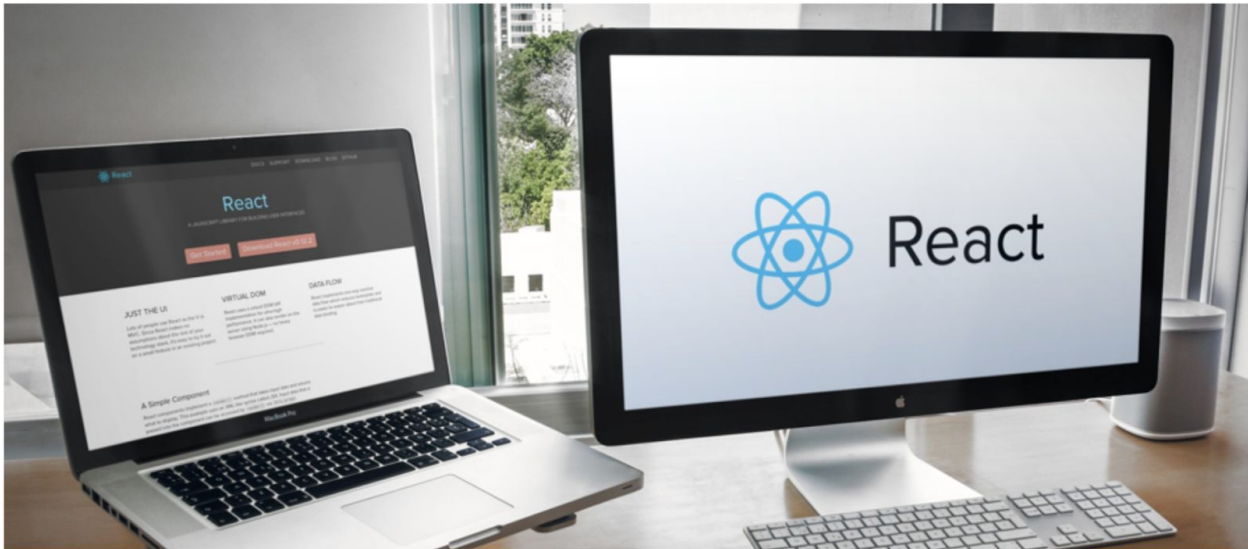


Opis szkolenia

React + Redux dla programistów



Zapraszamy na 3-dniowe intensywne warsztaty szkoleniowe z React + Redux.

Na szkoleniu poznasz tajniki pracy z Reactem, cały kod napiszesz w ES2015 i JSX, poznasz architekturę Flux oraz inspirowany Fluxem kontener stanu jakim jest Redux.

Zaimplementujesz też funkcję undo/redo i zdebugujesz aplikację korzystając z *Time Traveling Debugger*. Część czasu poświęcimy też na *performance* aplikacji, dowiesz się czym jest *Isomorphic JavaScript*. Wyposażony w tę wiedzę będziesz w stanie dobrać odpowiednie narzędzia do zadań postawionych przed Twoim zespołem.

Kod na szkoleniu będzie poparty testami, omówimy najlepsze praktyki w pisaniu dobrze testowanego kodu. Stworzymy setup projektu wykorzystując narzędzia takie jak Webpack i Babel.

Dlaczego React?

React to wydajna biblioteka do tworzenia dynamicznych interfejsów użytkownika. Facebook stworzył ją w celu rozwiązania swoich problemów z budowaniem dużych aplikacji, w których dane zmieniają się w czasie. React oparty o idę tworzenia aplikacji z mniejszych komponentów okazuje się też świetnym

narzędziem do pracy również nad mniejszymi aplikacjami czyniąc je prostymi w testowaniu, rozwijaniu i późniejszym utrzymaniu.

React to nie tylko biblioteka, ale cały jej ekosystem. Silna społeczność wypracowała najlepsze praktyki pracy z Reactem czyniąc je *de facto* standardem.

Dla kogo?

Szkolenie i przekazywana na nim wiedza jest skierowana do osób programujących w JavaScript i chcących budować wydajne i łatwe w utrzymaniu SPA.

Wymagana wiedza

Od uczestnika szkolenia wymaga się znajomości języka JavaScript przynajmniej w specyfikacji ES5 oraz podstawowej wiedzy z zakresu aplikacji webowych.

Agenda

Dzień 1

1. ES6 vs ES7 vs ES8

- czym są kolejne wersje języka, gdzie szukać informacji o tym, co i kiedy staje się częścią JS

2. Przegląd części języka dodanych w ES6 i ES7:

- arrow functions
- klasy
- destructuring assignment
- rest/spread operator
- template strings
- object literal enhancements
- modules
- promises
- fetch

3. Babel

- Jak pisać kod ES6/ES7 i transpilować przez Babel
- Budowa i konfiguracja

- Korzystanie z presetów
- Babel-preset-env: autoprefixer pozwalający na zdefiniowanie wspieranych funkcji języka na podstawie wersji przeglądark
- Wykraczając poza ES6/ES7: nowe funkcje (proposals) wspierane przez Babel typu class properties, async/await, decorators, a nie wspierane jeszcze natywnie

4. Automatyzacja pracy frontend developera z wykorzystaniem Webpacka

- Bundlowanie dowolnych modułów i zasobów z użyciem Webpack
- Build developerski i produkcyjny
- Customowa konfiguracja vs create-react-app + "eject"
- Wiązanie CSS'ów z komponentami przy pomocy styled-components

5. Początki z React

- Koncepcja budowania interfejsów z komponentów
- Pierwszy komponent „Hello World!”
- JSX, a HTML
- Kompozycja komponentów
- Przekazywanie danych między komponentami

6. Pierwsza część aplikacji w React

- Obsługa zdarzeń
- Obsługa formularzy
- Referencja do komponentów
- Stan komponentów

Dzień 2

1. Ciąg dalszy esencji React'a

- Komponenty funkcyjne / functional components
- Cykl życia komponentu - lifecycle hooks
- Definiowanie typów wejściowych komponentu przez prop types
- Wartości domyślne komponentu / default props

2. React-router

- Renderowanie ścieżek
- Zagnieżdżone ścieżki
- Używanie routera w kodzie z poziomu JS'a

3. Zaawansowane koncepty React'a przydatne w codziennej pracy:

- Kontekst komponentu / context
- Higher order components
- Containers
- Hooks

4. Flux - wprowadzenie

- Wprowadzenie do architektury Flux
- Koncepcje store'ów, akcji, dispatcher'ów
- Flux Standard Action

5. Redux – wstęp

- Wprowadzenie i podstawowe założenia
- Podstawowe koncepcje reduxa: store, action, reducer, dispatcher
- Czym jest reducer

Dzień 3

1. Kontener stanu Redux

- Dodanie redux do istniejącej aplikacji react'owej
- Przeniesienie stanu aplikacji do Store'ów
- Obsługa asynchroniczności (m. in. obsługa side effects)
- Redukcja boilerplate'u

2. Toolkit

- Funkcja Undo/Redo
- Time Traveling Debugger

3. TDD i testowanie w projektach React'owych:

- Kiedy, jak i dlaczego testować kod?
- Konfiguracja środowiska do testów jednostkowych
- Testowanie poszczególnych części aplikacji
- Strategie ułatwiające testowanie

100% praktyki

Implementacja aplikacji

Podczas warsztatów uczestnicy stworzą pod okiem trenera od podstaw prosty, aczkolwiek kompletny sklep internetowy - poznając omawiane narzędzia oraz techniki implementacji w React + Redux.

Oprócz aspektów związanych z implementacją, przejdziemy samodzielnie przez przygotowanie środowiska developerskiego, przygotowanie aplikacji do wypuszczenia na produkcję, testowanie oraz deployment.