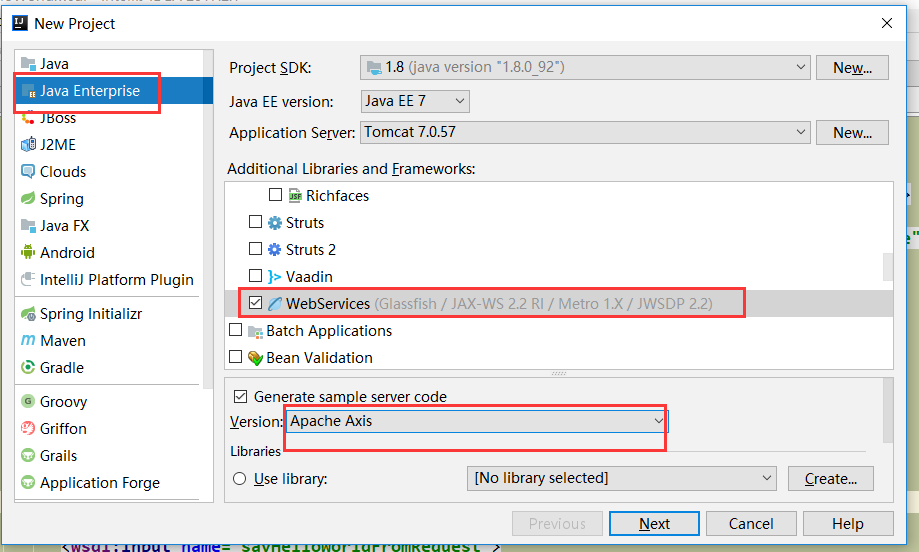
上篇博客给大家整理了WebService的基本概念及组成元素，对概念或原理还不太清楚的可以去看看。

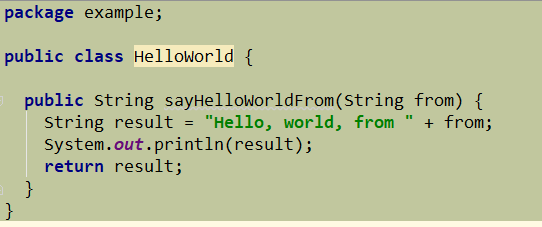
这篇我们用Intellij IDEA来搭建一个WebService，并看看如何请求。

首先，new project， 选择WebService项目，并选择Apache Axis:



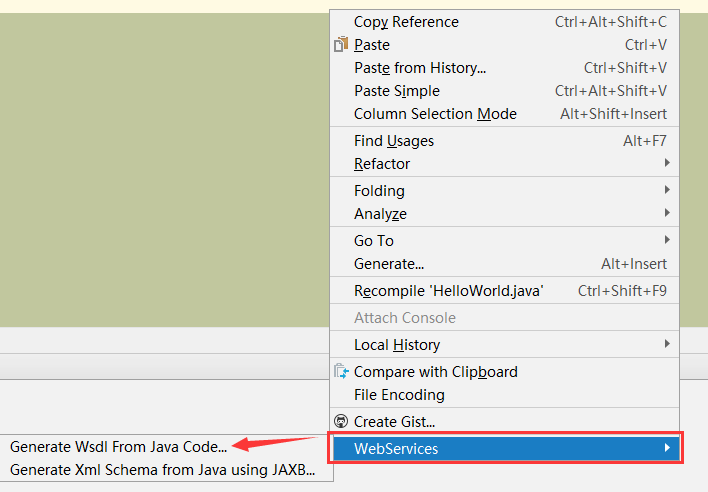
点击next，随便去个项目名称，比如webservice。

finish之后，会有一些示例文件，默认有一个HelloWorld类，这个类里面有个方法sayHelloWorldFrom(String from)



这是一个很简单的java方法，我们通过这个简单的java方法来看看WebService的结构和用法。

在这个类里面右键 -> WebService 生成wsdl文件：



生成好了之后，在HelloWorld.java同一个包下有一个.wsdl文件，打开这个文件。

这个WSDL由五个部分组成，分别是service, binding, portType, message, types.

service里面主要描述这个服务接口路径，相当于REST里面定义了一个API：

<**wsdl:service name="HelloWorldService"**>  
  
 <**wsdl:port binding="impl:HelloWorldSoapBinding" name="HelloWorld"**>  
  
 <**wsdlsoap:address location="http://localhost:8080/services/HelloWorld"**/>  
  
 </**wsdl:port**>  
  
</**wsdl:service**>

binding里面将Request和Response与SoapAction绑定到了一起：

<**wsdl:binding name="HelloWorldSoapBinding" type="impl:HelloWorld"**>  
  
 <**wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"**/>  
  
 <**wsdl:operation name="sayHelloWorldFrom"**>  
  
 <**wsdlsoap:operation soapAction="test"**/>  
  
 <**wsdl:input name="sayHelloWorldFromRequest"**>  
  
 <**wsdlsoap:body use="literal"**/>  
  
 </**wsdl:input**>  
  
 <**wsdl:output name="sayHelloWorldFromResponse"**>  
  
 <**wsdlsoap:body use="literal"**/>  
  
 </**wsdl:output**>  
  
 </**wsdl:operation**>

端类型里面描述了这个接口（方法）的输入端（Request message）和输出端（Response message） 以及参数。

<**wsdl:portType name="HelloWorld"**>  
  
 <**wsdl:operation name="sayHelloWorldFrom" parameterOrder="from"**>  
  
 <**wsdl:input message="impl:sayHelloWorldFromRequest" name="sayHelloWorldFromRequest"**/>  
  
 <**wsdl:output message="impl:sayHelloWorldFromResponse" name="sayHelloWorldFromResponse"**/>  
  
 </**wsdl:operation**>  
  
</**wsdl:portType**>

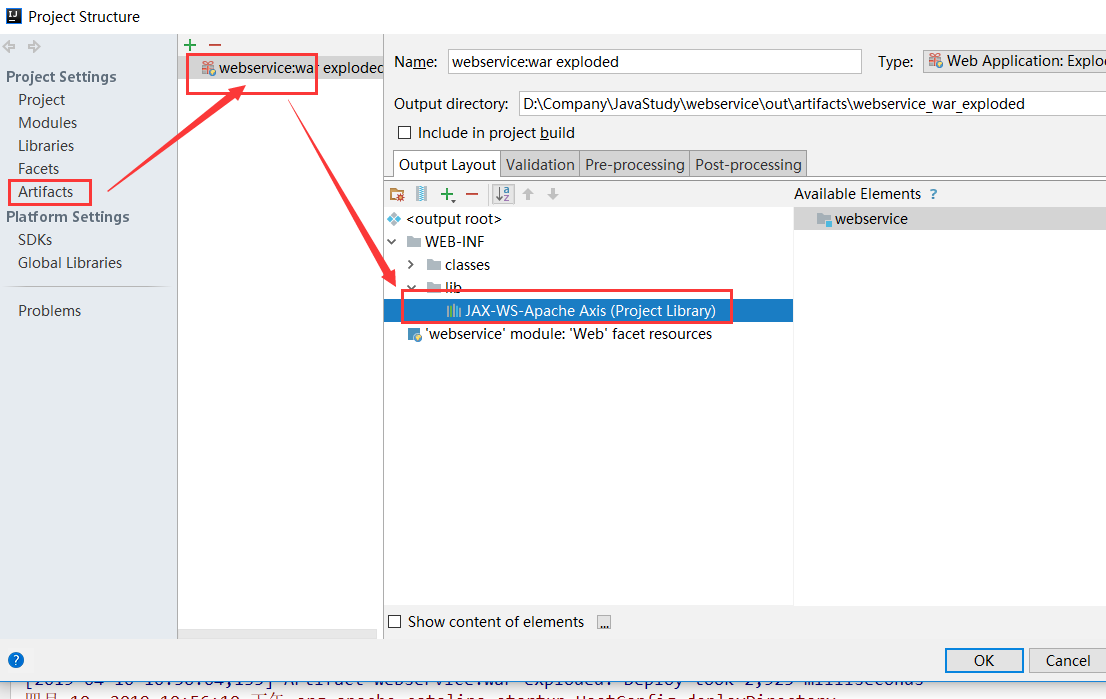
message里面定义的是Request的请求参数 和 Response的响应类型

<**wsdl:message name="sayHelloWorldFromResponse"**>  
  
 <**wsdl:part element="impl:sayHelloWorldFromReturn" name="sayHelloWorldFromReturn"**/>  
  
</**wsdl:message**>  
  
<**wsdl:message name="sayHelloWorldFromRequest"**>  
  
 <**wsdl:part element="impl:from" name="from"**/>  
  
</**wsdl:message**>

types里面基本就是把请求和响应的类型给罗列出来了，一直描述到基本数据类型。

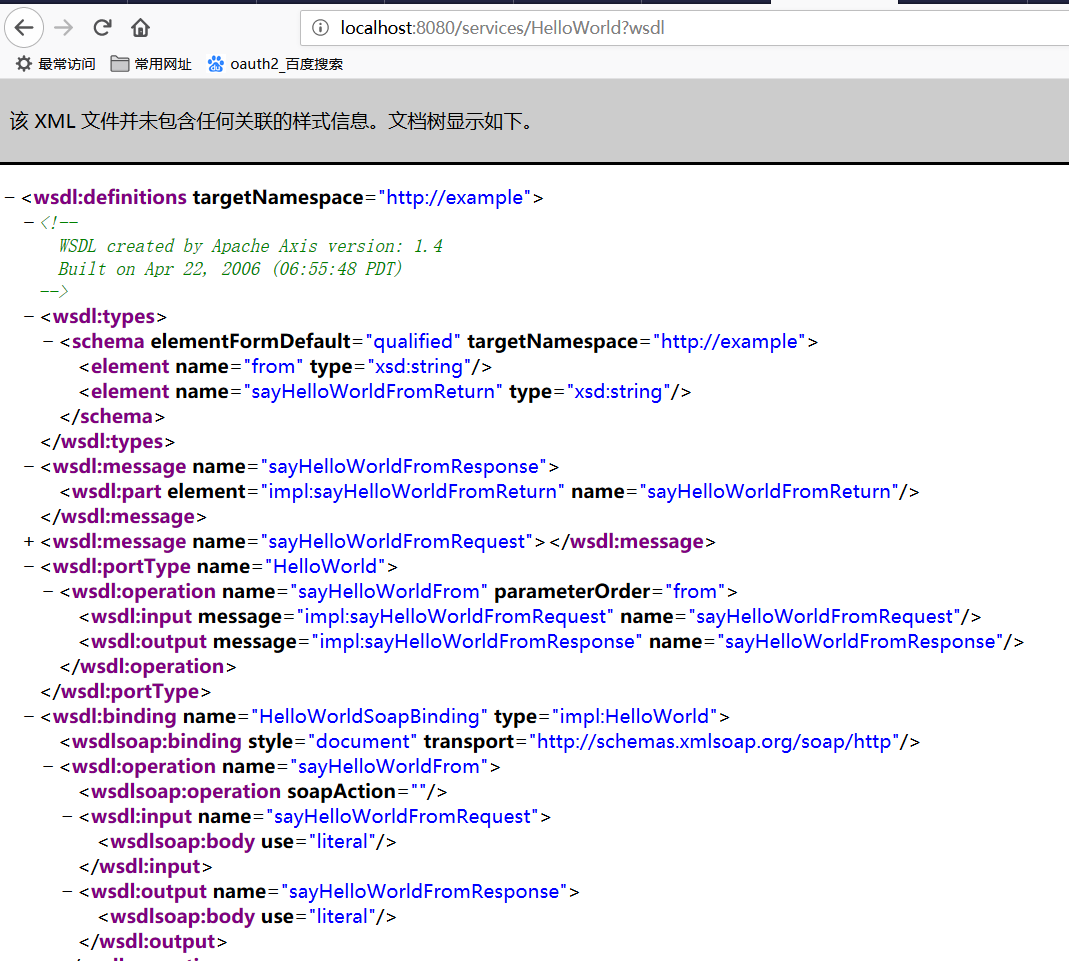
<**wsdl:types**>  
 <**schema elementFormDefault="qualified" targetNamespace="http://example" xmlns="http://www.w3.org/2001/XMLSchema"**>  
 <**element name="from" type="xsd:string"**/>  
 <**element name="sayHelloWorldFromReturn" type="xsd:string"**/>  
 </**schema**>  
</**wsdl:types**>

OK，现在请配置一下Tomcat，并将项目部署到Tomcat。注意，手动给项目加入这个依赖，不然访问会报错的。



在浏览器访问HelloWorld的wsdl：

<http://localhost:8080/services/HelloWorld?wsdl>



**请求WebService**

wsdl只是接口的描述语言，现在我们可以根据这些描述来封装SOAP请求了。

打开PostMan，或者SOAPUI以及其他Rest客户端。

**Request URL**：<http://localhost:8080/services/HelloWorld>

**Request Headers:**

SoapAction: (留空即可)

Content-Type: application/xml

**Request Body:** (请遵守SOAP规范)

<?xml version="1.0" encoding="UTF-8"?>

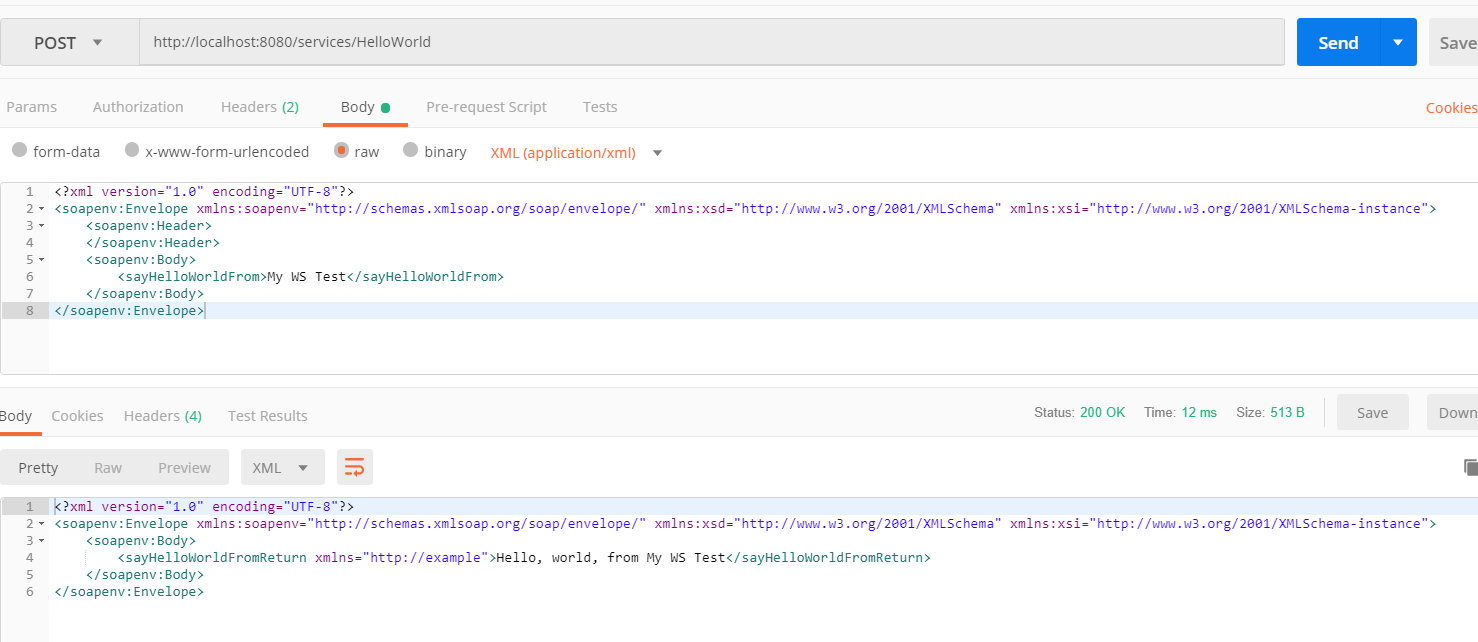
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soapenv:Body>

<sayHelloWorldFromReturn xmlns="http://example">Hello, world, from My WS Test</sayHelloWorldFromReturn>

</soapenv:Body>

</soapenv:Envelope>



OK，请求成功，看到刚才接口的返回了。

可以试试定义更加复杂的参数或返回类型，里面加入更多的业务处理。