

Salesforce Additional Information

Review the Additional Information for using Salesforce Connector.

On this page:

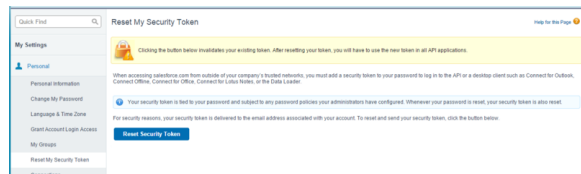
- [Salesforce Security Token](#)
- [Salesforce WSDL](#)
- [Salesforce Chatter App](#)
- [Salesforce Consumer Key and Consumer Secret](#)
- [Salesforce sQueries File](#)
 - [Getting Knowledge Article Specific Info](#)
- [Salesforce Connector Comparison](#)
 - [Object Construction](#)
 - [SOQL Queries](#)
 - [Credentials](#)
 - [StoreTypes](#)
 - [Crawl Scope](#)

Salesforce Security Token

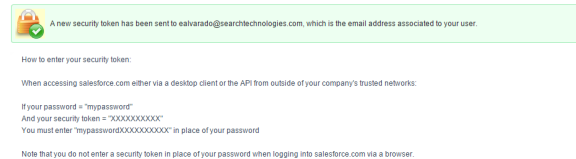
You are given a security token the first time you log into Salesforce. Don't know your security token?

To reset it:

1. Click on your username in the top right section
2. Choose "My Settings" from the menu
3. Click "Personal"
4. Click "Reset your security token" and follow the steps provided



Reset My Security Token
Reset Security Token



This steps will send to your user email the info with the "Security Token" like:



support@salesforce.com
salesforce.com security token confirmation

To: [redacted]
This message was sent with High importance.

Your Security Token Is Enclosed.

Dear [redacted],

When accessing salesforce.com from outside of your company's trusted networks, you must add a security token to your password to log in to a desktop client, such as Connect for Outlook, Connect Offline, Connect for Office, Connect for Lotus Notes, or the Data Loader.

New security tokens are automatically sent to you when your salesforce.com password is changed or when you request to reset your security token.

Your new security token is below. Note that security tokens are case sensitive.

User Name:

Security Token:

Please add your security token to your password. Note that you do not enter a security token in place of your password when logging into salesforce.com via a browser.

For Outlook Edition and Connect for Outlook:

1. Go to Outlook > Tools > Salesforce Options
2. Enter the password and security token in the password window. For example, if your password is "myPassword" and your security token is "XXXXXXX", you will enter "myPasswordXXXXXXX" into the password window.
3. Click verify. This tests that you have correctly entered your password and token.
4. If verification succeeds, Click 'OK'

For Offline Edition, Connect Offline, Office Edition, and other salesforce.com applications, please visit the Setup > My Personal Information > Reset Security Token page.

Thank you,
salesforce.com

Salesforce WSDL

The follow steps will provide you the information to get the Enterprise WSDL and generate the necessary jar file for the connector

To get the Enterprise WSDL file, please follow the next steps:

1. Click on Setup in the top right section.
2. Click on Build/Develop in the left menu.
3. Click "API".
4. Download the Enterprise WSDL in "Generate Enterprise WSDL".

API WSDL

Salesforce's WSDL allows you to easily integrate salesforce.com with your applications, and to build new applications that work with salesforce. to a place accessible to your development environment. For complete documentation, sample code, and developer community, visit <http://developer.salesforce.com>

WSDL and Client Certificates

Enterprise WSDL
A strongly typed WSDL for customers who want to build an integration with their salesforce.com organization only.
[Generate Enterprise WSDL](#)

Partner WSDL
A loosely typed WSDL for customers, partners, and ISVs who are building client applications for multiple organizations. It can be used to access data with
[Generate Partner WSDL](#)


Apex WSDL
Click on the link below to download an Apex programming WSDL

i For complete documentation, sample code, and developer community, visit [Salesforce Developer](#).
For more specific documentation about WSDL on Salesforce, please visit [Salesforce Web Services](#).

To generate the Enterprise Jar file, please follow the next steps:

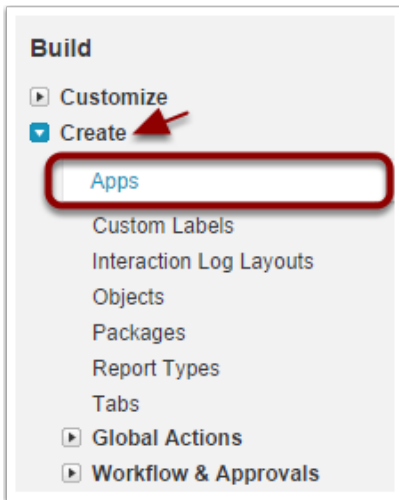
1. Generate and download WSDLs.

2. Download the last version of force-wsc JAR. (link [here](#))
3. Download the last version of stringtemplate JAR. (link [here](#))
4. Download the last version of antlr-runtime JAR. (link [here](#))
5. Generate Jar file using this instruction:

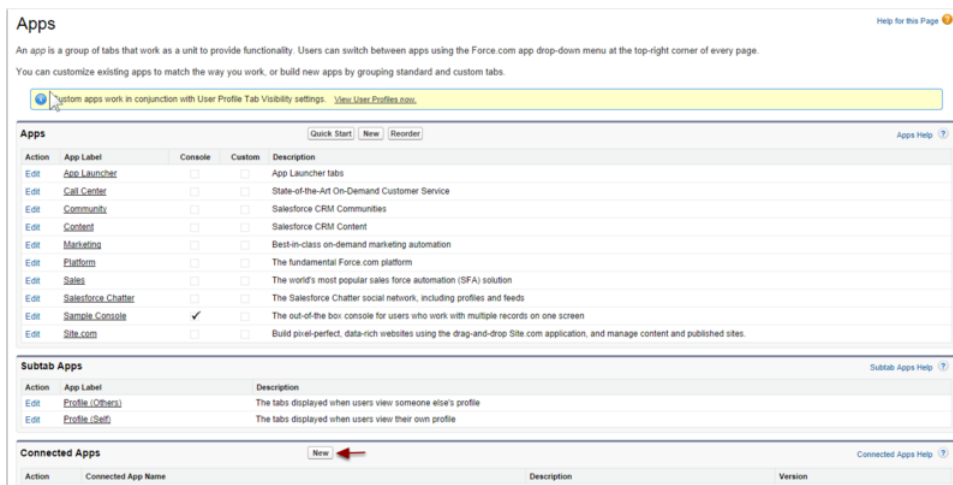
 `java -classpath force-wsc-XXX.jar;stringtemplate-XXX.jar;antlr-runtime-XXX.jar com.sforce.ws.tools.wsdlc enterprise.wSDL enterprise.jar`

Salesforce Chatter App

1. In your Sales Force Server, go to the Setup page
2. Go to Build / Create / Apps



3. In this new page, click on New Connected Apps



4. Fill the Basic Information:
 - a. Connected App Name
 - b. API Name
 - c. Contact Email

5. In the API (Enable OAuth Settings):
 - a. Check "Enable OAuth Settings"
 - b. Set the "Callback Url"
 - c. Select the "Selected OAuth Scopes"

6. Save the application
7. In the next screen you can copy the "Consumer Key" and the "Consumer Secret" information.

Salesforce Consumer Key and Consumer Secret

Once you created your "Connected App" you can get the "Consumer Key" if you go to Setup / Build / Create / Apps and select your connected app.

- Copy the "Consumer Key"
- To get the "Consumer Secret" click in "Click to reveal"

Salesforce sQueries File

This [document](#) represents the data retrieved from Salesforce using SOQL, please modify this document only if you know the available fields and types of the sObject you want to modified, and you know how to use SOQL.

Each sObject tag represents a element type inside Salesforce, following the structure of this sObjects will be explain (please use the API Name for every field, you can check it in the setup/customize/<item>/fields section in Salesforce).

Attributes

For each sObject it can be up to three attributes:

- * @type = Name of the a specific element type inside Salesforce (Not Optional)
- * @attachment = Boolean flag which indicates if this element type can have attachments. If not specified the default is false (Optional)
- * @nested = Boolean flag which indicates if this sObject has nested subfields for the specified references. If not specified the default is false (Optional)

Children

Each sObject can have up to two children:

- * fields = These are the fields that will be retrieved by the connector
- * nested = Indicates the fields to retrieve for a reference (all the fields that ends with "Id" e.g ParentId)

Nested

To uses nested subfields you put subFields(<{name-of-the-reference}>) at the end of the fields inside fields tag

Inside the nested you put a tag with the name of the reference () and the reference's fields in this way:

```
<name-of-the-reference>{name-of-the-reference}.Field1, {name-of-the-reference}.Field2, ...</name-of-the-reference>
```

In the case the reference can be map to many types you can do:

```
<name-of-the-reference>
  TYPEOF {name-of-the-reference}
    WHEN {Type1} THEN Field1, Field2, ...
    WHEN {Type2} THEN FieldA
    WHEN {Type3} THEN FieldX1
  ELSE FieldDefault
END
</name-of-the-reference>
```



- The sQueries.xml is a template document from which the user can decide the fields to crawl, each SQueries.xml can be different for each Salesforce and is responsibility of the developer in charge of the installation of the connector to update or modify this document.
- sQuieres.xml can use the SOQL TYPEOF, but this SOQL needs to be activated in the Salesforce environment.

Getting Knowledge Article Specific Info

You can crawl the KA (knowledge article) info in two ways:

1. Generic: This one will crawl only the standard fields of the KA.
2. Specific: This one will crawl all the custom fields for KA (you need to edit the sQueries file and add one for the specific KA type and fields).

Example: Do you want to add a new type for Offer KA type with custom fields myMail and lastAddress, use as base the query for KnowledgeArticleVersion:

```
<sObject type="KnowledgeArticleVersion" nested="true">
  <fields>Id, KnowledgeArticleId, OwnerId, IsDeleted, PublishStatus, VersionNumber,
  IsLatestVersion, ... , SourceId, ArticleType</fields>
</sObject>

<sObject type="Offer__kav" nested="true">
  <fields>Id, KnowledgeArticleId, OwnerId, IsDeleted, PublishStatus, VersionNumber,
  IsLatestVersion, ... , SourceId, ArticleType, myMail__c, lastAddress__c</fields>
</sObject>
```

Salesforce Connector Comparison

Detailed information comparing the Aspire Salesforce Connector against the Google Salesforce Connector.



A couple of important things to emphasize about the Google connector:

- This connector is not supported by Google and the latest documentation update was in August 2010.
- This connector only support one query to retrieve objects and if you want to include another object, you need to install another connector. On the other hand, the Aspire Salesforce connector uses multiple queries to retrieve many objects.
- Both connectors fetch data directly from Salesforce via the SOAP API (documents/objects) and transforms it into content feeds.

The main differences between the Google Salesforce connector and the Aspire Salesforce connector are:

Object Construction

- The Google Salesforce connector uses the Salesforce WebService APIs to construct object summary documents. Data is retrieved from Salesforce using its proprietary Query language over the WebService APIs.
- The Aspire Salesforce connector uses the Enterprise WSDL of the organization together with other files to generate the enterprise-api.jar

SOQL Queries

- When you configure a Google Salesforce connector, you provide a SOQL query describing the main object and its related objects which will retrieve the data. The SOQL query must already have all the Parent-Child relationships defined in Salesforce such that all the data required is available in the Query response stream. The connector does not support nested queries so all the objects and related data must be available through one SQL statement.
- When you configure an Aspire Salesforce connector, you provide a SOQL queries file with one query per object type in the Salesforce server and it supports nested queries.

Credentials

- The Google Salesforce connector uses the username/password to acquire data from Salesforce, must have at least read access to all the fields requested in the select query to get the data.
- The Aspire Salesforce connector also uses the username/password, but also uses a Security token (optional) to acquire data from Salesforce. User must have at least read access to all the fields requested in the queries in the SQueries file. Also, if you want to crawl the Chatter Feeds, you need to provide a Consumer Key of an App in Salesforce with access to the chatter feed.

StoreTypes

- The Google Salesforce connector supports Store Types.
- The Aspire Salesforce connector does not support Store Types functionality.



The SOAP response XML documents can be stored in several ways. StoreTypes are configurable per connector and are used to optionally save the documents for later use by another GSA or for failure recovery. These are the available StoreTypes:

- **MemoryStore:** SOAP response documents are stored in memory until its read by the connector-manager thread. After its read, the entry is automatically removed from memory.
- **FileStore:** The SOAP response is saved in the filesystem under /WEB-INF/connectors/salesforce-connector/connectorname/filestore folder.
- **JDBCStore:** SOAP responses will get compressed and stored in a database. Each connector instance will use the same database but will read/write from its own indexed table.

Crawl Scope

- As mentioned above, the Google Salesforce Connector only allows crawling one object type at the time.
- The Aspire Salesforce connector can crawl many object types at the same time.



For more info about the Aspire Salesforce connector you can visit: [Aspire Salesforce Connector](#).

For more info about the Google Salesforce connector you can visit: [Google Salesforce](#).

