Web Services

Definition:

1. If we want to share information between two application, then we should go for Web Services.

2. We Services is just a specification, not a technology, it is just providing set of rules and guideline.

Before Web service we have below Technology:

1 1. Socket Programming.

2. RMI

3. EJB

4. DCOM(Distributed component)

5. RPC

6. CORBA(Common object request broker architecture).

How information sharing between two application:

11. Consider we are sharing information between cpp and java.

2. We will create a generic format that format can be understandable to any language.

3. We will choose XML language that is understandable to any language, it is universal language.

4. We will store below details in XML file :

i. Service Name

ii. Method Name

iii. Parameter of the method and data type.

iv. Return Type.

v. End point URL of the service.(The location where your service is running)

5. This XML file is called as WSDL(Web Service Description Language).

6. Service provider will create this documents(WSDL).

7. We have tool to generate this file is call WSDL generation tool, it is a predefine class.

8. How this file share to client….???

i. Send email by attaching this file.

ii. Store this file to UDDI registry and provide end point to client about this file.

9. How client access this file :

i. Client will generate some classes on their specific language.

ii. This classes are called as STUBS(proxy).

iii. We will use some tool to generate this STUBS, this tool is call STUBS gen tool or proxy.

iv. STUDS is predefined class.

v. Client will create XML document, this document contains predefined tags called as SOAP.

10. Once again we will create class that will read soap request and invoke service method, it will create service class object. This call is called as Skeleton, it is a servlet, it will handle the request.

11. Response send back to client through HTTP protocols.

Web Services components:

**1. WSDL(Web Service Description Language).**

**2. Skeleton**

**3. UDDI registry**

**4. STUB**

**5. SOAP**

**6. HTTP**

Interface Used to develop Web Services:

**1. JAX-RPC (Java API for XML Remote Procedure call) ( JDK 1.4 )**

**2. JAX-M (Java API for XML - Messaging) ( JDK 1.4 )**

**3. JAX-WS (Java API for XML - Webservice) ( JDK 1.5 )**

**4. JAX-RS (Java API for XML - Restful Webservice) ( JDK 1.6 )**

Two Type of Web service Synchronous vs Asynchronous:

1. We will develop only synchronous web services. For Synchronous we will use JAX-RPC, JAX-WS, JAX-RS.

2. For Asynchronous we will use JAX-M.

Synchronous web services two types:

**1. SOAP base web services**

**2. RESTFUL web services.**

**3. To develop SOAP base service we will use JAX-RPC and JAX-WS.**

**4. To develop Restful web service we will use JAX-RS.**

Implementation Provider:

**1. JAX-RPC**

i. JAX-RPC-SI(Sun Implementation)

ii. Axis 1

iii. WS

iv. JW (j Boss)

**2. JAX-WS**

i. JAX-WS-RI(Reference implementation) (Sun). **( JDK 1.6 )**

ii. Metro(Sun)

Note : (JAX-WS-RI and Metro only Jar file are different)

iii. Axis 2 (Apache)

iv. ApacheCFX.(Apache)

Most important implementation is ApacheCFX because it support spring integration.

v. Weblogic, Web spear, JBoss, Glass Fish

**3. JAX-RS**

i. Jersey imp(sun)

ii. RESTeasy(Red Hat)

iii. Restlet

iv. Apache wink.

v. Apache CFX :

Most important implementation is ApacheCFX because it support spring integration.

Steps to prepare Web service application:

1. Create Web Application.

2. Download webservice implementation jar and copy then into LIB folder.

3. Configure Skeleton in web.xml file with some URL pattern.

4. Create a service class and copy it into WEB-INF classes folder.

5. Create web service configuration file.

6. Deploy application in server.

Eclipse support only three implementation ::

1. Axis 1 (JAX-RPC)

2. Axis 2 (JAX-WS)

3. Apache CFX. (JAX-WS)

Restful web services::

REST : Representational State Transform :

WADL: (Web application description language)

3. Apache Converting Object state into different formates ie., xml, json, plane text, html

1. JAVA to JSON & JSON to JAVA API:

i. Jackson Api

ii. JSON Api

iii. Simple Json.

JAX=RS annotation:

JAX-RS having lot of annotations and these annotations we will use

irrespective of the implementation.

1. @Path(javax.ws.rs.Path)

2. @GET (javax.ws.rs.GET) NM ARR

3. @POST(javax.ws.rs.POST) DUR NG NS) OF1

4. @PUT (javax.ws.rs.PUT)

5. @DELETE (javax.ws.rs.DELETE) picks

6. @QueryParam(javax.ws.rs.QueryParam)

7. @PathParam(javax.ws.rs.PathParam)

8. @MatrixParam(javax.ws.rs.MatrixParam)

9. @FormParam(javax.ws.rs.FormParam)

10. @Consumes(javax.ws.rs.Consumes)

11. @Produces(javax.ws.rs.Produces)

The rest method can accept only below types as a return type ie.,

i. String

ii. StringBuffer.

iii. StringBuilder.

iv. User define type.

v. Response.(We want some status code like 400, 401, 200 then we will use this Response)

It will not take any premitive and wrapper type as rest method return type.

Steps to prepare Restful web services :

1. Create Dynamic Web project.

2. Download jar files and copy into WEB-INF/lib folder.

3. Create service class with jax-rs annotation.

4. Configure skeleton in web.xml file with some url-pattern.

5. Deploy appliction into server.

1.

1