

实验二：创建 SOAP Web Services

BY 王鸿吉

目录

1.概述	1
2.JDeveloper 的代码补全功能设置（可选）	1
3.使用 JDeveloper 创建 web services	2
3.1 创建一个 web service	2
3.2 测试 web service.....	12
4.总结和要求.....	13

1.概述

Web service 有两种典型类型：SOAP web service 和 RESTful web service。开发 web service 也有两种模式：（1）自底向上：从 java 开始；（2）自顶向下：从 WSDL 开始。Java API for Web Services (JAX-WS) 是一种基于标准的 API，用于编写、组装和部署 Java web service。本实验介绍利用 JAX-WS 机制，通过创建 java 类实现 SOAP web service 的方法（注：也可以实现 RESTful web service）。

实现的基本思路：

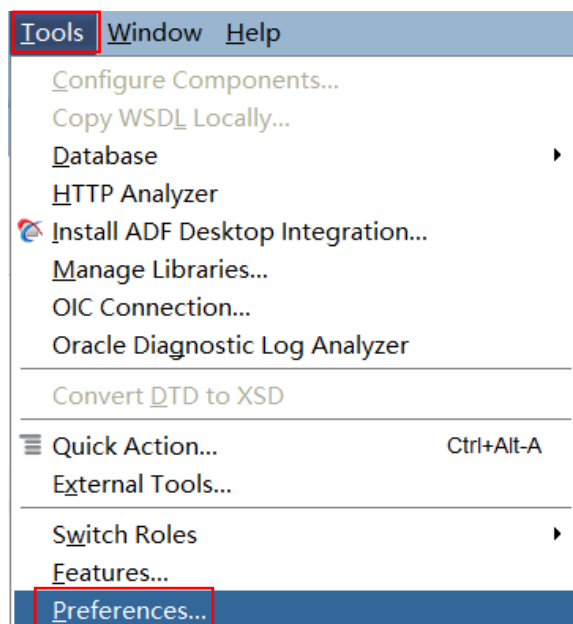
- （1）在 JDeveloper 创建一个 Java 类；
- （2）使用 JDeveloper 向导往该 Java 类中添加@WebService 等必要注释；
- （3）在类的相应方法上添加@WebMethod 并将类方法发布为 web service。

创建一个带注释的 Java 文件>使用 Ant 编译它为 Java 类和产生所有相关的 artifacts

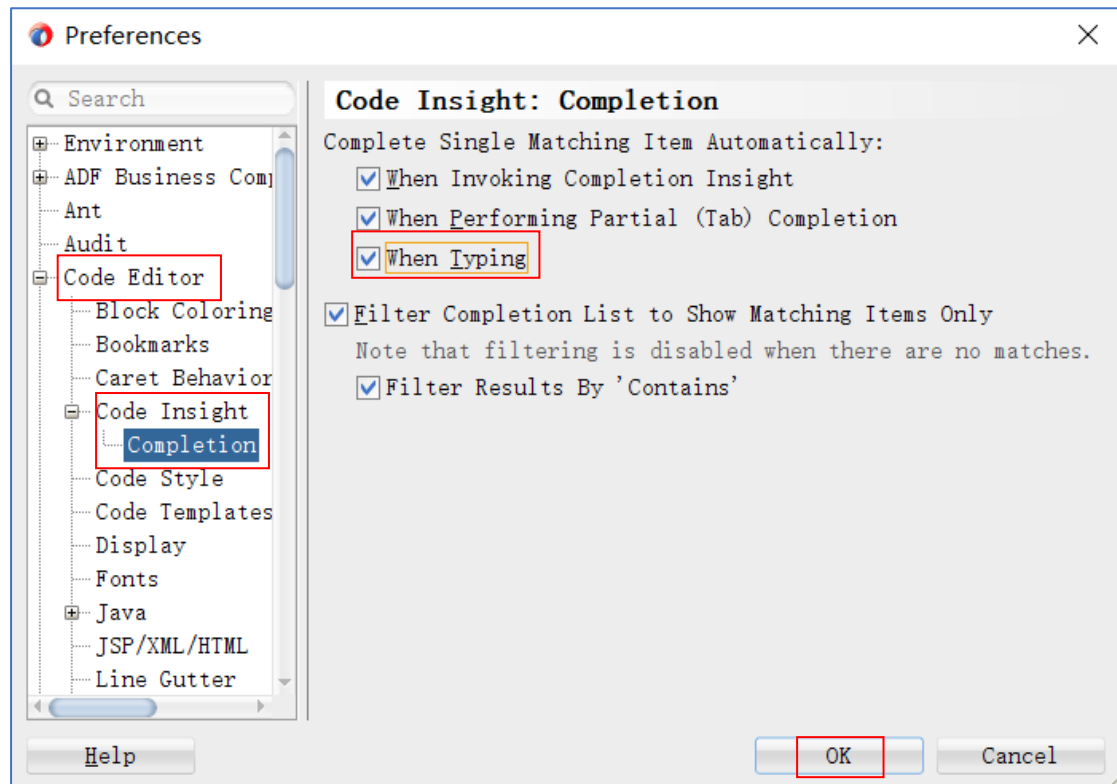
2.JDeveloper 的代码补全功能设置（可选）

当在 JDeveloper 编辑 Java 类时,如果需要键入 annotations 时,JDeveloper 中的 web services code insight 可以帮助实现 annotations 的自动补全功能，其设置方法如下：

打开 JDeveloper > 点击菜单栏上的【Tools】。



点击【Preferences...】。



依次选择左侧栏的【code Editor】>【code insight】>【completion】> 勾选右侧栏的【when typing】>点击【OK】关闭设置。

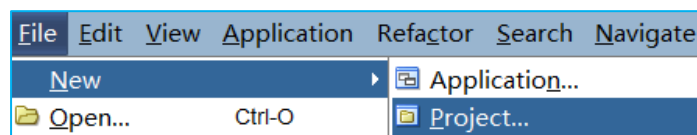
3.使用 JDeveloper 创建 web services

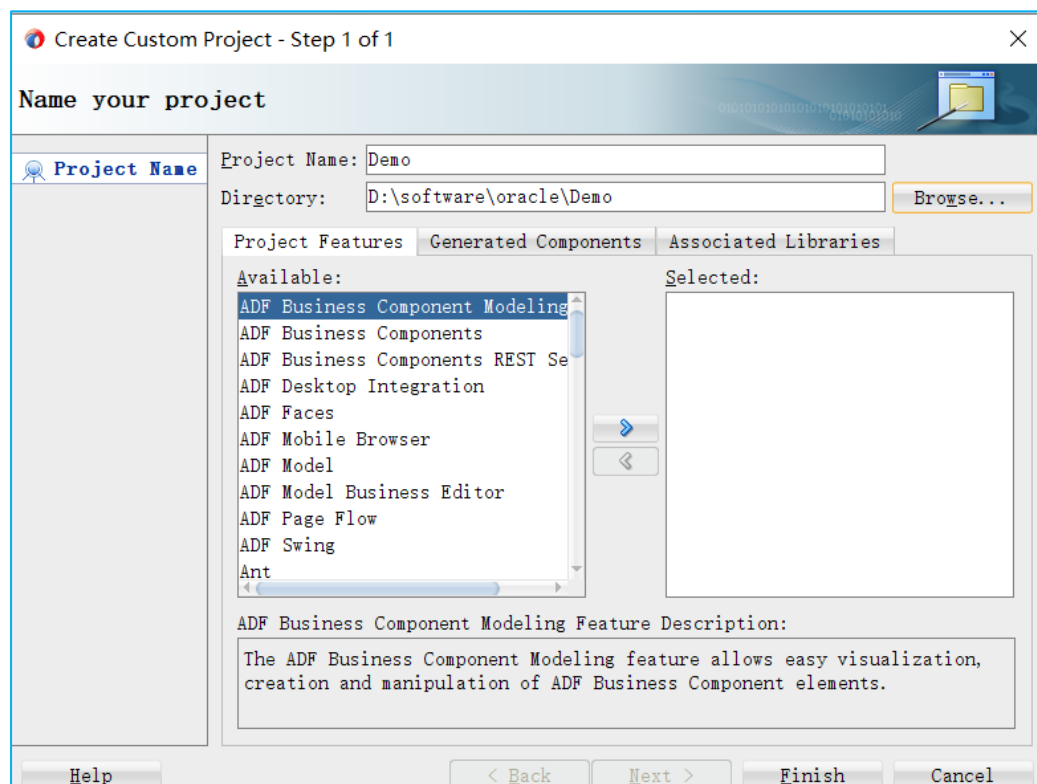
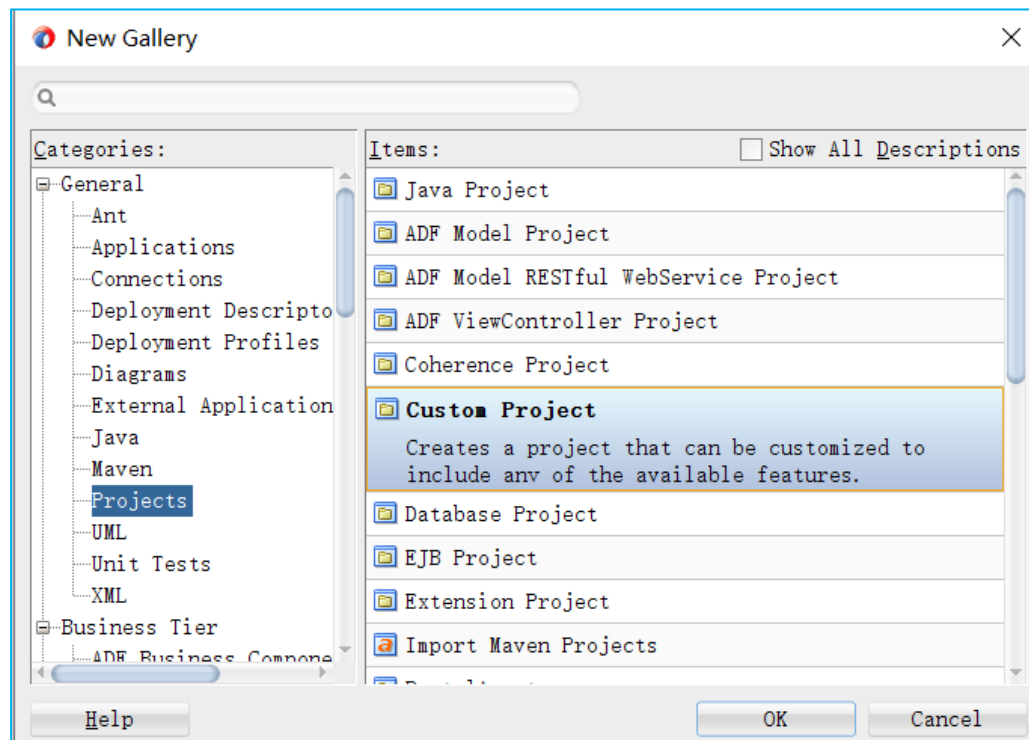
3.1 创建一个 web service

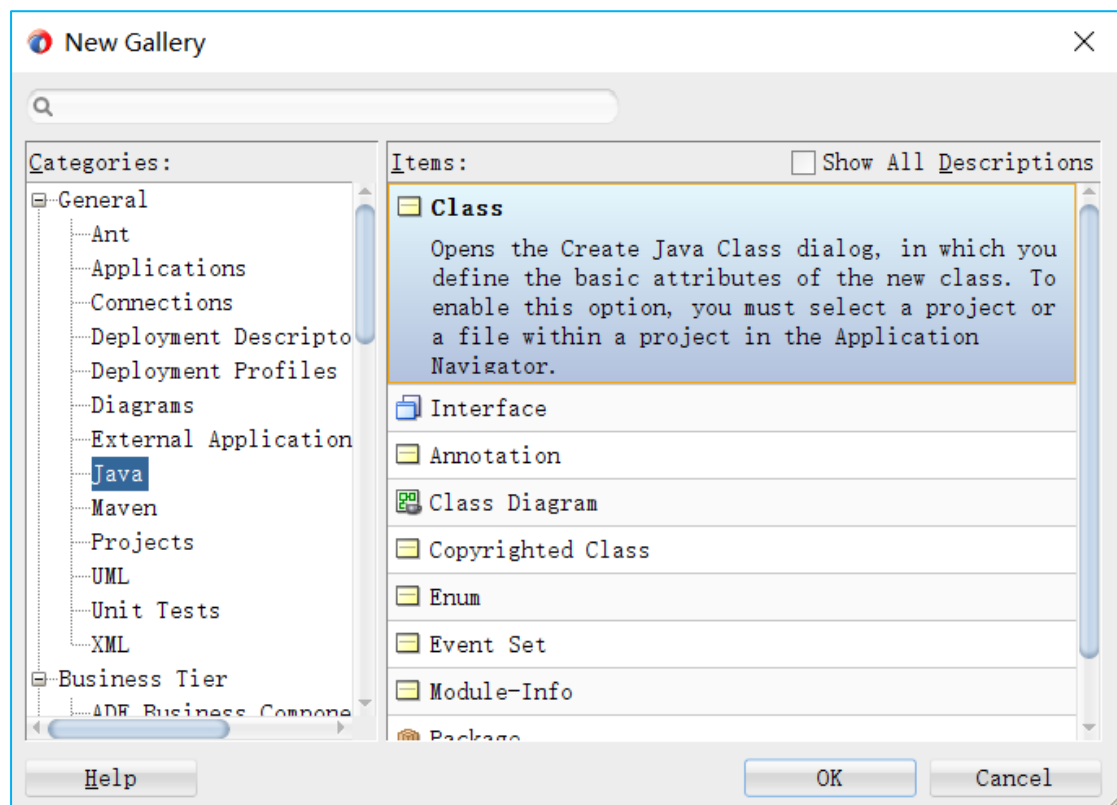
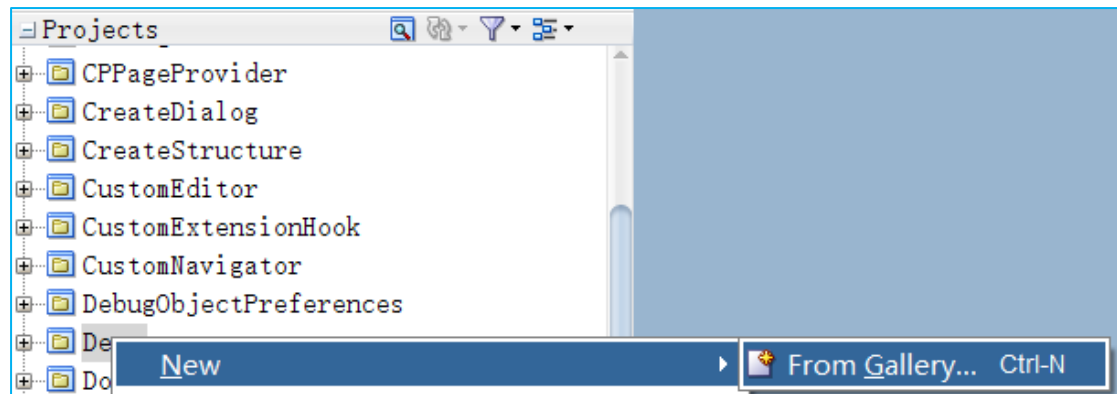
步骤 1：创建一个 Plain Old Java Object（POJO）对象

创建一个新的空项目。

依次点击【File】>【New】>【Project...】。或使用快捷键：Ctrl+N。







Create Java Class

Enter the details of your new class.

Name:

Package:

Extends:

Optional Attributes

Implements:

Access Modifiers

☒ public

☐ package private

Other Modifiers

☒ <None>

☐ abstract

☐ final

☒ Constructors from Superclass

☒ Implement Abstract Methods

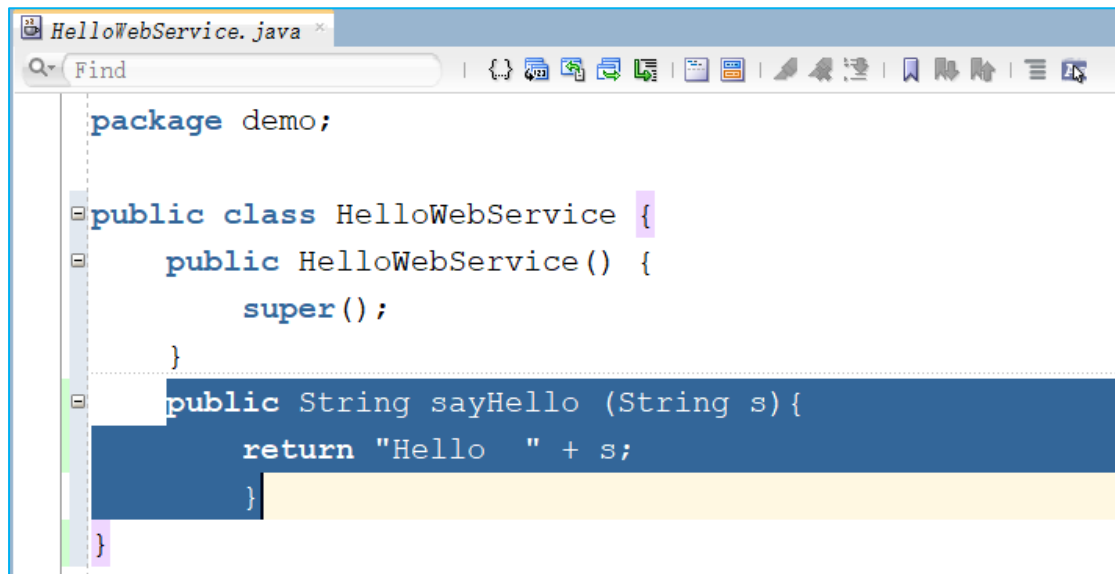
☐ Main Method

Messages:

Help OK Cancel

Creating Java Class

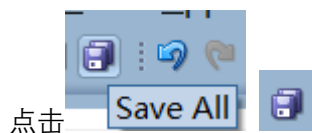
Cancel



```
package demo;

public class HelloWebService {
    public HelloWebService() {
        super();
    }

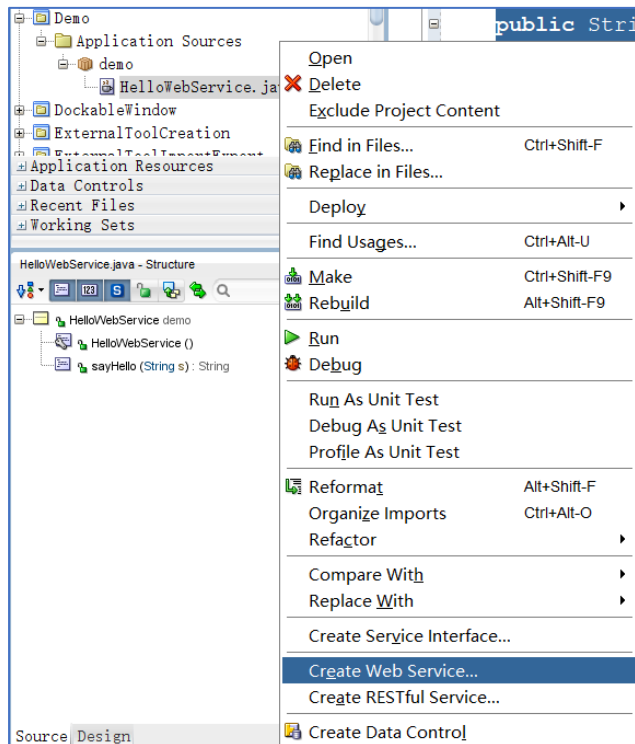
    public String sayHello (String s){
        return "Hello " + s;
    }
}
```



步骤 2: 创建 web service 类

有了 POJO 之后，就可以使用 JDeveloper 创建注释，这些注释是将任何方法发布为 web service 所必须的。

依次选择【Application Sources】>【demo】，右键【HellowebService.java】>【create web service...】。



在创建 java web service 页面，按图选择。

Select Deployment Platform

Platform

☒ **Java EE 6, with support for JAX-WS Annotations**
Support for deploying to 12.1 WLS with Java annotations using JAX-WS annotation specification.

☐ **Java EE 6, with support for JAX-WS RI**
Support for deploying to any container that supports the Sun JAX-WS Reference Implementation.

Help < Back Next > Finish Cancel

点击【next】。

Generation Options

Select or type the Java class or Stateless Session EJB you want to publish (Note: If you type a name for which no class exists, you will be offered the option to have the wizard generate a default implementation class with that name).

Component To Publish:
demo.HelloWebService

Web Service Name:
MyWebService

Port Name
MyWebServicePort

☐ Add SEI

Help < Back Next > Finish Cancel

- Web service name 可修改，修改时 port name 也随之同步修改。

点击【next】。

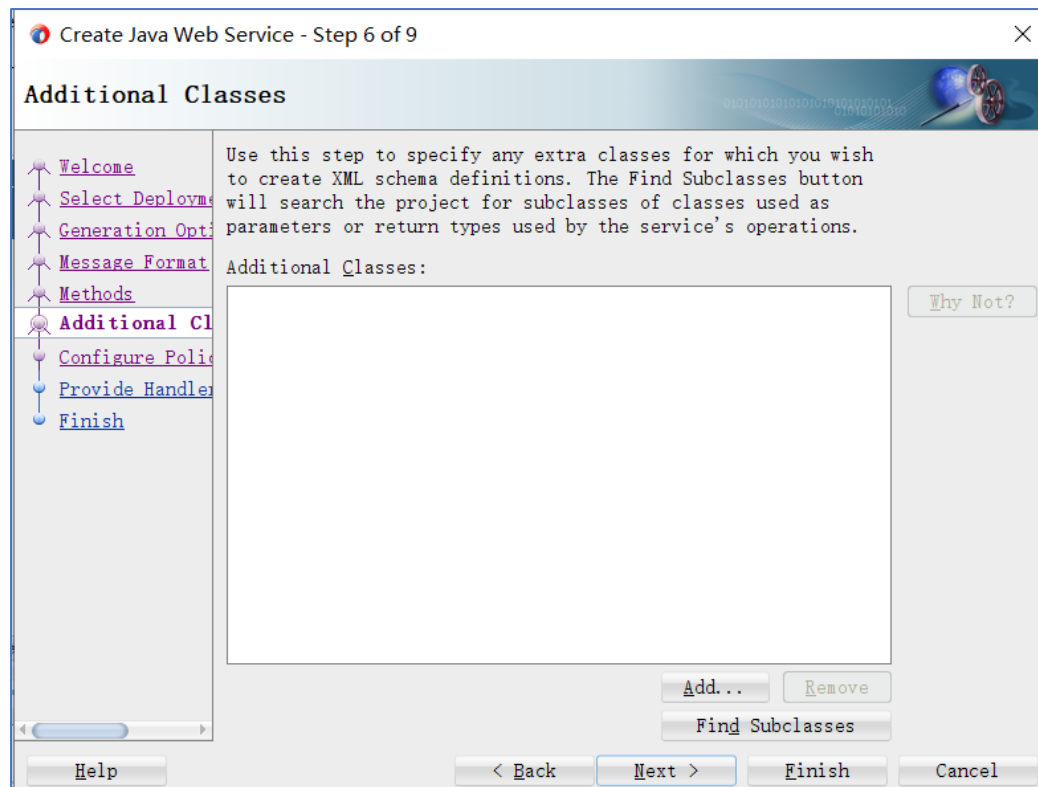
The screenshot shows the 'Create Java Web Service - Step 4 of 10' dialog box. The 'Message Format' tab is selected in the left sidebar. The main area contains instructions: 'Use the controls on this page to choose the binding option and the settings that control the structure of the SOAP messages transmitted to and from the web service.' There are two checkboxes: 'SOAP 1.1 Binding' (unchecked) and 'SOAP 1.2 Binding' (checked and highlighted with a red box). Below this is a dropdown menu for 'SOAP Message Format' set to 'Document/Wrapped'. Further down, there is text about MTOM: 'Binary encoding style for this service is MTOM. Enable MTOM to write MTOM annotation for this service.' with an unchecked 'Enable MTOM' checkbox. At the bottom, there is text: 'Select to enable Fast Infoset for this service' with an unchecked 'Enable Fast Infoset' checkbox. The bottom of the dialog has buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

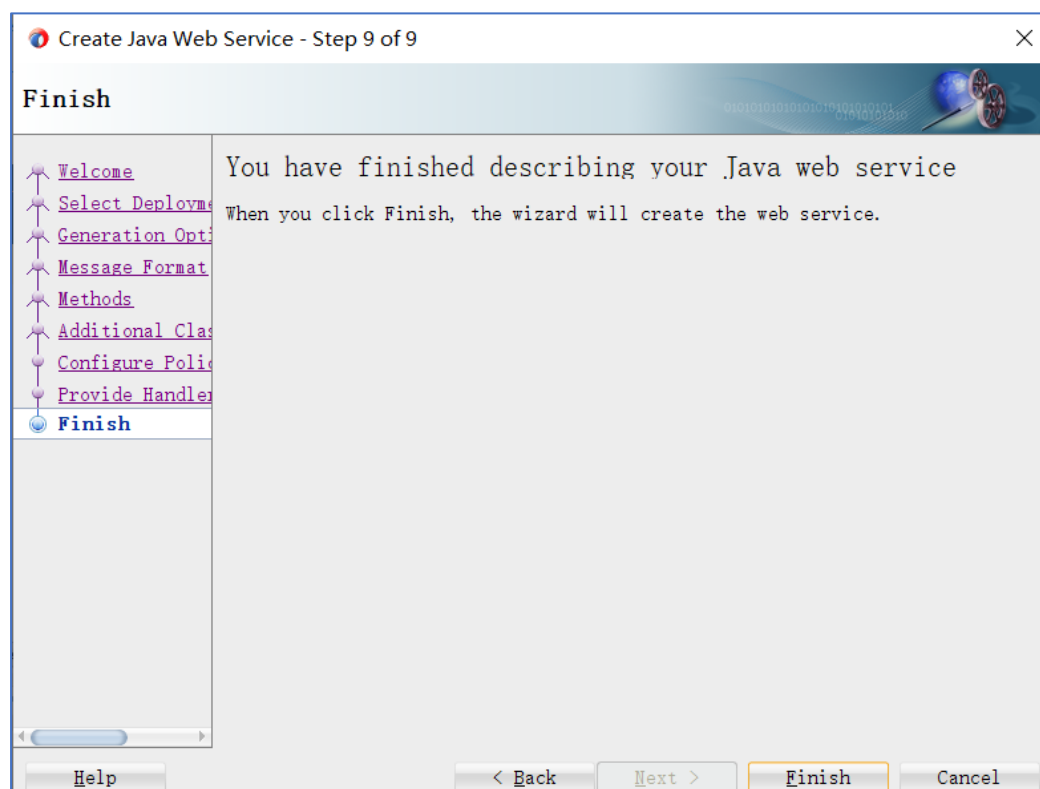
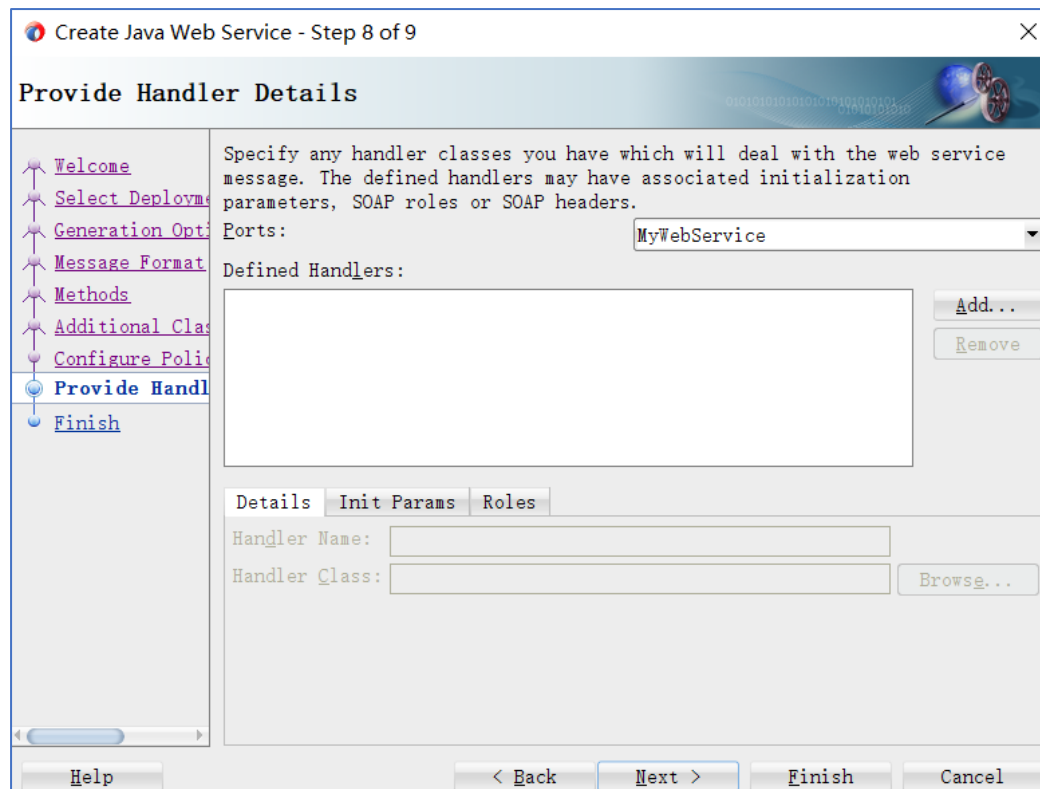
点击【next】。

The screenshot shows the 'Create Java Web Service - Step 5 of 10' dialog box. The 'Methods' tab is selected in the left sidebar. The main area contains instructions: 'Select at least one method for the web service to expose.' Below this is a list titled 'Available Methods:' with one entry: 'sayHello(java.lang.String s) : java.lang.String', which is checked. To the right of the list are three buttons: 'Why Not?', 'Select All', and 'Deselect All'. The bottom of the dialog has buttons for 'Help', '< Back', 'Next >', 'Finish', and 'Cancel'.

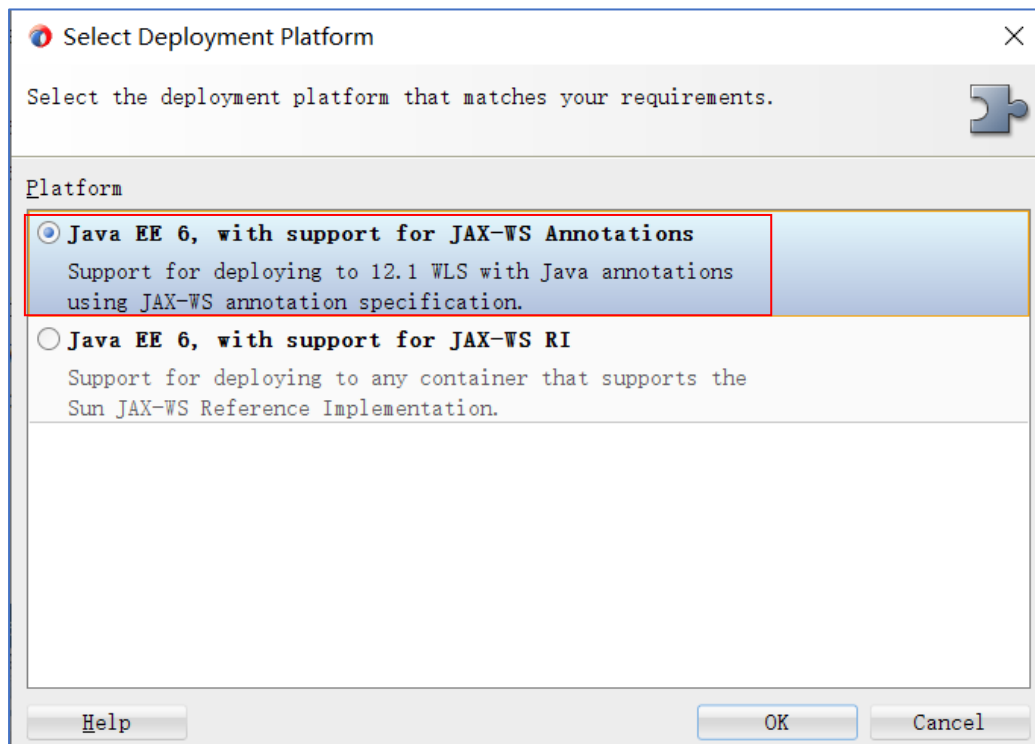
- 所有可发布的方法都将在此显示，选择其中一种。

点击【next】。



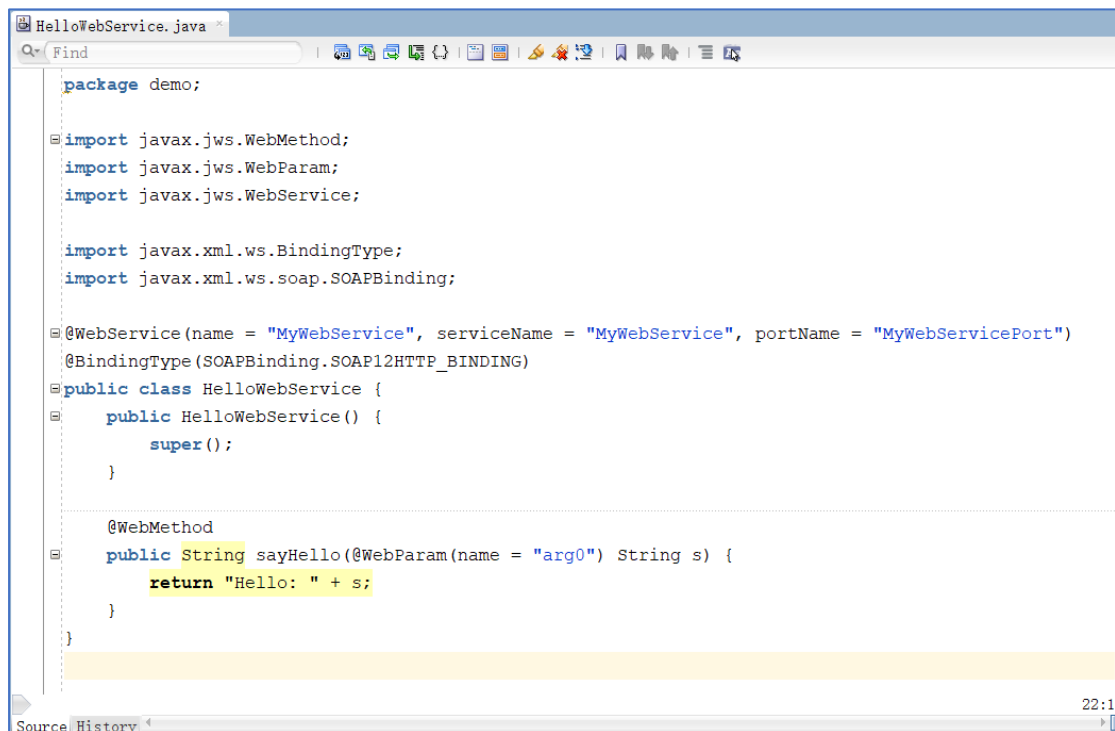


点击【finish】后进入选择部署平台。



点击【OK】。

原 java 类中添加了新的注释 annotations: @WebService、@BindingType、@WebMethod 和 @WebParam。



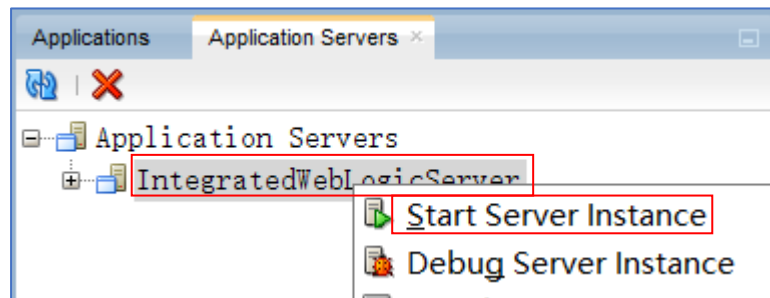
点击  保存所有工作。

3.2 测试 web service

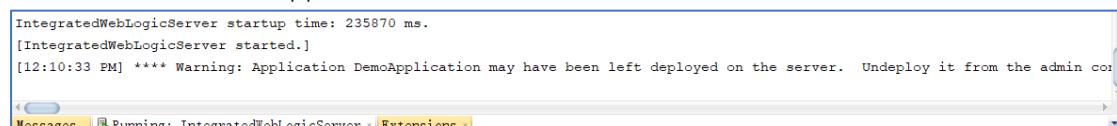
测试工具：JDeveloper 的 HTTP Analyzer。当使用 HTTP Analyzer 进行测试时，web service 自动在 WebLogic 服务器中被编译和部署，Analyzer 可以发送和接收来自 web service 的值。

步骤 1：启动 IntegratedWebLogicServer

依次选择【Application servers】>【IntegratedWebLogicServer】>【Start server instance】。

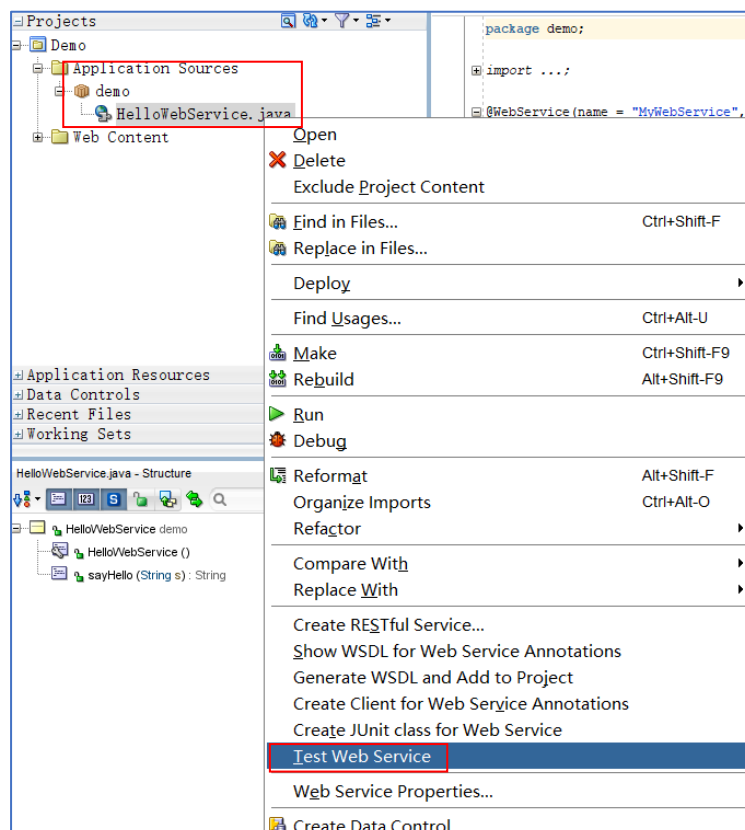


- 若未出现【Application Servers】标签，可按快捷键 Ctrl+Shift-G 或选择菜单栏的【windows】>【Application Servers】

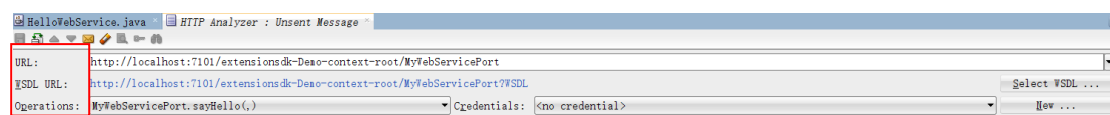


步骤 2：测试 web service

在应用窗口右键【HelloWebService.java】>【Test Web Service】。



在 HTTP Analyzer 编辑器中显示了 web service 的 URL，WSDL URL 和暴露的操作。



在请求区域，在 arg0 字段输入名字（如，厦门大学）点击【send request】，在响应区域就可以看到结果。



至此，自底向上创建 web service 的方法就结束了。

4.总结和要求

本实验介绍在 JDeveloper 12c 中如何采用自底向上模式创建和测试 web service。

要求：

- 理解并分析 HTTP Analyzer 编辑器中显示的 web service 的 URL，WSDL URL 和暴露的操作。