

# 빈 등록을 위한 어노테이션

- @Bean
- @Configuration
- @Component

## Spring Bean 이란?

- Spring에서는 Spring의 DI Container에 의해 관리되는 POJO(Plain Old Java Object)를 Bean이라고 부르며, 이러한 Bean들은 Spring을 구성하는 핵심 요소이다. Spring의 Bean을 정리하면 아래와 같다.
  - POJO(Plain Old Java Object)로써 Spring 애플리케이션을 구성하는 핵심 객체이다.
  - Spring IoC 컨테이너(또는 DI 컨테이너)에 의해 생성 및 관리된다.
  - class, id, scope, constructor-arg 등을 주요 속성으로 지닌다.

## Spring Bean 구성 요소

- class: Bean으로 등록할 Java 클래스
- id: Bean의 고유 식별자
- scope: Bean을 생성하기 위한 방법(singleton, prototype 등)
- constructor-arg: Bean 생성 시 생성자에 전달할 파라미터
- property: Bean 생성 시 setter에 전달할 인수

## SpringBean 등록 방법(@Bean, @Configuration, @Component)

### Bean 어노테이션과 Configuration 어노테이션

```
public class MangKyuResource {  
  
}
```

- @Bean으로 수동으로 빈을 등록 해줄 때 빈 이름은 메소드 이름으로 결정, 중복이름이 존재해선 안된다.

```
@Configuration  
public class ResourceConfig {  
  
    @Bean  
    public MangKyuResource mangKyuResource() {  
        return new MangKyuResource();  
    }  
  
}
```

## 등록 과정

1. 스프링 컨테이너는 @Configuration이 붙은 클래스를 자동으로 빈으로 등록
2. 해당 클래스를 파싱해 @Bean 메소드를 찾아 빈을 생성
3. @Configuration 안에서 @Bean을 사용해야 싱글톤을 보장

#### 수동으로 Bean을 등록해줘야 하는 상황

1. 개발자가 직접 제어가 불가능한 라이브러리를 활용 시
2. 애플리케이션 전범위적으로 사용되는 클래스를 등록 할 시
3. 다형성을 활용해 여러 구현체를 등록할 시

#### 예시

1. 개발자가 직접 제어가 불가능한 라이브러리를 활용 시
  - 외부 라이브러리 사용시, 해당 클래스를 싱글톤으로 사용해야 메모리상의 이점을 얻음
  - 그러나 직접 만든게 아니라, 가져다 쓰는 클래스이므로 @Bean으로 등록해줘야함
2. 애플리케이션 전범위적으로 사용되는 클래스와, 다형성을 활용해 여러 구현체를 등록할 시
  - 빈으로 등록해줘야 한눈에 파악하여 유지보수가 좋음
  - @Configuration 클래스만 보면 되기 때문

## @Component 어노테이션

직접 빈으로 등록하는 경우, 빈 등록 클래스가 늘어날 수록 많은 시간의 소요

- @Component 어노테이션이 있으면, @ComponentScan을 통해 자동으로 빈으로 등록해줌

```
public class MangKyuResource {  
  
}
```

•