Tuesday
16-6-2020

WT Assignment-5

UNIT-5

N.V.S.K.kalyani
174N1A0584
3rd B.Tech CSE-C

2 Marks

1A) A web service is a unit of managed code that can be remotely invoked using HTTP. That is, it can be activated using HTTP requests. Web services allow you to expose the functionality of your existing code over the network. once it is exposed on the network, other applications can use the functionality of your program.

2A) Components of WSDL:-

1. <types> - Defines the (XML schema) data types used by the web services.
2. <message> - Defines the data elements for each operation.
3. <portType> - Defines the operations that can be performed & the messages involved.
4. <binding> - Defines the protocol and data format for each port type.

3A) Technologies used by ajax:-

- HTML/XHTML and CSS.
- DOM
- XML or JSON
- ASP or JSP
- XMLHttp Request
- Javascript

4A) jax-ws:- jax-ws stands for Java API for XML web services. jax-ws is xml based java API to build web services, server and client application. Its part of standard Java API, so we don't need to include anything else which working with it.

5A) UDDI:- universal Description, Discovery & Integration (UDDI) specifications define a registry service for web services and for other electronic & non-electronic services. A UDDI registry service is a web service that manages information about service provides, service implementations & service metadata.

10 Marks

1A) The AJAX Framework is a cross browser framework that allows developers to quickly develop web pages that can call web services, web pages and other types of content through Javascript without having to submit the current page.

The AJAX Framework is :-

- Easy to use.
- work well with other existing frameworks.
- supports both "GET" and "POST".
- works with Internet Explorer 6+, opera 8+, safari 3+ Firefox 2+, Google's chrome and other Mozilla based browser.

Example:-

call page

```
function pagecallback(response) {
    alert(response);
}
var ajax=new GLM.AJAX();
ajax.callpage("mypage.html", pagecallback);
```

servicecall Example:-

```
function servicecallback (response) {
    alert(response);
}
var ajax = new GLM.AJAX();
ajax.callservice("MyWebService.asmx","MyMethodTo
                call", servicecallback);
```

Servicecall with parameters Ex:-

```
function servicecallback(response) {
    alert(response);
}
var ajax = new GLM.AJAX();
```

error handling Ex:-

```
function myErrorHandler (error){
    alert(error);
}
var ajax = new GLM.AJAX();
ajax.onError = myErrorHandler;
```

24) WSDL:- It is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations & messages are described abstractly, and then bound to a concrete network protocol & message format to define an endpoint.

Structure:-

```
<definitions>

<types>
   data type definitions....
</types>
<message>
   definition of the data being communicated....
</message>
<portType>
    set of operations....
</portType>
<binding>
   protocol and data format specifications....
</binding>
</definitions>
```

Example:-

```
<definitions name = "HelloService"
  targetNamespace = "http://www.examples.com/wsdl/
                     HelloService.wsdl"
  xmlns = "http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap = "http://schemas.xmlsoap.org/wsdl/
                soap/"
  xmlns:tns = "http://www.examples.com/wsdl/Hello
               Service.wsdl"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchem
              >

<message name = "SayHelloRequest">
  <part name = "firstName" type = "xsd:string"/>
</message>
<message name = "SayHelloResponse">
  <part name = "greeting" type = "xsd.string"/>
</message>
<portType name = "Hello_PortType">
  <operation name = "sayHello">
    <Input message = "tns:SayHelloRequest"/>
    <output message = "tns:SayHelloResponse"/>
  </operation>
</portType>
<binding name = "Hello_Binding" type = "tns:Hello_portTy
  <soap:binding style = "rpc"
    transport = "http://schemas.xmlsoap.org/soap/
                http"/>
  <operation name = "SayHello">
```

```
<soap:operation soapAction="sayHello"/>
<input
  <soap:body
    encodingstyle:"http://schemas.xmlsoap.org/
                   soap/encoding/"
    namespace = "urn:example:helloservice"
    use = "encoded"/>
</input>
<output>
  <soap:body
    encodingstyle:"http://schemas.xmlsoap.org/
                   soap/encoding/"
    namespace = "urn:examples:helloservice"
    use = "encoded"/>
  </output>
 </operation>
</binding>
<service name = "Hello_service">
 <documentation>wsdl file for Helloservice
  </documentation>
 <port binding = "tns:Hello_Binding" name = "Hello_port">
  <soap:address
    location = "http://www.examples.com/sayHello"/>
 </port>
</service>
</definitions>
```