

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

2 @SpringBootApplication

@SpringBootApplication

2. @SpringBoot Application

- 스프링 부트 애플리케이션의 시작점

```
@SpringBootApplication
public class FastcampusSpringBootApplication {

    public static void main(String[] args) {
        SpringApplication.run(FastcampusSpringBootApplication.class, args);
    }

}
```

@SpringBootApplication

2. @SpringBoot Application

- 스프링 부트 애플리케이션의 시작점

```
@SpringBootApplication
@EnableAutoConfiguration
@ComponentScan(excludeFilters = { @ComponentScan.Filter(type = FilterType.CUSTOM, classes = TypeExcludeFilter.class),
    @ComponentScan.Filter(type = FilterType.CUSTOM, classes = AutoConfigurationExcludeFilter.class) })
public class FastcampusSpringBootApplication {

    public static void main(String[] args) {
        SpringApplication.run(FastcampusSpringBootApplication.class, args);
    }
}
```

@SpringBootApplication

2. @SpringBoot Application

- @SpringBootConfiguration: 스프링 부트용 @Configuration
- @EnableAutoConfiguration: 사전에 정의한 라이브러리 빈을 등록
- @ComponentScan: 각종 스프링 빈 애노테이션을 베이스 패키지에서부터 스캔하여 스프링 빈으로 스프링 IoC 컨테이너에 등록

@SpringBootApplication 의 속성(attribute)들

2. @SpringBoot Application

- exclude
- excludeName
- scanBasePackages
- scanBasePackageClasses
- nameGenerator
- proxyBeanMethods

@SpringBootApplication 의 속성(attribute)들

2. @SpringBoot Application

- 나를 스프링의 마스터라 불러라!...

```
@SpringBootApplication(  
    exclude = {WebMvcAutoConfiguration.class},  
    excludeName = {"org.springframework.boot.autoconfigure.web.servlet.WebMvcAutoConfiguration"},  
    scanBasePackages = {"com.fastcampus.springbootpractice"},  
    scanBasePackageClasses = {MainController.class},  
    nameGenerator = BeanNameGenerator.class,  
    proxyBeanMethods = true  
)  
  
public class FastcampusSpringBootPracticeApplication {  
  
    public static void main(String[] args) {  
        SpringApplication.run(FastcampusSpringBootPracticeApplication.class, args);  
    }  
  
}
```

부록 1 - Spring Application 에 고급 설정 추가하기

2. @SpringBoot Application

- SpringApplication 인스턴스를 만들어 run() 전에 각종 설정 가능

```
@SpringBootApplication
public class FastcampusSpringBootApplication {

    public static void main(String[] args) {
        SpringApplication app = new SpringApplication(FastcampusSpringBootApplication.class);
        app.setBannerMode(Banner.Mode.OFF);
        app.run(args);
    }
}
```


부록 2 - 프로젝트 코드 구조 만들기

2. @SpringBoot Application

- 메인 애플리케이션 클래스는 루트 패키지에 놓는 것을 권장
- <https://docs.spring.io/spring-boot/docs/current/reference/html/using.html#using.structuring-your-code>

```
com
+- example
    +- myapplication
        +- MyApplication.java
        |
        +- customer
            +- Customer.java
            +- CustomerController.java
            +- CustomerService.java
            +- CustomerRepository.java
        |
        +- order
            +- Order.java
            +- OrderController.java
            +- OrderService.java
            +- OrderRepository.java
```