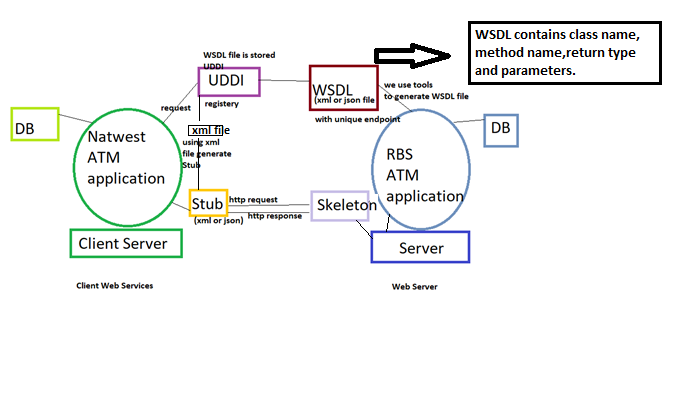
**Web Services: Web Services are s set of technologies and rues enable two are more components on the web to communicate to each other.**



**Web services provides only specifications (set of rules) not the API.**

**Specifications:**

1. **WSDL -** Web Service Description Language. (is xml file or json file and contains unique endpoint URL and also contains class name, methods, return type, paramaters).
2. **UDDI -** Universal Description, Discovery and Integration. (Online registry to store WSDL file) [optional]
3. **Skeleton**
4. **Stub protocol**
5. **SOAP protocol**
6. **HTTP protocol [ Transportation protocol ]**

**Two types of Web Services.**

1. **SOAP - Simple Object Access Protocol.**

**A web service that compiles to the SOAP web services specifications is a SOAP Web Services.**

**In case of SOAP is XML format. (Envelop containing Header provides information about message, Body contains all the actual message)**

**Jax – RPC -- API provided by Sun microsystems** (API contains abstract classes and interface)

**Jax – WS – API provided by Sun microsystems** (API contains abstract classes and interface)

**Axis1 – provided by apache foundations** (API contains abstract classes and interface)

**Note:** All API’s contains WSDL generating tools, Skeleton generating tools and Stub generating tools. using this API’s we can create both Server and Client Web Services for web.

1. **REST – Representational State Transfer.**

**Is an architectural style.**

**REST defines a set of principles to be followed while designing a service for communication/ data exchange between 2 applications.**

**When these principles are applied while designing web services (for client-server interactions) we get RESTful Web Services.**

**Jax – RS -- API provided by Sun microsystems** (API contains abstract classes and interface)

**Note:** All API’s contains WSDL generating tools, Skeleton generating tools and Stub generating tools. using this API’s we can create both Server and Client Web Services for web.

**Axis1 Web Services using Eclipse:**

**First, we have to create Axis1 Web service Server and run on Server.**

we need to create dynamic web project.

Create service class.

Follow few steps to convert call.class service into web service.

Configure the skeleton in web.xml config.wsdl [ mapping details], web.INF

**[ Note: JDK, Eclipse Enterprise version and Tomcat is already installed]**

**Step 1: add Tomcat Server to the project.**

**Eclipse IDE > Window > preference > choose Server > and then Runtime environment > add > Apache Tomcat v9.0 > and next Browser to the folder Tomcat directory > finish > apply and close.**

**[ to check the server is configured go to > Window > show view > others > Server> and can run on Servers tab in near console]**

**Step 2: next to create Axis1 Web service.**

**Eclipse > File > New > Others > Web > Dynamic Web Project and name the folder.**

**Create a cal.class file in src folder and write methods in this class file.**

**Step 3: convert cal service into web service using Axis1 [ i.e it generates wsdl file, configure skeleton in web,xml with url pattern, generate server configuration server-config wsdl and web.xml].**

**Right click cal service >new > other> expand web services folder > select web service > next > bottom up approach> finish button.**

**Second, we have to create Axis1 Web service client.**

**create java project [ we need wsdl file, means copy wsdl file to project using wsdl file we will generate stubs and we will invoke web services]**

**Type of Web service client:**

1. **Proxy based client** [ if client generate is stub and client uses those stubs in client appliaction ]
2. **DII client [ Dynamic invocation interface,** they not create stubs but will use API web servIces implementation providers**]**

**Step 1: create a java project**

**File>new>project>java project>name calserviceclient>finish.**

**Step 2: copy wsdl file into client project. [ we need wsdl file, means copy wsdl file to project using wsdl file we will generate stubs and we will invoke web services]**

**Step 3: generate stubs or proxies**

**Right click src folder> new>others>web services > web service client>next >choose the service definition> browse to wsdl file> next> finish.**

**Step 4: create java file with main method and use generated stubs to invoke web serives.**

**Before cresting stub object we need to two obects URL object and service object.**

**To create stub object in java file [i.e CalServiceSoapBindingStub(URL, Service)**

**public class TestClient{**

**public static void main (Stirng[] args)**

**{**

**// create URL object by passing Endpoint URL from wsdl file, in end <service> <portal> <address> <location> attribute copy EndpointUrl;**

**Java.net.URL endpointUrl= new java.net.URL(“ paste endpointurl”);**

**// create service object**

**Org.apache.axis.client.Service serive= new Org.apache.axis.client.Service**

**CalSericveSoapBindingStub stub= new CalServiceSoapBinding(endpointUrl, service);**

**// then we can invoke methods from the Web service using stub object.**

**}**

**}**

**Then run the java project in eclipse.**

WSDL – Web Service Description Language.

* Is an XML based interface that is used to describe the functionalities of the Web Services.

UDDI - Universal Description, Discovery and Integration.

* is online registry where all the Service providers put their WSDL