# 北京邮电大学软件学院

# 2020-2021学年第1学期实验报告

**课程名称： XML & Web Service**

**实验名称： 实验3 Web Server 实验**

**实验完成人：**

**指导教师：**\_\_\_\_\_\_\_\_\_\_吴国仕、傅湘玲\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

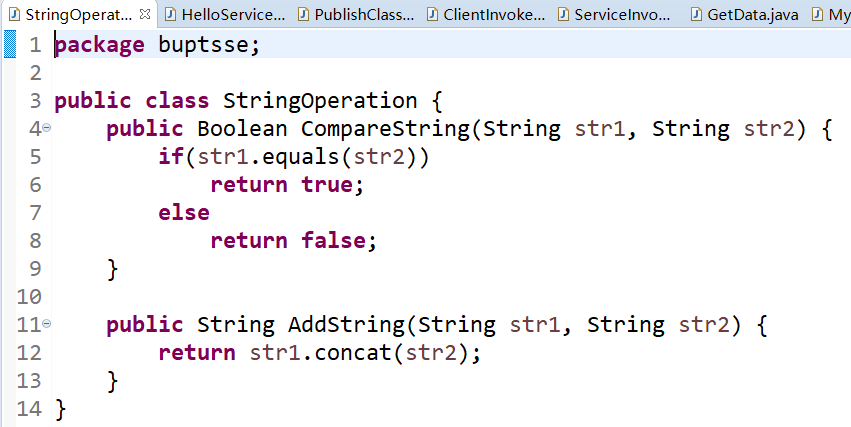
**日 期： 2020 年 12 月 12 日**

1. **实验目的**
2. 学习了解Web Service、SOAP协议、WSDL等原理以及使用规范；
3. 掌握Web Service项目基于JAX\_WS package以及JDK1.6+版本的开发过程；
4. 学会使用Web Service开发的相关开发工具。
5. **实验内容**
   1. 创建一个Web Service项目，要求具有以下功能：
      1. 输入字符串string1和字符串string2，比较string1和string2是否相同；
      2. 字符串合并AddString功能：能够将string1和string2合并为一个字符串。
   2. 此Web Service项目需要分别实现基于JAX\_WS package的开发和基于JDK1.6+版本的开发
6. **实验要求**
7. 使用JAX\_WS package 来实现上面两个功能，要求写出服务端和客户端：
   1. 使用SOAPUI工具查看服务端和客户端信息交换过程中的SOAP request、response 以及WSDL文件
   2. 使用MyEclipse的Web Service Explorer查看服务端和客户端信息交换的过程
8. 使用JDK1.6+版本环境来实现上面两个功能，要求写出服务端和客户端：
   1. 使用SOAPUI工具查看服务端和客户端信息交换过程中的SOAP request、response 以及WSDL文件
   2. 使用MyEclipse的Web Service Explorer查看服务端和客户端信息交换的过程
9. **实验环境**

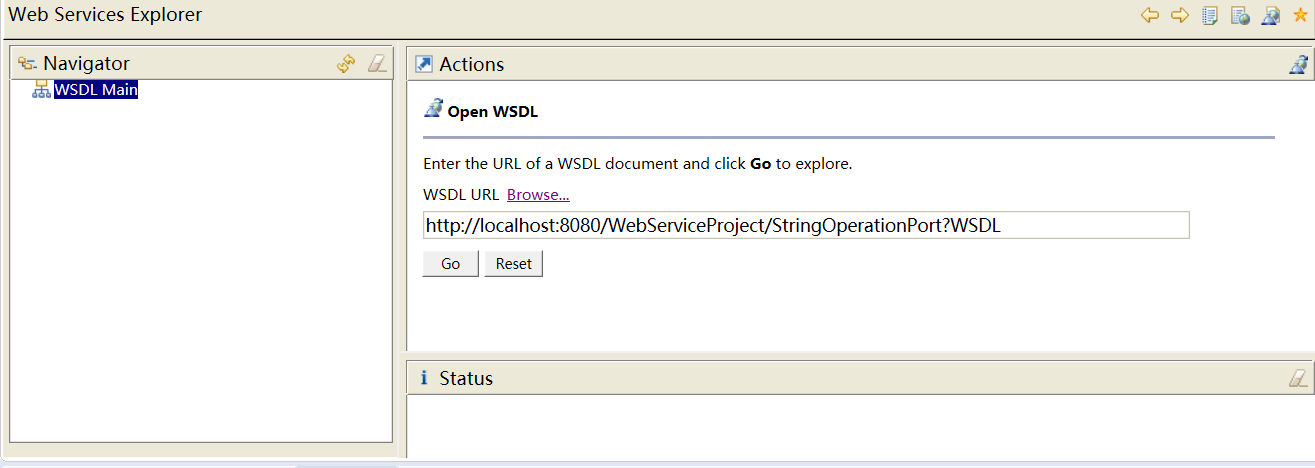
Windows10 Java语言 MyEclipse IDE

1. **实验结果**
2. **创建WebServiceProject项目, 建立AddString和CompareString两个服务**

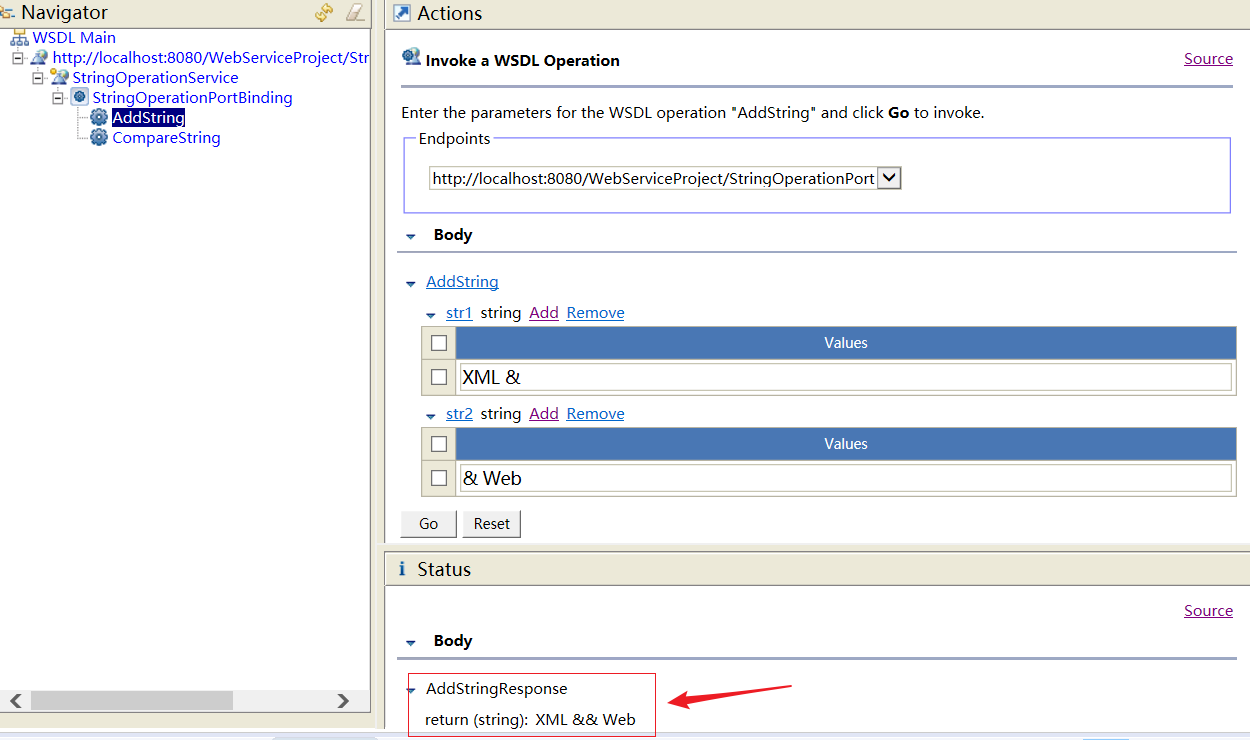
在项目src目录下创建buptsse包，之后创建StringOperation类，代码如下：



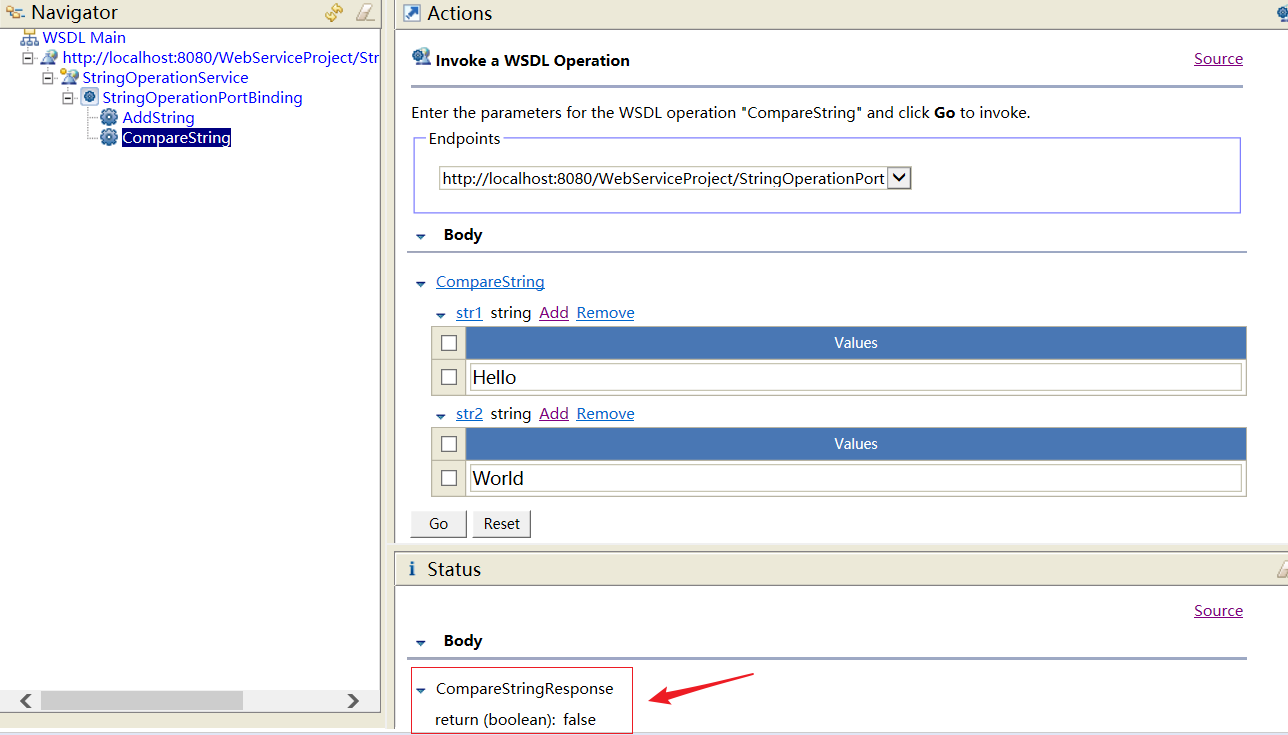
**WebServer测试：输入WSDL URL**



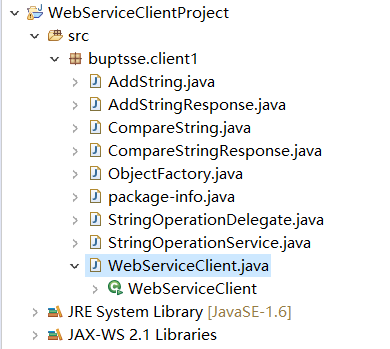
**字符串相加**



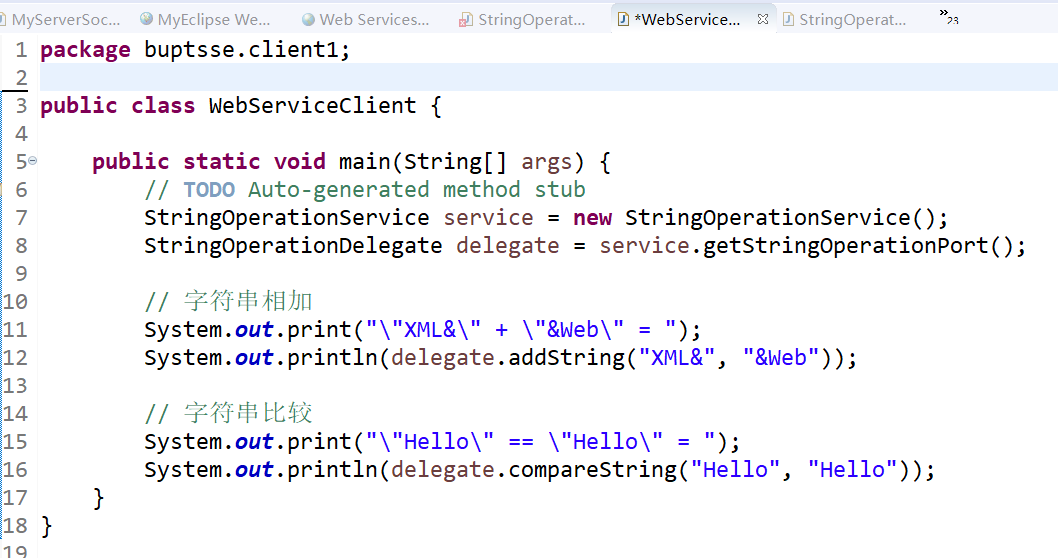
**字符串比较**



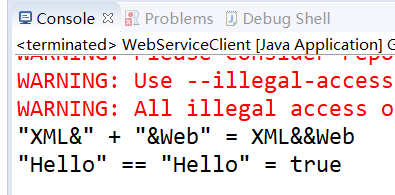
1. **创建Web服务客户端直接与Web服务及其所有公开的操作进行交互**



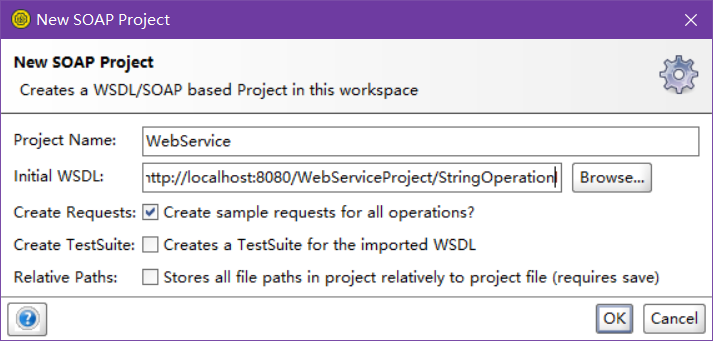
**创建WebServiceClient类，写出如下代码，使用Web服务**

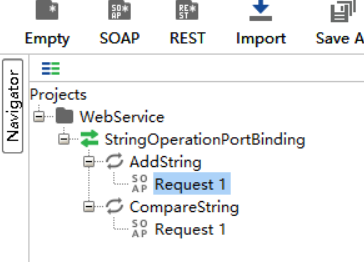


**运行代码，结果如下；**

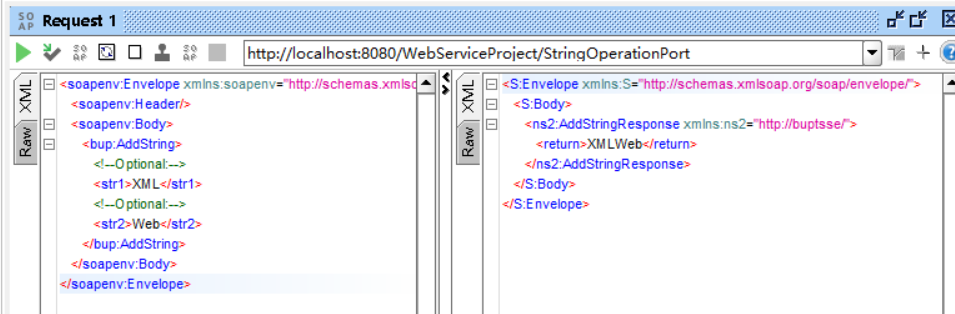


1. **在SOAPUI中创建的SOAP项目，测试Web Service**

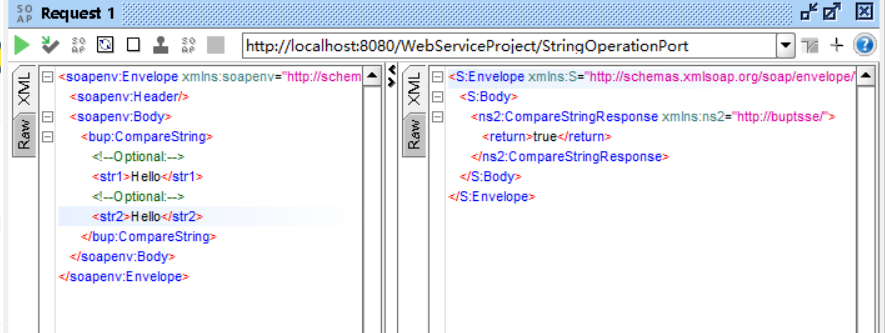
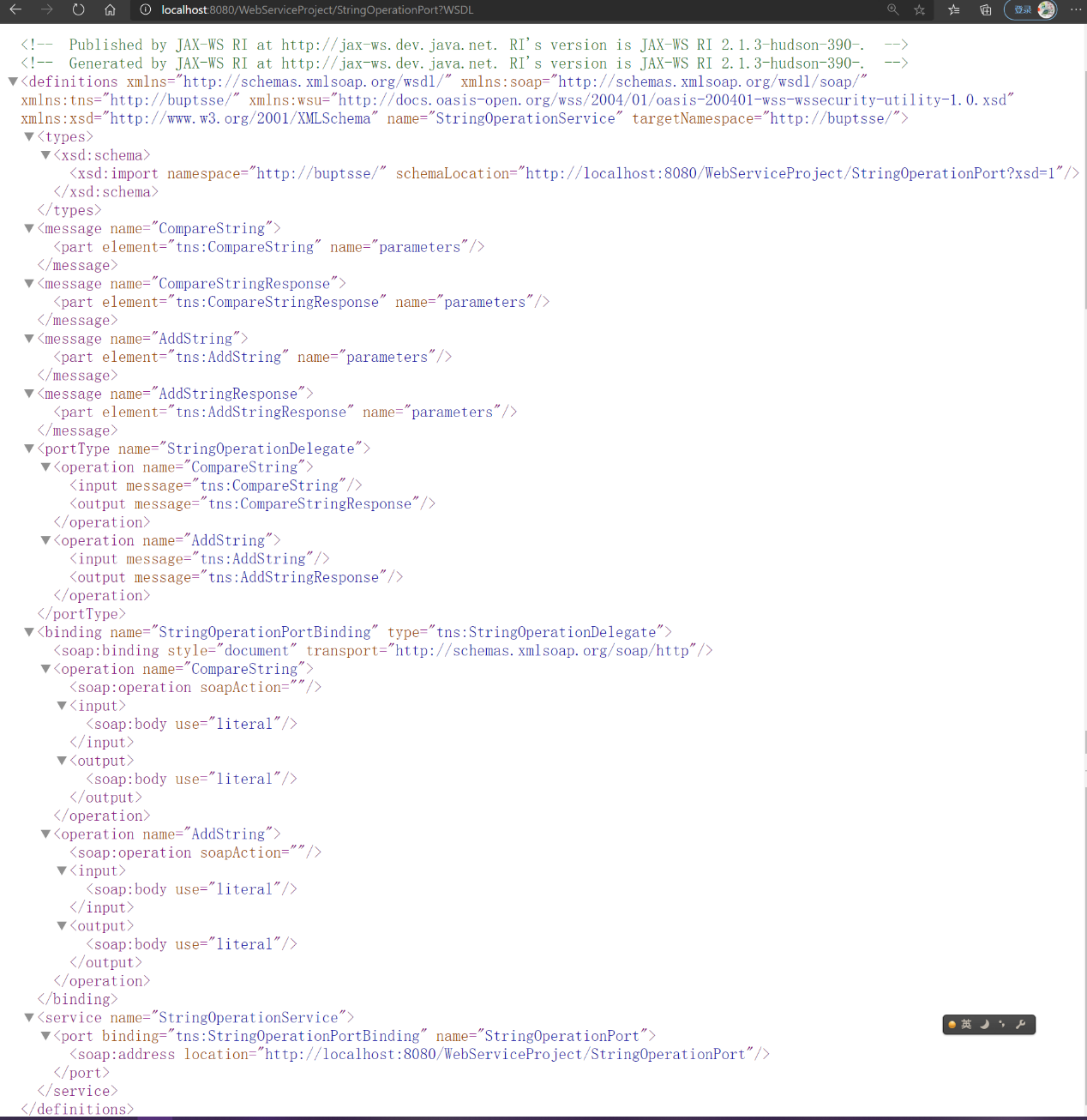




**测试AddString，结果如下：**

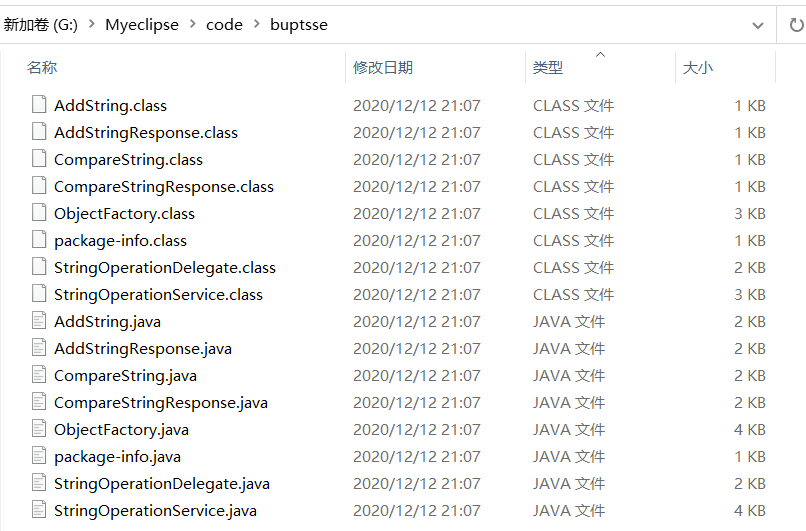


**测试CompareString，结果如下：**

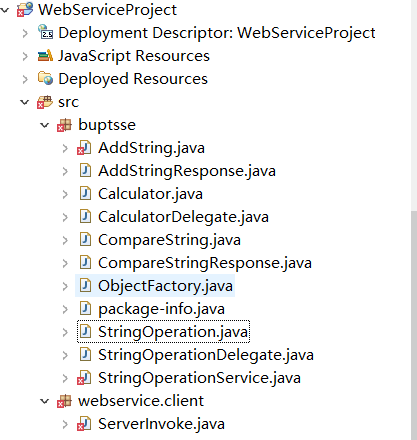
**打开浏览器，键入http://localhost:8080/WebServiceProject/StringOperationPort?WSD****L，查看生成的wsdl：**

1. **在命令行中生成代码**

**生成的代码如下：**



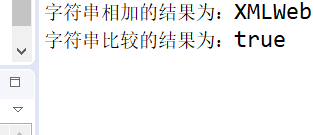
**将生成的代码放到项目中并新建webservice.client包，在保中新建ServerInvoke类**



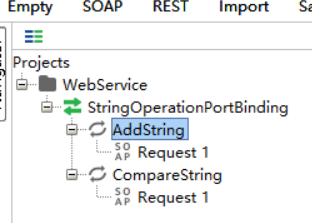
**ServerInvoke类代码如下：**



**运行代码结果如下：**



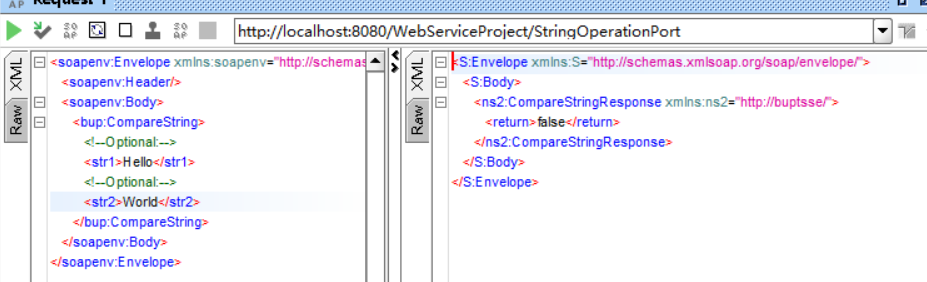
1. **在SOAPUI中创建的SOAP项目，再次进行测试Service**



**测试AddString，结果如下：**



**测试CompareString，结果如下：**



**再次打开浏览器，键入**[**http://localhost:8080/WebServiceProject/StringOperationPort?WSD**](http://localhost:8080/WebServiceProject/StringOperationPort?WSD)**，结果如下：**



1. **实验源代码**

StringOperation类

1. **package** buptsse;
3. **public** **class** StringOperation {
4. **public** Boolean CompareString(String str1, String str2) {
5. **if**(str1.equals(str2))
6. **return** **true**;
7. **else**
8. **return** **false**;
9. }
11. **public** String AddString(String str1, String str2) {
12. **return** str1.concat(str2);
13. }
14. }

WebServiceClient类

1. **package** buptsse.client1;
3. **public** **class** WebServiceClient {
5. **public** **static** **void** main(String[] args) {
6. // TODO Auto-generated method stub
7. StringOperationService service = **new** StringOperationService();
8. StringOperationDelegate delegate = service.getStringOperationPort();
10. // 字符串相加
11. System.out.print("\"XML&\" + \"&Web\" = ");
12. System.out.println(delegate.addString("XML&", "&Web"));
14. // 字符串比较
15. System.out.print("\"Hello\" == \"Hello\" = ");
16. System.out.println(delegate.compareString("Hello", "Hello"));
17. }
18. }

**ServerInvoke类**

1. **package** webservice.client;
2. **import** java.net.URL;
3. **import** javax.xml.namespace.QName;
4. **import** javax.xml.ws.Service;
5. **import** buptsse.StringOperation;
7. **public** **class** ServerInvoke {
9. **public** **static** **void** main(String[] args) {
10. **try** {
11. // 访问web service说明书地址
12. URL url = **new** URL("http://localhost:8080/WebServiceProject/StringOperationPort?WSDL");
14. // 指定命名空间，服务的名称，作用定位到那个服务
15. QName serviceName = **new** QName("http://buptsse/", "StringOperationService");
16. Service service = Service.create(url, serviceName);
18. // 获取接口的实例
19. StringOperation stringOperation = service.getPort(**new** QName("http://buptsse/", "StringOperationPort"),
20. StringOperation.**class**);
22. // 调用接口
23. String addString = stringOperation.AddString("XML", "Web");
24. Boolean isEqual = stringOperation.CompareString("Hello", "Hello");
26. // 输出结果
27. System.out.println("字符串相加的结果为：" + addString);
28. System.out.println("字符串比较的结果为：" + isEqual);
30. }**catch**(Exception e) {
31. System.out.println(e);
32. }
33. }
34. }