**What is a Web Service?**

Web Services work on client-server model where client applications can access web services over the network. Web services provide endpoint URLs and expose methods that can be accessed over network through client programs written in java, shell script or any other different technologies.  
Web services are stateless and doesn’t maintain user session like web applications.

**What are the advantages of Web Services?**

Some of the advantages of web services are:

* Interoperability: Web services are accessible over network and runs on HTTP/SOAP protocol and uses XML/JSON to transport data, hence it can be developed in any programming language. Web service can be written in java programming and client can be PHP and vice versa.
* Reusability: One web service can be used by many client applications at the same time.
* Loose Coupling: Web services client code is totally independent with server code, so we have achieved loose coupling in our application.
* Easy to deploy and integrate, just like web applications.
* Multiple service versions can be running at same time.

**What are different types of Web Services?**

There are two types of web services:

1. SOAP Web Services: Runs on SOAP protocol and uses XML technology for sending data.
2. Restful Web Services: It’s an architectural style and runs on HTTP/HTTPS protocol almost all the time. REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services.

### What is SOAP?

SOAP stands for Simple Object Access Protocol. SOAP is an XML based industry standard protocol for designing and developing web services. Since it’s XML based, it’s platform and language independent. So our server can be based on JAVA and client can be on .NET, PHP etc. and vice versa.

1. **What are advantages of SOAP Web Services?**

SOAP web services have all the advantages that web services has, some of the additional advantages are:

* + WSDL document provides contract and technical details of the web services for client applications without exposing the underlying implementation technologies.
  + SOAP uses XML data for payload as well as contract, so it can be easily read by any technology.
  + SOAP protocol is universally accepted, so it’s an industry standard approach with many easily available open source implementations.

1. **What are disadvantages of SOAP Web Services?**

Some of the disadvantages of SOAP protocol are:

* + Only XML can be used, JSON and other lightweight formats are not supported.
  + SOAP is based on the contract, so there is a tight coupling between client and server applications.
  + SOAP is slow because payload is large for a simple string message, since it uses XML format.
  + Anytime there is change in the server side contract, client stub classes need to be generated again.
  + Can’t be tested easily in browser

### What is WSDL?

WSDL stands for Web Service Description Language. WSDL is an XML based document that provides technical details about the web service. Some of the useful information in WSDL document are: method name, port types, service end point, binding, method parameters etc.

**What are different components of WSDL?**

Some of the different tags in WSDL xml are:

* xsd:import namespace and schemaLocation: provides WSDL URL and unique namespace for web service.
* message: for method arguments
* part: for method argument name and type
* portType: service name, there can be multiple services in a wsdl document.
* operation: contains method name
* soap:address for endpoint URL.

### What is UDDI?

UDDI is acronym for Universal Description, Discovery and Integration. UDDI is a directory of web services where client applications can lookup for web services. Web Services can register to the UDDI server and make them available to client applications.

### What is difference between Top Down and Bottom Up approach in SOAP Web Services?

In Top Down approach first WSDL document is created to establish the contract between web service and client and then code is written, it’s also termed as contract first approach. This is hard to implement because classes need to be written to confirm the contract established in WSDL. Benefit of this approach is that both client and server code can be written in parallel.

In Bottom Up approach, first web service code is written and then WSDL is generated. It’s also termed as contract last approach. This approach is easy to implement because WSDL is generated based on code. In this approach client code have to wait for WSDL from server side to start their work.

### What is REST Web Services?

REST is the acronym for REpresentational State Transfer. REST is an architectural style for developing applications that can be accessed over the network. REST architectural style was brought in light by Roy Fielding in his doctoral thesis in 2000.

REST is a stateless client-server architecture where web services are resources and can be identified by their URIs. Client applications can use HTTP GET/POST methods to invoke Restful web services. REST doesn’t specify any specific protocol to use, but in almost all cases it’s used over HTTP/HTTPS. When compared to SOAP web services, these are lightweight and doesn’t follow any standard. We can use XML, JSON, text or any other type of data for request and response.

**What are advantages of REST web services?**

Some of the advantages of REST web services are:

* Learning curve is easy since it works on HTTP protocol
* Supports multiple technologies for data transfer such as text, xml, json, image etc.
* No contract defined between server and client, so loosely coupled implementation.
* REST is a lightweight protocol
* REST methods can be tested easily over browser.

**What are disadvantages of REST web services?**

Some of the disadvantages of REST are:

* Since there is no contract defined between service and client, it has to be communicated through other means such as documentation or emails.
* Since it works on HTTP, there can’t be asynchronous calls.
* Sessions can’t be maintained.

### What is a Resource in Restful web services?

Resource is the fundamental concept of Restful architecture. A resource is an object with a type, relationship with other resources and methods that operate on it. Resources are identified with their URI, HTTP methods they support and request/response data type and format of data.

### What are different HTTP Methods supported in Restful Web Services?

Restful web services supported HTTP methods are – GET, POST, PUT, DELETE and HEAD.

**Compare SOAP and REST web services?**

|  |  |
| --- | --- |
| SOAP | REST |
| SOAP is a standard protocol for creating web services. | REST is an architectural style to create web services. |
| SOAP is acronym for Simple Object Access Protocol. | REST is acronym for REpresentational State Transfer. |
| SOAP uses WSDL to expose supported methods and technical details. | REST exposes methods through URIs, there are no technical details. |
| SOAP web services and client programs are bind with WSDL contract | REST doesn’t have any contract defined between server and client |
| SOAP web services and client are tightly coupled with contract. | REST web services are loosely coupled. |
| SOAP learning curve is hard, requires us to learn about WSDL generation, client stubs creation etc. | REST learning curve is simple, POJO classes can be generated easily and works on simple HTTP methods. |
| SOAP supports XML data format only | REST supports any data type such as XML, JSON, image etc. |
| SOAP web services are hard to maintain, any change in WSDL contract requires us to create client stubs again and then make changes to client code. | REST web services are easy to maintain when compared to SOAP, a new method can be added without any change at client side for existing resources. |
| SOAP web services can be tested through programs or software such as Soap UI. | REST can be easily tested through CURL command, Browsers and extensions such as Chrome Postman. |

### What are different ways to test web services?

SOAP web services can be tested programmatically by generating client stubs from WSDL or through software such as Soap UI.

REST web services can be tested easily with program, curl commands and through browser extensions. Resources supporting GET method can be tested with browser itself, without any program.

### Can we maintain user session in web services?

Web services are stateless so we can’t maintain user sessions in web services.

### What is difference between SOA and Web Services?

Service Oriented Architecture (SOA) is an architectural pattern where applications are designed in terms of services that can be accessed through communication protocol over network. SOA is a design pattern and doesn’t go into implementation.

Web Services can be thought of as Services in SOAP architecture and providing means to implement SOA pattern.

### What is JAX-WS API?

JAX-WS stands for Java API for XML Web Services. JAX-WS is XML based Java API to build web services server and client application. It’s part of standard Java API, so we don’t need to include anything else which working with it.

### What is JAX-RS API?

Java API for RESTful Web Services (JAX-RS) is the Java API for creating REST web services. JAX-RS uses annotations to simplify the development and deployment of web services. JAX-RS is part of JDK, so you don’t need to include anything to use it’s annotations.

**Name some implementations of JAX-RS API?**

There are two major implementations of JAX-RS API.

1. Jersey: Jersey is the reference implementation provided by Sun. For using Jersey as our JAX-RS implementation, all we need to configure its servlet in web.xml and add required dependencies. Note that JAX-RS API is part of JDK not Jersey, so we have to add its dependency jars in our application.
2. RESTEasy: RESTEasy is the JBoss project that provides JAX-RS implementation.

**Name important annotations used in JAX-RS API?**

Some of the important JAX-RS annotations are:

* @Path: used to specify the relative path of class and methods. We can get the URI of a webservice by scanning the Path annotation value.
* @GET, @PUT, @POST, @DELETE and @HEAD: used to specify the HTTP request type for a method.
* @Produces, @Consumes: used to specify the request and response types.
* @PathParam: used to bind the method parameter to path value by parsing it.

REST stands for REpresentational State Transfer.

REST is web standards based architecture and uses HTTP Protocol for data communication. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. REST was first introduced by Roy Fielding in 2000.

In REST architecture, a REST Server simply provides access to resources and REST client accesses and presents the resources. Here each resource is identified by URIs/ global IDs. REST uses various representations to represent a resource like text, JSON and XML. Now a days JSON is the most popular format being used in web services.

Following well known HTTP methods are commonly used in REST based architecture −

* **GET** − Provides a read only access to a resource.
* **PUT** − Used to create a new resource.
* **DELETE** − Ued to remove a resource.
* **POST** − Used to update a existing resource or create a new resource.
* **OPTIONS** − Used to get the supported operations on a resource.

A web service is a collection of open protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer.

Web services based on REST Architecture are known as RESTful web services. These web services use HTTP methods to implement the concept of REST architecture. A RESTful web service usually defines a URI, Uniform Resource Identifier a service, provides resource representation such as JSON and set of HTTP Methods.

REST architecture treats every content as a resource. These resources can be text files, html pages, images, videos or dynamic business data. REST Server simply provides access to resources and REST client accesses and modifies the resources. Here each resource is identified by URIs/ global IDs.

REST uses various representations to represent a resource where text, JSON, XML. XML and JSON are the most popular representations of resources.

RESTful web services make use of HTTP protocol as a medium of communication between client and server.

A client sends a message in form of a HTTP Request and server responds in form of a HTTP Response. This technique is termed as Messaging. These messages contain message data and metadata i.e. information about message itself.

A HTTP Request has five major parts −

* **Verb** − Indicate HTTP methods such as GET, POST, DELETE, PUT etc.
* **URI** − Uniform Resource Identifier (URI) to identify the resource on server.
* **HTTP Version** − Indicate HTTP version, for example HTTP v1.1 .
* **Request Header** − Contains metadata for the HTTP Request message as key-value pairs. For example, client ( or browser) type, format supported by client, format of message body, cache settings etc.
* **Request Body** − Message content or Resource representation.

A HTTP Response has four major parts −

* **Status/Response Code** − Indicate Server status for the requested resource. For example 404 means resource not found and 200 means response is ok.
* **HTTP Version** − Indicate HTTP version, for example HTTP v1.1 .
* **Response Header** − Contains metadata for the HTTP Response message as key-value pairs. For example, content length, content type, response date, server type etc.
* **Response Body** − Response message content or Resource representation.

Addressing refers to locating a resource or multiple resources lying on the server. It is analogous to locate a postal address of a person.

A URI is of following format −

<protocol>://<service-name>/<ResourceType>/<ResourceID>

Following are the benefits of statelessness in RESTful web services −

* Web services can treat each method request independently.
* Web services need not to maintain client's previous interactions. It simplifies application design.
* As HTTP is itself a statelessness protocol, RESTful Web services work seamlessly with HTTP protocol.

Following is the disadvantage of statelessness in RESTful web services −

Web services need to get extra information in each request and then interpret to get the client's state in case client interactions are to be taken care of.

PUT and POST operation are nearly same with the difference lying only in the result where PUT operation is idempotent and POST operation can cause different result.

As RESTful web services work with HTTP URLs Paths so it is very important to safeguard a RESTful web service in the same manner as a website is be secured. Following are the best practices to be followed while designing a RESTful web service −

* **Validation** − Validate all inputs on the server. Protect your server against SQL or NoSQL injection attacks.
* **Session based authentication** − Use session based authentication to authenticate a user whenever a request is made to a Web Service method.
* **No sensitive data in URL** − Never use username, password or session token in URL , these values should be passed to Web Service via POST method.
* **Restriction on Method execution** − Allow restricted use of methods like GET, POST, DELETE. GET method should not be able to delete data.
* **Validate Malformed XML/JSON** − Check for well formed input passed to a web service method.
* **Throw generic Error Messages** − A web service method should use HTTP error messages like 403 to show access forbidden etc.

JAX-RS stands for JAVA API for RESTful Web Services. JAX-RS is a JAVA based programming language API and specification to provide support for created RESTful Webservices. Its 2.0 version was released in 24 May 2013. JAX-RS makes heavy use of annotations available from Java SE 5 to simplify development of JAVA based web services creation and deployment. It also provides supports for creating clients for RESTful web services.