

18.使用@SpringBootApplication注解

许多Spring Boot开发者想让他们的应用使用自动配置、组件扫描，并且能够在应用类上定义自己的额外配置。一个简单的@SpringBootApplication注解能够去启动这些特点：

- @EnableAutoConfiguration:启动Spring Boot的自动配置机制。
- @ComponentScan:启动@Component扫描。
- @Configuration:允许在上下文中注册额外的bean或者导入额外的配置类。

@SpringBootApplication注解相当于同时使用@Configuration、@EnableAutoConfiguration和默认属性的@ComponentScan注解：

```
package com.example.myproject;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
//等同于@Configuration@EnableAutoConfiguration@ComponentScan
public class Application
{
    public static void main(String[] args)
    {
        SpringApplication.run(Application.class,args);
    }
}
```

@SpringBootApplication也提供了别名去自定义@EnableAutoConfiguration和@ComponentScan的属性。

这些功能都不是强制性的，您可以选择使用它启用的任何功能替换此单个注释。例如您可能不想使用组件扫描：

```
package com.example.myproject;

import org.springframework.boot.SpringApplication;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Import;

@Configuration
@EnableAutoConfiguration
@Import({MyConfig.class,MyAnotherConfig.class})
public class Application
{
    public static void main(String[] args){
        SpringApplication.run(Application.class,args);
    }
}
```

在这个例子中，`Application`和其他的Spring Boot应用一样，除了被`@Component`注解的类不会被自动扫描，并且用户定义的bean要被自己导入。