

# Using the Force.com SOAP API to Integrate with Enterprise Apps

---



**Richard Seroter**

VICE PRESIDENT OF PRODUCT, CENTURYLINK

@rseroter [www.seroter.com](http://www.seroter.com)



# Overview



Anatomy of a SOAP API call

Authenticating users

SOAP WSDL options

Explaining SOAP objects, actions

Handling exceptions

Working with SOAP headers

Building custom SOAP services

API limits

Monitoring usage

Summary



# SOAP API

SOAP protocol access to Salesforce data and functionality.



```
POST /services/Soap/c/35.0 HTTP/1.1
Host: na11.salesforce.com
Content-Type: text/xml
SoapAction: "
```

Interact over HTTP POST

XML payloads

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
xmlns:soapenv=http://schemas.xmlsoap.org/soap/envelope
/ xmlns:urn="urn:enterprise.soap.sforce.com">
  <soapenv:Header>
    <urn:SessionHeader>
      <urn:sessionId>...</urn:sessionId>
    </urn:SessionHeader>
  </soapenv:Header>
  <soapenv:Body>
    <urn:query>
      <urn:queryString>SELECT Id, Name FROM
Account</urn:queryString>
    </urn:query>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP wrapper

Secured with session header

Synchronous operations



# When Do You Use It?

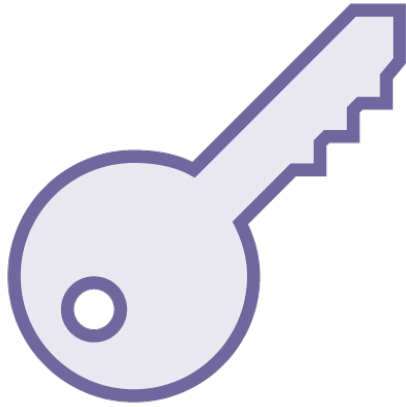


**Integrating with enterprise apps**

**Want generated objects**

**Require broad functional coverage**

# Authenticating SOAP API Users



**Acquire session ID**



**User, object, field  
security applies**



**Source IP rules  
available**



```
<?xml version="1.0" encoding="utf-8" ?>
<env:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
  <env:Body>
    <n1:login xmlns:n1="urn:enterprise.soap.sforce.com">
      <n1:username>USER</n1:username>
      <n1:password>PASSWORD + TOKEN</n1:password>
    </n1:login>
  </env:Body>
</env:Envelope>
```

← Issue "login" request

← Provide password and token

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns="urn:enterprise.soap.sforce.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <loginResponse>
      <result>
        <metadataServerUrl>...</metadataServerUrl>
        <passwordExpired>>false</passwordExpired>
        <sandbox>>false</sandbox>
        <serverUrl>https://na11.salesforce.com/...</serverUrl>
        <sessionId>00DA000000...</sessionId>
        <userId>005A00000...</userId>
        ...
      </result>
    </loginResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

← Get back target URL  
← Short-lived session ID



# Two SOAP API WSDL Options

## Enterprise WSDL

Strong typed

Easy to use

Standard, custom objects

Best for single org solutions

## Partner WSDL

Weak typed

Flexible to use

describeSObjects() for details

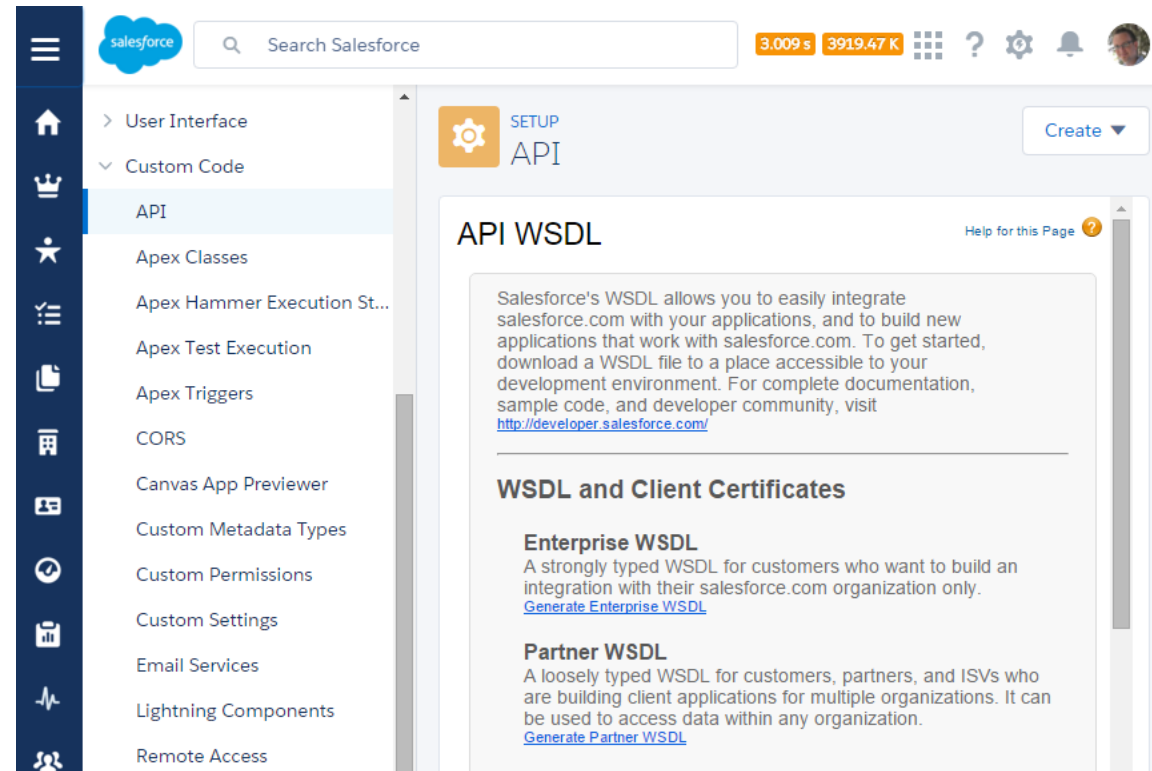
Best for apps targeting multiple orgs





Access via **API** menu

Dynamically  
generated



# Demo



Log into Force.com account

Navigate to WSDL view

Save Enterprise WSDL

Inspect Enterprise WSDL



# What Are SOAP Objects? – About

**Objects represent  
database tables**

**Records are like  
database rows**

**Multiple types of  
objects**



# What Are SOAP Objects? – Object Types

**Standard objects**

**Custom objects**

**External objects**



# What Are SOAP Objects? – Data Types

**Primitive types**

**Field types**

**Compound fields**



# What Are SOAP Objects? – Field Details

**System fields**

**Required fields**

**Field properties**



# What Are SOAP Objects? – Relationships

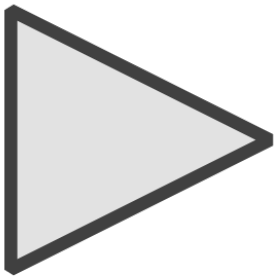
Master-detail (1:n)

Many-to-many

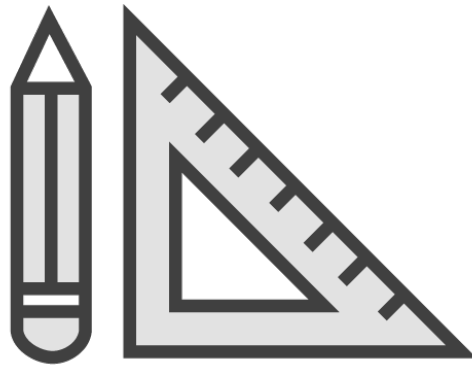
Lookup (1:n)



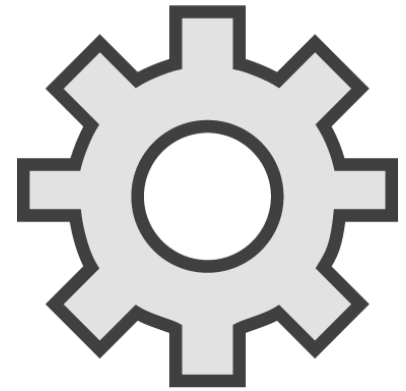
# Types of SOAP API Calls



Core calls



Describe calls



Utility calls



# Demo



Create SOAP request in Postman

Generate session ID

Query “Voter” records

Retrieve the Force.com timestamp



# Handling Faults



**API Fault Element with Exception Code**  
**Error with Status Code**  
**Resolution Handled Client Side**

**Optional**  
**Part of both WSDLs**  
**In request or response**

SOAP Headers



# Demo



Observe SessionHeader in API calls

API call with MruHeader

Confirm added to “recent” list



```
POST /services/Soap/class/SoapVoterService
HTTP/1.1
Host: na11.salesforce.com
Content-Type: text/xml
SoapAction: "
```

← Has “class” in URL

-----

```
global class SoapVoterService {
  webservice static Voter__c getVoter(
    String voterId) {
    Voter__c voter = [
      SELECT ID,
      Name
      FROM Voter__c
      WHERE ID = :voterId];
    return voter;
  }
}
```

← Global class as SOAP service  
← webservice keyword  
← Accepts variety of types

← Run in system context

Can generate a WSDL



# Demo



Create Apex class

Add Apex code to retrieve Voter ID

Decorate with webservice keyword

Call service from Postman



# API Limits

10

Concurrent queries per user

25

Concurrent calls in production org

1M

Daily API calls in production org

200

Max record count in create, update requests



# Monitoring API Usage



View requests for last  
24 hours



API usage reports



API usage  
notifications





# Demo



Viewing 24 hours of API calls

Running SOAP API call report

Creating usage notifications



# Summary



Overview

Anatomy of a SOAP API call

Authenticating users

SOAP WSDL options

Explaining SOAP objects, actions

Handling exceptions

Working with SOAP headers

Building custom SOAP services

API limits

Monitoring usage

