

스프링 부트 3 백엔드 개발자 되기 (2판)

JPA+OAuth2+JWT+AWS와 배우는 스프링 부트 3
Java 백엔드 입문자를 위한 풀 패키지

04 스프링 부트 3와 테스트

4.1 테스트 코드 개념 익히기

- test 디렉터리에서 작업
- 코드에는 다양한 패턴이 있음
 - 본 도서는 given-when-then 패턴으로 진행

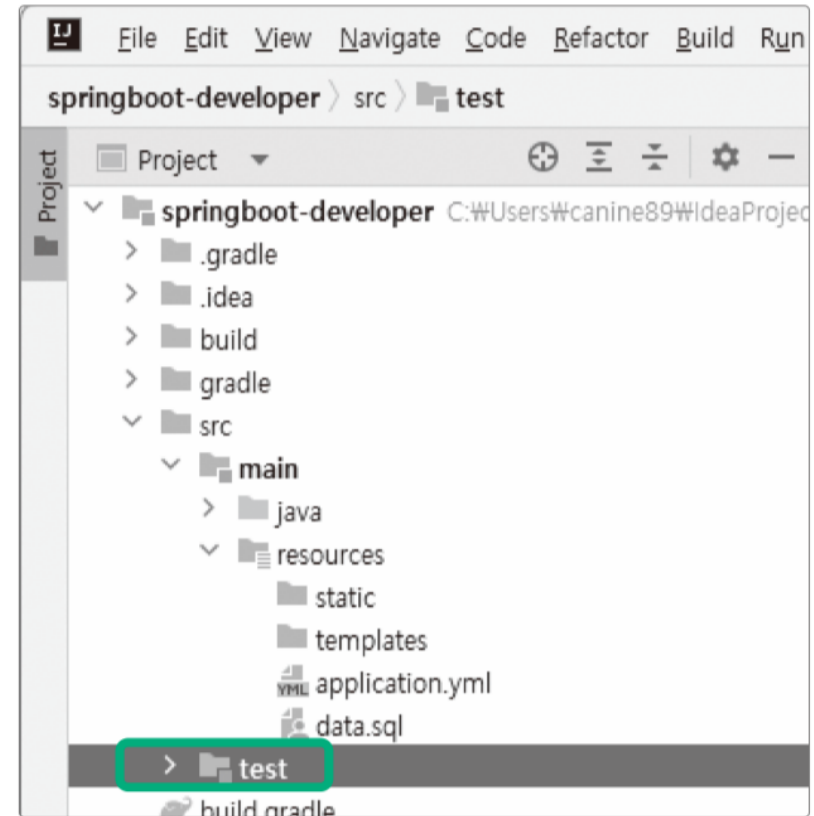
▼ given-when-then 패턴의 테스트 코드 예

```
@DisplayName("새로운 메뉴를 저장한다.")
@Test
public void saveMenuTest() {
    // given : 메뉴를 저장하기 위한 준비 과정
    final String name = "아메리카노";
    final int price = 2000;

    final Menu americano = new Menu(name, price);

    // when : 실제로 메뉴를 저장
    final long savedId = menuService.save(americano);

    // then : 메뉴가 잘 추가되었는지 검증
    final Menu savedMenu = menuService.findById(savedId).get();
    assertThat(savedMenu.getName()).isEqualTo(name);
    assertThat(savedMenu.getPrice()).isEqualTo(price);
}
```



4.2 스프링 부트 3와 테스트

- 스프링 부트 스타터 테스트 목록 참고
- JUnit, AssertJ를 가장 많이 사용



스프링 부트 스타터 테스트 목록

- JUnit : 자바 프로그래밍 언어용 단위 테스트 프레임워크
- Spring Test & Spring Boot Test : 스프링 부트 애플리케이션을 위한 통합 테스트 지원
- AssertJ : 검증문인 어설션을 작성하는 데 사용되는 라이브러리
- Hamcrest : 표현식을 보다 이해하기 쉽게 만드는 데 사용되는 Matcher 라이브러리
- Mockito : 테스트에 사용할 가짜 객체인 Mock 객체를 쉽게 만들고, 관리하고, 검증할 수 있게 지원하는 테스트 프레임워크
- JSONassert : JSON용 어설션 라이브러리
- JsonPath : JSON 데이터에서 특정 데이터를 선택하고 검색하기 위한 라이브러리

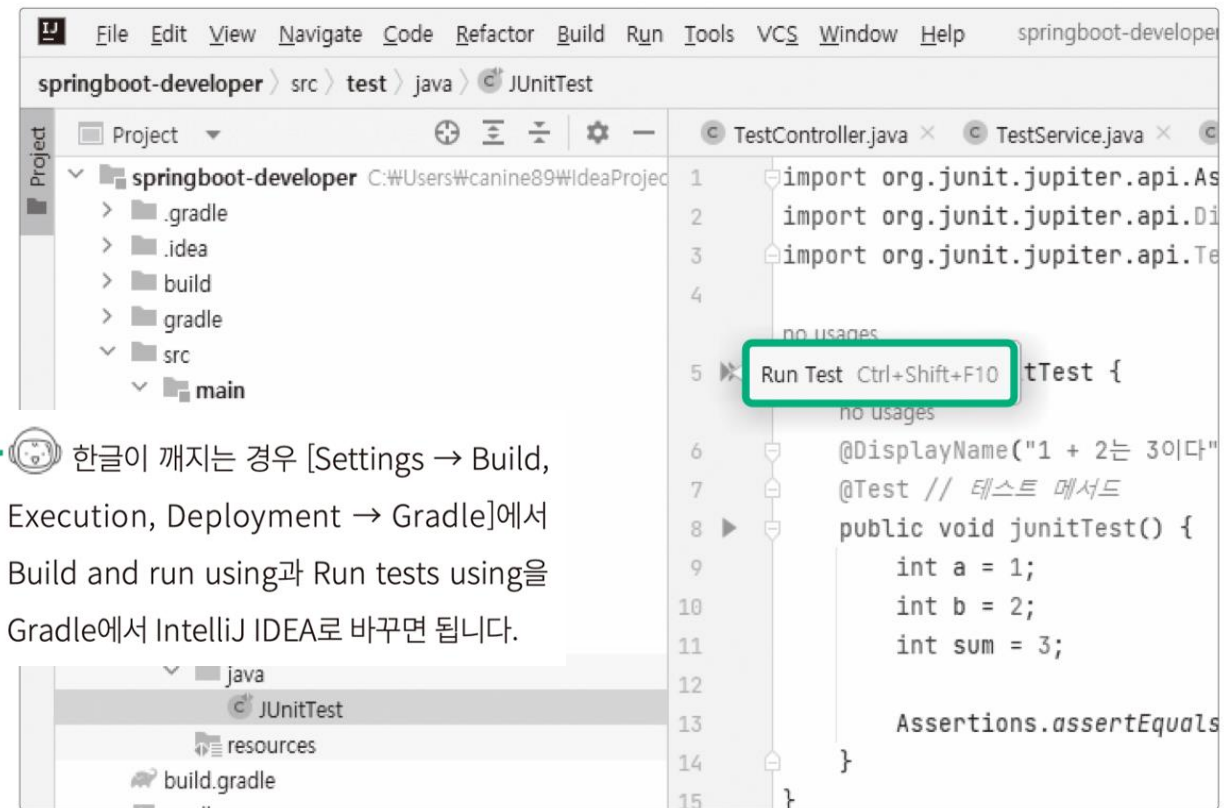
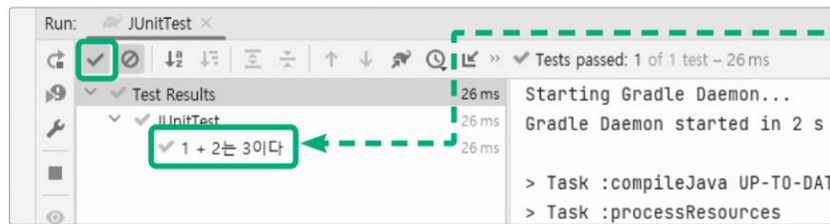
4.2 스프링 부트 3와 테스트(cont.)

- JUnit으로 단위 테스트 코드 만들기

- 91쪽부터 참고

- JUnitTest.java

- Assertions.assertEquals(sum, a + b)
- sum과 a+b가 같은지 확인
- Run test로 실행

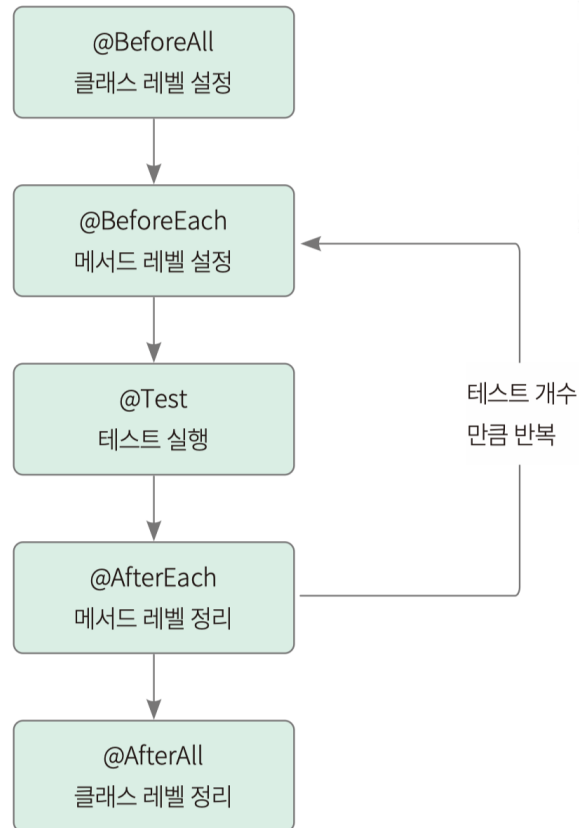


한글이 깨지는 경우 [Settings → Build, Execution, Deployment → Gradle]에서 Build and run using과 Run tests using을 Gradle에서 IntelliJ IDEA로 바꾸면 됩니다.

4.2 스프링 부트 3와 테스트(cont.)

- JUnitCycleTest.java

- 93쪽부터 참고
- @BeforeAll
- @BeforeEach
- @BeforeEach
- @Test
- @AfterAll
- @AfterEach



Tests passed: 3 of 3

Test Results	38 ms	
JUnitCycleTest	38 ms	> Task :compile
test1()	28 ms	> Task :process
test2()	2 ms	> Task :classes
test3()	8 ms	> Task :compile

4.2 스프링 부트 3와 테스트(cont.)

- 가독성이 좋은 AssertJ
 - isEqualTo 메서드 외 다양한 메서드가 있음
 - 자세한 내용은 97쪽 참고

메서드 이름	설명
isEqualTo(A)	A 값과 같은지 검증
isNotEqualTo(A)	A 값과 다른지 검증
contains(A)	A 값을 포함하는지 검증
doesNotContain(A)	A 값을 포함하지 않는지 검증
startsWith(A)	접두사가 A인지 검증

```
assertThat(a + b).isEqualTo(sum);
```

a + b가

sum과 같은가?

4.3 제대로 테스트 코드 작성해 보기

- testController.java
 - [Create Test]
 - test/java/<패키지> 아래에 파일 생성
- @SpringBootTest
- @AutoConfigureMockMvc
- @BeforeEach
- @AfterEach

```
@SpringBootTest // 테스트용 애플리케이션 컨텍스트 생성
@AutoConfigureMockMvc // MockMvc 생성
class TestControllerTest {

    @Autowired
    protected MockMvc mockMvc;

    @Autowired
    private WebApplicationContext context;

    @Autowired
    private MemberRepository memberRepository;

    @BeforeEach // 테스트 실행 전 실행하는 메서드
    public void mockMvcSetUp() {
        this.mockMvc = MockMvcBuilders.webAppContextSetup(context)
            .build();
    }

    @AfterEach // 테스트 실행 후 실행하는 메서드
    public void cleanUp() {
        memberRepository.deleteAll();
    }
}
```

TestControllerTest.java

4.3 제대로 테스트 코드 작성해 보기(cont.)

- testControllerTest.java
 - Given-When-Then 패턴 참고

Given	멤버를 저장합니다.
When	멤버 리스트를 조회하는 API를 호출합니다.
Then	응답 코드가 200 OK이고, 반환받은 값 중에 0번째 요소의 id와 name이 저장된 값과 같은지 확인합니다.

- HTTP 주요 응답 코드 참고

코드	매핑 메서드	설명
200 OK	isOk()	HTTP 응답 코드가 200 OK인지 검증
201 Created	isCreated()	HTTP 응답 코드가 201 Created인지 검증
400 Bad Request	isBadRequest()	HTTP 응답 코드가 400 Bad Request인지 검증
403 Forbidden	isForbidden()	HTTP 응답 코드가 403 Forbidden인지 검증
404 Not Found	isNotFound()	HTTP 응답 코드가 404 Not Found인지 검증
400년대 응답 코드	is4xxClientError()	HTTP 응답 코드가 400년대 응답 코드인지 검증
500 Internal Server Error	isInternalServerError()	HTTP 응답 코드가 500 Internal Server Error인지 검증
500년대 응답 코드	is5xxServerError()	HTTP 응답 코드가 500년대 응답 코드인지 검증

```
@SpringBootTest // 테스트용 애플리케이션 컨텍스트 생성
@AutoConfigureMockMvc // MockMvc 생성
class TestControllerTest {
```

... 생략 ...

@DisplayName("getAllMembers: 아티클 조회에 성공한다.")

@Test

public void getAllMembers() throws Exception {

// given

final String url = "/test";

Member savedMember = memberRepository.save(new Member(1L, "홍길동"));

// when

final ResultActions result = mockMvc.perform(get(url) // ①
 .accept(MediaType.APPLICATION_JSON)); // ②

// then

result

.andExpect(status().isOk()) // ③

// ④ 응답의 0번째 값이 DB에서 저장한 값과 같은지 확인

.andExpect(jsonPath("\$.id").value(savedMember.getId()))

.andExpect(jsonPath("\$.name").value(savedMember.getName()));

```
}
```

```
}
```

