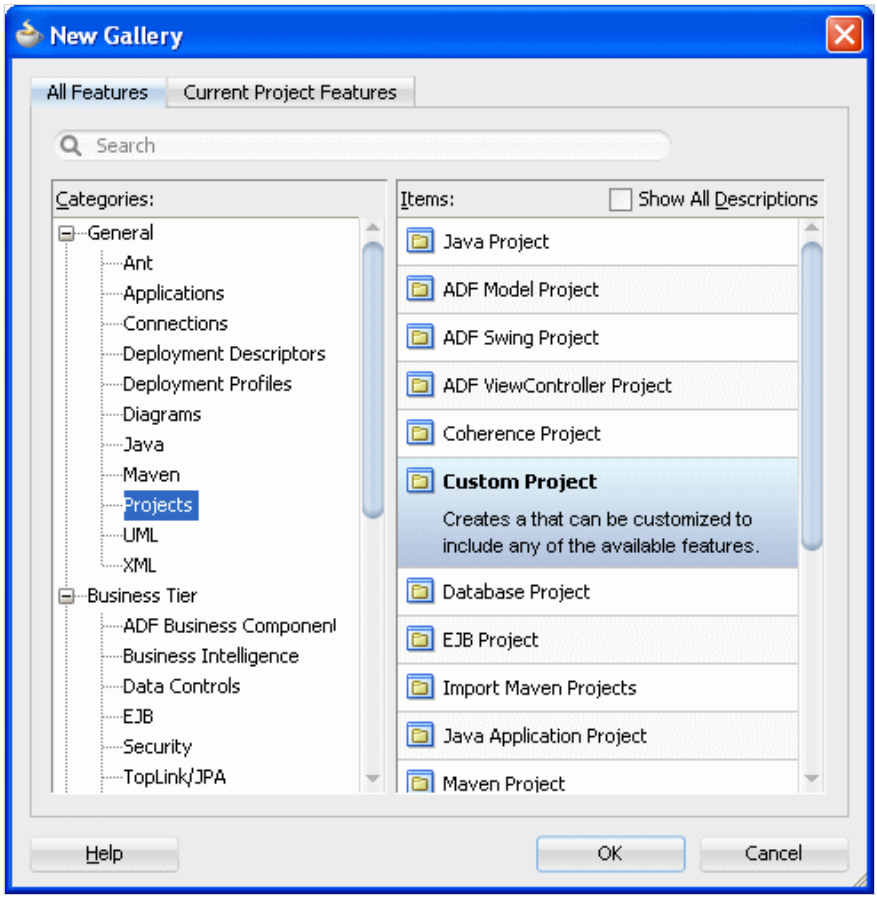
Part 4: Building an Annotation Driven EJB Web Service

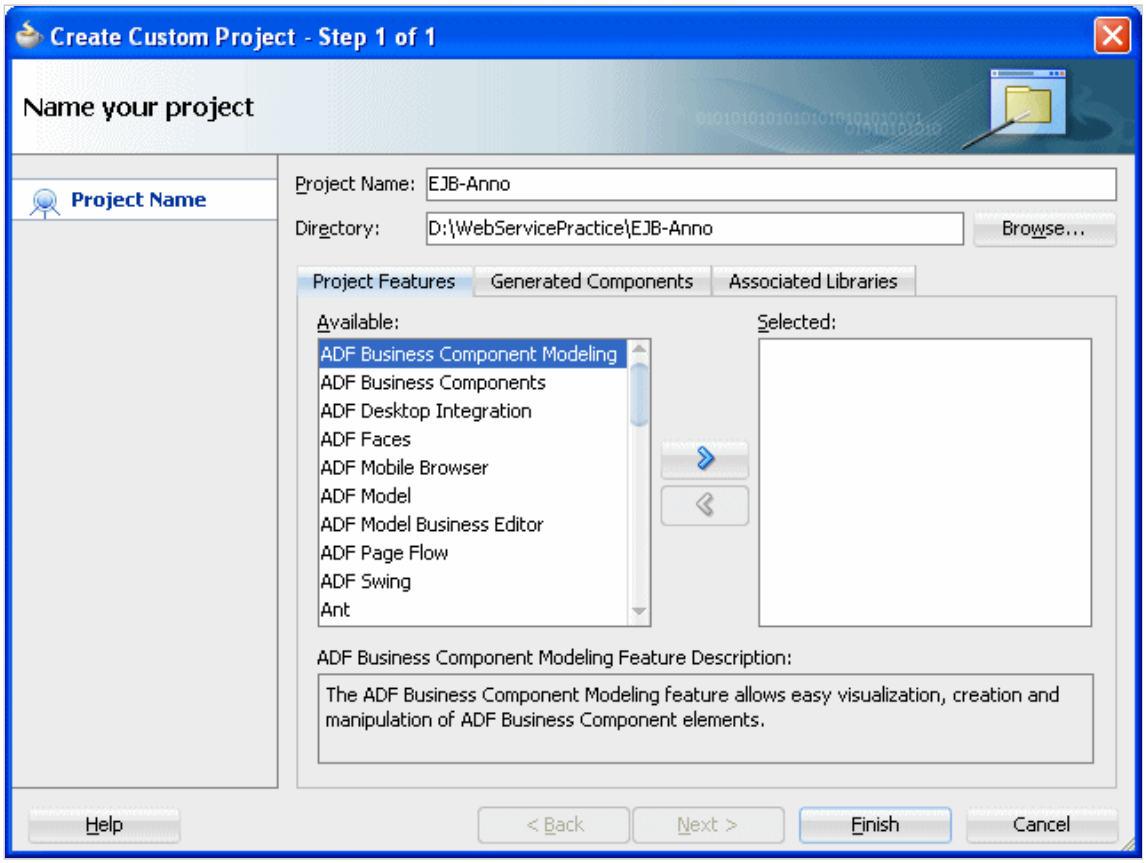
You can use any Java class as a Web Service, even an EJB Session Bean.

Step 1: Creating the EJB Web Service

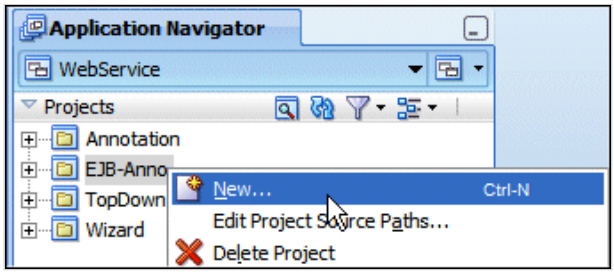
1. Create a new empty project. Right-click the Annotation project node and select **New**. In the New Gallery, click the **All Features** tab and then select **General > Projects** in the Categories list and **Custom Project** in the Items list. Click **OK**



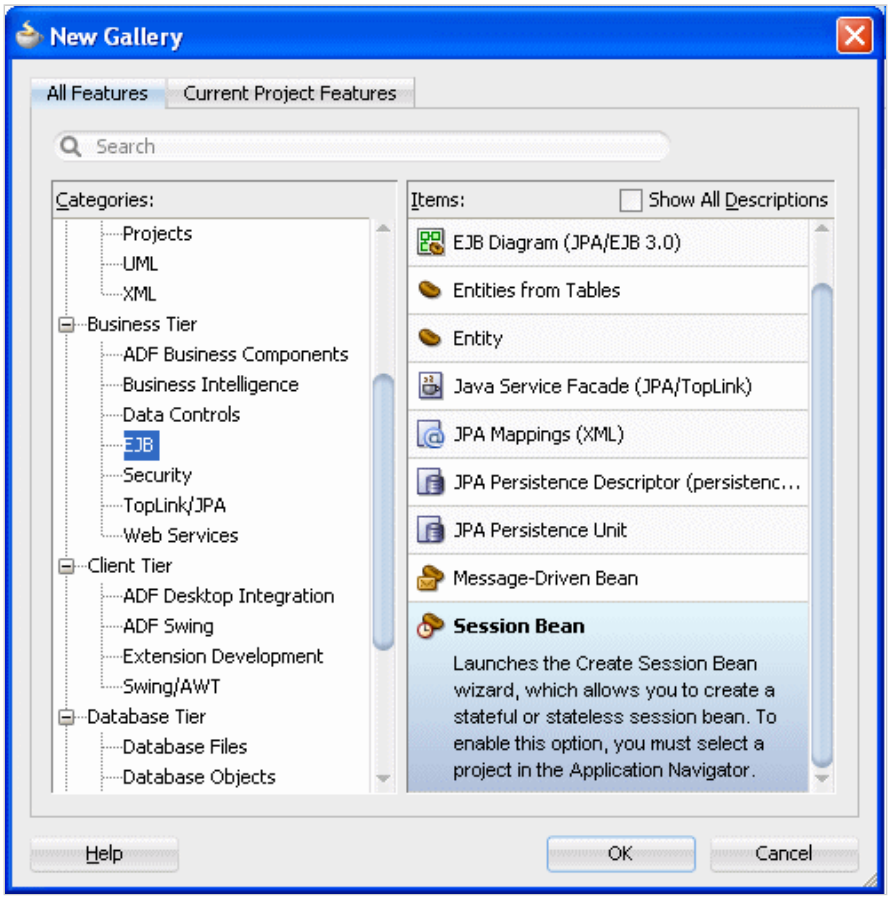
2. Name the new project **EJB-Anno**, leaving other values at their default. Click **Finish**.



3. Right-click the **EJB-Anno** project and select **New**.

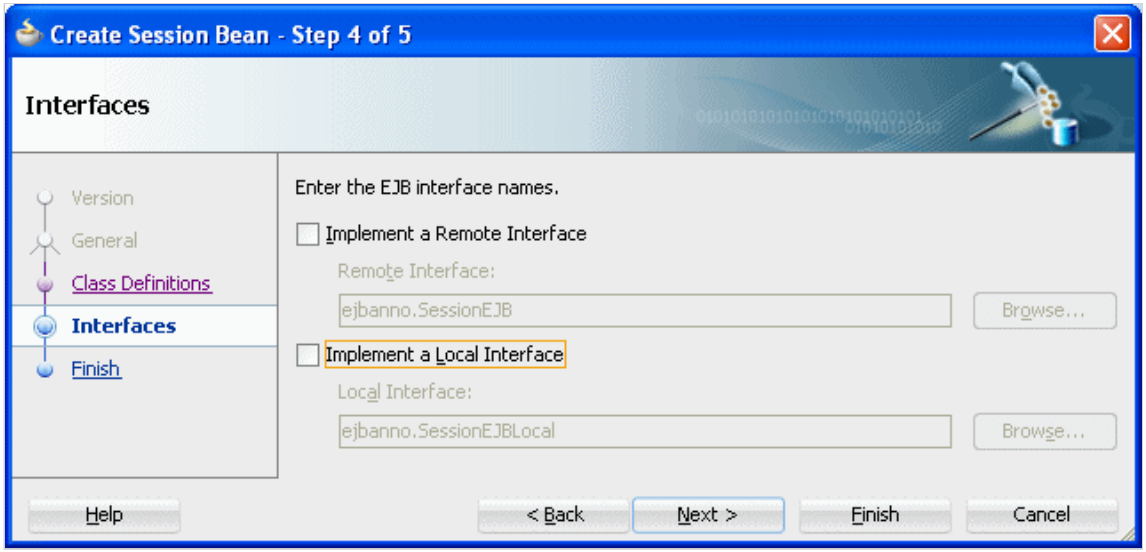


4. In the New Gallery, expand the **Business Tier** node and select **EJB** in the Categories list . In the Items column, select **Session Bean** and click **OK**.



5. In the new Session Bean Wizard, click **Next** up to Step 4. Accepting the defaults for the **EJB Version (Enterprise JavaBeans 3.0)**, **Name** and **Class Definition** pages.

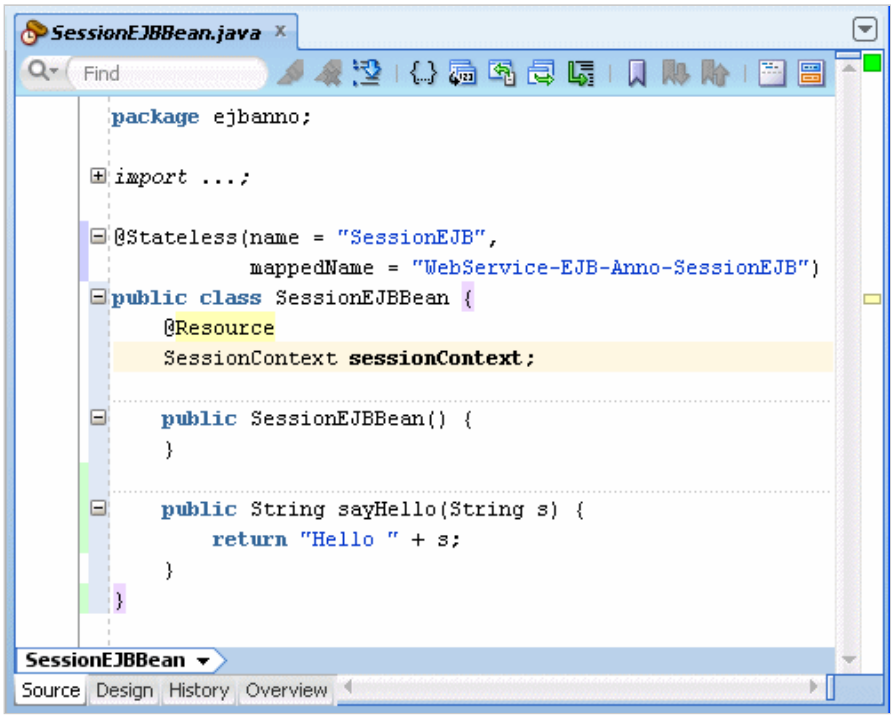
6. In the **EJB Home and Component Interfaces** page, deselect both check boxes so that no interfaces are implemented. Click **Next**, then **Finish** to create the bean.



7. The **SessionEJBBean.java** file opens in the editor. Add the same `sayHello()` method as in previous scenarios:

```
public String sayHello (String s) {
     return "Hello " + s;
}
```

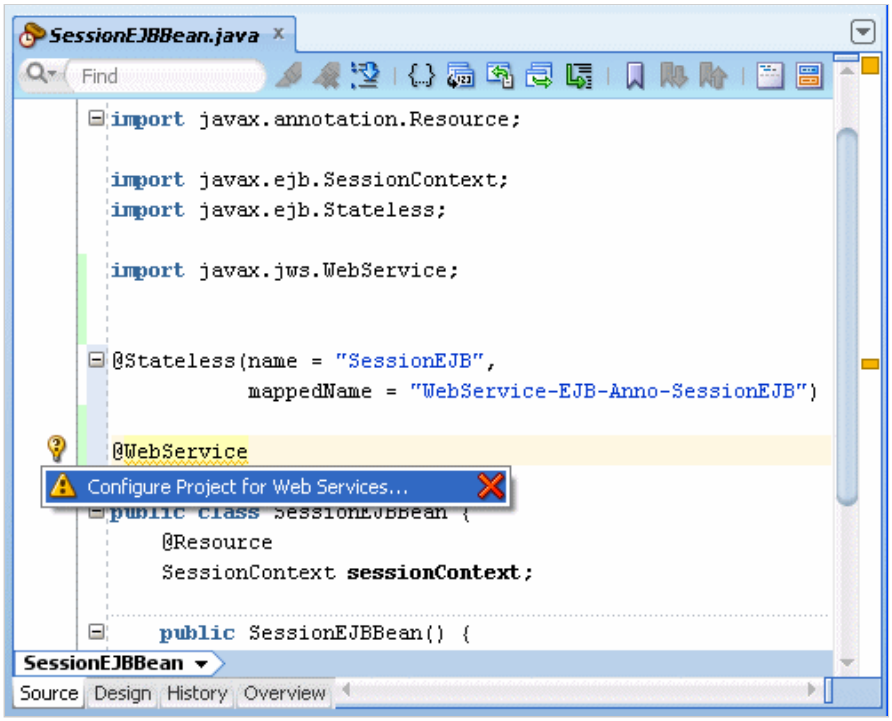
The class should now look like:



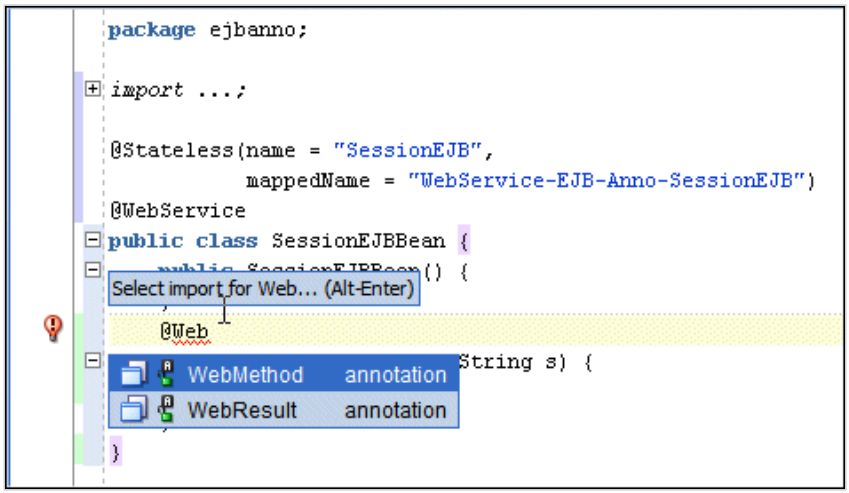
8. Click **Save All** to save your work.

   At this point you just have a class with simple business logic to return the word Hello, followed by the value entered for a parameter. Next you add the web service annotations as you did before.

9. Above the class definition add the **@WebService** annotation. JDeveloper will automatically add the import for javax.jws.WebService. Click the @WebService line of code and then click the lightbulb in the left margin. Sselect **Configure project for web services** to configure the rest of the project for web services.



10. Above the sayHello method, add the **@WebMethod** annotation, just as you did with the POJO. If requested, press [Alt]+[Enter] to add the import statement (although this statement may be added automatically.)

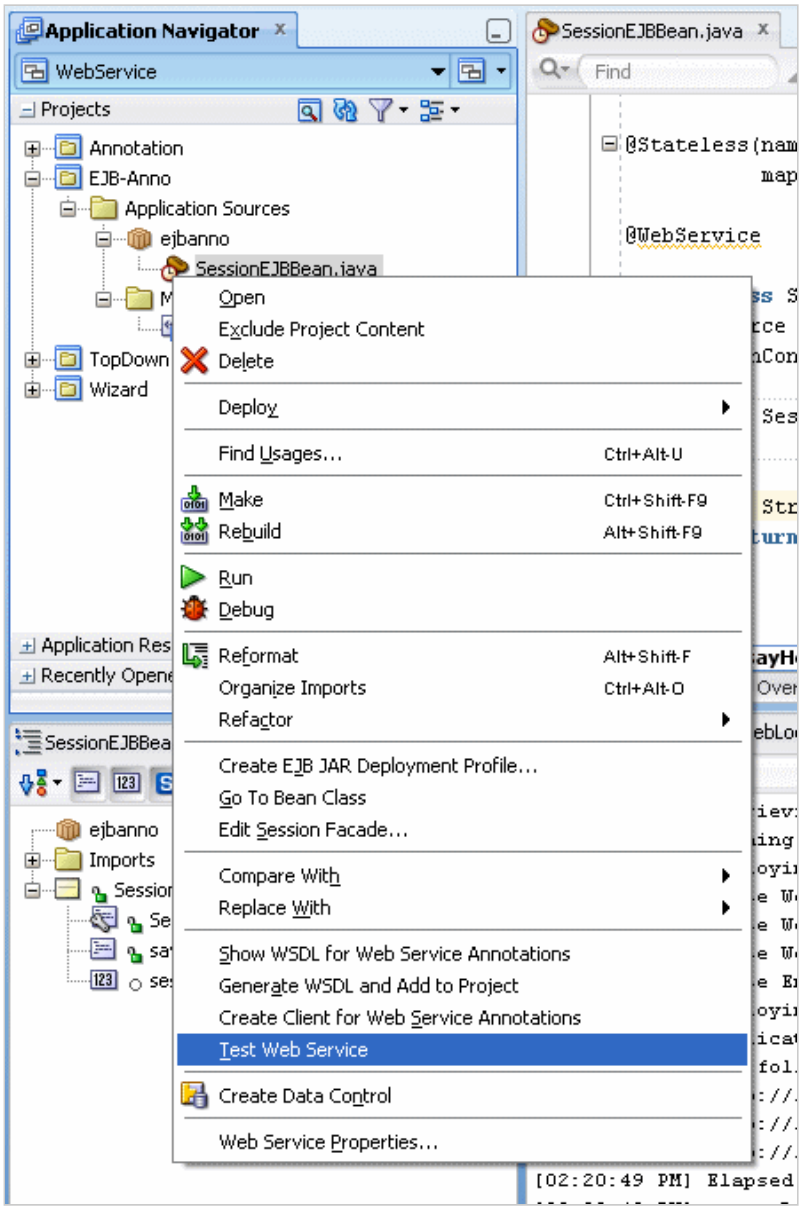


11. Click **Save All** to save your work.

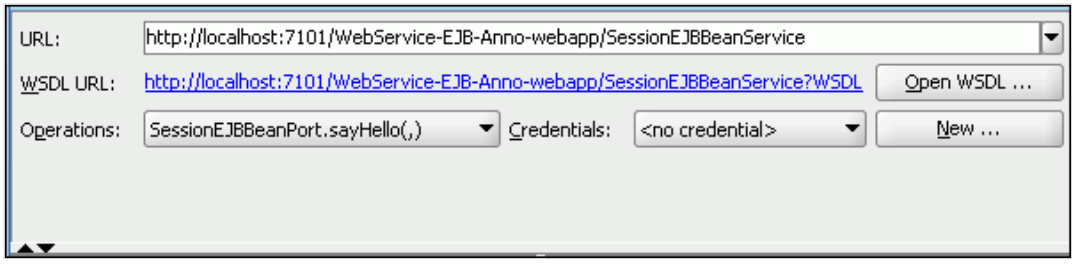
Step 2: Testing the Web Service

In this section you compile, deploy and test the web service. Just as before, you use the HTTP Analyzer for testing the web service. When you test web services using the analyzer, the service is compiled and deployed to the integrated server. The analyzer is then invoked, enabling you to send and receive values from the web service.

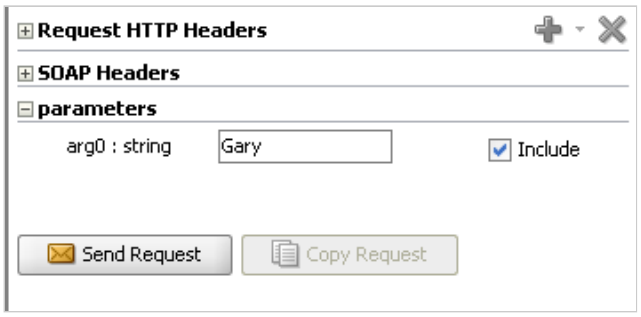
1. In the Application Navigator, right-click the **SessionEJBBean.java** node and in the context menu, select **Test Web Service**



Like earlier, the top portion of the HTTP Analyzer editor displays the URL for the web service, WSDL URL, and exposed Operations.



2. In the Request area, enter <your name> in the **arg0** field and click **Send Request**.



The analyzer then sends the request to the service, and after a bit the return parameter is displayed.

3. **Close** all the tabs and collapse the **EJB-Anno** project.