ECMA SCRIPT 6
REACT JS

Hello, World!

KAJ BIAŁAS
JavaScript Developer

Single Page Application

Czym jest?

SPA to technologia, która pozwala wyświetla poszczególne elementy strony, bez potrzeby ponownego załadowania całej strony.

React JS

Czym jest?

- biblioteka do budowania interfejsów użytkownika
- pozwala w łatwy sposób zbudować aplikację SPA
- nie jest framework'iem JavaScript

Create React App

Boilerplate

Create React App

Użycie

Utworzenie aplikacji w bieżącym katalogu

```
$ create-react-app .
```

Utworzenie aplikacji w nowym katalogu

```
$ create-react-app <folder_name>
```

Create React App

Komendy

Instalacja zależności (node_modules)

```
$ npm install
```

Uruchomienie serwera developerskiego

```
$ npm start
```

Create React App

Komendy

Zbudowanie paczki produkcyjnej

```
$ npm build
```

Więcej informacji:

- https://github.com/facebook/create-react-app

Issue #0

- Utwórz nowy katalog
- Utwórz w nim repozytorium GIT
- Uruchom create-react-app
- Spróbuj uruchomić serwer developerski

React JS

virtual dom

- wirtualna instancja całego drzewa
- przy zmianie porównywane są zmiany i wyszukiwane konkretne elementy

React JS

jsx

- jest to syntax, wykorzystywany przez React JS
- jest kompilowany do JavaScript'u

React JS

jsx

```
const App = () => (
  <div>
    <h1>Hello</h1>
  </div>
)
```

```
const App = () => (
  React.createElement(
    'div',
    {},
    React.createElement(
      'h1',
      {},
      'Hello'
    )
  )
)
```

React JS

react dom

Wykorzystując React w projekcie musimy załączyć

bibliotekę react oraz component

```
import React, { Component } from 'react';
```

React JS

komponenty

Komponent w React JS może być:

- funkcją
- klasą ES6 rozszerzoną o React.component

React JS

komponent funkcyjny

```
// “normal” function
function App(props) {
  return <div><h1>{props.title}</h1></div>
}

// arrow function - preferred way !
const App = props => <div><h1>{props.title}</h1></div>

// component call
const result = <App title="Hello world" />
```

React JS

class component

```
import React, { Component } from 'react';

class Header extends React.Component {
    render() {
      return <div><h1>{this.props.name}</h1></div>
    }
}

export default Header;
```

```
import React, { Component } from 'react';
import Header from './header.component.js';

class App extends Component {
    render() {
      return <div><Header name="XXX" /></div>
    }
}
```

Issue #5

Stwórz 2 dodatkowe komponenty Header i Footer, zaimplementuj je w aplikacji. Komponent Header ma wyświetlać nazwę aplikacji, natomiast Foter imię i nazwisko autora.

React JS

propertisy komponentu

- jest to syntax, wykorzystywany przez React JS
- jest kompilowany do JavaScript'u

React JS

stany komponentu

montowanie (ang. “mount”) - poniższe metody

- constructor()
- componentWillMount()
- render()
- componentDidMount()

React JS

stany komponentu

Odświeżanie (ang. “update”)

- componentWillReceiveProps()
- shouldComponentUpdate()
- componentWillUpdate()
- render()
- componentDidUpdate()

React JS

stany komponentu

Demontowanie (ang. “unmount”)

componentWillUnmount()

React JS

Komponent z call backiem

```
class App extends Component {
  state = {};

  componentWillMount() {
      this.setState({
          count: 0
      });
  }

  increseCounter = () => {
      this.setState({
          count: this.state.count + 1,
      });
  };

  render() {
    return (
      <div className="App">
          <Header handleClick={this.increseCounter} />
          <div>Current count: {this.state.count}</div>
      </div>
    );
  }
}
```

React JS

Komponent z call backiem

```
import React, { Component } from 'react';

class Header extends Component {

    render() {
        return (
            <header>
                <button onClick={this.props.handleClick}>+1</button>
            </header>
        );
    }
}

export default Header;
```

Issue #0

Stwórz strukturę:

- app controller z funkcją callback do przekazania stanu licznika

- header w którym będzie się znajdował button do zwiększenia stanu licznika

- footer w którym będzie się wyświetlał stan licznika

React JS

Pure Components

- nie mutuje danych komponentu
- podnosi perfomance aplikacji

https://codeburst.io/when-to-use-component-or-purecomponent-a60cfad01a81

React JS

PureComponents

```
import React, { PureComponent } from 'react';

class Header extends PureComponent {

    render() {
        return (
            <header>
                Header
            </header>
        );
    }
}

export default Header;
```

Issue #1

Zastąp dotychczasowe komponenty na pure-componenty

React JS

React Router

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


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

React JS

React Router

Należy doinstalować bibliotekę react-router-dom.

```
// index.js

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


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

Issue #2

Stwórz 3 podstorny:

Home, About Us, Contact

Dla każdej podstrony utwórz unikalną treść.

Nagłówek i stopka mają być identyczne na wszystkich podstronach. W nagłówku ma się wyświetlać tytuł unikalny podstrony i menu.

React JS

React Router - Link

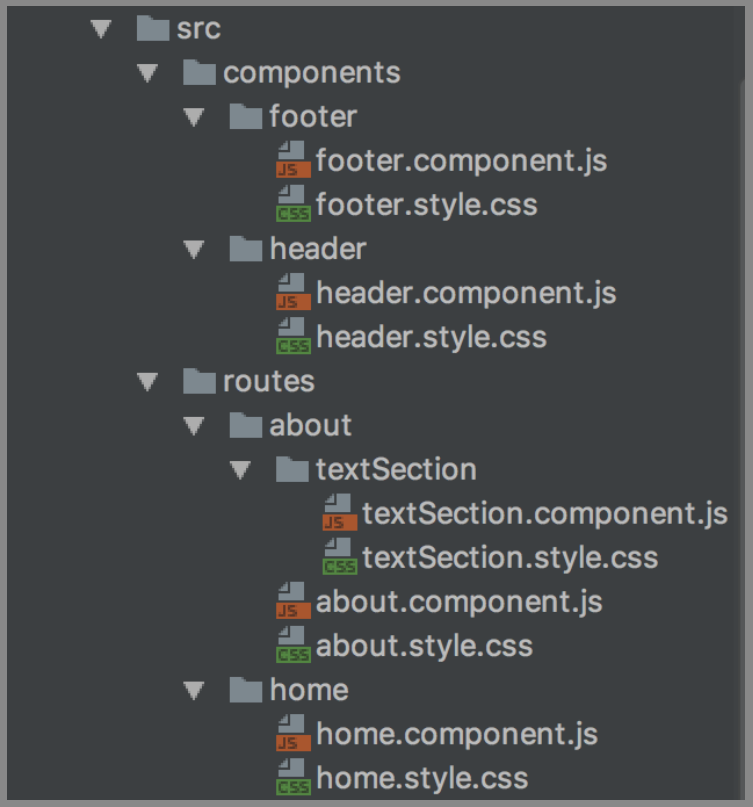
```
import React, { PureComponent } from 'react';
import { Link } from 'react-router-dom';


class App extends PureComponent {

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

export default App;
```

Issue #3

Zaimplementuj użycie componentu Link.

React JS

Struktura projektu



React JS

Api Request

```
constructor(props) {
    super(props);
    this.state = {
        error: null,
        isLoaded: false,
        items: []
    };
}
```

```
componentWillMount() {
    fetch('http://nflarrest.com/api/v1/team')
        .then(res => res.json())
            .then(
                (result) => {
                    console.log('result');
                    this.setState({
                        isLoaded: true,
                        items: result
                    });
                    console.log(this.state);
                },
                (error) => {
                    this.setState({
                        isLoaded: true,
                        error
                    });
                }
        )
}
```

```
 error, isLoaded, items } = this.state;
(error) {
 return <div>Error: {error.message}</div>;
lse if (!isLoaded) {
 return <div>Loading...</div>;
lse {
 return (
    <ul>
        {items.map(item => (
            <li key={item.Team}>
                {item.Team} {item.Team_name}
            </li>
        ))}
    </ul>
);
```

Issue #4

Utwórz podstronę Crime, która będzie zawierała informacje o przestępstwach, pobrane z API

React JS

Material UI

http://www.material-ui.com/#/components/

```
import React, { PureComponent } from 'react';
import { MuiThemeProvider, FlatButton } from 'material-ui';

class Material extends PureComponent {

    render() {
        return (
            <MuiThemeProvider>
                <FlatButton label="Default" />
                <FlatButton label="Primary" primary={true} />
                <FlatButton label="Secondary" secondary={true} />
                <FlatButton label="Disabled" disabled={true} />
            </MuiThemeProvider>
        );
    }
}

export default Material;
```

Issue #5

Użyj material-ui w dotychczasowym projekcie