# SpringBoot的启动方式

内置的Tomcat和外部的Tomcat，

## 使用内置的Tomcat

启动方式：

1. main方法直接启动
2. mvn springboot-run
3. java -jar XXX.jar

配置内置Tomcat属性：

关于Tomcat的属性在org.springframework.boot.autoconfigure.web.ServerProperties配置类中做了定义，我们只需要在application.properties配置属性即可，通用的Servlet容器配置搜以”server”作为前缀，而Tomcat特有配置都以”server.tomcat”作为前缀。

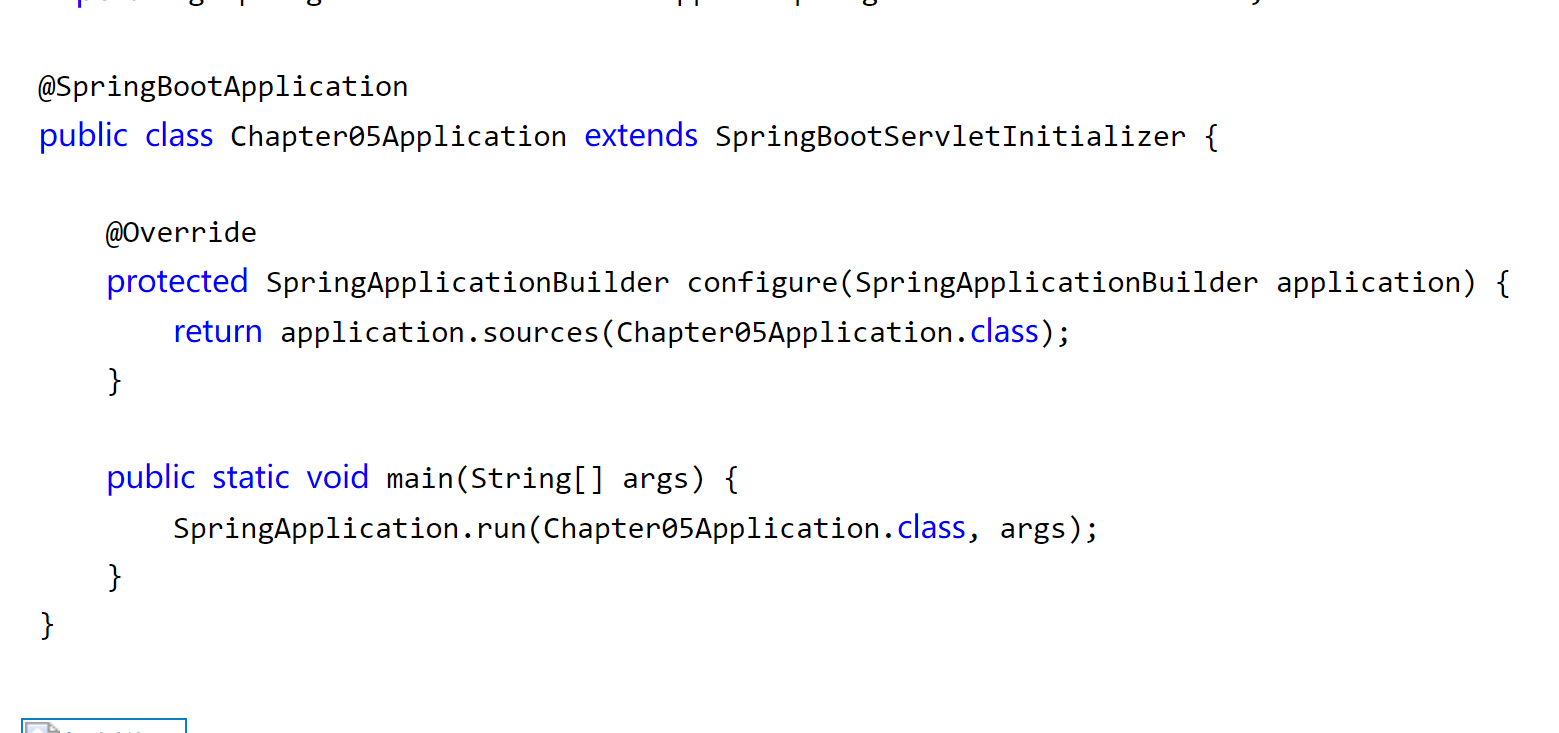
server.port = 8080

server.session.timeout = 10秒做单位

server.tomcat.uri-encoding = UTF-8

## 使用外部Tomcat

就不能依赖Application的main函数了，而是要以类似于web.xml文件配置的方式来启动Spring应用上下文。此时需要在启动类中继承SpringBootServletInitializer并实现configure方法。



pom.xml修改tomcat的相关配置

这里又两种方式解决

方法一

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<exclusions>

<exclusion>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

</exclusion>

</exclusions>

</dependency>

方法二

部署为war包时使用

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<scope>provided</scope>

</dependency>

# Springboot优点

是一个简化spring开发的框架，简单说，springboot就是整合了很多优秀的框架，不用我们去写一堆xml配置。

1. 快速创建独立运行的Spring项目以及与主流框架集成。
2. 使用嵌入式的Servlet容器，应用无需打成war包。
3. Starters自动依赖与版本控制。
4. 大量的自动配置，简化开发，也可以修改默认值。
5. 使用springboot可以生成独立的微服务功能单元。