Creating Web services and an EJB skeleton from a WSDL file

Contents

Creating Web services and an EJB skeleton from a WSDL file

Learning objectives

You will learn how to create a WS-I compliant Web service from a WSDL file. The wizard generates an EJB skeleton that contains a set of methods corresponding to the operations described in the WSDL document. When the EJB is created, each method has a trivial implementation that can be easily replaced by editing the EJB.

Time required

To complete this tutorial, you will need approximately **1 hour and 30 minutes**. If you decide to explore other facets of Web services or EJBs while working on the tutorial, it could take longer to finish.

Prerequisites

In order to complete this tutorial end to end, you should be familiar with:

- Basic Web services concepts, such as SOAP and WSDL
- Basic XML
- EJB programming

When you are ready, begin Lesson 1.1: Set up the workspace and create the required projects

Feedback

Related information

View the PDF version

Lesson 1.1: Set up the workspace and create the required projects

Create a WebSphere Application Server v6.1 server

To create a WebSphere Application Server do the following:

1. From the **File** menu, select **New** → **Other** → **Server** → **Server** → **Next**.
2. Select **WebSphere v6.1 Server** as the server type. Click **Next**.
3. If this runtime has not been created in your workspace, you will be prompted to select the installation directory for the server. Click **Next**.
4. Accept the default server port and name. Click **Finish**.
5. Wait for the server to start. once it has started the console will display `Server server1 open for e-business;`

Setting the WS-I compliance level

WS-I refers to Web service interoperability; this includes interoperability across platforms, operating systems, and programming languages.

The WS-I organization sets out standards collected in documents called Profiles that define the requirements needed to make a Web service interoperable. The Rational Developer products validate Web services against the WS-I Simple SOAP Binding Profile 1.0 (WS-I SSBP) and the WS-I Attachments Profile 1.0 (WS-I AP). For more information on WS-I, refer to their Web site: http://www.ws-i.org/

By default, the WS-I SSBP compliance level is set to **Ignore**. With this setting, no warning will be given if non-compliant choices are made. This compliance level is used by the Web service wizards and the WSDL validation tool. This sample generates a WS-I compliant Web service, therefore you should set the WS-I compliance level to **Require**.

You can change the WS-I compliance level by following the proceeding steps:

1. On the main menu bar, click **Window** → **Preferences**. The Preferences dialog box opens.
2. Expand the **Web Services** branch and select **WebSphere** → **WS-I BSP Compliance**.
3. Select the **Require** option from the drop-down list beside WS-I SSBP.
4. Click **OK**.

Creating the Web service EJB project

The remaining steps in this tutorial will be done in the J2EE perspective. If you are asked if you want to change to another perspective after performing a task, select **No**.

The EJB project will contain the business logic for the Web service as well as the WSDL file.

1. On the main menu bar, click **File** → **New** → **Project** → **EJB** → **EJB Project**. Click **Next**.
2. Type `TempEJB` in the Name text field. Under Target Runtime ensure that the target server is WebSphere Application Server v6.1. In the **EAR Project Name** field, enter `TempEJBEAR` as the EAR name. Click **Next**.
3. By default the correct facets for this type of project will be selected. Click **Next**.
4. Clear the checkbox for creating a client JAR module. The Web services wizard will create this module for you. Click **Finish**.

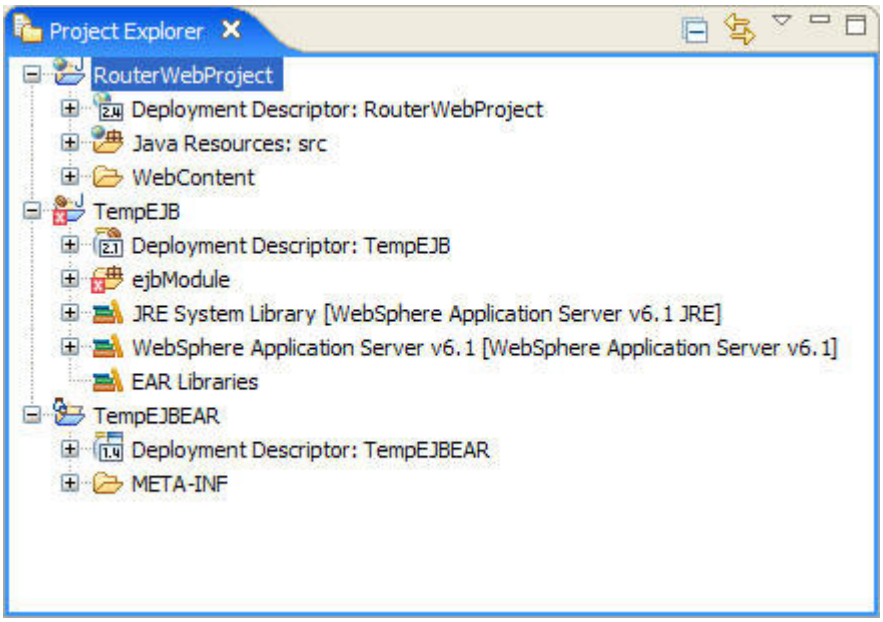
The EJB project that will contain the Web service logic and the associated EAR are created. The EJB project will have an error associated with it because it does not contain an enterprise bean. The bean will be generated by the Web services wizard.

Creating the Web service router project

EJB Web services require a router project. This project contains the router servlet that acts as the endpoint for the service and will call out to the EJB. If you are using SOAP over JMS as your transport method the router project needs to be an EJB project. If you are using SOAP over HTTP as we are in this tutorial, the router project should be a Web project. The project you create must be added to the same EAR as the EJB project that will contain the enterprise bean. This project should not contain any of the business logic for your Web service.

You can create a Web project by following the proceeding steps:

1. On the main menu bar, click **File** → **New** → **Project** → **Web** → **Dynamic Web Project**. Click **Next**.
2. Type `RouterWebProject` in the Name text field. Under Target Runtime ensure that the target server is WebSphere Application Server v6.1. In the **EAR Project Name** field, ensure `TempEJBEAR` is selected. This will ensure that the enterprise bean that you will create later and your router project are both referenced in the same EAR. Click **Finish**.
3. You have now created your router project and your workspace should look similar to the following:



Adding the projects to the server

You can associate the project with the server that your Web service will run on by following these steps:

1. Right-click the server in the Servers view and select **Add and Remove projects**. If the Servers view is not open in your workspace, open it from the **Window** menu by selecting **Show View → Servers**.
2. In the window that opens, select TempEJBEAR which contains your router and EJB projects, and click **Add**.
3. Click **Finish**.

Lesson checkpoint

Now you are ready to begin Lesson 1.2: Import and validate the WSDL file.

Feedback

Lesson 1.2: Import and validate the WSDL file

The Temperature Conversion WSDL document has been provided for you. The WSDL file that you will use in this tutorial converts temperature from Fahrenheit to Celsius and Celsius to Fahrenheit.

Before you begin, you must complete Lesson 1.1: Set up the workspace and create the required projects.

You can create a new WSDL document or import an existing one. The Temperature Conversion WSDL document used in this tutorial has been provided for you in a simple project. Complete the following steps to import the Temperature Conversion WSDL document into the workbench:

1. In the Project Explorer view, expand **TempEJB**, right-click and click **New → Folder** to create a folder called `wsdl`. Click **Finish**.
2. Import the simple project containing the WSDL file
3. Expand the imported simple project called TempConversionWSDL and expand the WebContent folder. The ConvertTemperature.wsdl file will be located here.
4. Right-click the ConvertTemperature.wsdl file and select **Move...**
5. Select the `wsdl` folder you created in the TempEJB project as your destination and click **OK**.

Validating the WSDL document

The WSDL Validator can validate WSDL semantics and WS-I compliance.

You can validate the Temperature Converter WSDL document by following the proceeding steps:

1. Select the ConvertTemperature.wsdl document in the Project Navigator.
2. Right-click, and click **Validate**. Any errors will be displayed in the Tasks view.

If no errors occur during validation, you can proceed to create the Web service.

Lesson checkpoint

Now you are ready to begin Lesson 1.3: Create the Web service.

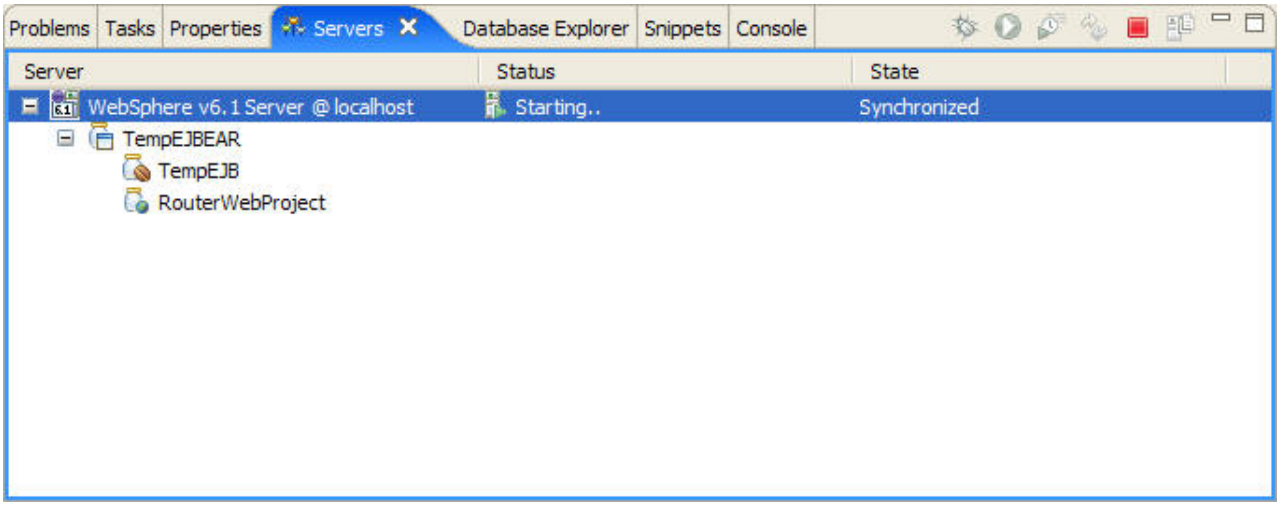
Feedback

Lesson 1.3: Create the Web service

Before you begin, you must complete Lesson 1.2: Import and validate the WSDL file.

Before you attempt to create a Web service it is strongly suggested that you start the WebSphere Application Server on which the Web service will run. Although you can start the server in the Web service wizards, since it may take several minutes to start depending on the speed of your machine, starting the server before you begin will both increase the speed with which you complete the wizard and reduce the chance that the wizard will generate an error because the server is taking too long to start.

To start the server, right-click the server in the Servers view and select **Start**:



If the Servers view is not open in your workspace, open it from the **Window** menu by selecting **Show View** → **Servers**.

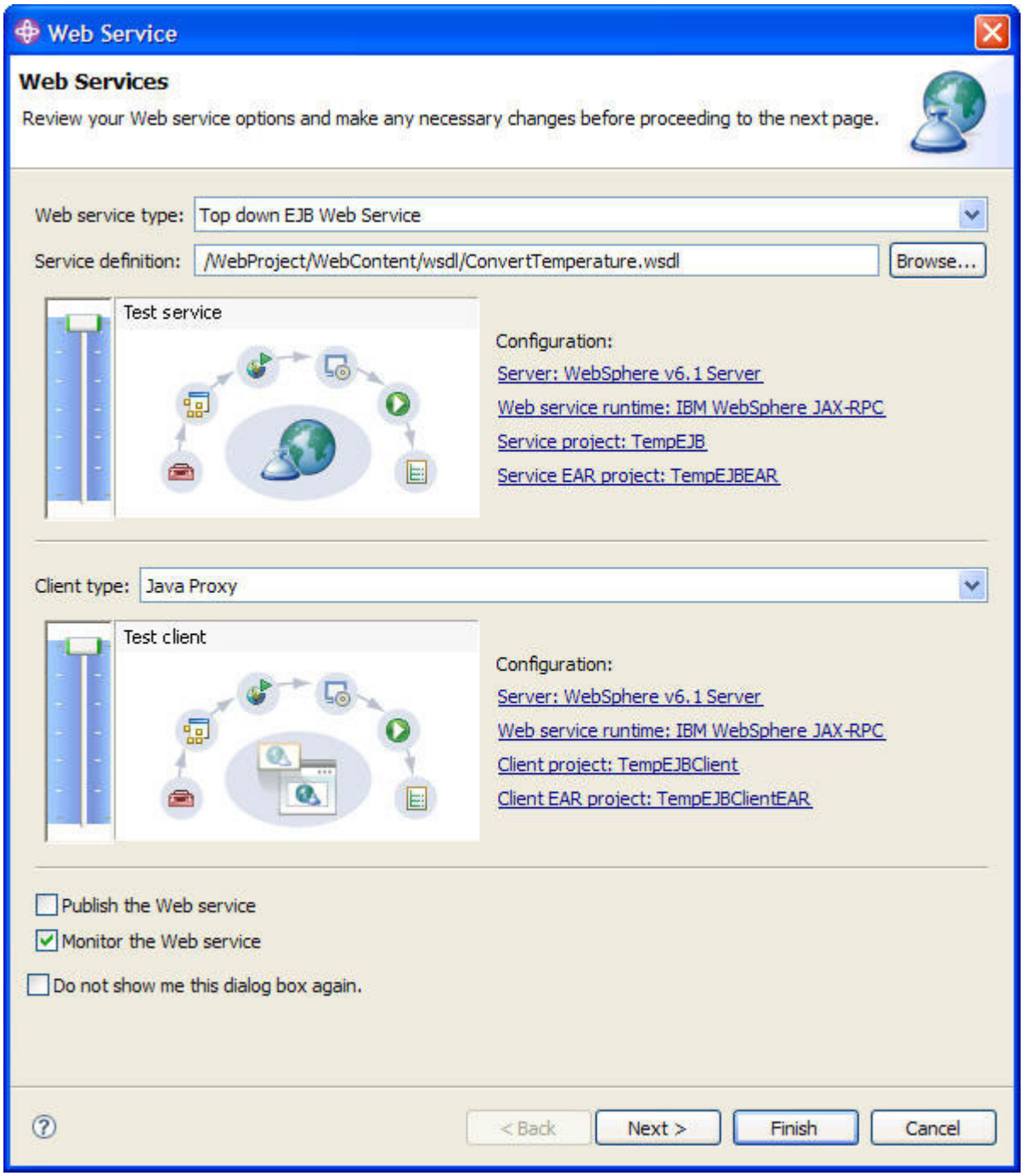
Create a Web service from a WSDL file

The Web Service wizard assists you in creating a new Web service, configuring it for deployment, and deploying the Web service to a server. Once your Web service is deployed, the wizard assists you in generating the client proxy and sample application to test the Web service.

When you have completed testing, you can publish your Web service to a UDDI Business Registry using the Export wizard.

1. In the Project Explorer, select the **ConvertTemperature.wsdl** document in your EJB project.
2. Click **File** → **New** → **Other**. Select **Web Services** in order to display the various Web service wizards. Select the **Web Service** wizard. Click **Next**.
3. Select the following options on the first page of the wizard:
   - Web service type: Top down EJB Web service
   - Service definition: ensure the ConvertTemperature.wsdl file that you imported is selected.
   - Level of service generation slider: move the slider to Test service.
   - Service configuration: ensure that WebSphere v6.1 Server and the IBM WebSphere JAX-RPC runtime environment are selected. Click **Service project** and enter `TempEJB` as your service project name. TempEJBEAR should be selected as your service EAR project. Do not
   - Level of client generation slider: move the slider to Test client.
   - Client configuration: ensure that WebSphere v6.1 Server and the IBM WebSphere JAX-RPC runtime are selected. The wizard will create a client and client EAR project. You can accept the default names or enter a different name.
   - Monitor the Web service.

   Once you have selected the correct options the wizard should look similar to the following:



Click **Next**.

4. On the Web Service Skeleton EJB Configuration page, select `RouterWebProject` as your router project if it is not already selected.
5. In the Web Service Test page, you can select a test facility to test your Web service before a client or proxy is developed. Select Web Services Explorer as the test facility for your Web service and click **Launch**. This step may take several seconds for the WebSphere Application server to start.
6. The Web Services Explorer is displayed in a Web browser. Select **fahrenheitToCelsius** or **celsiusToFahrenheit** from the operations list. Enter a number in the value field and click **Go**. A trivial implementation of each of these operations is provided, and a default value of -3 is returned. If both operations complete successfully, close the browser window and click **Next** in the Web services wizard.

7. In the Web Service Proxy page, keep the Security Configuration selection at `No Security` to remain WS-I compliant. Click **Next**.
8. In the Web Service Client Test page, ensure **Test the generated proxy** and **Run test on server** are both selected. In the Methods section. ensure that all methods are selected, or click **Select All** to select all methods. If you want to publish your Web service to a UDDI Registry, click **Next** to configure the Web Service Publication options. However this step will not be covered in this tutorial. Otherwise, click **Finish**.
9. The sample application is launched in a Web browser. You can use this application to test the Web service by selecting a method in the Methods frame, entering an input value in the Inputs frame, and clicking **Invoke** to view the result in the Result frame. Do not close the TestClient.jsp browser window yet - it will be used to test the Web service traffic for WS-I compliance later in this tutorial.

Lesson checkpoint

Now you are ready to begin Lesson 1.4: Implement the temperature conversion methods.

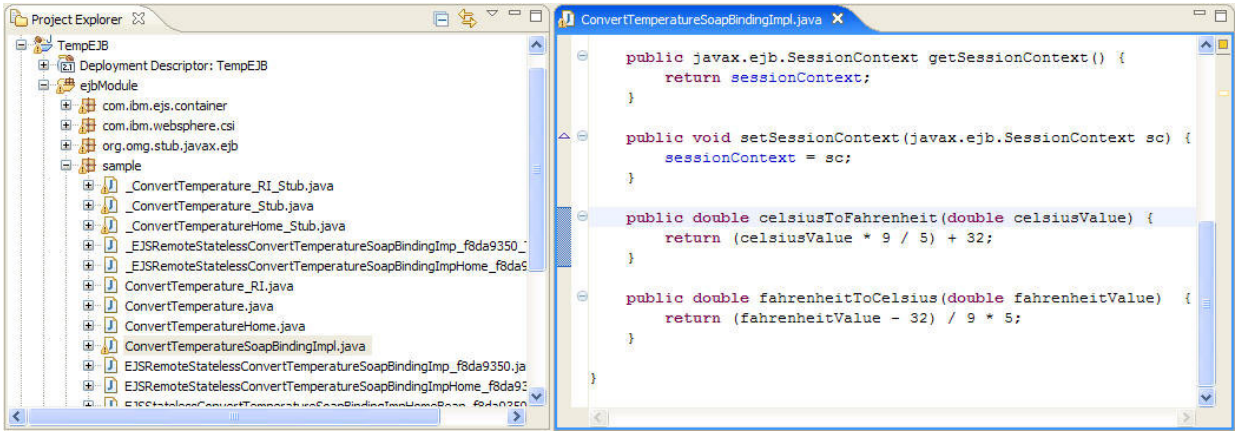
Feedback

Lesson 1.4: Implement the temperature conversion methods

Before you begin, you must complete Lesson 1.3: Create the Web service.

Trivial implementations of the **fahrenheitToCelsius** and **celsiusToFahrenheit** methods were automatically generated when you created a Web service from your WSDL document. In this section you will replace these trivial implementations with more meaningful code, and perform the necessary steps to test your new methods.

1. In the Project Explorer view, select **ConvertTemperatureSoapBindingImpl.java** under **TempEJB** → **ejbModule** → **sample**.
2. Locate the `fahrenheitToCelsius` method and replace the current implementation with the following:
   `return (fahrenheitValue - 32) / 9 * 5;`
3. Locate the `celsiusToFahrenheit` method and replace the current implementation with the following:
   `return (celsiusValue * 9 / 5) + 32;`



4. Save your updates by clicking **File** → **Save**.
5. Restart the EAR by expanding **WebSphere v6.1 Server** in the Servers view and right-clicking **TempEJBEAR** → **Restart TempEJBEAR**.
6. Click **Run** → **Launch** the Web Services Explorer from the main menu bar and repeat the instructions from the previous section to test your **fahrenheitToCelsius** and **celsiusToFahrenheit** methods.

Lesson checkpoint

Now you are ready to begin Lesson 1.5: Validate the Web service traffic WS-I compliance.

Feedback

Lesson 1.5: Validate the Web service traffic WS-I compliance

Before you begin, you must complete Lesson 1.4: Implement the temperature conversion methods.

To ensure that the SOAP envelope request and response pairs are WS-I compliant, you need to direct your Web service traffic through the TCP/IP Monitor:

When creating a Web service using the Web service or Web service client wizards, you can select to set up and run the TCP/IP Monitor automatically. Since you chose this option when creating the Web service, the TCP/IP monitor view should be in your workspace. If it is not, you can open this view by selecting **Window** → **Show View** → **Other** → **Debug** → **TCP/IP Monitor**.

Alternately, you can set up the TCP/IP Monitor manually by completing the following steps:

1. In the sample application, invoke the getEndPoint method. Record this endpoint.
2. Create a server to act as the TCP/IP Monitor:
   a. From the **Window** menu, select **Preferences**.
   b. In the Preferences window, expand **Run/Debug** and then select **TCP/IP Monitor**.
   c. Select the **Show TCP/IP Monitor View when there is activity** check box.
   d. Under the TCP/IP Monitors lists, click **Add**. A New Monitor dialog box opens.
   e. Specify the following settings:

Option	Description
Local monitoring port	Specify a unique port number on your local machine.
Host name	Specify the host name or IP address of the machine where the server is running.
Port	Specify the port number of the remote server.
Type	Specify whether the request type from the Web browser are sent by HTTP or TCP/IP. If the HTTP option is selected the requests from the Web browser are modified so that the HTTP header points to the remote machine and separated if multiple HTTP requests are received in the same connection. If the TCP/IP option is selected, all the requests are sent byte for byte.

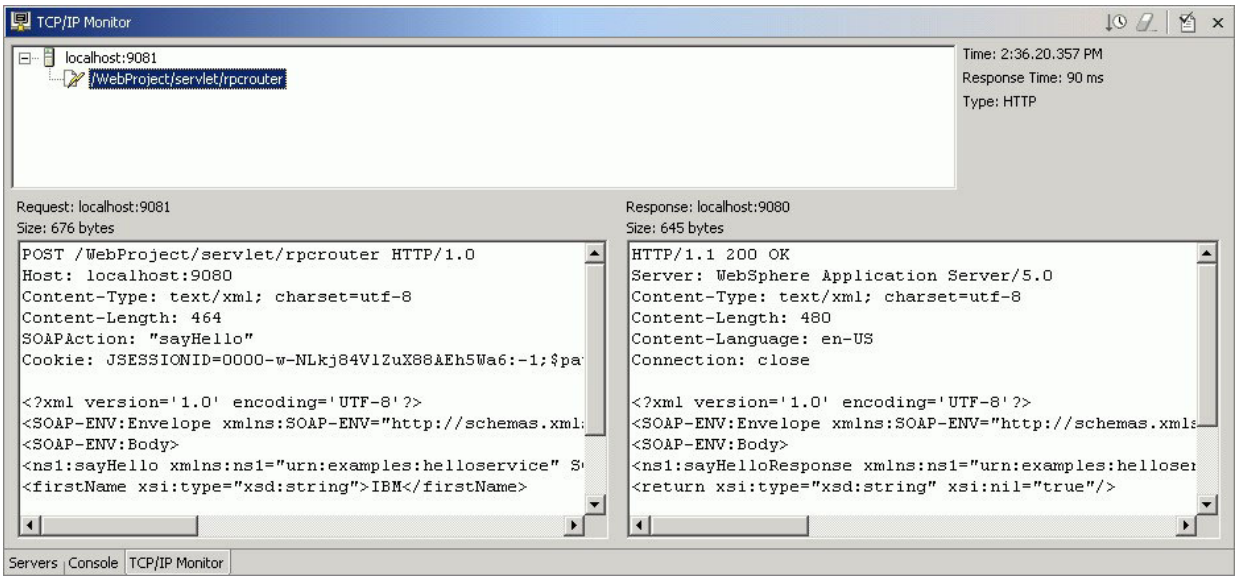
3. In order to route the Web service through the monitor, the endpoint of the Web service client needs to be changed. The TCP/IP Monitor listens on port 9081. In the Web browser window used in step 1, invoke the setEndPoint method, and change the endpoint so that it directs to port 9081. For example, the default would be: `http://localhost:9081/web_module_context_root/servlet/rpcrouter` Invoke the getEndPoint method again to ensure that your change has been implemented.

Describe or show the results of the steps they just followed.

Routing traffic and verifying WS-I compliance

You can route traffic through the TCP/IP monitor and test the traffic for WS-I compliance by following the proceeding steps:

1. Select a Web service method in the Methods pane. Invoke this method.
2. Change to the TCP/IP Monitor view by clicking the TCP/IP Monitor tab in the Servers view. This will display request and response pairs that are being routed through the TCP/IP Monitor. It will look similar to the following picture:



3. To ensure that your Web service SOAP traffic is WS-I compliant, you can generate a log file by clicking the  icon. In the dialog box that opens, select a name for the log file and specify where you want it to be stored. This log file will be validated for WS-I compliance. You can open the log file in an XML editor to examine its contents.

Lesson checkpoint

Finish your tutorial by reviewing the materials in the Summary.

Feedback

Summary

You have created a WS-I compliant Temperature Conversion Web service and an EJB skeleton from a WSDL file, and learned to implement some basic EJB methods.

Lessons learned

If you have completed all of the exercises, you should now be able to:

- Set the WS-I compliance level.
- Create a Web project called WebProject.
- Import the Temperature Conversion WSDL document.
- Validate the WSDL, and validate the WSDL file's WS-I compliance.
- Create the Web service and skeleton EJB, and test the methods included in the Web service using the Web Services
- Implement the fahrenheitToCelsius and celsiusToFahrenheit methods (optional)
- Deploy the Web service to the WebSphere Application Server.
- Test the Web service traffic for WS-I compliance.

More information

For more information about Web services, WSDL, SOAP, and the WebSphere v6 run-time environment, consult the online help for WebSphere Studio by clicking **Help → Help Contents**. For more in-depth technical articles on Web services, consult DeveloperWorks

Feedback

(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.