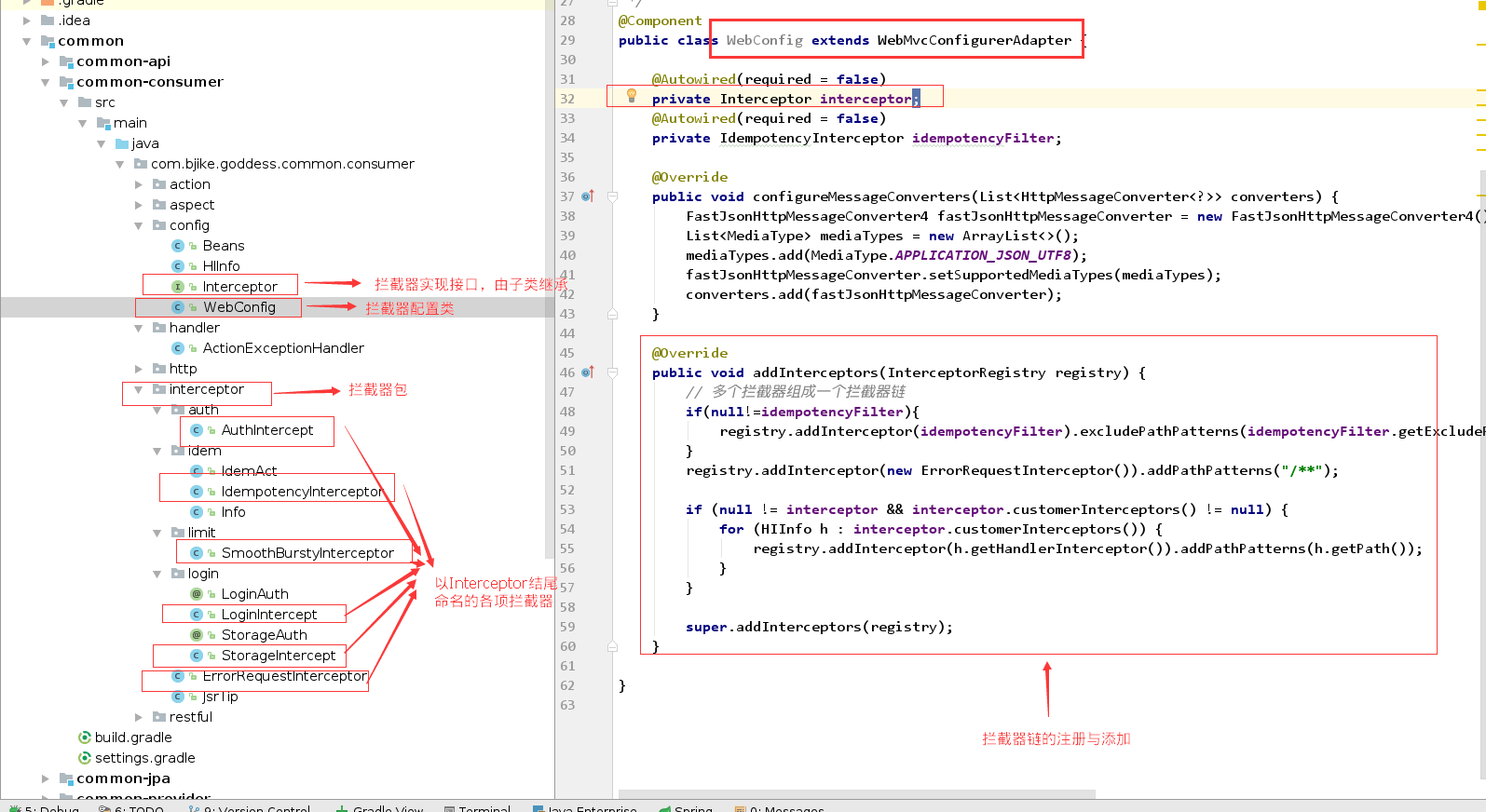
## GODDESS通用功能说明

**common-consumer:**

1. **拦截器**

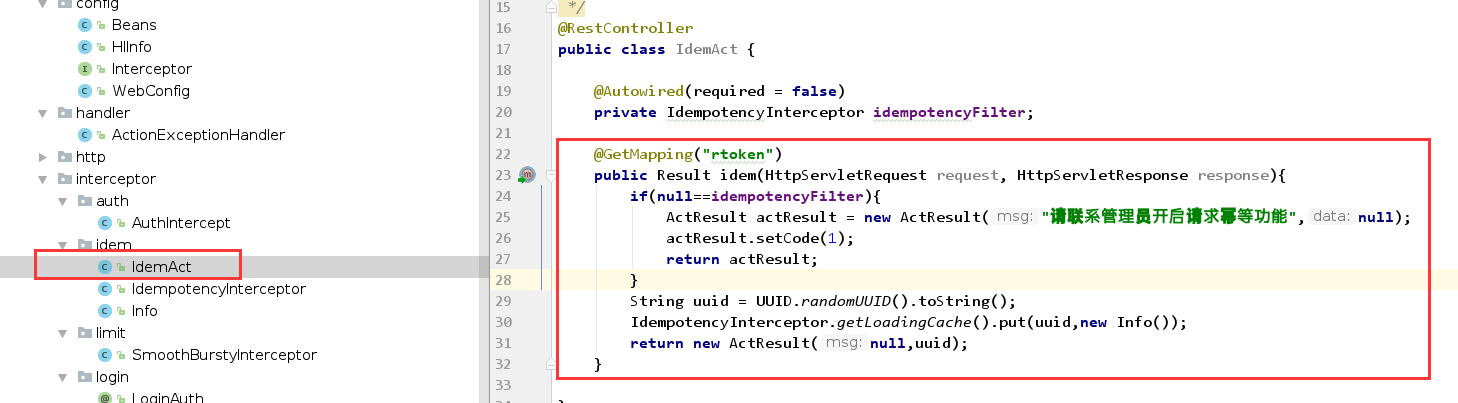
拦截器命名以Interceptor为命名结尾，拦截器位于common包下，但

都均没有扫描注入，而是由子类继承自由注入添加自己的拦截器链，请看下图拦截器目录结构：



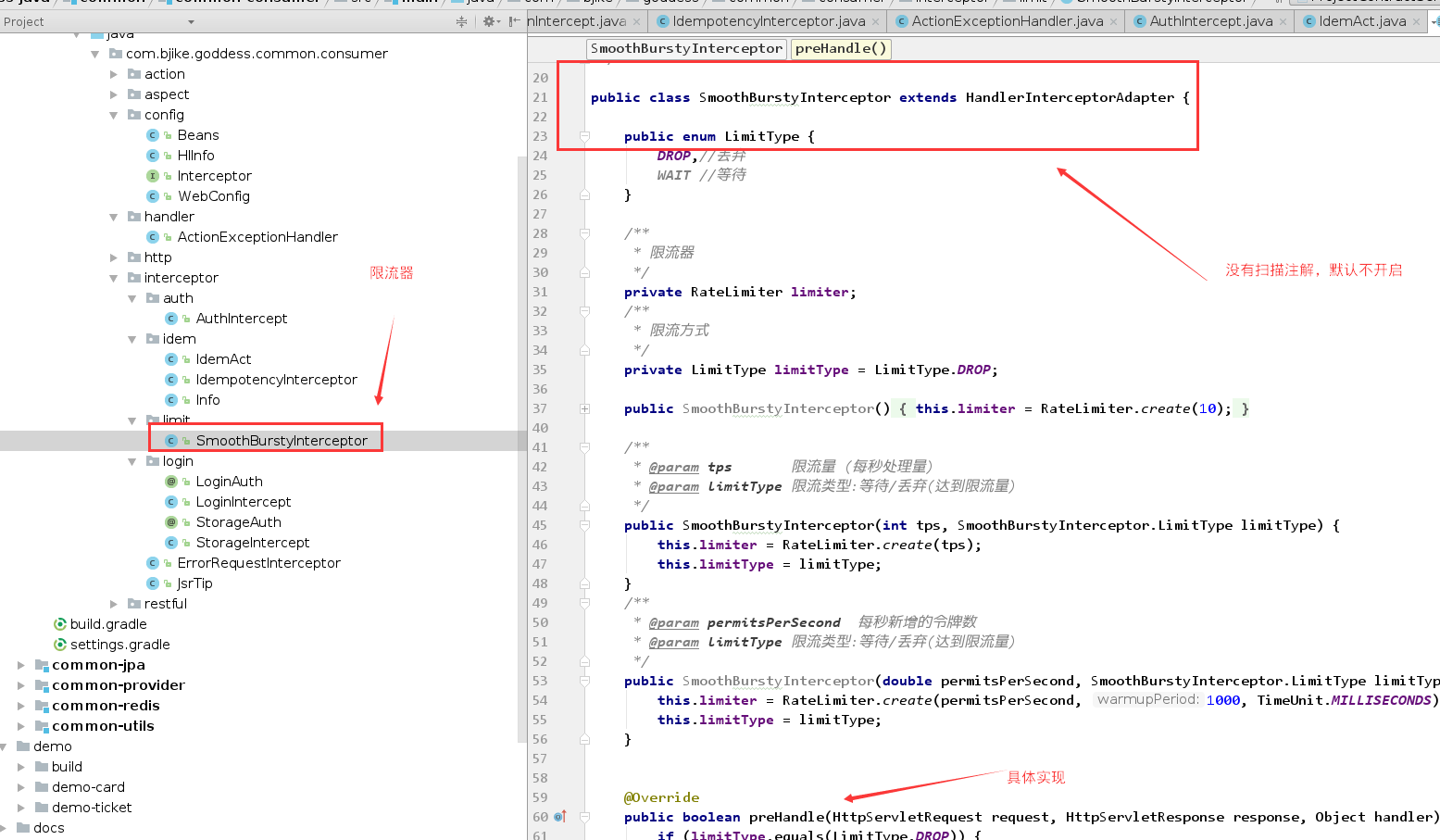
**IdempotencyInterceptor**（幂等拦截）：

针对多次提交做出处理，该拦截器默认全局开启，在webConfig.java注入，不由子类控制。一但开启此拦截器，则每次请求需要获取到rtoken，且请求时header携带新的rtoken，rtoken获取如下：

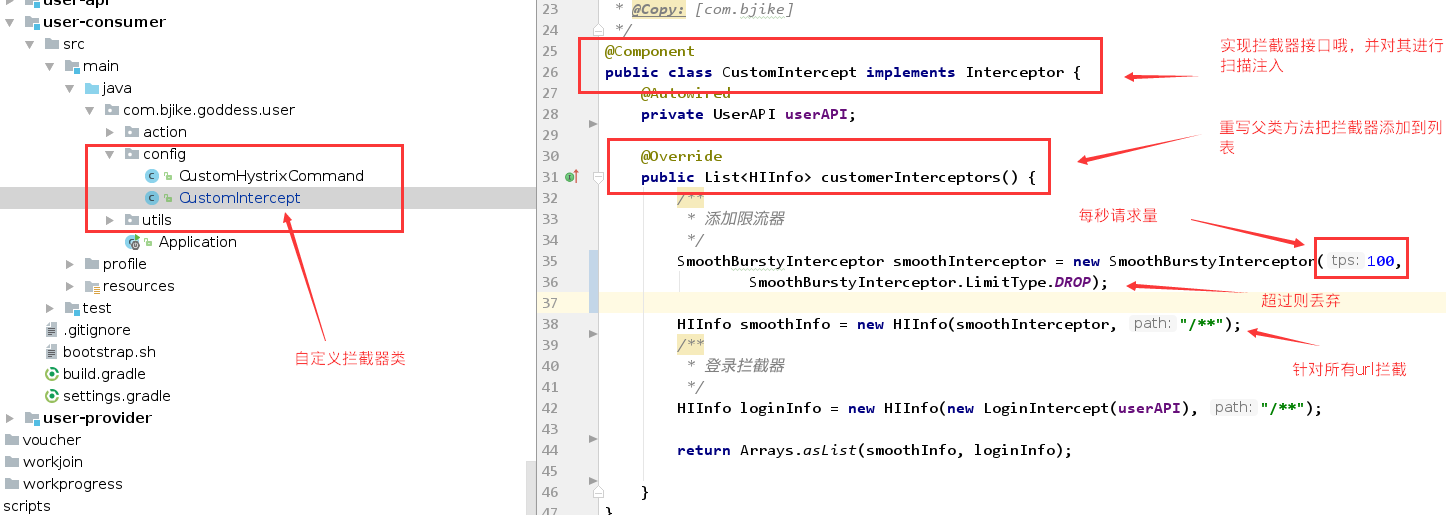


**SmoothBurstyInterceptor（限流器）**

该拦截器默认不开启，由子类控制开启，针对恶意的频繁访问进行限流，以保证程序的稳定性，限流器的实现有两种方式（请求等待，请求丢弃，默认为丢弃），默认限流量为每秒10次，超出限量抛出“网络错误”异常。见下图：



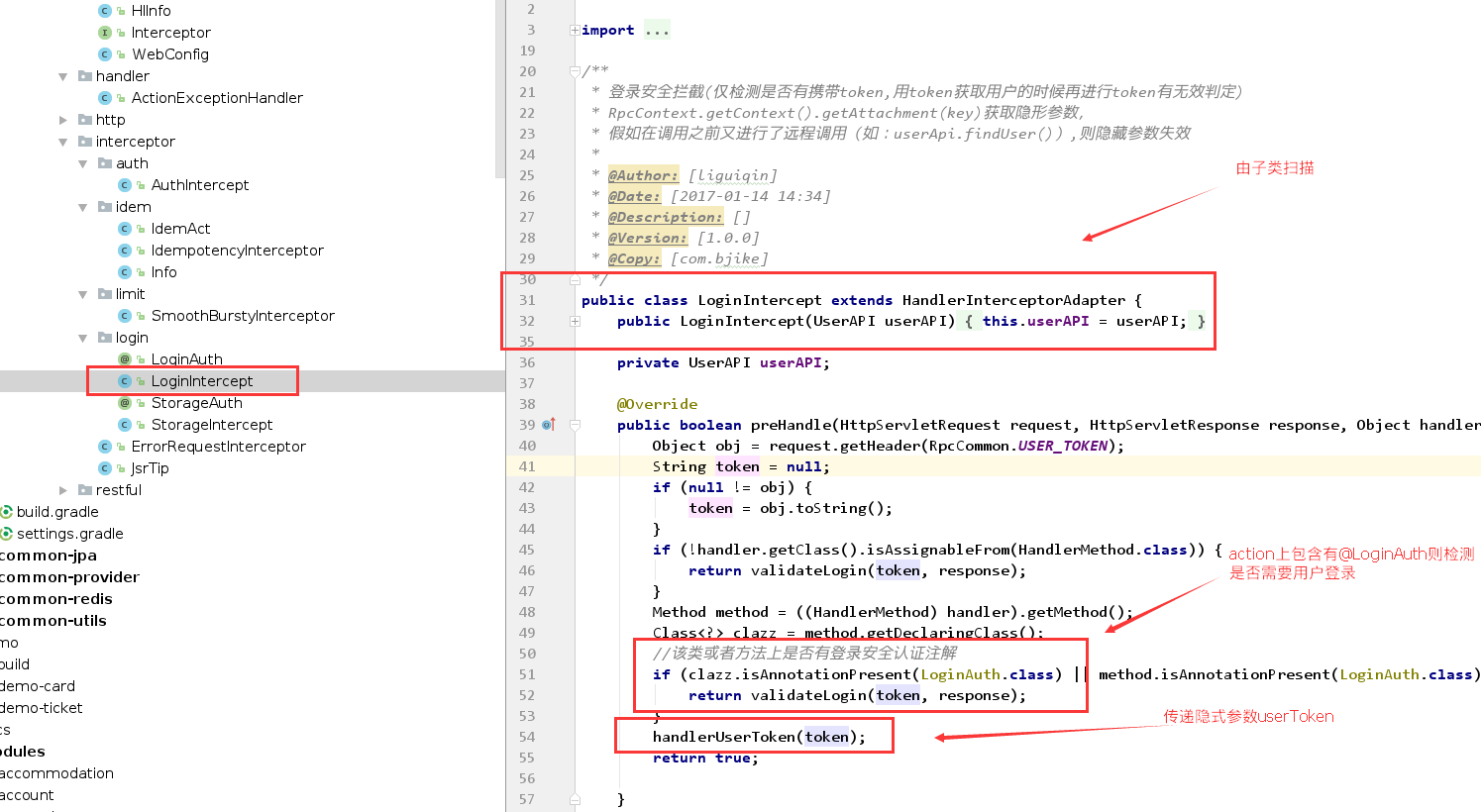
**如何开启：**

在consumer添加拦截器来对先流器的继承并扫描即可，以user-consumer为例，见下图：

**其他拦截器的用法跟上图的用法是一样的，同样是添加到 List<Info>就可以了，**

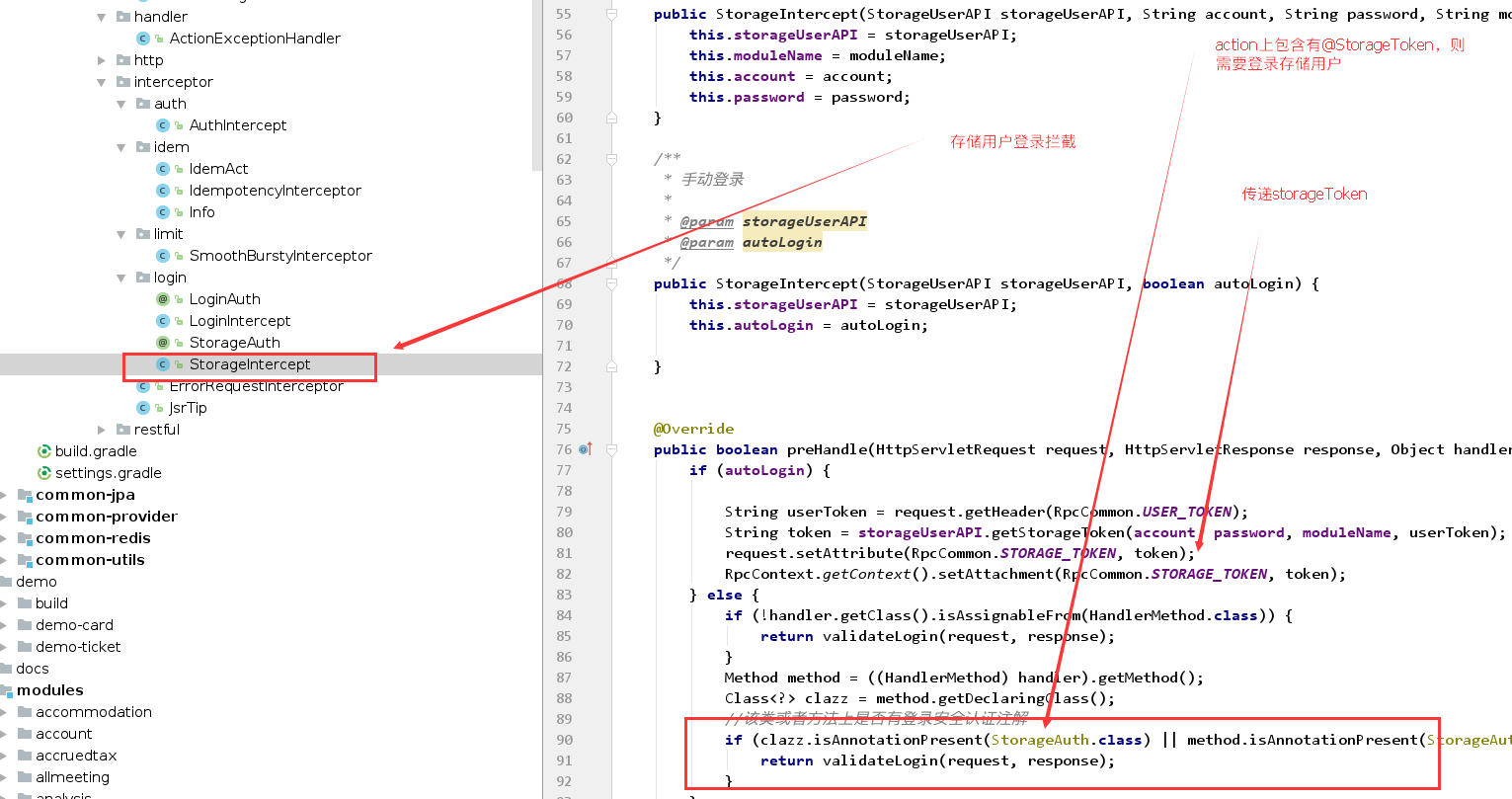
**比如上图就添加了两个拦截器。所下面的拦截器使用方法就不多做解释了，只对其作用做简单的讲解**

**LoginIntercept（用户登录拦截）：**

****

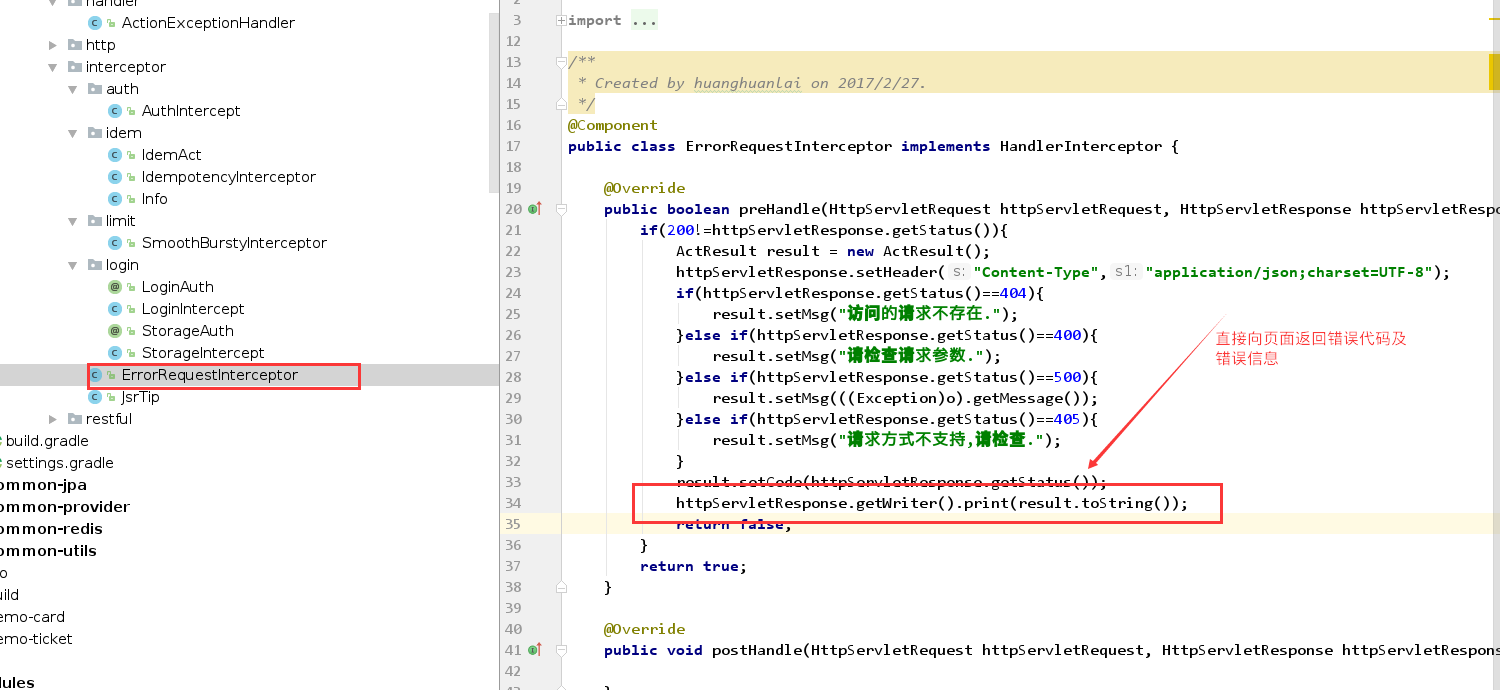
**StorageIntercept（存储用户登录）**

该拦截器默认不开启，由子类控制开启，其用法有点一样（具体使用会在下面介绍上传使用时详细介绍），可配置自动登录用户或者手动登录，拦截器如下：

****

**ErrorRequestInterceptor（错误请求拦截）：**

该拦截器默认开启，对请求方式的错误，参数错误，请求url错误，业务错误进行拦截



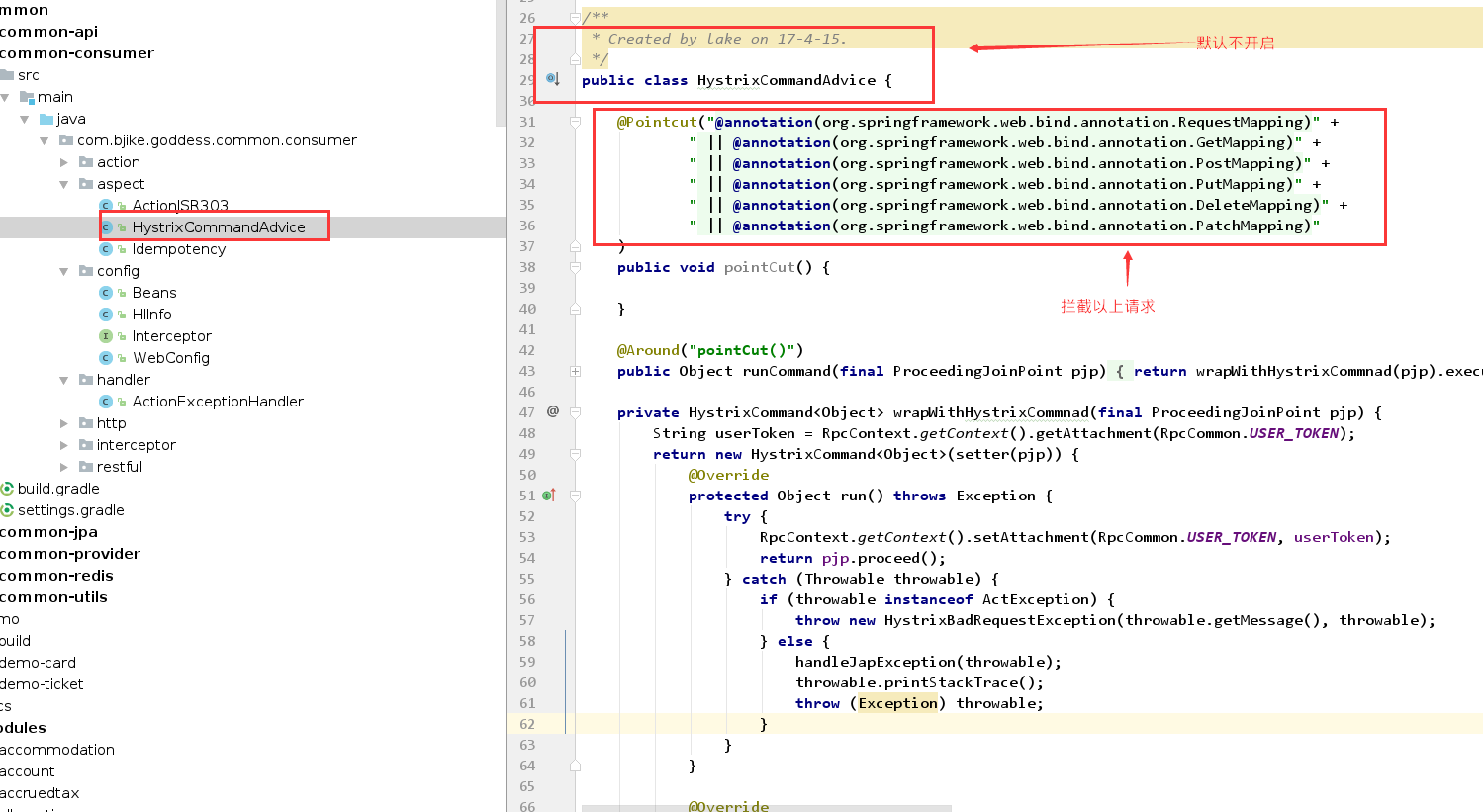
**2.AOP（切面环绕处理）**

**ActionJSR303：**该AOP默认开启，对action的入参进行数据校验，及幂等操作校验（是否重复提交）等，见下图



**HystrixCommandAdvice（断路器）：**

该AOP默认不开启，由子类实现扫描。其作用是增强系统稳定性，尽量减少故障对性能的影响.当客户端请求某方法（post，get，delete，patch），但程序多次错误而没有运行正常时阻断请求（拦截错误不直接返回页面，经由ActionExceptionHandler返回）

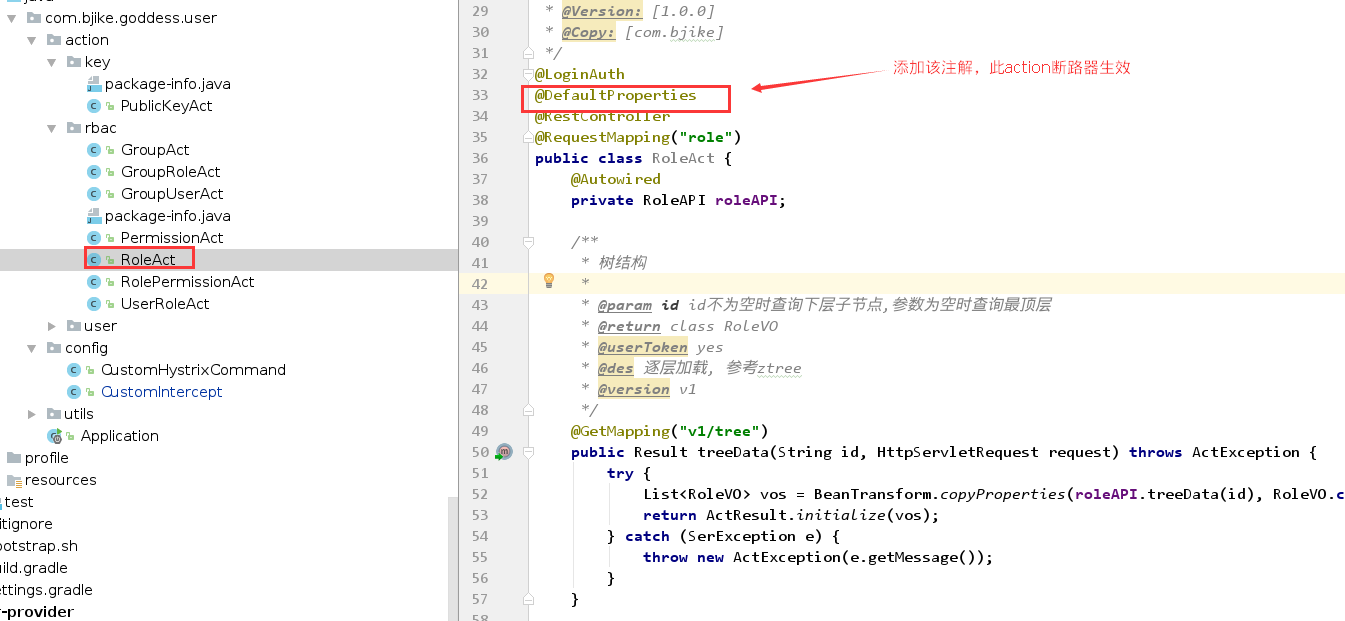


其用法也简单，见下图（user-consumer为例）：

1.添加扫描配置



1. action类添加@DefaultProperties注解：



1. **处理器**

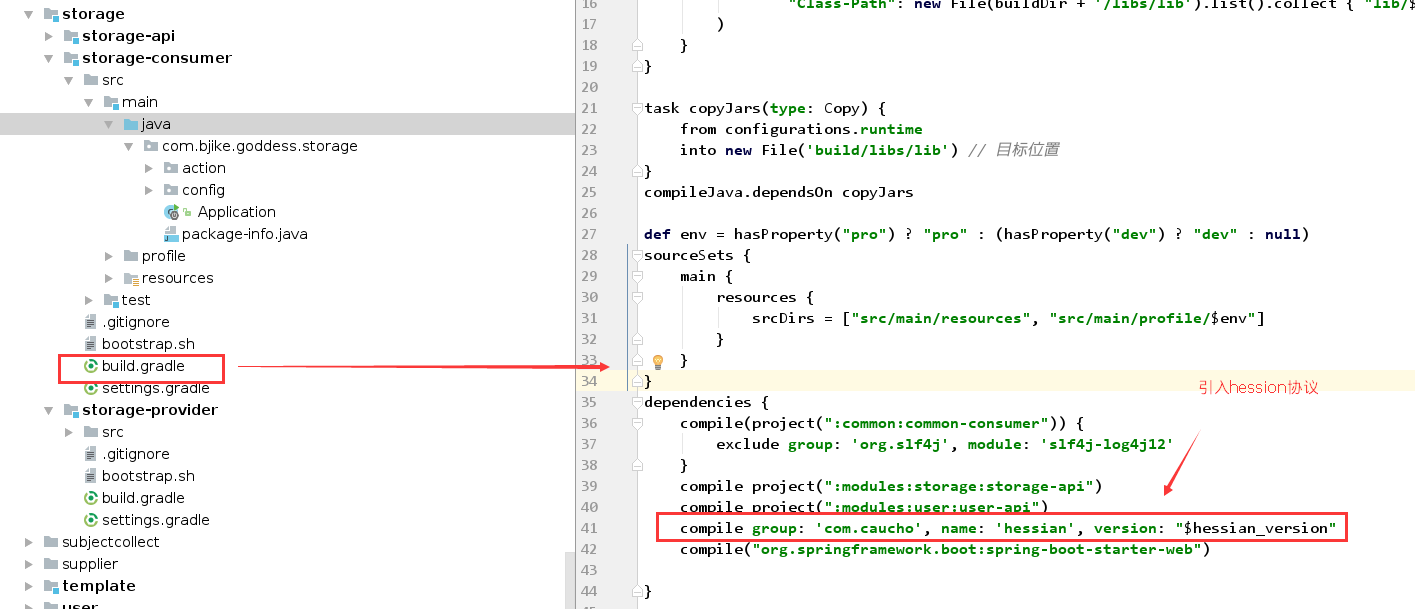
**ActionExceptionHandler（异常处理器）：**

该异常处理器默认开启，对登录用户的错误代码进行识别及处理，数据库的错误进行简单的处理，大部分错误都会由经此处理器再返回页面，不再详述。

1. **Storage（文件上传）**

文件上传比较特殊，consumer使用的时hessian服务协议，而不再时dubbo，因为hessian在上传大文件上有绝对的优势，文件上传的开发主要在consumer，而不在provider，一般如果没有什么特殊业务处理，直接调用storage-consumer里提供的api方法即可，如果要处理自己的逻辑业务，那么须引入storage-api，然后见下图步骤，(亦可参照storage-consumer的程序代码)

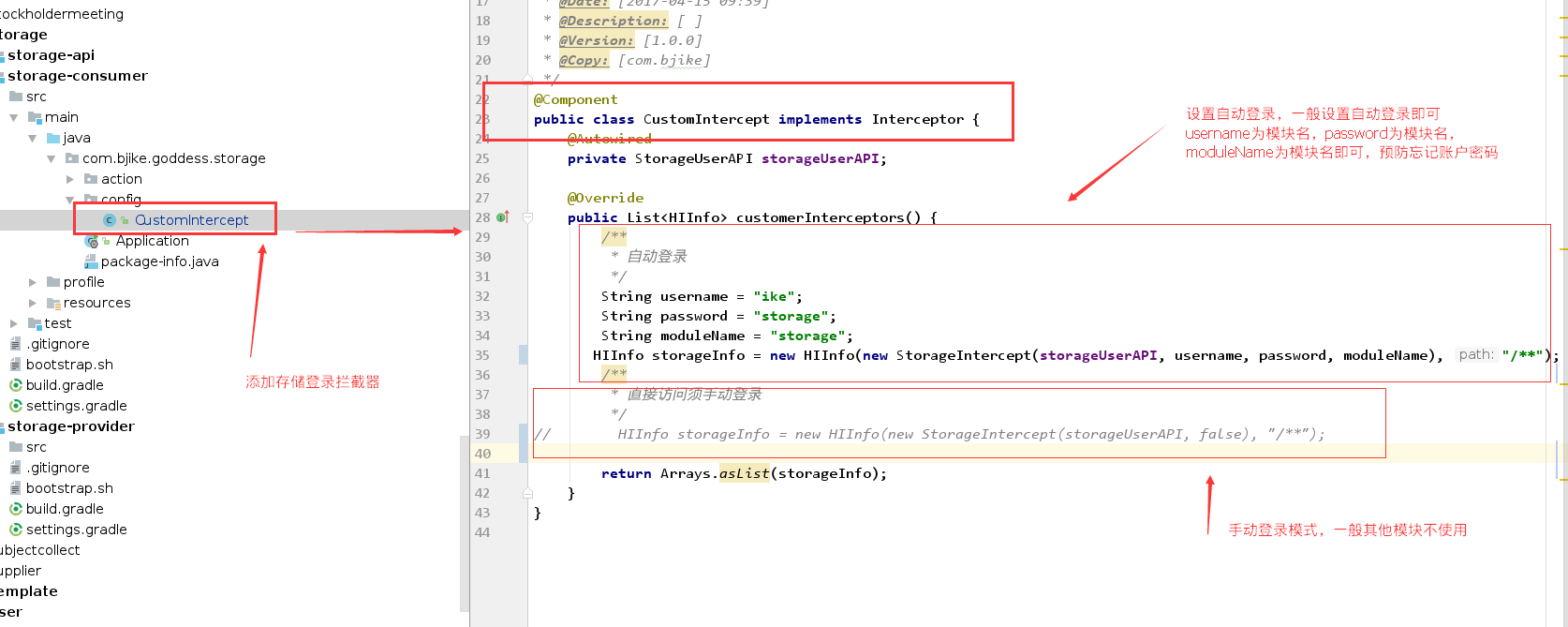
**1.引入hession协议**



1. **设置文件上传大小限制**

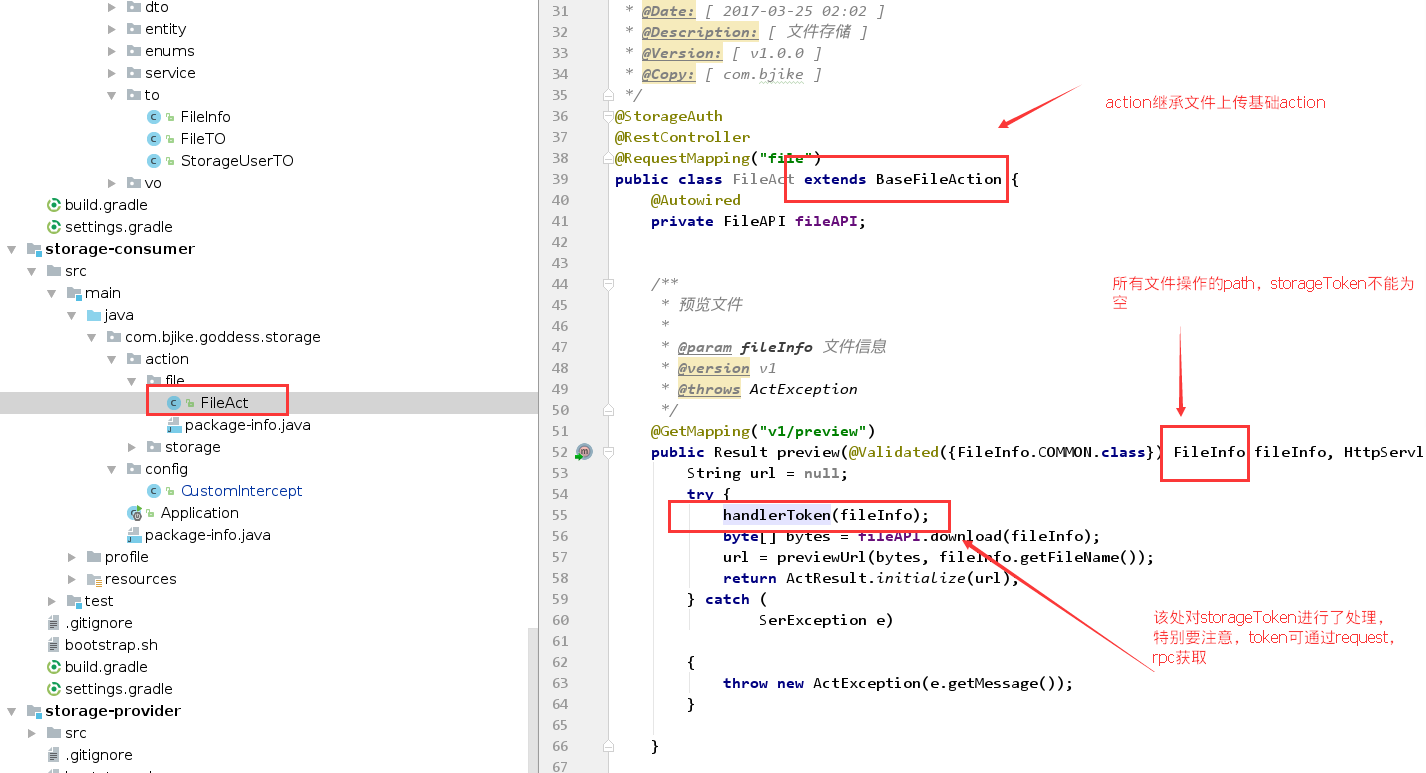
****

1. **添加存储登录拦截器，及设置自动登录帐号密码**



1. **action的编写**

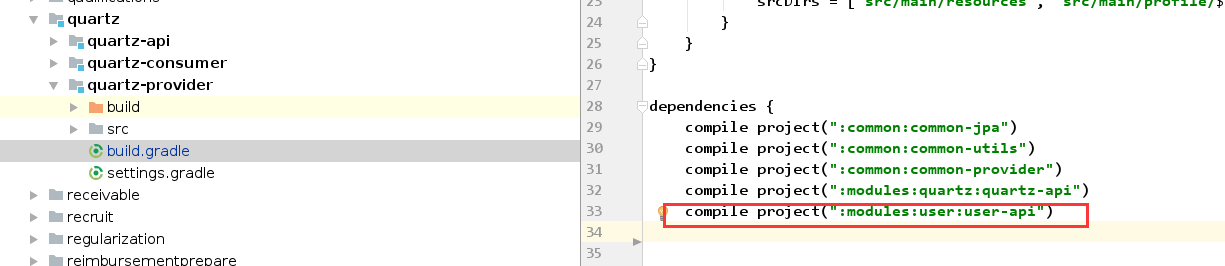
**注意storgaeToken在params传递，而不时header**

****

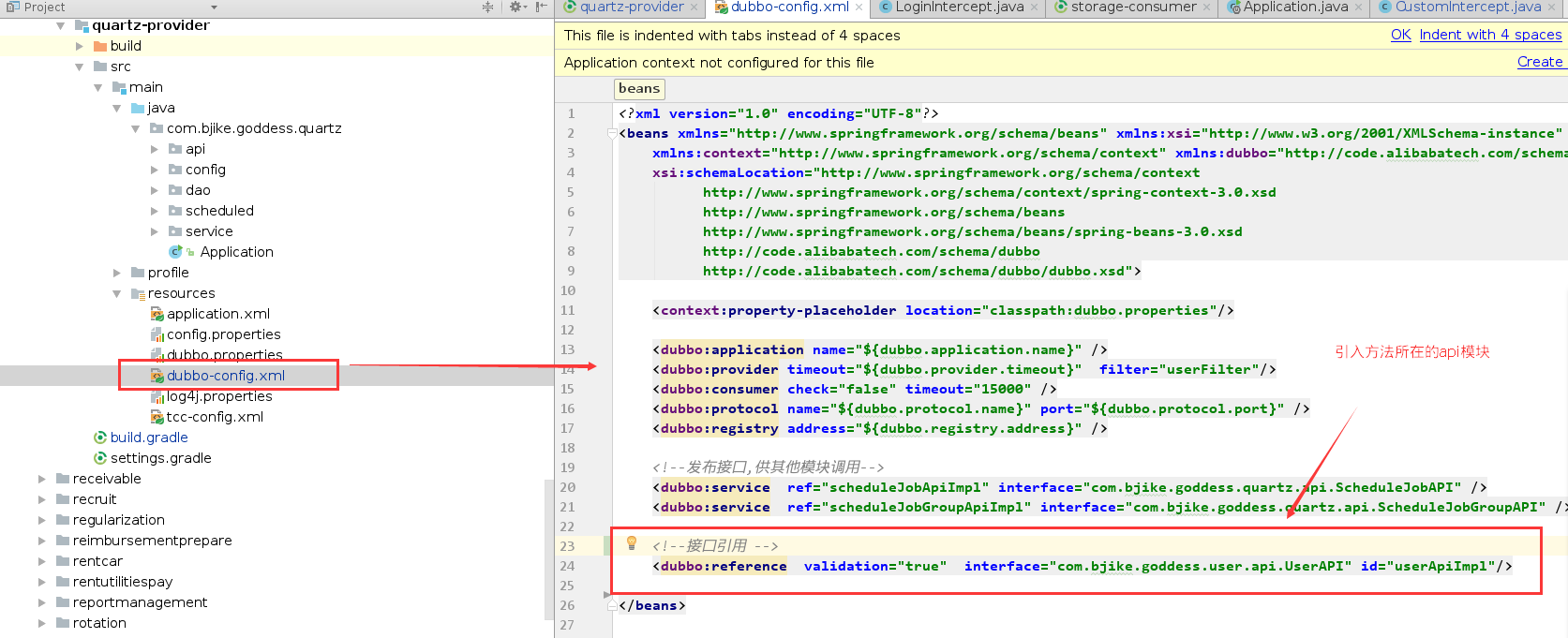
1. **Quartz（定时器）**

定时器的用法相对比较简单，但是由于依赖的模块比较多，可能打包会比较慢，以下定时器添加以user-provider的publicKey定时方法为例：

**1.导入依赖的api：**



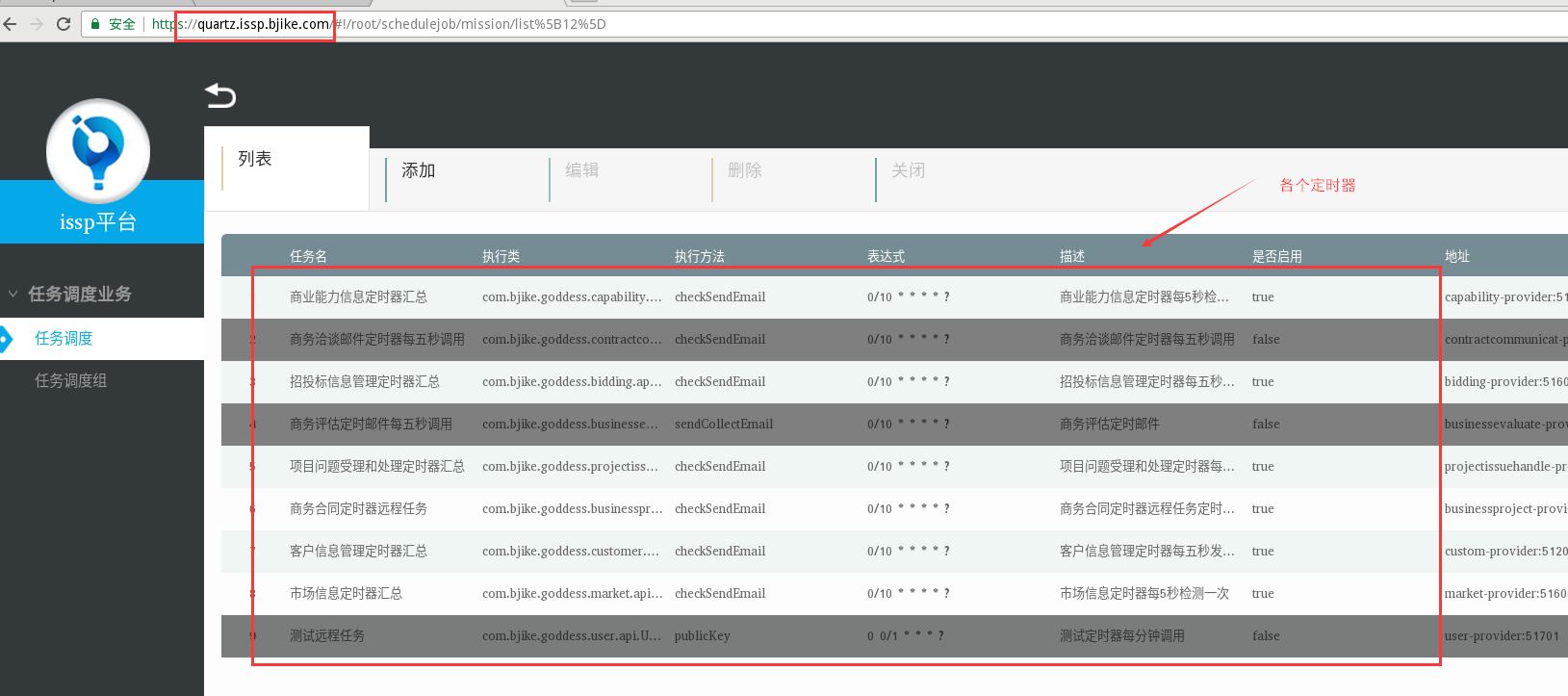
**2.Api方法接口引用：**

****

添加完成后就可以打包，并且进行部署了，部署完成进入定时器添加的页面：

地址为：<https://quartz.issp.bjike.com>

见下图：

****

**3.添加定时器组及任务**

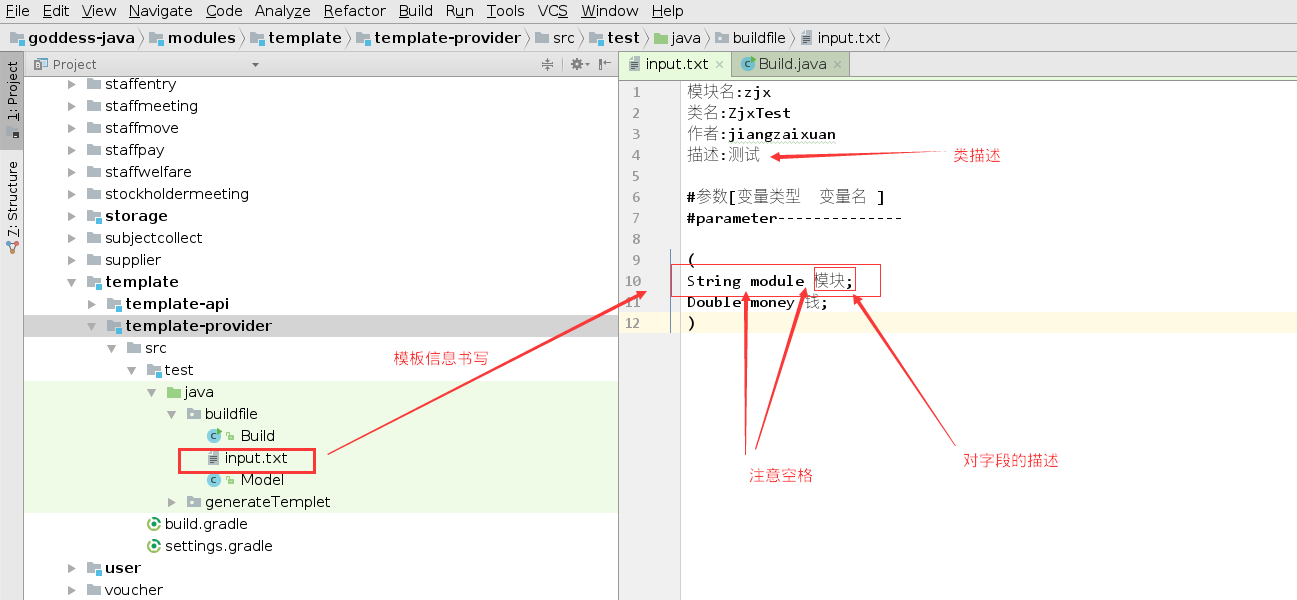
定时器组的组名为你的模块名，添加完组，然后再添加你的任务：



1. **Template模板**

通用的模板生成类，生成模块的基础信息，类信息等，只要填写的信息格式正确即可正确生成，生成后注意类的代码缩进一定要自己手动ctrl+alt+L整理好，同时注意entity类的关系映射，因为自动生成的不一定对，（也可参考goddess开发流程）

1.填写模板信息



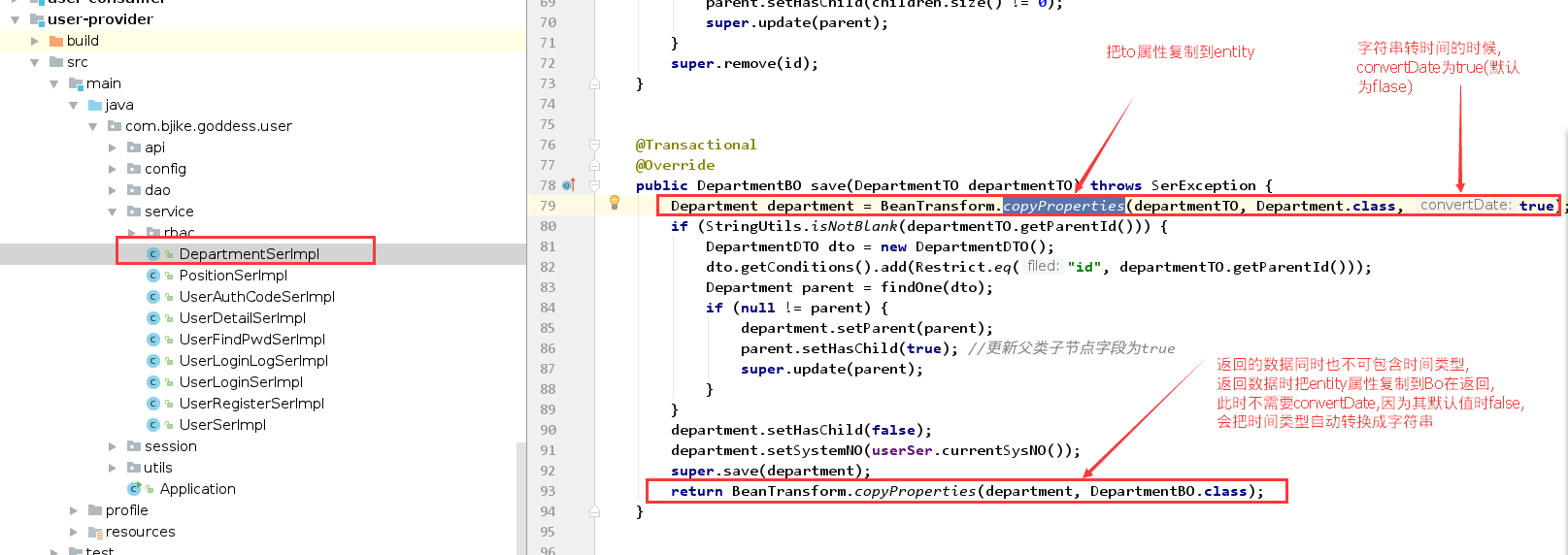
2.填写完以上信息后，执行下图方法：



1. **BeanTransform（类复制工具）**

**该类主要的作用**

1. 对类的属性进行复制,及对时间的转换,避免在远程调用传递过程当中,日期无法被序列的情况,这种情况主要体现在action->api->service层的传递,所以必须以to传递到service层,to不包含localdate,localtime,localdatetime类型的属性,而在使用beanTransfrom.copyProperties()方法对其转换,如下图:



1. 在action层对属性传递的过滤,需要什么数据,什么字段由前端控制,减少网络传输数据量,使用方法为传入request即可,见下图



前端传入过滤或者导入参数为**\_excludes,\_includes**数组