Unit : 6 : SOA Platforms

* Explain JavaAPIFor XML-Based Web Services.

=> For Create XML-Based Web Services, we have to use Jax-WS and JAXB.

JAX-WS stands for JavaAPIFor XML-Based Web Services which allows to build web services using XML-based messaging.

JAXB stands for Java Architecture for XML Binding which helps in marshaling and unmarshaling Java objects to and from XML.

There are two step to create JavaAPI for XML-Based web services.

1) Creating XML-Based web Services.
2) Consuming XML-Based web Services.

1 Creating XML-Based Web Services.

This are the step for create web services.

- Step 1: Define Your Service Interface.

Ex.

```
import javax.jws.*;

public interface WS
{
    String say(String name);
}
```

- Step 2: Implement the Service Interface

Ex.

```
import javax.jws.*;
public class WS1 implements
        WS
{
    String say(String name)
    {
        return "Hello " tname;
    }
}
```

- Step 3 : Publish the services

Ex.

```
import javax.xml.ws.Endpoint;

class Publish
{
  public static void main(String
    args[])
  {
    Endpoint.publish("http://
      localhost:8080/ws", new
      WS1());
  }
}
```

2 Consuming XML - Based Web Services.

This are the Basic step:

- Step 1 : Generate Java classes from the WSDL
- Step 2 : Create a client to consume the web services.

Page No.
Date : / /

Ex.

```
import com.example.WS1;
import com.example.WS;

class Consumer
{
  public static void main
    (String, args[])
  {
    Service S = new Service();
    MyService S1 = S.getMyWS
        ImplPort();

    String response = port.
        say("Hello");

    System.out.println(response);
  }
}
```

* Explain Java Architecture for XML Binding (JAXB).

=> Java Architecture for XML Binding allows to convert Java Object into XML Representation.

JAXB is convert Java Objects into XML which is called Marshaling.

JAXB is Convert XML to Java Objects which is called Unmarshaling.

JAXB is automatically generate the mapping between XML elements and Java classes using annotations or configuration Files.

This are the basic step for JAXB:

Step 1: Annotating Class creation

Ex.

```
import javax.xml.bind.annotation.
        XmlRootElement;

Class Person{
        private String name;
        private int age;

}
```

Step 2 : Marshaling :

Ex.

```
import javax.xml.bind.*;
import java.io.*;

Person person = new Person
                ("K", 20);

JAXBContext context =
    JAXBContext.newInstance
        (Person.class);

Marshaller marshaller =
        context.createMarshaller();

StringWriter writer = new
        StringWriter();

marshaller.marshal(person,
        writer);

String xml = writer.toString();

System.out.println(xml);
```

Step 3: Unmarshaling :

Ex.

```
import javax.xml.bind.*;
import java.io.*;

String xml = "<person><name>k
        </name><age>20</age>
        </person>";

JAXBContext context = JAXBContext.
    newInstance(Person.class);

Unmarshaller unmarshaller =
    context.createUnmarshaller();

StringReader reader = new
        StringReader(xml);

Person person = (Person)
    unmarshaller.unmarshal
        (reader);

System.out.println(person.getName
    () + person.getAge());
```

Date : / /

* Draw Diagram of SOA support in .NET Platform.

Page No.
Date : / /

* Draw the Diagram of SOA support J2EE