Boost your Architecture by Test-Driven Development

Hands-on training on how to use Test-Driven Development (TDD) to boost the architecture of your projects. The course covers writing unit, integration, and end-to-end tests, showing how you can guard and enforce the rules of your architecture. We show the ways TDD helps with building and improving sustainable architecture. JUnit, Kotest, and Spock frameworks are used in the examples, however, the knowledge gained during the course is easily applicable to other testing libraries. During the course, attendees will learn about the challenges of maintaining a huge number of tests in a project, and how to avoid the usual pitfalls.

Audience: Developers and Tech Leads
Duration: 2 days

- Day 1: Introduction, Test doubles, Properly written tests
- Day 2: Integration tests based on Spring Boot setup, Testing Dates, Tests Maintenance

Format: 60% workshop / 40% lecture

Training program

1. Introduction
    a. Why do we need tests?
    b. Unit, integration, contract, and end-to-end tests
    c. Shapes of tests
    d. TDD/BDD
2. Kotest or Spock
    a. Features and examples
    b. Killer features
3. Test doubles
    a. Dummies
    b. Fakes
    c. Stubs
    d. Mocks
    e. Spies
4. Properly written tests
    a. Good unit tests
    b. Code smells in unit tests
    c. Patterns
5. Integration tests (with examples in Spring Boot)
    a. Why do we need integration tests?
    b. Different flavours of integration tests
    c. Setup for integration tests
        i. Testing integration with database

        ii. Testing application services
        iii. Testing REST
        iv. Testing communication with external services

6. Testing Dates
7. Tests Maintenance
    a. Challenges with maintaining a large number of tests
    b. Readable and maintainable big suites of tests