

Creating a Web service client using Faces

Contents

Creating a Web service client using Faces 1

Lesson 1.1: Setting up the workspace and creating a Web project	1
Creating a Web project	1
Adding Faces facets to the Web project	1
Setting the WS-I compliance level for the project	2
Lesson checkpoint	2

Lesson 1.2: Create a Faces JSP file	2
Lesson checkpoint	3
Lesson 1.3: Finding a Web service and creating a Web service proxy	3
Running the proxy on the server	5
Lesson checkpoint	5
Summary	5

Creating a Web service client using Faces

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

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

When you are ready, begin Lesson 1: Setting up a workspace and create a Web project

Feedback

Related information

[View the PDF version](#)

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

Creating a Web project

You must create a dynamic Web project to contain the Faces JSP file. You can create the Web project by following the proceeding steps:

1. Click **File** → **New** → **Other** → **Web** → **Dynamic Web Project**. Click **Next**.
2. Type JSPWebProject in the **Name** text field. Select your runtime from the **Target Runtime** list - if your preferred server is not listed you can click **New** and add the required information to configure it. For this tutorial Apache Tomcat 5.5 has been selected. In the **Configurations** field, select Faces Project from the list. Click **Finish**.
3. If you are prompted to switch to the J2EE perspective, select **Yes**.

Adding Faces facets to the Web project

Facets define characteristics and requirements for J2EE projects. When you add a facet to a project, that project is configured to perform a certain task, fulfill certain requirements, or have certain characteristics.

If you selected Faces Project as the configuration when creating your dynamic Web project, the correct facets have already been added to your project. If you created a Web project without selecting that configuration, you need to add the correct facets to the project so that it supports Faces:

1. Right-click your Web project and select **Properties**.
2. From the list of properties select **Project Facets** → **Add/Remove Project Facets**.
3. From the list of available facets, select:
 - a. JSTL
 - b. Base Faces Support

Click **Finish** and then **OK** to exit the Properties window.

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 Explorer and click **Properties**.
2. Click **Profile Compliance and Validation** from the list of preferences.
3. Select **Ignore WS-I compliance** for both profiles, 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.

Lesson checkpoint

Now you are ready to begin Lesson 1.2: Create a Faces JSP file.

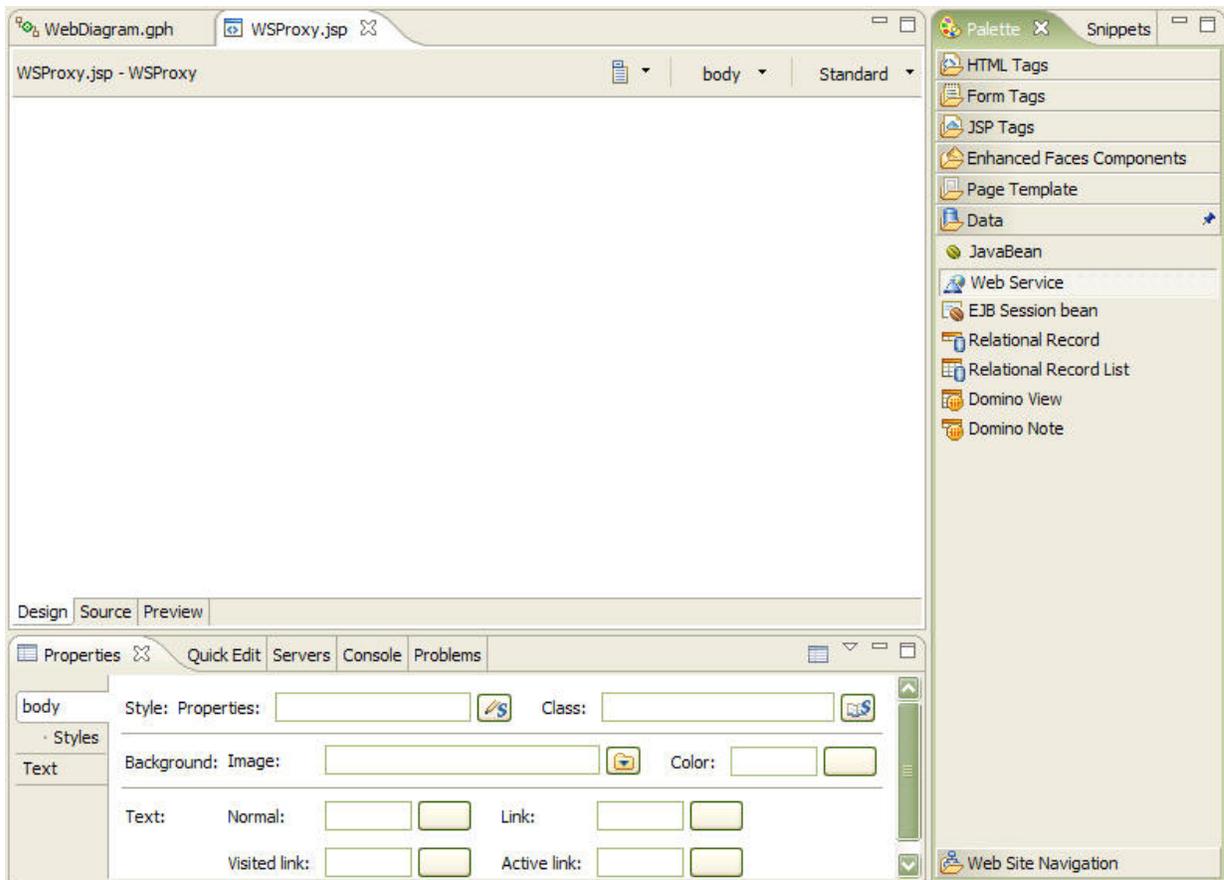
Feedback

Lesson 1.2: Create a Faces JSP file

Before you begin, you must complete Lesson 1.1: Setting up the workspace and creating a Web project.

You can create a Faces JSP file on which the created Web service proxy code will reside by following the proceeding steps:

1. Expand the Web project that you created, and select the WebContent folder.
2. From the **File** menu, click **New** → **Other** → **Web** → **Web Page**.
3. Select a name for the Faces JSP file. Ensure that the JSP template is selected from the template list, 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:



Lesson checkpoint

Now you are ready to begin Lesson 1.3: Finding a Web service and creating a Web service proxy.

Feedback

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

Provide a short description for this tutorial lesson.

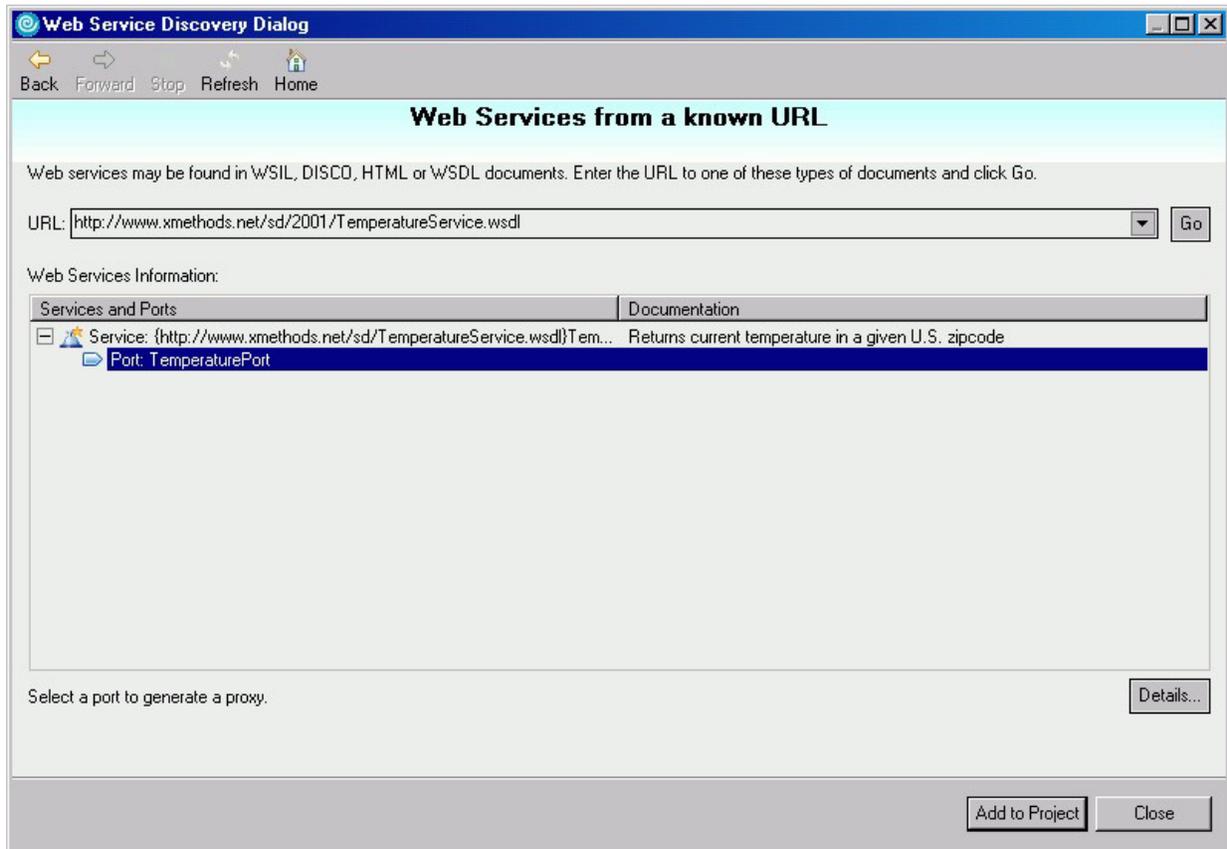
Before you begin, you must complete Lesson 1.2: Create a Faces JSP file.

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.

You can find a Web service and create a proxy for the Web service using the Web Service Discovery wizard:

1. On the **Palette** select the **Data** tab. Select **Web Service**, right-click and click **Insert**.
2. Select **Web services from a known URL**. Although we know the URL for the WSDL file, by using a Web Service Inspection Language (WSIL) file you can search all the Web services which are available. WSIL files are frequently found at a location similar to the following: `http://www.web_service_provider.com/inspection.wsil`
3. Enter the following WSDL URL in the URL field: `http://www.xmethods.net/sd/2001/TemperatureService.wsdl` and click **Go**.

- The Web service we have selected is now displayed in the table. Click **Port: TemperaturePort** to select it.



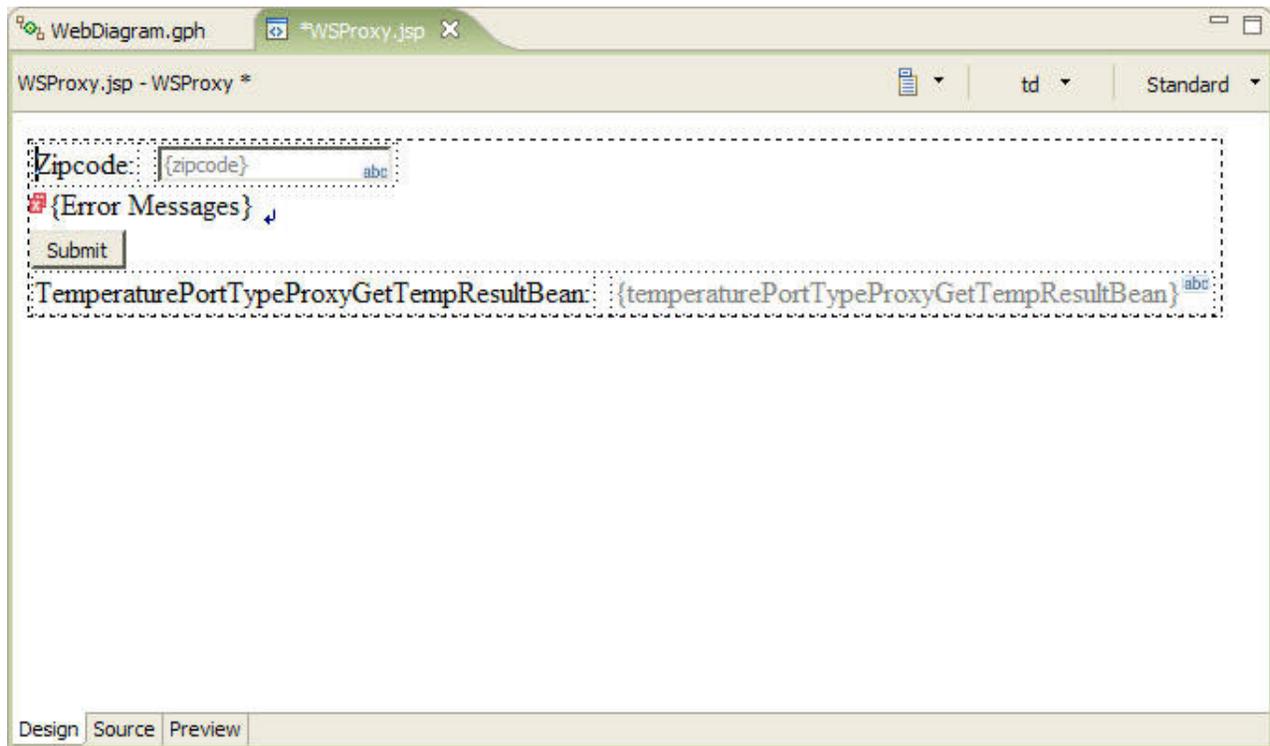
To test the Web service, click **Details** to launch the Web service in the Web Services Explorer. For more information about using the Web Services Explorer, refer to the online help.

- 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**.
- The Web service you selected is now listed in the list of existing Web services. When you select it the only method available for this Web service is listed in the methods list. Click **Finish** to add it to your JSP.

The proxy should now be shown in the Page Data view. If the Page Data view is not displayed you can open it by clicking **Window** → **Show View** → **Page Data**:



Additionally, the method and the action to invoke it have been added to the JSP file. It should look similar to the following:



Running the proxy on the server

1. Save WSProxy.jsp by clicking **File** → **Save**. Close the WSProxy.jsp editor.
2. In the Project Navigator, right-click **WSProxy.jsp** and click **Run** → **Run on Server**.
3. Select the server that is associated with the dynamic Web project, and click **Finish**. If you want to run the Faces JSP on another server, you would need to re-target the Web project that you created to the new server.

Once the JSP is running on the server, it 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**.

Lesson checkpoint

Finish your tutorial by reviewing the materials in the Summary.

Feedback

Summary

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

Lessons learned

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 WSDL 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.

Additional resources

For more information about Web services, WSDL, and SOAP, consult the online help (**Help** → **Help Contents**). For more in-depth technical articles on Web services, consult DeveloperWorks.

Feedback