# Servlet容器

## 嵌入式Servlet容器

嵌入式Servlet容器：应用打成可执行的jar

优点：简单、便携；

缺点：默认不支持JSP、优化定制比较复杂（使用定制器【ServerProperties、自定义

EmbeddedServletContainerCustomizer】，自己编写嵌入式Servlet容器的创建工厂

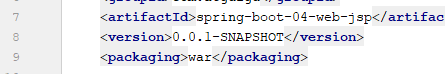
【EmbeddedServletContainerFactory】）；

## 外置的Servlet容器

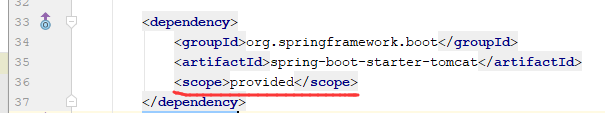
外置的Servlet容器：外面安装Tomcat---应用war包的方式打包；

# 步骤

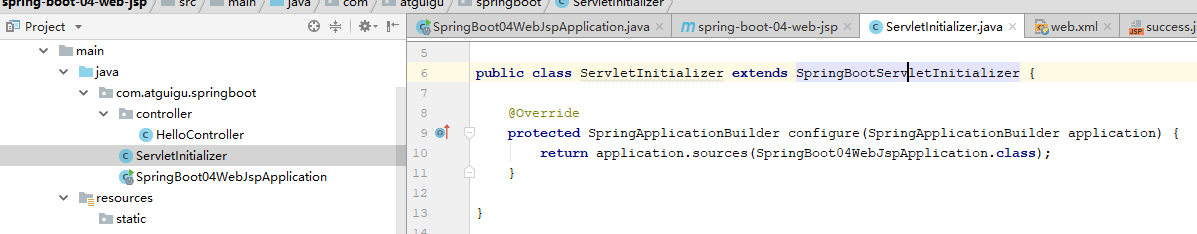
1）、必须创建一个war项目；（利用idea创建好目录结构）



2）、将嵌入式的Tomcat指定为provided；



3）、必须编写一个SpringBootServletInitializer的子类，并调用configure方法



4）、启动服务器就可以使用；

# 原理

## Jar包和war包

jar包：执行SpringBoot主类的main方法，启动ioc容器，创建嵌入式的Servlet容器；

war包：启动服务器，服务器启动SpringBoot应用【SpringBootServletInitializer】，启动ioc容器

## tomacat启动war包原理

### 规则

8.2.4 Shared libraries / runtimes pluggability：

1）、服务器启动（web应用启动）会创建当前web应用里面每一个jar包里面ServletContainerInitializer实例：

2）、ServletContainerInitializer的实现放在jar包的META-INF/services文件夹下，有一个名为

javax.servlet.ServletContainerInitializer的文件，内容就是ServletContainerInitializer的实现类的全类名

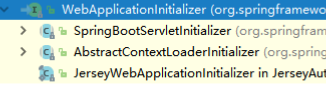
3）、还可以使用@HandlesTypes，在应用启动的时候加载我们感兴趣的类；

### 流程

1）、启动Tomcat

2）、org\springframework\spring-web\4.3.14.RELEASE\spring-web-4.3.14.RELEASE.jar!\META-INF\services\javax.servlet.ServletContainerInitializer：Spring的web模块里面有这个文件：org.springframework.web.SpringServletContainerInitializer

3）、SpringServletContainerInitializer将@HandlesTypes(WebApplicationInitializer.class)标注的所有这个类型

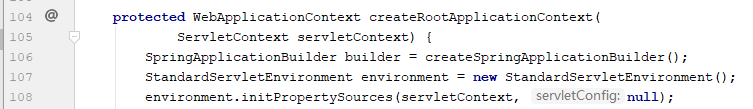


的类都传入到onStartup方法的Set>；为这些WebApplicationInitializer类型的类创建实例；

4）、每一个WebApplicationInitializer都调用自己的onStartup；

5）、相当于我们的SpringBootServletInitializer的类会被创建对象，并执行onStartup方法

6）、SpringBootServletInitializer实例执行onStartup的时候会createRootApplicationContext；创建容器



//1、创建SpringApplicationBuilder



 //调用configure方法，子类重写了这个方法，将SpringBoot的主程序类传入了进来

//使用builder创建一个Spring应用



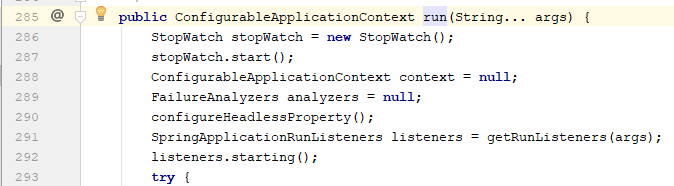
 //启动Spring应用



7）、Spring的应用就启动并且创建IOC容器



  //刷新IOC容器



启动Servlet容器，再启动SpringBoot应用