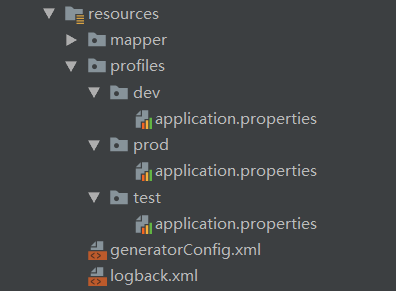
后端项目部署主要分为以下3个步骤：分别是配置编译环境，打包并上传，启动服务。

1. 配置编译环境

正常的项目开发会涉及到多个环境，后端主要会有dev开发环境，test测试环境，prod生产环境，配置不同的编译环境也有很多种方法，但中心思想还是通过参数控制打包不同的环境，我这里采用的是通过不同文件夹区分不同的环境，具体配置如下：

1）Resources资源文件配置，通过profiles文件下建立不同的文件夹。



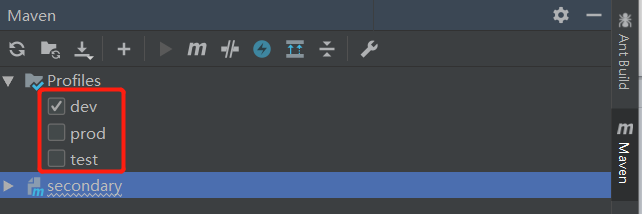
2）pom.xml配置打包环境参数

<profiles>  
 <profile>  
 <id>dev</id>  
 <properties>  
 <package.environment>profiles/dev</package.environment>  
 </properties>  
 </profile>  
 <profile>  
 <id>test</id>  
 <properties>  
 <package.environment>profiles/test</package.environment>  
 </properties>  
 </profile>  
 <profile>  
 <id>prod</id>  
 <properties>  
 <package.environment>profiles/prod</package.environment>  
 </properties>  
 </profile>  
</profiles>

并且在<build></build>标签中加入如下打包配置

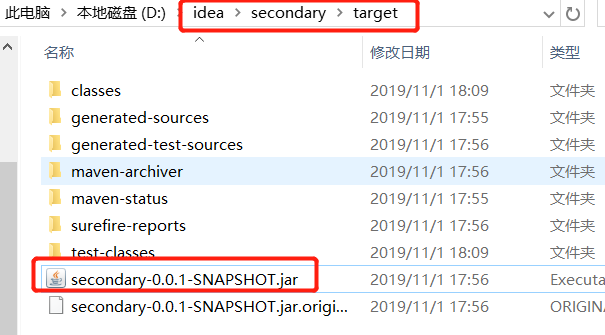
<resources>  
 <resource>  
 <directory>src/main/resources</directory>  
 <filtering>false</filtering>  
 <excludes>  
 <exclude>profiles/\*\*</exclude>  
 </excludes>  
 </resource>  
 <resource>  
 <directory>src/main/resources/${package.environment}</directory>  
 <filtering>true</filtering>  
 <excludes>  
 <exclude>profiles/\*\*</exclude>  
 </excludes>  
 </resource>  
</resources>

这样配置完成之后，就可以勾选对应的环境进行打包，如下图。也可以通过maven命令打包 mvn clean package –P dev。



1. 打包并上传

1）打包完成之后，在项目目录下会生成一个target目录，该目录下有一个jar包，如下图。



2）上传jar包至服务器。通过命令mkdir -p home/secondary 创建文件路径/home/secondary，紧接着进入该目录 cd /home/secondary，通过命令rz 上传jar包。

1. 启动服务

jar包上传成功之后就可以启动项目了，通过如下命令即可启动。

nohup java -jar -Xmx1024M secondary-0.0.1-SNAPSHOT.jar >/dev/null 2>&1 &。

nohup表示不挂断的运行命令，不会随着ctrl+c退出后导致项目停止。

-Xmx1024M是设置JVM最大可用内存为1024M。

>/dev/null表示输出重定向到null，即不输出到控制台。

2>&1表示将错误信息定位到输出。

& 表示后台运行，一般跟nohup配合使用。

至此，项目部署就完成了。