# 服务器前后端开发文档

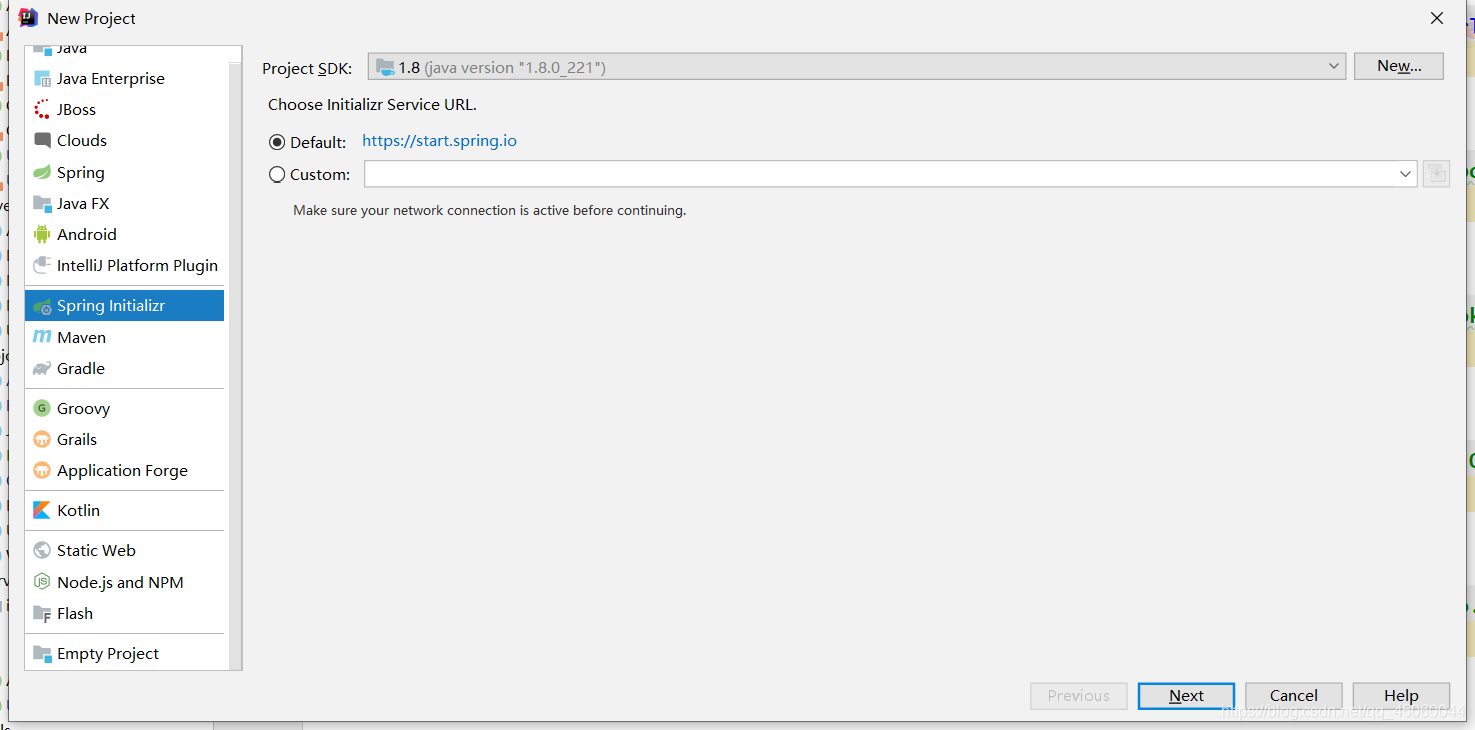
1基于SpringBoot框架的Web应用后端,基于Android的app程序

2使用Mysql数据库,使用MQTT服务器

3开发工具为IDEA2020.3版本

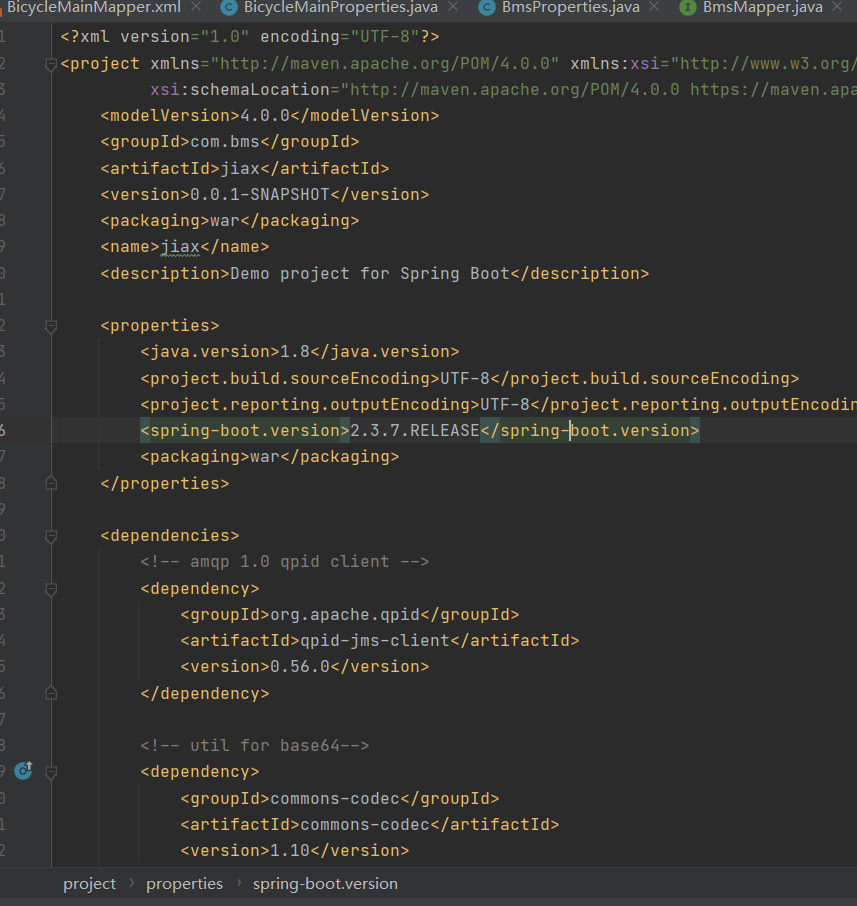
## 一.SpringBoot开发框架

### （1）新建SpringBoot项目

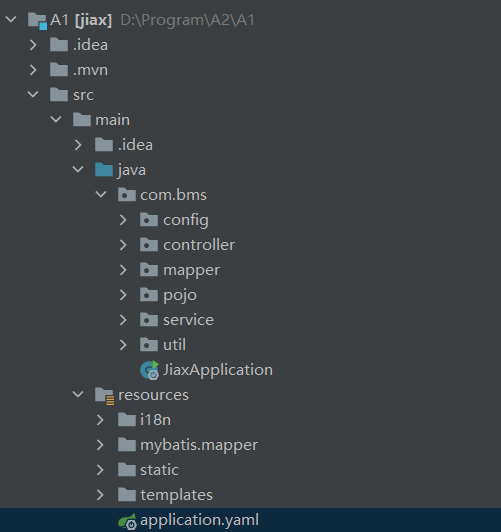


### SpringBoot依赖

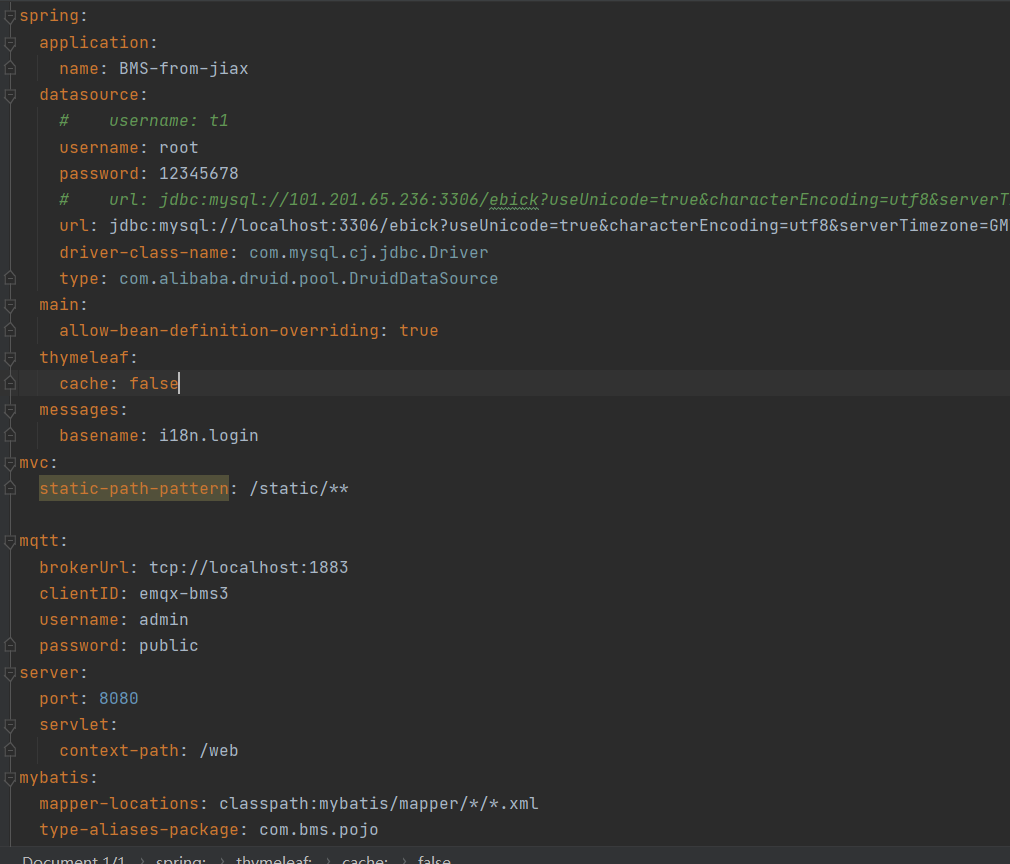
导入MAVEN依赖



### （3）项目目录下包分类

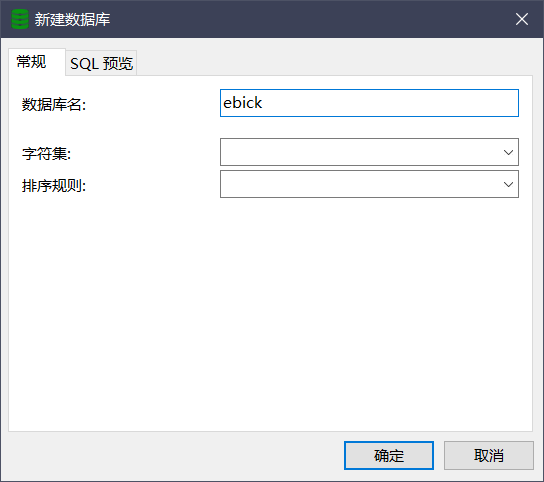


### （4）SpringBoot全局配置



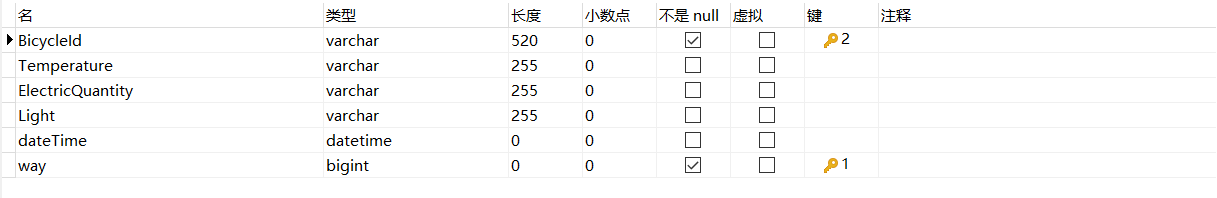
## Mysql数据库表建立

### （1）新建数据库ebick



### （2）新建表

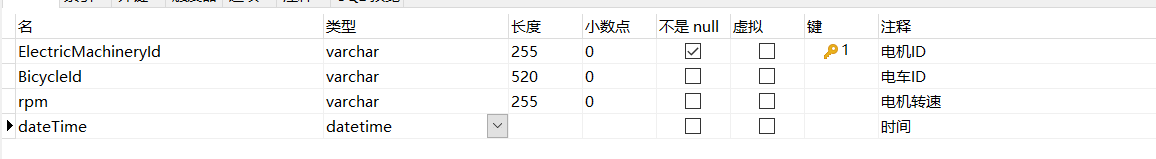
表bicmain 电车数据表



表bicid 电车关系表

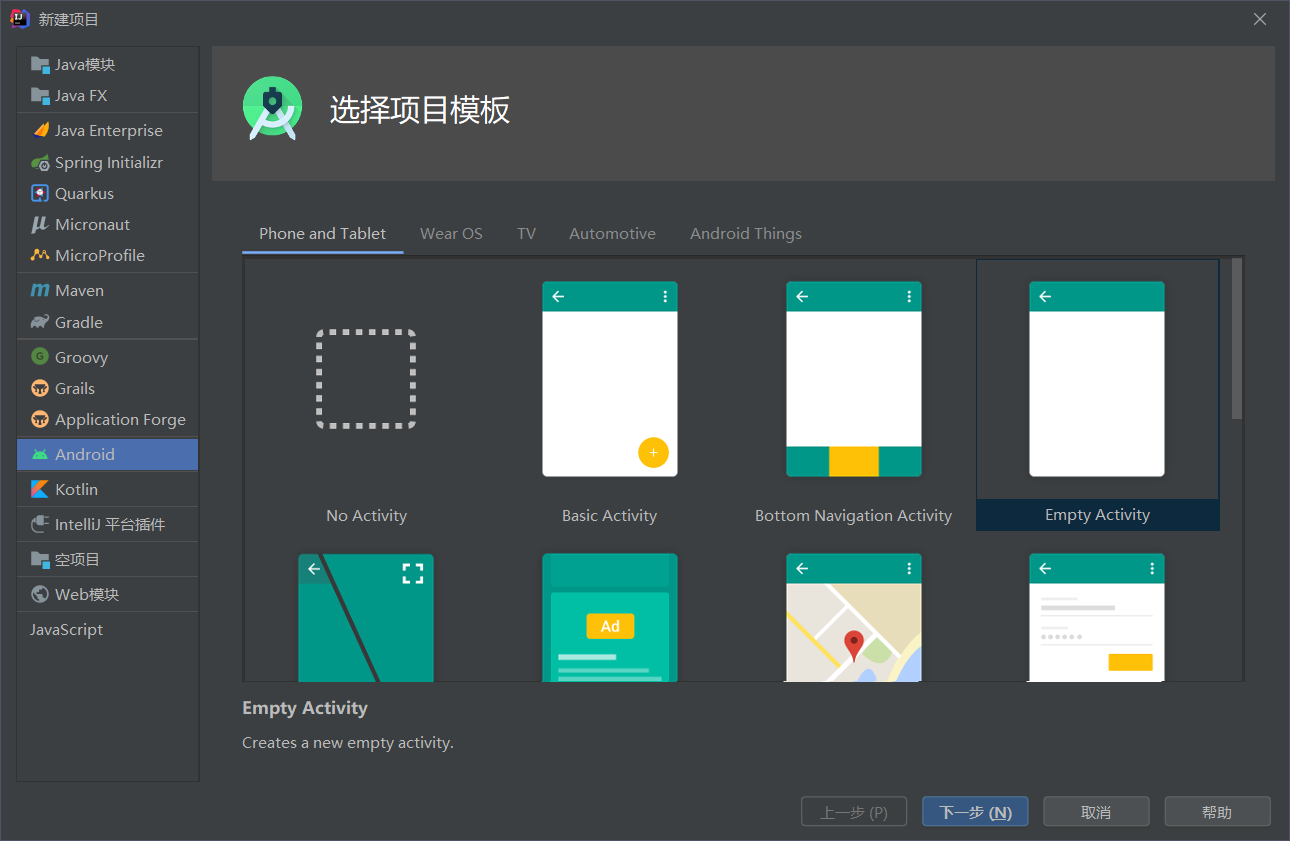


表electricmachinery 电机表



## 安卓程序建立

### （1）新建安卓项目



## MQTT服务器搭建

### （1）云服务器--通过 Docker 容器运行

EMQX 提供了一个容器镜像，

2.获取 Docker 镜像

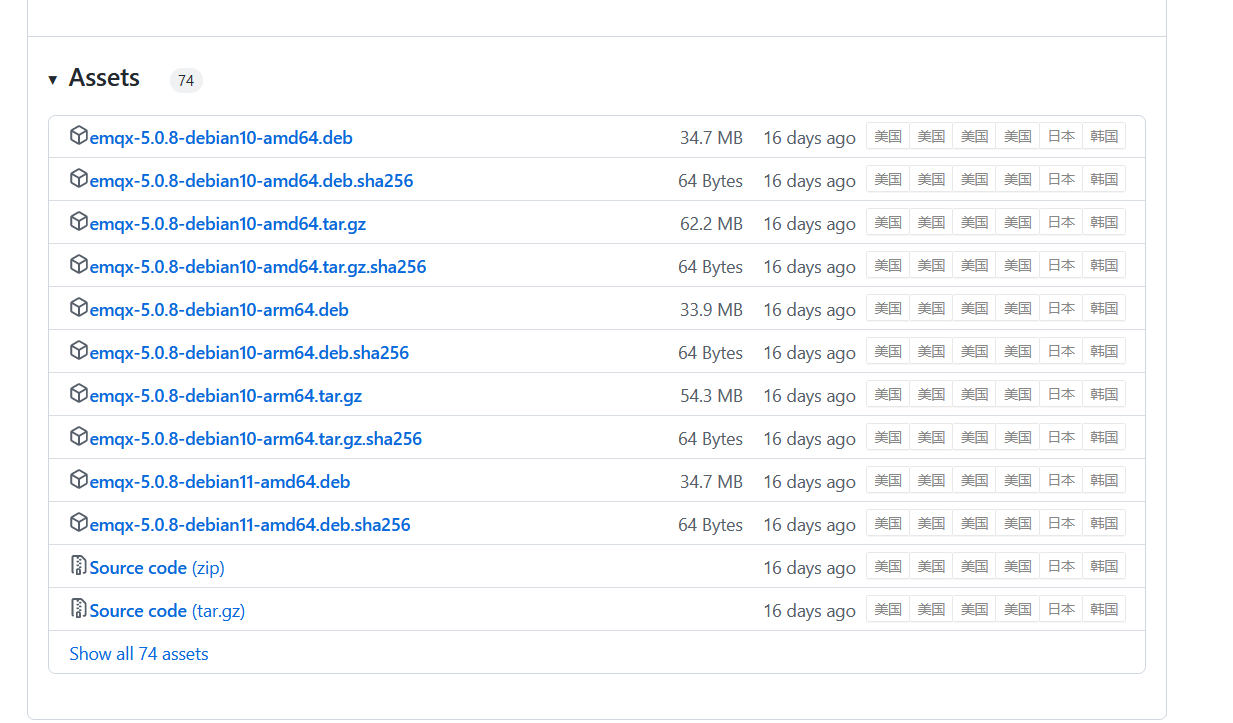
docker pull emqx/emqx:latest

2.启动 Docker 容器

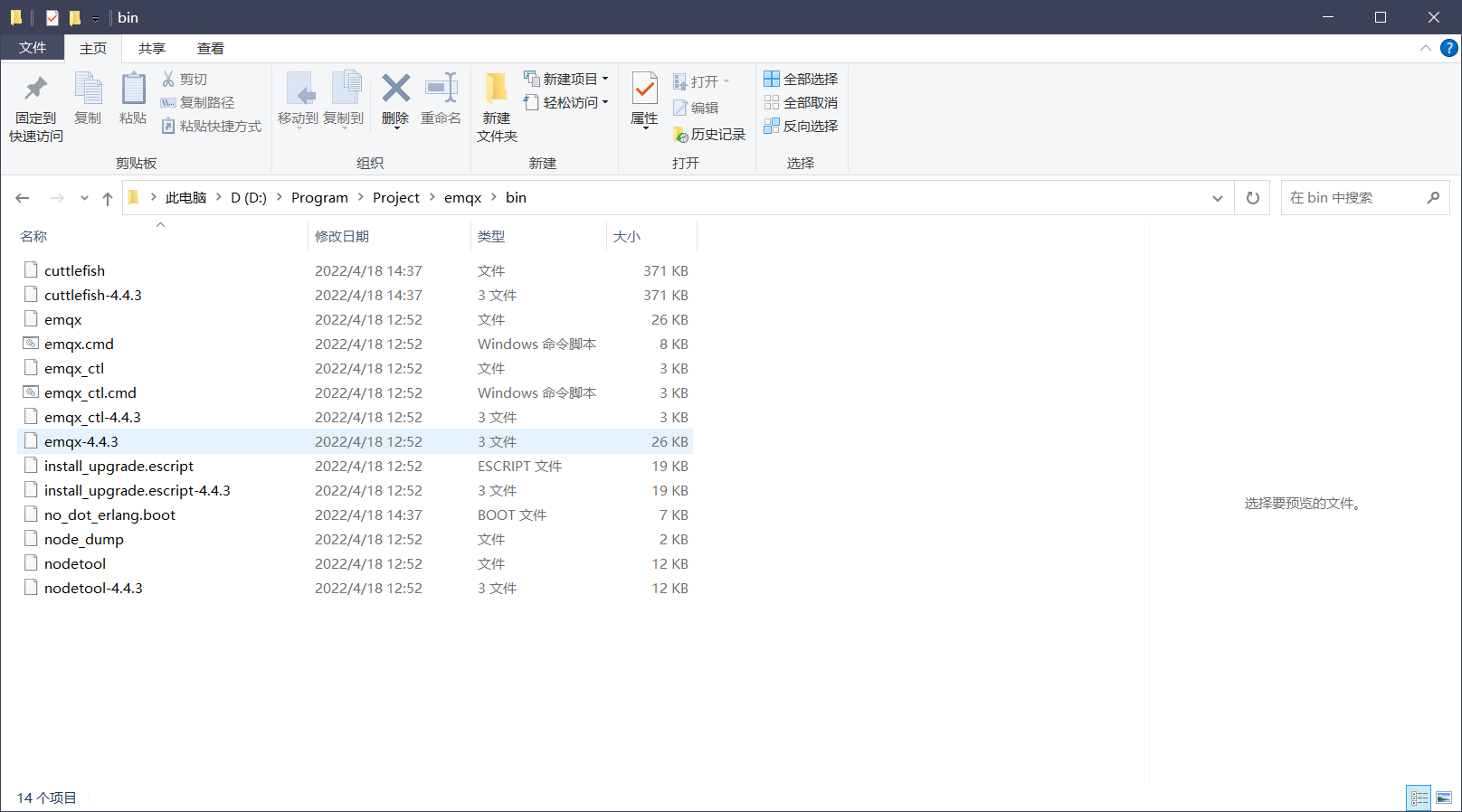
docker run -d --name emqx -p 1883:1883 -p 8081:8081 -p 8083:8083 -p 8084:8084 -p 8883:8883 -p 18083:18083 emqx/emqx:latest

（2）本地服务器--本地运行EMQ服务器

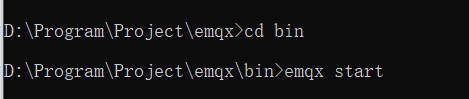
访问GitHub: <https://github.com/emqx/emqx/releases> 安装合适系统版本的EMQ服务器



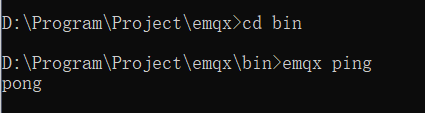
本地解压安装包后进入emqx/bin 目录



进入cmd界面 通过命令 emqx start 启动mqtt服务器



服务器启动后, 输入emqx ping命令 返回pong代表启动成功



## 项目具体实现

### MQTT消息格式定义

客户端接收Topic:Device1,

消息结构体

{

"Temperature":"22",

"ElectricQuantity":"22",

"Light":"1"

}

用于发布电车的温度,光照,电量数据

客户端接收Topic:Device2,

消息结构体

{

"Temperature":"22",

"ElectricQuantity":"22",

"Light":"1"

}

用于发布断电重启后历史的电车的温度,光照,电量数据

客户端接收Topic:Device3, 客户端发布Topic:keepAlive

发布消息结构体

{

“time”:20220163025

}

用于校准硬件部分时间

客户端发布Topic:HeartBeat

发布消息结构体

{

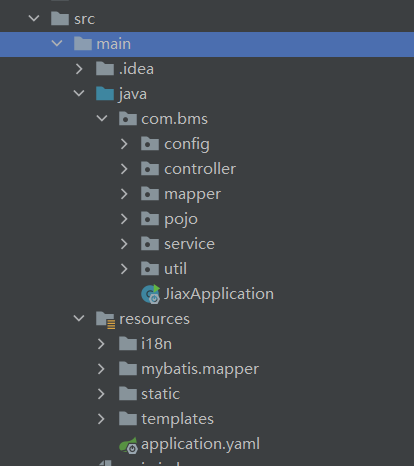
“live1”:20220163025

}

定时为25秒发布至硬件段,用做心跳包

### 服务器后端

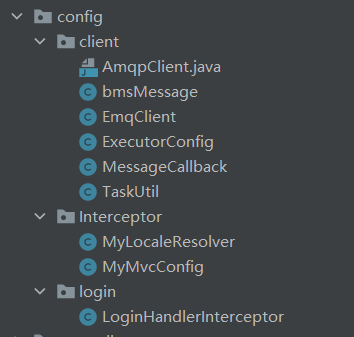
1基本结构



分为JAVA代码部分 与resources 资源部分 以及application.yml配置文件

2.1 java代码部分

Config包:



2