# Integrating with the Salesforce Force.com Platform

## Introduction

One application couldn't cater to all the needs of an organization. Most enterprise-level applications don't work in a siloed manner and they tend to integrate with other sets of applications. They tend to connect with different layers like Data, Presentation, Security, etc based on the needs of the customer. This integrated system would provide an organization with an effective way to maintain customer relationships, increase overall operational efficiency.

Salesforce is a powerful CRM solution and it can work even better when connected with applications that your company already uses. In this article, we summarize a few ways of integrating Force.com utilizing web services. There are several ways to integrate, but depending on the situation at hand we can determine the best way to integrate. For instance, if there is a simple data exchange between a system and Salesforce then SOAP API would suffice. On the other hand, if there is an event to be completed after data flows from one system to another, then Apex Web services, Apex callouts would be the suitable integration option.

## Ways to integrate with Salesforce Force.com

The need for integration is quite common when it comes to Salesforce implementation. There are usually three different layers in any enterprise-level application, commonly called the 3 Tier architecture:

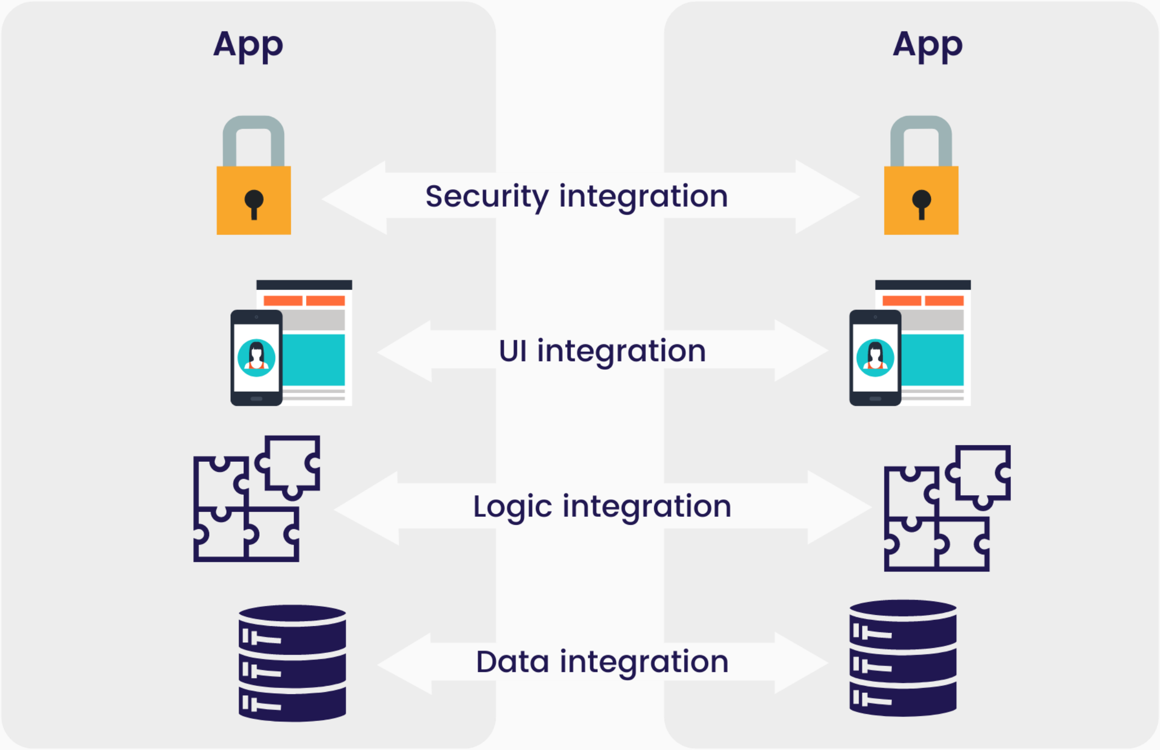
- ✓ Database Tier
- ✓ Application Tier
- ✓ Presentation Tier

Based on the above layers, there are three different integration features:

01 User Interface Integration

02 Business Logic Integration

03 Data Integration



**Data Integration:** It uses SOAP APIs and REST APIs to sync data between Salesforce and external systems. The advantage in Salesforce is there is no coding requirement for syncing data from primary data sources.

**Business Logic Integration:** In this type of Integration for inbound scenarios, Apex Web Services is used and for outbound scenarios, Apex callouts are used. Complex business logic can be in place for bigger organizations. The business logic can be distributed across multiple applications and it can be integrated into one.

**User Interface Integration:** It is a type of integration in which the external applications are redesigned and displayed within Salesforce. It makes a user's life easier by providing a single entry point into multiple applications.

When it comes to integrating external client applications to Salesforce, security is a major concern for many organizations. Proper authentication is required and this is very similar to the authentication or login process in Salesforce UI. If correct credentials are used, then the authentication to Salesforce is valid and Salesforce provides the client application with session ID and server URL.

# Deep Dive into Apex Web Services, Callouts

Apex Web Services: Apex is Salesforce's proprietary language. You could write and expose business logic as a REST or SOAP web service to external applications. While exposing you could add custom logic too. The external applications will call the exposed service and integrate with the application.

To expose a class as a web service, do the following steps:

01 Create a global class that can be accessed by all Apex scripts.

02 Create an inner apex class. This class would contain a request message to the external application. Here is where the variables that need to be passed to other applications are declared.

03 Create a method in Apex. This method will be exposed as a web service. You can add any business logic here.

04 At last, you could generate a WSDL from **Setup -> App Setup -> Develop -> Apex Classes** and it can be shared with external application

Expose a class as REST Service example:

```
@RestResource(urlMapping='/Account/*')global with sharing class MyRestResource{
@HttpGet
global static Account getRecord() {// Add your code
}
}
```

Expose a class as SOAP Service example:

```
global with sharing class MySOAPWebService {
   webservice static Account getRecord(String id) {
      // Add your code
    }
}
```

**Apex Callouts:** They enable you to call an external web service (or) send an HTTP request via Apex code and receive a response. With callouts, you could integrate external web services to Salesforce.

There are two ways to generate Apex callouts. They are alike in the way of sending requests to a web service and receiving a response. However, WSDL applies only for SOAP web services and HTTP callouts work with both SOAP, REST web services.

**01** **Import Web Service Definition Language(WSDL) into Apex:** You can get a WSDL from the external client upload and import it straightaway to Salesforce. This WSDL document is necessary for code generation. The Apex class makes your life easier. It constructs the SOAP XML during the invocation of web service, data transfer, and response parsing.

**02** **Using HTTP Service class:** In this method, you could interact with an external application using a REST-based service. APEX classes like HTTPRequest, HTTPResponse handles the request and response accordingly between two systems.

HTTP service is a preferred method as it requires lesser code, simple to interact with, and has easily readable JSON. SOAP web services are usually used for integrating with legacy applications.

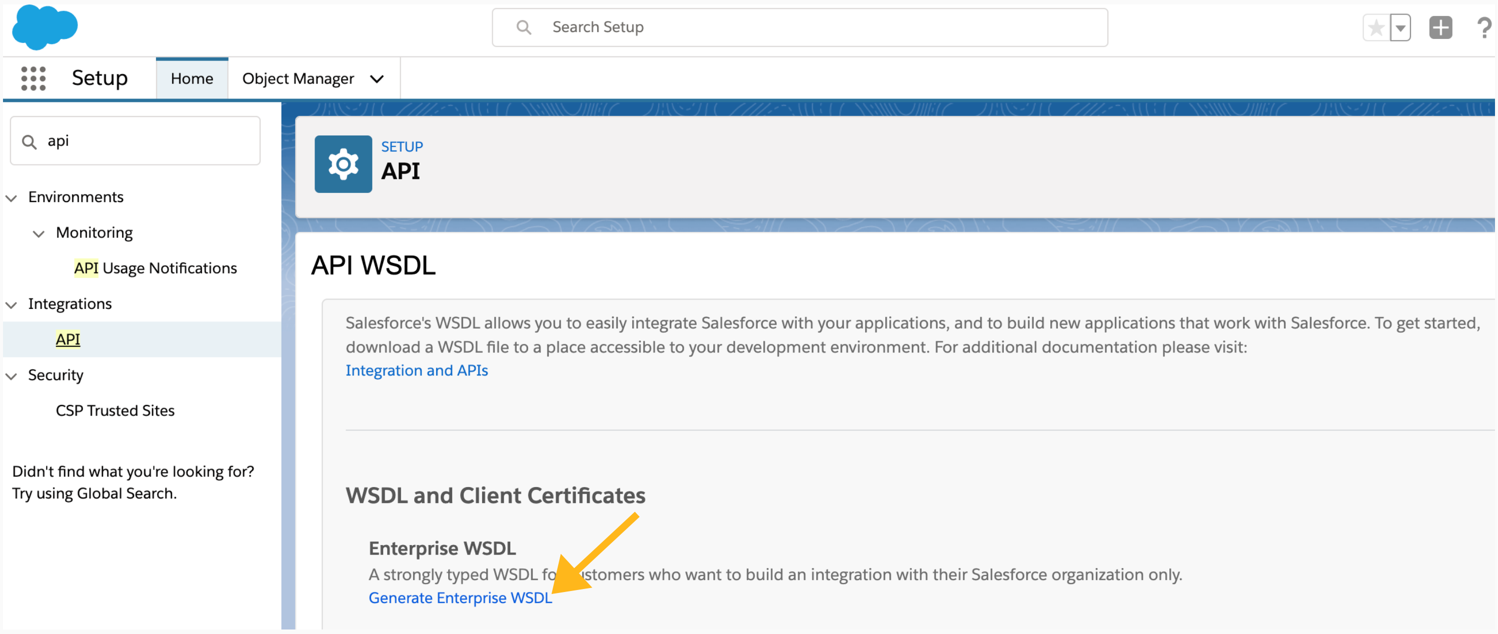
## Deep Dive into Data Integration

In this type of integration, data is synced between two or more applications. The data is consistent in applications once integrated. One of the ways to attain real-time sync is by using SOAP APIs. These APIs let you create, delete, change records, or even perform operations like search within records. SOAP API also allows you to maintain passwords, perform searches, and much more.

You could use SOAP API in any language that supports web services. It provides two WSDLs to choose from - Enterprise WSDL, Partner WSDL. The differences between these two WSDLs can be found below:

| Enterprise Web Services WSDL | Partner Web Services WSDL |
|---|---|
| This type of WSDL is used for building client applications for a specific Salesforce organization. | If there are multiple organizations involved in building a client application then this type of WSDL is used. |
| It has objects and fields with distinct data types. | Unlike Enterprise WSDL, partner WSDL works with name-value pairs of field names and values instead of data types. |
| Developers must download WSDL whenever a change is made in Salesforce or whenever the API version is changed and consume it. | Only when the API version changes, the WSDL needs to be downloaded and consumed. |

To generate WSDL from Salesforce, follow the path **Setup -> Integrations-> API -> Generate Enterprise/Partner WSDL.**



Didn't find what you're looking for? Try using Global Search.

API WSDL

Salesforce's WSDL allows you to easily integrate Salesforce with your applications, and to build new applications that work with Salesforce. To get started, download a WSDL file to a place accessible to your development environment. For additional documentation please visit:
Integration and APIs

WSDL and Client Certificates

Enterprise WSDL

A strongly typed WSDL for customers who want to build an integration with their Salesforce organization only.
Generate Enterprise WSDL

Another way of data integration or sync is by using REST APIs. The REST APIs provide an easy to use REST-based web services interface for interacting with Salesforce. It’s an exemplary choice when it comes to integration due to its ease and it's also widely used in development projects for web and mobile.

# Testing with SOAPUI

SOAPUI is an open-source API testing tool for developers and testers. This tool can check both REST and SOAP-based web services easily. You could download the Salesforce WSDL and test it with SOAP UI before using it with a real client application.

You could import the WSDL into SOAPUI and modify the request message and start testing. While modifying the request message, add the custom fields and objects that are a part of the operation.

For inbound integration to Salesforce, there are two steps to be done:

**01** Get a session ID and server URL using SOAP API login() call. Use the valid credentials of a user. This session ID needs to be added in the session header and the server URL is set as the endpoint for an operation.

**02** Start the operation. For instance operation create()

# Business Scenario

Problem scenario: We will run through a hypothetical integration scenario for a business. In this scenario, we will integrate Force.com with the Oracle Siebel CRM application. CloudIdeas company uses Oracle Siebel CRM at their German headquarters but they expanded to France and the French management team introduced Salesforce Professional Edition.

They did initial bulk data load and synced up master data like accounts, contacts, leads, etc. They decided to build a user interface to sync the data in both applications. Oracle Siebel CRM is used in HQ and the Cloudideas team wants to have this CRM as the main application for viewing their customer information. Once an entry is created in Oracle CRM, this needs to be seen in Salesforce as well and an interface needs to be built.

Considerations for Siebel to Salesforce scenario:

**01** What type of integration feature and what type of API to use? Cloudideas went for SOAP API to consume on the salesforce side and just a web service from Oracle CRM. They thought this would be the best approach because there is no complex logic involved here for data exchange.

**02** Cloudideas used the SOAP API login() call and passed the credentials to authenticate. They also modified the response and tested it with SOAPUI. Oracle CRM has to send this session ID from API call to salesforce.

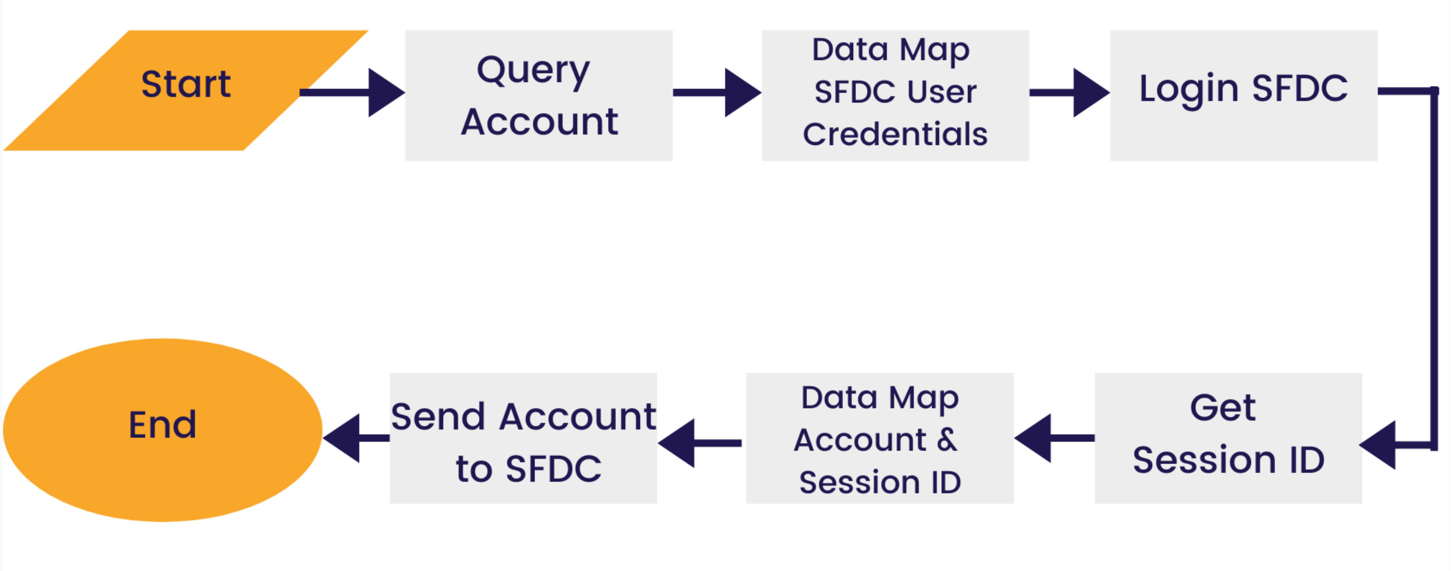
**03** Two records need to be set for Siebel to Salesforce integration. The two endpoints are the login URL and server URL.

Solution:

There is no development to be done on Salesforce for this scenario, you need to just create new fields that are present in Oracle CRM and not in Salesforce. The steps to be completed for implementing Siebel to Salesforce scenario are

**01** Generate Salesforce Enterprise WSDL, download it, and import to Siebel tools. Once you do this, outbound web service and integration objects are created in Oracle CRM. The XML web service administration document must be imported to the Oracle CRM application to create the Outbound web service record. This record is copied once more and pointed to the Server URL.

**02** In Oracle CRM, you could modify the Integration object and add fields from Salesforce that need to be synced. Then create a workflow like the one shown below:



**03** Make a request filter service that changes the Oracle CRM message to Salesforce message format. Do not forget to include the session header and body within the SOAP envelope

Resources:

https://developer.salesforce.com/blogs/developer-relations/2015/06/integrating-force-com.html