**作业 1: 服务开发和调用**

Class person

{

String name;

int age;

boolean gender;

set/getName();

set/getAge();

set/getGender();

String sayHello()

{

Return “Hello world!”+name;

}

}

1.1 基于 Apache Dubbo 将上述类的方法对外提供 RPC 服务并调用；

1.2 基于 Java Spring 将上述类的方法对外提供 RESTful 服务并调用；

1.3 基于 Apache Axis2 将上述类的方法对外提供 Web 服务，生成 WSDL 文件，以及调用

服务。

1、Apache Axis2目前最新版本为1.7.1，官方下载地址：

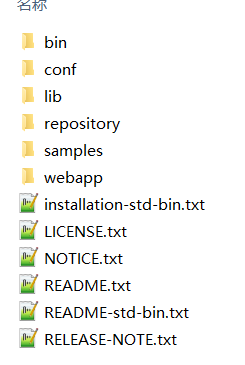
<http://axis.apache.org/axis2/java/core/download.html>



axis2-1.7.1-bin.zip为Axis2完整的发行版本。

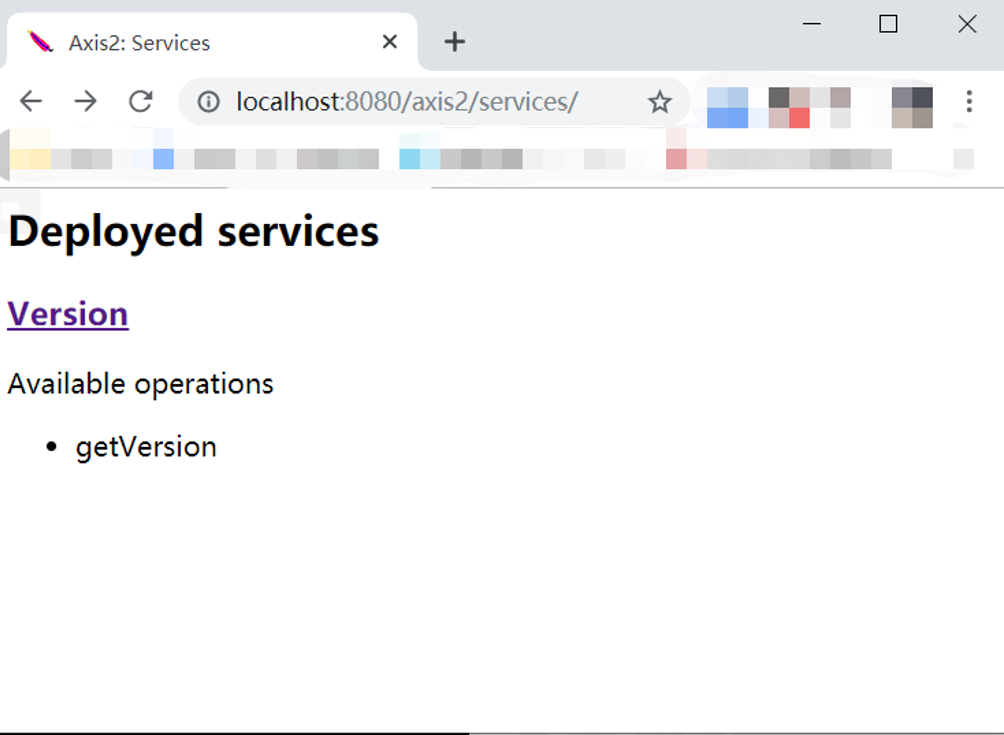
axis2-1.7.1-war.zip解压后可以获取一个war包，该war包可以部署到应用服务器，用于和Web容器集成。虽然axis2-1.7.1-bin.zip已经包含了该项，但是需要我们自己使用ant构建生成war包，为了方便起见,我们直接从官网下载war包。

bin目录为axis2的一些可执行脚本，axis2提供了一个内置的应用服务器及一个Web应用，conf为Web应用配置，当使用内置服务器发布WebService时，相关的\*.aar包可以存放到repository目录下。samples目录为axis2提供的一些案例项目，webapp即为axis2-1.7.1-war.zip中的内容，用于將axis2安装到web容器（例如tomcat）中。



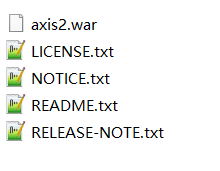
2、进入bin目录下执行axis2server.bat，启动内置服务器，打开浏览器访问http://localhost:8080/axis2/services/

出现下面界面说明axis2内置服务器启动成功。



3、將Axis2安装到Tomcat中

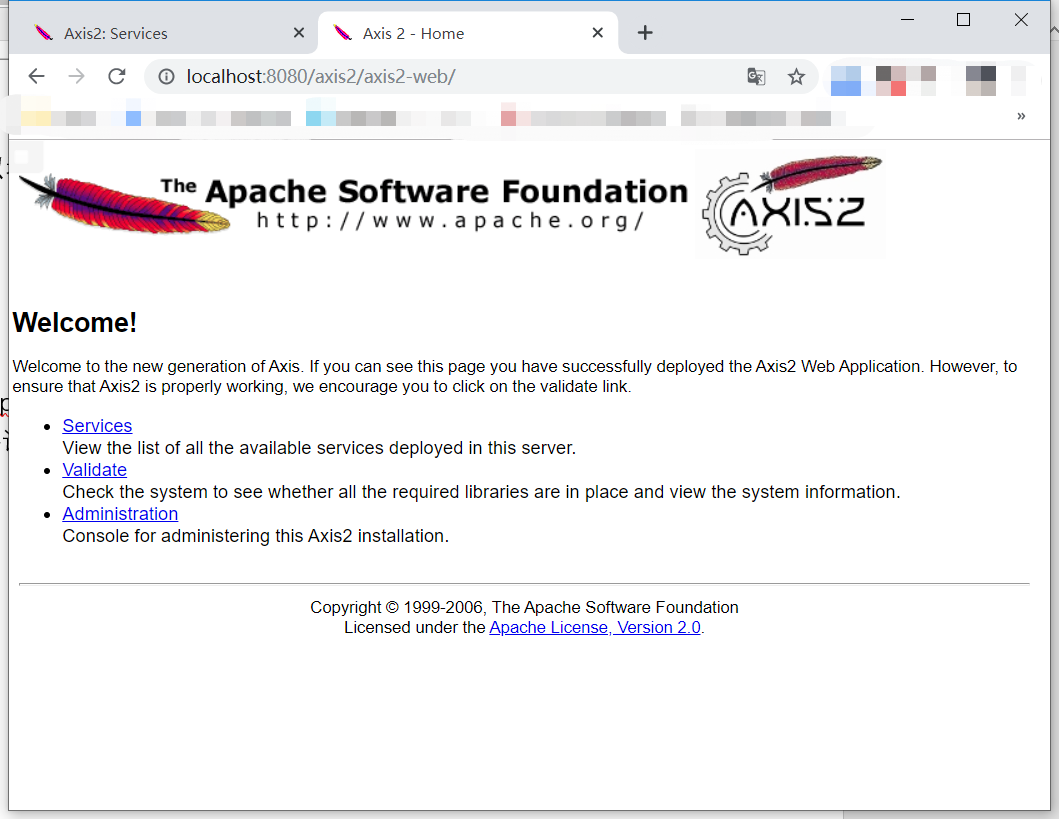
3.1將axis2-1.7.1-war.zip解压后可以看到axis2.war包



3.2拷贝axis2.war到tomcat的webapps目录下

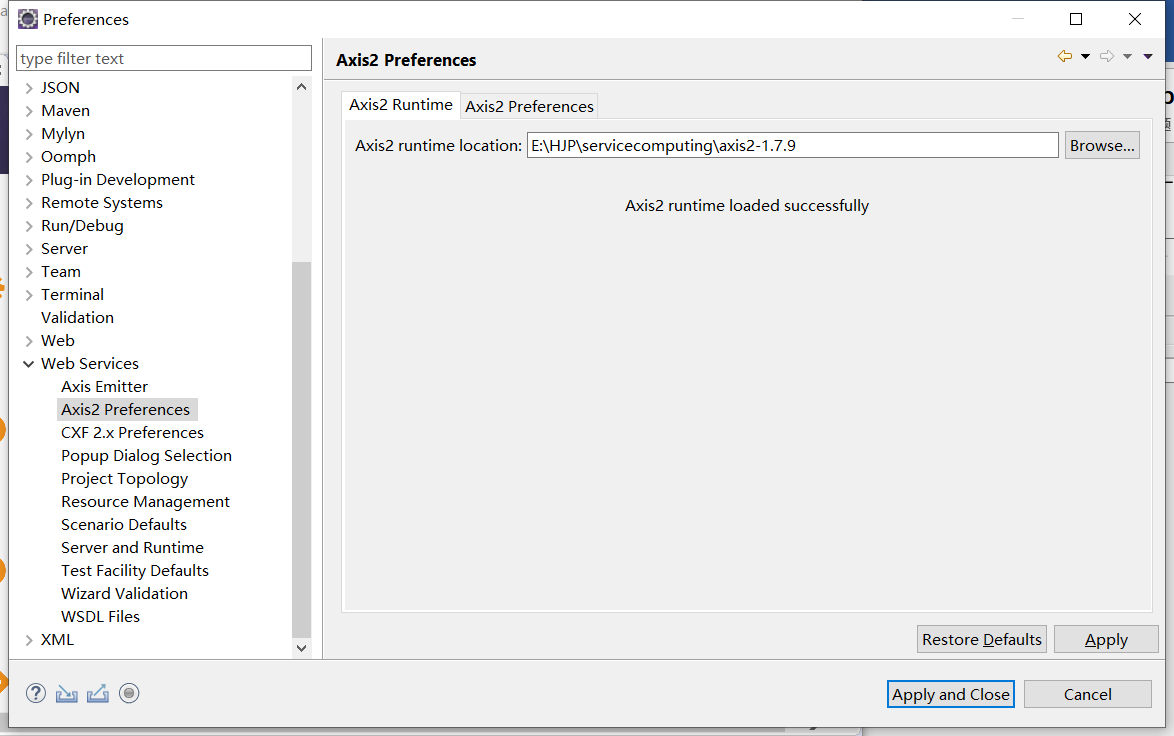
3.3启动tomcat服务器，打开浏览器访问http://localhost:8080/axis2/axis2-web/

出现下图所示界面说明安装成功



3.4点击Services可以查看当前所有对外发布的WebService。

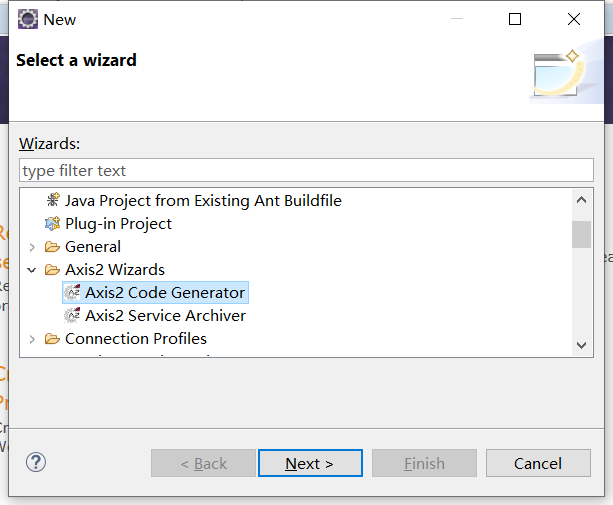
4、在Eclipse中安装axis2插件，将下载的axis2-eclipse-codegen-plugin，axis2-eclipse-service-plugin解压出来的jar包，复制到Ecilpse的 dropins（注意不是plugins，网上很多教程都是这个目录，错误）文件夹下，重启Eclipse，配置axis2的运行环境。图中的路就是下载的axis2-1.7.9-bin的解压目录。



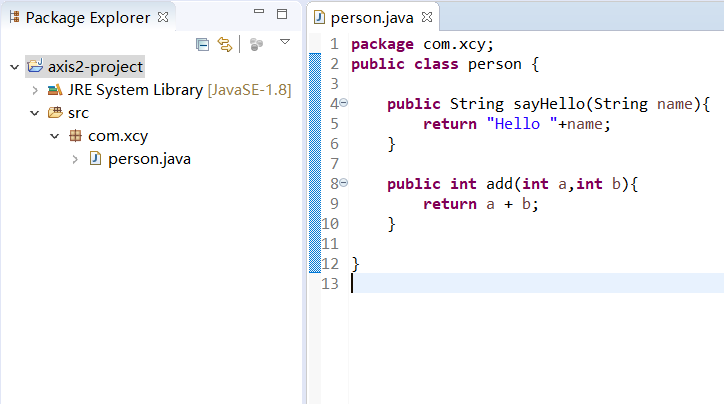
5、深坑问题：Eclipse最新版本2019-09与Axis2-1.7.9冲突，File->New->other无法载入如下的Axis2 Wizards内容，查阅网上许多的解决方法均无效。

最终解决方案：查阅Axis2-1.7.9 Last Published: 2018-11-16，所以最新版本的Eclipse是无法兼容Axis2的，因此下载一个在此之前的eclipse-jee-2018-09版本，然后重复上述步骤即可解决问题。下载地址：

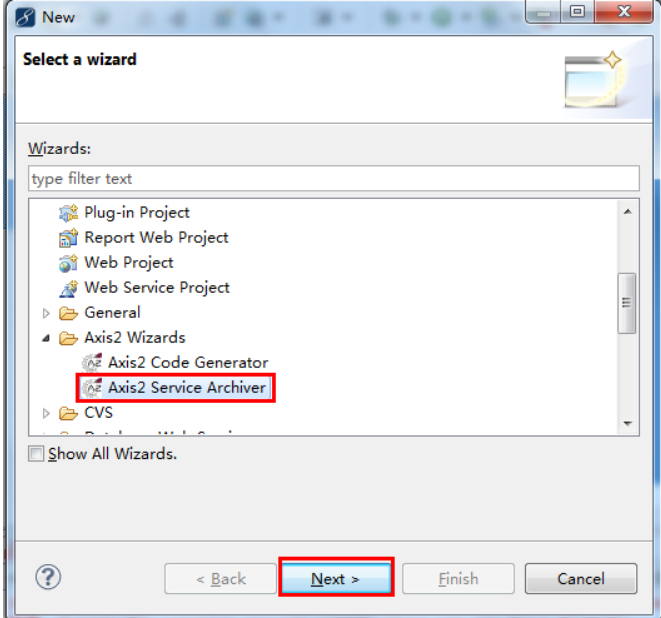
<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2018-09/R/eclipse-jee-2018-09-win32-x86_64.zip>



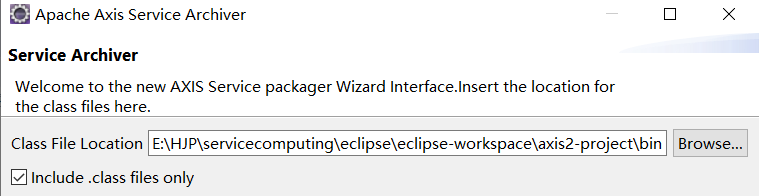
6、接下来编写服务代码，首先创建一个Java项目，File->New->Java Project,创建如下的程序，先简单编写两个服务sayHello和add。



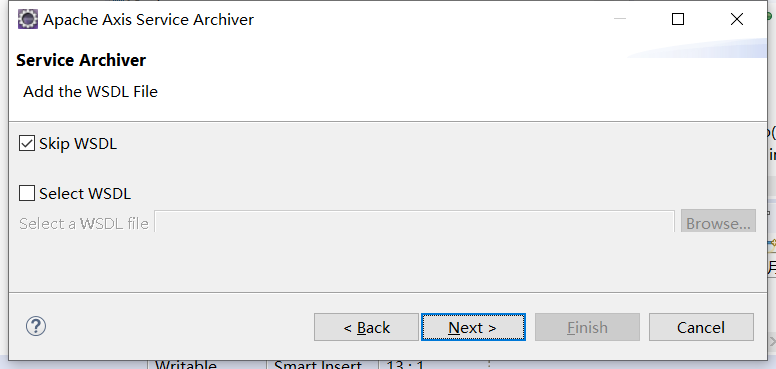
7、将服务代码打包成arr文件：Eclipse菜单- New - File - Other -Axis2 Service Archiver



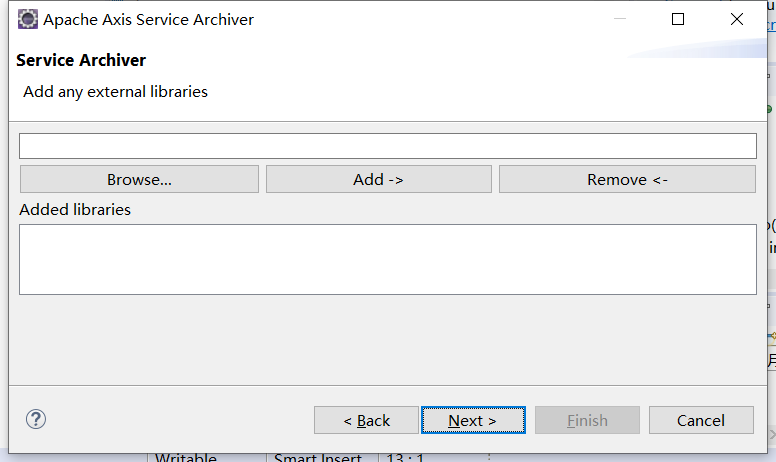
class file location：为刚刚写的Axis2WB类所在工程的bin目录



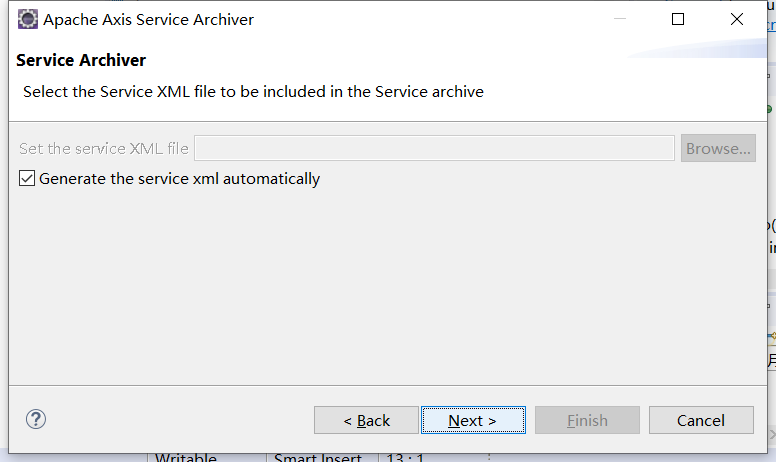
选择skip wsdl



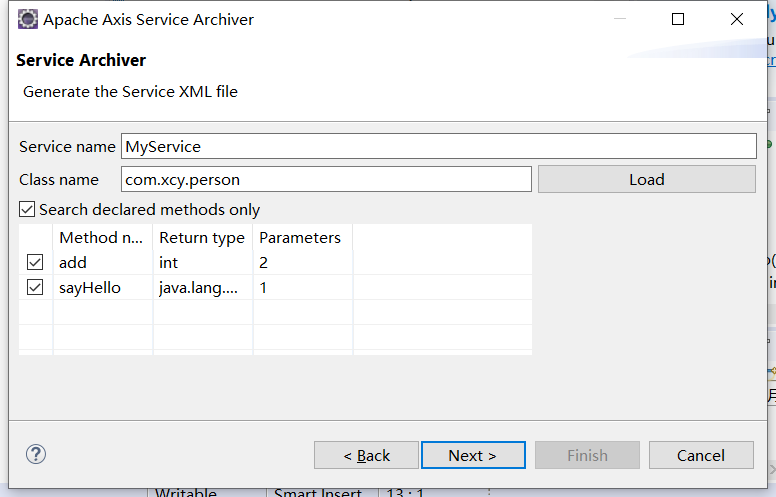
如果你的person有引用jar包，则在这里选择。这里写的没有，所以next。



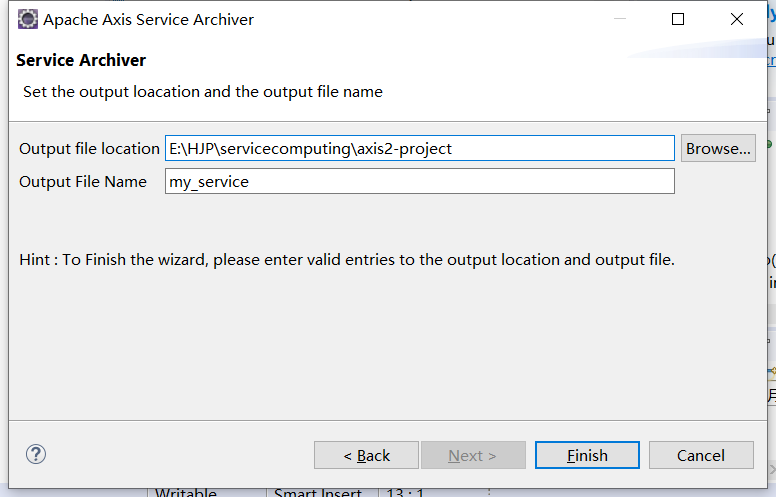
由于我们没有编写service.xml，所以勾选让它自动生成，next。



输入服务名称(随意)、类全名、load、next。

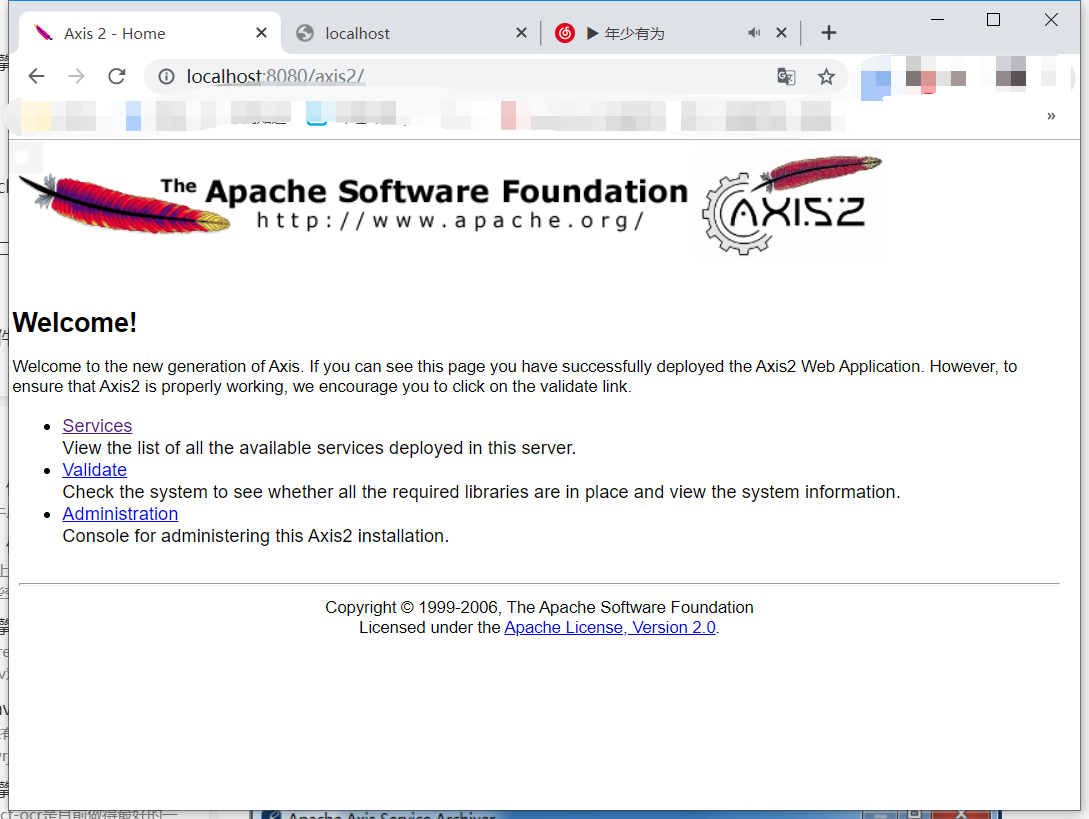


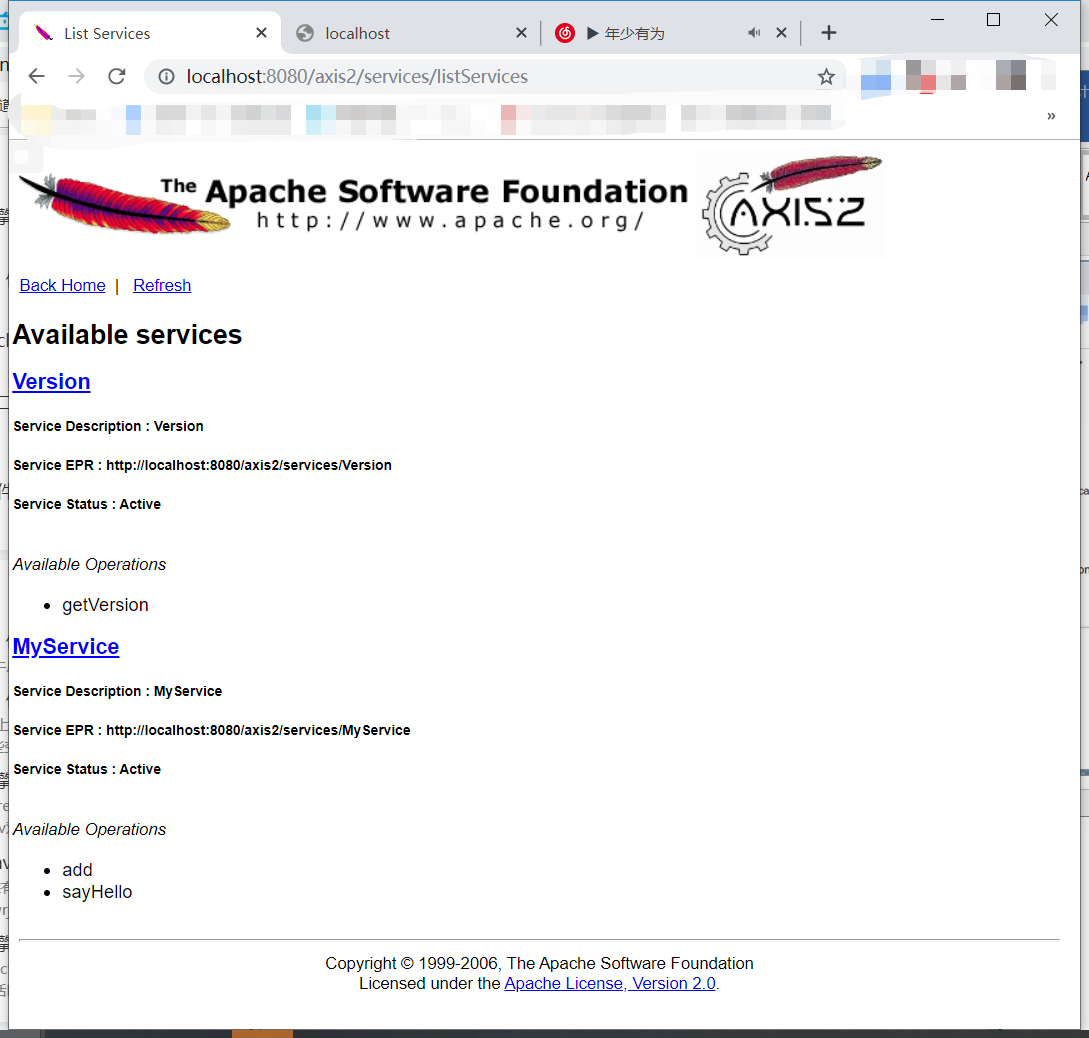
设置aar文件名以及存放目录，Finish。



8、发布

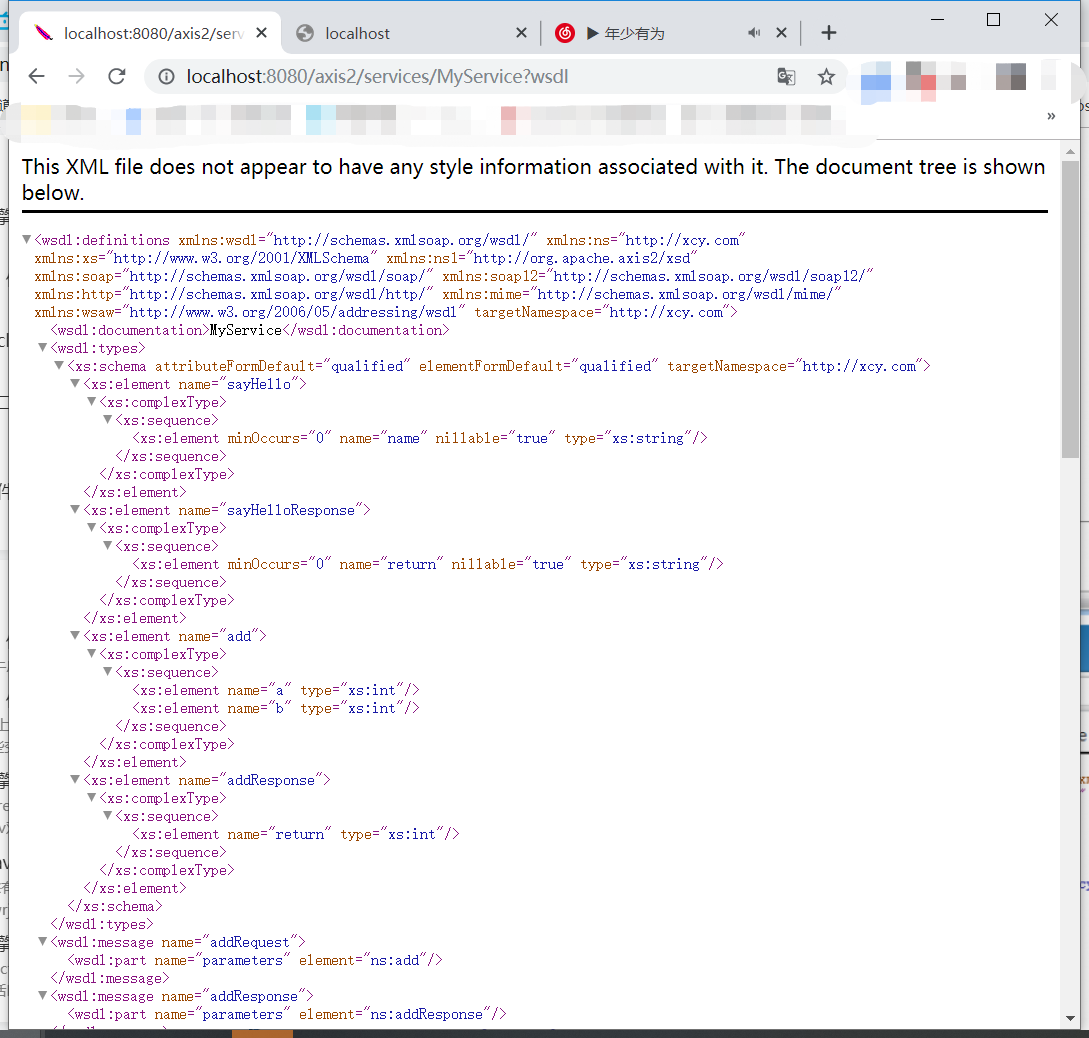
将my\_service.aar文件丢到之前部署的axis2应用的WEB-INF\services\目录下面，重启tomcat, 再访问http://localhost:8080/axis2/,点击Service。



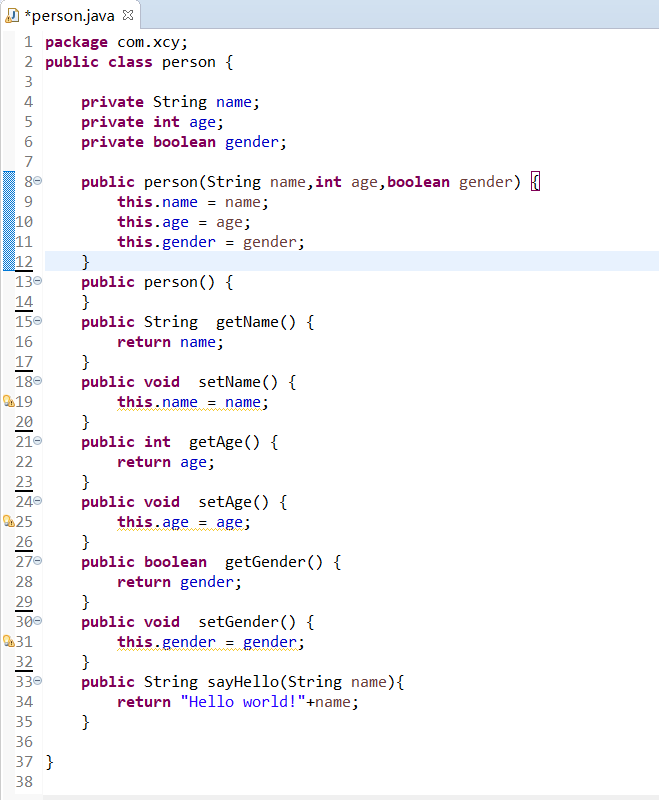


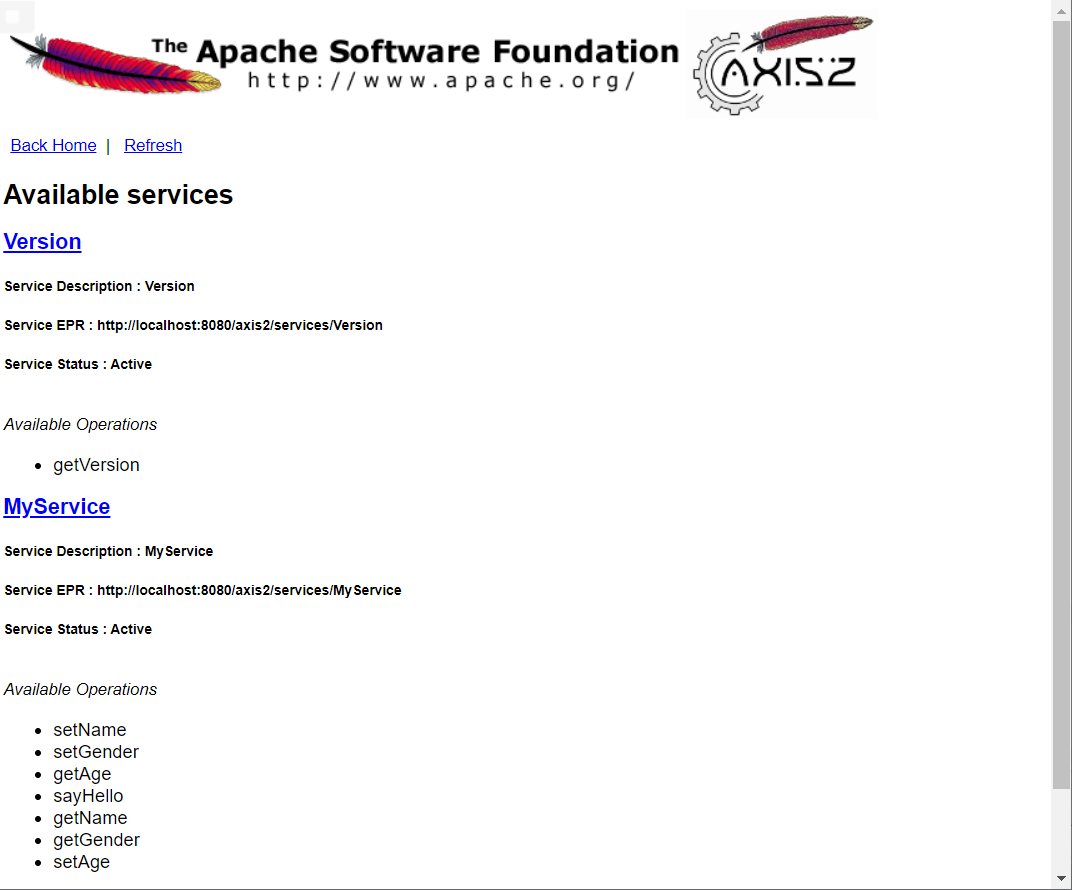
点击MyService,查看WSDL

<http://localhost:8080/axis2/services/MyService?wsdl>

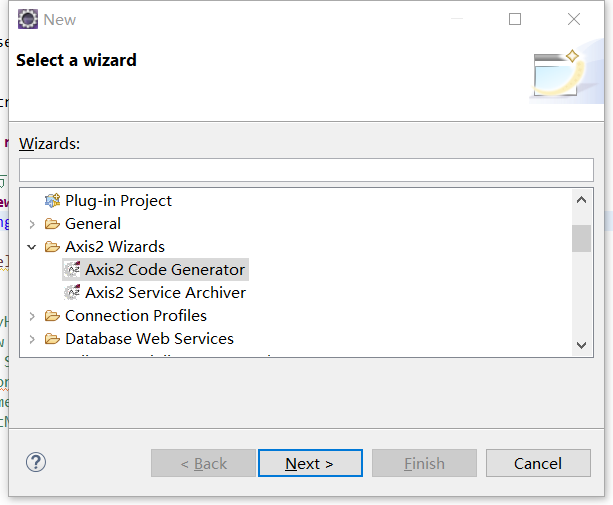


9、发布成功之后，我们重新改写下服务端代码，并重复上述步骤。



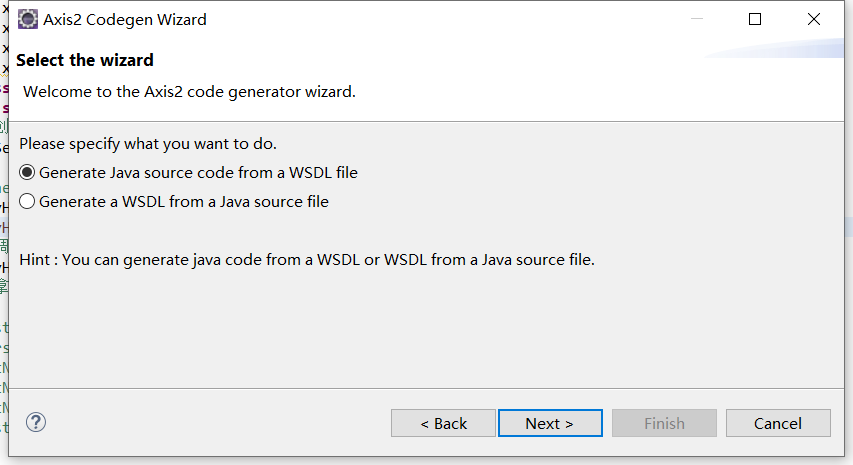


10、生成客户端代码：Eclipse菜单-File-New-Other-Axis2 Code Generator

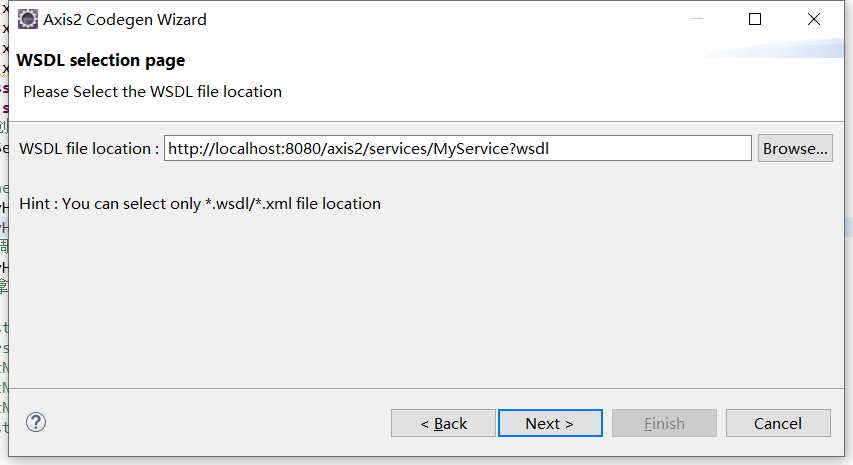


Generate Java source code from a WSDL file：根据WSDL生成webservice客户端的java代码。(在这里，我们选择这个)

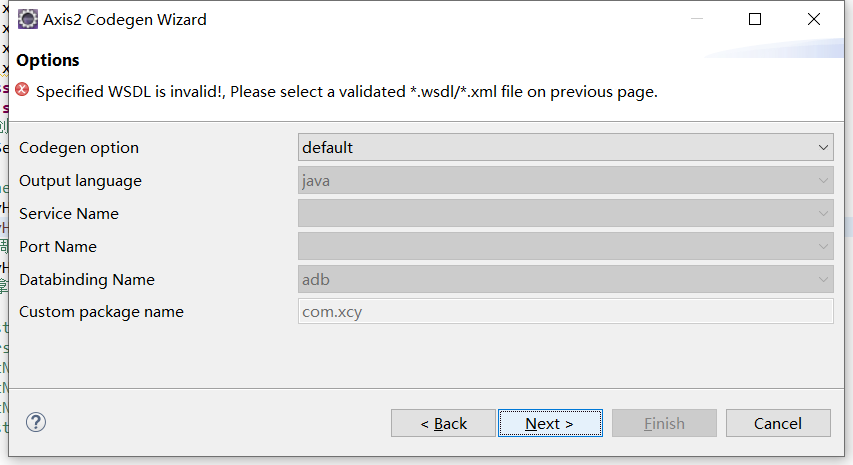
Generate a WSDL from a Java source file：根据一个java源文件生成wsdl文件(这个源文件是打算发布成Web服务的java源文件,例如本demo中的person.java)。



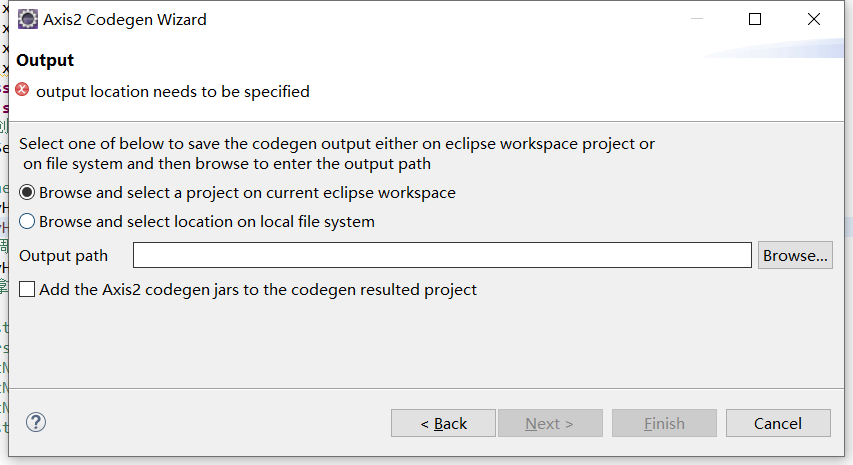
输入我们刚才点开的WSDL网页。



选择默认即可。



选项一是把代码生成到当前的工作空间指定的项目中去，第二个是生成到硬盘指定目录中去，这里选择第一个，选中我们刚才的axis2-project项目。

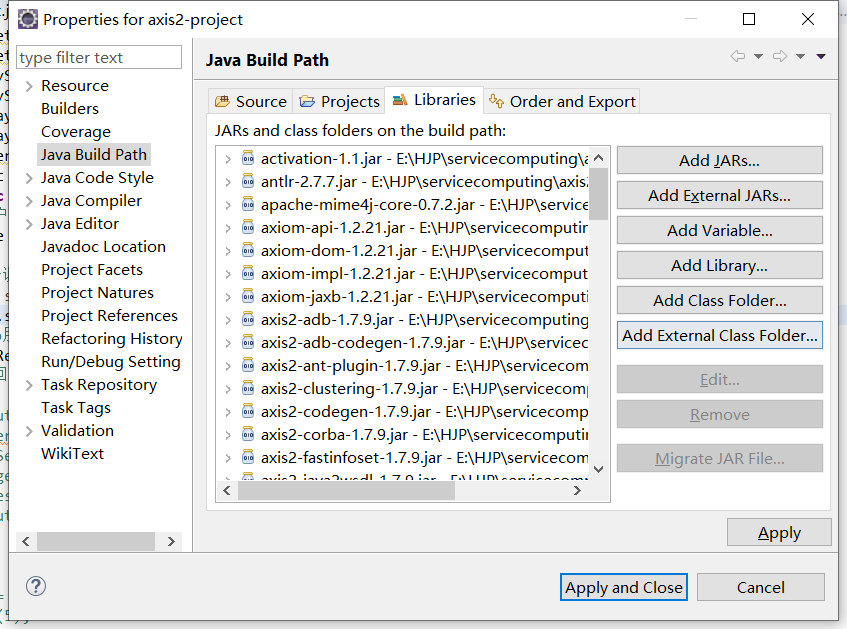


生成完代码后，你会发现报错，原因是因为缺少相关的jar包。

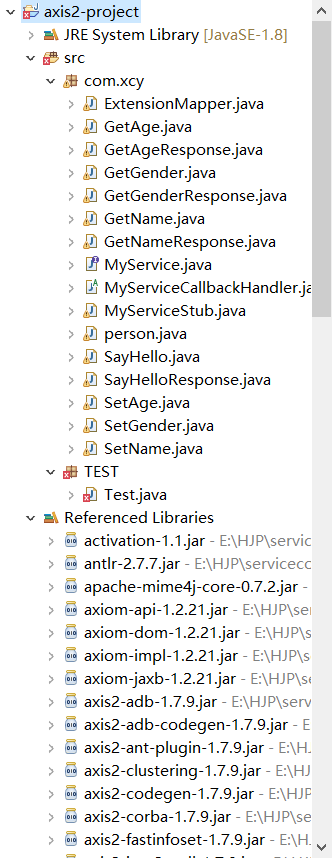
解压在第一步中下载的axis2.zip将解压后的目录中的lib下面的所有jar包拷进来，添加到class path中：

在axis2-projects上面右键Build Path->Configure Build Path->Libraries->Add External JARS

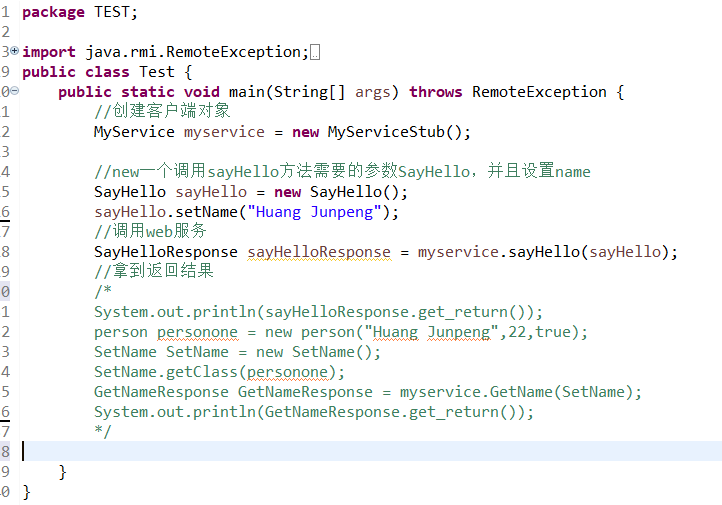
然后选中axis2里面lib所有的jar包。



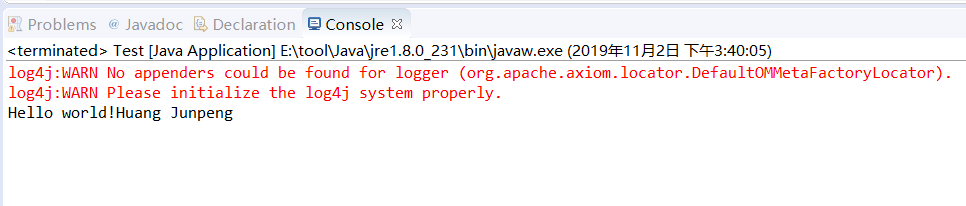
最后的代码结构如下，我们再编写一个调用Web服务的客户端（如图中TEST.Test.java）



Test.java代码如下：先简单调用SayHello()函数,运行时记得打开Tomcat，否则出错。



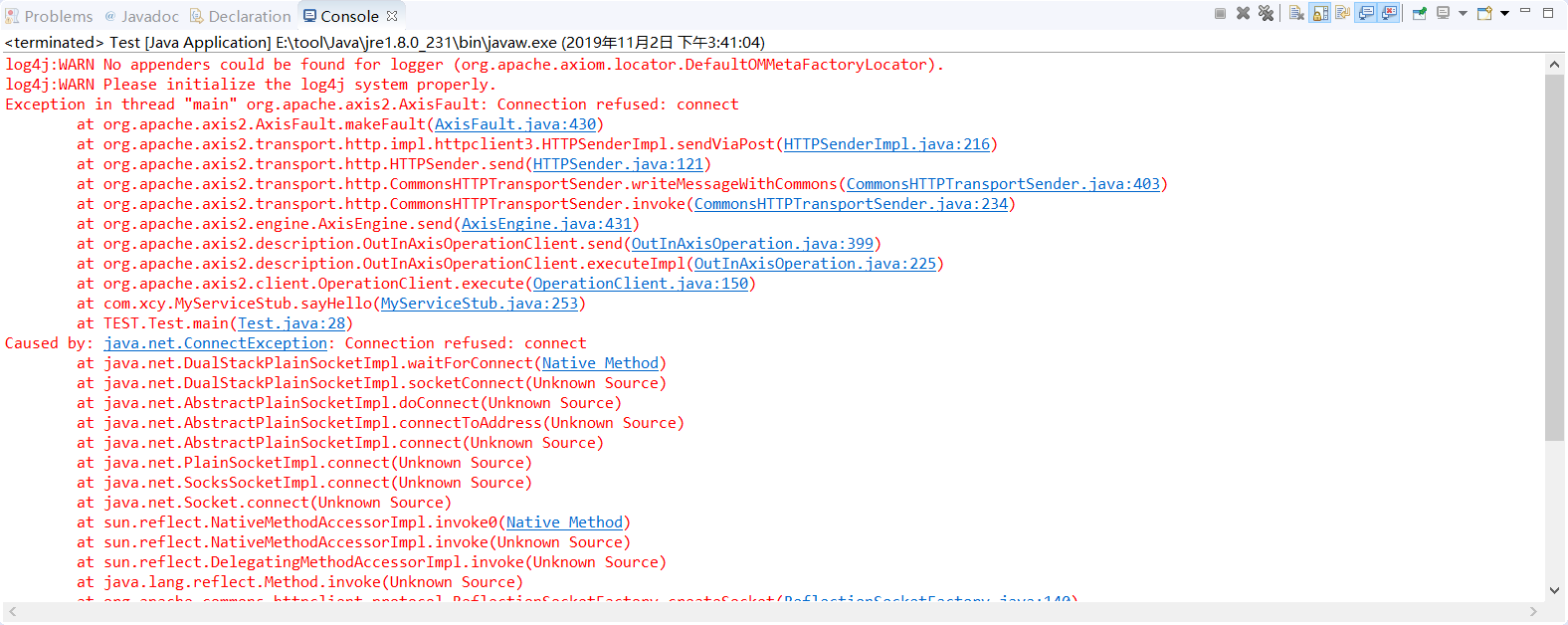
对Test.java右键run as java application



调用服务成功。

而这里我还没搞明白怎么调用写的类对象服务，暂时完成基本全过程步骤（不知道怎么写调用对象的代码）。

PS1:假如Tomcat没有打开会出现如下问题：



PS2:假如ECLIPSE报错：

Access restriction: The type Resource is not accessible due to restriction on required library：

解决：Eclipse 默认把这些受访问限制的API设成了ERROR。只要把Windows-Preferences-Java-Complicer- Errors/Warnings里面的Deprecated and restricted API中的Forbidden references(access rules)选为Warning就可以编译通过。