

IDEA 2017 建立简单的 spring boot [war 方式]

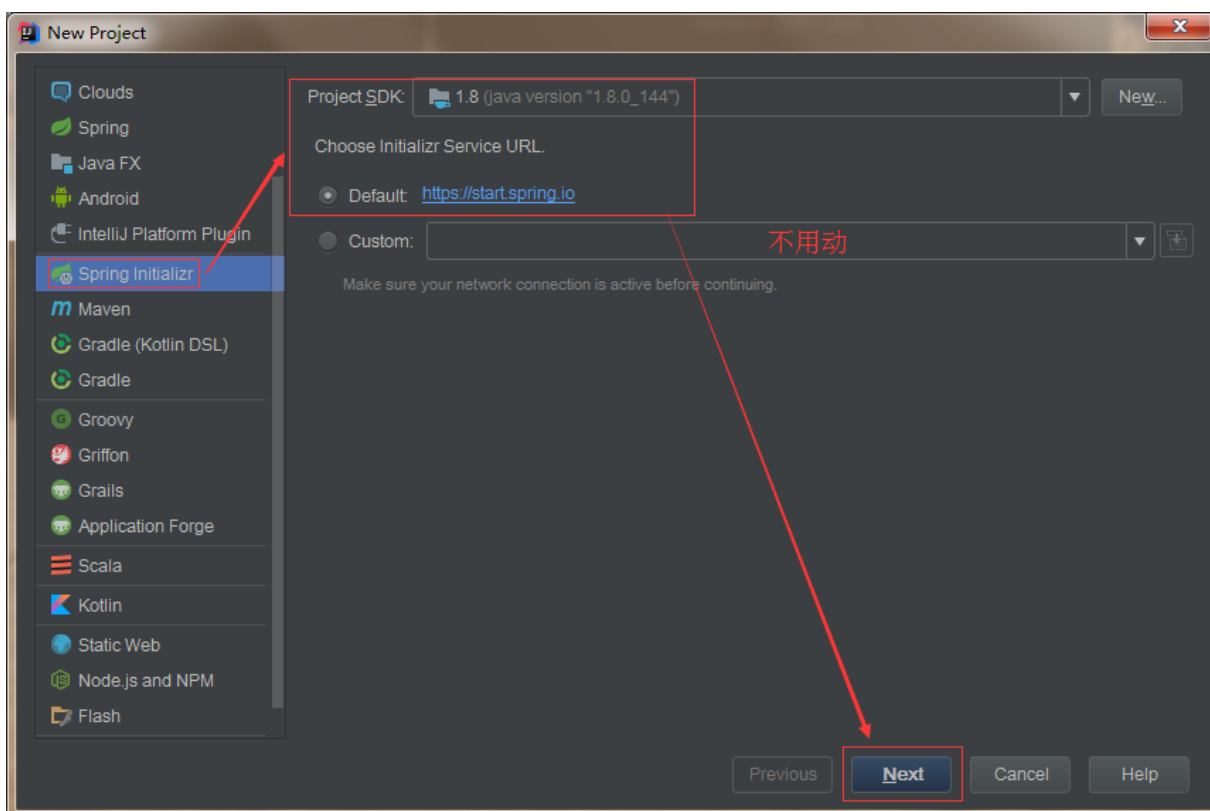
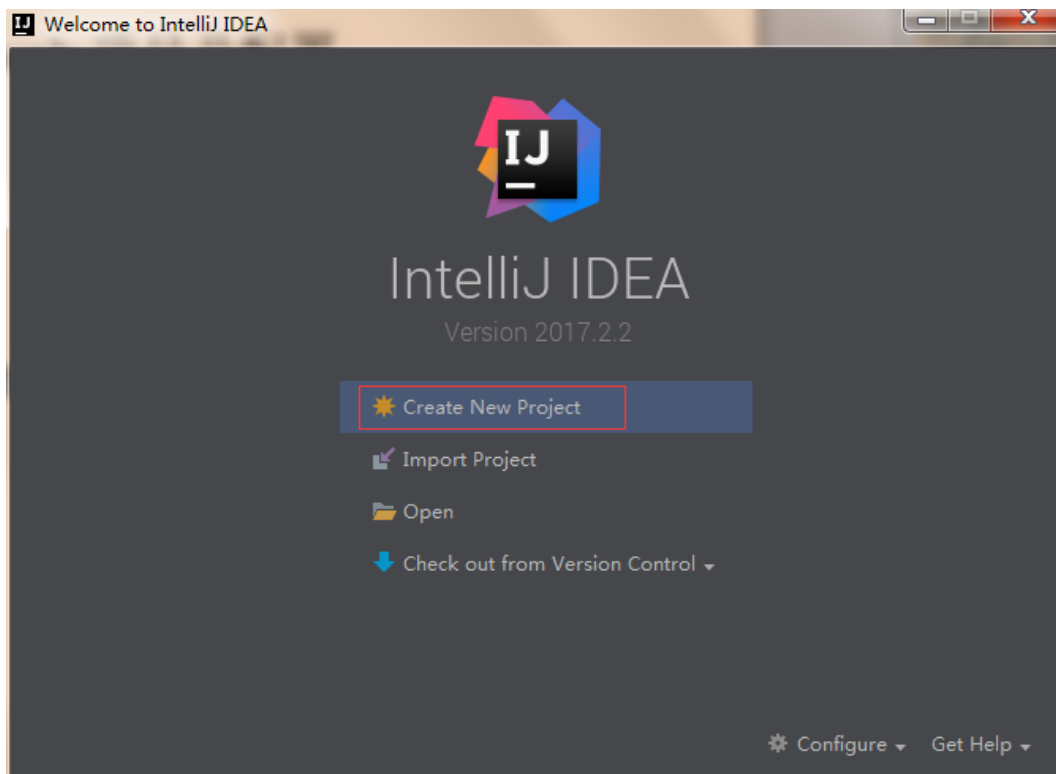
一、环境说明.....	1
二、建立 spring Boot 项目	1
三、用 Spring boot 建立一个简单 helloWorld	5
3.1 编写代码.....	5
#3.1.1 编写 HelloWorld 代码	5
#3.1.2 检测是否有语法错误.....	5
3.2 测试	6
#3.2.1 添加 tomcat 配置	6
#3.2.2 运行 tomcat 测试一下效果	8
3.3 打包成 war 发送到生产环境的 tomcat 上	8
#3.3.1 idea 生成 war 包.....	8
#3.3.2 配置线上环境 tomcat 虚拟主机并访问	9
1) 上传 war 文件.....	9
2) 配置 tomcat 虚拟主机	9
3) 测试.....	9
四、jar 方式转 war 打包	9
4.1 修改打包形成.....	10
4.2 去掉内置 tomcat	10
4.3 修改启动类，并重写初始化方法.....	10
4.4 配置项目结构 Project structure.....	11
附录一、参考文章.....	13

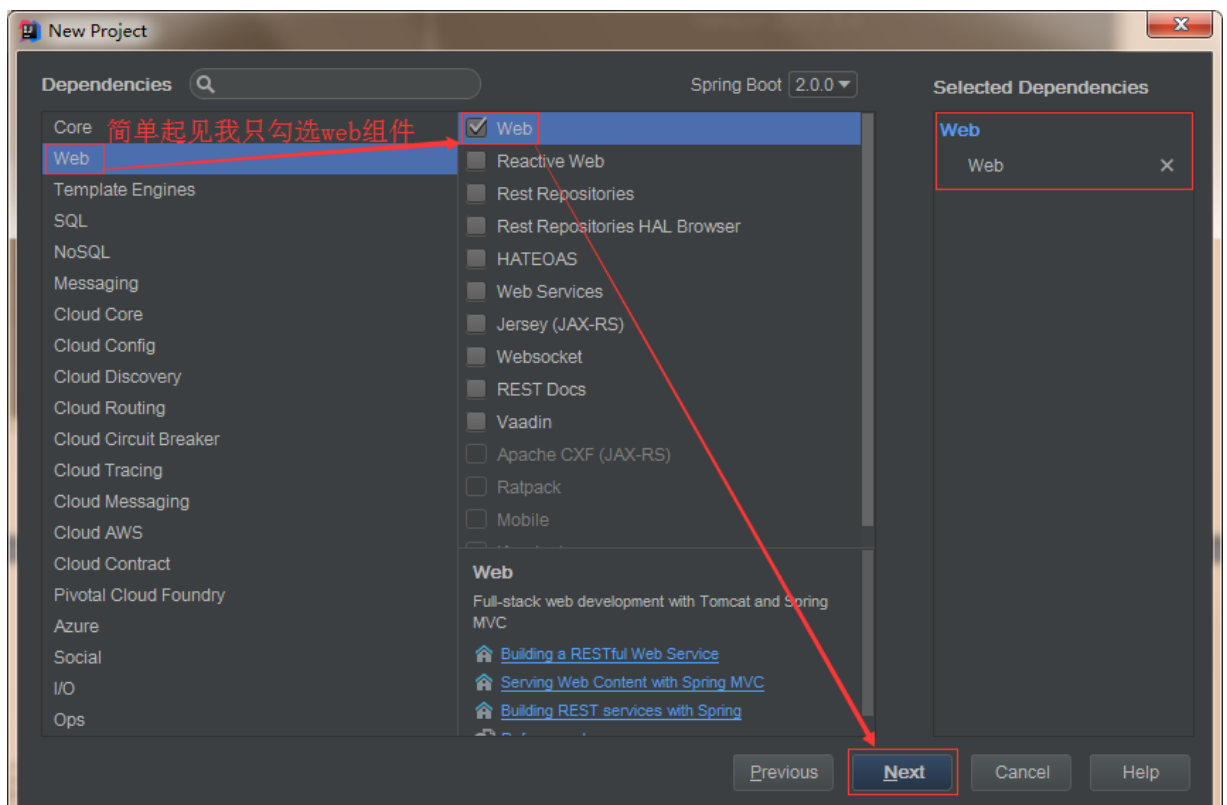
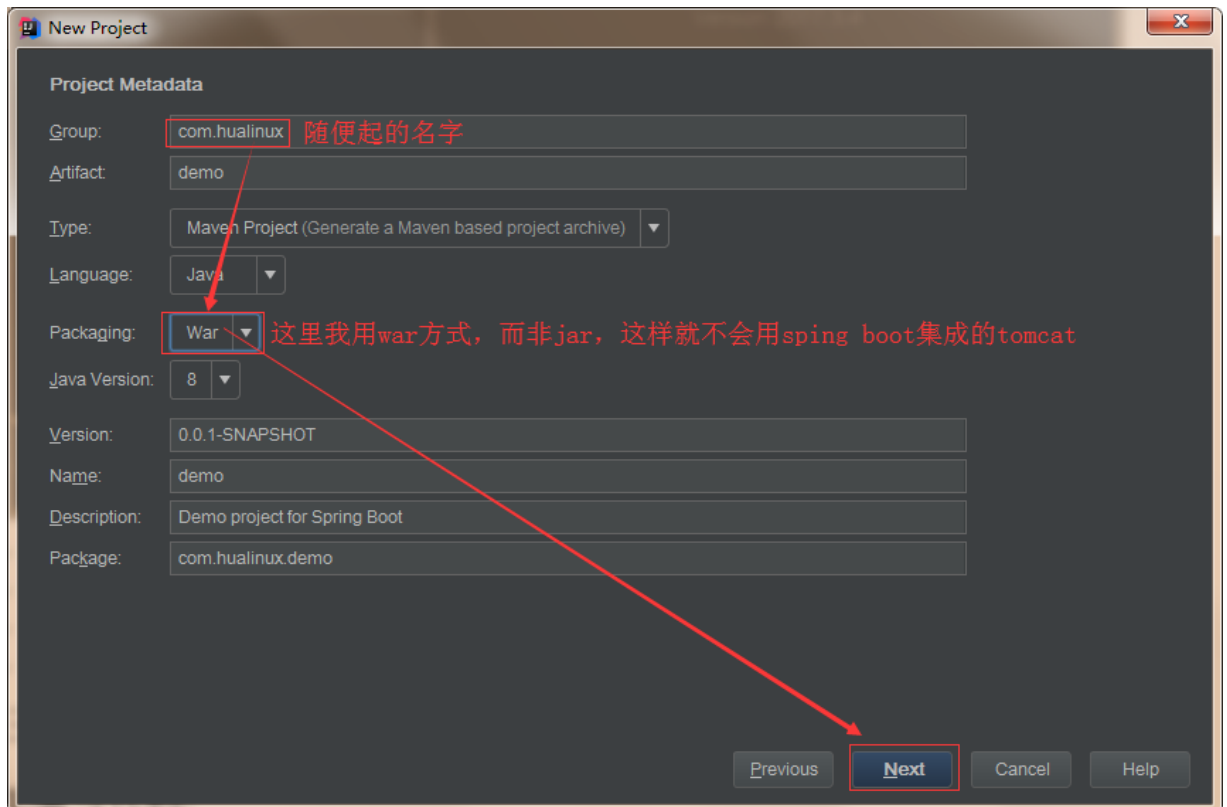
一、环境说明

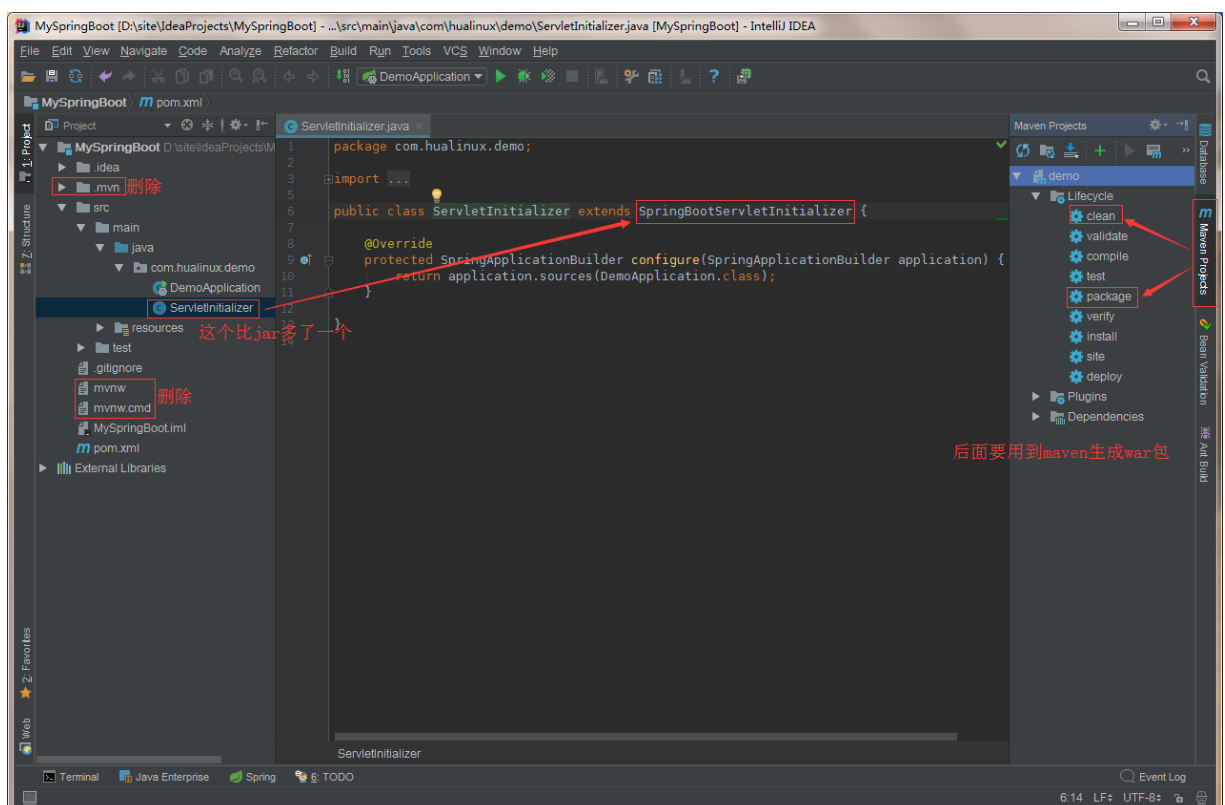
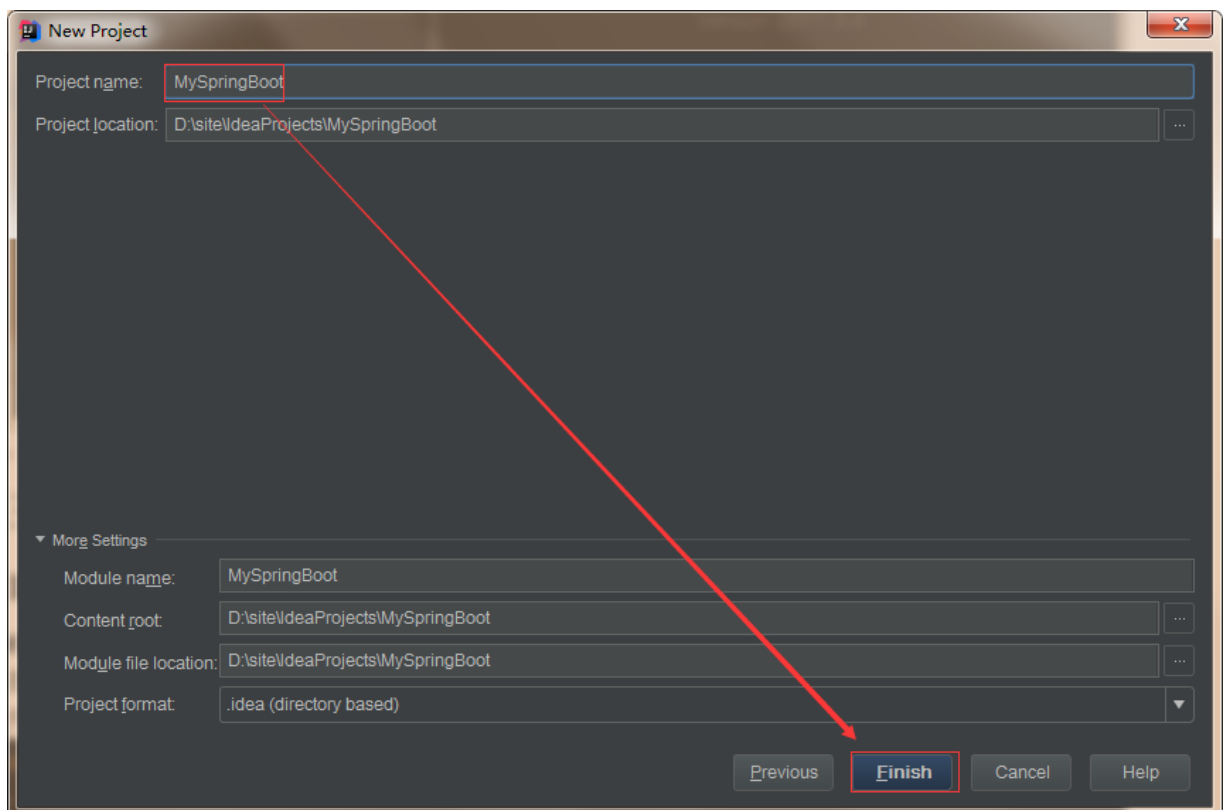
已经配置好了 JDK、tomcat8.5.20（二进制）、安装好了 idea 2017.3.4
spring boot 直接通过 idea 下载，省去安装的麻烦。

二、建立 spring Boot 项目

如果你之前有建立过项目的话，先关闭项目 “File→Close Project”







注:

我们需要类似于 web.xml 的配置方式来启动 spring 上下文, 所以在 DemoApplication 同级 idea 自动帮我们生成多了个 ServletInitializer 类, 继承 SpringBootServletInitializer 并重写 configure 方法。

三、用 Spring boot 建立一个简单 helloWorld

3.1 编写代码

#3.1.1 编写 HelloWorld 代码

打开 src→main→java→com.hualinux.demo→DemoApplication.java，修改成如下代码：

粗体为添加部分

```
package com.hualinux.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

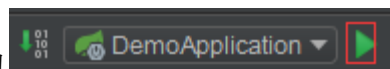
@RestController
@SpringBootApplication
public class DemoApplication {

    @RequestMapping("/")
    String index() {
        return "Hello Spring Boot";
    }

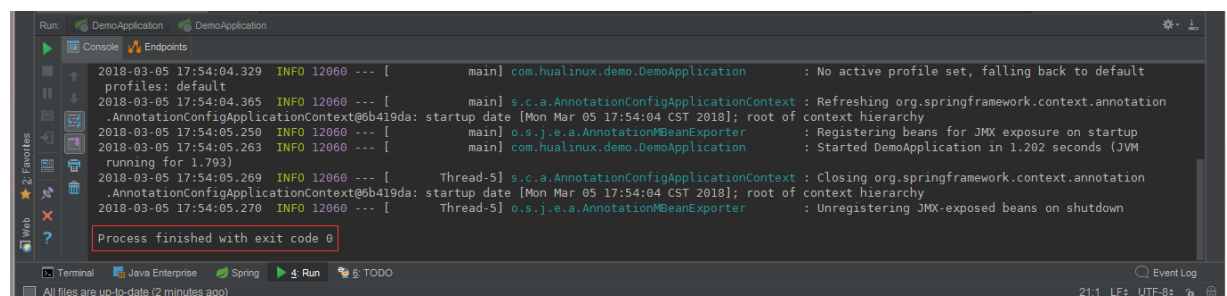
    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }
}
```

#3.1.2 检测是否有语法错误

按工具栏上的运行按钮



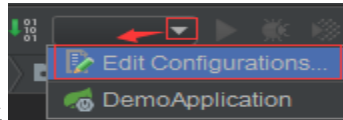
如通过，则语法没有错误



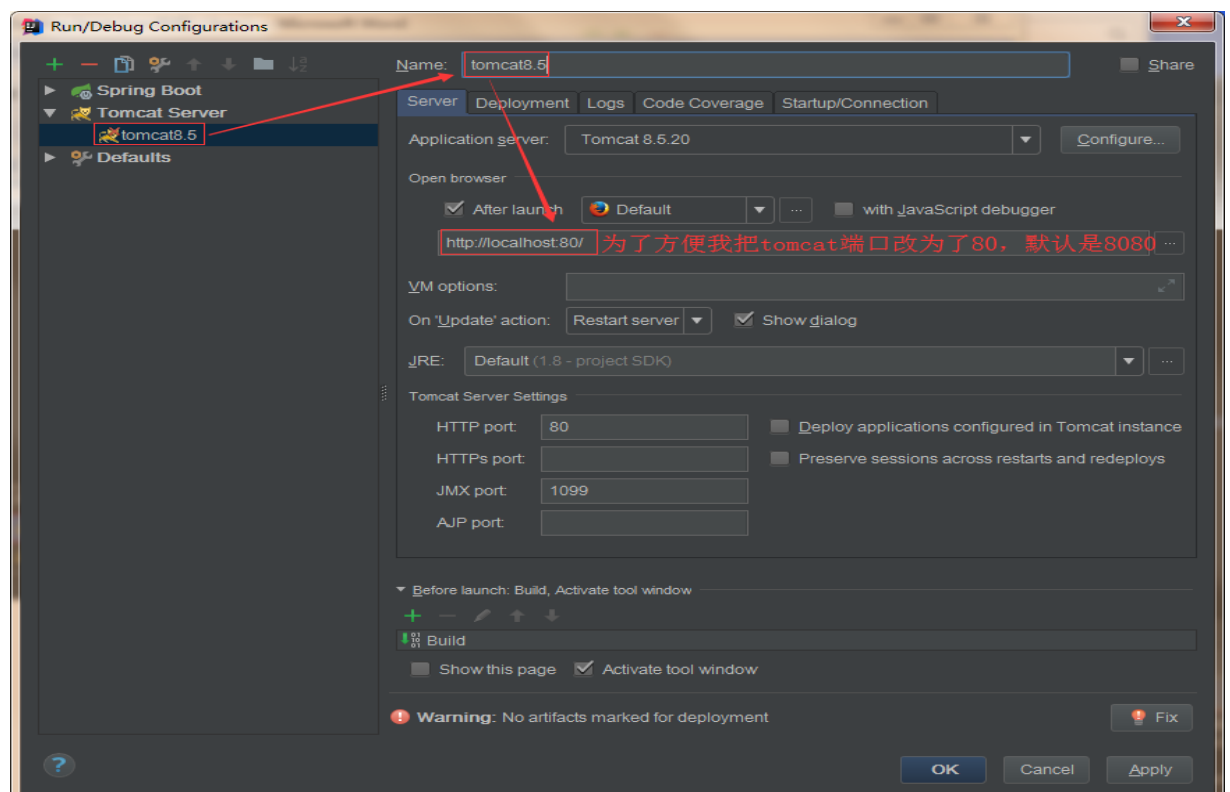
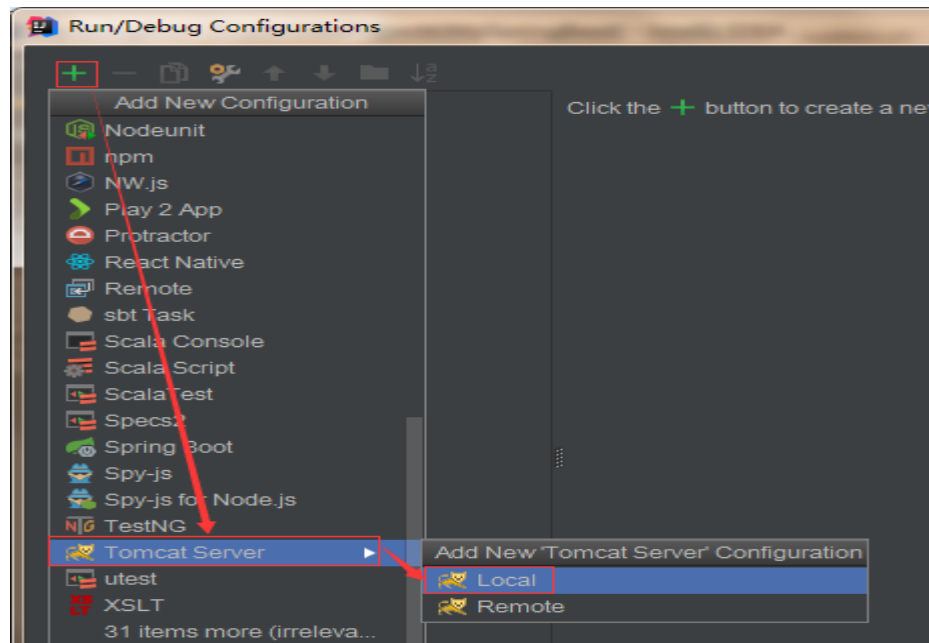
3.2 测试

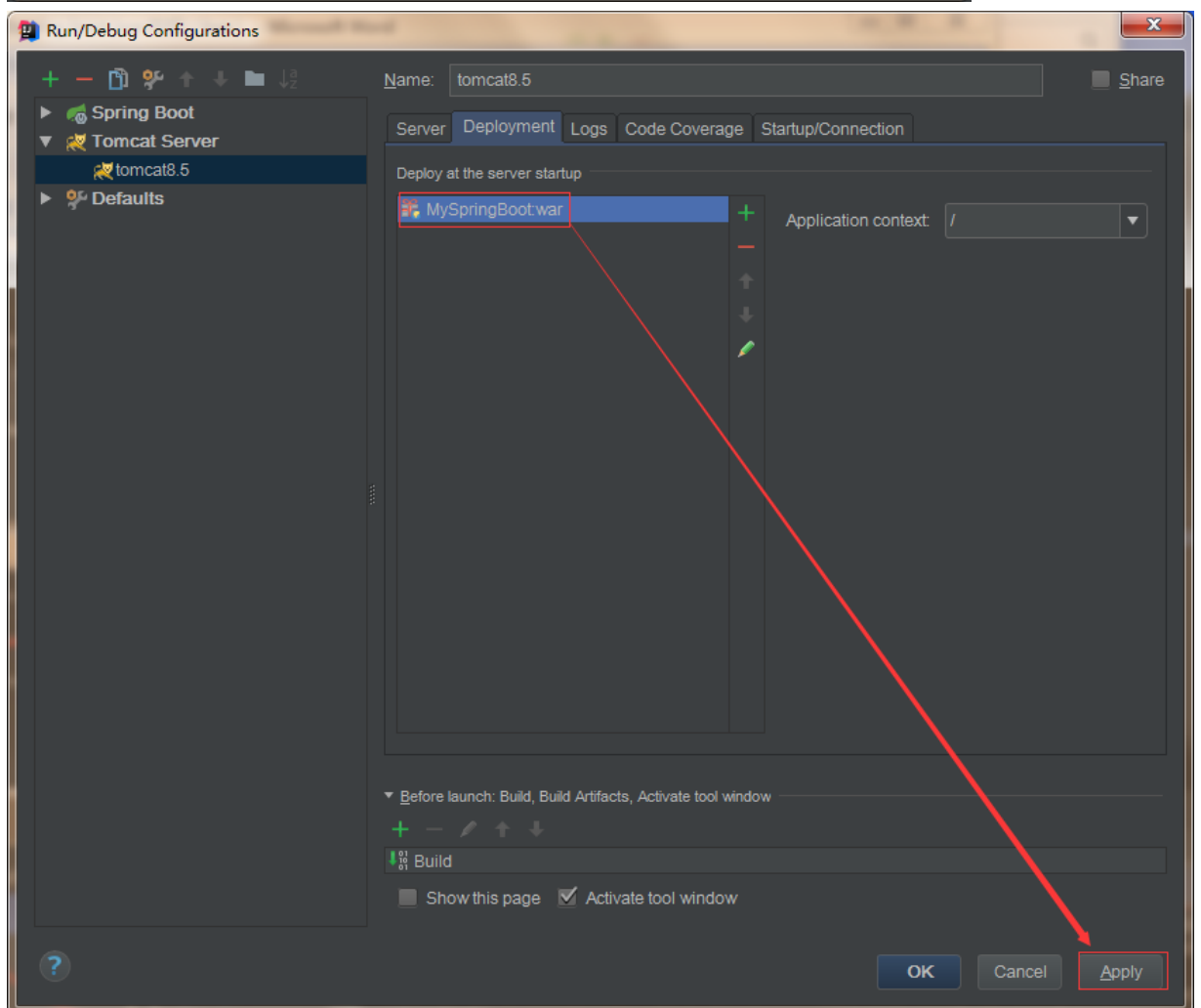
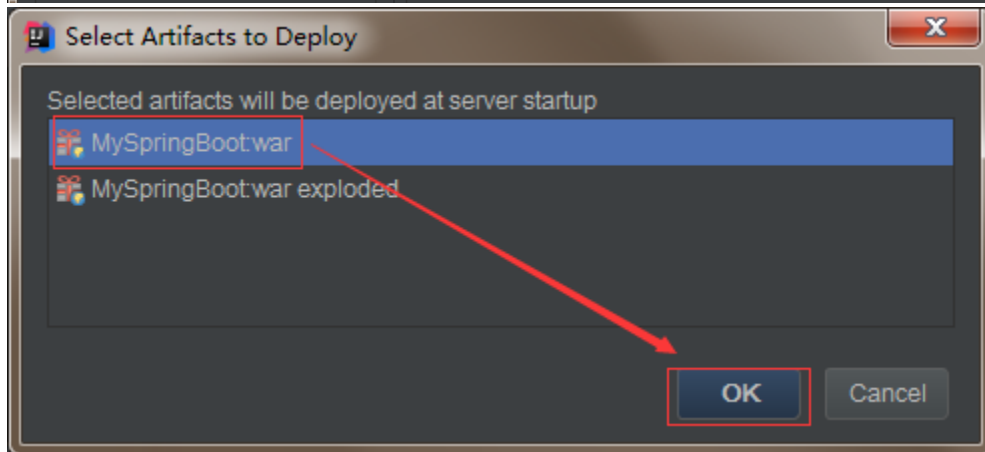
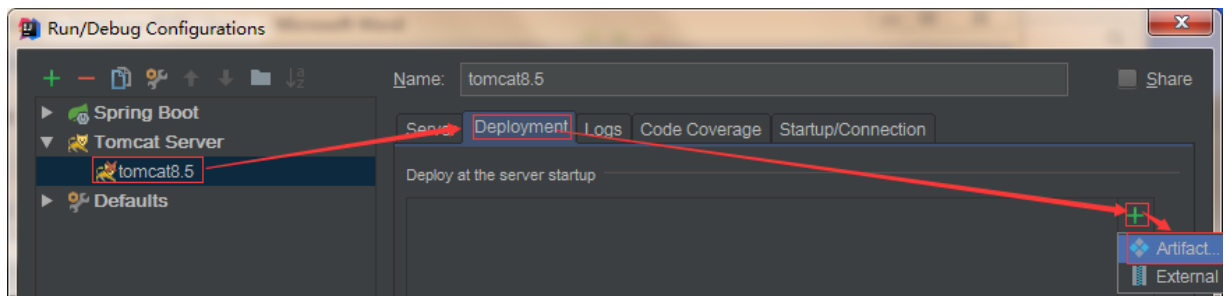
#3.2.1 添加 tomcat 配置

因为我们使用的是 war 包，所以不会使用 spring boot 自动带的 tomcat，这样我们主需要配置一个 tomcat，tomcat 我下载的是 apache-tomcat-8.5.20-windows-x64.zip，



点工具栏上

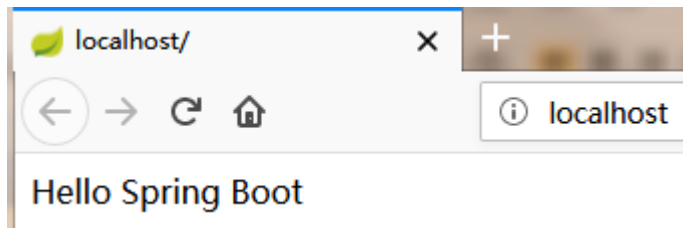




这样 tomcat 就配置成功了

#3.2.2 运行 tomcat 测试一下效果

点工具栏上 ，会自动弹出一个浏览器窗口，如下图：

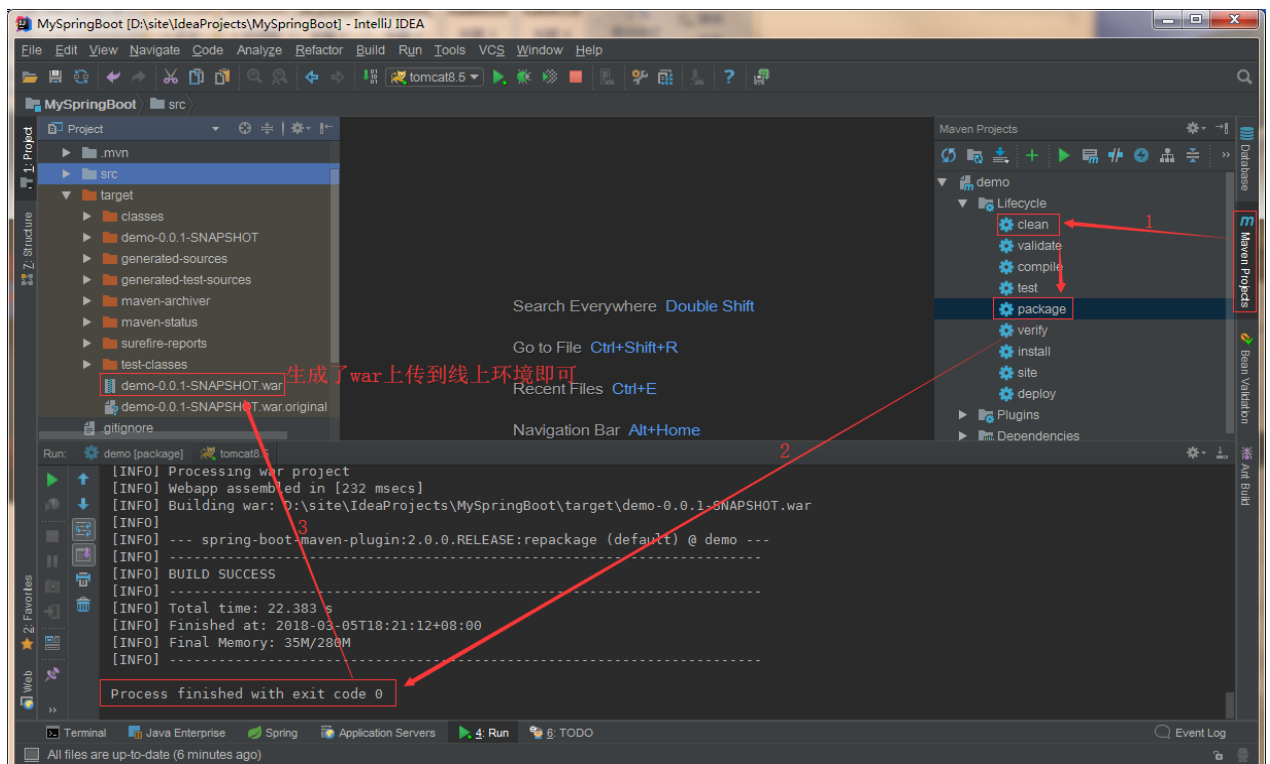
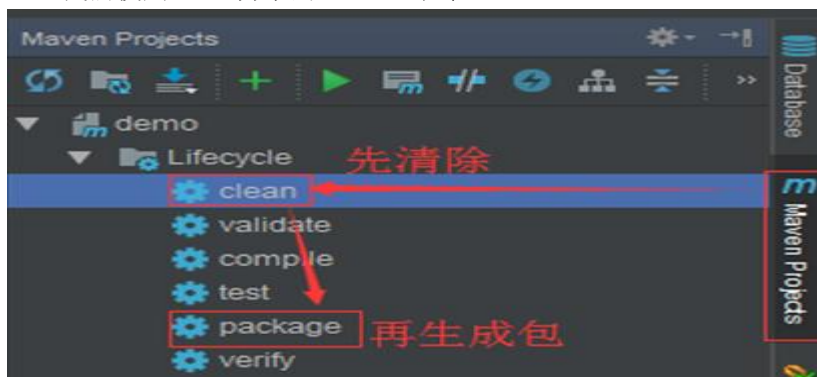


3.3 打包成 war 发送到生产环境的 tomcat 上

本地测试没有问题了，我们需要打包成 war 放到生产环境了，为了方便生产环境安装了 tomcat，其它默认

#3.3.1 idea 生成 war 包

这里我们使用 idea 自带的 maven 命令



#3.3.2 配置线上环境 tomcat 虚拟主机并访问

1) 上传 war 文件

我建立了个 tomcat 的网站目录: /disk1/www/hualinux.com

上传上图中 idea 的 “demo-0.0.1-SNAPSHOT.war” 文件并重命名为 demo.war

2) 配置 tomcat 虚拟主机

配置线上环境的 tomcat 虚拟主机, 这里为了方便我用 IP 访问, 没有配置域名

进入 tomcat 的 “conf/Catalina/localhost” 并建立一个名为 “springboot.xml”, 内容如下:

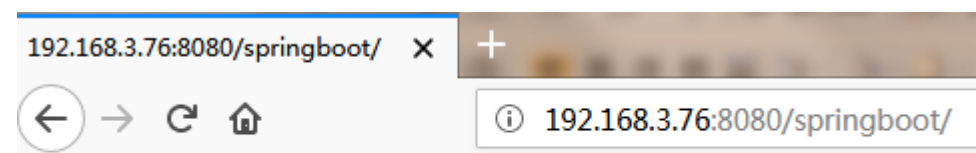
```
<!-- <?Xml version='1.0' encoding='utf-8'?> -->
<Context path="/springboot" docBase="/disk1/www/hualinux.com/demo.war" debug="0"
reloadable="true" privileged="true" >
</Context>
```

3) 测试

配置好线上 tomcat, 重启 tomcat 后, 现在访问测试一下效果

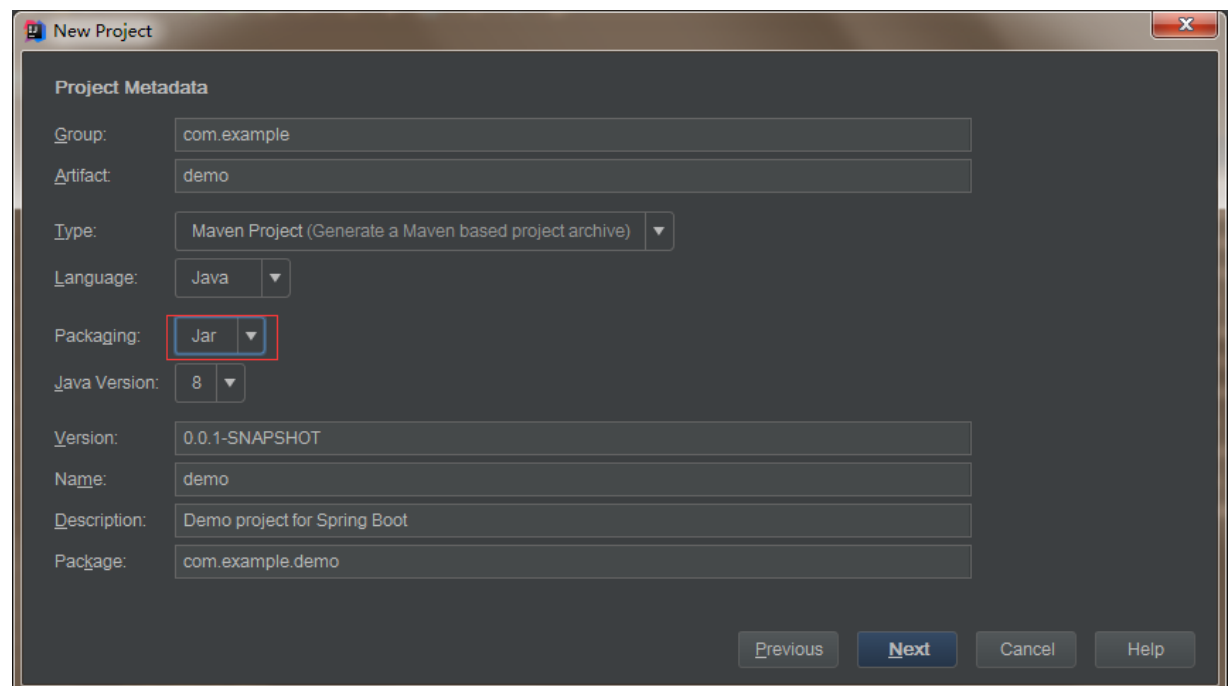
http://ip:8080/springboot, 我这里的 ip 地址为 192.168.3.76, 打开浏览器输入

http://192.168.3.76:8080/springboot/



四、jar 方式转 war 打包

如果一开始配置 spring Boot 使用的是 jar 包, 如下图:



则需要修改一个配置即可

4.1 修改打包形成

在 pom.xml 里设置 `<packaging>war</packaging>`

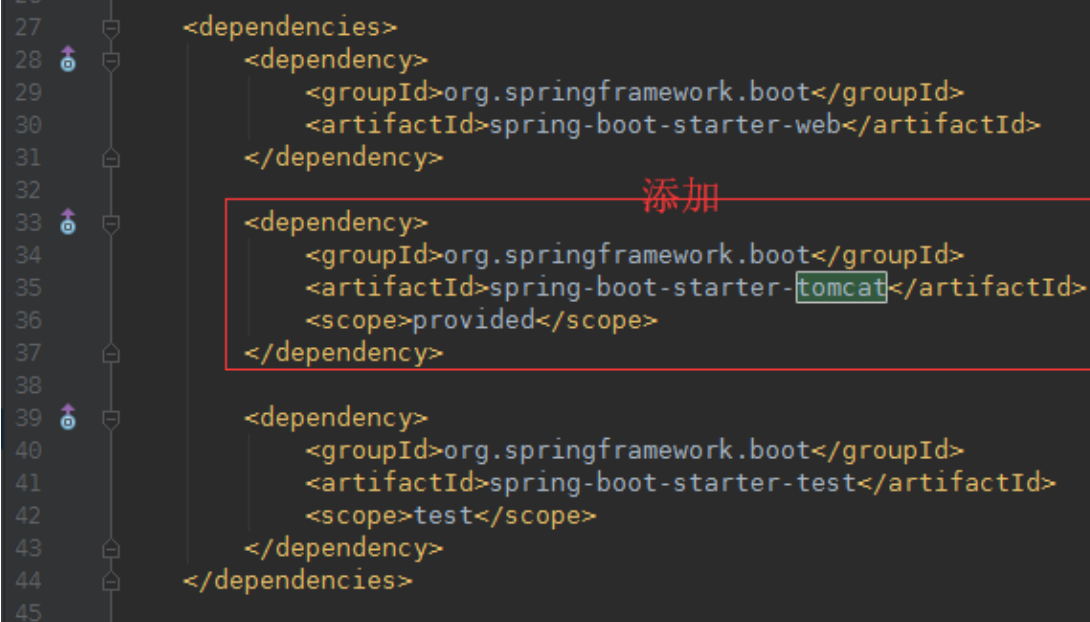
```
<groupId>com.hualinux</groupId>
<artifactId>demo</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
```

4.2 去掉内置 tomcat

在 pom.xml 里添加如下代码:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
  <scope>provided</scope>
</dependency>
```

如图:



```
27 <dependencies>
28   <dependency>
29     <groupId>org.springframework.boot</groupId>
30     <artifactId>spring-boot-starter-web</artifactId>
31   </dependency>
32   添加
33   <dependency>
34     <groupId>org.springframework.boot</groupId>
35     <artifactId>spring-boot-starter-tomcat</artifactId>
36     <scope>provided</scope>
37   </dependency>
38
39   <dependency>
40     <groupId>org.springframework.boot</groupId>
41     <artifactId>spring-boot-starter-test</artifactId>
42     <scope>test</scope>
43   </dependency>
44 </dependencies>
45
```

4.3 修改启动类，并重写初始化方法

在 DemoApplication 同级目录中添加 ServletInitializer.java 类，内容如下:

```
package com.hualinux.demo;

import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;

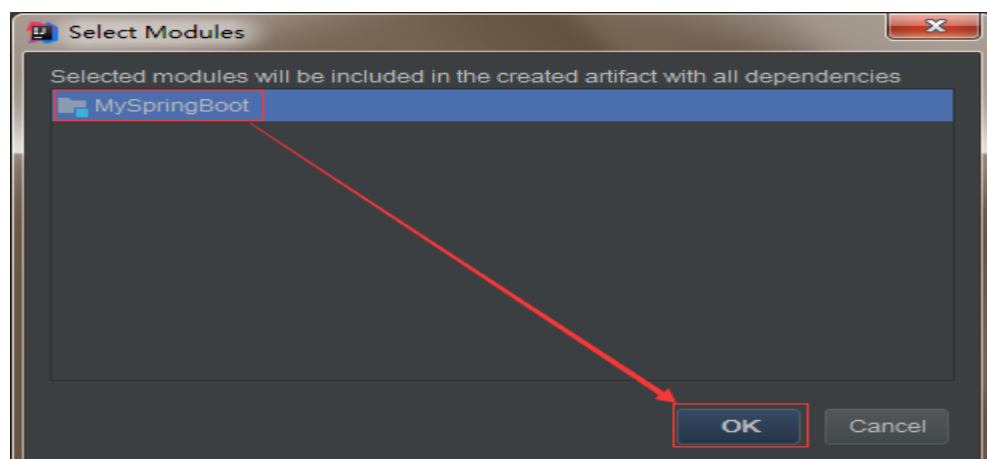
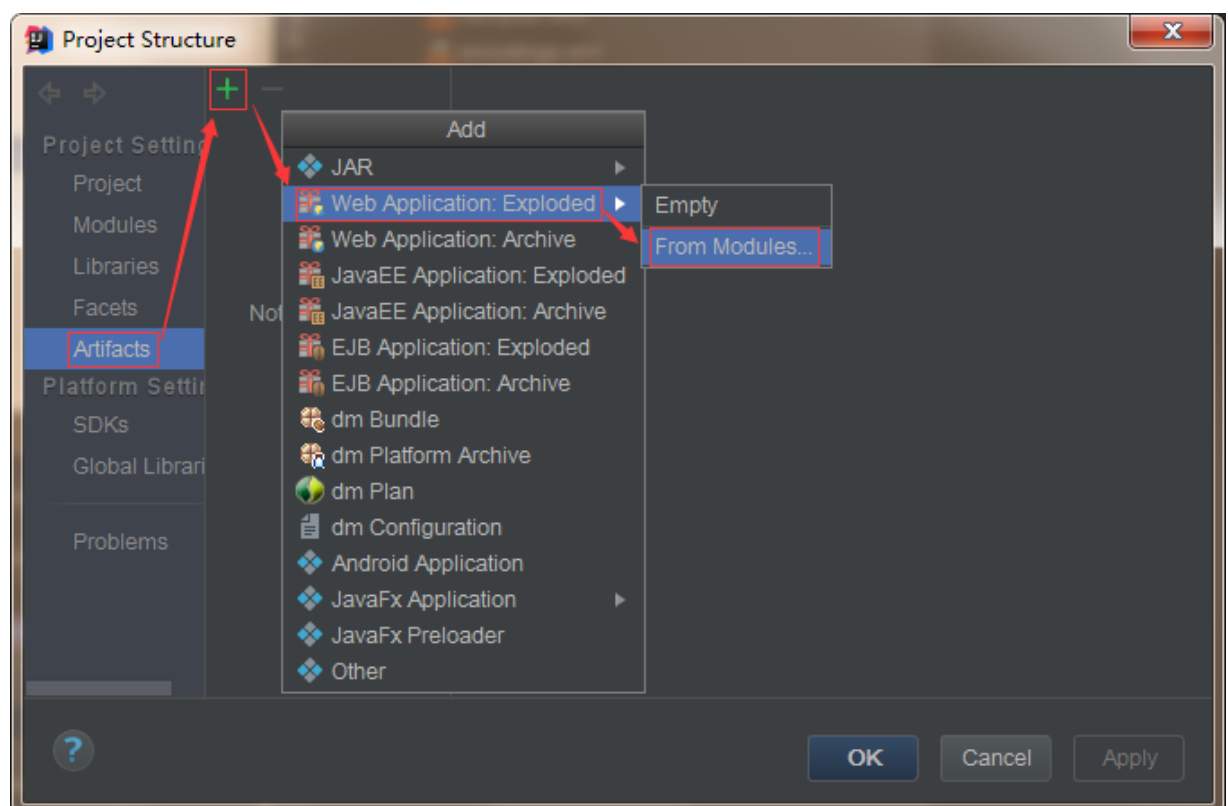
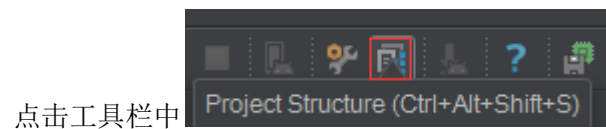
public class ServletInitializer extends SpringBootServletInitializer {
```

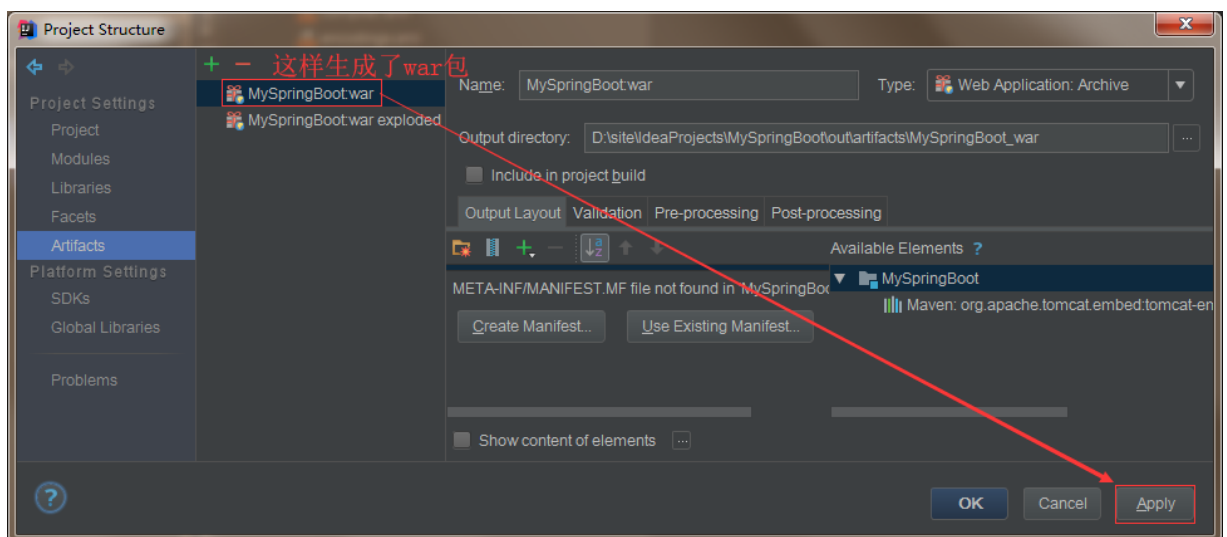
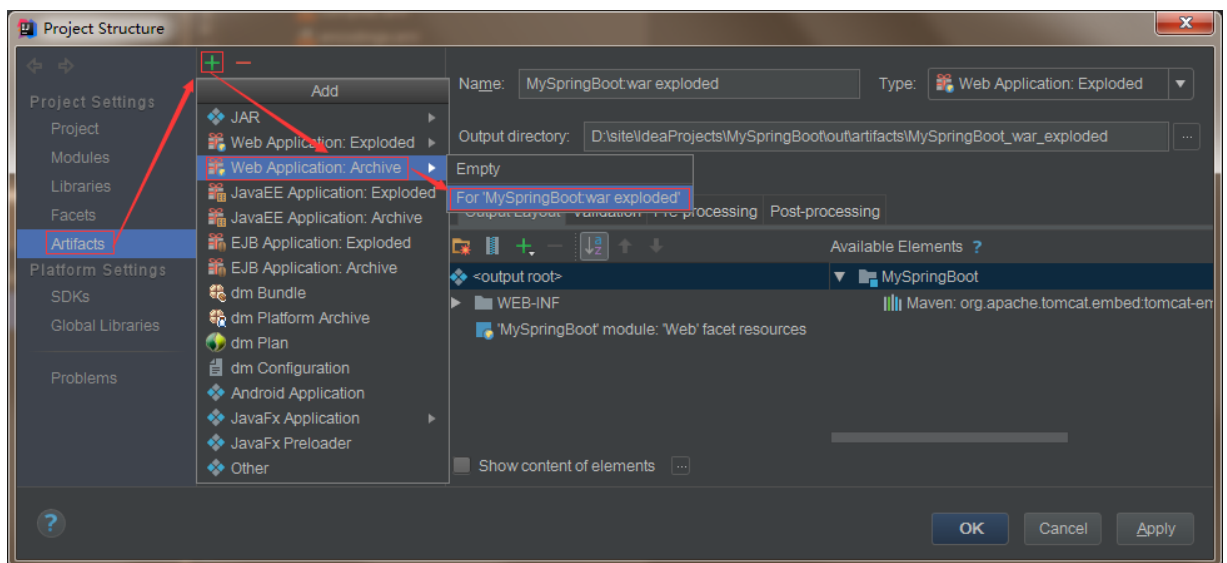
```

@Override
protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {
    return application.sources(DemoApplication.class);
}
}

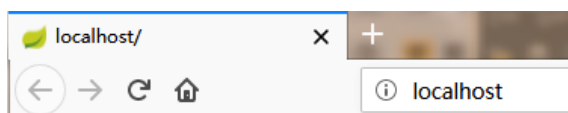
```

4.4 配置项目结构 Project structure



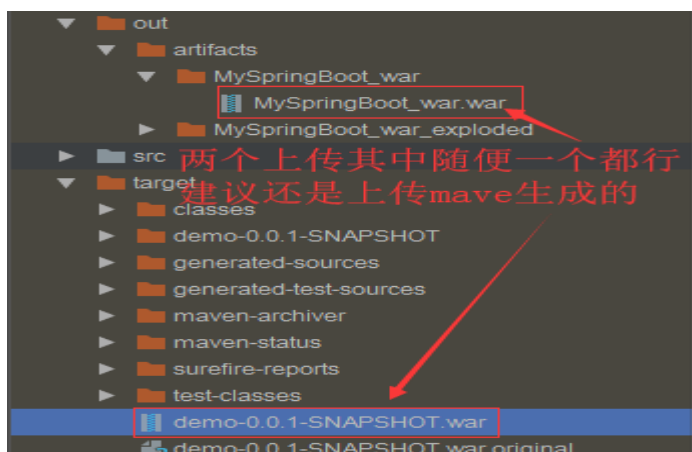


接下来就像“#3.2.1 添加 tomcat 配置”那样配置 tomcat 并运行看是否成功，如图所示则成功

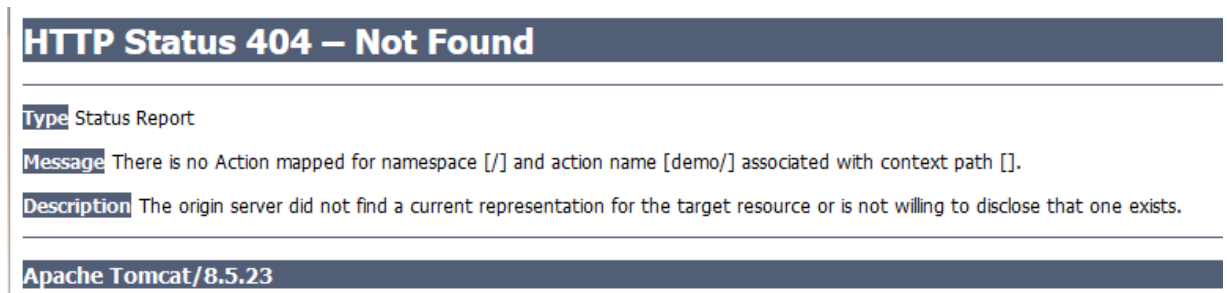


Hello Spring Boot

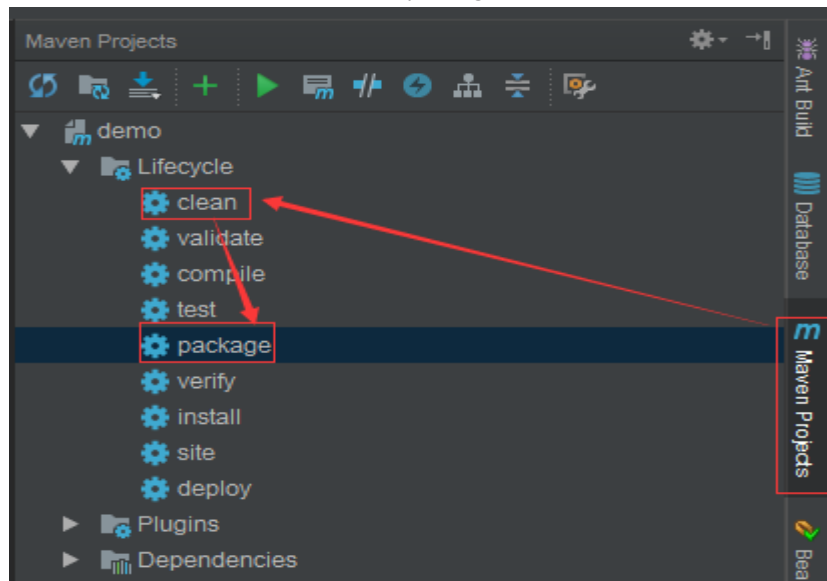
最后如果放到生产环境则按“3.3 打包成 war 发送到生产环境的 tomcat 上”操作即可，这里不赘述。



注，如果报下图错误



1.用 maven clean，再执行 maven package 再上传试下



2.tomcat 配置不正确，找不到

附录一、参考文章

IntelliJ IDEA Spring Boot（9）打成 war 包

<http://blog.csdn.net/u011998835/article/details/78389740>

spring boot 项目打包成 war 并在 tomcat 上运行的步骤

<http://blog.csdn.net/yalishadaa/article/details/70037846>

SpringBoot idea maven 打包 war 及运行 war 包

http://blog.csdn.net/qq_35498405/article/details/78054536

flying 飞翔

QQ: 715031064

广州 linux 运维: 478477301

linux 运维菜鸟: 313184229

2018.3.5