



Program szkolenia

JUnit dla zaawansowanych

Szczegóły

Kategoria: Testowanie i QA

Czas trwania: 24h / 3 dni

Termin: ustalany indywidualnie

Cena: ustalana indywidualnie

Opis

Zaawansowane szkolenie z testów jednostkowych - JUnit zostało zaplanowane na 3 dni szkoleniowe. Pierwszego dnia omówimy bardzo dokładnie architekturę i sposób na wykorzystanie w praktyce wszystkich funkcjonalności, które oferuje JUnit, Mockito oraz AssertJ. Drugiego dnia do całości dorzucimy Spring Boota i poznamy ułatwienia, które nam oferuje, aby ułatwić i przyspieszyć pisanie testów, które wymagają uruchomione kontekstu aplikacji. Poznamy również najlepsze praktyki związane z testowaniem API (w tym również zabezpieczonego), interakcji z bazą danych oraz integracji z innymi serwisami. Trzeciego dnia skupimy się na dodatkowych narzędziach ułatwiających pisanie wartościowych testów oraz powtórzymy całość materiału. Szkolenie zakończymy napisaniem testów do projektu, który wymusi zastosowanie wszystkich nabytych podczas szkolenia umiejętności. Całe szkolenie będzie mocno praktyczne i składające się głównie z ćwiczeń.

Program

Poniżej przedstawiamy przykładowy program szkolenia, który może zostać zmodyfikowany zgodnie z oczekiwaniami oraz poziomem grupy szkoleniowej. Przed przygotowaniem docelowego programu szkolenia przeprowadzamy rozmowę techniczną, w której bierze udział trener oraz osoba techniczna lub cały zespół developerów reprezentujący klienta, w celu ustalenia szczegółów szkolenia.

Dzień: 1 (8h)

JUnit 5

1. Teoria testów jednostkowych

- a) Motywacja do pisania testów jednostkowych
- b) Zasady pisania dobrych testów
- c) Dobre praktyki, powszechne błędy - tak naprawdę ten punkt będzie przewijał się przez całe szkolenie

2. Trzy podstawowe narzędzia - JUnit5, Mockito, AssertJ

Powtórka i rozwinięcie najpopularniejszej kombinacji narzędzi spotykanej przy pisaniu testów jednostkowych w Javie.

- a) Wstęp do architektury JUnit
- b) Podpięcie JUnit do projektu
- c) Mockito - dlaczego jest ważne, jak poprawnie z niego korzystać, mniej popularne funkcjonalności przydatne w codziennej pracy
- d) AssertJ - najczytelniejsza biblioteka do asercji w Javie

3. JUnit 5 - advanced

- a) Assumptions
- b) Lifecycle - omówienie dwóch dostępnych rodzajów
- c) Repeated i Nested tests - use cases
- d) Testy parametryzowane - różne rodzaje parametryzacji
- e) Praca z plikami
- f) Wstrzykiwanie do metod testowych - wstęp do ExtensionModel

4. Extension Model - deep dive

- a) Build-in extensions
- b) Jak pisać własne rozszerzenia
- c) W jaki sposób współdzielić powtarzalny kod między testami przy pomocy rozszerzeń

5. JUnit in Real Life

- a) Konfiguracja JUnit 5 z Maven / Gradle
- b) Optymalizacja czasu wykonania testów
- c) Generowanie raportów i integracja z CI / CD
- d) Migracja JUnit 4 -> JUnit 5

Dzień: 2 (8h)

JUnit 5 i Spring Boot

1. Czym testy kontekstu Spring Boota różnią się od typowych testów jednostkowych

- a) Kiedy wystarczy nam zwykły test, a kiedy powinniśmy postawić kontekst Spring Boota

2. Wsparcie Spring Boota do pisania testów

- a) Adnotacja @SpringBootTest
- b) Konfiguracja testowa + test properties
- c) Bean Mocks
- d) Testy z uruchomionym serwerem
- e) MockMvc
- f) Slice tests

3. Testowanie integracji z bazą danych

- a) Jak dostarczyć bazę danych dla testów
- b) Jak jednostkowo testować interakcje z bazą danych
- c) Kiedy testować jednostkowo, a kiedy integracyjnie
- d) Jak testować kod, który wymaga transakcji

4. Integracja ze Spring Security

Dzień: 3 (8h)

Zewnętrzne narzędzia

- 1. Wiremock**
 - a) Testowanie integracji z innymi serwisami
 - b) TestWebClient vs TestRestTemplate
 - c) Testy integracyjne Wiremock
- 2. TestContainers** - co oferują i kiedy warto skorzystać
- 3. Contract Tests - demo**
- 4. Powtórzenie i podsumowanie materiału**
- 5. Ostatni projekt**

Kontakt



Przemysław Wołosz

Key Account Manager

infoShare Academy

+48 730 830 801

przemyslaw.wolosz@infoshareacademy.com