

Opis szkolenia

Power-up React + Redux



Piszesz już aplikacje w Reakcie i redux'ie? Zapraszamy na **3-dniowe** intensywne warsztaty szkoleniowe z zaawansowanych aspektów związanych z tymi technologiami!

Czego nauczysz się podczas warsztatów?

- Pisać bardziej wydajny kod w React.js (np. dzięki Pure Components, unikanie niepotrzebnych renderów)
- Rozwiązywać nietrywialne problemy, jak skomplikowane wyrażenia warunkowe w JSX, reużywalność komponentów (higher order components)
- Obsługiwać efekty uboczne w reduxie (czystsza architektonicznie komunikacja z backendem)
- Pisać aplikacje, które są "future-proof", gotowe na platformy mobilne czy renderowanie po stronie serwera.
- Poznać najnowsze aspekty i możliwości Reacta
- Dobrze testować reacta i reduxa.

Dla kogo?

Szkolenie kierujemy do programistów, którzy już piszą aplikacje Reactowe. Umiejętności, które powinieneś mieć przychodząc na warsztaty:

- React: umiesz stworzyć komponent, zrobić hierarchię / drzewko komponentów i przekazywać dane
- Redux: umiesz dostarczyć stan do komponentów czy zmienić stan aplikacji przez akcje

Agenda

1. Redux side effects

- redux-thunk - najprostszy sposób na asynchroniczne akcje
- jak dobrze implementować komunikację aplikacji “ze światem zewnętrznym”
- przechwytywanie akcji - jak wywołać zapytanie do API za pomocą akcji
- połączenie z routerem - powiadomienie store’a o zmianie ścieżki
- połączenie z serwisami i asynchroniczność (http, websockets)
- autoryzacja z użyciem JWT (JSON Web Tokens)

2. Redux side effects z redux-saga

- wstęp do sagi - jak wykorzystać generatory do asynchronicznych akcji
- porównanie redux-saga z redux-thunk
- zarządzanie bardziej skomplikowanymi flow side effect’ów (np. akcje zależne od siebie)
- integracja z zewnętrznym API z użyciem sag
- łączenie sag - zarządzanie równoległymi efektami
- redux-saga vs redux-observable - dlaczego wybraliśmy sagi?

3. React patterns

- Hooks
- Higher Order Components - kompozycja komponentów jako sposób na zarządzanie złożonością
- function as child
- recompose - przydatna paczka pozwalająca na wielopoziomą kompozycję
- wyrażenia warunkowe w JSX
- asynchroniczny setState

- Event switch - zarządzanie eventami w komponencie
- Jak zbudować "Provider" z Reduxa - zrozumienie contextu w Reacie
- Dekoratory

4. Wydajność react'a

- jak działa React: Virtual-DOM i reconciliation, React Fiber
- profilowanie aplikacji pod kątem problemów z wydajnością
- PureComponent - wydajny komponent w React
- immutability - jak wpływa na wydajność aplikacji: koszty vs zysk
- złożone komponenty
- shouldComponentUpdate - zarządzanie zmianami drzewa komponentów w Reacie
- antypatterny - np. Bindowanie funkcji w renderze

5. Progressive Web Apps w react.js

- co to jest i po co?
- podział na moduły i dynamiczne ładowanie
- first load - poprawienie czasu pierwszego ładowania
- manifest - strona jako aplikacja
- podejście Offline-first
- service-workers - javascript w tle
- lighthouse – narzędzie do analizy zgodności aplikacji z koncepcją PWA

6. Server-side rendering

- First load - poprawienie czasu pierwszego ładowania
- Next.js - serwer do renderowania Reacta
- Dostarczenie danych do Reacta po stronie serwera
- Obsługa routingu po stronie serwera

7. Silnie typowany React: Flow czy Typescript

- Typescript vs Flow - dwa sposoby na statyczne typowanie kodu
- React + TypeScript - przepisanie aplikacji do Typescript czy dodanie statycznych typów
- React + Flow - dodanie typów do istniejącej aplikacji

8. Testowanie react'a przy pomocy Jest i Enzyme

- Jest - testowanie Reacta

- Enzyme - tworzenie zmockowanych komponentów (shallow vs mount)
- Testowanie logiki biznesowej
- Testowanie komponentów: DOM
- Testowanie komponentów: Snapshot
- Testowanie Redux - jak przetestować kontener stanu Reduxa
- Testowanie Sag

9. Dodatkowe tematy

- Stylowanie komponentów - różne podejścia (CSS modules vs styled-components)
- Animacje - react-transition-group i React Motion
- MobX - alternatywa do Reduxa

100% praktyki

Podczas warsztatów pod okiem trenera uczestnicy od podstaw stworzą klon popularnego narzędzia Trello, poznając omawiane narzędzia oraz techniki zaawansowanych aspektów związanych z react.js i redux.

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