

ReactJS

REACT ROUTER

Hello, World!



KAJ BIAŁAS _____

JavaScript Developer

Issue #0

Utwórz nowy projekt korzystając z `$ create-react-app .`

Zaimplementuj aplikację zgodnie ze strukturą,
korzystając z obiektu props komponentów:

Nazwa aplikacji

Treść

Informacje o autorze

React Router

Czym jest?

Biblioteka służąca do podmiany wyświetlanego komponentu głównego, podczas przechodzenia między podstronami.

React Router

Instalacja biblioteki

Nazwa biblioteki: `react-router-dom`

Instalacja paczki przez NPM: `$ npm install react-router-dom --save`

React Router

Utworzenie route'ingu

Import biblioteki:

```
import {BrowserRouter, Route} from 'react-router-dom';
```

Przykładowe utworzenie routingu:

```
<BrowserRouter>  
  <Route exact path="/" component={Home} />  
  <Route path="/about" component={AboutMe} />  
  <Route path="/contact" component={Contact} />  
  <Route path="/projects" component={Projects} />  
</BrowserRouter>
```

No fajnie!

Ale jak to wygląda w praktyce?

React Router

Utworzenie route'ingu

Plik: `index.js`

```
import React from 'react';
import ReactDOM from 'react-dom';
import { BrowserRouter, Route } from 'react-router-dom';

import Home from './Home';
import About from './About';

ReactDOM.render(
  <BrowserRouter>
    <div>
      <Route exact path="/" component={Home} />
      <Route path="/about" component={About} />
    </div>
  </BrowserRouter>,
  document.getElementById('root')
);
```


Issue #1

Wykorzystaj kod oraz strukturę podstron z: `Issue #0`

Zaimplementuj aplikację zawierającą podstrony:

Home

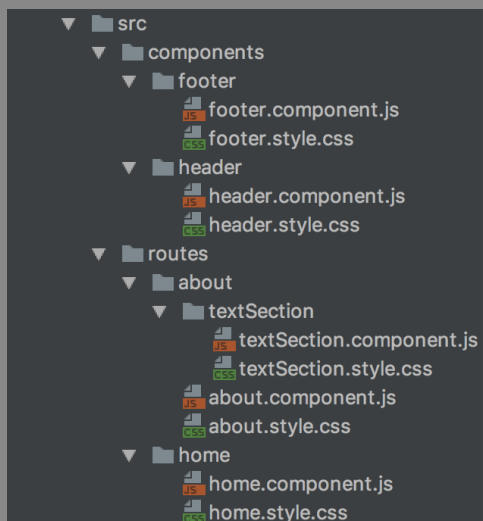
About Us

Contact

Komponenty podstron muszą być unikalne, wykorzystaj wspólne komponenty dla nagłówka i stopki.

Struktura plików

Propozycja



katalog: `components` zawiera re-używalne komponenty używane w aplikacji

katalog: `routes` zawiera główne komponenty podstron

w katalogów route'ów umieszczamy komponenty używane tylko w ich obrębie

Issue #2

Wykorzystaj aplikację z: `Issue #1`

Zastosuj w aplikacji bardziej czytelną strukturę.

Utwórz komponent `menu.component.js` zawierający linki do utworzonych podstron.

React Router

Obiekt

```
<Link />
```

Nazwa biblioteki: `react-router-dom`

Import obiektu:

```
import { Link } from 'react-router-dom';
```

React Router

Obiekt `<Link />`

Użycie obiektu `<Link />` :

```
import React, { Component } from 'react';
import { Link } from 'react-router-dom';

class App extends Component {

  render() {
    return (
      <div className="App">
        <Link to="/about">About</Link>
      </div>
    );
  }
}

export default App;
```

React Router

Dlaczego obiekt `<Link />` a nie `` ?

React Router

Dlaczego obiekt `<Link />` a nie `` ?

Użycie normalnego linku powoduje pełne prze-renderowanie strony oraz komponentów.

Issue #3

Wykorzystaj aplikację z: `Issue #2`

Wykorzystaj obiekt `<Link />` do utworzenia menu.

React Router

Przekazywanie parametrów

Definicja użycia parametrów:

```
<Route path="user/:id" component={User}/>
```

Przekazanie parametru

```
<Link to="users/user1">  
  User1  
</Link>
```

React Router

Przekazywanie parametrów

Przechowywanie parametrów z URL

```
{this.props.match.params.id}
```

Issue #4

Wykorzystaj aplikację z: `Issue #3`

Dodaj podstronę `Users`

W stopce utwórz link po kliknięciu w swoje nazwisko.

Wykorzystaj parametr w URL, żeby na podstronie `Users` wyświetlić przekazane id.

Issue #5

Wykorzystaj aplikację z: `Issue #4`

Utwórz komponent `usersList.component.js`, zrób tablicę obiektów zawierającą użytkowników (id oraz imię i nazwisko). Wyświetl użytkowników używając funkcji: `map()`

Issue #6

Wykorzystaj aplikację z: `Issue #5`

Utwórz linki użytkowników do podstrony: `Users` ,
która wyświetla id użytkownika.

React Router

Przekierowanie 404

Utworzenie nowego route'a:

```
<Route component={NotFound} />
```

Całkowity wygląd routera

```
<BrowserRouter>  
  <Switch>  
    <Route exact path="/" component={Home} />  
    <Route component={NotFound} />  
  </Switch>  
</BrowserRouter>
```

Issue #7

Wykorzystaj aplikację z: `Issue #6`

Utwórz komponent dla podstrony 404 i przekieruj do niego nieznalezione adresy URL.

React Router

Przekierowanie 404

Utworzenie nowego route'a:

```
<Redirect from='*' to='/' />
```


Issue #8

Wykorzystaj aplikację z: `Issue #7`

Utwórz przekierowanie dla nieistniejących adresów URL do strony startowej.

Issue #9

Wykorzystaj aplikację z: `Issue #8`

Utwórz przekierowanie dla nieistniejących adresów URL do podstrony 404, wykorzystując: `<Redirect />`