# 1.开发服务端

1.apache下载axis2压缩包

<http://axis.apache.org/axis2/java/core/download.html>

  
 下载war包即可

2.eclipse中新建web项目

3.将下载的war包解压



4.进入WEB-INF文件夹，将conf、modules、services三个文件夹复制到自己项目的WEB-INF下

5.进入lib文件夹，将lib文件夹里的jar包全部复制到自己的项目的lib文件夹下，并导入jar包

6.打开web.xml文件，添加配置

<servlet>

<servlet-name>AxisServlet</servlet-name>

<servlet-class>org.apache.axis2.transport.http.AxisServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

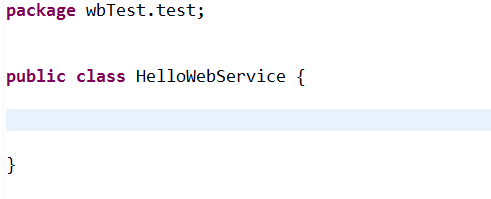
<servlet-name>AxisServlet</servlet-name>

<url-pattern>/services/\*</url-pattern>

</servlet-mapping>

7.创建一个package，并在包下创建一个普通的class





8.在类中简单定义两个方法，一个带参，一个无参

/\*\*

\* 带参方法

\* **@param** name

\* **@return**

\*/

**public** String hello(String name){

**return** "my nmae is " + name;

}

/\*\*

\* 无参数方法

\* **@return**

\*/

**public** String helloWorld(){

**return** "hello world";

}

9.在WEB-INF下的services文件中新建一个跟类名一致的文件夹



在新建的文件夹中在新建一个文件夹 名字为META-INF



10.在META-INF下新建services.xml配置文件



<!-- webService 服务名 -->

<service name=*"HelloWebService"*>

<!-- 描述 -->

<description>

HelloWorld Service Example

</description>

<parameter name=*"ServiceClass"*>

<!-- 类的全限定名字 包名+类名 -->

wbTest.test.HelloWebService

</parameter>

<!-- 方法名 -->

<!-- 这里要注意，当没有返回值时才用

class="org.apache.axis2.rpc.receivers.RPCInOnlyMessageReceiver"，

跟方法参数无关，没有参数还是用RPCMessageReceiver-->

<operation name=*"hello"*>

<messageReceiver class=*"org.apache.axis2.rpc.receivers.RPCMessageReceiver"* />

</operation>

<operation name=*"helloWorld"*>

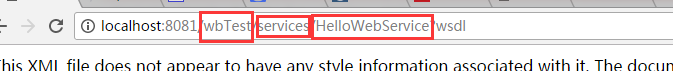
<messageReceiver class=*"org.apache.axis2.rpc.receivers.RPCMessageReceiver"* />

</operation>

</service>

11.将项目加入tomcat，启动项目，打开浏览器，输入

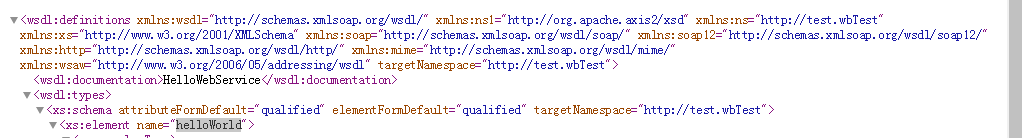
<http://localhost:8081/wbTest/services/HelloWebService?wsdl>



3个红框分别为 项目名、固定的services、和services.xml中的服务名



看到如下图则证明服务端配置成功



# 2.开发客户端

编写一个测试类

package wbTest.test;

import javax.xml.namespace.QName;

import org.apache.axis2.addressing.EndpointReference;

import org.apache.axis2.client.Options;

import org.apache.axis2.rpc.client.RPCServiceClient;

public class ClientWebService {

public static void main(String[] a) {

// webService服务地址 无需加 ?wsdl

String url = "http://localhost:8081/wbTest/services/HelloWebService";

String result = null;

try {

// 使用RPC方式调用WebService

RPCServiceClient serviceClient = new RPCServiceClient();

Options options = serviceClient.getOptions();

// 指定调用WebService的URL

EndpointReference targetEPR = new EndpointReference(url);

options.setTo(targetEPR);

// 在创建QName对象时，QName类的构造方法的第一个参数表示WSDL文件的命名空间名，

//也就是<wsdl:definitions>元素的targetNamespace属性值

//第二个参数为需要调用哪一个方法

QName opAddEntry = new QName("http://test.wbTest",

"hello");

// 指定hello方法的参数值，如果有多个，继续往后面增加即可，不用指定参数的名称

Object[] opAddEntryArgs = new Object[] { "mutianyang" };

// 指定hello方法返回值的数据类型的Class对象.....

Class[] classes = new Class[] { String.class };

// 返回参数类型，这个和axis1有点区别

// invokeBlocking方法有三个参数，其中第一个参数的类型是QName对象，表示要调用的方法名；

// 第二个参数表示要调用的WebService方法的参数值，参数类型为Object[]；

// 第三个参数表示WebService方法的返回值类型的Class对象，参数类型为Class[]。

// 当方法没有参数时，invokeBlocking方法的第二个参数值不能是null，而要使用new Object[]{}

// 如果被调用的WebService方法没有返回值，应使用RPCServiceClient类的invokeRobust方法，

// 该方法只有两个参数，它们的含义与invokeBlocking方法的前两个参数的含义相同

// 调用sayHello方法并输出该方法的返回值.......

result = (String) serviceClient.invokeBlocking(opAddEntry,

opAddEntryArgs, classes)[0];

System.out.println(result);

// 调用无参数方法

opAddEntry = new QName("http://test.wbTest",

"helloWorld");

//调用无参方法时 不能直接写null 要new Object[]{}

opAddEntryArgs = new Object[] {};

classes = new Class[] { String.class };

result = (String) serviceClient.invokeBlocking(opAddEntry,

opAddEntryArgs, classes)[0];

System.out.println(result);

} catch (Exception e) {

e.printStackTrace();

}

}

}

按照注释修改上述代码即可。

# 3.调用.net webservice

1.下载axis2 bin压缩包

http://axis.apache.org/axis2/java/core/download.html 

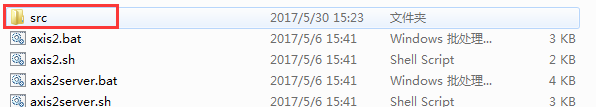
2.解压到本地硬盘

3.进入到解压后bin目录

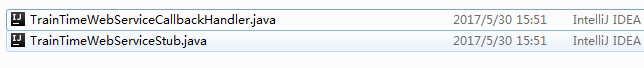


4.会将<http://www.webxml.com.cn/WebServices/TrainTimeWebService.asmx?wsdl> 的webservice描述编译为java类

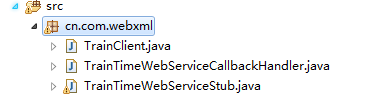
5.编译成功后在bin目录下会生成一个src文件夹



6.将src下的两个java文件复制到项目中



7.编写测试代码



**public** **static** **void** main(String[] args) {

**try**{

TrainTimeWebServiceStub ws = **new** TrainTimeWebServiceStub();

ws.\_getServiceClient().getOptions().setProperty(

org.apache.axis2.transport.http.HTTPConstants.***CHUNKED***,

Boolean.***FALSE***);

//要調用的方法名

GetStationName gcf = **new** GetStationName();

//方法所需參數

// gcf.setTrainCode("D32");

//接收返回值

ArrayOfString gt =

ws.getStationName(gcf).getGetStationNameResult();

**for** (String string : gt.getString()) {

System.***out***.println(string);

}

}**catch**(Exception e){

e.printStackTrace();

}

}