

Program szkolenia:

Architektura Flux i React.js

Informacje:

Nazwa:	Architektura Flux i React.js
Kod:	JS-react
Kategoria:	JavaScript
Grupa docelowa:	architekci
Czas trwania:	3 dni
Forma:	40% wykłady / 60% warsztaty

Celem szkolenia jest nabycie konkretnych umiejętności projektowania aplikacji po stronie klienta, których złożoność odpowiada realnym problemom typowych projektów.

Szkolenie odpowiada na pytania: jaką strukturę i porządek wyższego rzędu powinna mieć aplikacja, która będzie żyć kilka lat i która ma być testowalna oraz relatywnie łatwa w utrzymaniu.

Zalety szkolenia:

- Nastawienie na testowalność
- Całościowe podejście do architektury
- Najlepsze praktyki bazujące na doświadczeniu trenera

Szczegółowy program:

1. React

1.1. Why React, what problems does it solve

1.2. React philosophy

1.2.1. Re-rendering whole page

1.2.2. Shadow DOM

1.2.3. Immutability

1.3. Components

1.3.1. Component basics

1.3.2. Component composition

1.3.3. Passing data between components

1.3.4. Keeping and modifying state

1.3.5. Inter-component communication

1.3.6. JSX tags

1.3.7. Functional stateless components

1.3.8. Component lifecycle

1.3.9. Integration with non react libraries

1.3.10. Mixins

1.3.11. Higher order components

1.4. User interaction

1.4.1. Event handlers

1.4.2. Managed unmanaged components

1.4.3. Forms and inputs

1.5. External libraries

1.5.1. Routing

1.5.2. Component libraries

1.5.3. Internationalization

1.6. Unit testing

1.6.1. How to unit test components

1.6.2. Making components easy to test

1.6.3. Using mocks

1.6.4. Using Jest for testing

1.6.5. Using Jasmine/Karma for testing

1.7. Code coverage

1.7.1. UI testing

1.7.2. How to test components using Selenium

1.7.3. Selenium setup

1.7.4. Understanding asynchronous Selenium tests

1.7.5. How to create Selenium tests so UI changes won't break them

1.8. Best practices

2. Flux

2.1. What is Flux app architecture

2.2. How data flows in Flux application

2.3. Keeping and changing application state

2.4. Immutability in application state

2.4.1. Solutions

2.4.2. Performance implications

2.5. Asynchronous data loads and Flux

2.6. Actions and events

2.7. Testing Flux components

2.8. Best practices

3. Environment and process:

3.1. Project setup

3.1.1. How to structure project

3.1.2. Where to keep tests

3.2. Tools

3.2.1. Bundlers

3.2.2. Transpilers

3.2.3. Test runners

3.2.4. Production and development builds