

Java Web Services

The services that are accessible across various networks are Java Web Services. Since JavaEE 6, it has two APIs defined by Java for creating web services applications.

- JAX-WS for SOAP web services
- JAX-RS for RESTful web services.

It is not necessary to add any jars in order to work with either of these APIs because they both use heavy annotations and are included in the standard JDK.

JAX-WS

The Jakarta EE API, known as JAX-WS, is used to develop and create web services, especially for SOAP service users. The development and deployment of web services clients and endpoints are made simpler by using annotations.

Two ways to write JAX-WS application code are:

- RPC Style
- Document Style

JAX-RS

JAX-RS is a framework for building RESTful web applications. Currently, there are two implementations for building JAX-RS :

- Jersey
- RESTeasy

Implementation of Java Web Services

1. Implementing SOAP Web Services with JAX-WS

There are certain steps to implement SOAP Web Services with JAX-WS

1. First, you need to define Service endpoint interfaces (SEI) which specify the methods to expose as web service.
2. Next, you need to implement SEI with a Java class.
3. Then you need to Annotate the SEI and its implementation class with JAX-WX annotations to specify the web service details.
4. Package web service classes to the WAR file and deploy it to a web server.

2. Implementing RESTful Web Services with JAX-RS

There are certain steps to implement RESTful Web Services with JAX-RS

1. First you need to define the resources that represents the web services and its methods.
2. Then you need to annotate the resource class and its methods with JAX-RS annotations to specify the web service package.
3. At last we need to package the web service classes to the WAR file and deploy it to a web server.

Examples of Java Web Services

This is an example of how we can use JAX-WS to create a Java Web Service (JWS) using the data from the search results.

1. The SEI (Service Endpoint Interface), which shows the procedures of web services

- Java

```
import javax.xml.ws.WebMethod;  
import javax.xml.ws.WebService;  
  
@WebService  
public interface HelloWorld {  
    @WebMethod  
    String sayHello(String name);  
}
```

2. Implement the SEI with java class

- Java

```
import javax.xml.ws.WebService;  
  
@WebService(endpointInterface = "com.example.HelloWorld")  
public class HelloWorldImpl implements HelloWorld {  
    public String sayHello(String name) {  
        return "Hello " + name + "!";  
    }  
}
```

3. Annotate the SEI and it's implementation class with JAX-WX annotations to specify the web services.

- Java

```
import javax.xml.ws.WebMethod;
```

```
import javax.xml.ws.WebService;
```

```
@WebService
```

```
public interface HelloWorld {
```

```
    @WebMethod
```

```
    String sayHello(String name);
```

```
}
```

```
import javax.xml.ws.WebService;
```

```
@WebService(endpointInterface = "com.example.HelloWorld")
```

```
public class HelloWorldImpl implements HelloWorld {
```

```
    public String sayHello(String name) {
```

```
        return "Hello " + name + "!";
```

```
    }
```

```
}
```

