

Kod szkolenia: **REACT/TDD**

Tytuł szkolenia: **Test-Driven Development w ekosystemie React**

Dni: 2

Opis:

Adresaci szkolenia:

Szkolenie adresowane jest do programistów tworzących aplikacje wykorzystujące bibliotekę React. Kierowane jest do osób chcących zwiększyć pewność siebie podczas tworzenia oprogramowania (mniej regresji i nieprzewidzianych defektów) oraz uzyskać lepszą jakość i czytelność kodu.

Cel szkolenia:

W wyniku szkolenia uczestnicy poznają założenia metodologii TDD oraz podstawy pracy z narzędziami stosowanymi przy testowaniu aplikacji budowanych za pomocą biblioteki React.

Celem szkolenia jest omówienie oraz przećwiczenie pracy w cyklu TDD oraz testowania poszczególnych fragmentów aplikacji. Uczestnicy będą pracować z bibliotekami ułatwiającymi stosowanie TDD w kontekście komponentów, dyrektyw i usług w typowych konfiguracjach.

Mocne strony szkolenia:

Charakterystyczną cechą szkolenia jest szerokie spektrum poruszanych zagadnień przy jednoczesnym skupieniu na aspekcie praktycznym. Największy nacisk położony jest na sytuacje i narzędzia, które występują się w rzeczywistych projektach. Część warsztatowa szkolenia pozwala na osiągnięcie poziomu umiejętności pozwalającego na samodzielne wykorzystanie zdobytej wiedzy od razu po zakończeniu kursu.

Wymagania:

Od uczestników wymagana jest przynajmniej podstawowa umiejętność używania biblioteki React.

Parametry szkolenia:

2*8 godzin (2*7 godzin netto) wykładów i warsztatów (z wyraźną przewagą warsztatów).

Program szkolenia:



1. Podstawy TDD w JavaScript

- Dlaczego TDD może Ci pomóc
- Stosowanie cyklu TDD Red-Green-Refactor
- Definicja TDD
- Rodzaje i poziomy testów
- Specyfika testowania aplikacji JS
- Struktura testu jednostkowego
- Definiowanie asercji
- Testowanie asynchronicznego kodu
- Mocks, spies, stubs
- Wyświetlanie wyników testu
- Raport pokrycia testami

2. TDD React

- Możliwości React Test Utilities
- Praca z enzyme
- Shallow/full rendering komponentów
- Testowanie stanu komponentu
- Testowanie struktury komponentu
- Symulowanie zdarzeń przeglądarki
- Snapshot testing
- Fixtures
- Debugging testów
- Komunikacja z API w testach

3. TDD Redux

- Testowanie akcji
- Testowanie reducerów

4. Refaktoryzacja

- Rozpoznawanie i poprawa kodu trudnego do testowania
- Pisanie czystego kodu w JavaScript - przydatne wzorce
- Refaktoryzacja w praktyce

5. Testy E2E



- Kiedy potrzebujemy testów end-to-end
- Zalety i wady testów z poziomu przeglądarki
- Co testować?
- Budowa utrzymywanych testów e2e

