

# Spring Annotation 정리

## @Component

개발자가 직접 작성한 Class를 Bean으로 등록하기 위한 어노테이션

## @ComponentScan

@Component, @Service, @Repository, @Controller, @Configuration이 붙은 빈들을 찾아서 Application Context에 bean을 등록해 주는 어노테이션

## @Autowired

필드, setter 메소드, 생성자에 사용하며 객체에 대한 의존성을 주입해주는 어노테이션

## @RestController

Spring에서 Controller 중 View로 응답하지 않는 Controller를 의미하며 method의 반환 결과를 JSON 형태로 반환. `@RestController`가 적혀있는 Controller의 method는 `HttpResponse`로 바로 응답이 가능하며 `@ResponseBody` 역할을 자동적으로 해주는 어노테이션

## @Service

Service 모듈에서 쓰이는 어노테이션으로, `비즈니스 로직`을 수행하는 Class라는 것을 나타내는 용도

## @Repository

DAO class에서 쓰이는 어노테이션. `DB`에 접근하는 method를 가지고 있는 Class에서 쓰임

## @GetMapping

HTTP method 중 GET 메서드에 Controller의 메서드를 매핑시켜주는 어노테이션.

## @PostMapping

HTTP method 중 POST 메서드에 Controller의 메서드를 매핑시켜주는 어노테이션.

## @PutMapping

HTTP method 중 PUT 메서드에 Controller의 메서드를 매핑시켜주는 어노테이션.

## @DeleteMapping

HTTP method 중 DELETE 메서드에 Controller의 메서드를 매핑시켜주는 어노테이션.

## @PathVariable 과 @RequestParam

method 파라미터 앞에 사용하면서 해당 URL에서 특정값들을 변수로 받아올 수 있음

- **@RequestParam**: 요청의 매개변수가 URL 경로에 포함되지 않고, **?**를 사용하여 쿼리 문자열로 전달됨. 예를 들어, `/hello?name=John&age=30` 과 같은 요청에서 `name` 과 `age` 라는 매개변수 값을 받아옴
- **@PathVariable**: URL 경로의 일부로 매개변수가 전달되며, 경로 변수는 `{}` 로 표시됨. 예를 들어, `/user/{id}` 와 같은 요청에서 `{id}` 라는 경로 변수 값을 받아옴