主要实现的功能

通过Swagger Codgen生成前端调用后端的接口

前端Springboot+Themeleaf

后端SpringBoot

打开SwaggerUI

**踩过的坑**

**[使用Swagger-codgen必须指定返回值类型]**

**[**@RequestMapping(value={""}, method=RequestMethod.GET,**produces = "application/json"**)**]**

[root@192 ~]# cd record/

[root@192 record]# ls

swagger-ui-master swagger-ui-master.zip

[root@192 record]# cd swagger-ui-master

[root@192 swagger-ui-master]# ls

bower.json CONTRIBUTING.md Dockerfile index.js karma.conf.js lib nginx.conf README.md test

composer.json dist gulpfile.js ISSUE\_TEMPLATE.md lang LICENSE package.json src

[root@192 swagger-ui-master]# docker build -t swagger-ui-builder .

Sending build context to Docker daemon 5.178 MB

Step 1 : FROM alpine:3.3

---> 6c2aa2137d97

Step 2 : MAINTAINER Roman Tarnavski

---> Using cache

---> 9218e2df36d1

Step 3 : RUN apk add --update nginx

---> Using cache

---> 6bf063f1abc2

Step 4 : COPY nginx.conf /etc/nginx/

---> Using cache

---> 8c1cd4119ddc

Step 5 : ADD ./dist/ /usr/share/nginx/html

---> Using cache

---> ceda690de9c8

Step 6 : EXPOSE 8080

---> Using cache

---> fbd94214b1d9

Step 7 : CMD nginx -g 'daemon off;'

---> Using cache

---> 52e7dfd9b86f

Successfully built 52e7dfd9b86f

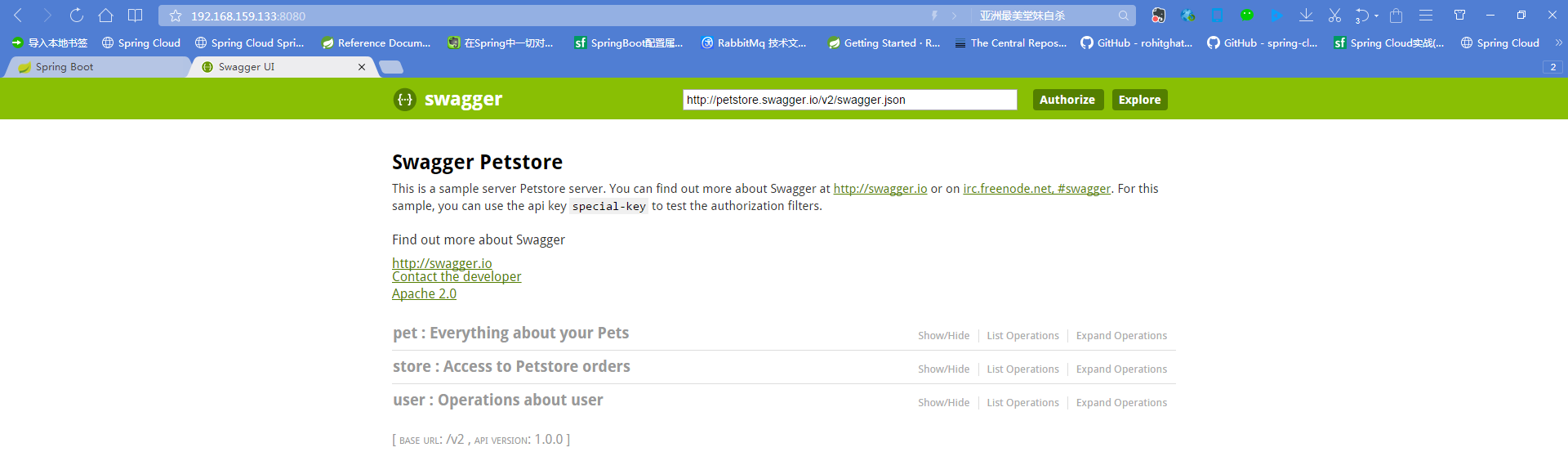
[root@192 swagger-ui-master]# docker run -p 8080:8080 swagger-ui-builder

[root@192 ~]# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

35a884805a98 swagger-ui-builder "/bin/sh -c 'nginx -g" 41 seconds ago Up 39 seconds 0.0.0.0:8080->8080/tcp drunk\_sinoussi

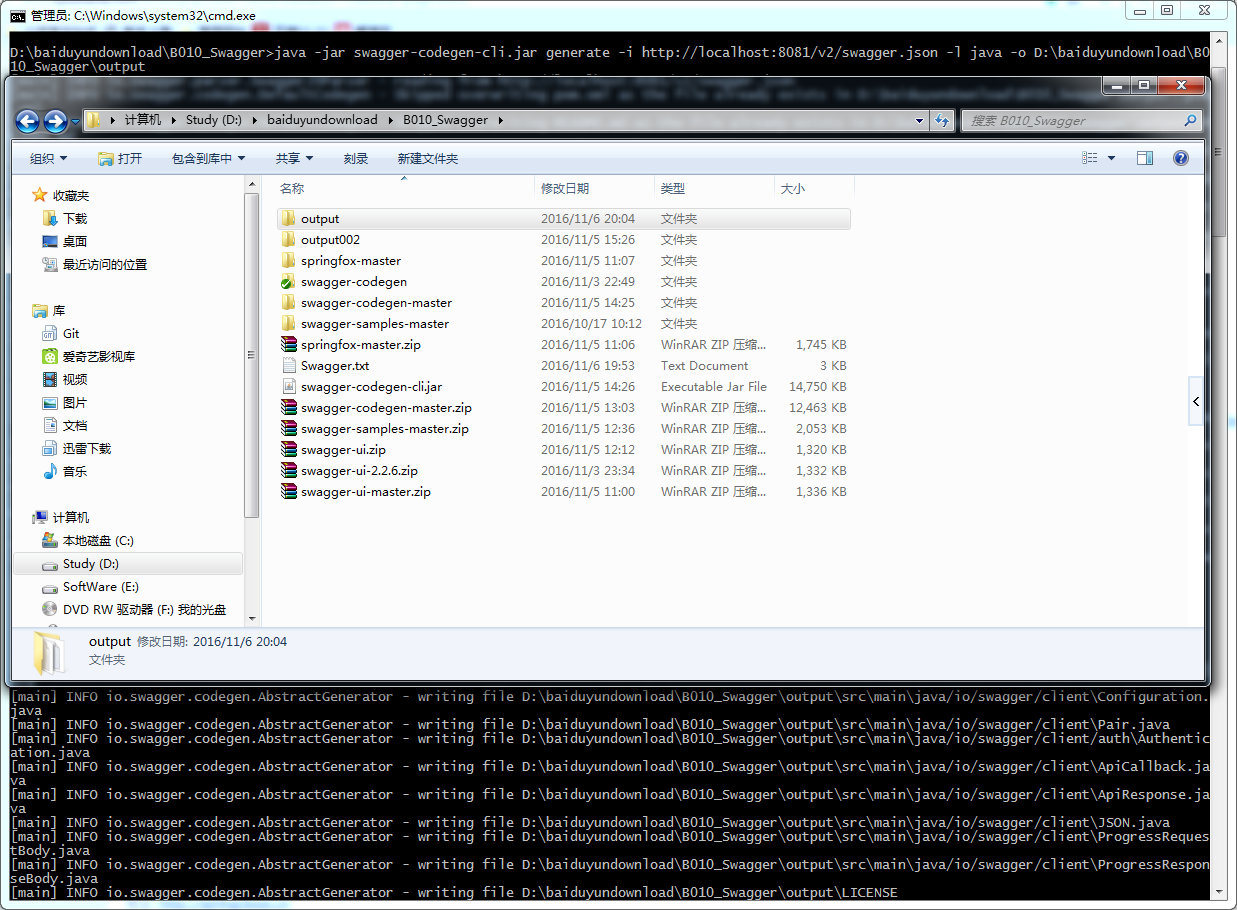
[root@192 ~]#



2启动后端为服务项目



3使用swagger-codegen生成接口工程

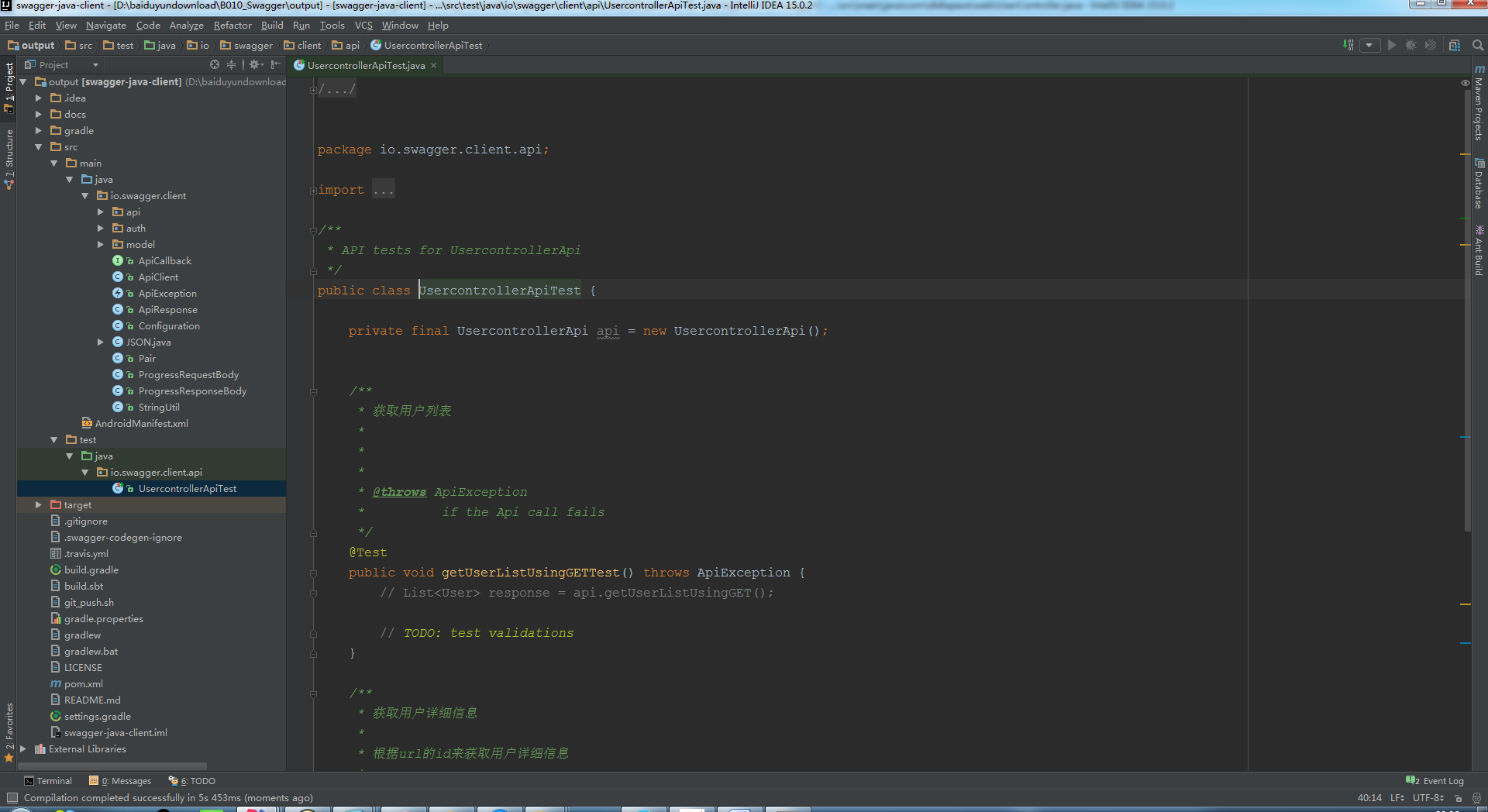


4.打开工程

<java.version>1.7</java.version>

变成

<java.version>1.8</java.version>



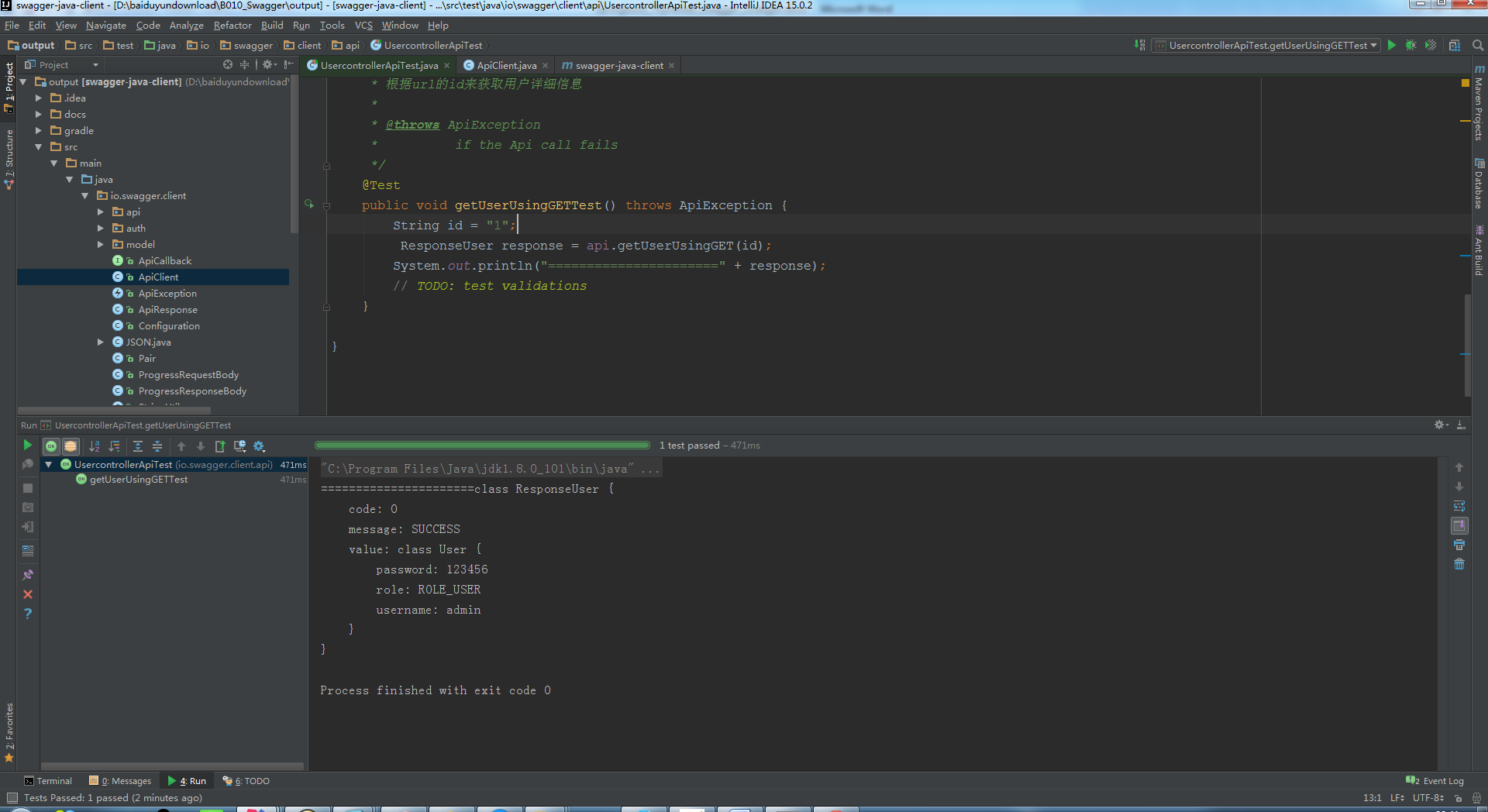
修改地址：

public class ApiClient {

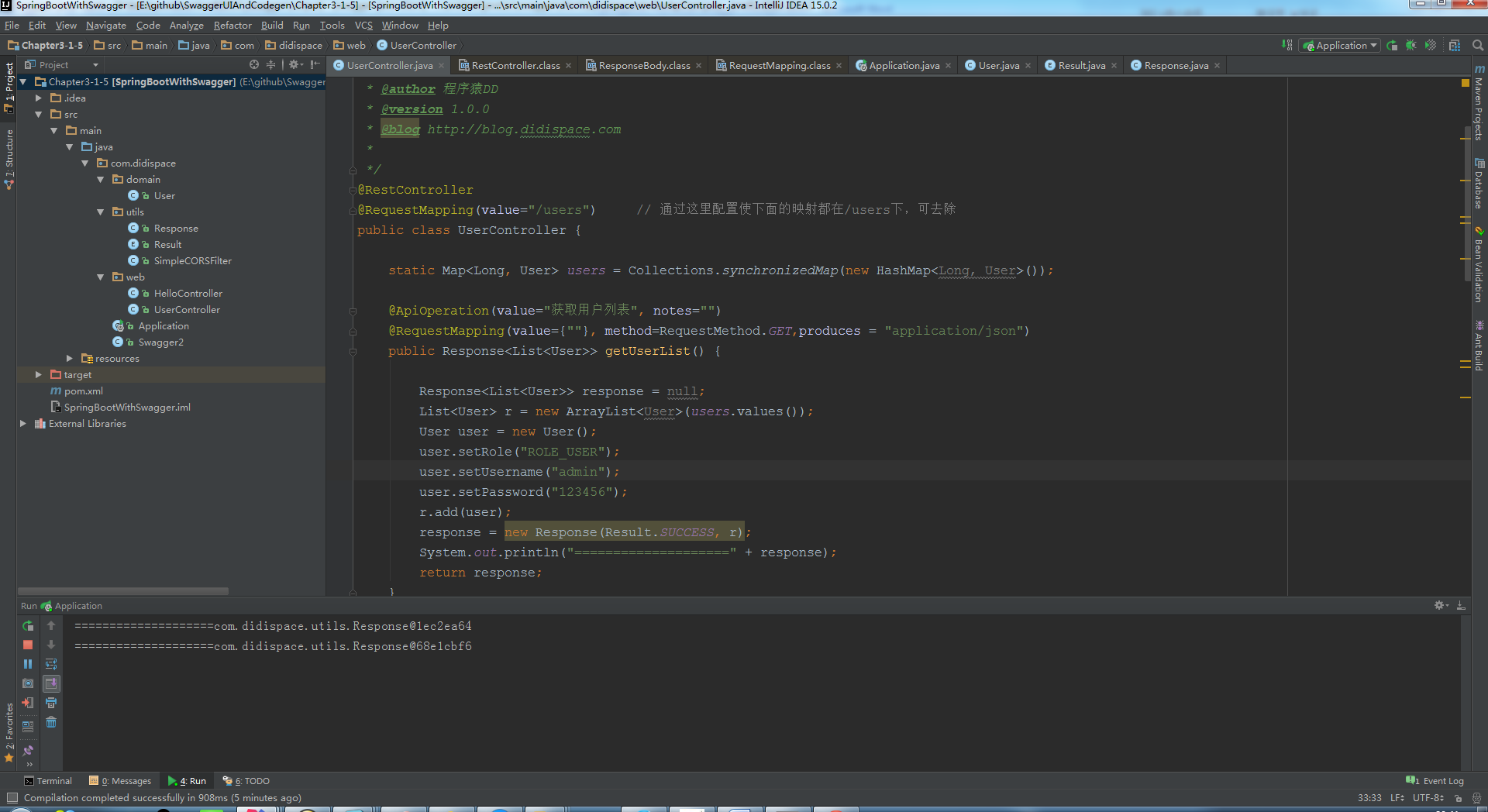
// private String basePath = "https://localhost:8081/v2";  
 private String basePath = "http://localhost:8081/v2";

进行测试

生成的接口工程

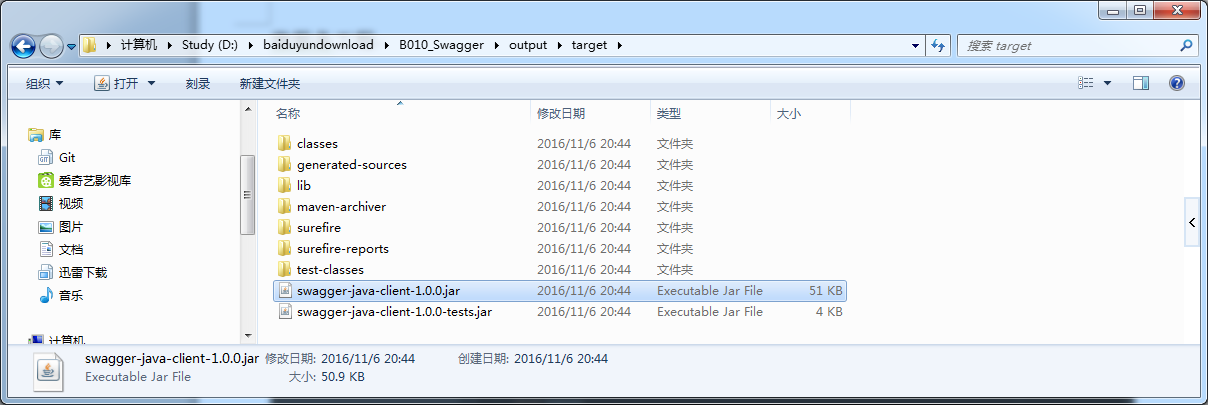


微服务工程



将生成的接口工程打包：

Mvn clean install



画面侧追加Mvn依赖

<dependencies>

<dependency>

<groupId>io.swagger</groupId>

<artifactId>swagger-java-client</artifactId>

<version>1.0.0</version>

<scope>system</scope>

<systemPath>${project.basedir}/src/main/resources/lib/ swagger-java-client-1.0.0.jar</systemPath>

</dependency>

添加的配置文件

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<java.version>1.8</java.version>

<swagger-core-version>1.5.9</swagger-core-version>

<okhttp-version>2.7.5</okhttp-version>

<gson-version>2.6.2</gson-version>

<jodatime-version>2.9.3</jodatime-version>

<maven-plugin-version>1.0.0</maven-plugin-version>

<junit-version>4.12</junit-version>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependency>

<groupId>io.swagger</groupId>

<artifactId>swagger-annotations</artifactId>

<version>${swagger-core-version}</version>

</dependency>

<dependency>

<groupId>com.squareup.okhttp</groupId>

<artifactId>okhttp</artifactId>

<version>${okhttp-version}</version>

</dependency>

<dependency>

<groupId>com.squareup.okhttp</groupId>

<artifactId>logging-interceptor</artifactId>

<version>${okhttp-version}</version>

</dependency>

<dependency>

<groupId>com.google.code.gson</groupId>

<artifactId>gson</artifactId>

<version>${gson-version}</version>

</dependency>

<dependency>

<groupId>joda-time</groupId>

<artifactId>joda-time</artifactId>

<version>${jodatime-version}</version>

</dependency>