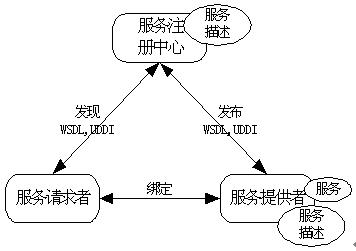
[**Web Service工作原理及实例**](http://blog.csdn.net/yangwenxue_admin/article/details/51059125)

**一、Web Service基本概念**

Web Service也叫XML Web Service WebService是一种可以接收从Internet或者Intranet上的其它系统中传递过来的请求，轻量级的独立的通讯技术。是:通过SOAP在Web上提供的软件服务，使用WSDL文件进行说明，并通过UDDI进行注册。  
  
XML：(Extensible Markup Language)扩展型可标记语言。面向短期的临时数据处理、面向万维网络，是Soap的基础。  
  
Soap：(Simple Object Access Protocol)简单对象存取协议。是XML Web Service 的通信协议。当用户通过UDDI找到你的WSDL描述文档后，他通过可以SOAP调用你建立的Web服务中的一个或多个操作。SOAP是XML文档形式的调用方法的规范，它可以支持不同的底层接口，像HTTP(S)或者SMTP。  
  
WSDL：(Web Services Description Language) WSDL 文件是一个 XML 文档，用于说明一组 SOAP 消息以及如何交换这些消息。大多数情况下由软件自动生成和使用。  
  
UDDI (Universal Description, Discovery, and Integration) 是一个主要针对Web服务供应商和使用者的新项目。在用户能够调用Web服务之前，必须确定这个服务内包含哪些商务方法，找到被调用的接口定义，还要在服务端来编制软件，UDDI是一种根据描述文档来引导系统查找相应服务的机制。UDDI利用SOAP消息机制（标准的XML/HTTP）来发布，编辑，浏览以及查找注册信息。它采用XML格式来封装各种不同类型的数据，并且发送到注册中心或者由注册中心来返回需要的数据。

**二、调用原理**



实现一个完整的Web服务包括以下步骤：  
  
◆ Web服务提供者设计实现Web服务，并将调试正确后的Web服务通过Web服务中介者发布，并在UDDI注册中心注册； （发布）  
  
◆ Web服务请求者向Web服务中介者请求特定的服务，中介者根据请求查询UDDI注册中心，为请求者寻找满足请求的服务； （发现）  
  
◆ Web服务中介者向Web服务请求者返回满足条件的Web服务描述信息，该描述信息用WSDL写成，各种支持Web服务的机器都能阅读；（发现）  
  
◆ 利用从Web服务中介者返回的描述信息生成相应的SOAP消息，发送给Web服务提供者，以实现Web服务的调用；（绑定）  
  
◆ Web服务提供者按SOAP消息执行相应的Web服务，并将服务结果返回给Web服务请求者。（绑定）

**三、调用方式：**

1. Net下采用GET/POST/SOAP方式动态调用WebService的简易灵活方法(C#)  
  
webservice 的调用有3种方式  
  
1). httpget   
2). httppost  
3). httpsoap  
  
soap 的优点是 可以传递结构化的 数据，而前两种不行。  
btw, soap 最终也是使用 HTTP 传送 XM

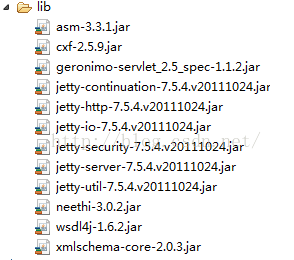
**Webservice实例**

**一、使用CXF开发Web Service服务端：**

每个Web Service组件需要2个部分：接口和实现类：

步骤：

1、准备开发需要的jar包【[apache-cxf-2.5.9下载](http://download.csdn.net/detail/yangwenxue_admin/9214993" \t "_blank)】



2、开发一个webservice业务接口，方法使用@WebService修饰。

**[java]** [view plain](http://blog.csdn.net/yangwenxue_admin/article/details/51059125) [copy](http://blog.csdn.net/yangwenxue_admin/article/details/51059125)

1. **package** com.ywx;
3. **import** javax.jws.WebService;
5. @WebService
6. **public** **interface** HelloWorld {
7. String sayHi(String name);
8. }

3、写一个这个方法的实现类，方法也需要使用@WebService修饰，并指定其中的参数中，如下指定了所需要实现的接口、并指定服务名称。

**[java]** [view plain](http://blog.csdn.net/yangwenxue_admin/article/details/51059125) [copy](http://blog.csdn.net/yangwenxue_admin/article/details/51059125)

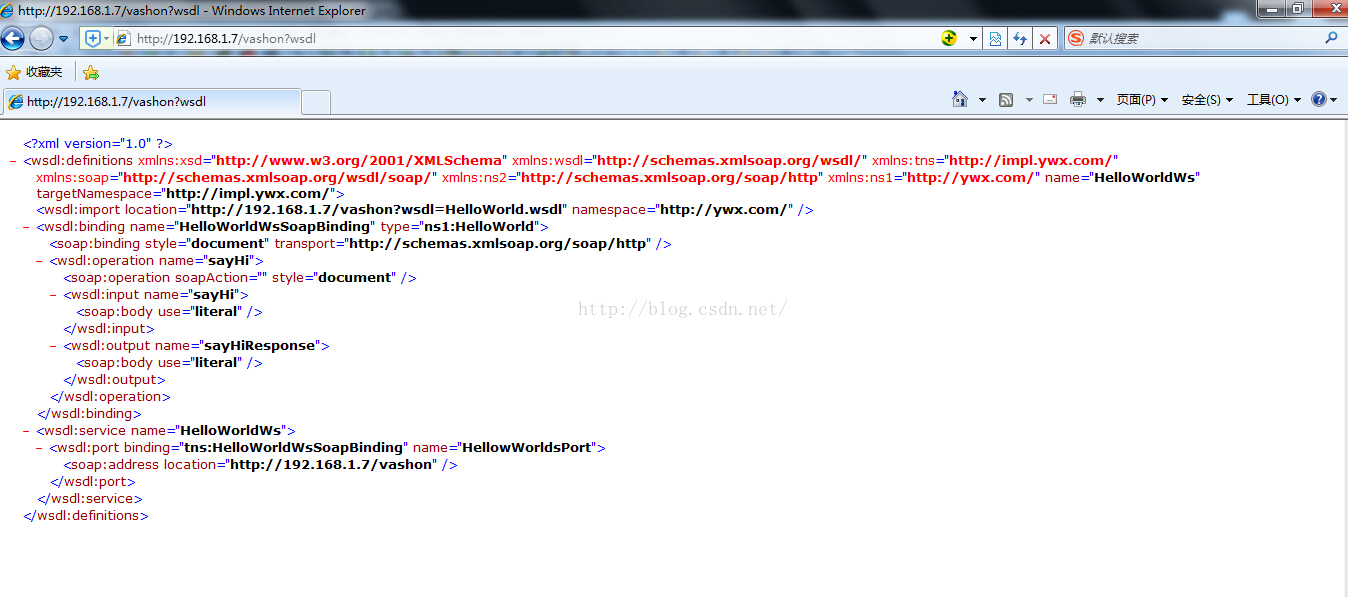
1. **package** com.ywx.impl;
3. **import** java.util.Date;
5. **import** javax.jws.WebService;
7. **import** com.ywx.HelloWorld;
8. @WebService(endpointInterface="com.ywx.HelloWorld",serviceName="HelloWorldWs")//指定webservice所实现的接口以及服务名称
9. **public** **class** HellowWorlds **implements** HelloWorld{
11. @Override
12. **public** String sayHi(String name) {
13. **return** name+"您好！现在时间是："+**new** Date();
14. }
16. }

4、暴露Web Service的函数，运行函数暴露Web Service：

**[java]** [view plain](http://blog.csdn.net/yangwenxue_admin/article/details/51059125) [copy](http://blog.csdn.net/yangwenxue_admin/article/details/51059125)

1. **package** com.ywx.lee;
3. **import** javax.xml.ws.Endpoint;
5. **import** com.ywx.HelloWorld;
6. **import** com.ywx.impl.HellowWorlds;
8. **public** **class** ServiceMain {
9. **public** **static** **void** main(String args[]){
10. HelloWorld hw = **new** HellowWorlds();
11. //调用Endpoint的publish方法发布Web Service
12. Endpoint.publish("192.168.1.7/vashon", hw);
13. System.out.println("Web Service暴露成功！");
14. }
15. }

然后运行浏览器，输入：http://192.168.1.7/vashon?wsdl 查看结果，如果成功生成如下wsdl文档则表示Web Service暴露成功。

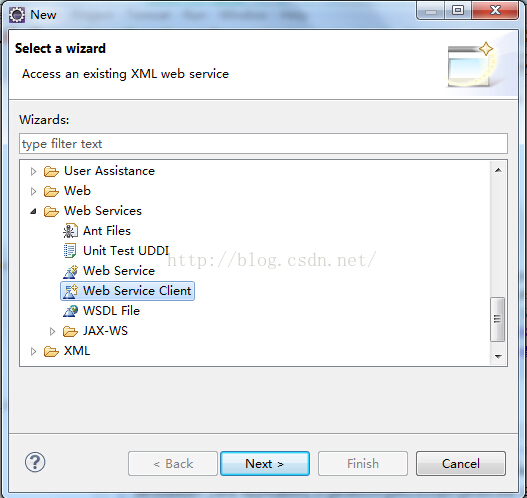


**二、使用CXF开发Web Service客户端：**

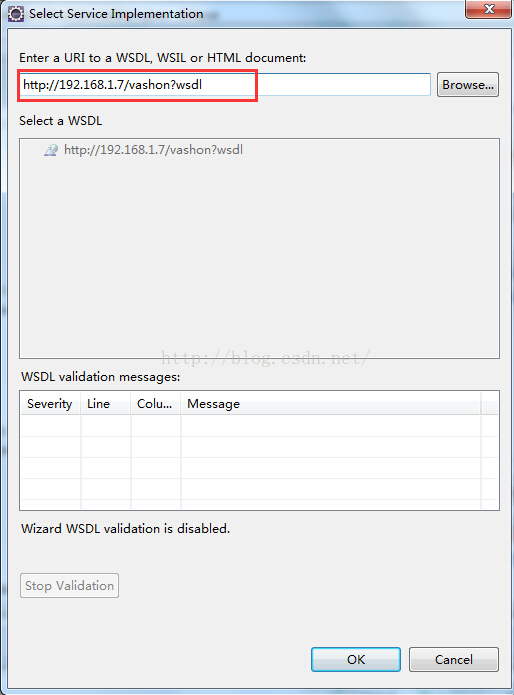
步骤：

1、新建一个客户端工程

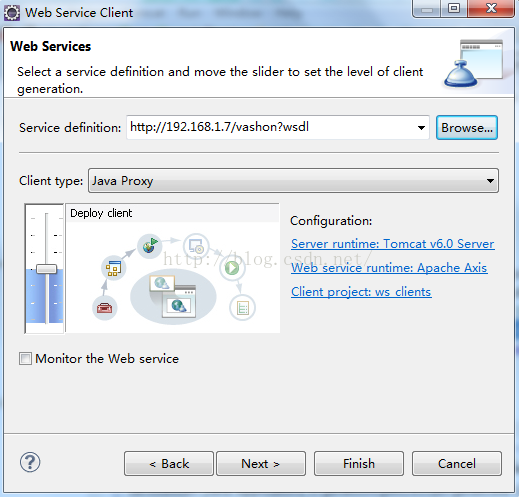
2、调用CXF提供的wsdl2java工具或使用eclipse/myeclipse的new Web Service生成客户端代码（这里使用第二种方式）：



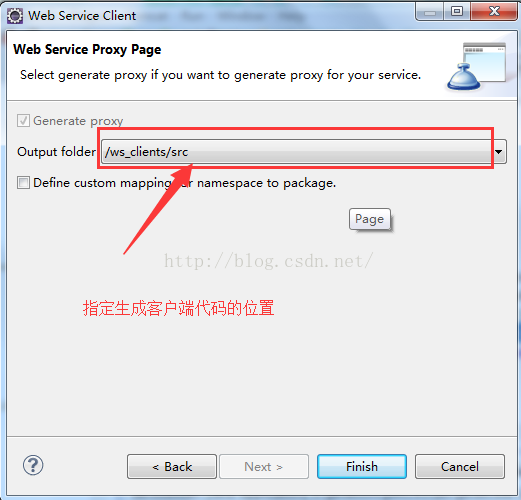
输入wsdl链接：



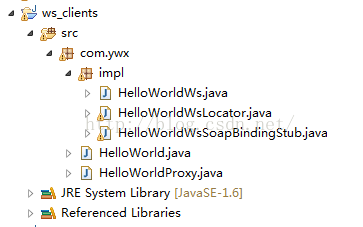
点击next：



选择生成客户端代码的位置：



点击finish，生成客户端代码如下：



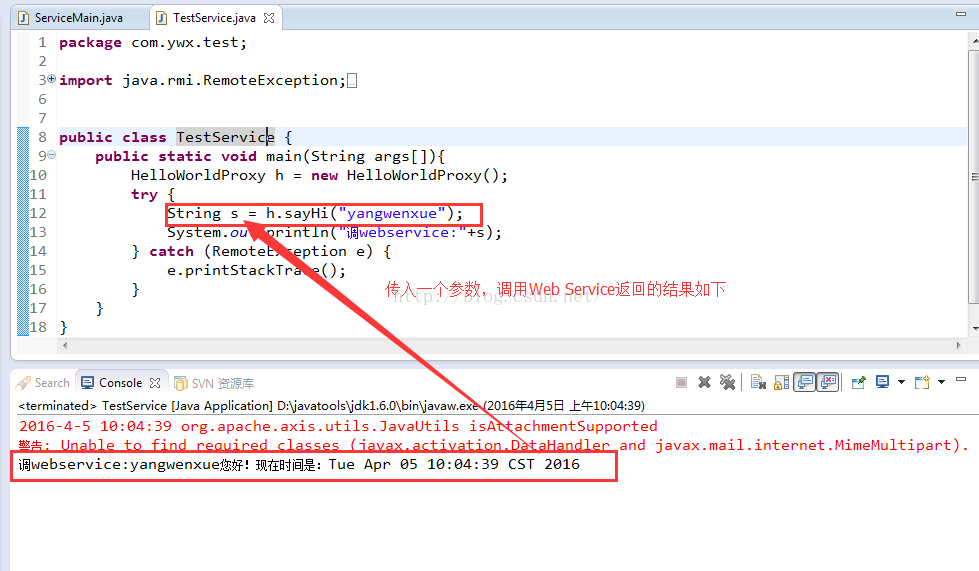
3、在客户端写测试类测试：

**[java]** [view plain](http://blog.csdn.net/yangwenxue_admin/article/details/51059125) [copy](http://blog.csdn.net/yangwenxue_admin/article/details/51059125)

1. **package** com.ywx.test;
3. **import** java.rmi.RemoteException;
5. **import** com.ywx.HelloWorldProxy;

8. **public** **class** TestService {
9. **public** **static** **void** main(String args[]){
10. HelloWorldProxy h = **new** HelloWorldProxy();
11. **try** {
12. String s = h.sayHi("yangwenxue");
13. System.out.println("调webservice:"+s);
14. } **catch** (RemoteException e) {
15. e.printStackTrace();
16. }
17. }
18. }

运行结果（传入一个参数，调用Web Service返回的字符串结果如下）：



其调用生成的格式已经有服务端定义好了，看上面贴出来的代码或者下面的截图说明：



Web Service服务端和客户端工程结果截图如下：

