UNIT-V

XML

❖ **XML:**

- Xml (Extensible Markup Language) is a markup language.
- XML is designed to store and transport data.
- Xml was released in late 90's. It was created to provide an easy to use and store self-describing data.
- XML is not a replacement for HTML.
- XML is designed to be self-descriptive.
- XML is designed to carry data, not to display data.
- XML tags are not predefined. You must define your own tags.
- XML is platform independent and language independent.

Structure of XML:

- XML documents create a hierarchical structure looks like a tree so it is known as XML Tree that starts at "the root" and branches to "the leaves".
- XML documents must contain a root element**.** This element is "the parent" of all other elements.
- The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree. All elements can have sub elements (child elements).
- XML documents uses a self-describing and simple syntax:

```
<root>
    <child>
        <subchild>.....</subchild>
    </child>
</root>
```

Example of XML Document:

XML documents uses a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
```

```
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

Output:

Note

To: Tove

From: Jani

Reminder

Don't forget me this weekend!

HTML Program :

```
<!DOCTYPE html>
<html>
<h1>Note</h1>
<body>
<p>To:RAJ
<br>
From:RAVI
</p>
<h1>Reminder</h1>
<p>Meeting at 8am</p>
</body>
</html>
```

Output:

Note

To:RAJ
From:RAVI

Reminder

Meeting at 8am

Differences between XML and HTML:

- XML and HTML were designed with different goals:
- XML is designed to carry data emphasizing on what type of data it is.
- HTML is designed to display data emphasizing on how data looks
- XML tags are not predefined like HTML tags.
- HTML is a markup language whereas XML provides a framework for defining markup languages.
- HTML is about displaying data, hence it is static whereas XML is about carrying information, which makes it dynamic.

❖ **XSL:**

- XSL is a language for expressing style sheets. An XSL style sheet is, like with CSS, a file that describes how to display an XML document of a given type.
- A transformation language for XML documents: XSLT. Originally intended to perform complex styling operations, like the generation of tables of contents and indexes, it is now used as a general purpose XML processing language.
- XSLT is thus widely used for purposes other than XSL, like generating HTML web pages from XML data.
- In HTML documents, tags are predefined but in XML documents, tags are not predefined

An XSL document specifies how a browser should render an XML document.

Main parts of XSL Document:

1. **XSLT:** It is a language for transforming XML documents into various other types of documents.
2. **XPath:** It is a language for navigating in XML documents.
3. **XQuery:** It is a language for querying XML documents.
4. **XSL-FO:** It is a language for formatting XML documents.

❖ **XSLT:**

- XSLT stands for Extensible Style sheet Language Transformation.
- XSLT stands for XSL Transformation. It is used to transform XML documents into other formats (like transforming XML into HTML).
- XSLT is used to transform XML document from one form to another form.

- XSLT uses Xpath to perform matching of nodes to perform these transformation .
- The result of applying XSLT to XML document could be an another XML document, HTML, text or any another document from technology perspective.
- The XSL code is written within the XML document with the extension of *(.xsl)*.
- In other words, an XSLT document is a different kind of XML document.

XML Namespace: XML Namespaces are the unique names .

- XML Namespace is a mechanism by which element or attribute is assigned to a group.
- XML Namespace is used to avoid the name conflicts in the XML document.
- XML Namespace is recommended by W3C.

XML Namespace Declaration:

It is declared using reserved attribute such as the attribute is *xmlns* or it can begin with *xmlns:*

Syntax:

<element xmlns:name = "URL">

where

- Namespace starts with the xmlns.
- The word name is the namespace prefix.
- the URL is the namespace identifier.

- **Example:**

  Consider the following xml document named Table.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="rule.css"?>
<tables>
<table>
<tr>
    <td>Apple</td>
    <td>Banana</td>
</tr>
</table>
<table>
<height>100</height>
<width>150</width>
```

</table>

</tables>

Xpath:

- Xpath is an important component of XSLT standard.
- Xpath is used to traverse the element and attributes of an XML document.
- Xpath uses different types of expression to retrieve relevant information from the XML document.
- Xpath contains a library of standard functions.

  Example:

  - bookstore/book[1] => Fetches details of first child of bookstore element.
  - bookstore/book[last()] => Fetches details of last child of bookstore element.

Templates:

- An XSL stylesheet contains one or more set of rules that are called templates.
- A template contains rules that are applied when the specific element is matched.
- An XSLT document has the following things:
  - The root element of the stylesheet.
  - A file of extension .xsl .
  - The syntax of XSLT i.e what is allowed and what is not allowed.
  - *The standard namespace whose URL is http://www.w3.org/1999/XSL/Transform.*

Example:

In this example, creating the XML file that contains the information about five students and displaying the XML file using XSLT.

XML file:

Creating Students.xml as:

<?xml version="1.0" encoding="UTF-8"?>

<?xml-stylesheet type="text/xsl "href="Rule.xsl" ?>

<student>

<s>

<name> Divyank Singh Sikarwar </name>

<branch> CSE</branch>

<age>18</age>

<city> Agra </city>

</s>

<s>

<name> Aniket Chauhan </name>

<branch> CSE</branch>

<age> 20</age>

<city> Shahjahanpur </city>

</s>

<s>

<name> Simran Agarwal</name>

<branch> CSE</branch>

<age> 23</age>

<city> Buland Shar</city>

</s>

<s>

<name> Abhay Chauhan</name>

<branch> CSE</branch>

<age> 17</age>

<city> Shahjahanpur</city>

</s>

<s>

<name> Himanshu Bhatia</name>

<branch> IT</branch>

<age> 25</age>

<city> Indore</city>

</s>

</student>

XSLT Code:

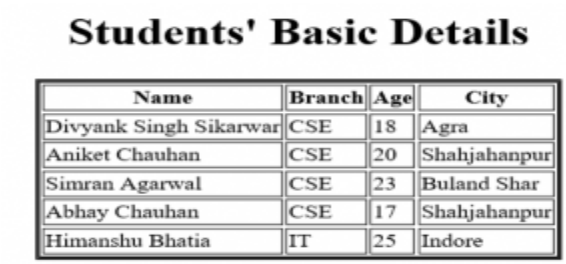
Creating Rule.xsl as:

<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"

```
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h1 align="center">Students' Basic Details</h1>
<table border="3" align="center" >
<tr>
    <th>Name</th>
    <th>Branch</th>
    <th>Age</th>
    <th>City</th>
</tr>
    <xsl:for-each select="student/s">
<tr>
    <td><xsl:value-of select="name"/></td>
    <td><xsl:value-of select="branch"/></td>
    <td><xsl:value-of select="age"/></td>
    <td><xsl:value-of select="city"/></td>
</tr>
    </xsl:for-each>
    </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Output:



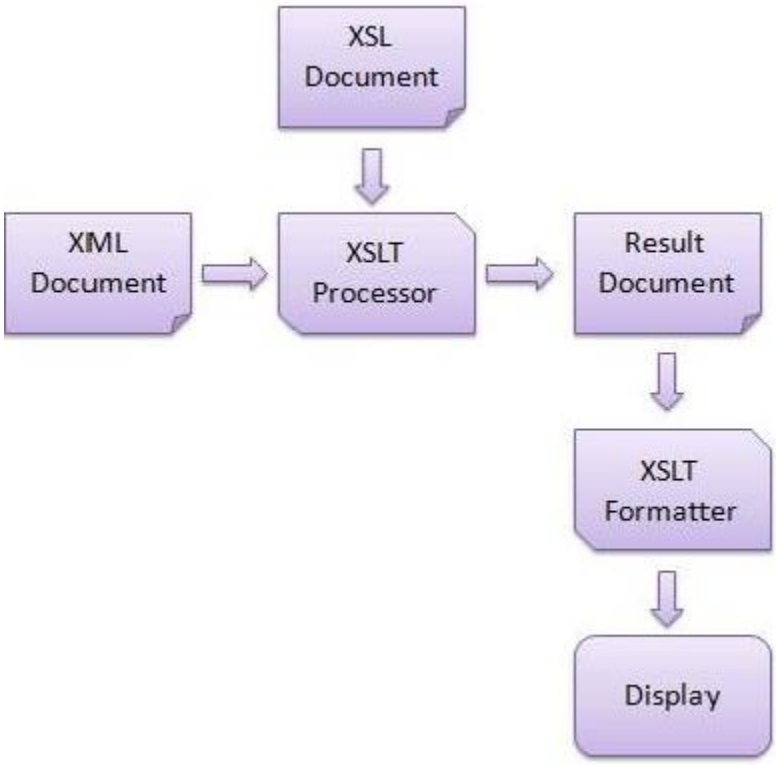
Students' Basic Details

Name	Branch	Age	City
Divyank Singh Sikarwar	CSE	18	Agra
Aniket Chauhan	CSE	20	Shahjahanpur
Simran Agarwal	CSE	23	Buland Shar
Abhay Chauhan	CSE	17	Shahjahanpur
Himanshu Bhatia	IT	25	Indore

XSLT Works:

- The XSLT style sheet is written in XML format.It is used to define the transformation rules to be applied on the target XML document.
- The XSLT processor takes the XSLT style sheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format.
- At the end it is used by XSLT formatter to generate the actual output and displayed on the end-user.

Image representation:



- XSLT provides an easy way to merge XML data into presentation because it applies user defined transformations to an XML document and the output can be HTML, XML, or any other structured document.
- XSLT provides Xpath to locate elements/attribute within an XML document. So it is more convenient way to traverse an XML document rather than a traditional way, by using scripting language.
- XSLT is template based. So it is more resilient to changes in documents than low level DOM and SAX.

- By using XML and XSLT, the application UI script will look clean and will be easier to maintain.
- XSLT templates are based on XPath pattern which is very powerful in terms of performance to process the XML document.
- XSLT can be used as a validation language as it uses tree-pattern-matching approach.
- You can change the output simply modifying the transformations in XSL files.

❖ **XML document :**

- An XML (Extensible Markup Language) Document contains declarations, elements, text, and attributes.
- It is made up of entities (storing units) and It tells us the structure of the data it refers to.It is used to provide a standard format of data transmission.
- As it helps in message delivery, it is not always stored physically, i.e. in a disk but generated dynamically but its structure always remains the same.

XML Standard structure and its rules:

Rule 1:

- Its standard format consists of an XML prolog which contains both XML Declaration and XML DTD (Document Type Definition) and the body.
- If the XML prolog is present, it should always be the beginning of the document. The XML Version, by default, is 1.0, and including only this forms the shortest XML Declaration.
- UTF-8 is the default character encoding and is one of seven character-encoding schemes. If it is not present, it can result in some encoding errors.

Syntax of XML Declaration:

<?xml version="1.0" encoding="UTF-8"?>

Syntax of DTD:

<!DOCTYPE root-element [<!element-declarations>]>

Example:

```
<!DOCTYPE website [
  <!ELEMENT website (name,company,phone)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT company (#PCDATA)>
  <!ELEMENT phone (#PCDATA)>
```

```
]>
<website>
  <name>GeeksforGeeks</name>
  <company>GeeksforGeeks</company>
  <phone>011-24567981</phone>
</website>
```

Rule 2:

- XML Documents must have a root element (the supreme parent element) and its child elements (sub-elements).
- To have a better view of the hierarchy of the data elements, XML follows the XML tree structure which comprises of one single root (parent) and multiple leaves (children).

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<website>
   <company category="geeksforgeeks">
      <title>Machine learning</title>
      <author>aarti majumdar</author>
      <year>2022</year>
   </company>
   <company category="geeksforgeeks">
      <title>Web Development</title>
       <author>aarti majumdar</author>
      <year>2022</year>
   </company>
   <company category="geeksforgeekse">
      <title>XML</title>
      <author>aarti majumdar</author>
      <year>2022</year>
   </company>
</website>
```

Rule 3: All XML Elements are required to have Closing and opening Tags(similar to HTML).

<message>Welcome to GeeksforGeeks</message>

Rule 4: The opening and closing tags are case-sensitive. For Example, <Message> is different from <message> from above example.

Rule 5: Values of XML attributes are required to have quotations:

```
<website category="open source">
   <company>geeksforgeeks</company>
</website>
```

Rule 6: White-Spaces are retained and maintained in XML.

<message>welcome to geeksforgeeks</message>

Rule 7: Comments can be defined in XML enclosed between <*!*– and –> tags.

<!-- XML Comments are defined like this -->

Rule 8: XML elements must be nested properly.

```
<message>
   <company>GeeksforGeeks</company>
</message>
```

❖ **XML schema**:

- XML schema is a language which is used for expressing constraint about XML documents. There are so many schema languages which are used now a days for example Relax- NG and XSD (XML schema definition).
- An XML schema is used to define the structure of an XML document. It is like DTD but provides more control on XML structure.
- XML schema defines the elements, attributes and data types. Schema element supports Namespaces. It is similar to a database schema that describes the data in a database

XML Schema Example:

employee.xsd

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.javatpoint.com"
xmlns="http://www.javatpoint.com"
elementFormDefault="qualified">
<xs:element name="employee">
```

```
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
      <xs:element name="email" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Let's see the xml file using XML schema or XSD file.

employee.xml

```
<?xml version="1.0"?>
<employee
xmlns="http://www.javatpoint.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.javatpoint.com employee.xsd">
  <firstname>vimal</firstname>
  <lastname>jaiswal</lastname>
  <email>vimal@javatpoint.com</email>
</employee>
```

Description of XML Schema:

<xs:element name="employee"> : It defines the element name employee.

<xs:complexType> : It defines that the element 'employee' is complex type.

<xs:sequence> : It defines that the complex type is a sequence of elements.

<xs:element name="firstname" type="xs:string"/> : It defines that the element 'firstname' is of string/text type.

<xs:element name="lastname" type="xs:string"/> : It defines that the element 'lastname' is of string/text type.

<xs:element name="email" type="xs:string"/> : It defines that the element 'email' is of string/text type.

XML Schema Data types:

There are two types of data types in XML schema.

1. simpleType
2. complexType

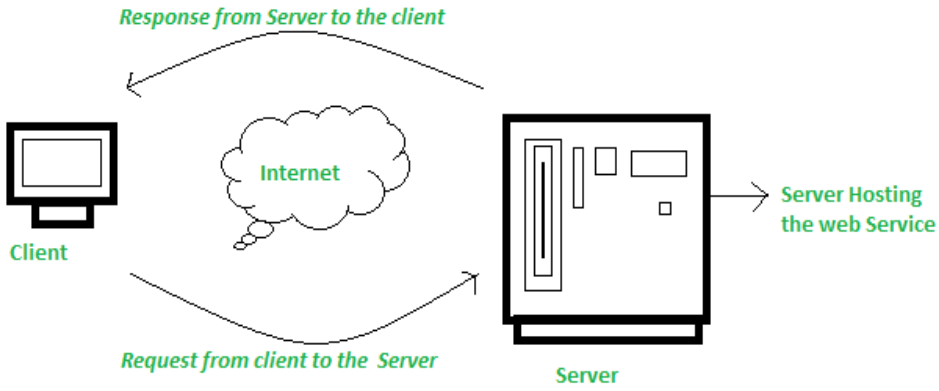
1.simpleType:

The simpleType allows you to have text-based elements. It contains less attributes, child elements, and cannot be left empty.

2.complexType:

The complexType allows you to hold multiple attributes and elements. It can contain additional sub elements and can be left empty.

- **Web Service**:
  - It is a client-server application or application component for communication.
  - The method of communication between two devices over the network.
  - It is a software system for the interoperable machine to machine communication.
  - It is a collection of standards or protocols for exchanging information between two devices or application.



Types of Web Services:

There are mainly two types of web services.

1. SOAP web services.
2. RESTful web services.

1.SOAP Web Services:

- SOAP stands for Simple Object Access Protocol. It is a XML-based protocol for accessing web services.
- SOAP is a W3C recommendation for communication between two applications. SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

Advantages of Soap Web Services:

WS Security: SOAP defines its own security known as WS Security.

Language and Platform independent: SOAP web services can be written in any programming language and executed in any platform.

Disadvantages of Soap Web Services:

Slow: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

WSDL dependent: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

2.RESTful web services:

- REST stands for Representational State Transfer. REST is an architectural style not a protocol.

Advantages of RESTful Web Services:

Fast: RESTful Web Services are fast because there is no strict specification like SOAP. It consumes less bandwidth and resource.

Language and Platform independent: RESTful web services can be written in any programming language and executed in any platform.

Can use SOAP: RESTful web services can use SOAP web services as the implementation.

Permits different data format: RESTful web service permits different data format such as Plain Text, HTML, XML and JSON.

Web Service Components:

There are three major web service components.

1. SOAP
2. WSDL

3. UDDI

1. SOAP (Simple Object Access Protocol):

- SOAP is stands for Simple Object Access Protocol.
- SOAP is a XML-based protocol for accessing web services.
- SOAP is XML based, so it is platform independent and language independent. In other words, it can be used with Java, .Net or PHP language on any platform.
- It is a transport-independent messaging protocol.
- SOAP is built on sending XML data in the form of SOAP Messages.
- A document known as an XML document is attached to each message.
- Only the structure of the XML document, not the content, follows a pattern.
- The best thing about Web services and SOAP is that everything is sent through HTTP, the standard web protocol.

2.WSDL (Web Services Description Language):

- WSDL is stands for Web Services Description Language.
- WSDL is a xml document containing information about web services such as method name, method parameter and how to access it.
- WSDL is a part of UDDI. It acts as a interface between web service applications.
- If a web service can't be found, it can't be used.
- The client invoking the web service should be aware of the location of the web service.
- Second, the client application must understand what the web service does in order to invoke the correct web service.
- The WSDL file is another XML-based file that explains what the web service does to the client application.
- The client application will be able to understand where the web service is located and how to use it by using the WSDL document.

3.UDDI (Universal Description, Discovery, and Integration):

- UDDI is stands for Universal Description, Discovery and Integration.
- UDDI is a XML based framework for describing, discovering and integrating web services.
- UDDI is a directory of web service interfaces described by WSDL, containing information about web services.

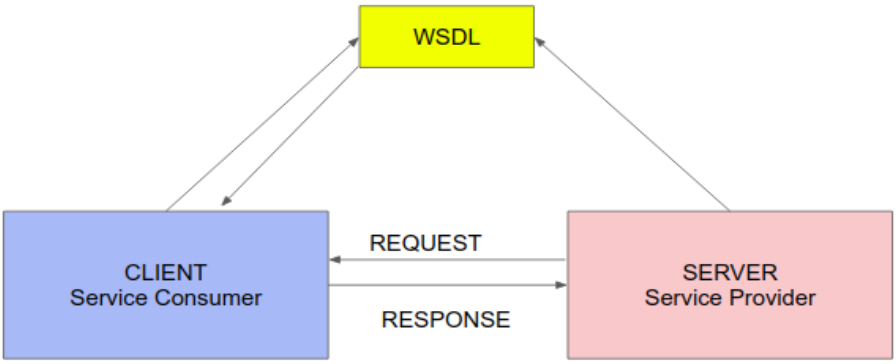
- UDDI is a standard for specifying, publishing and discovering a service provider's online services.
- UDDI provides a repository where WSDL files can be hosted so that a client application can discover a WSDL file to learn about the various actions that a web service offers.

Web Services Description Language (WSDL):

The Web Services Description Language (WSDL) forms the basis for the original Web Services specification. The following figure illustrates the use of WSDL. At the left is a service provider. At the right is a service consumer.

The steps involved in providing and consuming a service are:

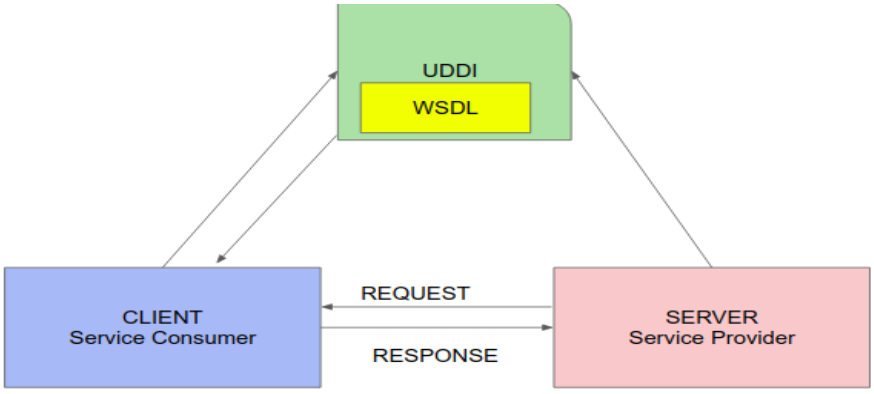
1. A service provider describes its service using WSDL. This definition is published to a repository of services. The repository could use Universal Description, Discovery, and Integration (UDDI). Other forms of directories could also be used.
2. A service consumer issues one or more queries to the repository to locate a service and determine how to communicate with that service.
3. Part of the WSDL provided by the service provider is passed to the service consumer. This tells the service consumer what the requests and responses are for the service provider.
4. The service consumer uses the WSDL to send a request to the service provider.
5. The service provider provides the expected response to the service consumer.



- Now finally we can say WSDL is an interface for the web service and the service provider creates this interface for his web service and service consumer to get this WSDL and use all the web services.

Universal Description, Discovery, and Integration (UDDI):

- Now there are two ways the Service consumer can get the WSDL document.
- One way is if the service consumer and the service provider all know each other then the service provider can handover directly the WSDL document or the WSDL URL to the client and then the client can use this web service.
- To enable this all the web services provides publishes their description of their web services to a WSDL to an online registry or a directory that can be searched by any consumer and can get the whole of their WSDL. And this online registry or directory is called UDDI.



- UDDI stands for Universal Description Discovery and Integration(UDDI)**.** And it is an online registry where a service provider can publish his WSDL and consumer can query and get the WSDL.
- Then once a consumer has the WSDL and they can make use of the web service. This is all about the WSDL and UDDI at a very basic level.

❖ **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

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

2. Implement the SEI with java class

```
import javax.jws.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.

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

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

❖ **Web resources:**

- Use XML Web resources to cache configuration settings or metadata for your solutions.
- An XML Web resource doesn't represent a robust solution for data multiple users frequently update.
- When one user is updating an XML Web resource, another user (or automated process) could update the Web resource. Data added by the second user is lost when the first user saves their changes.
- All XML files must use the .xml file name extension. Files that use XML data but don't use the .xml file name extension can't be uploaded as Web resources unless the file name extension is changed.

  Example:

```
<security-constraint>
   <web-resource-collection>
      <web-resource-name>profile</web-resource-name>
      <url-pattern>/profile/*</url-pattern>
   </web-resource-collection>
   <user-data-constraint>
      <transport-guarantee>CONFIDENTIAL</transport-guarantee>
   </user-data-constraint>
</security-constraint>
```