# 一、软件安装

**1.jdk**—java运行环境（请使用jdk1.8，否则后面会报很多错误，不好解决）

**2.IntelliJ IDEA**—java IDE集成开发软件

**3.lombok**-集成在intellij IDEA中（preference-plugin-browrepositories-搜索lombok-安装）

注：安装完后记得配置complier-annotation complier选择

**4.Docker**-如果是linux或mac建议按照（window比较费事）

注：如果安装Docker就可以新建mysql和redis容器本地跑起来

**5.Git**—进行版本推送（git指令见Study_Note）

**6.Postman**—方便的模拟get或者post或者其他方式的请求来调试接口（辅助用来获取token等）

# 二、服务功能测试

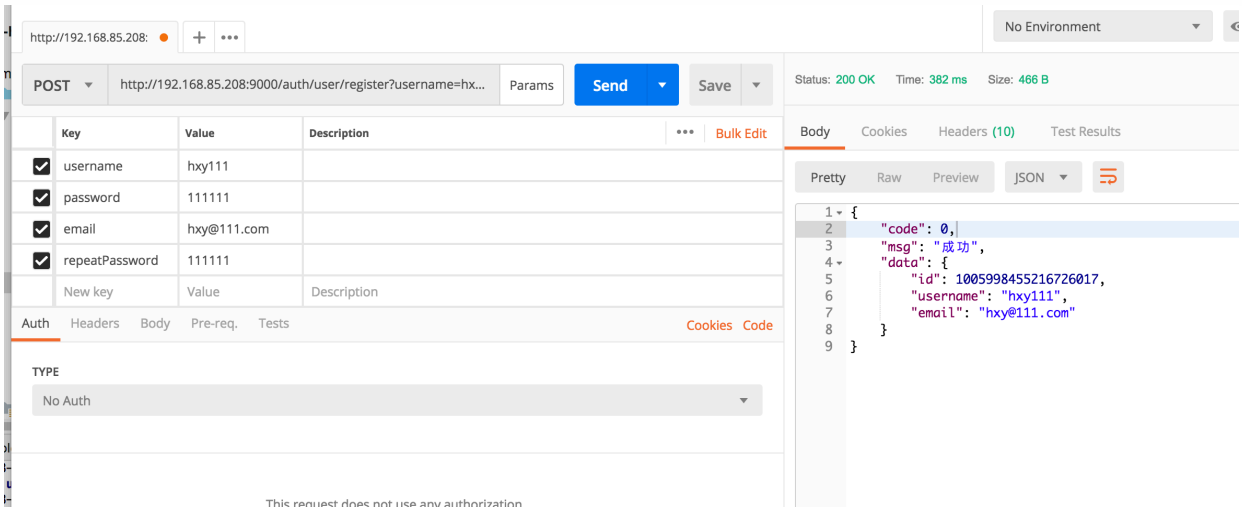
http://zuul网关启动的服务器的ip地址和端口，通过网关统一对后台微服务进行访问

## 1、注册用户（只有新用户不用token）

auth/user/register

Params:username、password、email、repeatPassword

注意：用户名 密码最小六位最大20位

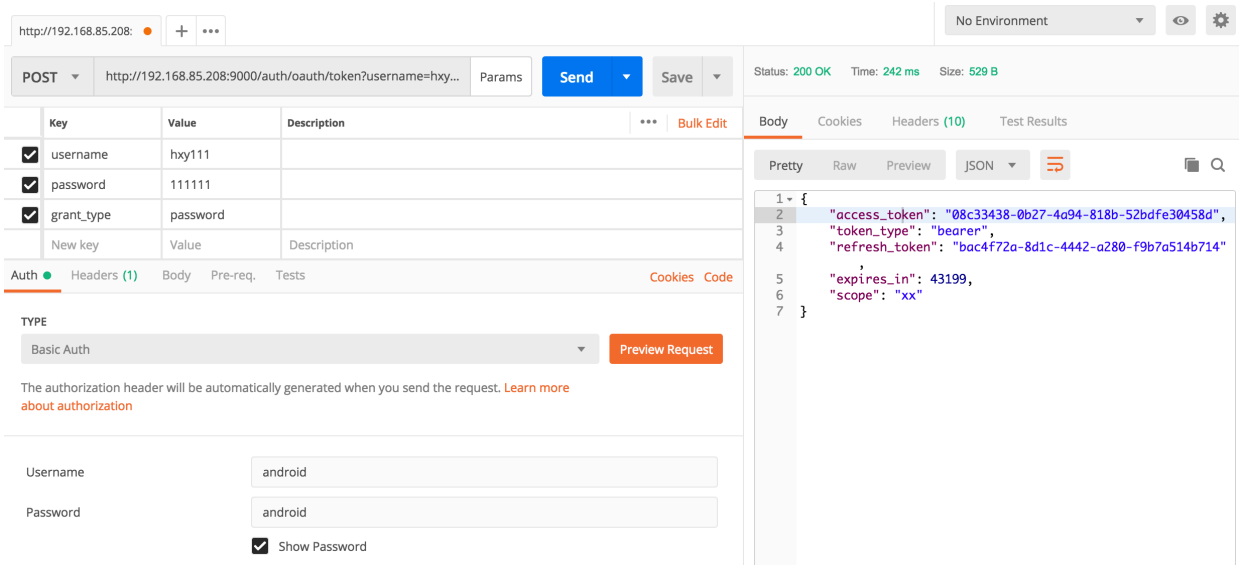



# 2、获取新用户token

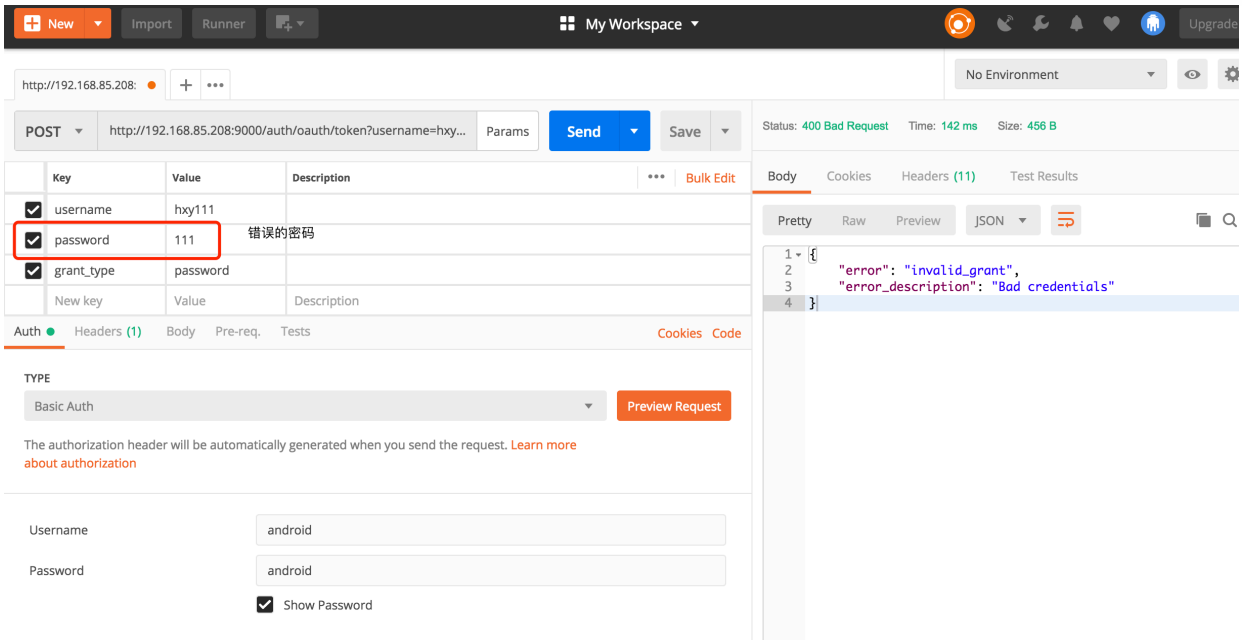
auth/oauth/token

params：username、password、grant_type（目前写死为password）

用户名密码必须存在在数据库中（即当前用户已经注册成功，且用户名密码匹配）



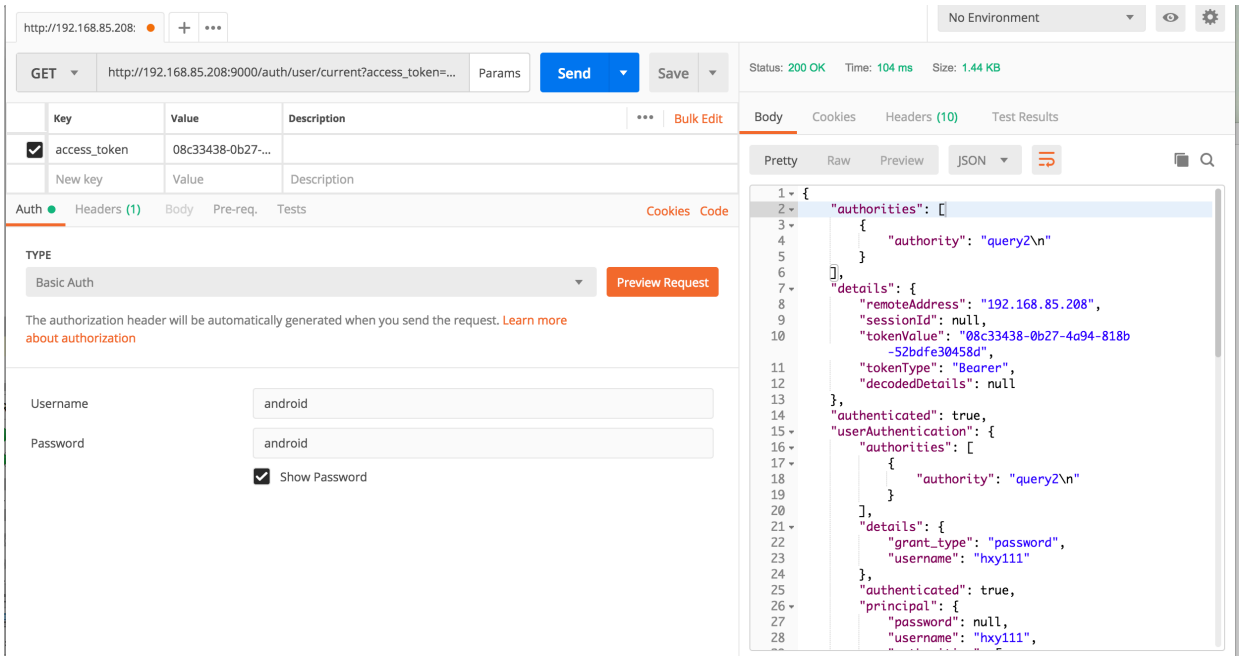

用户名密码不匹配情况:



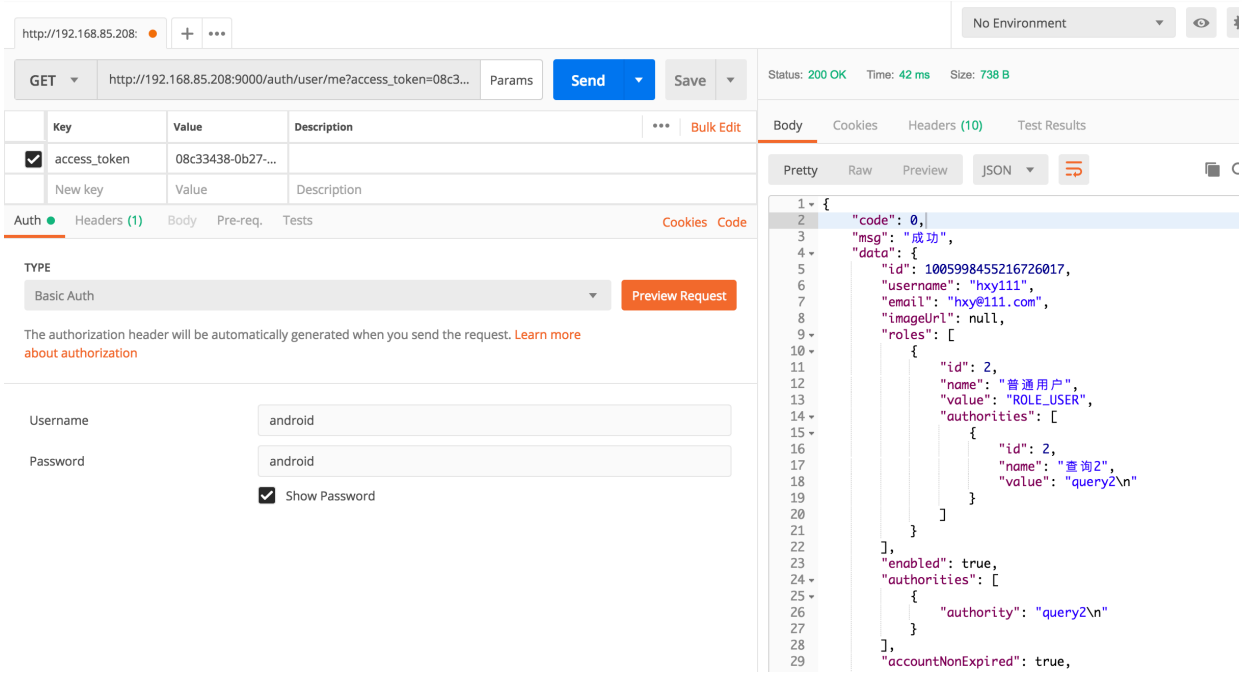

# 3、获取用户具体信息（例如权限等）

/auth/user/current—这是个get方法，不允许给服务器发送数据

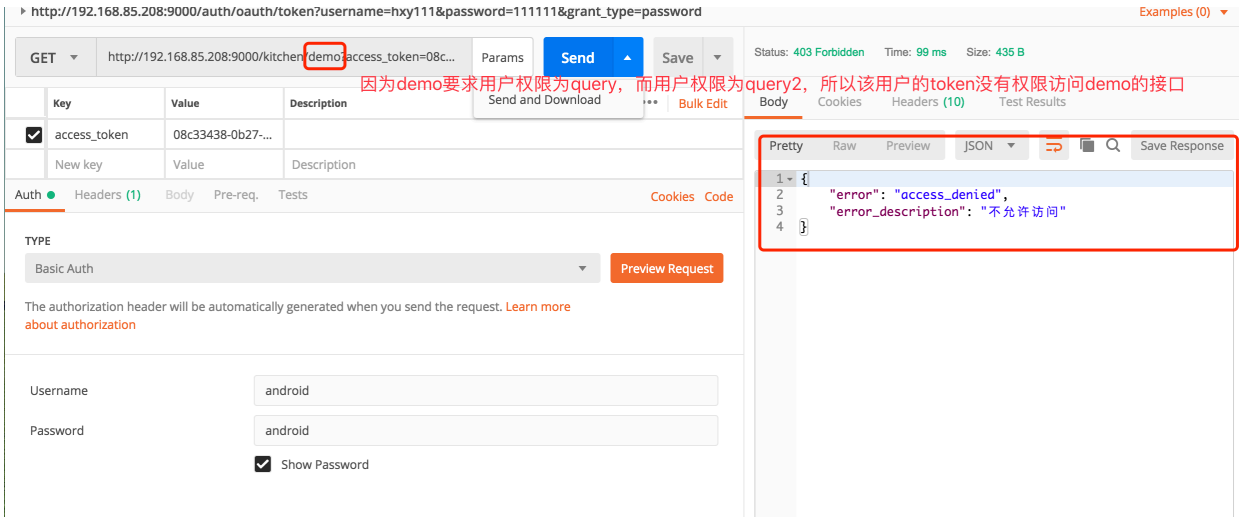
在params填写access_token

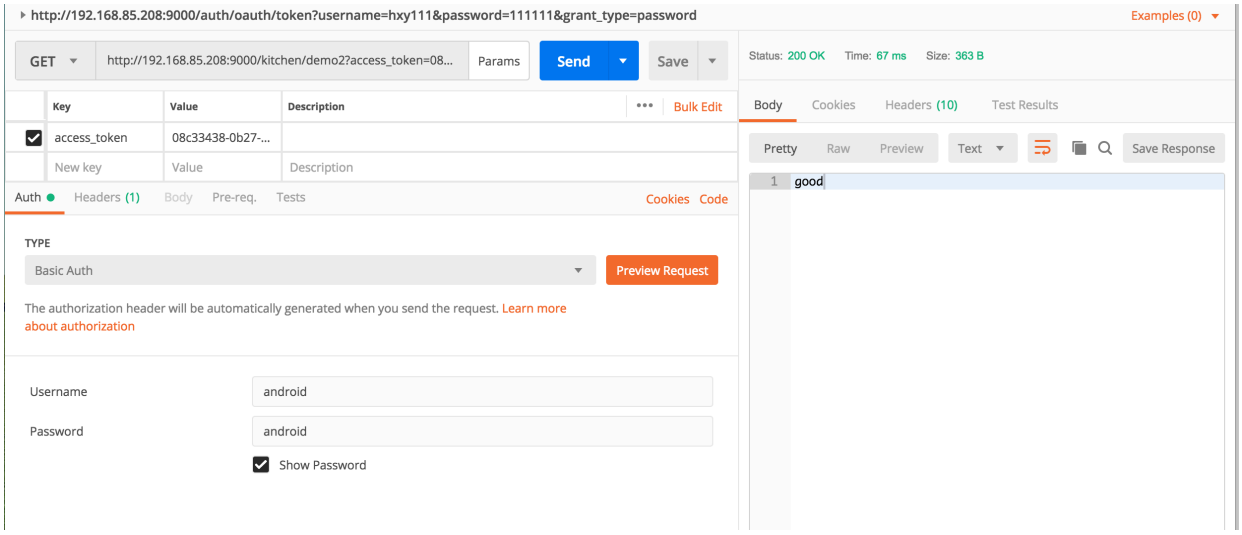



/auth/user/me




# 4、自定义资源服务模块使用（即功能模块）




# 三、新增服务配置

http://192.168.84.199:10080/earthchen/smart-home/src/master

1.后台新增微服务时

- 在新增对应服务的application.yml进行security配置

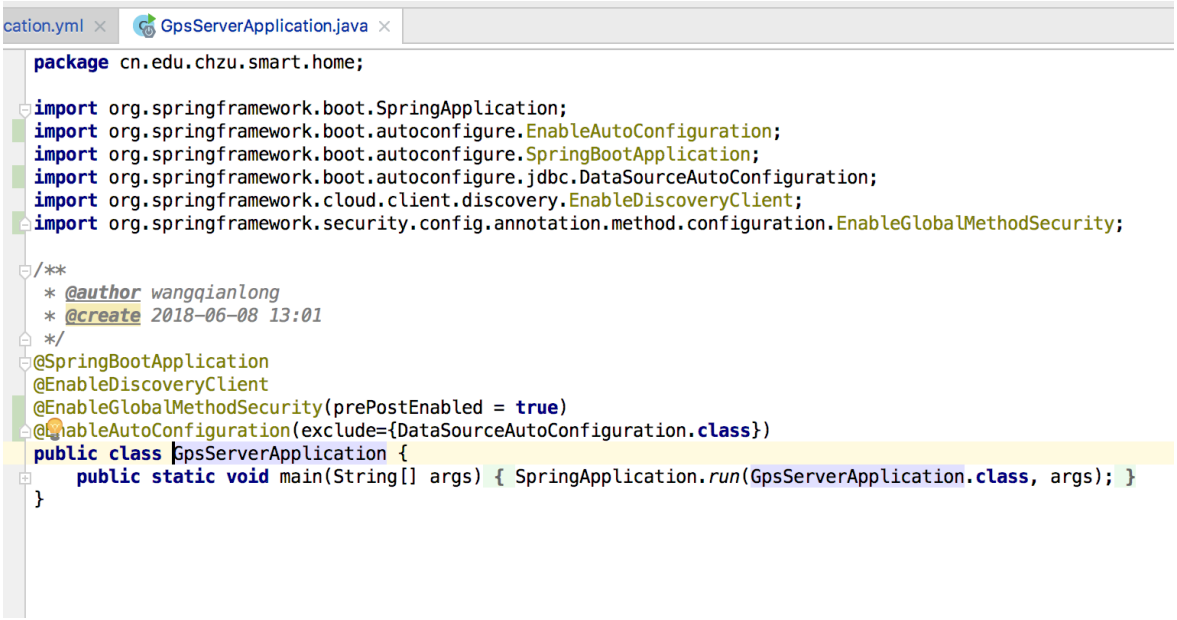
```
feign:
  hystirx:
    enable: true

security:
  oauth2:
    resource:
      id: smart-home-gps-server
      user-info-uri: http://192.168.85.208:9000/auth/user/current
      prefer-token-info: false
```

注释：若使用redis缓存—（当服务涉及到访问数据库使用redis缓存），还需要进行redis和mysql配置

- 微服务目录下新增cn.edu.chzu.smart.home.config包，创建配置类（复制已有的模块就可以）
- 网关zuul-gateway进行反向代理路由，zuul-gateway微服务的application.yml进行配置

```
zuul:
  routes:
    auth-server:
      path: /auth/**
      sensitiveHeaders:
      serviceId: auth-server
    kitchen-server:
      path: /kitchen/**
      sensitiveHeaders:
      serviceId: smart-home-kitchen-server
    gps-server:
      path: /gps/**
      sensitiveHeaders:
      serviceId: smart-home-gps-server
  add-proxy-headers: true
```

- 新增微服务的入口需要配置网关反向




2.客户端新增一个接口应用时，需要在auth-server（认证服务中）application.yml配置server-security

# 四、服务发布部署（推送到git并在服务器上运行）

## 1、版本推送

修改代码时，需要拉取最新版本代码（即从git上克隆版本历史中最新的版本代码），再次基础上进行修改

修改代码后，通过git push的方式推送到gitee平台自己的分支中，并注明修改的内容和版本

## 2、服务器部署运行

见服务器的readme.md的三种部署方式

http://192.168.84.199:10080/earthchen/smart-home/src/master

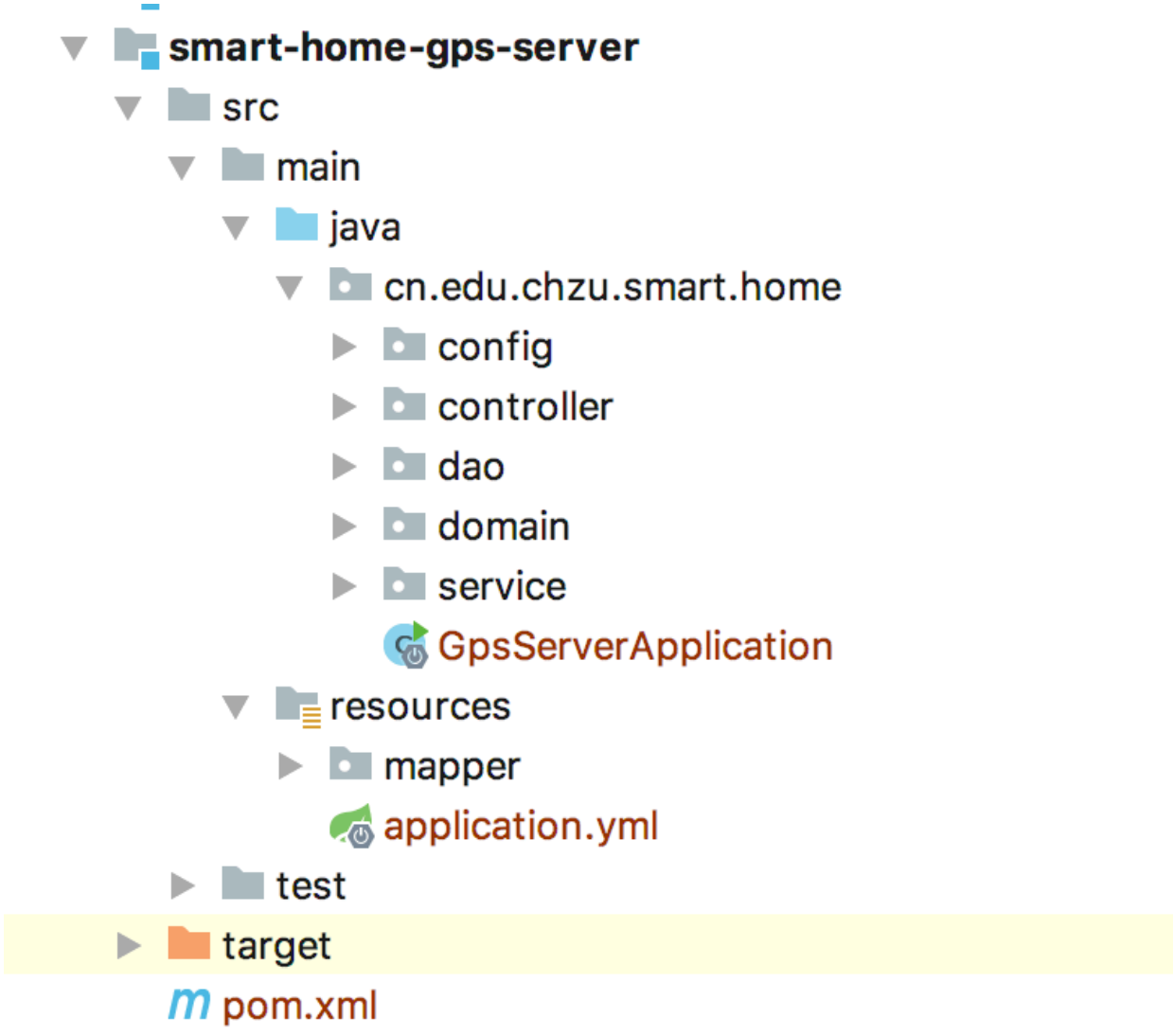
本文详细介绍docker方式进行服务器代码部署和运行

## 2.1 docker打包镜像

- 在自己开发的资源服务包下面新增一个Dockerfile文件（可以参考已有的文件）


注意：打包镜像时，请注意需要进入dockerfile所在的目录，运行如下命令:

# 五、资源服务文件结构




config

controller

dao

domain

service

GpsServerApplication——入口函数

resource-mapper

resource-application.yml