DEVELOPING CONTRACT - DRIVEN WEB SERVICES USING JDEVELOPER

Purpose:
The purpose of this tutorial is to develop a java web service using a top-down approach.

Topics:
This tutorial covers the following topics:
1. Overview
2. Prerequisites
3. Creating the WSDL contract document
4. Generating the web service from the WSDL contract document
5. Testing the web service

Overview:
In this tutorial we make use of JDeveloper’s tools for end to end contract driven development, including XML schema creation, WSDL editing, top-down web service generation and client creation and testing.

Prerequisites:
Before we start this tutorial all we need to have is:
1. The oracle JDeveloper must have been installed.
2. We must have started JDeveloper by double-clicking <jdev_home>jdeveloper.exe.
3. Start oracle application server containers for J2EE (OC4J)
4. And have created a connection in JDeveloper to OC4J.

The first step in this tutorial is to create the WSDL contract document and this includes the following steps:
1. Creating a schema
2. Creating a WSDL contract document
3. Defining the service interface
4. Defining the service implementation

First, let us create a schema and then import the schema into a WSDL document.

1. Click on the “Application Navigator”. On the Application, right click and choose the option “New Application”.

![Application Navigator](image)

- **New Application**: Ctrl-N
- **Add to Applications**: Ctrl+Shift-F9
- **Rebuild**: Alt+Shift+F9
- **Reformat**: Ctrl+Alt-L
- **Organize Imports**: Ctrl+Alt-O
2. Enter the “Top Down contract development” as the application name in the create application dialog. And click OK.
3. And enter the project name as WSDL document in the create project dialog and select new from the context menu by right clicking the WSDL document.

4. Expand **General** in the categories and then select XML then select XML schema. Here, in create XML schema dialog enter the name as CreditRating.xsd. The schema will be created and opened in the editor.

5. And now the following URL would be entered over the example name.

```
“http://myCompany.com/CreditRating/types”
```

6. Now lets begin adding 3 elements namely- ssn, rating and error to the schema. And also set type for each of these.

Click the example element and then enter the name as ssn and set type as xsd: string by right clicking it. The following snap shot appears. This includes even the second element rating.
7. Similarly set the type for element rating as xsd: int and include the third element as error and set its type as xsd: string using the same procedure. The following screen shot appears with 3 elements.

This finishes creation of the schema.

Now let us start creating the WSDL contract document.
1. Now, on right clicking the WSDL document and choosing new, choose web services by expanding the business tier node and select WSDL document.

```
<schema>
targetNamespace http://mycompany.com/CreditRating
</schema>
```

2. And now enter the WSDL name as Credit Rating and enter the target namespace as follows “http://myCompany.com/CreditRating and then click OK.
3. The CreditRating WSDL will be created and the following screen shot appears in the Design tab.
4. Right click the types in the structure window and choose insert types and then choose XML schema in the insert inside types.
5. Similarly choose insert inside schema by right clicking the schema sub node and then choose import.

6. In the property inspector specify the schemaLocation, id and namespace.
7. In the source tab, right click and choose Reformat in order to make the code clear.

Now let us define the service interface.

1. First we need to add 3 messages.

In the design tab, on clicking the switch view you will get to left most position. Then, under the design tab, right click the messages in order to insert message.

2. Enter the name as CreditRatingRequestMessage. and then click OK.
3. By right clicking the CreditRatingRequestMessage, choose insert inside message CreditRatingRequestMessage and then select part.

4. Now enter part name as ssn and enter the part type as types:ssn.
5. Add 2 more messages CreditRatingResponseMessage and CreditRatingFaultMessage and their part types.
6. Now right click PortTypes and select Insert portType. and then enter the port type name as CreditRating and click OK.

7. Select operation by right clicking on the port type to select insert inside portType-CreditRating.

8. Set the operation name to processCreditRating and input and output to CreditRatingRequestMessage and CreditRatingResponseMessage and fault name to NegativeCredit. The following screen shot appears.
9. After expanding the Creditrating we see that the messages will be associated to input, output and the fault message. The following screen shot shows that input is associated with ssn.

10. Now we need to define the bindings by choosing insert binding, from the structure window (on right clicking the Binding we will obtain this). Enter the Binding name as CreditRatingSoapHttp.
11. After creating the binding we will obtain the following. This shows relationship to other elements when you click on it.

12. Now we need to change the name to "NegativeCredit" after clicking the source tab, in the NegativeCredit fault definition.
Now we need to define the Service implementation.

1. Right click on the services and select insert service in the design tab of structure window and name that to CreditRatingService.

2. Now, by right clicking the service-CreditRatingService and choosing http://schemas.xmlsoap.org/wsdl/>port, we will obtain the port. And enter the portname as CreditRatingServiceSoapHttpPort
3. Choose the soap address by right clicking the port-CreditRatingServiceSoapHttpPort and select Insert inside port.
4. Enter the location name as tbd and then click OK.

5. Now right click the CreditRating.wsdl and then validate WSDL.

6. The following screenshots shows that the WSDL document has executed with out nay errors.
Now let us discuss about generating the web service from the WSDL contract document. We need to perform the following steps for this.

1. Right click the TopDownContractDevelopment and select New project. Select empty project in the items. Now enter the project name as Service.
2. After right clicking the service project, expand business tier and choose web services and then select Java web service from WSDL click OK.
3. On the web service description, choose the URL location of your WSDL document and click next.

4. Now, select the package name as service and then click on finish button.
5. Expand application sources and service package under it. Select CreditRating service and see associated files under it.

6. Now, in the start up choose HTTP as 8988.

7. Select CreditRatingService and by double clicking CreditRatingimpl.java. Replace return 0 with the following code.
int id;
try
{
// Parses integer value of first 3 numbers form SSN
id = Integer.parseInt(ssn.substring(0,3));
}
catch(NumberFormatException e)
{
// if SSN is invalid returns -1
return -1;
}
if(id < 300)
{
// If value of the first 3 numbers from customer SSN is less
// than 300, credit rate is 1.
return 1;
}
else if(id < 600)
{
// If value less than 600, credit rate is 2.
return 2;
}
else if(id <900)
{
// If value less than 900, credit rate is 3.
return 3;
}
else
{
// Otherwise, credit rate is 0.
return 0;
}

We will get the following screen shots and these show it got executed with no errors. and this also shows that the processCreditRating() method throws a CreditRatingFaultMessage Exception.
```java
public class CreditRatingImpl {

    public int processCreditRating(String ssn) throws CreditRatingFaultMessage {
        int id;
        try {
            // Parses integer value of first 3 numbers from SSN
            id = Integer.parseInt(ssn.substring(0, 3));
        } catch (NumberFormatException e) {
            // if SSN is invalid returns -1
            return -1;
        }
        if (id < 300) {
            // if value of the first 3 numbers from customer SSN is less
            // than 300, credit rate is 1.
            return 1;
        } else if (id < 600) {
            // if value less than 600, credit rate is 2.
            return 2;
        } else if (id < 900) {
            // if value less than 900, credit rate is 3.
            return 3;
        } else {
            // Otherwise, credit rate is 0.
            return 0;
        }
    }
}
```
8. The following screen shot appears on successful compilation by choosing run. By right clicking the CreditRatingservice.

![Screen Shot](image)

We need to test the web service now.

This involves following steps.

1. By right clicking the TopDownContractDevelopment and selecting NewProject we will obtain a Create project Dialog box where we need to enter the project name as Client.

![Create Project Dialog](image)
2. Now select new by right clicking the Client project. expand business tier and choose web services from where you have to choose web service proxy and the following screen shot shows that.

![New Gallery](image)

- Categories:
  - General
  - Business Tier
    - ADF Business Components
    - Content Repository
    - EJB
    - TopLink
  - Web Services
  - Client Tier
  - Database Tier
  - Integration Tier
  - Web Tier

- Items:
  - Java Web Service from WSDL
  - JAX-RPC Mapping File
  - JMS Adapter
  - JMS Web Service
  - MQ Series Adapter
  - Oracle Applications Adapter
  - PL/SQL Web Service
  - UDDI Registry Connection
  - URL Service Data Control
  - Web Service Data Control

- Web Service Proxy:
  - WSDL Document

Description:
Launches the Create Web Service Proxy wizard, in which you generate a Java proxy for calling a web service.

To enable this option, you must select a project or a file within a project in the Application Navigator.
3. We will obtain the welcome page of the web service proxy wizard as follows. Here we need to mention the URL of the WSDL document and the mapping file will populate automatically.
4. Now, select the radio button - Run against a service deployed to Embedded OC4J.
5. Add the following code to call the web service.

```java
int rating = myPort.processCreditRating("78964022");
System.out.println("Credit Rating is "+ rating);
```

6. Run the web service and the following screenshot shows that the process has executed successfully.

```
U:\jdk\bin\j ava.exe -client - classpath "C:\Documents and Settings\greshma\jdeveloper\mywork\TopDownContractDevelopment-Service-context-root\CreditRatingService\bin\creditRating.jar" -clientclass com.topdown.creditratingapi.CreditRating
Calling http://129.130.11.59:8988/TopDownContractDevelopment-Service-context-root/CreditRatingService/services/creditRating
Credit Rating is 3
Process exited with exit code 0.
```