# Spring Boot核心-基本配置

## 入口类和＠SpringBootApplication

Spring Boot 通常有一个名为XxxApplication 的入口类，入口类里有一个main 方法，这个main方法其实就是一个标准的Java 应用入口方法。在main 方法中使用SpringApplication.run( Xxx.class, args ），启动Spring Boot 应用项目。

@SpringBootApplication 是Spring Boot 的核心注解，它是一个组合注解。

|  |
| --- |
| @Target(ElementType.***TYPE***) @Retention(RetentionPolicy.***RUNTIME***) @Documented @Inherited @SpringBootConfiguration @EnableAutoConfiguration @ComponentScan(excludeFilters = {  @Filter(type = FilterType.***CUSTOM***, classes = TypeExcludeFilter.**class**),  @Filter(type = FilterType.***CUSTOM***, classes = AutoConfigurationExcludeFilter.**class**) }) **public** @**interface** SpringBootApplication {…} |

@EnableAutoConfiguration 让Spring Boot 根据类路径中的jar 包依赖为当前项目进行自动配置。Spring Boot 会自动扫描＠SpringBootApplication 所在类的同级包。

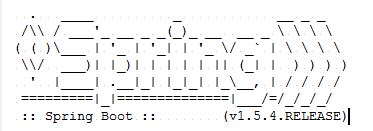
## 关闭特定的自动配置

通过上面的@SpringBootApplication 的源码我们可以看出，关闭特定的自动配置应该使用@SpringBootApplication 注解的exclude 参数。

|  |
| --- |
| @SpringBootApplication{exclude={Xxxx.class}} |

## 定制Banner

在Spring Boot 启动的时候会有一个默认启动图案



在src/main/resources 下新建一个banner.txt 。

通过http://patorjk.com/software/taag 网站生成字符，如敲入的为“ SUPERMAN ”，将网站生成的字符复制到banner.txt 中。

再启动程序，会使用新的图案。

关闭banner，使用SpringApplication.setShowBanner(false)

## Spring Boot 的配置文件

Spring Boot 使用一个全局的自己置文件application.properties 或application.yml ，放置在src/main/resources 目录或者类路径的/config 下。Spring Boot 不仅支持常规的properties 配置文件，还支持yaml语言的配置文件。Yaml是以数据为中心的语言，在自己置数据的时候具有面向对象的特征。Spring Boot 的全局配置文件的作用是对一些默认配置的配置值进行修改。

## 命令行参数配置

Spring Boot 可以是基于jar 包运行的，打成jar 包的程序可以直接通过下面命令运行：java -jar xx.jar，可以通过以下命令修改Tomcat 端口号：java -jar xx.jar –server.port=9090

## 常规属性配置

在Spring Boot里，我们只需在application.properties定义属性，直接使用@Value注入即可。

## 类型安全的配置

Spring Boot 提供了基于类型安全的配置方式，通过@ConfigurationProperties 将properties属性和一个Bean 及其属性关联，从而实现类型安全的配置。

## 日志配置

Spring Boot 文恃Java Util Logging 、Log4J 、Log4刀和Logback 作为日志框架，无论使用哪种日志框架， Spring Boot 己为当前使用日志、框架的控制台输出及文件输出做好了配置，Spring Boot 使用Logback 作为日志框架。