SOC-第二次作业

-1140310606 张茗帅

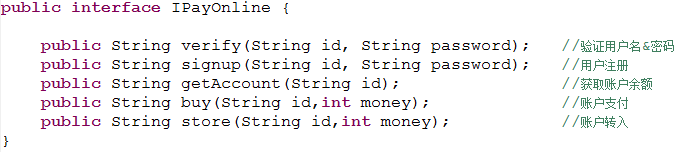
### 一、XFire技术实践

**XFire服务端：**

功能需求：

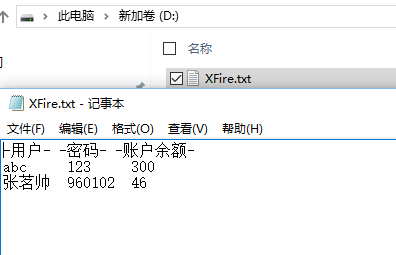
用XFire模拟在线支付功能，具体服务接口为：账户验证、用户余额查询、账户支付、账户转入。

实现步骤：

 按照老师的实验指导书的步骤建立工程，我的主要工作就是实现自己的PayOnlineImpl.java实现类，其中IPayOnline.java接口类定义抽象方法如下：

这五种方法分别对应实验要求的各种功能，在此，我添加了用户注册功能，方便对另外四种方法进行测试。

在实现类PayOnlineImpl.java中，对这5种抽象方法进行了具体的实现。其中对数据的存储形式采用的是文件存储，存储格式如下：



从图中可以看出，每一行主要有三列，分别对应用户名、密码、和账户余额。

各方法核心思想：

**verify方法：**

验证方法，主要通过给定用户输入的用户名和密码和文件中每一行进行匹配，匹配成功即验证成功，返回结果还有用户名正确但是密码错误、以及用户名不存在。

**signup方法：**

注册方法，通过用户给出的用户民和密码注册到服务器端的文件中，同时初始化用户的余额为0。

**getAcount方法：**

获取余额的方法，通过指定的用户名返回该用户的账户余额，实现起来十分简单，只需将用户名这个字符串与文件中的每一行进行比较即可，匹配成功，即返回文件该行该用户所对应的余额信息。

**buy方法：**

该方法用于实现用户支付功能，通过用户id和消费的金额money来返回该用户消费后剩余的金额。实现起来也不难，首先在文件中找到用户id名所对应的列之后，对该行的第三列余额进行改变，此时需要用到RandomAccessFile类来对文件进行读写，因为该类的seek方法可以在指定文件中的的位置修改相关信息，这一点非常适合我这个程序，可以直接对余额值进行修改即可。

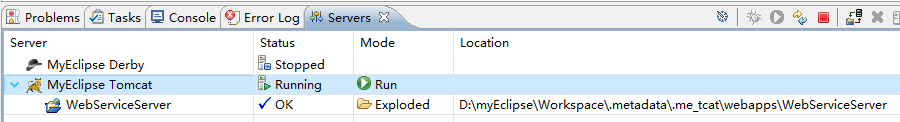
**store方法：**

该方法用于实现账户转入功能，该方法与buy方法十分类似，实现过程基本相同，唯一的区别就是将传入的money与用户原有的余额进行加法运算（在buy方法中，采用的是减法运算）然后更新该列的值即可。

**注：具体代码见WebServiceServer和WerServiceClient文件夹**

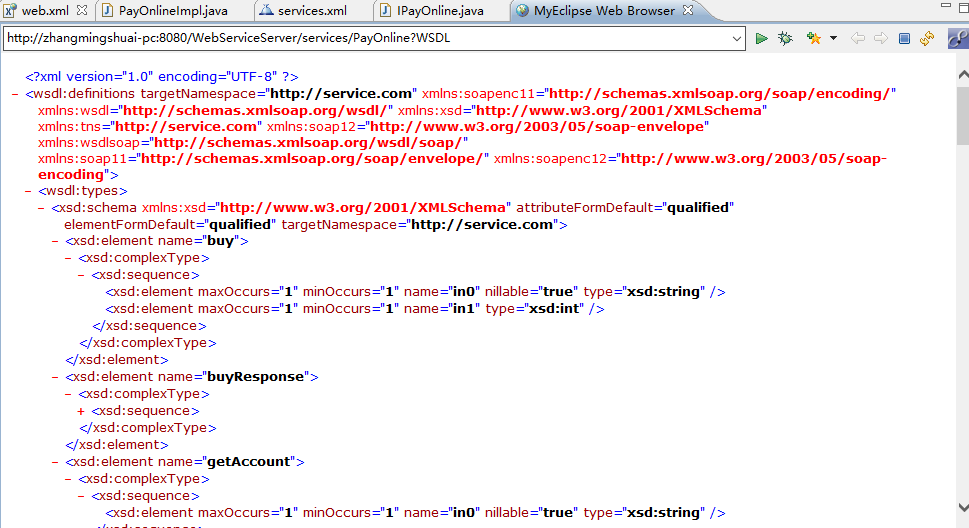
项目部署：

按照要求部署到tomcat上，在此我部署在了myeclipse自带的tomcat服务器上。



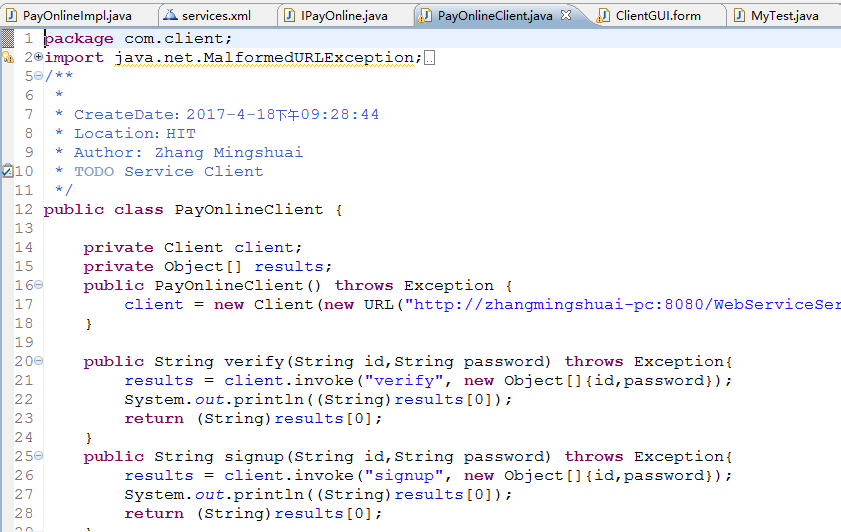
然后打开浏览器，输入<http://zhangmingshuai-pc:8080/WebServiceServer/services/PayOnline?WSDL>

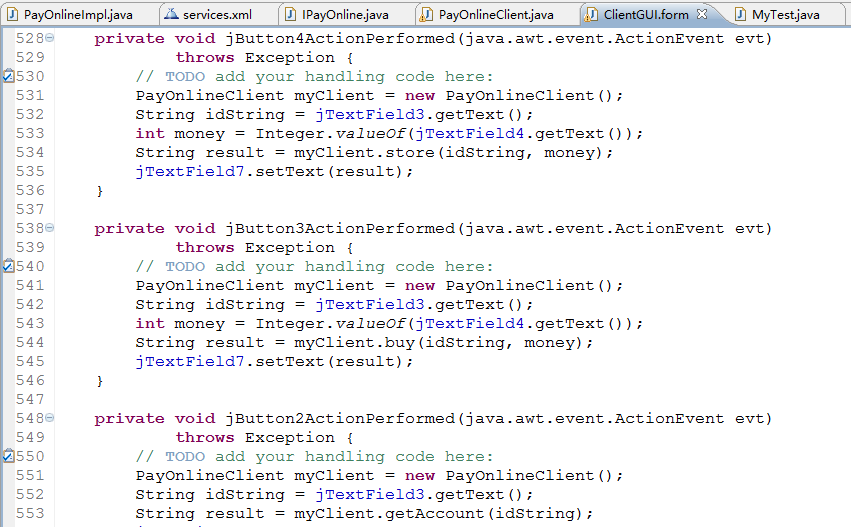
显示如下图：



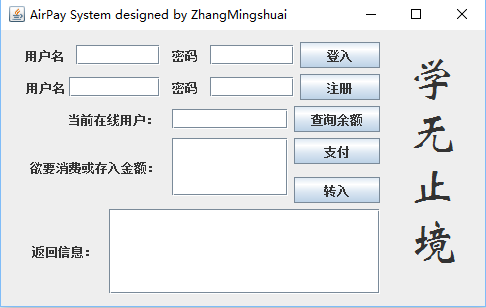
通过此WSDL文件可以看到此WebService服务各方法说明、参数类型等信息。至此WebService服务端建立完毕。

**XFire客户端：**

首先创建一个Java project，按照实验指导书的要求导入相关的libraries，然后实现WebServiceClient类，如下图

 这个类的实现还是十分简单的，在构造方法中通过与已经启动的服务器端的WSDL的URL来进行连接。接下来的每一个方法对应的服务器端相应的方法，在此，获取客户端输入参数通过图形化界面GUI来进行实现，核心实现部分见下图：

通过触发不同的按钮来选择不同的方法，得到的返回值传回至GUI界面中，我的GUI界面基于JFrame框架，大体图如下：



**实例测试：**

首先注册账户名为“服务计算”，密码为“soc”的一个账号



然后发现对应文件成功写入了相关信息，初始化账户余额为：0元



进入登入环节，故意输错密码



进入登入环节，故意输错账户名和密码



进入登入环节，输入正确的账户名和密码



登入成功后，当前在线用户一栏中显示登入成功的用户名“服务计算，点击查询余额”，在返回信息一栏中得到该账户对应的账户余额。



实现账户转入的功能



实现账户支付的功能



再次打印余额信息，依旧正确



## 二、JAX-WS技术实践

**JAX-WS服务器端：**

功能需求：

用JAX-WS模拟实现某公司的产品的信息管理服务，其中至少包括：产品列表查询、某种产品的详情查询（含描述获取，库存查询，价格查询等）、对某种产品的下单、收货确认回执。

实现思想：

仔细想一想，具体实现基本和上一个差不多，我依旧可以用文件保存公司产品的相关信息，分别设置5列：产品名称、产品描述、产品库存、产品价格、收货标志位（此位用于实现“收货确认回执”功能，默认为0，当用户下单后，设置此位为1，收到回执后设置此位为0）。

实现步骤：

按照实验指导书上的步骤进行操作，主要实现自己的创建的类，我创建的类叫做ProductManagement.java，其中包含了主要有8个方法

**add():**

理论上说客户端不应该调用此方法，在这里便于测试，我加入了这个方法。

**getOne(id):**

通过产品名称进行搜索返回该产品的相关全部信息，包含描述、价格、库存、收货标志位。

**getProductList():**

返回产品库中所有的产品名称。

**getDescription(id):**

根据产品名称返回该产品的描述信息。

**getCount(id):**

根据产品名称返回该产品的库存信息。

**getPrice(id):**

根据产品名称返回该产品的价格信息

**buy(id):**

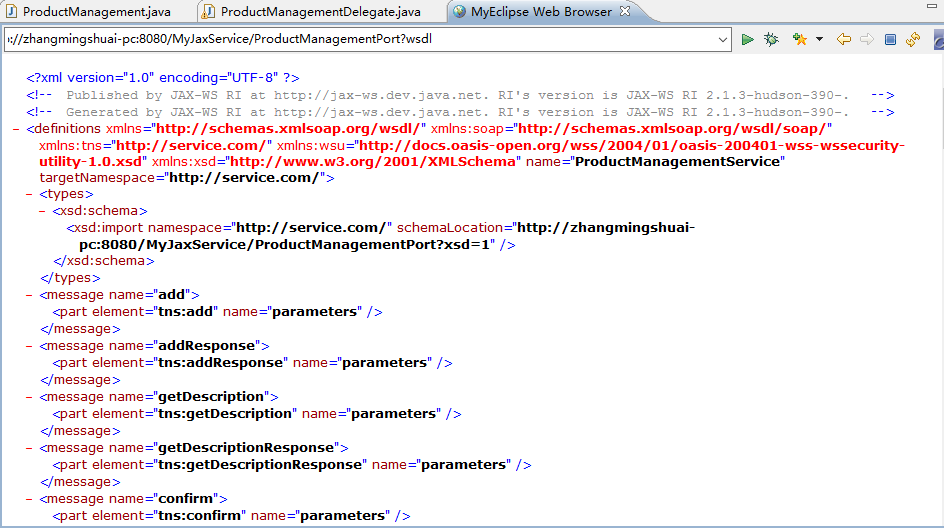
实现用户购买功能，服务端更新该产品的库存数目，同时设置该产品的收货标志位，标志着该产品正在运送途中。

**confirm(id):**

实现“收货确认回执”功能，服务器端将id对应产品名的收货标志位设置为0，代表货物成功送至用户手中。

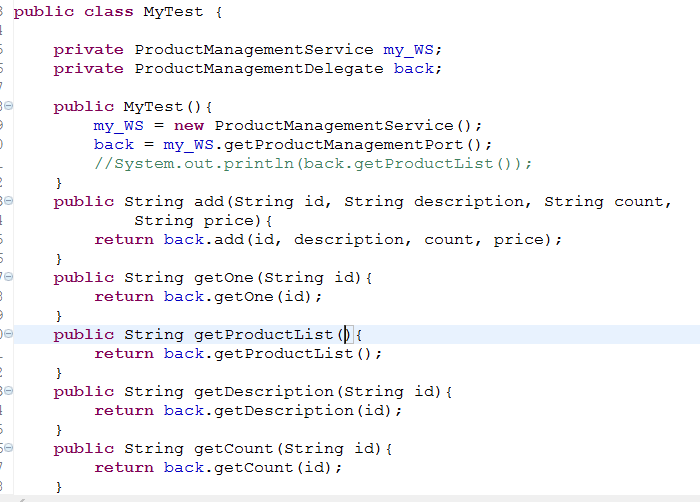
**注：具体代码见MyJaxServer和MyJaxClient文件夹**

然后 New Web Service，生成WSDL文件，部署到tomcat上，启动tomcat，输入 <http://zhangmingshuai-pc:8080/MyJaxService/ProductManagementPort?wsdl> 可得到下图：

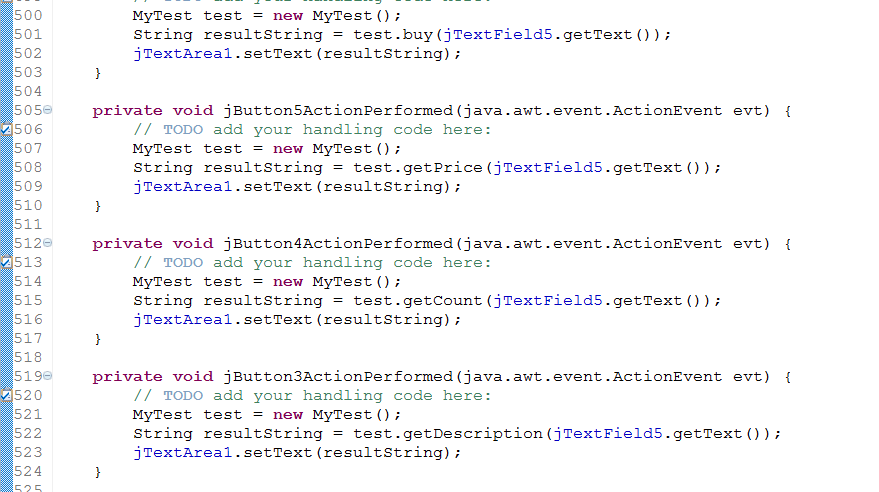


通过此WSDL文件可以看到此WebService服务各方法说明、参数类型等信息。至此基于JAX-WS的WebService服务端建立完毕。

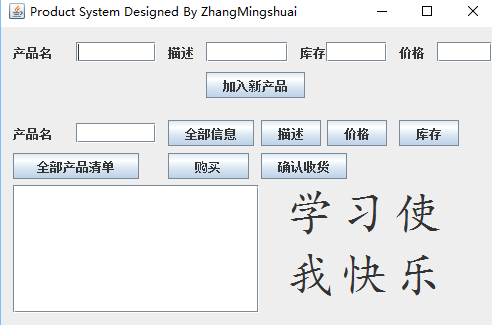
**JAX-WS客户端：**

 客户端的测试我同样增加了图形化UI界面，客户端的建立按照实验指导书一步一步来十分轻松简单，测试类语句很少，见图：

在UI界面中传入参数到MyTest.java即可，然后再得到返回参数显示在UI界面上。见下图部分UI界面代码：

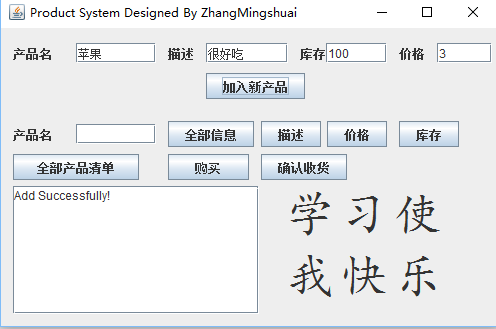


我的UI界面基于JFrame框架，整体面貌如下：

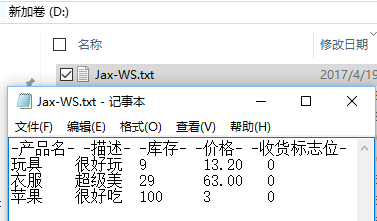


实例测试：

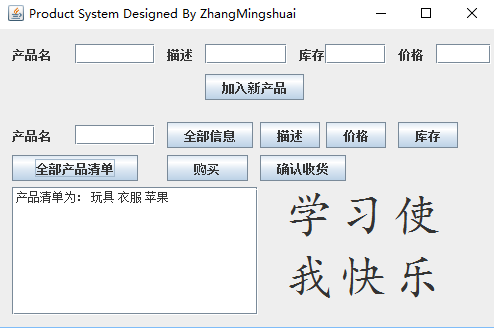
首先在客户端向服务器端加入一组数据（在此设计加入方法只是为了测试方便），我们知道在现实生活中，客户端用户一定是不可以随便更改服务器端的数据的



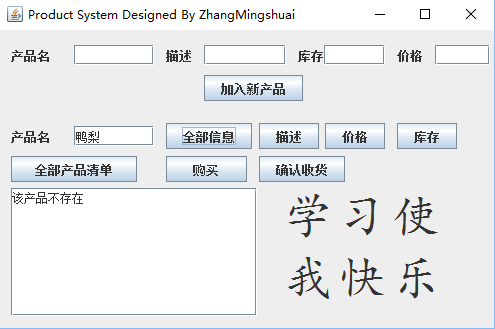
加入成功后可以看到服务器端存储数据的文件中多了一行刚才添加的数据



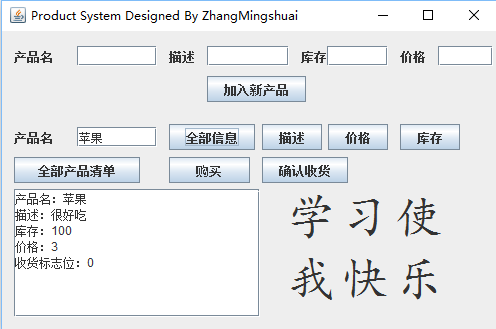
点击“全部产品清单”按钮，可以得到全部的产品名称



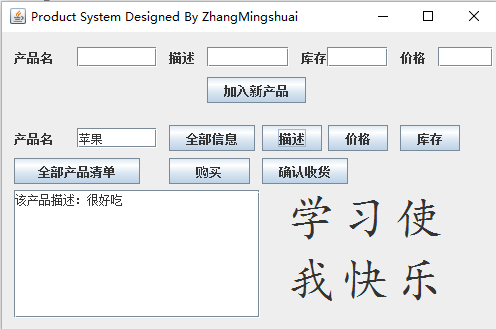
输入一个不存在的产品名，点击“全部信息”按钮，返回信息如下



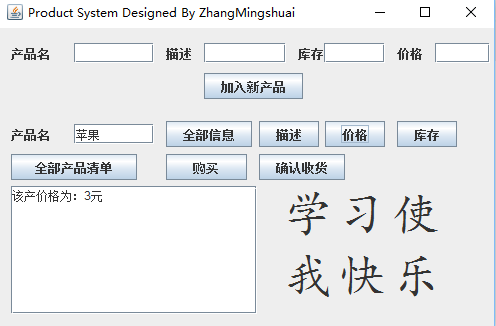
输入一个正确的产品名，点击“全部信息”按钮，返回信息如下



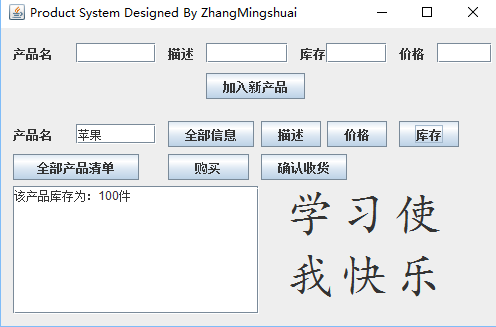
输入一个正确的产品名，点击“描述”按钮，返回信息如下



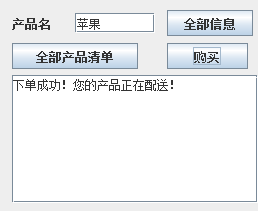
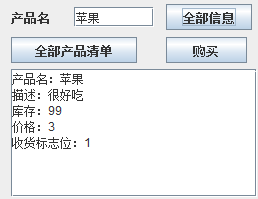
输入一个正确的产品名，点击“价格”按钮，返回信息如下



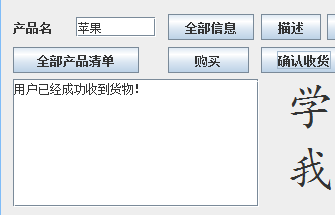
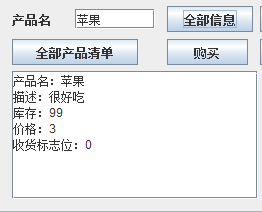
输入一个正确的产品名，点击“库存”按钮，返回信息如下



输入一个正确的产品名，点击“购买”按钮，返回信息如下，同时查看该产品的全部信息，发现该产品 库存少了一个，且收货标志位为1（代表用户还未收到货物）



输入一个正确的产品名，点击“确认收货”按钮，返回信息如下，同时查看该产品的全部信息，发现该产品收货标志位恢复为0（代表用户已经成功收到货物）

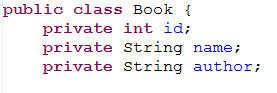


### 三、JAX-RS技术实践

实现功能：

一个简单的书籍管理系统，客户通过简单的URL请求可以获取书籍的详细信息以及添加书籍。

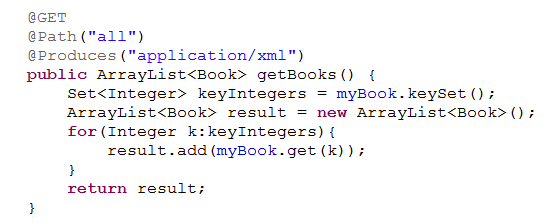
存储结构：主要包含图书id，图书名称，图书作者

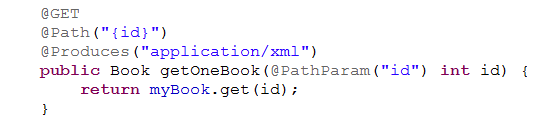


Rest Method：

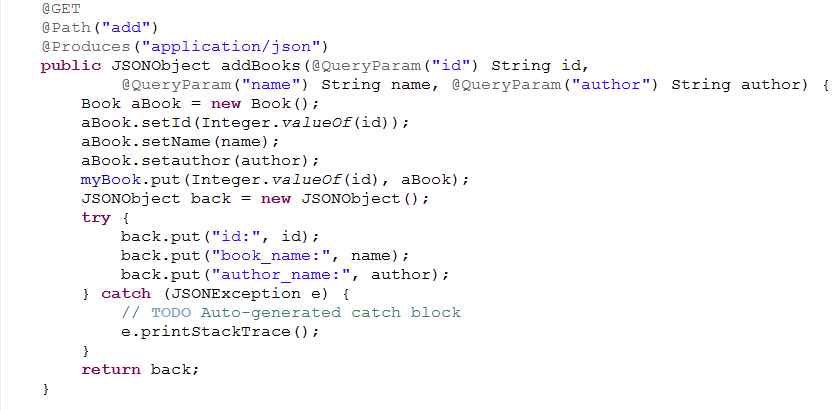
主要包含三个方法

1 此方法getBooks返回书库中每一本书的详细信息（之前每一本书作为值存储在了hashmap myBook中），以XML形式显示。



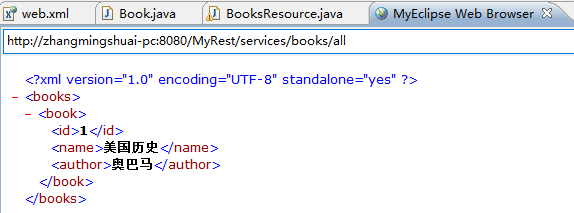
 2 此方法getOneBook通过用户输入的书籍的id号，返回对应书籍的详细信息，以XML形式显示。

3 此方法addBooks用于添加图书，用户在url上输入相关参数，书籍ID，书籍名称，书籍作者。服务器端接受这些参数并将他们保存到书库中，返回文本为json型。

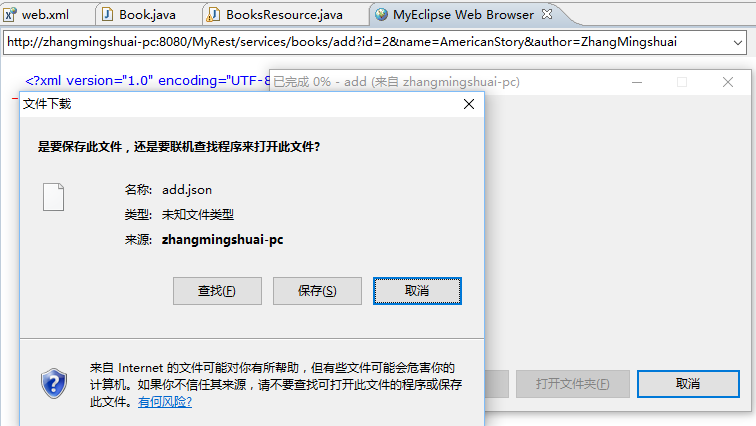


测试：

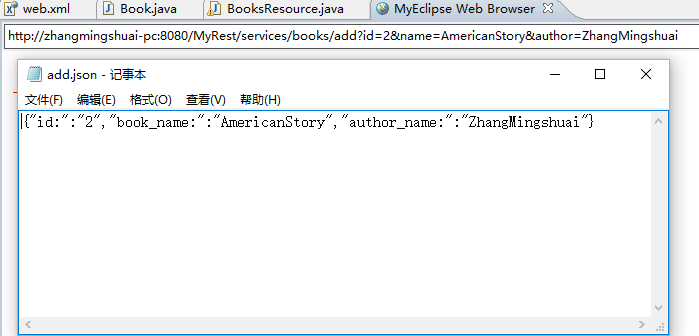
首先输入/books/all，则会找到对应getBooks方法，返回如下（书库中固有一本书籍）：



然后测试添加功能，输入各参数（QueryParameter型），由于文本返回为json，好像是网页不能直接读取显示，它提示我下载这个文件，我便下载了。



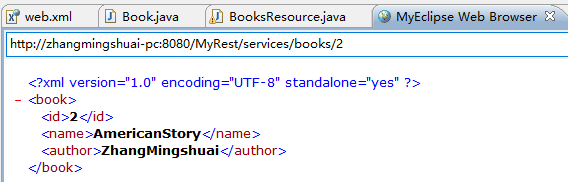
将json文件下载过后打开，可以看到，json中内容如下，为我刚才传递的参数值

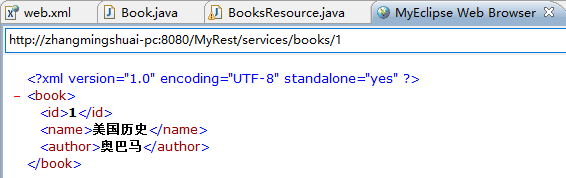


再查看一下全部书籍信息，发现书库中多了我刚才添加的那本书。



然后测试查询功能，按照PathParameter的参数传递方式分别查询id为1和2的书籍信息如下：





**注：本部分具体代码见MyRest文件夹**

### 四、Rest API技术实践

我在聚合数据网<https://www.juhe.cn/>上注册了一个账号，并且免费申请了一个“手机号码归属地查询”的API对应的AppKey



获取了这个AppKey，我就可以在java程序设计中调用它了，在此，我主要做了一个小程序，实现的功能就是对手机号码归属地的查询。

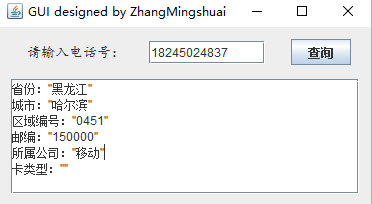
注：本部分代码见MyAPI文件夹

测试：

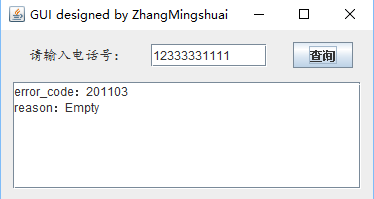
输入了我老家那头的电话，得到返回信息如下



再测试我们校园卡的一个号码，得到返回结果如下



输入一个错误的电话号码，得到的返回信息如下图



至此，全部实验已经顺利完成！源代码请参考各文件夹！