

# -parameters 컴파일 옵션 문제 상황

@RestController

```
public class HelloController {  
    @GetMapping("/hello")  
    public String hello(String name) {  
        return "Hello, " + name;  
    }  
}
```

위와 같은 코드 실행 시 잘되기도 하고, 예외 발생으로 -parameters 옵션 붙여 컴파일하라고 안내 나오기도 한다.

# 배경지식1

- Spring은 기존에 2가지 방법을 이용해서 Java 코드의 파라미터 이름을 가져오고 있었음
  - javac 옵션 중 ( -g Generate all debugging info:IDE 컴파일 기본 옵션값,
    - parameters: Generate metadata for reflection on method parameters )
  - 1. StandardReflectionParameterNameDiscoverer 구현체는 JDK의 Reflection 기능을 사용해 메서드 파라미터 이름을 가져오며 컴파일러의 -parameters 옵션에 의존하는 방법
  - 2. LocalVariableTableParameterNameDiscoverer 구현체는 LocalVariableTable로부터 메서드 파라미터의 이름을 가져오며 컴파일러의 -g 옵션에 의존하는 방법
- Spring Boot 3.2(Spring 6.1, JAVA17) 버전 부터  
LocalVariableTableParameterNameDiscoverer 제거됨.

# 배경지식2

- Gradle이 Spring 프로젝트를 빌드하는 방법

build.gradle 파일을 보면 plugins 으로 아래의 플러그인을 사용하도록 구성됨

```
7   plugins {  
8       id 'java'  
9       id 'org.springframework.boot' version '3.3.1'  
10      id 'io.spring.dependency-management' version '1.1.5'  
11  }
```

이 플러그인으로 인해 Gradle 빌드시 -parameters 옵션이 자동 적용됨.

# 근본 원인 및 해결책

- Spring Boot 3.2 이상에서는 Java 코드의 파라미터 이름을 가져오는 방법이 변경 되었기 때문에
  1. @RequestParam("name")을 이용해 명시적으로 파라미터 이름을 지정하던지
  2. Gradle로 빌드하던지
  3. 컴파일러 옵션에 -parameters 옵션을 붙여 컴파일 하자!

## 컴파일러 옵션에 `-parameters` 옵션을 붙여 컴파일하는 방법

- build.gradle 파일 안에 아래 코드 삽입

```
tasks.withType(JavaCompile) {  
    options.compilerArgs << "-parameters"  
}
```