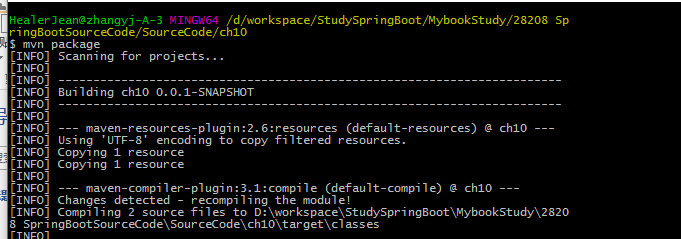
# 1.jar形式

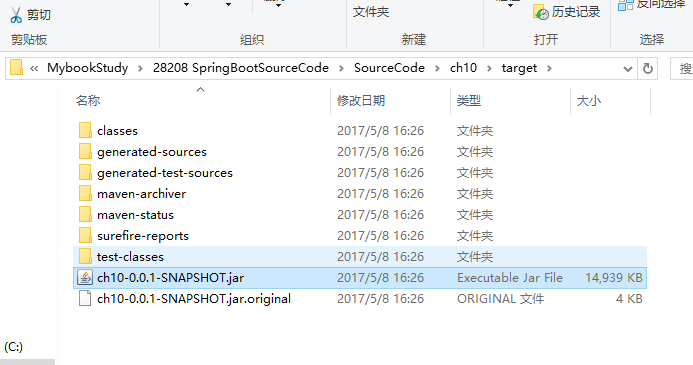
## 1.首先一定要安装marven

## 1.1移动打项目下面 以书上的为例

### mvn package

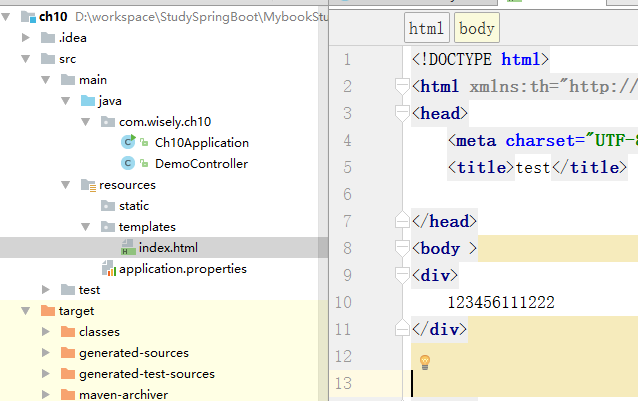


## 1.2 观察target下面的ｊａｒ包

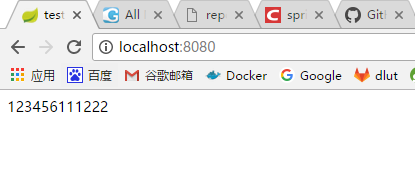


## 2.部署运行 相当于intellij idea 里面的启动项目

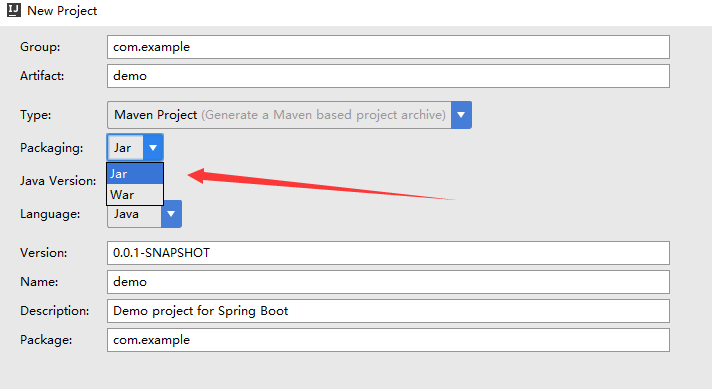
### $ Java -jar ch10-0.0.1-SNAPSHOT.jar



## 3.测试



# 2. war打包（新建项目时可以选择jar或者是war）



## 2.1 打包方式一样 mvn package

# 3.jar可以变为war

### 3.1 pom.xml中jar变为war

<**groupId**>com.studyBoot</**groupId**>  
<**artifactId**>studybook</**artifactId**>  
<**version**>0.0.1-SNAPSHOT</**version**>  
<**packaging**>jar</**packaging**> ----》war

## 3.2还有步骤，看书得了

# 4. 云部署 docker

## 1. 首先打包一个项目 ch10docker-0.0.1-SNAPSHOT.jar

## 2.在jar同级目录下创建一个 Dockerfile文件 (都放到项目根目录下面)

FROM java:8 //java版本为8  
  
MAINTAINER wyf //作者为wrf  
  
ADD ch10docker-0.0.1-SNAPSHOT.jar app.jar //将jar包添加到镜像中，并重命名为app.jar  
  
EXPOSE 8080 //运行镜像中的容器，端口为800  
  
ENTRYPOINT ["java","-jar","/app.jar"] 容器启动时候执行 这个命令

## 3.编译镜像

### $ docker build -t wisely/ch10docker .

#### 解释; wisely为前缀，可以自己取名称 ch10docker为项目的名字，后面的.表示Dockerfile 路径的，表示在当前路径下面

### 3.2测试 docker images

## 3 docker运行 （和基本命令中是一样的，只是顺序不一样而已）

### docker run–p 8080:8080 –name ch10 –d wisely/ch10docker

## 4.测试

## http://localhost:8080/

# 5.大工项目部署

## 1.gradlew clean

## 2.gradlew build