

Creating a Web service client using Faces introduction

Time required

To complete this tutorial, you will need approximately **30 minutes**.

Prerequisites

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

- Basic Web design concepts, such as Web sites, pages, browsers, and servers
- How to create a simple static Web Page
- The elements of a Web page, such as tables, hyperlinks, forms, and images
- Basic JavaServer Faces concepts
- Basic Web service concepts, such as WSIL, WSDL, and UDDI

Learning objectives

In this tutorial you will learn to:

- Create a Web project
- Create a Faces JSP file
- Set project-level WS-I compliance levels
- Locate a Web service from a known Web Service Inspection Language (WSIL) URL
- Create a proxy to the Web service and place one of its methods on the Faces JSP file for invocation

When you are ready, begin [Exercise 1: Setting up a workspace and create a Web project](#)


[Terms of use](#) | [Feedback](#)

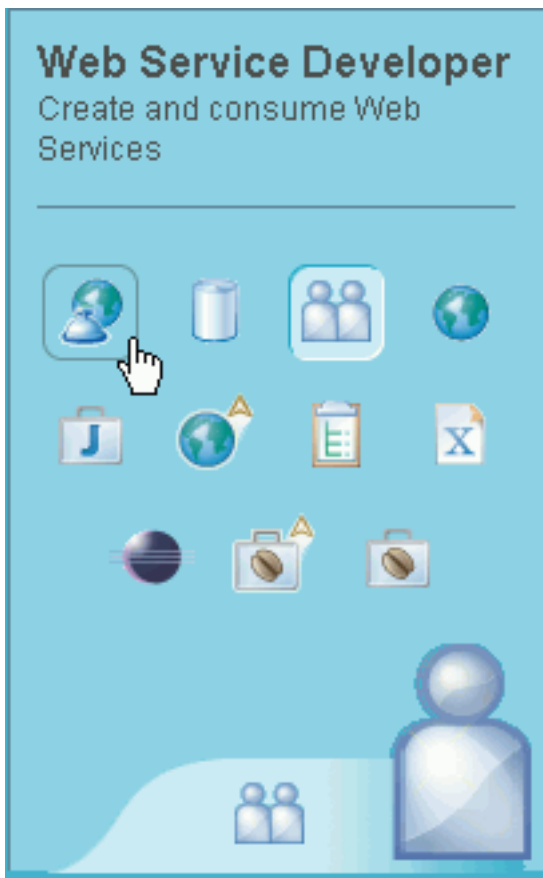
(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.

Exercise 1.1: Setting up the workspace and creating a Web project

Enabling Web service capabilities

To enable the capabilities required for Web services development:

1. On the Welcome page, check to see if Web services are enabled by looking for the Web services icon in the lower right-hand corner: . If the Welcome page has been closed, you can reopen it from the Help menu.
2. If Web services are not enabled, click the icon in the lower right corner that looks like a person. This generates a list of capabilities that you can select from.
3. Click the Web services icon in the top left corner:



You have now enabled the tools used in Web services development.

Creating a Web project

You must create a dynamic Web project to contain the Faces JSP file. To create the Web project:

1. Click **File > New > Other > Web > Dynamic Web Project**. Click **Next**.
2. Type `JSPWebProject` in the **Name** text field. Click **Finish**.
3. If you are prompted to switch to the Web perspective, select **No**.

Setting the WS-I compliance level for the project

WS-I refers 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/>

You can set the WS-I compliance level for an individual project or for all projects contained in your Workspace. The default WS-I compliance level is Suggest, which means that a warning is generated when a non WS-I compliant suggestion is made. In this tutorial, the Web service we are creating is not WS-I compliant, so we will set the level of WS-I compliance to Ignore for this project.

1. Right-click **JSPWebProject** in the Project Navigator and click **Properties**.
2. Click **WS-I Compliance** from the list of preferences.
3. Select **Ignore WS-I compliance** for both SSBP and AP, and click **Apply** and then **OK**.

The JSPWebProject will no longer generate WS-I compliance warnings, but any other projects in your workspace will retain their original WS-I compliance settings.

Now you are ready to begin [Exercise 1.2: Create a Faces JSP file](#).

[Terms of use](#) | [Feedback](#)

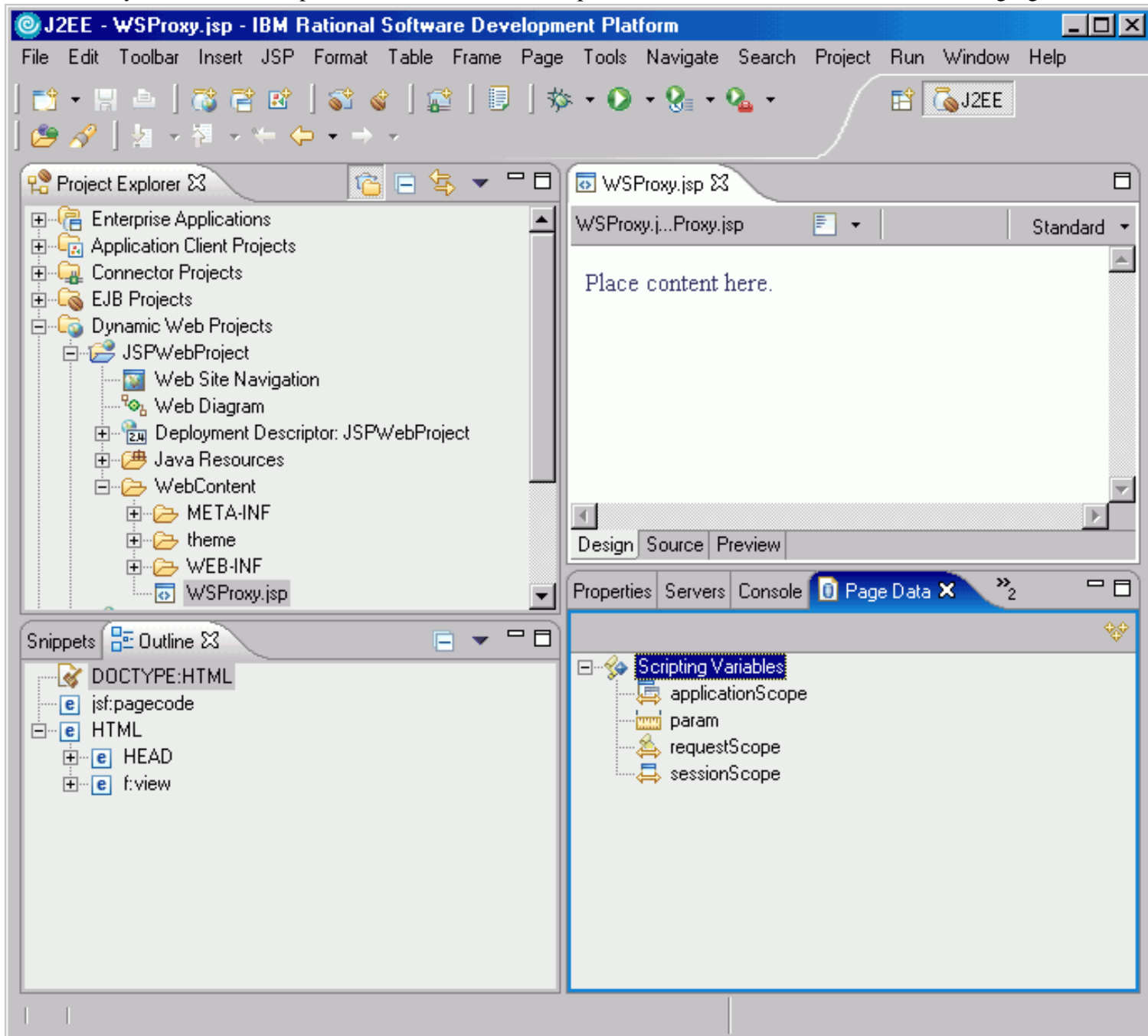
(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.

Exercise 1.2: Create a Faces JSP file

Before you begin, you must complete [Exercise 1.1: Setting up the workspace and creating a Web project](#).

To create a Faces JSP file on which the created Web service proxy code will reside:

1. Expand the Web project that you created, and select the **WebContent** folder.
2. From the **File** menu, click **New > Other > Web > Faces JSP File**.
3. Select a name for the Faces JSP file and click **Finish**. For this tutorial, the file will be called **WSProxy.jsp**. The Faces JSP file that you have created opens in an editor. Your workspace should now look similar to the following figure:



4. If the Page Data view is not visible, you can open it from the **Window** menu by clicking **Show View > Other > Web > Page Data**.

Now you are ready to begin [Exercise 1.3: Finding a Web service and creating a Web service proxy](#).

[Terms of use](#) | [Feedback](#)

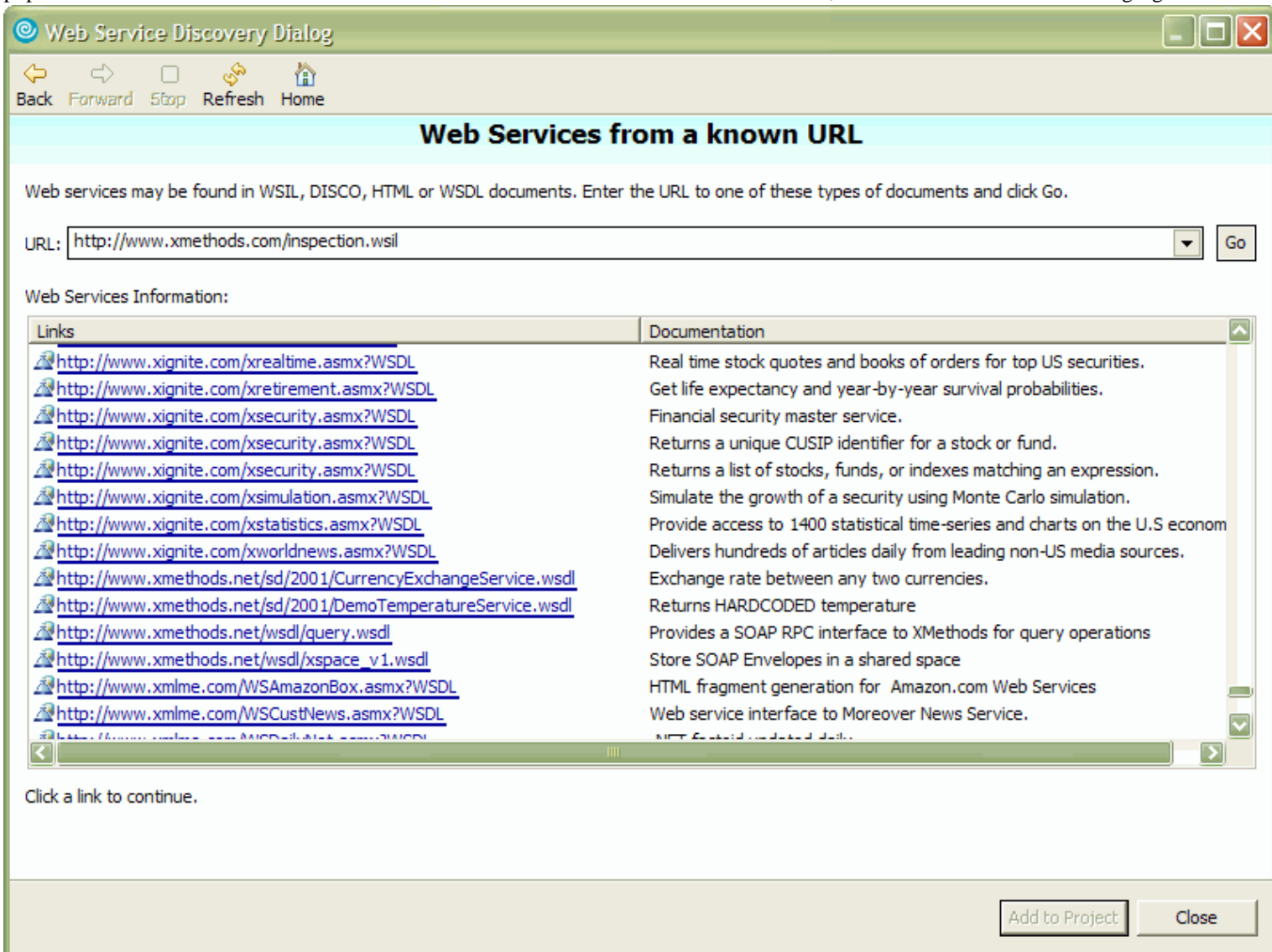
(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.

Exercise 1.3: Finding a Web service and creating a Web service proxy

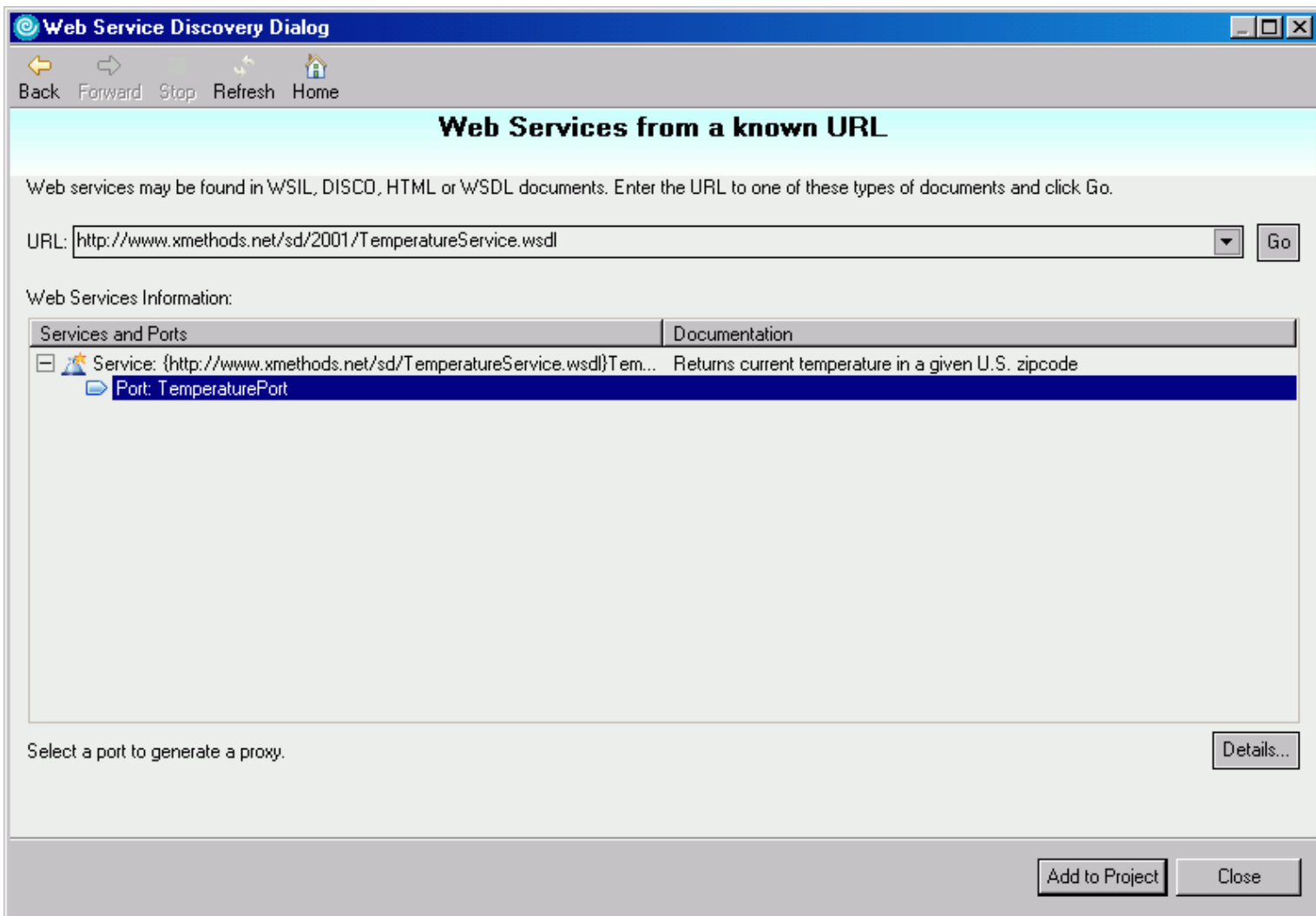
Before you begin, you must complete [Exercise 1.2: Create a Faces JSP file](#).

You must now locate a Web service for which you want to create a client. This Web service can be on a private or public UDDI registry, at a URL, or in your workspace. For this tutorial, we will search for a Web service that returns the temperature for a given zip code that is provided by www.xmethods.com. To find a Web service and create a proxy for the Web service using the Web Service Discovery wizard:

1. In the Page Data view, right-click on empty space and click **New > Web Service**.
2. Select **Web services from a known URL**. Although we do not know the URL for the WSDL file, by using a Web Service Inspection Language (WSIL) file we can search all the Web services listed on the site. WSIL files are frequently found at the following location: http://www.web_service_provider.com/inspection.wsil
3. Enter the following WSIL URL in the URL field: <http://www.xmethods.com/inspection.wsil> and click **Go**. The wizard should populate the Web Services Information table with a list of Web services available on the site, and look similar to the following figure:



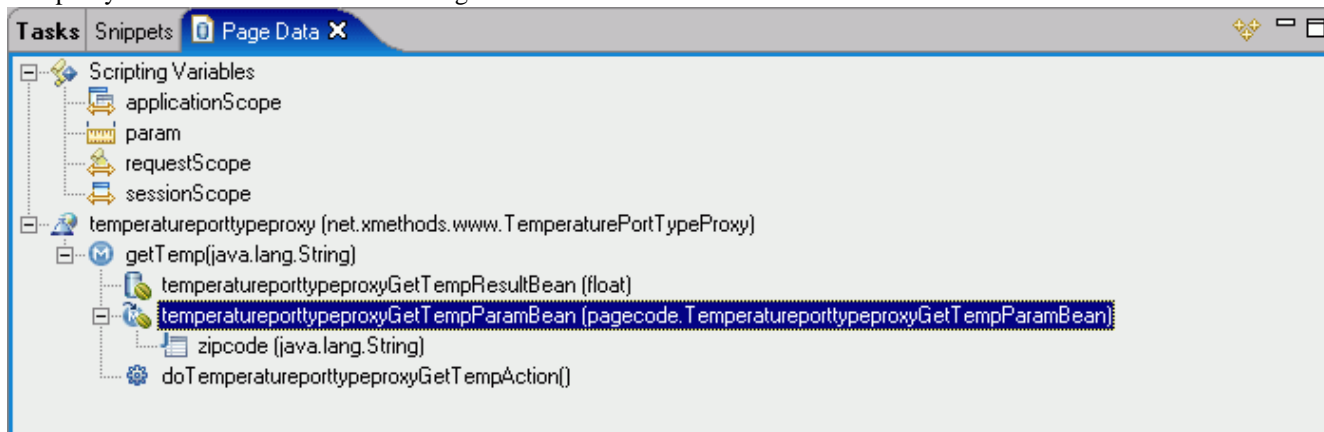
4. The Web services are listed alphabetically by provider. Since we are using a Web service provided by www.xmethods.net, so scroll down the list until the links listed all start with www.xmethods.net. Select the TemperatureService.wsdl located at <http://www.xmethods.net/sd/2001/DemoTemperatureService.wsdl> by clicking on the link.
5. The Web service we have selected is now displayed in the table. Click **Port: TemperaturePort** to select it.



To test the Web service, click **Details** to launch the Web service in the Web Services Explorer. For more information about using the Web Services Explorer, refer to the online help.

6. Once you have tested the Web service, click **Add to Project**. If you are prompted to overwrite files on your system click **Yes to All**.
7. The Web service you selected is now listed in the list of existing Web services. The only method available for this Web service is also listed. Click **Finish** to add it to your Page Data view.

The proxy should now be shown in the Page Data view:



Now you are ready to begin [Exercise 1.4: Adding a proxy method to the Faces JSP file](#).

[Terms of use](#) | [Feedback](#)

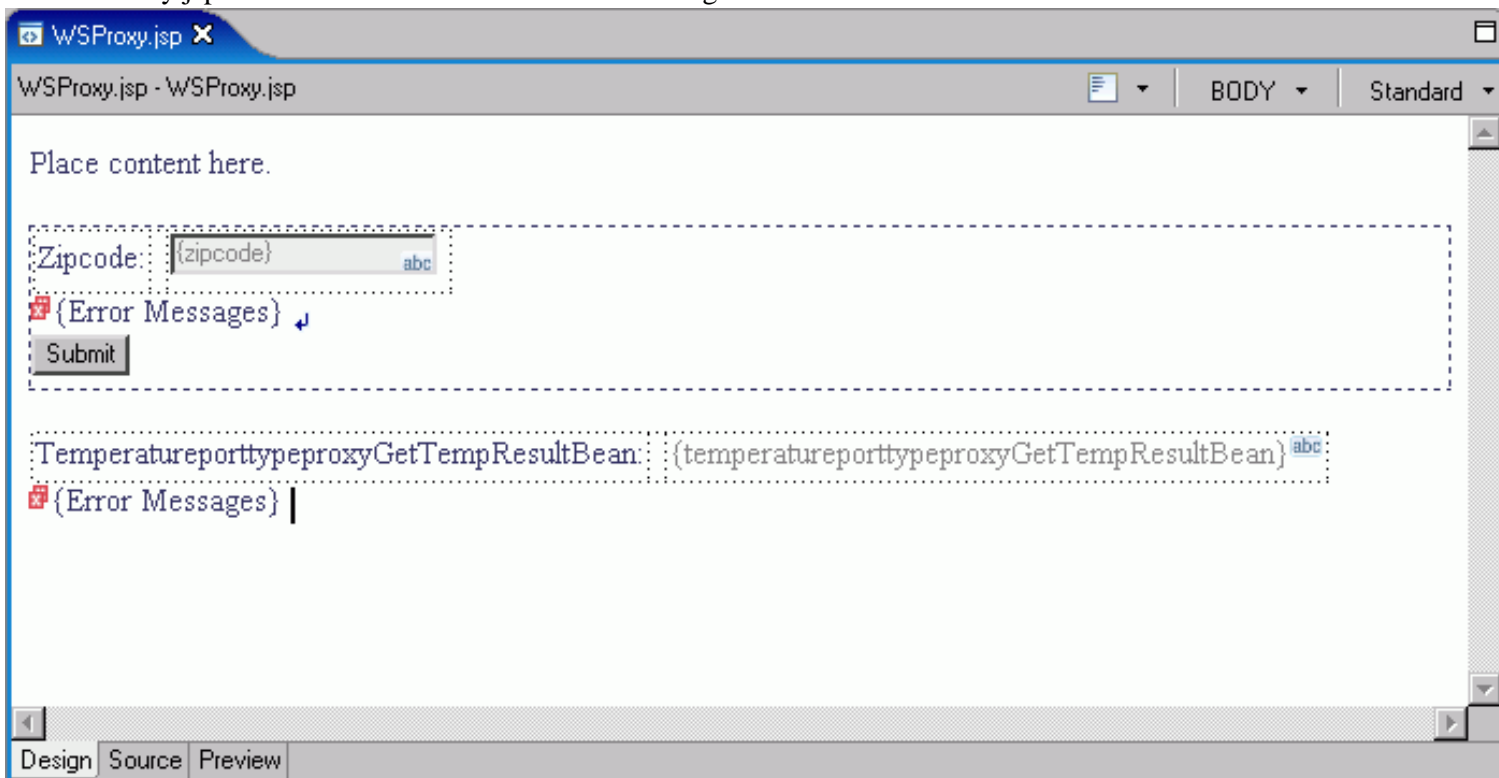
(C) Copyright IBM Corporation 2000, 2004. All Rights Reserved.

Exercise 1.4: Adding a proxy method to the Faces JSP file

Before you begin, you must complete [Exercise 1.3: Finding a Web service and creating a Web service proxy](#).

Now that you have generated a proxy for the Web service, you need to add its method to your Faces JSP file.

1. Click the **Design** tab of the WSProxy.jsp.
2. In the Page Data view, select **temperatureporttypeproxyGetTempParamBean(pagecode. TemperatureporttypeproxyGetTempParamBean)** and drag it onto WSProxy.jsp.
3. The Insert Web Service panel opens. The zip code field should be selected. Click **Finish**. This will create a GUI element that allows you to enter a zip code and click Submit to run the doGetTempAction.
4. In the Page Data view, select **temperatureporttypeproxyGetTempResultBean(float)** and drag it onto the WSProxy.jsp.
5. The Insert Web Service panel opens. The output field for the Web service should be selected. Click **Finish**. This will create a GUI element that displays the result.
6. Now you need to bind the GetTemp action to the Submit button on the JSP. To do this, select **doTemperatureporttypeproxyGetTempAction()** on the Page Data view, and drag it onto the Submit button on the JSP.
7. The WSProxy.jsp should now look similar to the following:



8. Save WSProxy.jsp by clicking **File > Save**. Close the WSProxy.jsp editor.
9. In the Project Navigator, right-click **WSProxy.jsp** and click **Run > Run on Server**.
10. Select the WebSphere Application Server v6 server that is already created and click **Finish**. If you want to run the Faces JSP on a server other than WebSphere Application Server v6, you would need to re-target the Web project that you created to another server.

Once the JSP is running on the server, the following should open in a Web browser in the workspace. You can test the Web service by entering a valid value in the zip code field and clicking **Submit**.

Finish your tutorial by reviewing the materials in the [Summary](#).

[Terms of use](#) | [Feedback](#)

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

Summary

In this tutorial you learned how to search for a Web service using WSIL, create a client for the WSDL file of the Web service, and place the client code on a Web page using Faces.

Completed learning objectives

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

- Create a Web project.
- Create a Faces JSP file.
- Set project-level WS-I compliance levels.
- Locate a Web service from a known WSIL URL.
- Create a proxy to the Web service and place one of its methods on the Faces JSP file for invocation.

When you have completed the tutorial and no longer require the Web projects for testing, right-click each project and select **Delete** to remove the projects from your workspace.

More information

For more information about Web services, WSDL, and SOAP, please consult the online help for WebSphere Studio (**Help > Help Contents**). For more in-depth technical articles on Web services, consult www.ibm.com/developerworks/webservices

[Terms of use](#) | [Feedback](#)

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