**What is Integration?**

**Integration is a process of connecting two applications.** A typical enterprise uses many applications, many or most of which are not designed to work with one another out of the box. Integrating separate but related apps helps organizations achieve greater levels of operational consistency, efficiency, and quality. Each application can have data, business logic, presentation, and security layers, all of which are possible targets for integration. This page is an overall launch point for exploring the various integration options related to Force.com and other Salesforce platform technologies.

**What are possible Ways to Integrate with Salesforce Force.com?**

* *User Interface Integration:*Combine the UIs of two or more apps to create composite apps with little or no rework on the UI of each individual app. One example is using force.com Canvas
* *Business Logic Integration:*It uses Apex Web Services for Inbound and Apex Callouts for Outbound. It typically handles a scenario where business logic is spread across several applications to implement the complete end-to-end business process. An example would be building complex logic on the data received before committing it into Salesforce.
* *Data Integration:*It uses SOAP APIs and REST APIs. It typically handles data synchronization requirements, where one application in an enterprise acts as the primary source for a particular business object, like Account.

**What is webservices?**

Webservices is a functionality or code which helps to us to do integration. Web services are open standard (XML, SOAP, HTTP, etc.) based web applications that interact with other web applications for the purpose of exchanging data. Web services can convert your existing applications into web applications.

**What is Protocal ?**

Protocal is a set of mutually accepted and implemented rules at both ends of the communications channel for the proper exchange of information.

**How many types of API’s avaliable in salesforce?**

* REST API
* SOAP API
* Bulk API
* Streaming API
* Metadata API
* Chatter REST API
* User Interface API
* Analytics REST API
* Apex REST API
* Apex SOAP API
* Tooling API

**When to Use REST API**

REST API provides a powerful, convenient, and simple REST-based web services interface for interacting with Salesforce. Its advantages include ease of integration and development, and it’s an excellent choice of technology for use with mobile applications and web projects. For certain projects, you may want to use REST API with other Salesforce REST APIs. To build UI for creating, reading, updating, and deleting records, including building UI for list views, actions, and dependent picklists, use User Interface API. To build UI for Chatter, communities, or recommendations, use Chatter REST API. If you have many records to process, consider using Bulk API, which is based on REST principles and optimized for large sets of data.

**When to Use SOAP API**

SOAP API provides a powerful, convenient, and simple SOAP-based web services interface for interacting with Salesforce. You can use SOAP API to create, retrieve, update, or delete records. You can also use SOAP API to perform searches and much more. Use SOAP API in any language that supports web services.

For example, you can use SOAP API to integrate Salesforce with your org’s ERP and finance systems. You can also deliver real-time sales and support information to company portals and populate critical business systems with customer information.

**When to Use Chatter REST API**

Use Chatter REST API to display Chatter feeds, users, groups, and followers, especially in mobile applications. Chatter REST API also provides programmatic access to files, recommendations, topics, notifications, Data.com purchasing, and more. Chatter REST API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter, but it also exposes Salesforce features beyond Chatter.

Use Chatter REST API to display Chatter feeds, users, groups, and followers, especially in mobile applications. Chatter REST API also provides programmatic access to files, recommendations, topics, notifications, Data.com purchasing, and more. Chatter REST API is similar to APIs offered by other companies with feeds, such as Facebook and Twitter, but it also exposes Salesforce features beyond Chatter.

### ****When to Use Streaming API****

Use Streaming API to receive near-real-time streams of data that are based on changes in Salesforce records or custom payloads. For Salesforce record changes, Salesforce publishes notifications when the changes occur. For custom notifications, you can publish event messages. Subscribers can receive notifications using CometD—an implementation of the Bayeux protocol that simulates push technology. Clients can subscribe to some types of events with Apex triggers or declaratively with Process Builder and flows.

### ****When to Use Apex REST API****

Use Apex REST API when you want to expose your Apex classes and methods so that external applications can access your code through REST architecture. Apex REST API supports both OAuth 2.0 and Session ID for authorization.

### ****When to Use Apex SOAP API****

Use Apex SOAP API when you want to expose Apex methods as SOAP web service APIs so that external applications can access your code through SOAP.

Apex SOAP API supports both OAuth 2.0 and Session ID for authorization.

**When to Use Tooling API**

Use Tooling API to integrate Salesforce metadata with other systems. Metadata types are exposed as sObjects, so you can access one component of a complex type. This field-level access speeds up operations on complex metadata types. You can also build custom development tools for Force.com applications. For example, use Tooling API to manage and deploy working copies of Apex classes and triggers and Visualforce pages and components. You can also set checkpoints or heap dump markers, execute anonymous Apex, and access logging and code coverage information.

**What is Call In and Call Out?**

Call In is used to exposing our webservices to another system.  
Call out enables you to tightly integrate your Apex with an external service by making a call to an external Web service or sending a HTTP request from Apex code and then receiving the response.

**What is WSDL ?**

WSDL (Web Services Description Language) is an XML document that describes a web service. WSDL is derived from Microsoft’s Simple Object Access Protocol (SOAP) and IBM’s Network Accessible Service Specification Language (NASSL). WSDL replaces both NASSL and SOAP as the means of expressing business services in the UDDI registry. It is used in combination with SOAP and XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL. There are two types of WSDL in  Salesforce:

* **Enterprise WSDL**
* **Partner WSDL**

### ****What is difference between Enterprise WSDL and Partner WSDL?****

#### Enterprise WSDL

* The Enterprise WSDL is strongly typed.
* The Enterprise WSDL is tied (bound) to a specific configuration of Salesforce (ie. a specific organization’s Salesforce configuration).
* The Enterprise WSDL changes if modifications (e.g custom fields or custom objects) are made to an organization’s Salesforce configuration.

#### Partner WSDL

* The Partner WSDL is loosely typed.
* The Partner WSDL can be used to reflect against/interrogate any configuration of Salesforce (ie. any organization’s Salesforce configuration).
* The Partner WSDL is static, and hence does not change if modifications are made to an organization’s Salesforce configuration.

**How SOAP can be accessed ?**

SOAP can be communicate through WSDL file, without WSDL file we can’t do integration.  
Message format in SOAP is XML

**How to generate WSDL file?**

* Go to Setup -> Develop  -> apex classes
* Click on Apex class
* Click on Generate WSDL button
* Generate WSDL code from class
* Save as .wsdl file in your system

**How to consume external WSDL file**

1. In the application, from Setup, enter **Apex Classes** in the Quick Find box, then select **Apex Classes**.
2. Click **Generate from WSDL**.
3. Click **Browse** to navigate to a WSDL document on your local hard drive or network, or type in the full path. This WSDL document is the basis for the Apex class you are creating.
4. Click **Parse WSDL** to verify the WSDL document contents. The application generates a default class name for each namespace in the WSDL document and reports any errors. Parsing fails if the WSDL contains schema types or constructs that aren’t supported by Apex classes, or if the resulting classes exceed the 1 million character limit on Apex classes. For example, the Salesforce SOAP API WSDL cannot be parsed.
5. Modify the class names as desired. While you can save more than one WSDL namespace into a single class by using the same class name for each namespace, Apex classes can be no more than 1 million characters total.
6. Click **Generate Apex**. The final page of the wizard shows which classes were successfully generated, along with any errors from other classes. The page also provides a link to view successfully generated code.

### ****What is remote site settings ?****

Remote site settings is used to authorize the endpoint and allow us to whom integrate(end user)

### ****How manys ways to XML parsing ?****

They are two ways of XML parsing  
1. XML streams  
2. XML DOM

### ****How to read root element in XML DOM**** ?

getroot element

### ****How to read child element in XML DOM ?****

getchild element

### ****How to read all child elements in XML DOM ?****

getchild elements

### ****How to read text between tags ?****

gettext

**How SOAP and REST will Communicate ?**

SOAP will communicate through WSDL file.  
REST will communicate through HTTP protocol.

**What are methods in REST**

* HTTPGET : Retrieve data identified by a URL.
* HTTPPOST : Create a resource or post data to the server.
* HTTPDELETE : Delete a resource identified by a URL.
* HTTPPUT : Create or replace the resource sent in the request body.

**What is difference between SOAP and REST?**

**SOAP API: The Simple Object Access Protocol (SOAP)** is an attempt to define a standard for creating web service APIs. It is a pattern, a web service architecture, which specifies the basic rules to be considered while designing web service platforms. It typically uses HTTP as a layer 7 protocol, although this is not mandatory.

The SOAP message itself consists of an envelope, inside of which are the SOAP headers and body, the actual information we want to send. It is based on the standard XML format, designed especially to transport and store structured data. SOAP may also refer to the format of the XML that the envelope uses. SOAP is a mature standard and is heavily used in many systems, but it does not use many of the functionality build in HTTP. While some consider it slow, it provides a heavy set of functionality which is a necessity in many cases. It might now be the best solution for browser-based clients, due to its custom format.

* Supports data in the form of XML only and not JSON.
* Requires WSDL for the integration
* Use SOAP API in any language that supports Web services.

**REST API: The Representational State Transfer (REST)** is another architectural pattern (resource-oriented), an alternative to SOAP. Unlike SOAP, RESTful applications use the HTTP build-in headers (with a variety of media-types) to carry meta information and use the GET, POST, PUT and DELETE verbs to perform CRUD operations. REST is resource-oriented and uses clean URLs (or RESTful URLs).

* Supports both XML and JSON format
* Preferred for mobile and web apps since JSON being Lighter the app runs smoother and faster

**What is JSON ?**

JSON stands for JavaScript Object Notation. JSON is light weighted than XML.  When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server.