WebService体系之——JavaBean的传递

摘要：相比上篇笔记、这篇是实现使用webservice接口来实现JavaBean的传递。

一简介：

主要搭建的步骤：

1、 到apache的cxf官网上下载相应的jar包。

2、 建立一个java工程、将下载的jar包引入到项目中。

3、 创建JavaBean。

4、 创建服务端功能接口。

5、 创建实现服务端功能接口的具体类。

6、 发布服务接口。

7、 创建测试类、充当客户端调用服务端提供的功能、获取服务调用服务端提供的服务（具体点就是调用服务端提供的方法）。

二：同一个项目时具体步骤及代码

1、创建java项目、引入jar包、方便起见就直接把jar包（除endorsed文件夹下的之外）全部引入即可下载之后的包中我们还可以发现有自带的spring的包、这样我们在使用spring集成它的时候就方便很多了、这里先不提集成。后面会补充jar和项目结构图。

2、创建JavaBean——UserInfo：

[java]view plaincopy

```
1. package com.chy.ws.model;
2. import javax.xml.bind.annotation.XmlAccessType;
3. import javax.xml.bind.annotation.XmlAccessorType;
4. import javax.xml.bind.annotation.XmlRootElement;
5. import javax.xml.bind.annotation.XmlType;
6. /**
7.  * 注解：@XmlRootElement-指定XML根元素名称（可选）
8.  *           @XmlAccessorType-控制属性或方法序列化 ， 四种方案：
9.  *                       FIELD-对每个非静态，非瞬变属性JAXB工具自动绑定成XML，除非注明XmlTransient
10.  *                        NONE-不做任何处理
11.  *                        PROPERTY-对具有set/get方法的属性进行绑定，除非注明XmlTransient
12.  *                        PUBLIC_MEMBER -对有set/get方法的属性或具有共公访问权限的属性进行绑定，除非注明XmlTransient
13.  *           @XmlType-映射一个类或一个枚举类型成一个XML Schema类型
14.  *
15.  */
16. @XmlRootElement(name="userInfo")
17. @XmlAccessorType(XmlAccessType.FIELD)
18. @XmlType(propOrder={"userName", "age"})
19. publicclass UserInfo {
```

```
20. private String userName;
21. privateint age;
22. public UserInfo(String userName, int age) {
23. super();
24. this.userName = userName;
25. this.age = age;
26.     }
27. public UserInfo() {
28. super();
29.     }
30. public String getUserName() {
31. return userName;
32.     }
33. publicvoid setUserName(String userName) {
34. this.userName = userName;
35.     }
36. publicint getAge() {
37. return age;
38.     }
39. publicvoid setAge(int age) {
40. this.age = age;
41.     }
42. public String toString(){
43. return"user name: " + userName + " age: " + age;
44.     }
45. }
```

3、创建服务端接口：注意别忘了在服务端接口类级别上加上@WebService ！ UserService代码：

[java]view plaincopy

```
1. package com.chy.ws.service;
2. import javax.jws.WebService;
3. import com.chy.ws.model.UserInfo;
4. @WebService//切记 别丢了!
5. publicinterface UserService {
6. public UserInfo getUserInfo(String userName, int age);
7. }
```

4、创建实现服务端功能接口的具体类——UserServiceImpl代码：

[java]view plaincopy

```
1. package com.chy.ws.service;
2. import com.chy.ws.model.UserInfo;
3. publicclass UserServiceImpl implements UserService {
4. @Override
5. public UserInfo getUserInfo(String userName, int age) {
6. returnnew UserInfo(userName, age);
7.     }
8. }
```

5、发布服务接口——WebServiceServer代码：

[java]view plaincopy

```
1. package com.chy.ws.server;
2. import org.apache.cxf.jaxws.JaxWsServerFactoryBean;
3. import com.chy.ws.service.UserService;
4. import com.chy.ws.service.UserServiceImpl;
5. publicclass WebServiceUserInfo {
6. public WebServiceUserInfo(){
7. //create a factory for create a web service interface
8.         JaxWsServerFactoryBean factory = new JaxWsServerFactoryBean();
9. //release the web service
10.         factory.setAddress("http://localhost:8080/userinfoservice");
11. //register the web service
12.         factory.setServiceClass(UserService.class);
13.         factory.setServiceBean(new UserServiceImpl());
14.         factory.create();
15. //another method to release the web service
16. //Endpoint.publish("http://localhost:8080/userinfoservice", new UserServiceImpl());
17.     }
18. publicstaticvoid main(String[] args) {
19. new WebServiceUserInfo();
20.          System.out.println("Server ready...");
21. try {
22.                 Thread.sleep(1000*300);//休眠五分分钟，便于测试
23.             } catch (InterruptedException e) {
24.                 e.printStackTrace();
25.             }
26.             System.out.println("Server exit...");
27.             System.exit(0);
28.     }
29. }
```

6、通过浏览器访问地址：http://localhost:8080/userinfoservice?wsdl 若有结果则发布成功！

7、测试类——WebServiceUserInfoClient代码：

[java]view plaincopy

```
1. package com.chy.ws.client;
2. import org.apache.cxf.jaxws.JaxWsProxyFactoryBean;
3. import com.chy.ws.model.UserInfo;
4. import com.chy.ws.service.UserService;
5. publicclass WebServiceUserInfoClient {
6. publicstaticvoid main(String[] args) {
7.         JaxWsProxyFactoryBean factory = new JaxWsProxyFactoryBean();
8.         factory.setAddress("http://localhost:8080/userinfoservice");
9.         factory.setServiceClass(UserService.class);
10.         UserService userService = (UserService) factory.create();
11.         System.out.println("invoke userinfo webservice...");
12. // 测试返回JavaBean对象的
13.         UserInfo user = userService.getUserInfo("vicky", 23);
14.         System.out.println(user.toString());
15.     }
16. }
```

三：不同项目时客户端搭建具体步骤及代码

1、创建一个java项目、引入与服务端相同的jar包。

2、创建JavaBean——UserInfo（注意一定要和服务器中的JavaBean是相同的包名和内容、可直接将服务端的复制过来）。

3、创建服务端功能接口（直接将服务端的java代码复制到客户端、注意：完整路径名一定要一样、及包名完全相同）。

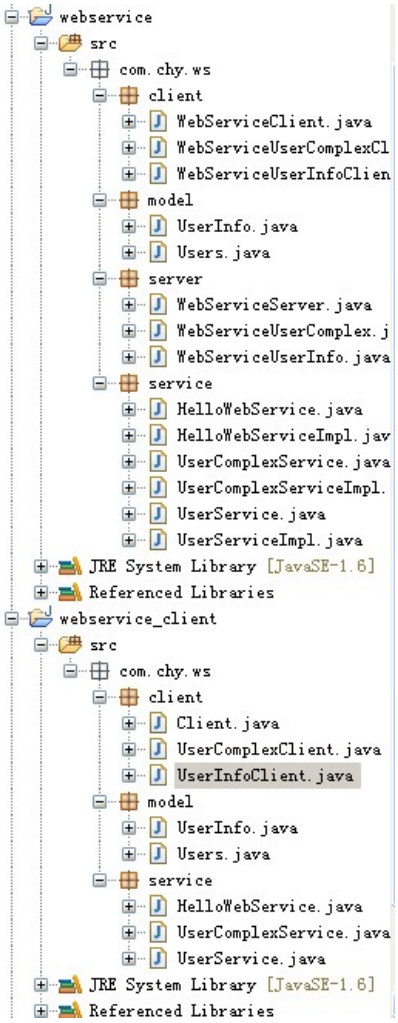
4、创建测试类来测试服务——UserInfoClient代码：

[java]view plaincopy

```
1. package com.chy.ws.client;
2. import org.apache.cxf.jaxws.JaxWsProxyFactoryBean;
3. import com.chy.ws.service.UserService;
4. publicclass UserInfoClient {
5. publicstaticvoid main(String[] args) {
6.         JaxWsProxyFactoryBean factory = new JaxWsProxyFactoryBean();
7.         factory.setAddress("http://localhost:8080/userinfoservice");
8.         factory.setServiceClass(UserService.class);
9.         UserService userService = (UserService) factory.create();
10.         System.out.println("invoking web service...");
11.         System.out.println("user info :" + userService.getUserInfo("chy", 22).toString());
12.         System.exit(0);
13.     }
14. }
```

四：补充——项目结构图及jar包

相对于前面的图这里明显有了多加的内容、但是并没有重新建一个新的项目。回头看的时候方便点。


webservice
src
com.chy.ws
client
WebServiceClient.java
WebServiceUserComplexCl
WebServiceUserInfoClien
model
UserInfo.java
Users.java
server
WebServiceServer.java
WebServiceUserComplex.j
WebServiceUserInfo.java
service
HelloWebService.java
HelloWebServiceImpl.jav
UserComplexService.java
UserComplexServiceImpl.
UserService.java
UserServiceImpl.java
JRE System Library [JavaSE-1.6]
Referenced Libraries
webservice_client
src
com.chy.ws
client
Client.java
UserComplexClient.java
UserInfoClient.java
model
UserInfo.java
Users.java
service
HelloWebService.java
UserComplexService.java
UserService.java
JRE System Library [JavaSE-1.6]
Referenced Libraries