

# บทที่ 10

## Testing in Spring Boot

### Unit Test & Integration Test

## หัวข้อที่จะเรียนรู้

- ความสำคัญของการทดสอบซอฟต์แวร์
- Testing Pyramid
- Unit Test vs Integration Test
- Frameworks: JUnit 5, Mockito, Spring Boot Test
- ตัวอย่างการเขียน Test ใน Spring Boot

## ทำไมต้องทดสอบ?

- ลด Bug ก่อนขึ้น Production
- เพิ่มความมั่นใจในการแก้ไขโค้ด (Refactor)
- ทำให้ระบบ **Maintainable** และ **Reliable**
- รองรับ **CI/CD Pipeline**

# Testing Pyramid

E2E Tests (น้อย)

Integration Tests (ปานกลาง)

Unit Tests (มากที่สุด)

- **Unit Test** → เร็ว, ทดสอบ logic เล็ก ๆ
- **Integration Test** → ทดสอบการทำงานร่วมกัน (DB, API)
- **E2E Test** → ทดสอบระบบเต็มรูปแบบ

# Unit Test

- ทดสอบ Method/Function เดี่ยว ๆ
- ไม่พึ่งพา external system
- ใช้ **Mockito** จำลอง dependency

ตัวอย่าง:

```
@Test
void testAddUser() {
    User u = new User("Alice");
    when(repo.save(u)).thenReturn(u);

    User result = service.addUser(u);
    assertEquals("Alice", result.getName());
}
```

# Integration Test

- ทดสอบหลายชั้นร่วมกัน (Controller + Service + Repository)
- ใช้ **Spring Boot Test** + Embedded Database
- ใช้ `MockMvc` หรือ `TestRestTemplate`

ตัวอย่าง:

```
@SpringBootTest
@AutoConfigureMockMvc
class UserControllerTest {
    @Autowired
    private MockMvc mockMvc;
    @Test
    void shouldReturnUsers() throws Exception {
        mockMvc.perform(get("/users"))
            .andExpect(status().isOk());
    }
}
```

## JUnit 5

- Framework มาตรฐานการทดสอบใน Java
- Annotation ที่ใช้บ่อย:
  - `@Test`
  - `@BeforeEach` , `@AfterEach`
  - `@BeforeAll` , `@AfterAll`

# Mockito

- ใช้จำลอง (Mock) dependencies
- ป้องกันการเรียก DB จริงหรือ service จริง
- Methods: `when()`, `verify()`, `any()`



# Spring Boot Test

- ใช้ Annotation
  - `@SpringBootTest` → โหลด context จริง
  - `@WebMvcTest` → โหลดเฉพาะ Controller Layer
  - `@DataJpaTest` → ทดสอบ Repository + DB

# Lab

## 1. เขียน Unit Test สำหรับ UserService

- mock UserRepository
- ทดสอบการสร้างผู้ใช้ใหม่

## 2. เขียน Integration Test สำหรับ /users API

- ใช้ MockMvc ตรวจสอบ response

# Assignment

- เขียน Test สำหรับ API `/users` :
  - Unit Test → `UserService` (เพิ่ม, ลบ, ค้นหา)
  - Integration Test → `/users` (GET, POST, DELETE)
- ส่ง:
  - โค้ด Test + ผลรัน (screenshot/coverage report)
  - ต้องมีอย่างน้อย **6 test cases** (Unit  $\geq 4$ , Integration  $\geq 2$ )