map must contain a value for type.



**Note**: Using sObject Collections to update blob data requires more values in the attributes map. For more information, see Using SObject Collections to Insert a Collection of Blob Records.

- Each object must contain an id field with a valid ID value.
- Objects are updated in the order they're listed. The SaveResult objects are returned in the order in which the update requests were specified.
- If the request body includes objects of more than one type, they are processed as chunks. For example, if the incoming objects are {account1, account2, contact1, account3}, the request is processed in three chunks: {{account1, account2}, {contact1}, {account3}}. A single request can process up to 10 chunks.
- You can't update records for multiple object types in one call when one of those types is related to a feature in the Salesforce Setup area.

## **Checking for Errors**

- If the request isn't well formed, the API returns a 400 Bad Request HTTP Status. Fix the syntax of the request and try again.
- If the request is well formed, the API returns a 200 OK HTTP Status. If an item was processed successfully, the success flag shows for that item. Error information is returned in the errors array.

## Request Body Example

```
PATCH /composite/sobjects
{
    "allOrNone" : false,
    "records" : [{
        "attributes" : {"type" : "Account"},
        "id" : "001xx000003DGb2AAG",
        "NumberOfEmployees" : 27000
},{
        "attributes" : {"type" : "Contact"},
        "id" : "003xx000004TmiQAAS",
        "Title" : "Lead Engineer"
}]
```

## **Response Body Examples**

This example shows a response when all the items were processed successfully.

This example shows a response when some items caused errors and allOrNone is false.

```
{
    "statusCode": "MALFORMED_ID",
    "message": "Contact ID: id value of incorrect type: 001xx000003DGb2999",
    "fields": [
        "Id"
    ]
}
]
```

This example shows a response when some items caused errors and allorNone is true.

```
HTTP/1.1 200 OK
      "id" : "001RM000003oCprYAE",
      "success" : false,
      "errors" : [
            "statusCode" : "ALL OR NONE OPERATION ROLLED BACK",
            "message" : "Record rolled back because not all records were valid and the
request was using AllOrNone header",
            "fields" : [ ]
      ]
   },
      "success" : false,
      "errors" : [
            "statusCode" : "MALFORMED ID",
            "message": "Contact ID: id value of incorrect type: 001xx000003DGb2999",
            "fields" : [
               "Id"
         }
      ]
   }
```

# Delete Multiple Records with Fewer Round-Trips

Use a DELETE request with sObject Collections to delete up to 200 records, returning a list of DeleteResult objects. You can choose to roll back the entire request when an error occurs.

## Request Syntax

DELETE /vXX.X/composite/sobjects?ids=recordId, recordId

## **Parameters**

Parameter	Description	
allOrNone	Optional. Indicates whether to roll back the entire request when the deletion of any object fails (true) or to continue with the independent deletion of other objects in the request. The default is false.	
ids	Required. A list of up to 200 IDs of objects to be deleted. The IDs can belong to different object types, including custom objects.	

## Usage Guidelines

- The DeleteResult objects are returned in the order in which the IDs of the deleted objects were specified.
- You can't delete records for multiple object types in one call when one of those types is related to a feature in the Salesforce Setup area.

## **Checking for Errors**

- If the request isn't well formed, the API returns a 400 Bad Request HTTP Status. Fix the syntax of the request and try again.
- If the request is well formed, the API returns a 200 OK HTTP Status. If an item was processed successfully, the success flag shows for that item. Error information is returned in the errors array.

## Request Example

DELETE /composite/sobjects?ids=001xx000003DGb2AAG,003xx000004TmiQAAS&allOrNone=false

## Response Body Examples

This example shows a response when all the items were processed successfully.

This example shows a response when some items caused errors and allOrNone is false.

```
HTTP/1.1 200 OK
```

Reference Headers

This example shows a response when some items caused errors and allorNone is true.

```
HTTP/1.1 200 OK
[
      "id" : "001RM000003oLruYAE",
      "success" : false,
      "errors" : [
         {
            "statusCode" : "ALL OR NONE OPERATION ROLLED BACK",
            "message" : "Record rolled back because not all records were valid and the
request was using AllOrNone header",
            "fields" : [ ]
      ]
   },
      "success" : false,
      "errors" : [
         {
            "statusCode" : "MALFORMED ID",
            "message" : "malformed id 001RM000003oLrB000",
            "fields" : [ ]
      ]
   }
```

# Headers

This section lists custom HTTP request and response headers used for REST API.

Reference Assignment Rule Header

### IN THIS SECTION:

## Assignment Rule Header

The Assignment Rule header is a request header applied when creating or updating Accounts, Cases, or Leads. If enabled, the active assignment rules are used. If disabled, the active assignment rules are not applied. If a valid AssignmentRule ID is provided, the AssignmentRule is applied. If the header is not provided with a request, REST API defaults to using the active assignment rules.

#### 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.

#### Limit Info Header

This response header is returned in each request to REST API. You can use the information to monitor API limits.

### 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.

#### **Query Options Header**

Specifies options used in a query, such as the query results batch size. Use this request header with the Query resource.

# Assignment Rule Header

The Assignment Rule header is a request header applied when creating or updating Accounts, Cases, or Leads. If enabled, the active assignment rules are used. If disabled, the active assignment rules are not applied. If a valid AssignmentRule ID is provided, the AssignmentRule is applied. If the header is not provided with a request, REST API defaults to using the active assignment rules.



**Note:** This header also gets applied when making REST API calls that indirectly result in creating or updating Accounts, Cases, or Leads. For example, if you use this header with a call that updates a record, and the update executes an Apex trigger that updates a Case, the assignment rules would be applied.

## Header Field Name and Values

#### Field name

Sforce-Auto-Assign

#### Field values

- TRUE. Active assignment rules are applied for created or updated Accounts, Cases, or Leads.
- FALSE. Active assignment rules are not applied for created or updated Accounts, Cases, or Leads.
- Valid AssignmentRule ID. The given AssignmentRule is applied for created Accounts, Cases, or Leads.

TRUE and FALSE are not case-sensitive.

If the header is not provided in the request, the default value is TRUE.

#### **Example**

Sforce-Auto-Assign: FALSE

# 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.

Reference Limit Info 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 gueries 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 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/nnn. The first number is the number of API calls used, and the second number is the API limit for the organization.

### 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
```

Reference Package Version Header

```
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/v47.0/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

# Status Codes and Error Responses

Either when an error occurs or when a response is successful, the response header contains an HTTP code, and the response body usually contains:

- The HTTP response code
- The message accompanying the HTTP response code
- The field or object where the error occurred (if the response returns information about an error)

HTTP response code	Description
200	"OK" success code, for GET or HEAD request.
201	"Created" success code, for POST request.
204	"No Content" success code, for DELETE request.
300	The value returned when an external ID exists in more than one record. The response body contains the list of matching records.
304	The request content has not changed since a specified date and time. The date and time is provided in a If-Modified-Since header. See Get Object Metadata Changes for an example.
400	The request couldn't be understood, usually because the JSON or XML body contains an error.
401	The session ID or OAuth token used has expired or is invalid. The response body contains the message and errorCode.
403	The request has been refused. Verify that the logged-in user has appropriate permissions. If the error code is REQUEST_LIMIT_EXCEEDED, you've exceeded API request limits in your org.
404	The requested resource couldn't be found. Check the URI for errors, and verify that there are no sharing issues.
405	The method specified in the Request-Line isn't allowed for the resource specified in the URI.
415	The entity in the request is in a format that's not supported by the specified method.

HTTP response code	Description
500	An error has occurred within Lightning Platform, so the request couldn't be completed. Contact Salesforce Customer Support.

## Incorrect ID example

Using a non-existent ID in a request using JSON or XML (request\_body.json or request\_body.xml)

```
"fields" : [ "Id" ],
   "message" : "Account ID: id value of incorrect type: 001900K0001pPuOAAU",
   "errorCode" : "MALFORMED_ID"
}
```

## Resource does not exist

Requesting a resource that doesn't exist, for example, if you try to create a record using a misspelled object name

```
"message" : "The requested resource does not exist",
"errorCode" : "NOT_FOUND"
}
```