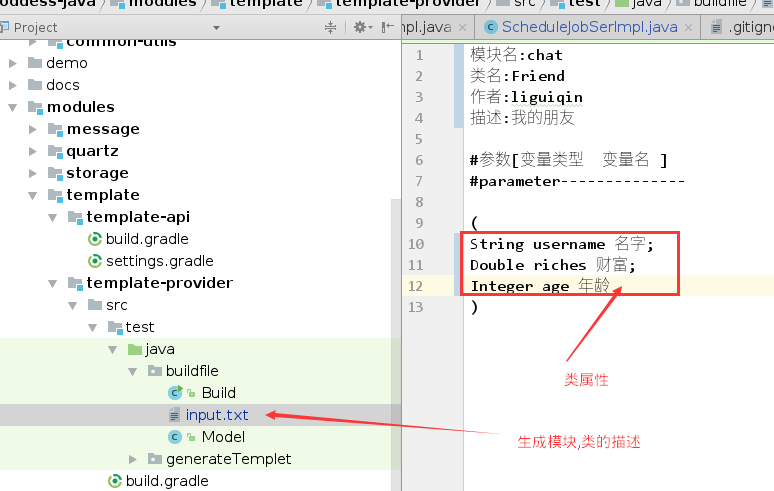
# 开发流程

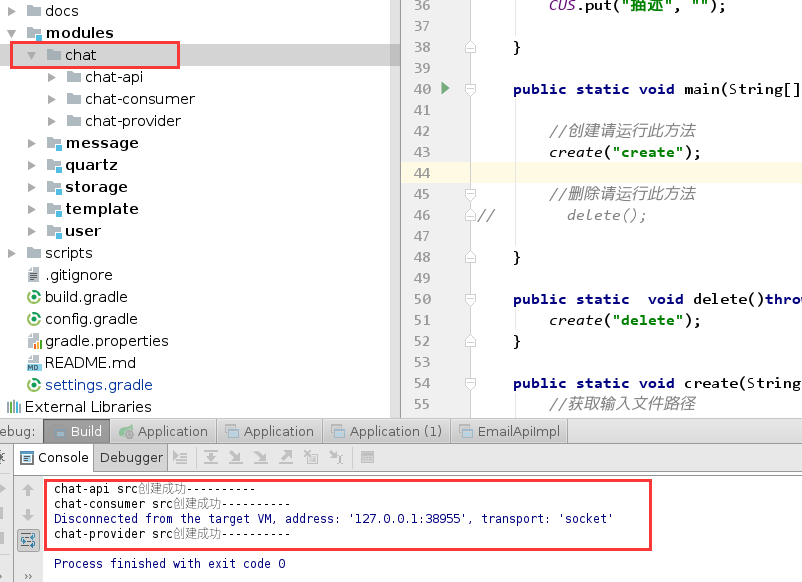
注:项目的大部分配置,直接copy其他模块的配置文件即可,配置文件的内容稍有改动,主要是服务端口及模块名的修改.项目的实体及其他包的创建生成,均使用template模块的工具类进行生成,完成一个新的简单的模块的功能开发步骤如下:

**1.填写创建模块及类的信息**

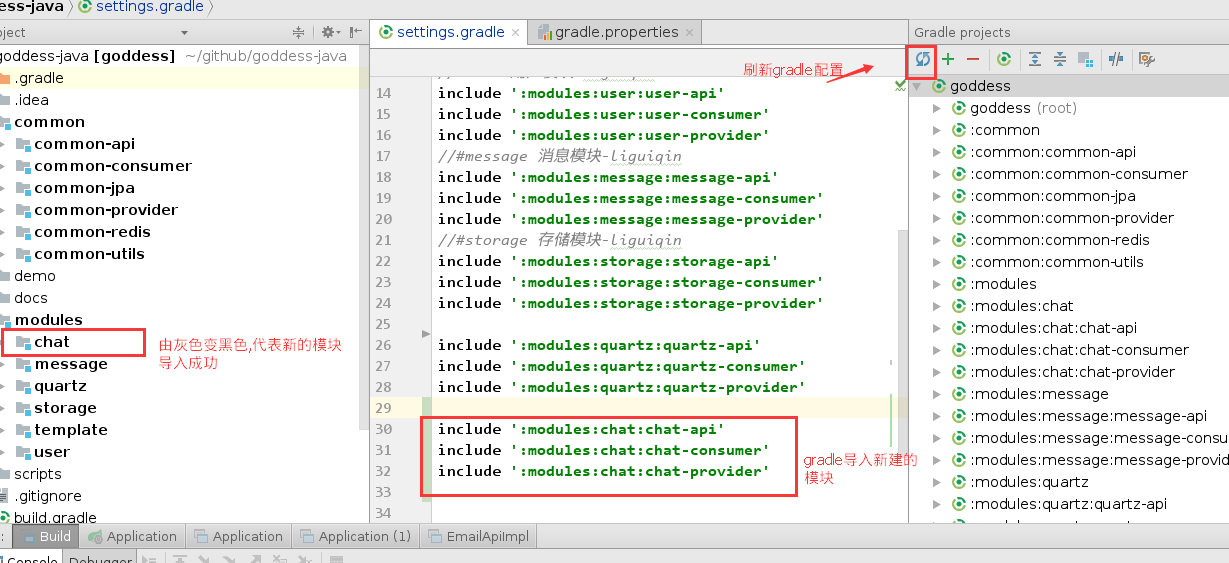




**2.创建成功,见下图:**



**3.导入模块项目:**



**4.打开api**

注:虽然类信息已经创建成功,但代码的格式是没有缩进的,需要手动的缩进

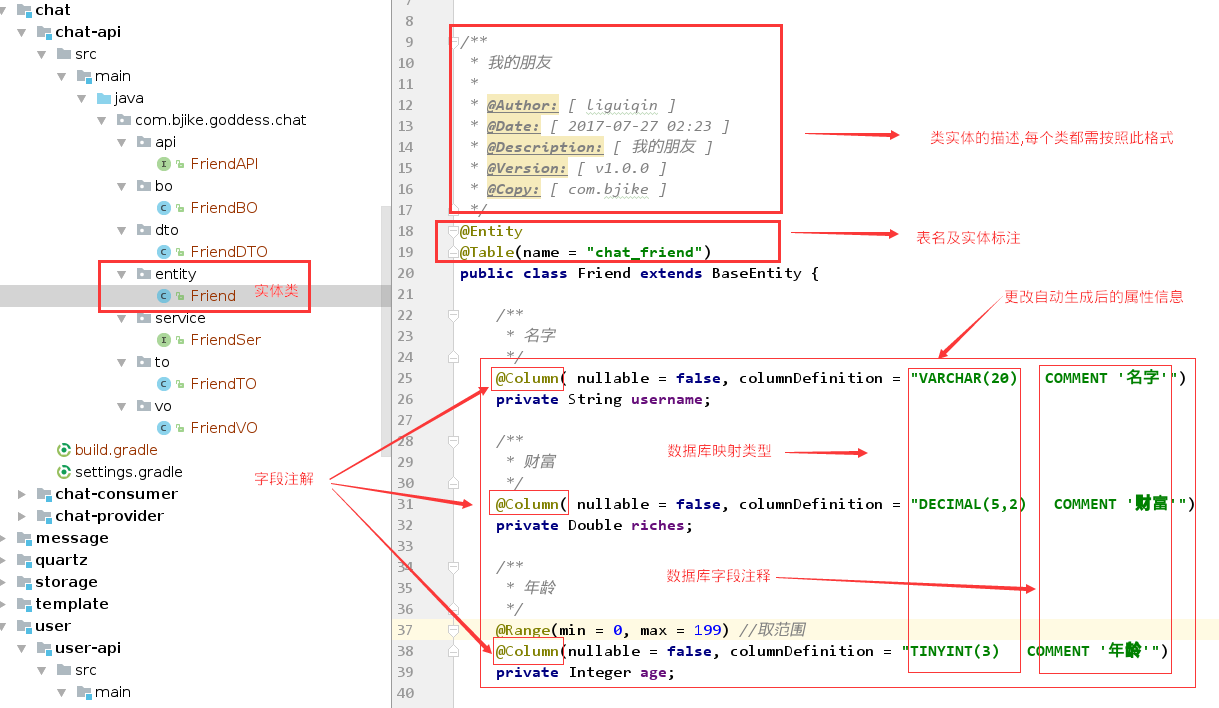
打开类,同时按住ctrl+alt+l即可.其次,实体信息生成的映射关系并不完全正确,

需要检查及更正.字段信息的映射,根据字段类型来配置,具体参照docs/EntityColumn.java,

实体默认字段有id,createTime ,modifyTime



先看生成的实体类,并进行修改,如下图.其他类api,bo,dto,ser,to,vo则对其进行格式的整理即可,bo,vo,to则根据实际业务情况,可对字段进行删减



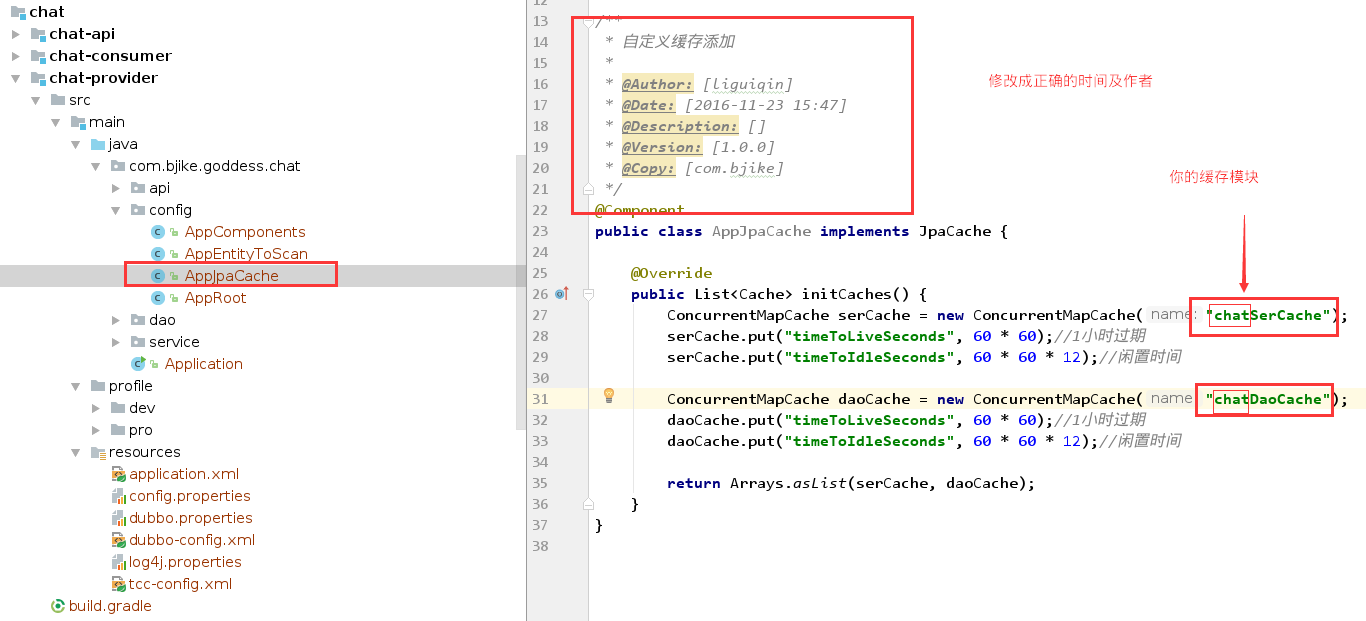
1. **provider的配置文件及修改**

**找到任意一个正常的模块(此处我用的是quartz的模块),对其配置文件进行复制**

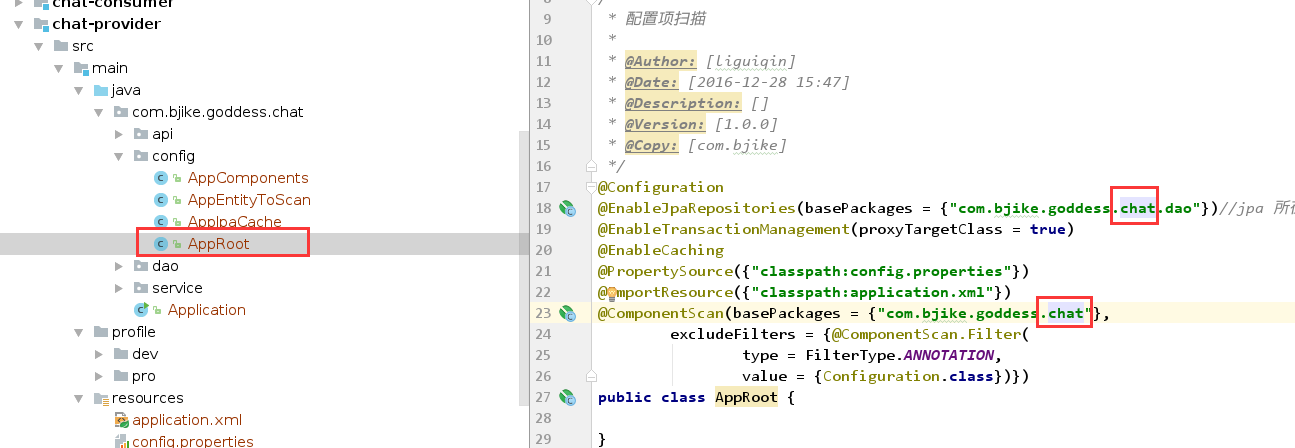
**如下图:**



**AppJpaCache更改缓存配置:**



**AppRoot:更改扫描包类:**



**修改配置文件:**

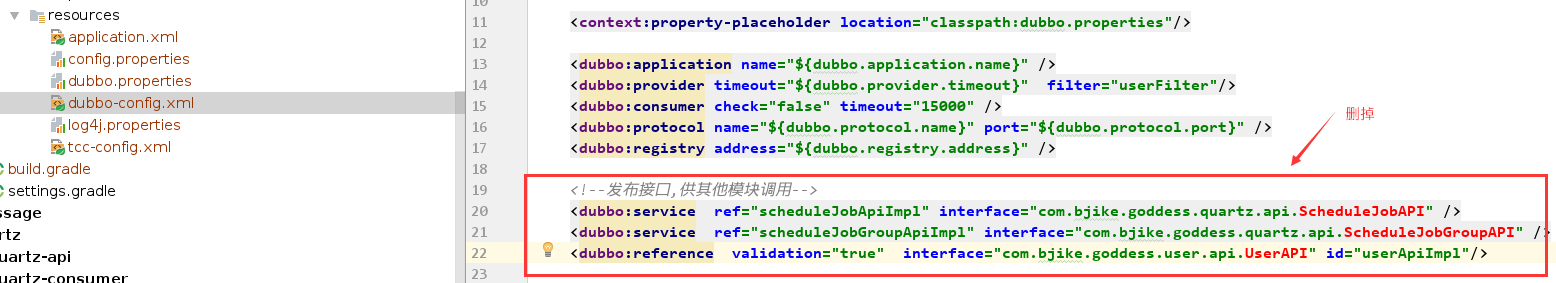
**Resource:**

Dubbo.properties:

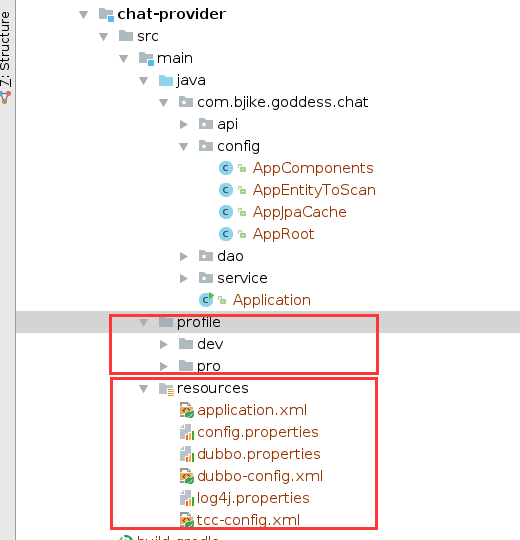
**修改dubbo.application.name为自己的模块名**

**dubbo.protocol.port 指定的端口范围,(问负责人要)**

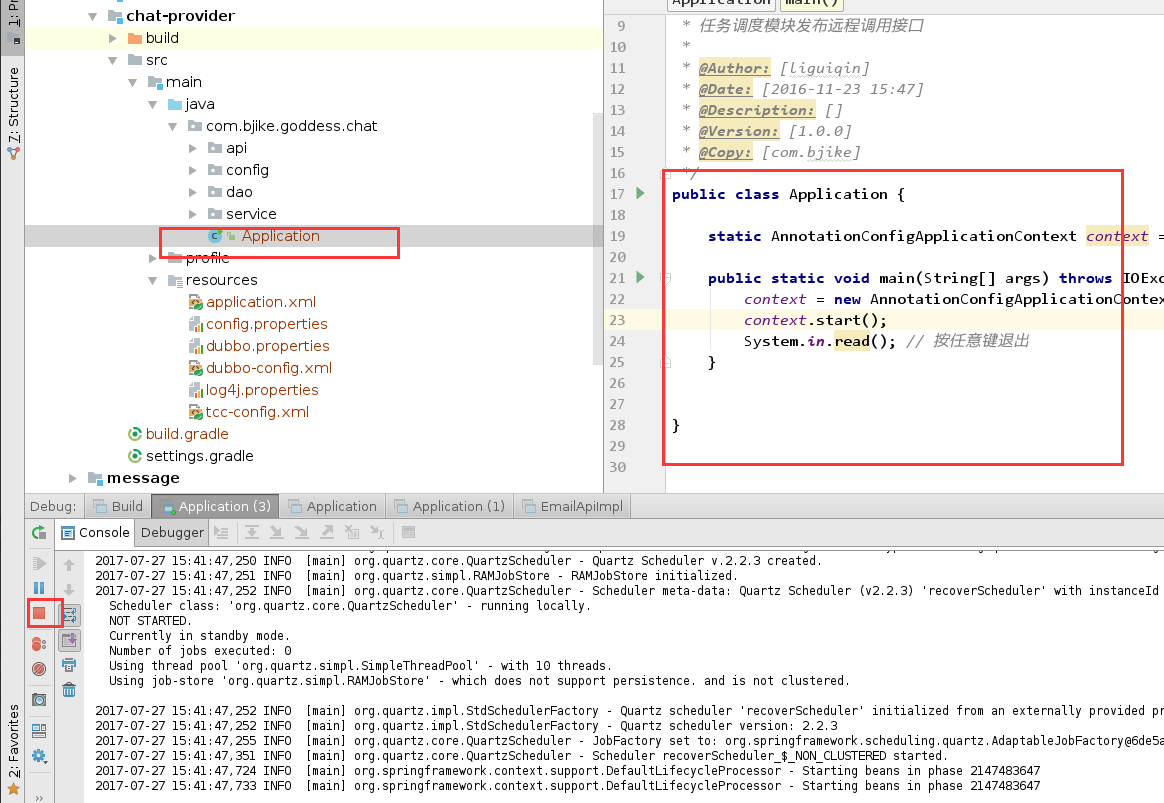
**dubbo-config.xml 删除掉别人的发布接口**



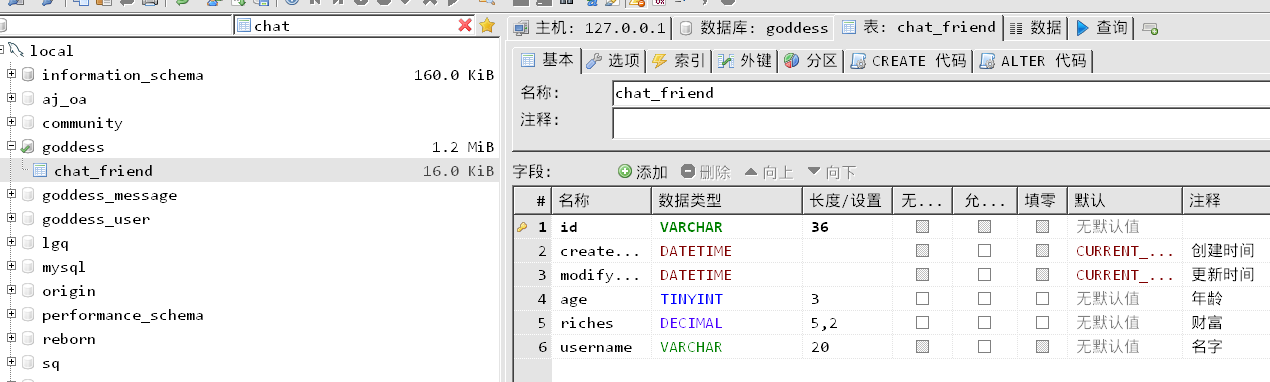
**Profile下的dev,pro为部署正在环境时候的配置,跟resource基本一致,开发完resource,再从resource复制依次替换掉dev,pro的文件即可(不包含dubbo-config.xml文件).然后再修改dev,pro里的地址及端口.包括config.properties里的mysql链接,dubbo.properties的zookeeper地址,tcc-config.xml的redis地址,具体地址可参照其他模块的配置**



运行成功见下图

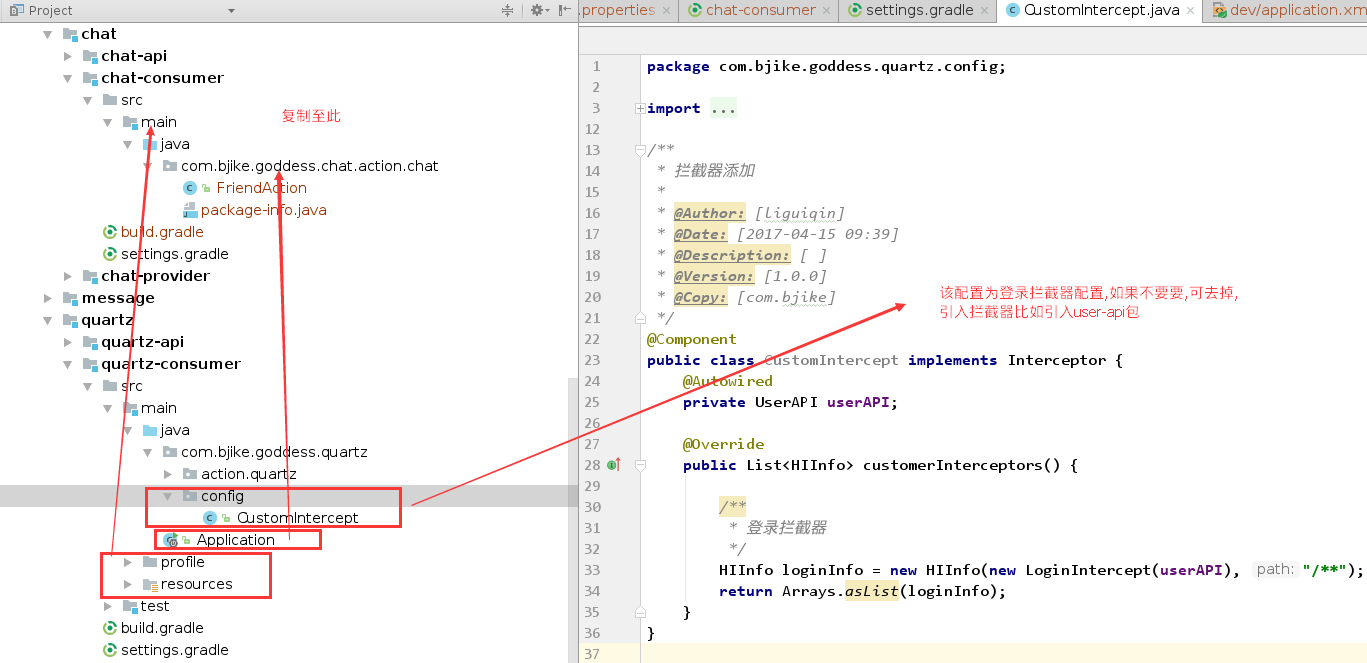


查看数据库,创表成功:

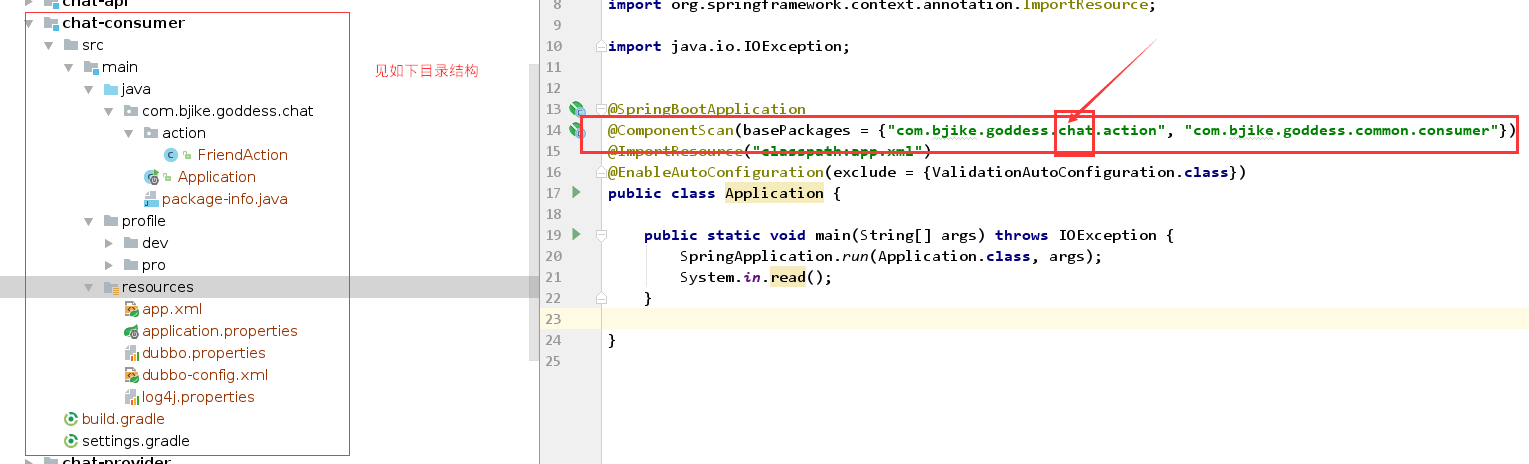


**6.consumer的配置文件及修改**

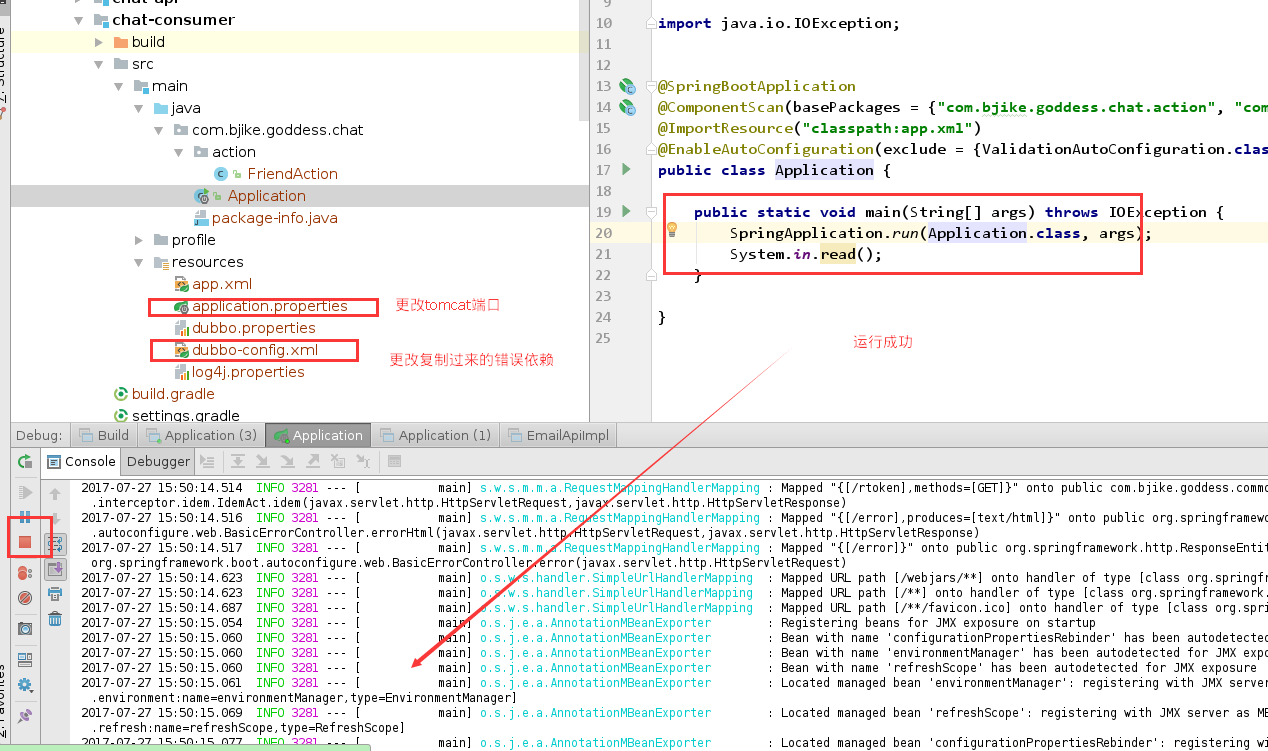
**复制配置文件并修改**



**然后见下图目录:**

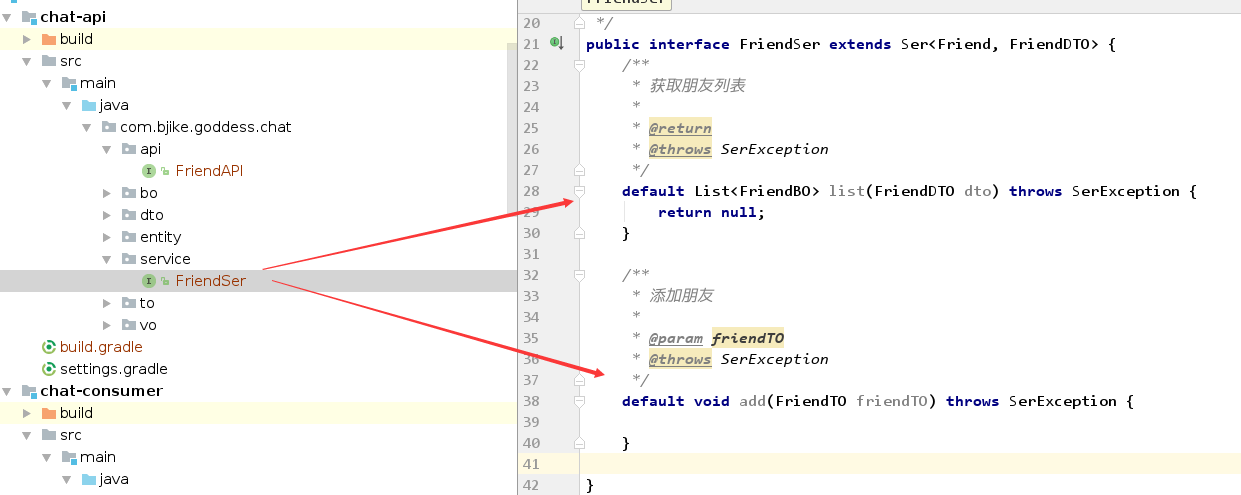


**运行成功见下图:**

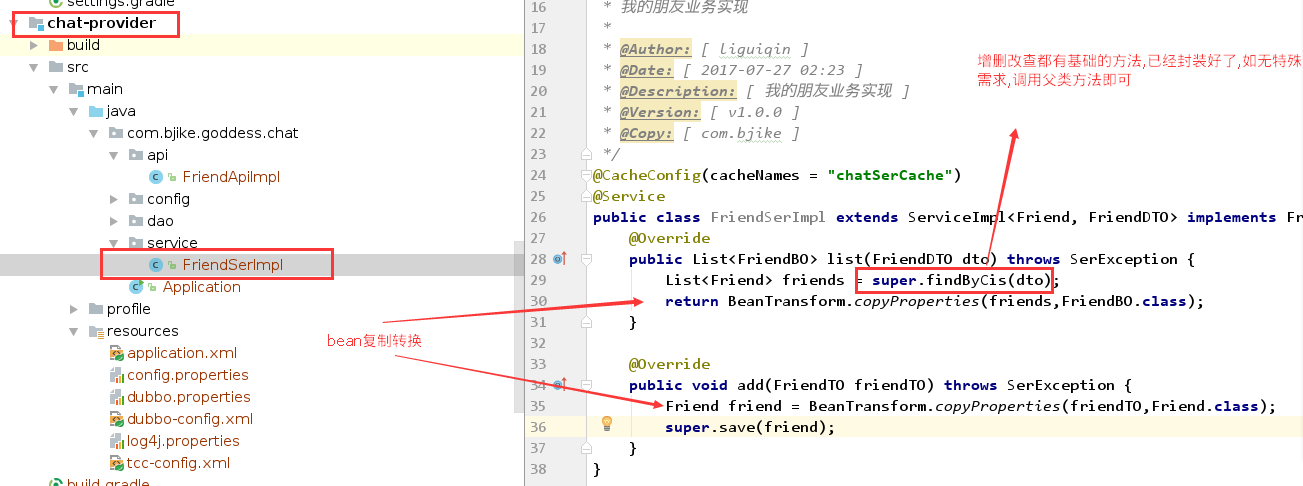


1. **开始业务编写**

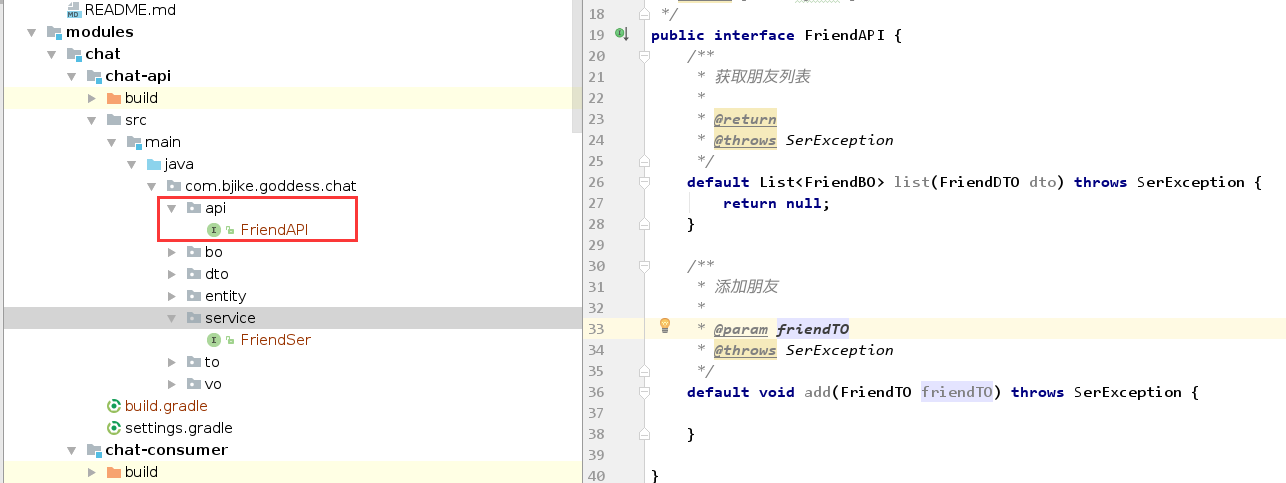
**Service层业务接口编写**



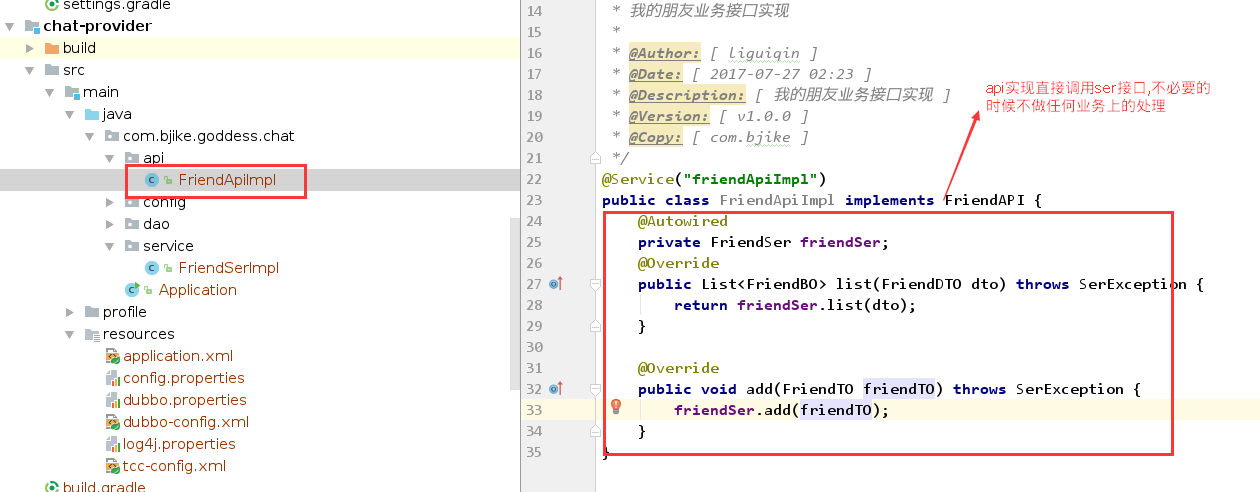
**Service实现层编写**



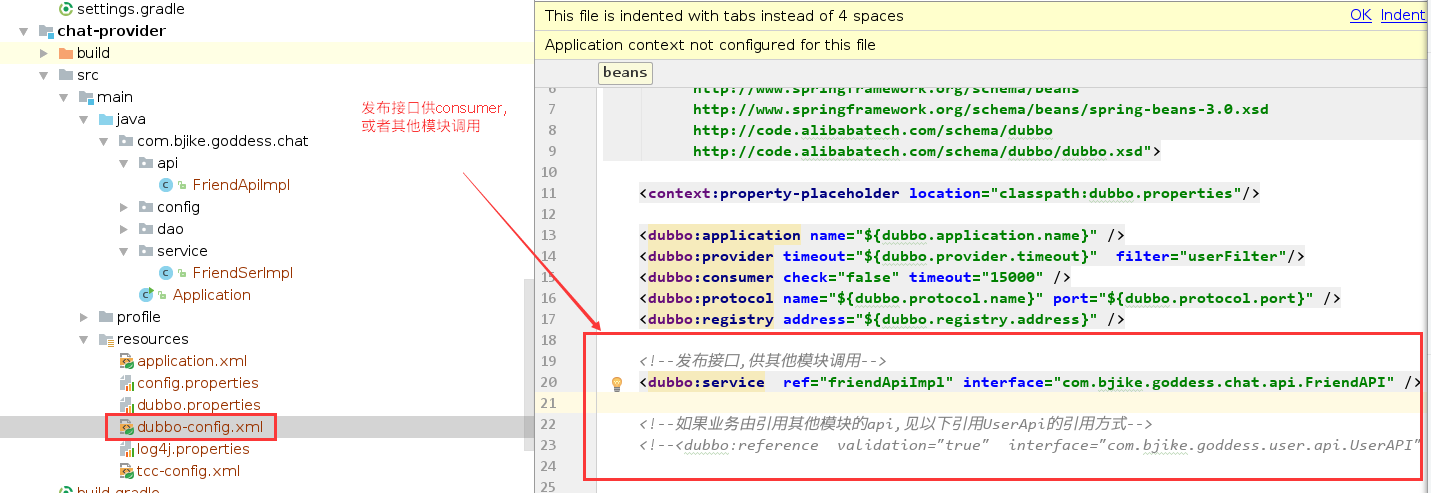
**对外发布api接口编写**



**对外发布api接口实现**



**provider发布接口:**

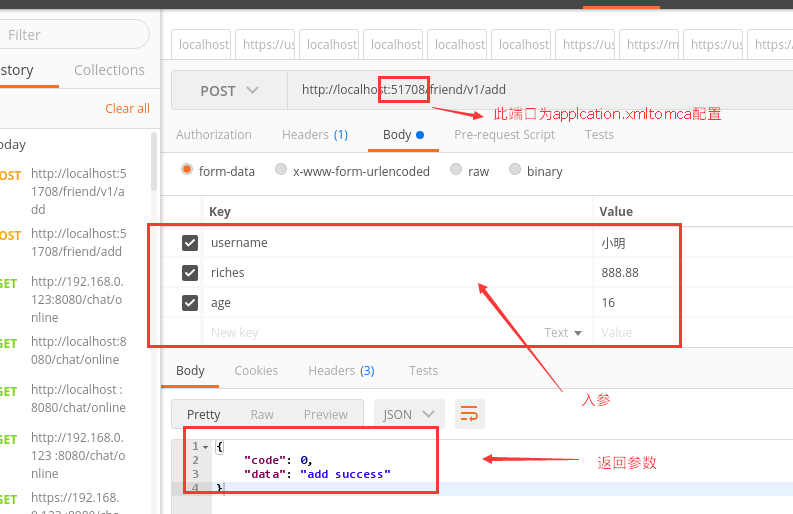


**consumer引入服务:**

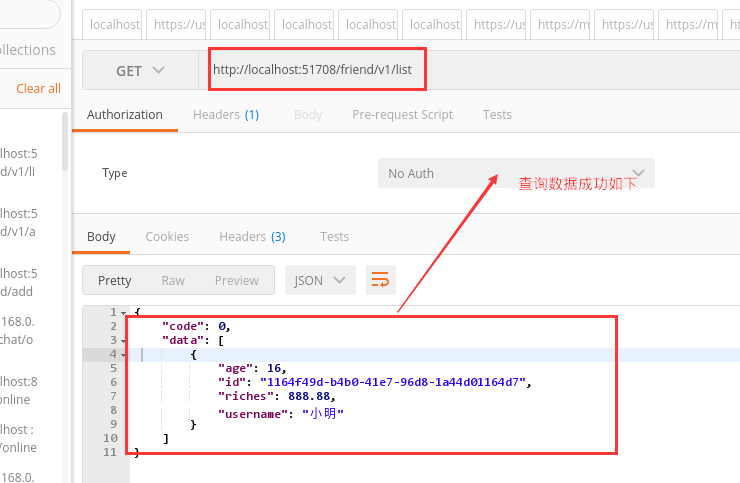


1. 重启**provider,consumer用postman进行测试:**

**添加数据成功如下**



**查询成功:**



**关于拦截器,文件上传,限流器,断路器的使用,或其他规范请参见项目里的README.md,**

**和其他已经完成的模块,**