# .代码

@SpringBootApplication  
**public class** Main {  
 **public static void** main(String[] args) {  
 SpringApplication.*run*(Main.**class**,args);  
 }  
}

# 注解SpringBootApplication

## 简介

翻译过来，就是springBoot应用，标注在某个类上说明这个类是SpringBootApplication的主配置类，SpringBoot就应该运行在这个类的main方法来启动SpringBoot应用；

## 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 {

## @SpringBootConfiguration

标注在某个类上，表示这是一个SpringBoot的配置类

这个配置又被@Configuration注解标注

**@Configuration**

此注解是Spring底层的注解，配置类上来标注这个注解；

配置类🡪配置文件

代替了配置文件

这个注解又被@Component标注，其实就是一个Spring的组件

## @EnableAutoConfiguration

开启自动配置功能，以前我们需要配置的东西，我们都不需要手动配置了，Spring Boot帮我们自动配置，@EnableAutoConfiguration告诉SpringBoot开启自动配置功能，这样自动配置才能生效。

也是一个组合注解

@Target({ElementType.TYPE})  
@Retention(RetentionPolicy.RUNTIME)  
@Documented  
@Inherited  
@AutoConfigurationPackage  
@Import({AutoConfigurationImportSelector.**class**})  
**public** @**interface** EnableAutoConfiguration {

### @AutoConfigurationPackage

自动配置包

@Import({Registrar.**class**})这个注解实现了这个功能

此注解的作用是给Spring容器中导入组件

Registrar类中定义了自动扫描包的规则，此类获取了被注解标识类的元数据，获取此类的包名，然后自动扫描此类包及子包下的所有类。

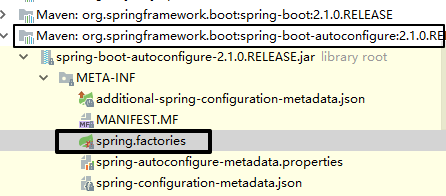
### @Import({AutoConfigurationImportSelector.class})

给容器中添加组件

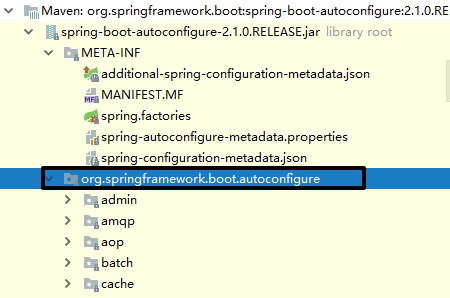
AutoConfigurationImportSelector里确定了具体导入哪些组件，它会将所有需要导入的组件以全类名的方式返回，这些组件就会被添加到容器中。

在底层有个list来存放组件的全类名，在list中有96个组件被导入（可能会更多），组件都以AutoConfiguration结尾，什么什么自动配置，有aop，admin，amqp，cache等等等等，会给容器中导入非常多的自动配置类，就是给容器中导入这场景所需要的全部组件，并自动配置好。

#### 自动配置组件所在



SpringBoot在启动时，从类路径下的META-INF/spring.factories中获取EnableConfiguration指定的值，将这些值作为自动配置类导入到容器中，自动配置类就生效，帮我们进行自动配置工作。



所有的自动配置都在这个文件夹中，以后我们不满意，可以来手动更改