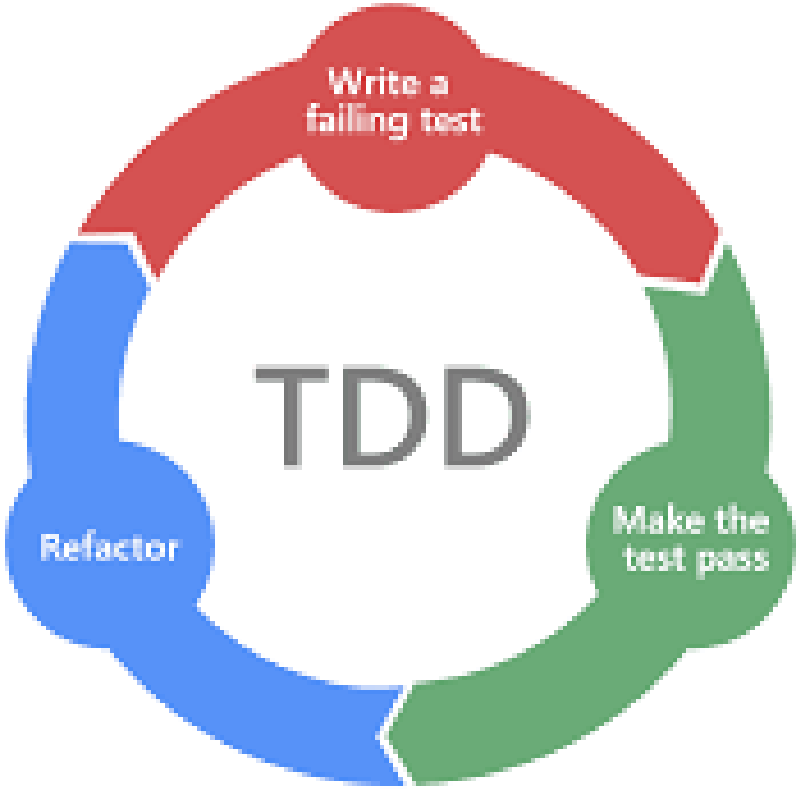
스프링 부트에서 테스트코드를 작성하자

2020-10-20

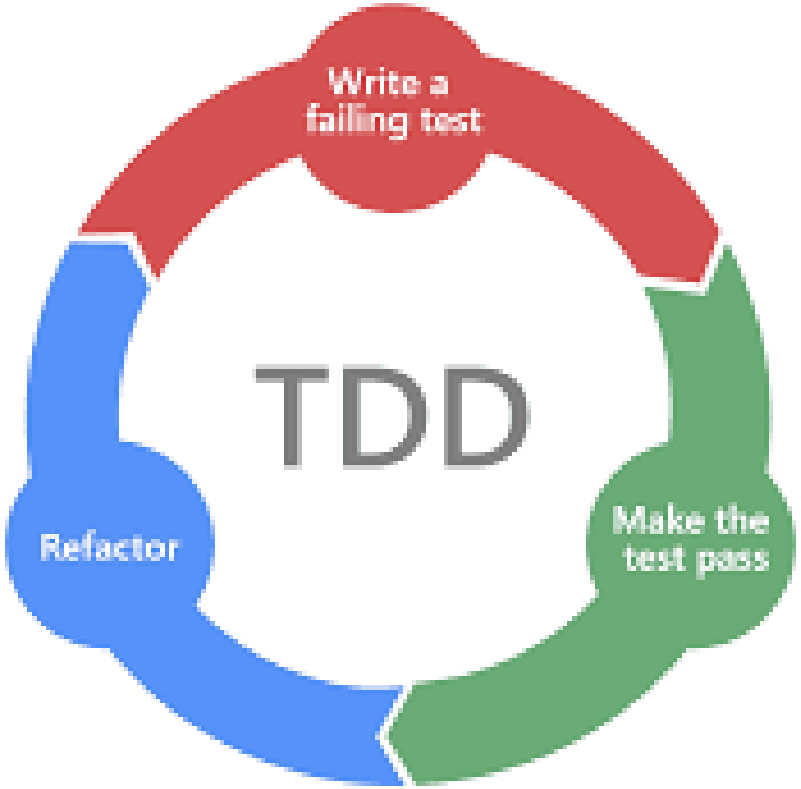
TDD란?



테스트 주도 개발은 매우 짧은 사이클을 반복하는 SW 개발 프로세스 중 하나이다.

TDD에서 Unit Test 작성을 제일 먼저 수행한다.
그러나 Unit Test를 사용한다고 해서 TDD를 보장하지 않는다. TDD는 반드시 Unit Test 코드 작성이 구현 이전에 이루어져야 하기 때문이다. 또한 TDD에서는 리팩토링도 이루어진다.

TDD란?



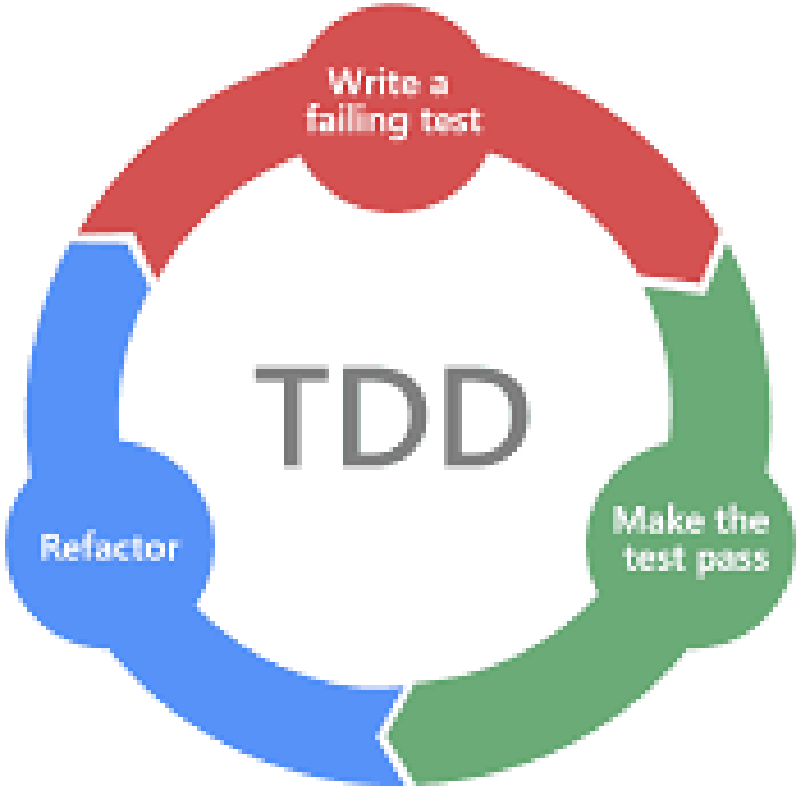
1. **테스트 코드를** 먼저 작성한다.
2. 테스트 코드가 패스되도록 **구현 코드**를 작성한다.
3. 테스트를 실행해본다. 성공하면 구현 코드를 **리팩토링**한다.

리팩토링이란?

인텔리제이에서 패키지 이름을 바꾸려면 Refactor 메뉴에 들어가야 한다.
이름을 바꾸려고 할 때, IDE가 리팩토링을 적용할 부분을 정하라고 한다.
이름을 바꾼 후, 해당 패키지 이름이 들어간 모든 파일에서 변경이 적용되었다는 것을 알 수 있다.

⇒ **리팩토링은 결과의 변경 없이 코드의 구조를 변경하는 것**을 뜻한다.
⇒ 따라서 TDD에서 리팩토링은 여러 cycle을 거치면서 코드를 좀 더 세련되게 만들어주는 단계라고 할 수 있다.

TDD가 왜 필요할까?



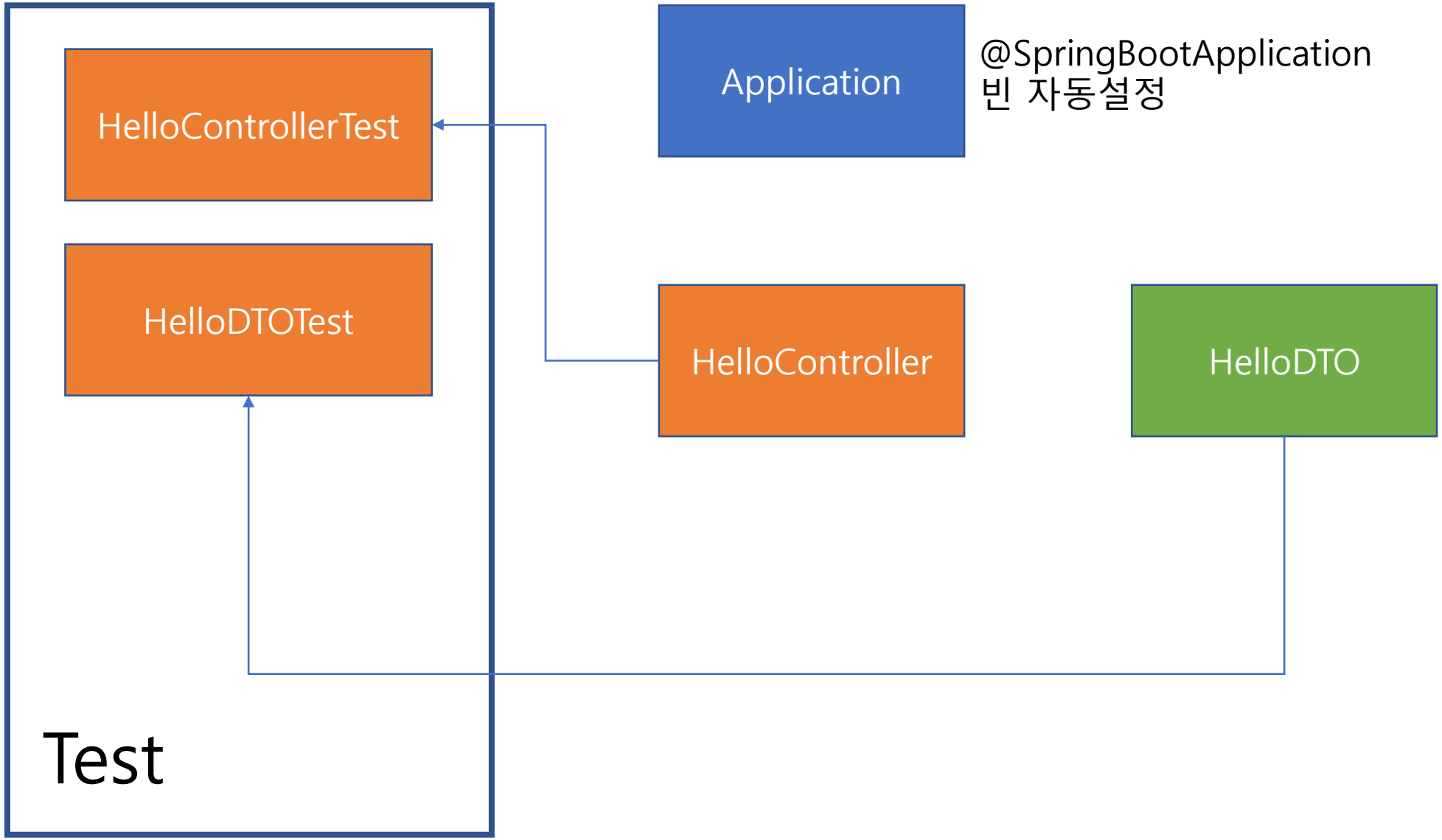
1. 테스트 코드를 미리 작성해 구현 코드 수정 시 피드백이 빠르다.
2. 자동 검증이 가능하다.
3. 서비스의 모든 기능을 테스트하지 않아도 된다.

JUnit

Junit은 자바 용 단위 테스트 프레임워크이다.
JUnit5가 출시되었지만 아직 JUnit4를 쓰는 추세이다.
JUnit5와 JUnit4의 차이는 @Before, @BeforeClass, @After 어노테이션을 좀더 직관적으로 이해할 수 있도록 @BeforeEach, @BeforeAll, @AfterEach로 바꾸었고, 접근제어자가 public이 기본이었는데, default로 바뀌었다.

JUnit4	JUnit5
@Before @BeforeClass @After	@BeforeEach @BeforeAll @AfterEach
Public void ... 접근제어자 public기본으로 사용.	void ... 접근제어자 Default로 사용.

접근 제어자	같은 클래스의 멤버	같은 패키지의 멤버	자식 클래스의 멤버	그 외의 영역
public	○	○	○	○
protected	○	○	○	X
default	○	○	X	X
private	○	X	X	X


HelloControllerTest
HelloDTOTest
Application
@SpringBootApplication
빈 자동설정
HelloController
HelloDTO
Test

@RestController

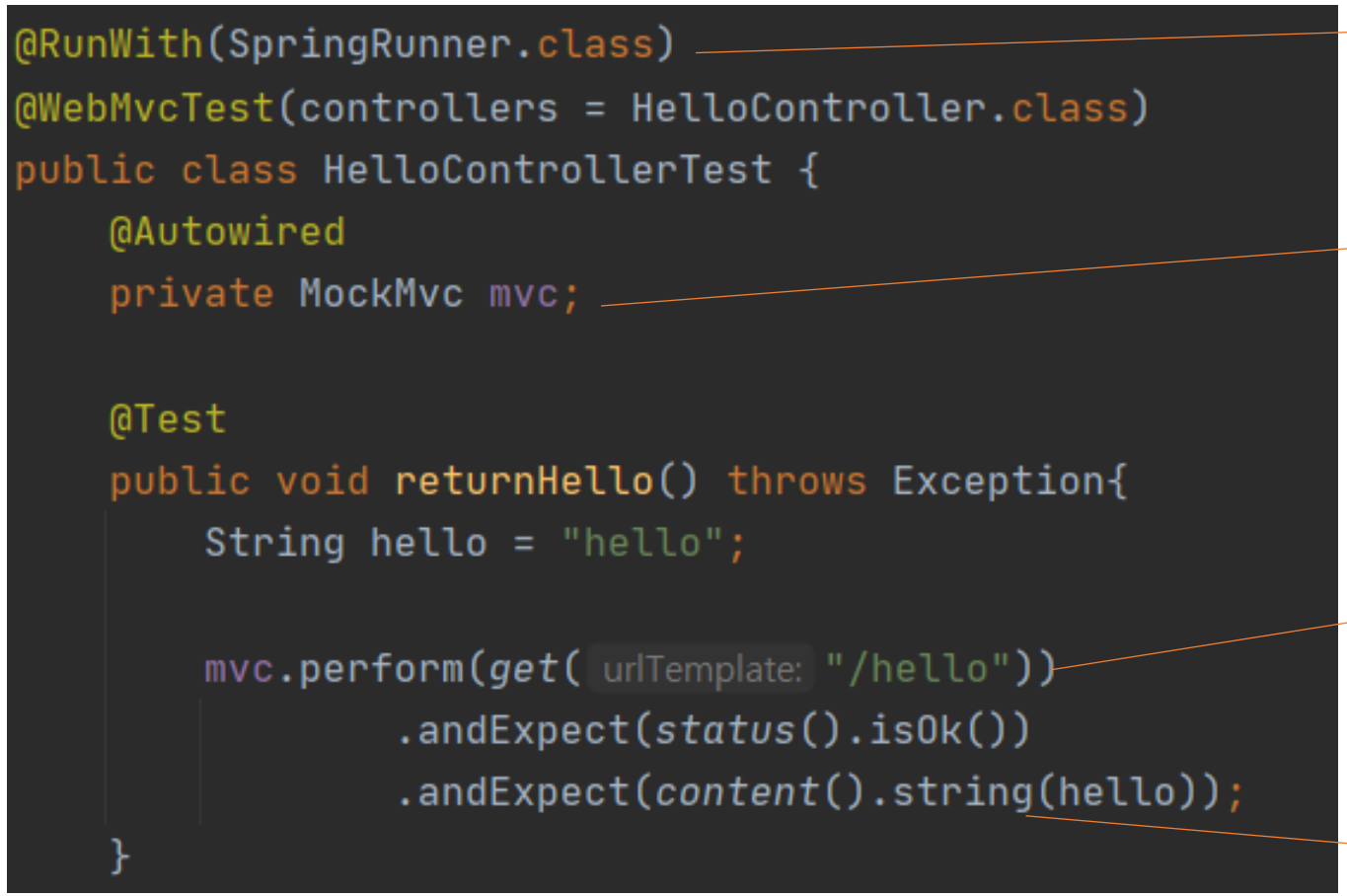
컨트롤러를 JSON으로 반환해주게 한다. RequestBody를 매번 쓰지 않아도 된다.

@Test

테스트 메서드에 붙이는 어노테이션이다.

@WebMvcTest

웹 MVC에 적합한 테스트 클래스에 붙이는 어노테이션이다.



테스트와 Junit 사이의 연결자 역할.
Junit에 내장된 실행자 외의 실행자를 사용할 수 있게 한다.

웹 API를 테스트하는
MockMvc 자동주입.

MockMvc를 통해
HTTP 요청을 보냄.

결과 검증

Lombok이란?

Getter, Setter, Constructor를 자동 생성해 주는 라이브러리다.
사용하려면 클래스 위에 필요한 Lombok 어노테이션을 붙이면 된다.

```
@Getter
@RequiredArgsConstructor
public class HelloResponseDto {
    private final String name;
    private final int amount;
}
```

@Getter:
모든 필드의 getter를 생성해준다.

@RequiredArgsConstructor
Final이 붙은 모든 필드를 정의할 수 있는 생성자를 만들어준다.