

【SpringBoot】 annotation

以下是常用的 Spring Boot Annotation 的說明

- `@SpringBootApplication`：用於標記Spring Boot應用程式的主要類，它結合了`@Configuration`、`@EnableAutoConfiguration`、`@ComponentScan`三個註解的功能。
- `@Controller`：用來標記一個類，表示這是一個Controller，可以處理HTTP請求。
- `@RestController`：用於標記一個控制器類，並表示該類中所有的方法都會以JSON格式返回結果。
- `@RequestMapping`：使用標記Controller中的方法，並指定該方法的URL路徑。
 - `@GetMapping`：類似於`@RequestMapping`，但只是處理HTTP GET請求。
 - `@PostMapping`：類似於`@RequestMapping`，但只是處理HTTP POST請求。
 - `@PutMapping`：類似於`@RequestMapping`，但只是處理HTTP PUT請求。
 - `@DeleteMapping`：類似於`@RequestMapping`，但只是處理HTTP DELETE請求。
- `@PathVariable`：用來獲取URL路徑上的變量值。
- `@RequestParam`：用來獲取HTTP請求中的參數值。
- `@RequestBody`：用於獲取HTTP請求中的請求體，一般用於處理POST請求。
- `@ResponseBody`：使用指定方法返回值的格式，可以是JSON、XML或者其他格式。
- `@Autowired`：用於將依賴註冊到Spring容器中，可以在成員變量、構造函數、Setter方法中使用。
- `@Value`：用於獲取配置文件中的值，例如數據庫連接信息、服務端接口等。
- `@Configuration`：用於標記一個配置類，配置類中通常包含`@Bean`註解的方法，用於提供Bean的定義。
- `@Bean`：用於將方法返回的對象註冊為Bean，Spring容器管理該Bean的生命週期。
- `@EnableAutoConfiguration`：用於啟動Spring Boot的自動配置功能，Spring Boot會根據當前的classpath和配置文件，自動配置所需的Bean。
- `@ComponentScan`：用於指定需要掃描的包，Spring Boot會掃描該包及其子包下面所有的`@Component`註解，並將其註冊為Bean。
- `@Transactional`：用於標記一個方法或類需要事務管理，Spring Boot會根據`@Transactional`註解的配置，自動創建事務，並管理其生命週期。
- `@Profile`：使用指定Bean的配置文件的Profile，不同的Profile可以在不同的環境下使用不同的配置。
- `@Async`：用於標記一個方法為異常方法，Spring Boot會自動將方法放入線程池中執行。

@SpringBootApplication

掛有@SpringBootApplication

main程式為執行時的**進入點**。同時也是主要的配置類別。

會掃描所在類別的package及其子package中掛有

@Component

@Controller

@Service

@Repository

等component類別並註冊為spring bean。

包含三種 annotation

@EnableAutoConfiguration

@ComponentScan

@Configuration