

Class:M.Tech(CSE) I Year-II Sem

Faculty:K.Yellaswamy

Date:03.07.2015

(B1532) WEB SERVICES LAB

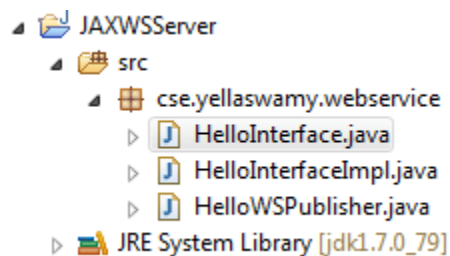
1. Write a Program to implement WSDL Service (Hello Service,WSDL File)

Using JAX-WS Web Service.we will see how we can develop JAX-WS endpoint and client step by step.

Steps for creating JAX-WS Webservice Endpoint:

- I. Open Eclipse Kepler IDE
- II. Create java project name "JAXWSServer"
- III. Create a new package name
`cse.yellaswamy.webservice`
- IV. Create JAXWSService Endpoint Interface
`HelloInterface.java`
- V. Create JAXWSService Endpoint implementation class
`HelloInterfaceImpl.java`
- VI. Create Endpoint Publisher
`HelloWSPublisher.java`

The following is the Directory Structure for above Steps:





```
// HelloInterface.java
package cse.yellaswamy.webservice;
```

```
import javax.jws.WebMethod;
import javax.jws.WebService;
```

```
@WebService
public interface HelloInterface
{
```

```
@WebMethod
public String sayHello(String name);
}
```

```
// HelloInterfaceImpl.java
package cse.yellaswamy.webservice;
```

```
import javax.jws.WebService;
```

```
@WebService(endpointInterface="cse.yellaswamy.webservice.HelloInterface")
public class HelloInterfaceImpl implements HelloInterface
{
```

```
    public String sayHello(String name) {
        // TODO Auto-generated method stub
        return "Hello From Webservice"+name;
    }
```

```
}
```

```
// HelloWorldPublisher.java

package cse.yellaswamy.webservice;

import javax.xml.ws.Endpoint;

public class HelloWorldPublisher
{
    public static void main(String args[])
    {

        Endpoint.publish("http://KANDULA:9090/cmrhellows/HelloInterface", new
HelloInterfaceImpl());
    }
}
/*
 * Run above program.Your webservice is published.You can check your service wsdl at
http://KANDULA:9090/cmrhellows/HelloInterface?wsdl
 */
```

URL: <http://kandula:9090/cmrhellows/HelloInterface?wsdl>



The screenshot shows a web browser window with the address bar containing the URL <http://kandula:9090/cmrhellows/HelloInterface?wsdl>. The browser's address bar also shows several tabs, including "Pearson - PowerPoi...", "Digital Cabel Net: L...", "Javatpoint - A Soluti...", "FOOD SECURITY - ...", "LoginPage", "Goddess Sri Nalla P...", "Listen to Edikonda S...", and "Research Programm...". Below the address bar, a message states: "This XML file does not appear to have any style information associated with it. The document tree is shown below." The XML document tree is displayed with a tree view on the left. The root element is a service named "HelloInterfaceImplService". The tree shows the following structure:

- <!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01. -->
- <!-- Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is JAX-WS RI 2.2.4-b01. -->
- <definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://webservice.yellaswamy.cse/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://webservice.yellaswamy.cse/" name="HelloInterfaceImplService">
- <types>...</types>
- <message name="sayHello">...</message>
- <message name="sayHelloResponse">...</message>
- <portType name="HelloInterface">...</portType>
- <binding name="HelloInterfaceImplPortBinding" type="tns:HelloInterface">
- <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
- <operation name="sayHello">
- <soap:operation soapAction="">
- <input>
- <soap:body use="literal"/>
- </input>
- <output>
- <soap:body use="literal"/>
- </output>
- </operation>
- </binding>
- </service name="HelloInterfaceImplService">

Steps for Creating JAXWS Client

- I. Open eclipse and create a new java project

JAXWSClient

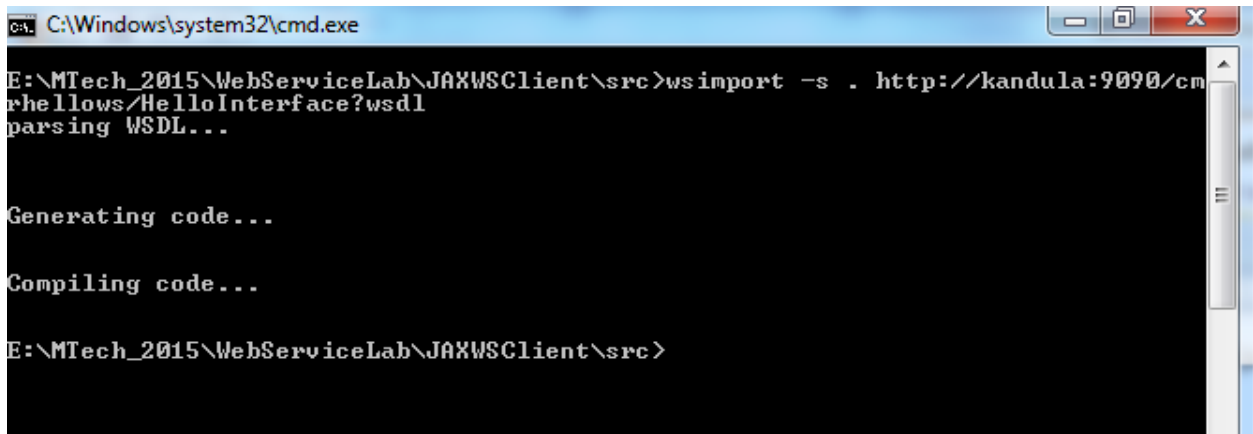
- II. Now we need to generate the client stubs so open your command line and enter wsimport command.

```
E:\MTech_2015\WebServiceLab\JAXWSClient\src>wsimport -s . http://kandula:9090/cmrhellows/HelloInterface?wsdl
```

parsing WSDL...

Generating code...

Compiling code...



```
C:\Windows\system32\cmd.exe
E:\MTech_2015\WebServiceLab\JAXWSClient\src>wsimport -s . http://kandula:9090/cmrhellows/HelloInterface?wsdl
parsing WSDL...

Generating code...

Compiling code...

E:\MTech_2015\WebServiceLab\JAXWSClient\src>
```

- III. Lets create client class now

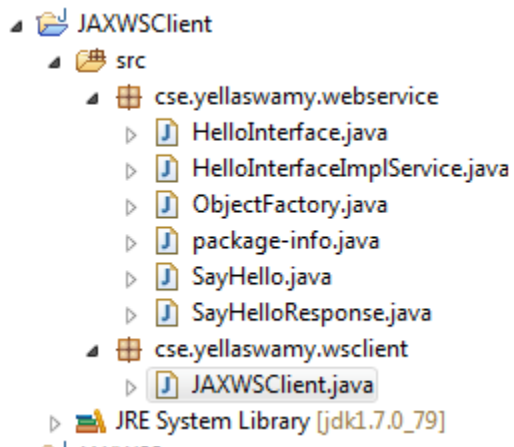
```
// JAXWSClient.java
```

```
package cse.yellaswamy.wsclient;
import cse.yellaswamy.webservice.HelloInterface;
import cse.yellaswamy.webservice.HelloInterfaceImplService;
public class JAXWSClient {

    public static void main(String[] args)
    {
        // TODO Auto-generated method stub
    }
}
```

```
HelloInterfaceImplService helloService = new HelloInterfaceImplService();  
HelloInterface hello = helloService.getHelloInterfaceImplPort();  
System.out.println(hello.sayHello("KANDULA ASHOK"));  
  
    }  
  
}
```

Directory Structure for Client:

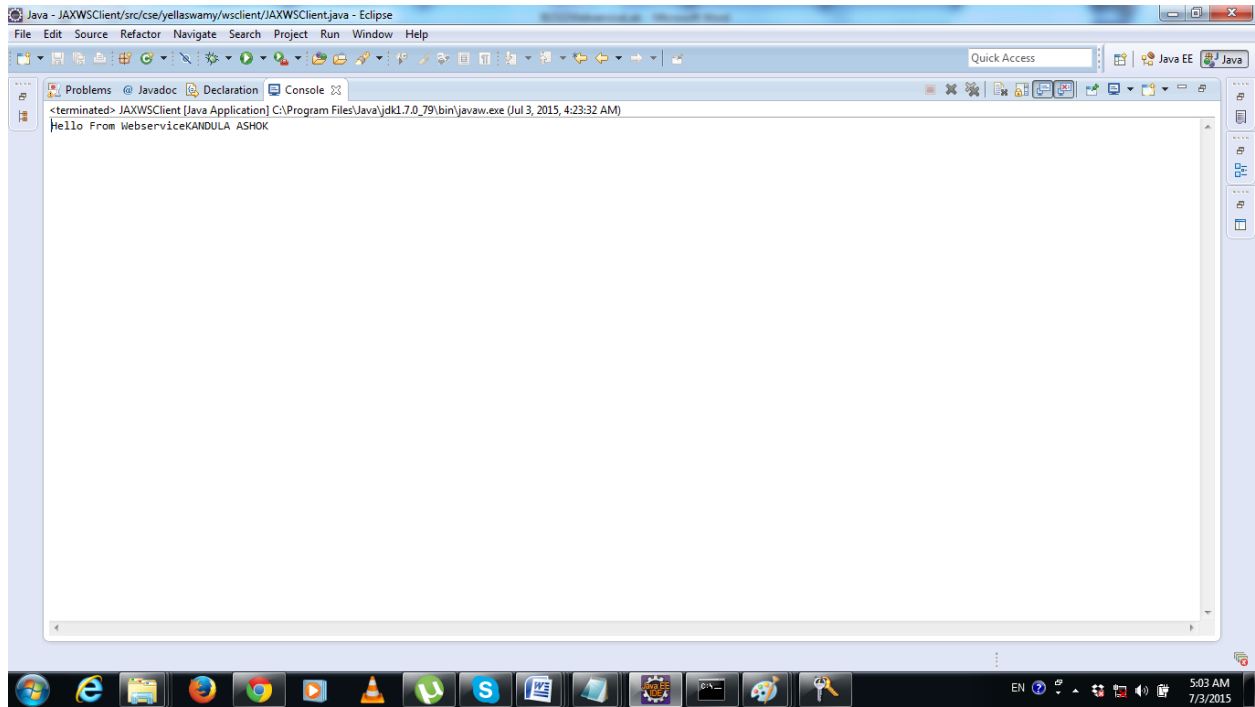


Output:



CMR College of Engineering & Technology

Department of Computer Science & Engineering

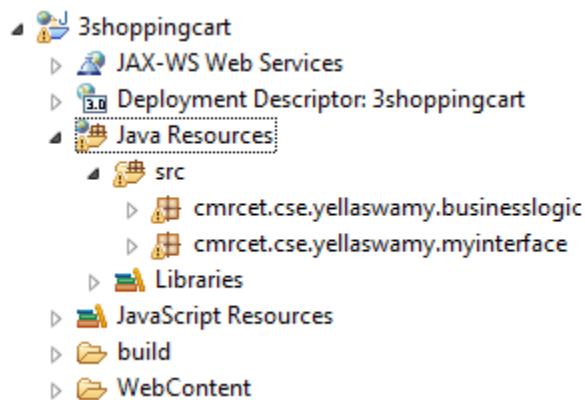


2. Write a Program the service provider can implement a single `getPrice()`, `static bind()` and get product operation

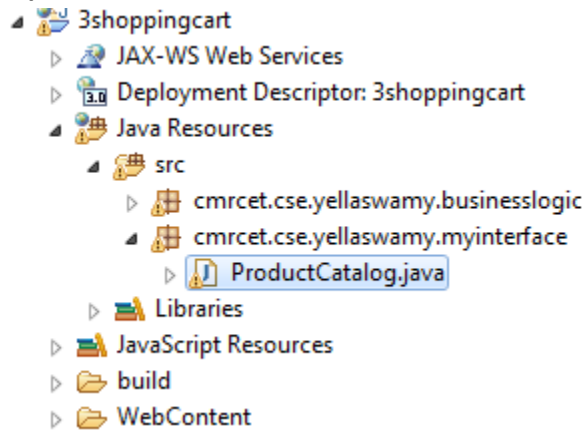
Steps:

- I. Create a Dynamic Web Application(i.e.,3shoppingcart)
- II. Create the following packages
 - i. `cmrcet.cse.yellaswamy.myinterface;`
 - ii. `cmrcet.cse.yellaswamy.businesslogic;`
- III. Write a interface in package `cmrcet.cse.yellaswamy.myinterface` (i.e .,ProductCatalog.java)
- IV. Write Implementation class for above interface in package `cmrcet.cse.yellaswamy.businesslogic` (i.e., ProductServiceImpl.java)
- V. Check the Directory Structure of WebApplication
- VI. Deploy in Glassfish Server
- VII. Test the Web Service

Step I&II:



Step III:



```
// ProductCatalog.java
```

```
package cmrcet.cse.yellaswamy.myinterface;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```
import javax.ws.WebMethod;  
import javax.ws.WebService;
```

```
import cmrcet.cse.yellaswamy.businesslogic.ProductServiceImpl;  
@WebService  
public class ProductCatalog  
{
```

```
    ProductServiceImpl productservice=new ProductServiceImpl();
```

```
    @WebMethod  
    public List<String> getProductCategories()  
    {  
        return productservice.getProductCategories() ;  
    }
```

```
    @WebMethod  
    public List<String> getProducts(String mychoice)  
    {  
        return productservice.getProducts(mychoice);  
    }
```

```
    @WebMethod  
    public boolean addProduct(String mychoice,String product)  
    {  
        return productservice.addProduct(mychoice,product);  
    }
```

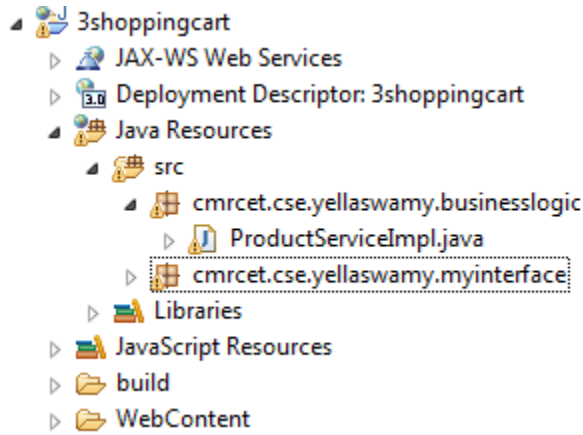



CMR College of Engineering & Technology

Department of Computer Science & Engineering

```
@WebMethod(operationName="BookPrice")
public double getBookPrice(String isbn)
{
    return productservice.getBookPrice(isbn);
}
}
```

Step IV:



```
//ProductServiceImpl.java
```

```
package cmrcet.cse.yellaswamy.businesslogic;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import javax.jws.WebMethod;
```

```
public class ProductServiceImpl
```

```
{
```

```
    //step4
```

```
    List<String> csebooklist=new ArrayList<>();
```

```
    List<String> ecebooklist=new ArrayList<>();
```

```
    List<String> eebooklist=new ArrayList<>();
```

```
    //step5 constructor
```

```
    public ProductServiceImpl()
```

```
    {
```

```
        //CSE BOOKS
```

```
        csebooklist.add("Thinking on the Web- Berners Lee");
```

```
        csebooklist.add("Web Services & SOA Principles and Technology-
```

```
P.Papazoglou");
```

```
        csebooklist.add("Developing Java Web Services-R.Nagappan");
```

```
        //ECE BOOKS
```

```
        ecebooklist.add("ece-book1");
```

```
        ecebooklist.add("ece-book2");
```

```
        ecebooklist.add("ece-book3");
```

```
//EEE BOOKS
eeebooklist.add("eee-book1");
eeebooklist.add("eee-book2");
eeebooklist.add("eee-book3");

}

public List<String> getProductCategories()
{
    List<String> categorories=new ArrayList<>();
    categorories.add("cse");
    categorories.add("ece");
    categorories.add("eee");
    return categorories;
}

public List<String> getProducts(String mychoice)
{

    switch(mychoice.toLowerCase())
    {
    case "cse":
        return csebooklist;
    case "ece":
        return ecebooklist;
    case "eee":
        return eeebooklist;

    }
    return null;
}

//Add Products Dynamically

public boolean addProduct(String mychoice1,String product)
{

    switch(mychoice1.toLowerCase())
    {
    case "cse":
        csebooklist.add(product);
        break;
    case "ece":
        ecebooklist.add(product);
        break;
    case "eee":
        eeebooklist.add(product);
    }
}
```

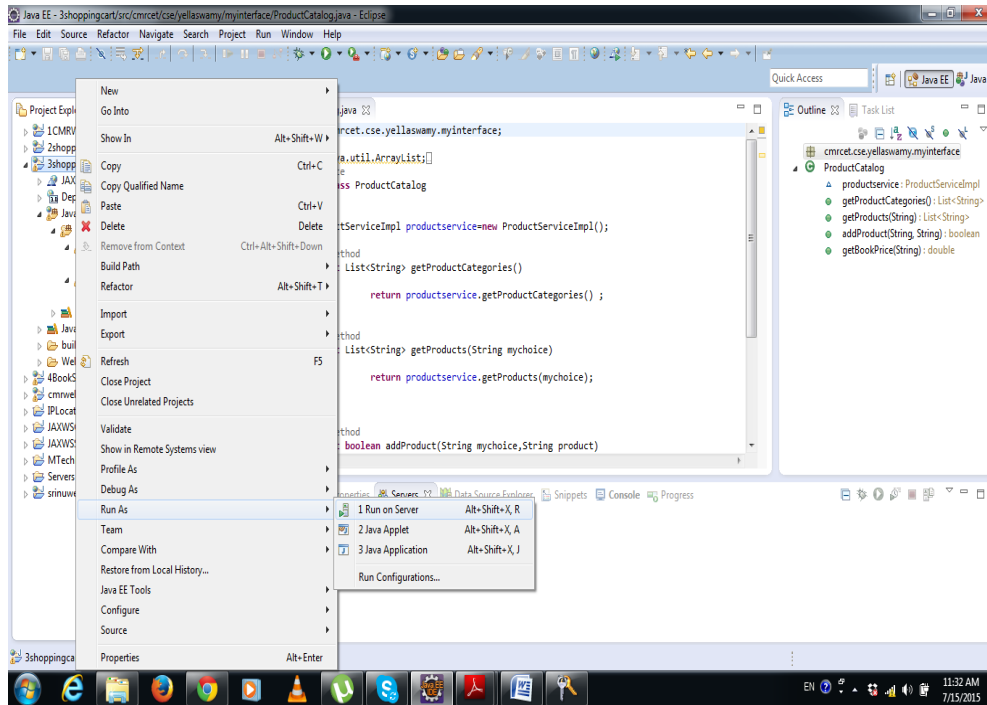
```
        break;
        default:
            return false;
    }
    return true;
}

public double getBookPrice(String isbn)
{
    // TODO Auto-generated method stub
    if("IS101".equalsIgnoreCase(isbn))
        return 230.29;
    else if("IS102".equalsIgnoreCase(isbn))
        return 459.56;
    else
        return 0;
}
}
```

Step V: Final Directory Structure

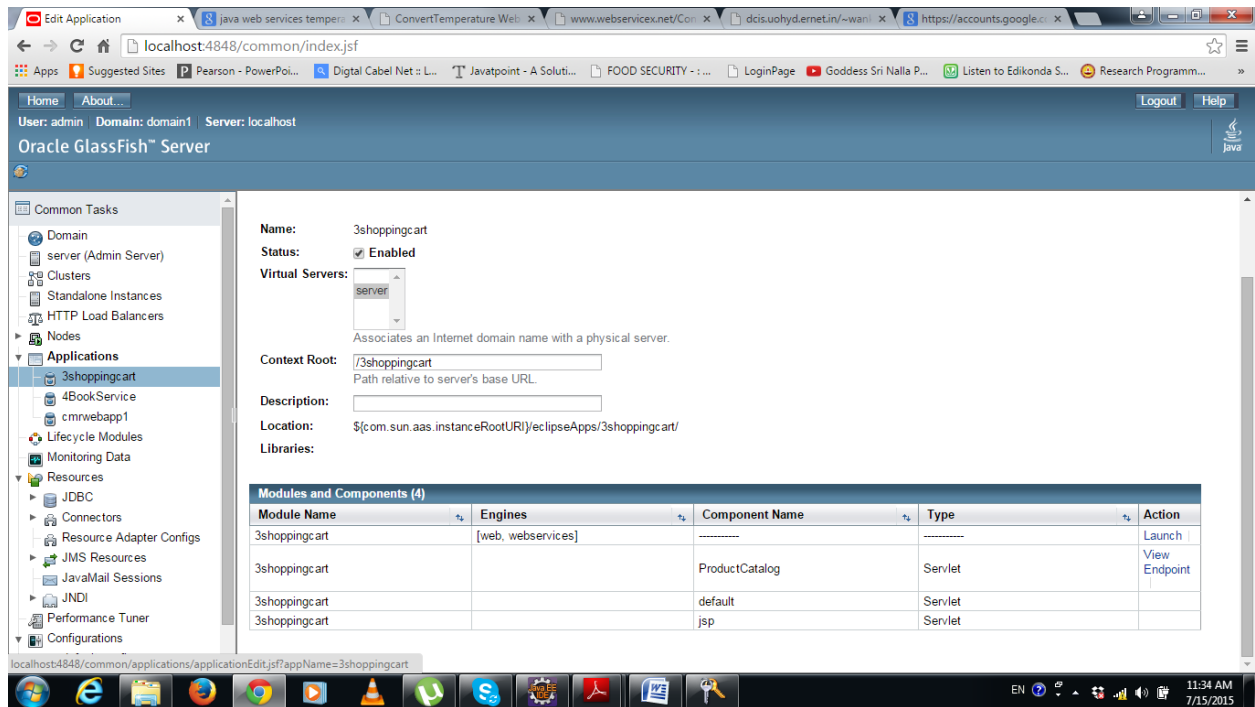
- └─ 3shoppingcart
 - └─ JAX-WS Web Services
 - └─ Deployment Descriptor: 3shoppingcart
 - └─ Java Resources
 - └─ src
 - └─ cmrcet.cse.yellaswamy.businesslogic
 - └─ ProductServiceImpl.java
 - └─ cmrcet.cse.yellaswamy.myinterface
 - └─ ProductCatalog.java
 - └─ Libraries
 - └─ JavaScript Resources
 - └─ build
 - └─ WebContent

Step VI: Deploy and Test with Glass Fish Application Server



<http://localhost:4848/>





Module Name	Engines	Component Name	Type	Action
3shoppingcart	[web, webservices]	-----	-----	Launch
3shoppingcart		ProductCatalog	Servlet	View Endpoint
3shoppingcart		default	Servlet	
3shoppingcart		jsp	Servlet	

<http://kandula:8181/3shoppingcart/ProductCatalogService?wsdl>

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
  Metro/2.2.0-1 (tags/2.2.0u1-7139; 2012-06-02T10:55:19+0000) JAXWS-RI/2.2.6-2
  JAXWS/2.2 svn-revision#unknown.
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is
  Metro/2.2.0-1 (tags/2.2.0u1-7139; 2012-06-02T10:55:19+0000) JAXWS-RI/2.2.6-2
  JAXWS/2.2 svn-revision#unknown.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-
policy"xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam
="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.x
mlsoap.org/wsdl/soap/"xmlns:tns="http://myinterface.yellaswamy.cse.cmrcet/" x
mlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org
/wSDL/"targetNamespace="http://myinterface.yellaswamy.cse.cmrcet/" name="Prod
uctCatalogService">
<types>
<xsd:schema>
<xsd:import namespace="http://myinterface.yellaswamy.cse.cmrcet/" schemaLocat
ion="http://kandula:8181/3shoppingcart/ProductCatalogService?xsd=1"/>
</xsd:schema>
</types>
<message name="getProductCategories">
```



CMR College of Engineering & Technology

Department of Computer Science & Engineering

```
<part name="parameters" element="tns:getProductCategories"/>
</message>
<message name="getProductCategoriesResponse">
<part name="parameters" element="tns:getProductCategoriesResponse"/>
</message>
<message name="BookPrice">
<part name="parameters" element="tns:BookPrice"/>
</message>
<message name="BookPriceResponse">
<part name="parameters" element="tns:BookPriceResponse"/>
</message>
<message name="getProducts">
<part name="parameters" element="tns:getProducts"/>
</message>
<message name="getProductsResponse">
<part name="parameters" element="tns:getProductsResponse"/>
</message>
<message name="addProduct">
<part name="parameters" element="tns:addProduct"/>
</message>
<message name="addProductResponse">
<part name="parameters" element="tns:addProductResponse"/>
</message>
<portType name="ProductCatalog">
<operation name="getProductCategories">
<input wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/getProductCategoriesRequest" message="tns:getProductCategories"/>
<output wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/getProductCategoriesResponse" message="tns:getProductCategoriesResponse"/>
</operation>
<operation name="BookPrice">
<input wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/BookPriceRequest" message="tns:BookPrice"/>
<output wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/BookPriceResponse" message="tns:BookPriceResponse"/>
</operation>
<operation name="getProducts">
<input wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/getProductsRequest" message="tns:getProducts"/>
<output wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/getProductsResponse" message="tns:getProductsResponse"/>
</operation>
<operation name="addProduct">
<input wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/addProductRequest" message="tns:addProduct"/>
<output wsam:Action="http://myinterface.yellaswamy.cse.cmrctet/ProductCatalog/addProductResponse" message="tns:addProductResponse"/>
</operation>
</portType>
<binding name="ProductCatalogPortBinding" type="tns:ProductCatalog">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
<operation name="getProductCategories">
<soap:operation soapAction=""/>
<input>
```



CMR College of Engineering & Technology

Department of Computer Science & Engineering

```
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="BookPrice">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="getProducts">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
<operation name="addProduct">
<soap:operation soapAction=""/>
<input>
<soap:body use="literal"/>
</input>
<output>
<soap:body use="literal"/>
</output>
</operation>
</binding>
<service name="ProductCatalogService">
<port name="ProductCatalogPort" binding="tns:ProductCatalogPortBinding">
<soap:address location="http://kandula:8181/3shoppingcart/ProductCatalogService"/>
</port>
</service>
</definitions>
```

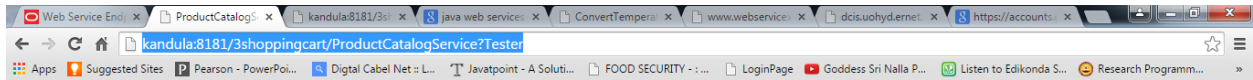



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!--
  Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.2.0-1 (tags/2.2.0u1-7139; 2012-06-02T10:55:19+0000) JAXWS-RI/2.2.6-2 JAXWS/2.2 svn-revision#unknown.
-->
<!--
  Generated by JAX-WS RI at http://jax-ws.dev.java.net. RI's version is Metro/2.2.0-1 (tags/2.2.0u1-7139; 2012-06-02T10:55:19+0000) JAXWS-RI/2.2.6-2 JAXWS/2.2 svn-revision#unknown.
-->
<definitions xmlns:wsu="http://docs.oasis-open.org/ws/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://myinterface.yellaswamy.cse.cmrcet/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  targetNamespace="http://myinterface.yellaswamy.cse.cmrcet/" name="ProductCatalogService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://myinterface.yellaswamy.cse.cmrcet/" schemaLocation="http://kandula:8181/3shoppingcart/ProductCatalogService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="getProductCategories">
    <part name="parameters" element="tns:getProductCategories"/>
  </message>
  <message name="getProductCategoriesResponse">
    <part name="parameters" element="tns:getProductCategoriesResponse"/>
  </message>
  <message name="BookPrice">
    <part name="parameters" element="tns:BookPrice"/>
  </message>
  <message name="BookPriceResponse">
    <part name="parameters" element="tns:BookPriceResponse"/>
  </message>
  <message name="getProducts">
    <part name="parameters" element="tns:getProducts"/>
  </message>
  <message name="getProductsResponse">
    <part name="parameters" element="tns:getProductsResponse"/>
  </message>
  <message name="addProduct">
    <part name="parameters" element="tns:addProduct"/>
  </message>
  <message name="addProductResponse">
    <part name="parameters" element="tns:addProductResponse"/>
  </message>
</definitions>
  
```

<http://kandula:8181/3shoppingcart/ProductCatalogService?Tester>



ProductCatalogService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract java.util.List cmrcet.cse.yellaswamy.myinterface.ProductCatalog.getProductCategories()
getProductCategories ( )
```

```
public abstract java.util.List cmrcet.cse.yellaswamy.myinterface.ProductCatalog.getProducts(java.lang.String)
getProducts ( )
```

```
public abstract boolean cmrcet.cse.yellaswamy.myinterface.ProductCatalog.addProduct(java.lang.String,java.lang.String)
addProduct ( )
```

```
public abstract double cmrcet.cse.yellaswamy.myinterface.ProductCatalog.bookPrice(java.lang.String)
bookPrice ( )
```



getProductCategories Method invocation



Method parameter(s)

Type	Value
------	-------

Method returned

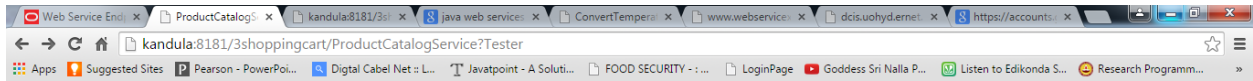
java.util.List : "[cse, ece, eee]"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:getProductCategories
xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/" />
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getProductCategoriesResponse
xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/">
      <return>cse</return>
      <return>ece</return>
      <return>eee</return>
    </ns2:getProductCategoriesResponse>
  </S:Body>
</S:Envelope>
```



ProductCatalogService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

```
public abstract java.util.List cmrcet.cse.yellawamy.myinterface.ProductCatalog.getProductCategories()  
getProductCategories ( )
```

```
public abstract java.util.List cmrcet.cse.yellawamy.myinterface.ProductCatalog.getProducts(java.lang.String)  
getProducts (cse )
```

```
public abstract boolean cmrcet.cse.yellawamy.myinterface.ProductCatalog.addProduct(java.lang.String,java.lang.String)  
addProduct ( )
```

```
public abstract double cmrcet.cse.yellawamy.myinterface.ProductCatalog.bookPrice(java.lang.String)  
bookPrice ( )
```



getProducts Method invocation

Method parameter(s)

Type	Value
java.lang.String	cse

Method returned

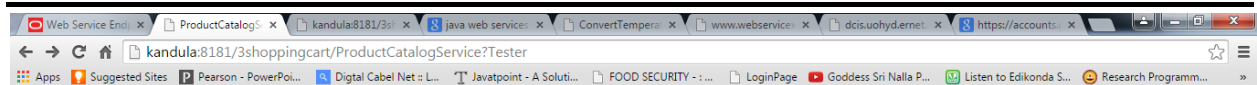
java.util.List : "[Thinking on the Web- Berners Lee, Web Services & SOA Principles and Technology-P.Papazoglou, Developing Java Web Services-R.Nagappan]"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:getProducts
xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/">
      <arg0>cse</arg0>
    </ns2:getProducts>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:getProductsResponse
xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/">
      <return>Thinking on the Web- Berners Lee</return>
      <return>Web Services & SOA Principles and Technology-
P.Papazoglou</return>
      <return>Developing Java Web Services-R.Nagappan</return>
    </ns2:getProductsResponse>
  </S:Body>
</S:Envelope>
```



ProductCatalogService Web Service Tester

This form will allow you to test your web service implementation ([WSDL File](#))

To invoke an operation, fill the method parameter(s) input boxes and click on the button labeled with the method name.

Methods :

public abstract java.util.List cmrcet.cse.yellaswamy.myinterface.ProductCatalog.getProductCategories()
getProductCategories ()

public abstract java.util.List cmrcet.cse.yellaswamy.myinterface.ProductCatalog.getProducts(java.lang.String)
getProducts (cse)

public abstract boolean cmrcet.cse.yellaswamy.myinterface.ProductCatalog.addProduct(java.lang.String,java.lang.String)
addProduct ()

public abstract double cmrcet.cse.yellaswamy.myinterface.ProductCatalog.bookPrice(java.lang.String)
bookPrice (S101)



bookPrice Method invocation

Method parameter(s)

Type	Value
java.lang.String	IS101

Method returned

double : "230.29"

SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <ns2:BookPrice xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/">
      <arg0>IS101</arg0>
    </ns2:BookPrice>
  </S:Body>
</S:Envelope>
```

SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:BookPriceResponse
xmlns:ns2="http://myinterface.yellaswamy.cse.cmrcet/">
      <return>230.29</return>
    </ns2:BookPriceResponse>
  </S:Body>
</S:Envelope>
```
