

초격차 패키지 Online.

# Boot의 기본기

## PART1 | Spring Boot Properties

이걸 안 쓸거면 boot 쓰지 마세요

## PART2 | @SpringBootApplication

스프링 부트의 시작점

## PART3 | @Component vs. @Configuration

둘의 차이가 뭐야?

## PART4 | Configuration Properties

프로퍼티를 설정하는 우아한 방법

# Boot의 기본기

## 4 Configuration Properties

## Externalized Configuration

### 4. Configuration Properties

### 각종 설정값을 외부로 분리해낸 것

- 서로 다른 환경에서도 사용할 수 있음
- 애플리케이션을 새로 컴파일하지 않고 설정값을 바꿀 수 있음
- 종류
  - Java properties file
  - YAML
  - environment variable
  - command-line argument

## 외부 설정의 우선 순위

## 4. Configuration Properties

외부 설정을 읽어들이는 순서: 아래 설정이 위에서 읽은 것을 덮어씀

- 1.디폴트 프로퍼티
- 2.@Configuration 클래스에 @PropertySource 로 정의된 것
- 3.설정 파일: application.properties
- 4.RandomValuePropertySource
- 5.OS 환경변수
- 6.자바 시스템 프로퍼티: System.getProperties()
- 7.JNDI 속성: java:comp/env
- 8.ServletContext - 초기 파라미터
- 9.ServletConfig - 초기 파라미터
- 10.SPRING\_APPLICATION\_JSON 안의 프로퍼티들
- 11.Command-line arguments
- 12.테스트에 사용된 프로퍼티들
- 13.@TestPropertySource
- 14.Devtools 글로벌 세팅: \$HOME/.config/spring-boot

## 설정 파일(Config data)의 우선 순위

### 4. Configuration Properties

- 1.JAR 패키지 안의 application.properties, application.yaml
- 2.JAR 패키지 안의, 프로파일이 지정된 파일: application-{profile}.properties
- 3.JAR 패키지 밖의 파일
- 4.JAR 패키지 밖의, 프로파일이 지정된 파일

## 설정 파일(Config data) 의 위치

### 4. Configuration Properties

1.classpath

1.1.classpath:/

1.2.classpath:/config

2.현재 디렉토리

2.1../

2.2../config

2.3../config/child

## 설정 파일(Config data) 을 읽는 방법

### 4. Configuration Properties

- @value
- Environment
- @ConfigurationProperties

## 1. @value

- SpEL 로 프로퍼티명을 표현
- type-safe 하지 않음
- 인스턴스화 이후에 주입하므로, final 쓸 수 없음
- 생성자 안에서 보이지 않음 (대안: @PostConstruct)
- Relaxed binding 지원 (kebab-case only)
- meta-data 없음 (javadoc은 적용 가능)

```
@Component
public class MyBean {

    @Value("${name}")
    private String name;

    // ...

}
```



## 2. Environment

## 4. Configuration Properties

- 애플리케이션 컨텍스트에서 꺼내오는 방법
- Environment 빈을 가져오는 방법
- 눈에 잘 안 들어옴

**생략 가능 (as of Spring Framework 4.3)**

```
@Autowired
public SortService(
    @Qualifier("bubbleSort") Sort<String> sort,
    Environment environment,
    ApplicationContext app,
    CustomProperties customProperties
) {
    System.out.println("Env - iam.duration: " + environment.getProperty("iam.duration"));
    System.out.println("app.Env - iam.duration: " + app.getEnvironment().getProperty("iam.duration"));
    System.out.println("conf.prop: " + customProperties.getDuration().toMillis() + " ms");
}
```

### 3. Configuration Properties

## 4. Configuration Properties

- 자바 클래스로 매핑하므로 type-safe
- 각 프로퍼티에 대응하는 meta-data 작성 가능
- Relaxed binding 지원
- 작성 방법
  - 기본
  - @Configuration 생략
  - @Bean 메소드
  - @ConstructorBinding

### 3. Configuration Properties - 기본

## 4. Configuration Properties

```
@ConfigurationProperties("iam")
@Configuration
public class CustomProperties {

    private Duration duration;

    public Duration getDuration() {
        return duration;
    }

    public void setDuration(Duration duration) {
        this.duration = duration;
    }
}
```

### 3. Configuration Properties - @Configuration 생략

## 4. Configuration Properties

```
@ConfigurationProperties("iam")
public class CustomProperties {

    private Duration duration;

    public Duration getDuration() {
        return duration;
    }

    public void setDuration(Duration duration) {
        this.duration = duration;
    }

}
```

```
@ConfigurationPropertiesScan
@SpringBootApplication
public class FastcampusSpringBootPracticeApplication {
```

### 3. Configuration Properties - @Bean 메소드

## 4. Configuration Properties

```
@Configuration
public class Config {

    @Bean
    @ConfigurationProperties("iam")
    public CustomProperties customProperties() {
        return new CustomProperties();
    }
}
```

### 3. Configuration Properties - @ConstructorBinding

## 4. Configuration Properties

- Immutable한 프로퍼티를 구현할 수 있는 방법 - 추천

```
@ConstructorBinding
@ConfigurationProperties("iam")
public class CustomProperties {

    /**
     * 와아 이건 configuration processor 테스트
     */
    private final Duration duration;

    public CustomProperties(@DefaultValue("1") Duration duration) {
        this.duration = duration;
    }

    public Duration getDuration() {
        return duration;
    }
}
```

## 결론

## 4. Configuration Properties

- @ConfigurationProperties 를 이용해서
- 상수처럼, immutable 하고 type-safe 하고 명확한 프로퍼티를 만들어 사용하자