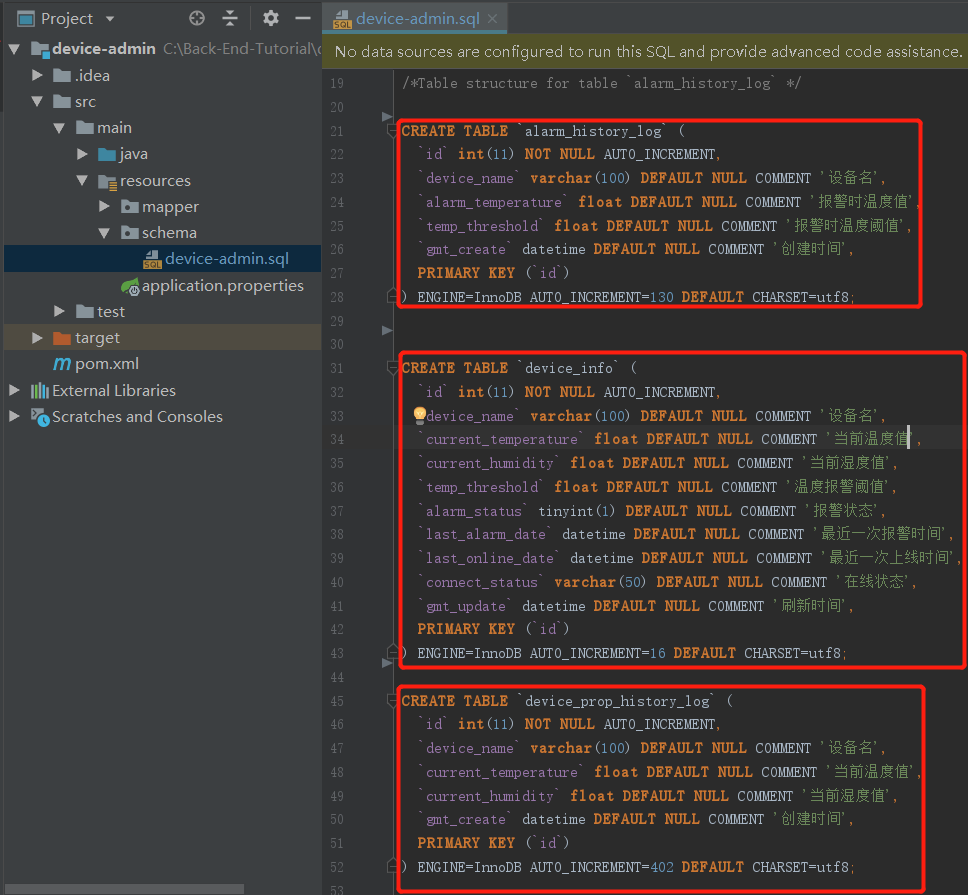
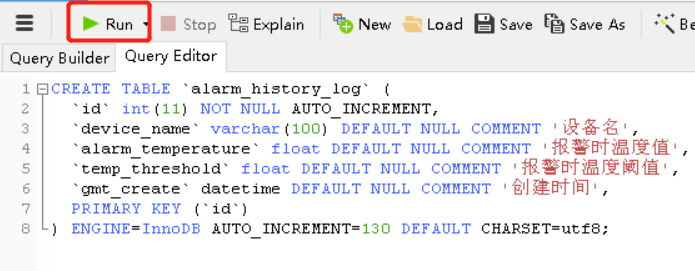
# 运行后端device-admin应用步骤

# 建立数据库

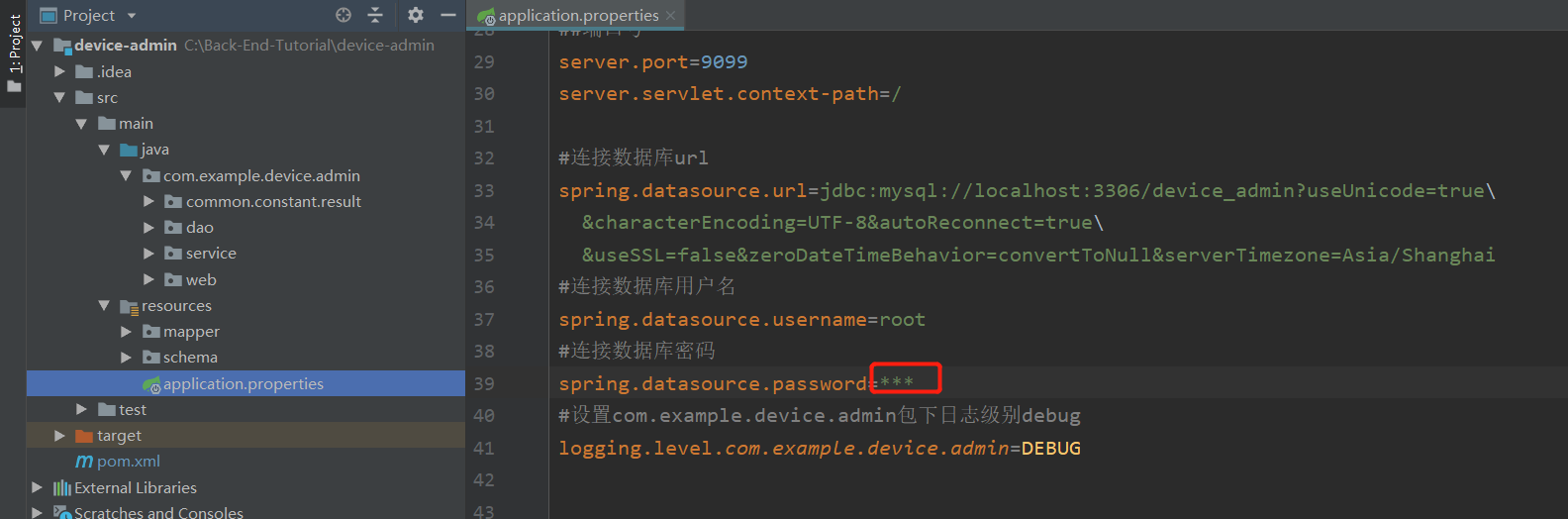
使用Navicat建立MySQL数据库device-admin，在device-admin项目中java/resource/schema文件下存放有 device-admin.sql文件，可以直接复制到navicat或者命令行执行。建立数据库device-admin，建立3张表。

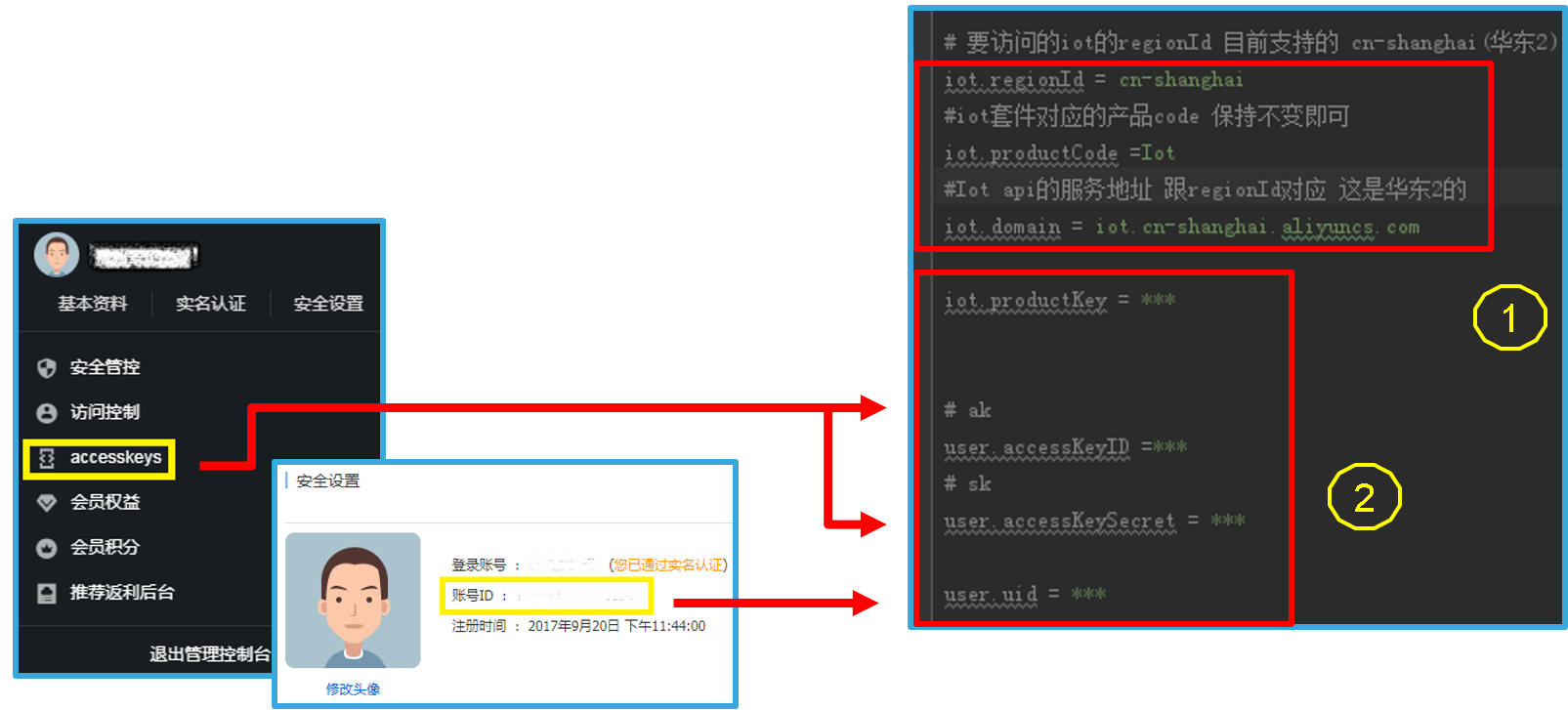




# 更改配置

在src/main/resource中的application.properties中修改数据库密码



在src/main/resource中的application.properties中新增对阿里云IoT平台访问的配置：阿里云IoT平台的账号信息，包含账户Id，用户AccessKey；阿里云IoT平台访问区域信息；阿里云IoT平台上要访问的产品信息，即iot.productKey,本项目中配置的是单个ProductKey。

# 服务器端订阅消息

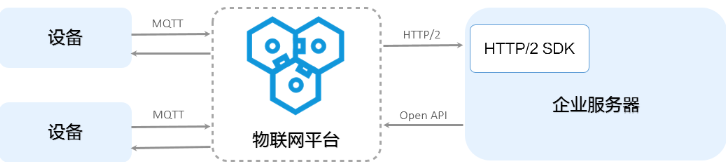
服务端可以直接订阅阿里云iot平台产品消息，阿里云iot平台是通过HTTP/2通道进行消息流转.同时阿里云HTTP/2 SDK提供身份验证、topic订阅、消息发送、消息接收等，HTTP/2订阅消息适用于平台和服务器间大量数据流转,开发之前需要在阿里云物联网平台配置服务端订阅.在服务端订阅中设置选择推送的消息类型：

设备上报消息：所有具有发布权限的topic

设备状态变化通知：上线、下线

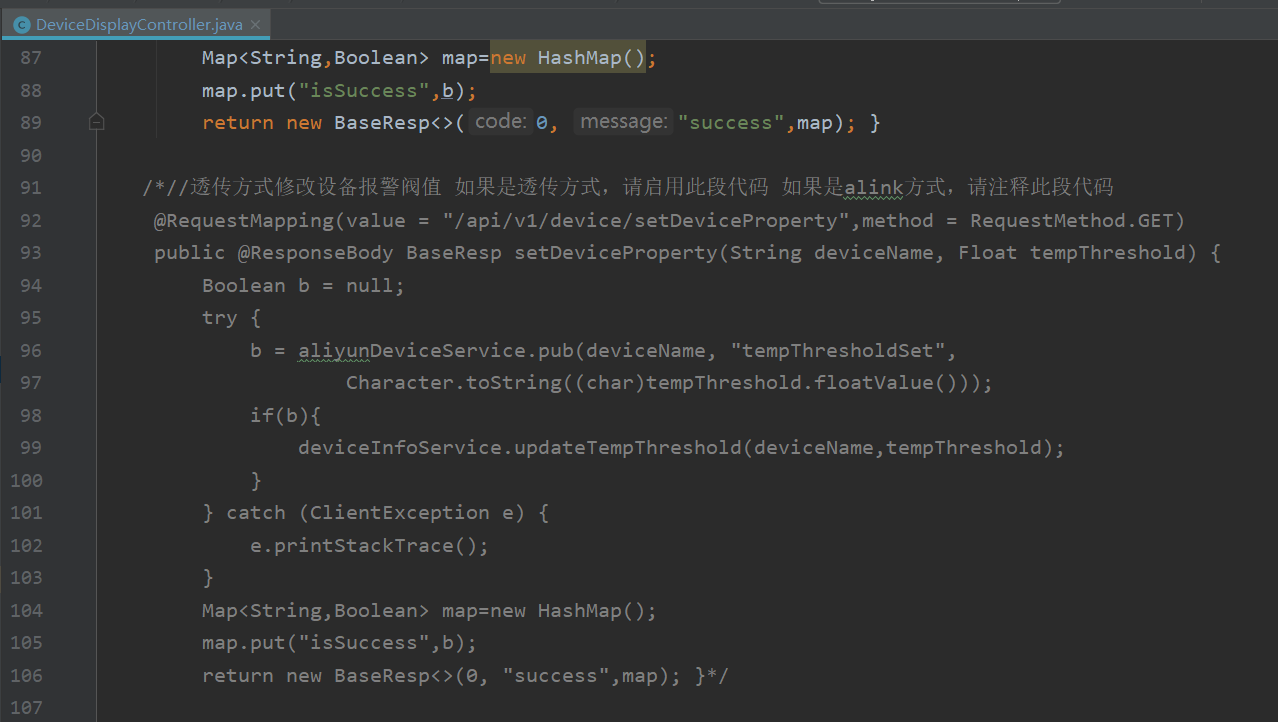
设备生命周期变更：设备创建、删除等.

详情可参考阿里云iot开发文档https[://help.aliyun.com/document\_detail/89227.html?spm=5176.11065259.1996646101.searchclickresult.214d5a42bGRoOj](https://help.aliyun.com/document_detail/89227.html?spm=5176.11065259.1996646101.searchclickresult.214d5a42bGRoOj)

****

# 启动透传代码

在com.example.device.admin.web.controller包下的DeviceDisplayController类启动透传代码透传方式修改设备报警阀值以及透传方式设备报警状态取消代码，注释掉Alink方式设备报警状态取消代码以及Alink方式设备报警状态取消代码。

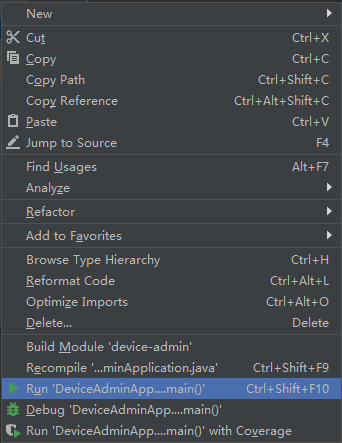


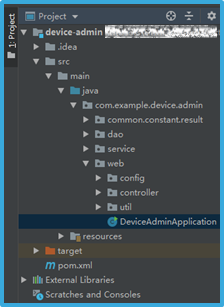
# 运行IDEA项目

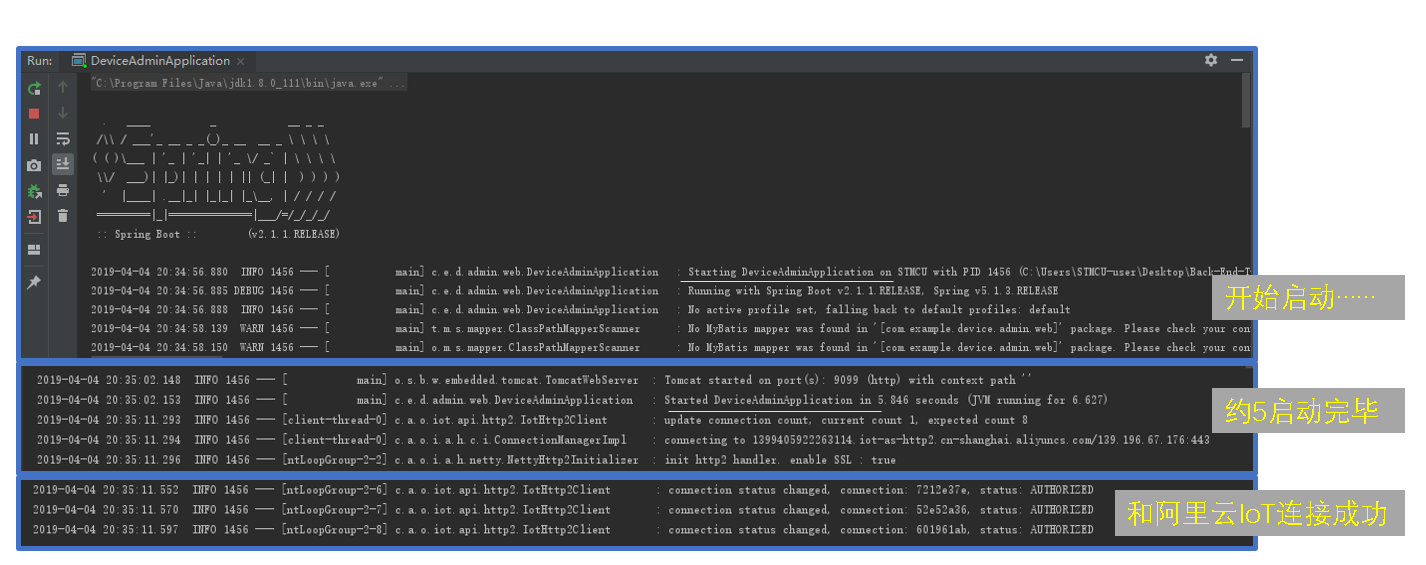
项目代码开发完毕以后在本地运行IDEA项目如图所示：

选中DeviceAdminApplication.java,鼠标右键 🡪 菜单中选择 “Run”，

可以看到控制台中打印了启动的日志信息，如下图：

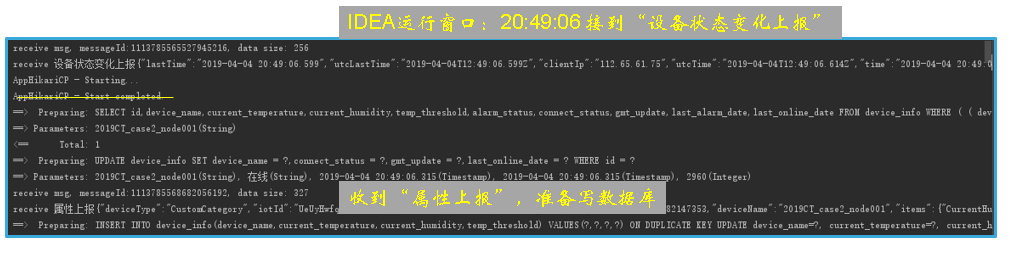






项目运行启动完成与IoT平台日志如下图所示：

节点设备连接阿里云IoT平台，定期上报数据设备属性，后端应用收到设备状态变化上报并写入数据库；收到属性上报以后，写入数据库，可结合阿里云iot平台查看设备日志，查询数据库查询数据。

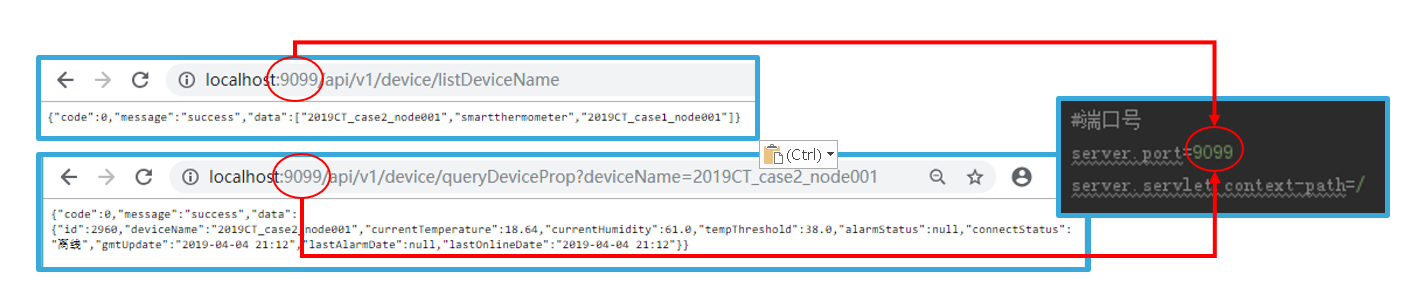




本地启动完成以可通过浏览器模拟测试前端发来的Get请求。

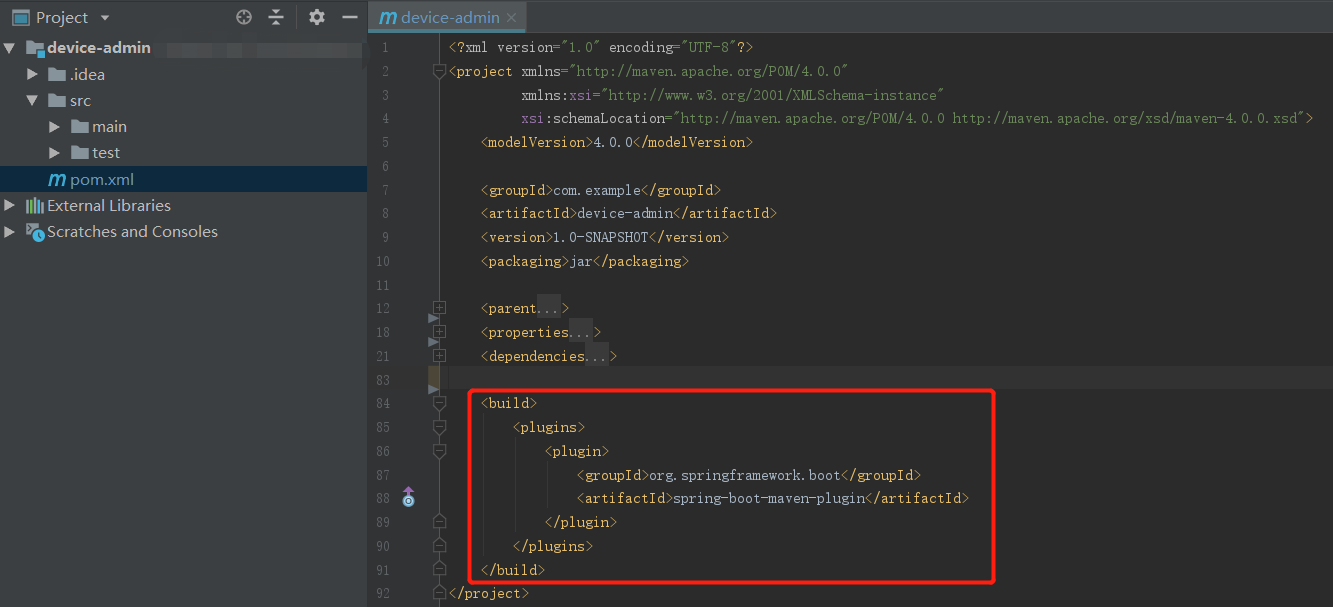
举例：查询所有设备名，通过向浏览器数据http://localhost:9099/api/v1/device/listDeviceName发起Get请求得到所有设备名结果。

查询2019CT\_case2\_node001的设备属性, 通过向浏览器数据http://localhost:9099/api/v1/device/queryDeviceProp?deviceName=2019CT\_case2\_node001发起Get请求得到设备属性。



## 后端打包

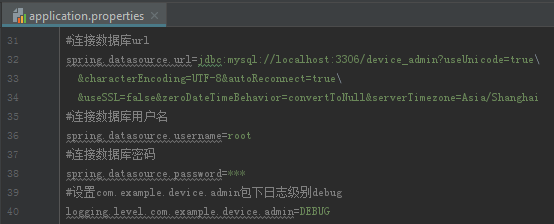
后端打包是目前后端开发的最终步骤，开始打包之前需要在pom.xml文件中添加spring-boot-maven-plugin插件，在添加了该插件之后，项目打包会打包成一个可以执行jar文件。



# 项目打包部署

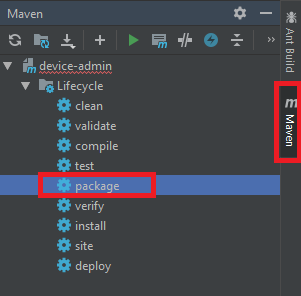
## 修改项目配置文件

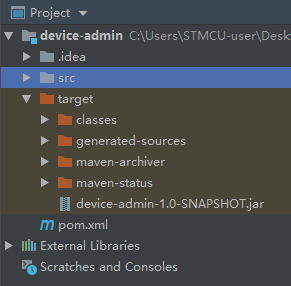
部署在公网服务器首先需要确认在application.properties中修改数据库配置的数据库ip地址，用户名，密码正确，确保服务器网络能连上数据库。



## 项目打包

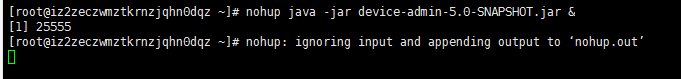
通过IDEA 右侧的Maven父工程下双击package即开始打包。打包内置了tomcat容器，所以我们只需要把项目打包，用命令提示符运行起来即可。打包完成后，在项目中web工程的target目录下出现一个可执行的jar文件。





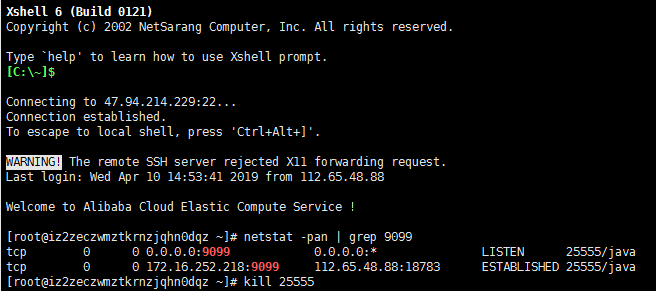
## 项目运行

把打包好的jar包通过Git Bash上传至公网服务器以后项目运行可以使用Xshell工具连接Linux系统，通过运行java -jar+文件名 命令即可以完成启动，若需要在公网上如何不挂断开启程序，可以使用nohup java -jar +文件名 & 命令，其中nohup 意思是不挂断运行命令,当账户退出或终端关闭时,程序仍然运行;&代表在后台运行。



若想关闭程序，可以使用netstat –pan |grep 端口号命令查看端口占用情况，

使用kill  pid命令终止进程,关闭程序。



项目在公网运行起来以后，需要在公网服务器开通项目端口号，即可通过其他电脑浏览器访问接口。

