

Force.com SOAP API Cheatsheet



Overview

Use the Force.com SOAP API to create, retrieve, update, or delete records in Force.com from any external system that supports SOAP-based Web services, such as Java, .NET, or PHP client applications. With more than 20 different calls, the API also allows you to maintain passwords, perform searches, retrieve metadata information about objects and more.

Getting Started

To generate the WSDL representing your environment, click **Setup | Develop | API**. The Enterprise WSDL is a strongly typed representation of your data, while the Partner WSDL is a loosely typed representation, ideal for writing generic clients that work across different organizations.

Summary Functions

These are the fundamental calls needed for logging in, querying, creating, and deleting data:

<code>login()</code>	Logs in and starts a client session.
<code>logout()</code>	Ends the session of the logged-in user.
<code>invalidateSessions()</code>	Ends one or more sessions.
<code>create()</code>	Adds one or more new individual records.
<code>delete()</code>	Deletes one or more individual records.
<code>update()</code>	Updates one or more existing records.
<code>undelete()</code>	Undeletes records from the Recycle Bin.
<code>emptyRecycleBin()</code>	Deletes records from the Recycle Bin immediately.
<code>getDeleted()</code>	Retrieves the IDs of individual deleted records for a specified timespan.
<code>getUpdated()</code>	Retrieves the IDs of individual updated records for a specified timespan.
<code>convertLead()</code>	Converts a Lead into an Account, Contact, or (optionally) an Opportunity.
<code>merge()</code>	Merges records of the same object type.
<code>process()</code>	Submits an array of approval process instances for approval, or processes an array of approval process instances to be approved, rejected, or removed.
<code>query()</code>	Executes a query against the specified object and returns data that matches the specified criteria.

Summary Functions cont.

<code>queryAll()</code>	Same as <code>query()</code> , but includes deleted and archived items.
<code>queryMore()</code>	Retrieves the next batch of records from a query.
<code>retrieve()</code>	Retrieves one or more records based on the specified record IDs.
<code>search()</code>	Executes a text search in your organization's data.

Utility Calls

These calls let you retrieve user information, send emails, and reset passwords:

<code>getServerTimestamp()</code>	Retrieves the current system timestamp from the API.
<code>getUserInfo()</code>	Retrieves personal information for the user associated with the current session.
<code>resetPassword()</code>	Changes a user's password to a system-generated value.
<code>sendEmail()</code>	Immediately sends an email message.
<code>setPassword()</code>	Sets the specified user's password to the specified value.

Primitive Data Types

The API uses the following primitive data types:

<code>base64</code>	Base64-encoded binary data.
<code>boolean</code>	Boolean fields have one of these values: true (or 1), or false (or 0).
<code>byte</code>	A set of bits.
<code>date</code>	Date data. Unlike <code>dateTime</code> fields, date fields contain no time value.
<code>dateTime</code>	Date/time values (timestamps).
<code>double</code>	It's a 64 bit decimal.
<code>int</code>	Fields of this type contain numbers with no fractional portion.
<code>string</code>	Character strings.
<code>time</code>	Time values.

Describe Calls

These calls let you dynamically retrieve information about objects and other aspects of an organization:

<code>describeAllTabs()</code>	Returns information about all the tabs—including Lightning Page tabs—available to the logged-in user, regardless of whether the user has chosen to hide tabs.
<code>describeAppMenu()</code>	Retrieves metadata about items either in the Salesforce1 navigation menu or Salesforce drop-down app menu.
<code>describeApprovalLayout()</code>	Retrieves metadata about approval layouts for the specified object type.
<code>describeAvailableQuickActions()</code>	Describes details about actions available for a specified context.
<code>describeCompactLayouts()</code>	Retrieves metadata about compact layouts for the specified object type.
<code>describeDataCategoryGroups()</code>	Retrieves available category groups for entities specified in the request.
<code>describeDataCategoryGroupStructures()</code>	Retrieves available category groups along with their data category structure for entities specified in the request.
<code>describeFlexiPages()</code>	Retrieves metadata details about a set of Lightning Pages. A Lightning Page is the home page for a mobile app that appears as a menu item in the Salesforce1 navigation menu.
<code>describeGlobal()</code>	Retrieves a list of available objects for your organization's data.
<code>describeGlobalTheme()</code>	Returns information about both objects and themes available to the current logged-in user.
<code>describeLayout()</code>	Retrieves metadata about page layouts for the specified object type.
<code>describePrimaryCompactLayouts()</code>	Retrieves metadata about the primary compact layout for each of the specified object types.
<code>describeQuickActions()</code>	Retrieves details about specified actions.
<code>describeSearchScopeOrder()</code>	Retrieves an ordered list of objects in the logged-in user's default global search scope, including any pinned objects in the user's search results page.
<code>describeSObjects()</code>	Retrieves metadata (field list and object properties) for the specified object types.
<code>describeSoftphoneLayout()</code>	Describes the SoftPhone layout(s) created for an organization.
<code>describeTabs()</code>	Describes the apps and tabs that have been configured for the user.
<code>describeTheme()</code>	Returns information about themes available to the current logged-in user.

Field Types

In addition to the primitive data types, the API defines these data types for object fields:

<code>anyType</code>	Polymorphic data type that represents a particular type depending on the kind of field involved.
<code>calculated</code>	Fields that are defined by a formula.
<code>combobox</code>	A combobox, which includes a set of enumerated values and allows the user to specify a value not in the list.
<code>currency</code>	Currency values.
<code>email</code>	Email addresses.
<code>encryptedstring</code>	Encrypted text fields contain any combination of letters, numbers, or symbols that are stored in encrypted form.
<code>ID</code>	Primary key field for the object.
<code>masterrecord</code>	When records are merged, the ID of the record that is saved (the other records are deleted).
<code>multipicklist</code>	Multi-select picklists, which include a set of enumerated values from which multiple values can be selected.
<code>percent</code>	Percentage values.
<code>phone</code>	Phone numbers. Values can include alphabetic characters.
<code>picklist</code>	Picklists, which include a set of enumerated values from which one value can be selected.
<code>reference</code>	Cross-references to a different object. Analogous to a foreign key field in SQL.
<code>textarea</code>	String that is displayed as a multiline text field.
<code>url</code>	URL values.

API Request Limits

The following table lists the total API requests (calls) per 24-hour period for an organization.

See table here: <http://bit.ly/TotalAPILimits>

Limits are enforced against the aggregate of all API calls made by the organization in a 24 hour period; limits are not on a per-user basis. When an organization exceeds a limit, all users in the organization may be temporarily blocked from making additional calls. Calls will be blocked until usage for the preceding 24 hours drops below the limit.

API Fault Element

The following table lists the API fault elements that the API returns if an error occurs when processing a service request:

ApiQueryFault	The row and column numbers where the problem occurred.
LoginFault	An error occurred during the login() call.
InvalidSObjectFault	An invalid sObject in a describeSObject(), describeSObjects(), create(), update(), retrieve(), or query() call.
InvalidFieldFault	An invalid field in a retrieve() or query() call.
MalformedQueryFault	A problem in the queryString passed in a query() call.
InvalidQueryLocatorFault	A problem in the queryLocator passed in a queryMore() call.
MalformedSearchFault	A problem in the search passed in a search() call.
InvalidIdFault	A specified ID was invalid in a setPassword() or resetPassword() call.
UnexpectedErrorFault	An unexpected error occurred. The error is not associated with any other API fault.

Compound Fields

Compound fields group together multiple elements of primitive data types, such as numbers or strings, to represent complex data types, such as a location or an address. Compound fields are an abstraction that can simplify application code that handles the values, leading to more concise, understandable code.

Compound fields are accessible as a single, structured field, or as individual component fields. The values contained within the compound field and the values in individual fields both map to the same underlying data stored in Salesforce; they always have identical values. Code that references individual component fields is unaffected by the new compound fields.

Address Compound Fields

Using API 30.0 and later, standard addresses are available as a compound field of type Address, a structured data type that combines the following fields.

City
Country
CountryCode
Latitude
Longitude
PostalCode
State
StateCode
Street

Geolocation Compound Field

In API versions 26.0 and later, geolocation fields are available as a compound field of type Location. This structured data type contains the following fields.

Latitude
Longitude

Getting Started

The following fields are read-only fields found on most objects. These fields are automatically updated during API operations. For example, the ID field is automatically generated during a create operation and the LastModifiedDate is automatically updated during any operation on an object.

See table here: <http://bit.ly/SystemFieldsTable>

If you import data into Salesforce, you can set the values for audit fields on the following objects: Account, Article, ArticleVersion, Attachment, CampaignMember, Case, CaseComment, Contact, ContentVersion, Contract, EmailMessage, Event, FeedComment, FeedItem, Holiday, Idea, IdeaComment, Lead, Opportunity, Question, Task, Vote, and custom objects.

1. From Setup, enter User Interface in the Quick Find box, then select User Interface under Customize.
2. Under Setup, select Enable "Set Audit Fields upon Record Creation" and "Update Records with Inactive Owners" User Permissions.
3. In the permission set or profile you want to set audit fields with, enable the permission, Set Audit Fields upon Record Creation.
4. Using the API, create a record and set its audit fields.

Not all standard objects have all audit fields. Check the Enterprise WSDL to verify which audit fields are available for a given object.