

JavaScript Automated Testing

Ducin IT Consulting - Program szkolenia

Czas trwania: 2-3 dni

Formuła: 25% teoria, 75% praktyka

Szkolenie ma na celu przybliżyć - w zależności od potrzeb grupy - podstawowe oraz zaawansowane zagadnienia związane z testowaniem automatycznym kodu JavaScriptowego. Kładzie nacisk nie tylko na poprawne pisanie testów, ale i wychwytywanie krytycznych fragmentów kodu, które najbardziej warto pokryć testami.

Praktyczne przykłady realizowane są m.in. w oparciu o TDD oraz BDD. Uczestnicy uczą się technik testowania Unitowego, Integracyjnego oraz End- to- End oraz analizują, dlaczego tzw. "piramida testów" ma taki, a nie inny kształt. Testowane są różnorodne elementy systemów - takie jak logika biznesowa, asynchroniczne żądania, timery, warstwa wizualna DOM - a także zaawansowane techniki takie jak keshowanie, restartowanie żądań i wiele, wiele innych. Testowanie End- to- End uwzględniają pokrycie testami istniejącej, popularnej aplikacji.

Szkolenie zakłada co najmniej podstawową wiedzę z zakresu JavaScriptu.

Kluczowe Aspekty:

- Skuteczne wyszukiwanie fragmentów kodu, które warto testować
- Techniki pisania testów, które nie łamią się przy okazji dowolnego refactoru. Testy, które badają funkcjonalność, a nie implementację
- Kluczowe techniki mockowania, stubowania - wspomagające pisanie szybkich i precyzyjnych testów unitowych
- Proste i skuteczne testy End-to-End

Program szkolenia:

1. Teoria

1.1. Rodzaje testów

1.1.1. Unitowe

1.1.2. Integracyjne

1.1.3. E2E

1.1.4. Property Testing

1.1.5. Testy Mutacyjne

1.2. Matchers, Expects

1.3. Test structure

1.4. Test-Driven Development

1.5. Behavior-Driven Development

2. Integracje

2.1. Code Coverage

2.2. integracja: CI/CD

3. Implementacja

3.1. Struktura testów

3.2. Setup, Teardown

3.3. Arrange, Act, Assert

3.4. Testy sync i async

3.5. Pętle testów

3.6. "System Under Test"

4. Testy kodu vs jego zależności

4.1. mockowanie

4.2. stubowanie

4.3. testowanie zachowań

5. Testowanie

5.1. testowanie logiki biznesowej

5.2. testowanie wzorców projektowych

5.3. testowanie operacji asynchronicznych

5.4. testowanie eventów

5.5. testowanie HTTP

6. Asercje

6.1. przegląd asercji

6.2. customowe asercje

6.3. biblioteka chai

7. Narzędzia

7.1. Istanbul/NYC

7.2. Mocha

7.3. Jest

7.4. Karma

7.5. Jasmine

7.6. Sinon.js

7.7. Lolex

7.8. Chai

7.9. ajv

7.10. Puppeteer

7.11. Node.js vs Przeglądarki

7.12. i wiele, wiele innych

8. (Opcjonalnie)

8.1. testowanie jquery

8.2. testowanie Reacta

8.3. testowanie Reduxa

8.4. testowanie Angulara

8.5. testowanie NGRX