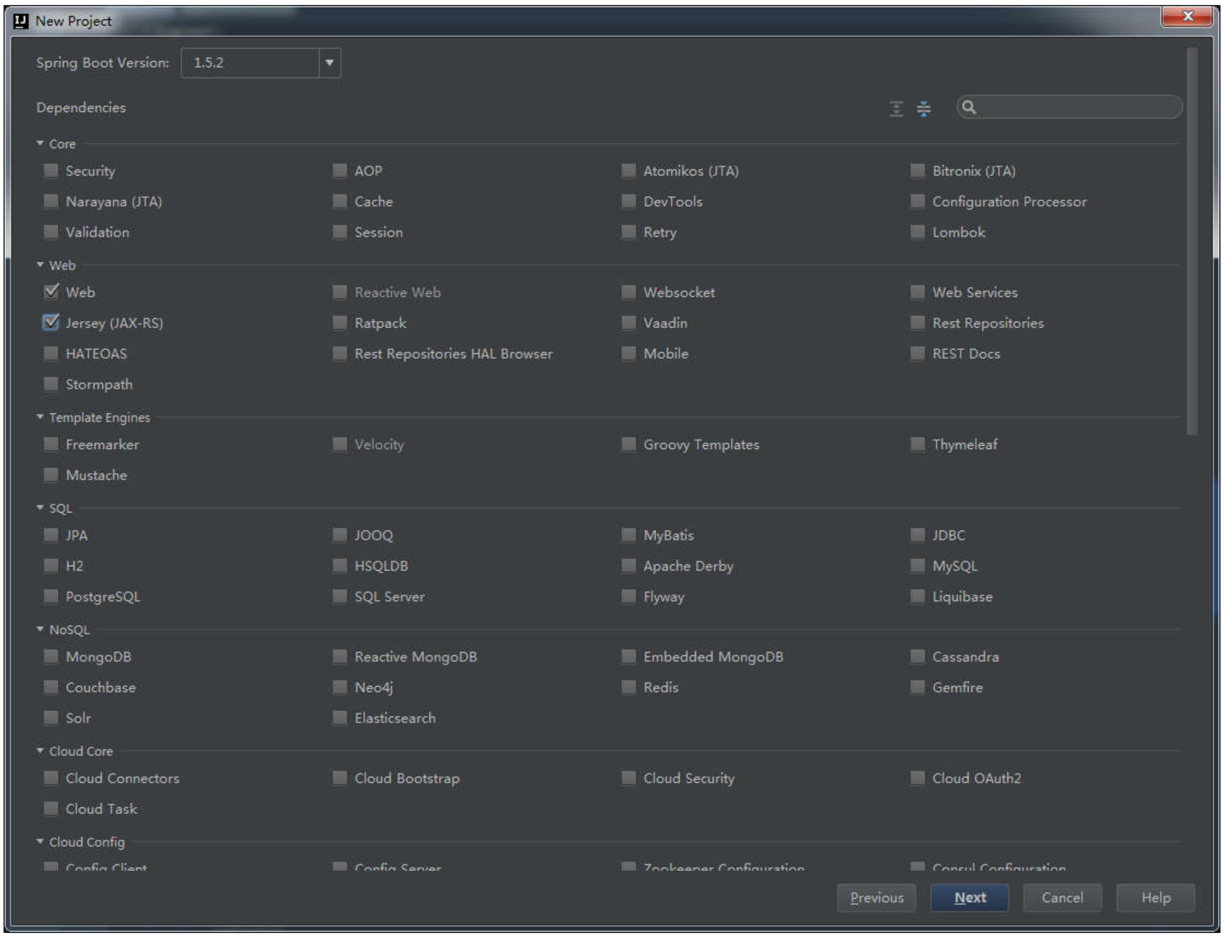
Springboot+Swagger

1 SpringBoot+Swagger

1.1 创建项目 SpringBoot+Jersey:


New Project

Spring Boot Version: 1.5.2

Dependencies

Web: [✓ Web]

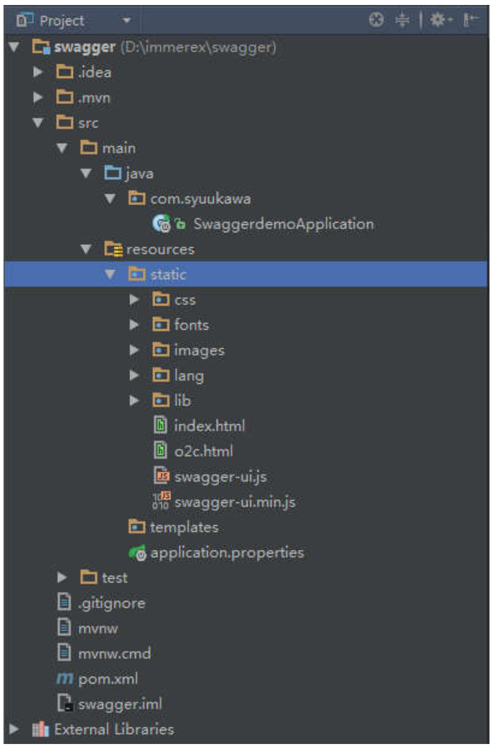
Jersey (JAX-RS): [✓ Jersey (JAX-RS)]

1.2 POM 文件中添加 Swagger 配置:

```
●     <!--swagger jar add-->
   <dependency>
      <groupId>io.swagger</groupId>
      <artifactId>swagger-jersey2-jaxrs</artifactId>
      <version>1.5.10</version>
   </dependency>
●     <dependency>
      <groupId>io.swagger</groupId>
      <artifactId>swagger-annotations</artifactId>
      <version>1.5.9</version>
   </dependency>
   <dependency>
      <groupId>io.swagger</groupId>
      <artifactId>swagger-annotations</artifactId>
```

```
        <version>1.5.9</version>
    </dependency>
```

1.3 在 resource-static 中添加 swagger 需要文件:



1.4 添加配置类:

ApiOriginFilter:

```
public class ApiOriginFilter implements Filter {
   @Override
   public void init(FilterConfig filterConfig) throws ServletException {

   }
   @Override
    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws
IOException, ServletException {
        HttpServletResponse response = (HttpServletResponse) res;
        response.setHeader("Access-Control-Allow-Origin", "*");
        response.setHeader("Access-Control-Allow-Methods", "GET, POST, DELETE, PUT, PATCH,
OPTIONS");
        response.setHeader("Access-Control-Max-Age", "3600");
        response.setHeader("Access-Control-Allow-Headers", "Origin, X-Requested-With,
Content-Type, Accept, api_key, Authorization");
//          response.setHeader("Content-Type", "application/json");
        chain.doFilter(req, res);
    }
   @Override
   public void destroy() {
    }
}
```

JerseyConfig:

```
@Component
public class JerseyConfig extends ResourceConfig {

    @Value("${spring.jersey.application-path:/}")
    private String apiPath;

    public JerseyConfig() {
        // Register endpoints, providers, ...
        this.registerEndpoints();
    }

    @PostConstruct
    public void init() {
        // Register components where DI is needed
        this.configureSwagger();
    }

    private void registerEndpoints() {
        packages("com.syuukawa");
        // Access through /<Jersey's servlet path>/application.wadl
        this.register(WadlResource.class);
    }

    private void configureSwagger() {
        // Available at localhost:port/swagger.json
        this.register(ApiListingResource.class);
        this.register(SwaggerSerializers.class);

        BeanConfig config = new BeanConfig();
        config.setConfigId("springboot-jersey-swagger-docker-example");
        config.setTitle("Spring Boot + Jersey + Swagger + Docker Example");
        config.setVersion("v1");
        config.setContact("Orlando L Otero");
        config.setSchemes(new String[]{"http", "https"});
        config.setBasePath(this.apiPath);
        //TODO need to modify
        config.setResourcePackage("com.syuukawa");
        config.setPrettyPrint(true);
        config.setScan(true);
    }
}
```

1.5 修改 application.properties:

```
server.port= 8081
spring.jersey.application-path= /api
```

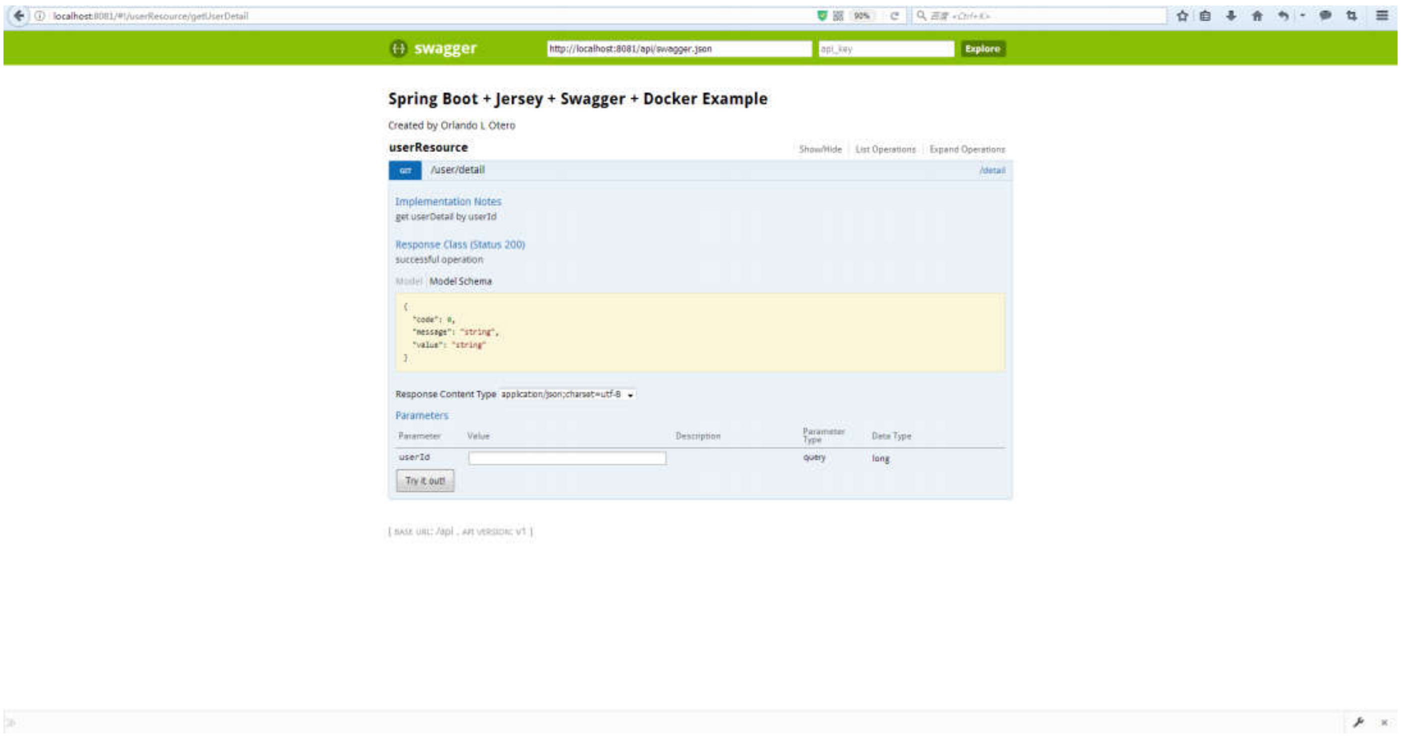
1.6 添加相关测试类:

```
@Path("/user")
@Api(value = "/userResource", description = "description")
@SuppressWarnings("all")
public class UserDetailResource {
    private static final Logger log = LoggerFactory.getLogger(UserDetailResource.class);

    /**
     * <p>Description: 这里用一句话描述这个方法的作用</p>
     * <p>param </p>
     * <p>author zhouhe </p>
     * <p>date 2017/3/17 14:28 </p>
     * <p>return </p>
     */
    @ApiOperation(value = "/detail", notes = "get userDetail by userId")
    @GET
    @Path("/detail")
    @Produces({"application/json;charset=utf-8"})
    @Consumes({"application/json;charset=utf-8"})
    public Response<String> getUserDetail(@QueryParam("userId") Long userId) {
        System.out.println("========="+ userId);
        return  new Response(Result.SUCCESS,userId);
    }
}
```

1.7 运行结果:

Localhost:8081


localhost:8081/#!/userResource/getUserDetail
swagger
http://localhost:8081/api/swagger.json
api_key
Explore

Spring Boot + Jersey + Swagger + Docker Example

Created by Orlando L Otero

userResource

Show/Hide
List Operations
Expand Operations
GET

/user/detail

Implementation Notes

get userDetail by userId

Response Class (Status 200)

successful operation

```
{
  "code": 0,
  "message": "string",
  "value": "string"
}
```

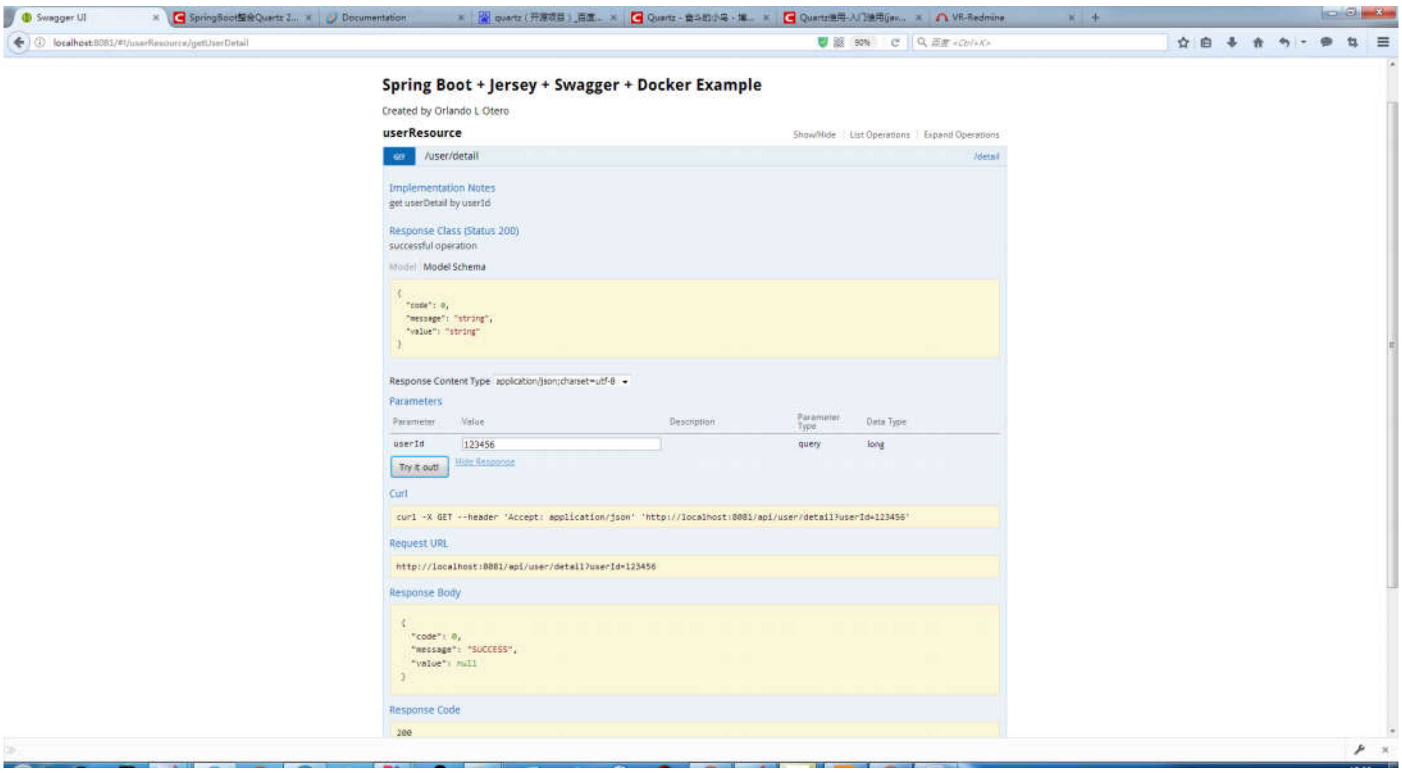
Response Content Type application/json;charset=utf-8

Parameters

Parameter	Value	Description	Parameter Type	Data Type
userId			query	long

[BASE URL: /api , API VERSION: v1]

1.8 进行接口测试：



Spring Boot + Jersey + Swagger + Docker Example

Created by Orlando L Otero

userResource

/user/detail

Implementation Notes

get userDetail by userId

Response Class (Status 200)

successful operation

```
{
  "code": 0,
  "message": "string",
  "value": "string"
}
```

Response Content Type application/json;charset=utf-8

Parameters

Parameter	Value	Description	Parameter Type	Data Type
userId	123456		query	long

Curl

```
curl -X GET --header 'Accept: application/json' 'http://localhost:8081/api/user/detail?userId=123456'
```

Request URL

```
http://localhost:8081/api/user/detail?userId=123456
```

Response Body

```
{
  "code": 0,
  "message": "SUCCESS",
  "value": null
}
```

Response Code

```
200
```

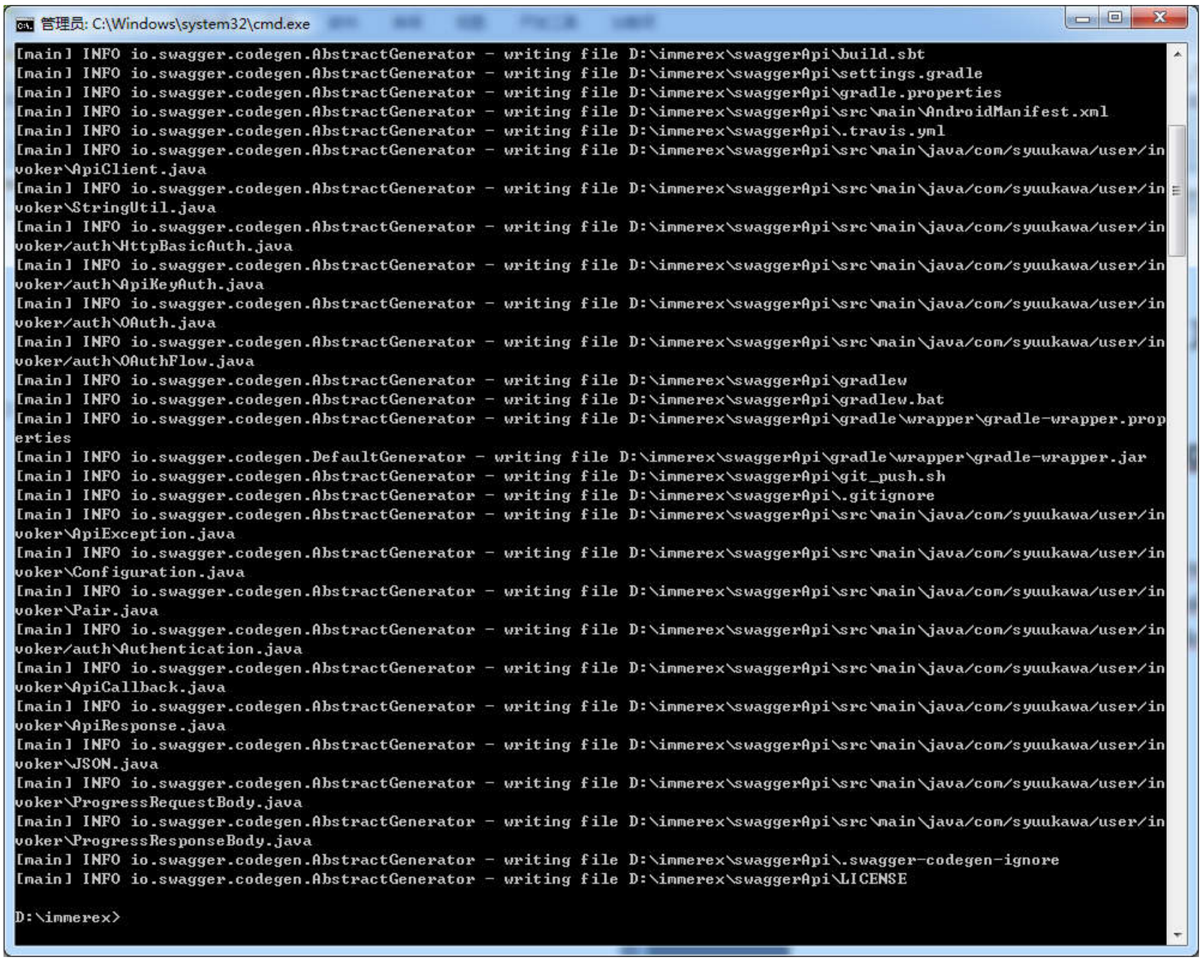
2 SwaggerApi 项目生成

2.1 Swagger-config.json

```
{
    "apiPackage":"com.syuukawa.user.api",
    "invokerPackage":"com.syuukawa.user.invoker",
    "modelPackage":"com.syuukawa.user.entity",
    "dateLibrary":"legacy",
    "serializableModel":true,
    "fullJavaUtil":true
  }
```

2.2 执行命令

```
java -jar swagger-codegen-cli-2.2.1.jar generate -i http://localhost:8081/api/swagger.json -l java -c D:\immerex\swagger_config.json -o D:\immerex\swaggerApi
```

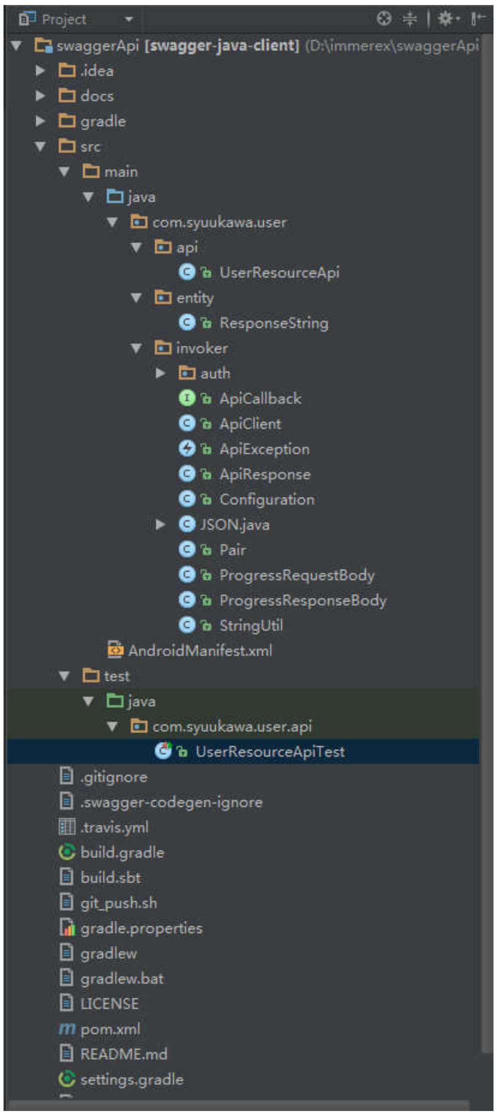


管理员: C:\Windows\system32\cmd.exe

```
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\build.sbt
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\settings.gradle
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\gradle.properties
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\AndroidManifest.xml
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\.travis.yml
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ApiClient.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\StringUtil.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker/auth\HttpBasicAuth.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker/auth\ApiKeyAuth.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker/auth\OAuth.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker/auth\OAuthFlow.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\gradlew
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\gradlew.bat
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\gradle\wrapper\gradle-wrapper.prop
erties
[main] INFO io.swagger.codegen.DefaultGenerator - writing file D:\immerex\swaggerApi\gradle\wrapper\gradle-wrapper.jar
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\git_push.sh
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\.gitignore
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ApiException.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\Configuration.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\Pair.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker/auth\Authentication.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ApiCallback.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ApiResponse.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\JSON.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ProgressRequestBody.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\src\main\java/com/syuukawa/user/in
voker\ProgressResponseBody.java
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\.swagger-codegen-ignore
[main] INFO io.swagger.codegen.AbstractGenerator - writing file D:\immerex\swaggerApi\LICENSE

D:\immerex>
```

2.3 SwaggerApi 项目结构



```
Project
swaggerApi [swagger-java-client] (D:\immerex\swaggerApi
  .idea
  docs
  gradle
  src
    main
      java
        com.syuukawa.user
          api
            UserResourceApi
          entity
            ResponseString
          invoker
            auth
            ApiCallback
            ApiClient
            ApiException
            ApiResponse
            Configuration
            JSON.java
            Pair
            ProgressRequestBody
            ProgressResponseBody
            StringUtil
      AndroidManifest.xml
    test
      java
        com.syuukawa.user.api
          UserResourceApiTest
  .gitignore
  .swagger-codegen-ignore
  .travis.yml
  build.gradle
  build.sbt
  git_push.sh
  gradle.properties
  gradlew
  gradlew.bat
  LICENSE
  pom.xml
  README.md
  settings.gradle
```

2.4 测试

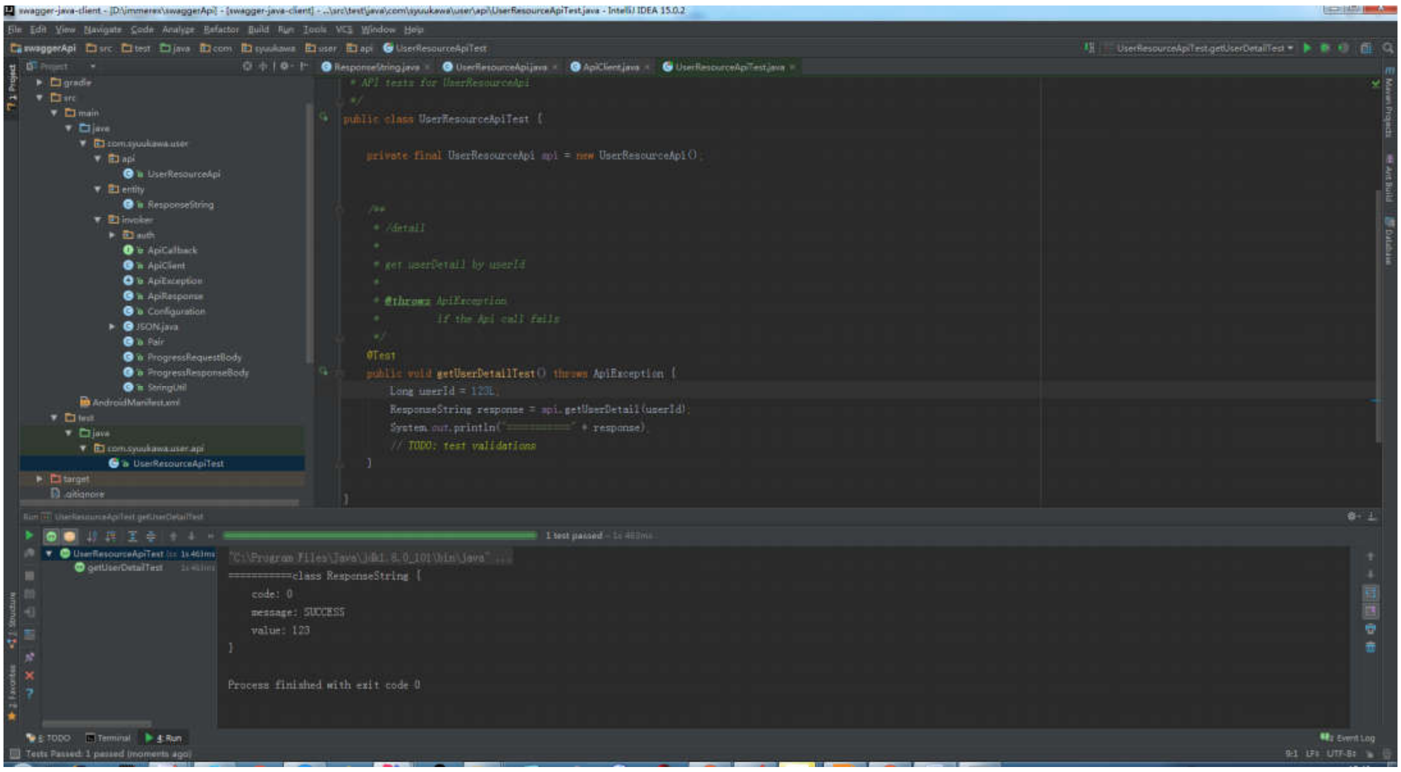
1、修改 ApiClient 中的 basePath

```
private String basePath = "http://localhost:8081/api";
```

2、修改测试类中的方法

```
@Test
public void getUserDetailTest() throws ApiException {
    Long userId = 123L;
    ResponseString response = api.getUserDetail(userId);
    System.out.println("==========" + response);
    // TODO: test validations
}
```

3、测试结果如下


```java
 * API tests for UserResourceApi
 */
public class UserResourceApiTest {

    private final UserResourceApi api = new UserResourceApi();


    /**
     * /detail
     *
     * get userDetail by userId
     *
     * @throws ApiException
     *          if the Api call fails
     */
    @Test
    public void getUserDetailTest() throws ApiException {
        Long userId = 123L;
        ResponseString response = api.getUserDetail(userId);
        System.out.println("==========" + response);
        // TODO: test validations
    }

}
```

```
"C:\Program Files\Java\jdk1.8.0_101\bin\java" ...
==========class ResponseString {
    code: 0
    message: SUCCESS
    value: 123
}

Process finished with exit code 0
```

2.5 后续调用

1、 SwaggerApi 项目可以打成 jar，放到前端项目中，实现前端---SwaggerApi---后端的调用，此处省略。

```xml
<dependency>
    <groupId>f5</groupId>
    <artifactId>iControl</artifactId>
    <version>11.4.1</version>
    <scope>system</scope>
    <systemPath>${project.basedir}/src/main/webapp/WEB-INF/lib/iControl.jar</systemPath>
</dependency>
```

2、 前端项目可以引入 SwaggerApi 项目，实现前端---SwaggerApi---后端的调用，此处省略。

```xml
<dependency>
    <groupId>SwaggerApi</groupId>
    <artifactId>SwaggerApi</artifactId>
    <version>1.0.0</version>
</dependency>
```

九天