Podstawy web developmentu: React i jego zastosowania

PUT, 14.04.2016

_PODSTAWY
ECMAScript 2015

_CECHY JĘZYKA ECMAScript

- Brak silnego typowania
- Język interpretowany
- Zarówno paradygmat funkcyjny jak i imperatywny
- Opcjonalne średniki
- Hoisting
- Dziedziczenie prototypowe

_ZMIENNE

```
function foo() {
   var variable = false;
   let variableArray = [];
   const constVariable = 2;

   if (variable) {
       let otherVariable = "Hello,
World!";
   } else {
       let otherVariable = {};
   }
}
```

_FUNKCJE

```
function foo(name) {
   return `Hello, ${name}`;
}

let bar = function (name) {
   return `Hello, ${name}`;
};

let baz = (name) => {
   return `Hello, ${name}`;
};

let foo2 = (name) => `Hello, ${name}`;
```

_KLASY

```
class App extends Component {
    constructor(name) {
        super();

        this.name = name;
    }

    foo() {
        return `Hello, ${this.name}!`;
    }

    static bar(name) {
        return `Hello ${name}!`;
    }
}
```

```
App.bar('Wojtek');
// Hello, Wojtek!

const app = new App('Wojtek');
app.foo();
// Hello, Wojtek!
```

_MODUŁY

```
//main.js
import React from 'react';
import App, {FOO_BAR} from './app';

const app = new App();
app.start(FOO_BAR);

//app.js
export default class App {
   start(value) {
       return `Hello, ${value}!`;
   }
}

export const FOO_BAR = 'foo-bar';
```

_PACKAGE MANAGER – npm

- package.json
- npm install [--save] [--save-dev]
- npm update [--save] [--save-dev]
- npm run <script-name>
- node_modules

_PODSTAWY
ReactJS

_ReactJS

- Single Page Application (SPA)
- Biblioteka do renderowania widoków
- Virtual DOM
- Wszystko opiera się na komponentach
- Aplikację można renderować na serwerze w NodeJS
- Pozwala na dowolny model danych

_DEFINICJA KOMPONENTU

```
import React from 'react';

export default class App extends React.Component {
   render() {
       return (
           <div>Hello, World!</div>
       );
   }
}
```

_RENDEROWANIE KOMPONENTU

```
//main.js
import ReactDOM from 'react-dom';
import App from './app';

ReactDOM.render(<App/>, document.getElementByid('react-root'));
```

```
//index.html
<html>
   <body>

       <div id="react-root"></div>

   </body>
</html>
```

_PODSTAWY JSX

_JSX

```
const render = () => {
  return (
     <div>Hello, World!</div>
  );
};
```

_JSX

```
const render = () => {
  const name = 'Wojtek';
  return (
     <div>Hello, {name}!</div>
  );
};
```

_JSX

```
const render = () => {
  let condition = true;
  const name = 'Wojtek';
  const nameElement = condition ? <span>{name}</span> : null;

  return (
     <div>
        Hello, {nameElement}!
     </div>
  );
};
```

_JSX

```
const render = () => {
  const cssClasses = 'class1 class2';
  return (
    <div>
      <div className="class3">Hello, World!</div>
      <div className={cssClasses}>Hello, World!</div>
    </div>
  );
};
```

_JSX

```
const render = () => {
  const onInputChange = (event) => {
     // deal with event
  };

  return (
     <input onChange={onInputChange}/>
  );
};
```

_MAP, FILTER, FIND

_FILTER

```
const data = [1, 2, 3, 4];

const getEven = () => {
  return data.filter((item) => item % 2 === 0)
};

// [2, 4]
```

_FIND

```
const data = [{id: 1}, {id: 2}, {id: 3}];

const getById = (id) => {
  return data.find((item) => item.id === id)
};

getById(2);
// {id: 2}
```

_MAP

```
const render = () => {
  const data = ['foo', 'bar', 'baz'];

  const listElements = data.map((item, index) => {
     return <li key={index}>{item}</li>
  });

  return (
     <ul>
        {listElements}
     </ul>
  );
};
```

_STAN KOMPONENTU

_STAN KOMPONENTU

```
export default class App extends React.Component {
  constructor(props) {
      super(props);
      this.state = {
        seconds: 0
      }
  }

  componentDidMount() {
      setInterval(() => {
        this.setState({
          seconds: this.state.seconds + 1
        });
      }, 1000);
  }

  render() {
      return (
        <div> {this.state.seconds} seconds elapsed! </div>
      );
  }
}
```

_PLAN DZISIEJSZYCH WARSZTATÓW

_REPOZYTORIUM

Repozytorium znajdziecie klikając w poniższy link:

https://github.com/apptension/cs-put-react

_COMPONENT LIFECYCLE

_CYKL ŻYCIA KOMPONENTU

```
class LifecycleComponent extends React.Component {
  componentWillMount() {
     console.log('Component WILL MOUNT!')
  }

  componentDidMount() {
     console.log('Component DID MOUNT!')
  }

  componentWillReceiveProps(newProps) {
     console.log('Component WILL RECEIVE PROPS!')
  }

  componentWillUpdate(nextProps, nextState) {
     console.log('Component WILL UPDATE!');
  }

  componentWillUnmount() {
     console.log('Component WILL UNMOUNT!')
  }
}
```

_PROPS

_PRZEKAZYWANIE PROPSÓW

```
render() {
  const value1 = 'value';
  const value2 = 5;
  return <ExampleComponent exampleProp1={value1}  exampleProp2={value2} />;
}
```

_UŻYWANIE PROPSÓW

```
render() {
  const title = `Hello, ${this.props.exampleProp1}`;
  return <div>{title}</div>;
}
```

_FETCH & PROMISES

_PODSTAWY PROMISE’ÓW

```
const promise = new Promise((resolve) => {
  setTimeout(() => {
     resolve('Hello, World! 5 seconds elapsed!');
  }, 5000);
});

promise.then((msg) => {
  console.log(msg);
});
```

_ODRZUCENIE PROMISE’A

```
const condition = false;
const promise = new Promise((resolve, reject) => {
   setTimeout(() => {
      if (condition) {
         resolve();
      } else {
         reject();
      }
   }, 500);
});

promise.then(() => {
   console.log('This will not be called');
}, () => {
   console.log('This will be called!');
});
```

_ŁĄCZENIE PROMISE’ÓW

```
const promise = new Promise((resolve) => {
  setTimeout(() => {
      resolve();
  }, 500);
});

promise
 .then(() => {
  return new Promise((resolve) => {
      console.log('Inner promise');
      resolve();
  });
 })
 .then(() => {
    console.log('Outer promise')
 });

// Inner promise
// Outer promise
```

_CZEKANIE NA WIELE PROMISE’ÓW

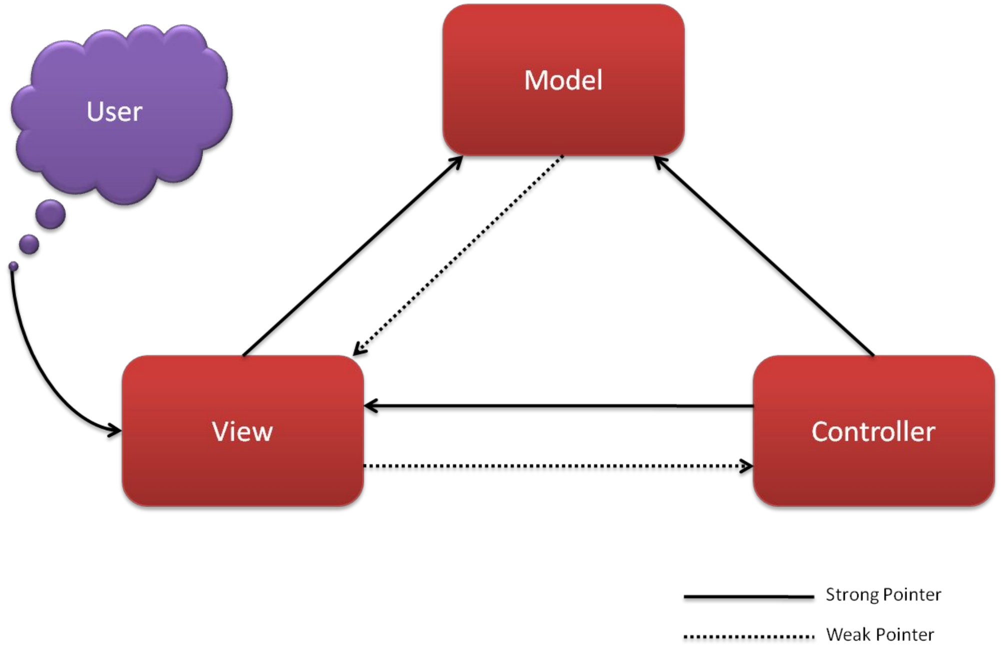
```
const promise1 = new Promise((resolve) => {
  setTimeout(() => {
     resolve(2);
  }, 500);
});

const promise2 = new Promise((resolve) => {
  setTimeout(() => {
     resolve(1);
  }, 5000);
});

const allPromise = Promise.all([promise1, promise2]);
allPromise.then((data) => {
  console.log('Will be called after 5s', data);
});
// Will be called after 5s [1, 2]
```

_FETCH

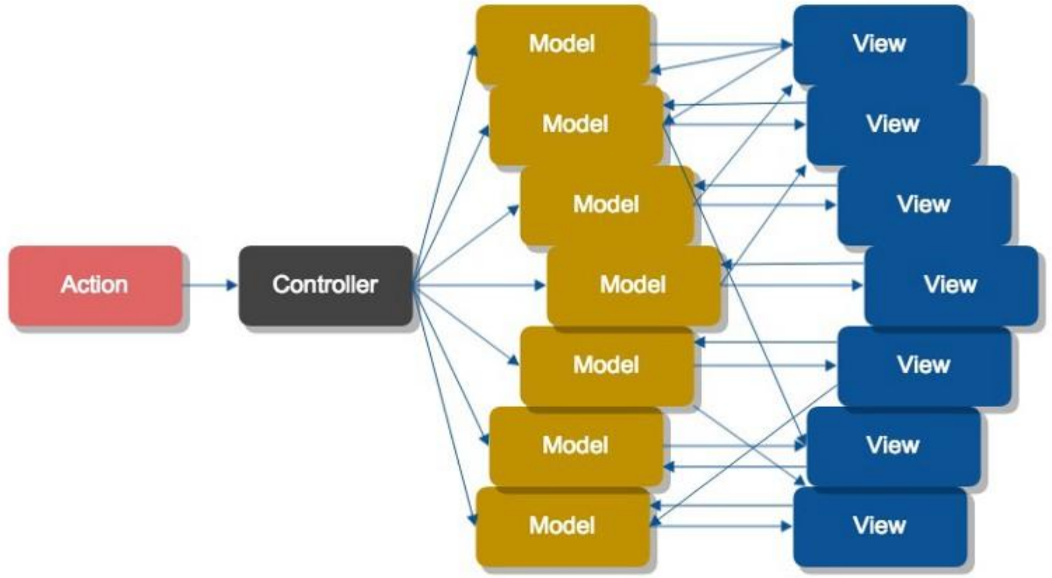
```
fetch('http://some-cool-api.com/users')
 .then((data) => data.json())
 .then((json) => console.log('Json data:', json));
```

_MVC

_MODEL VIEW CONTROLLER

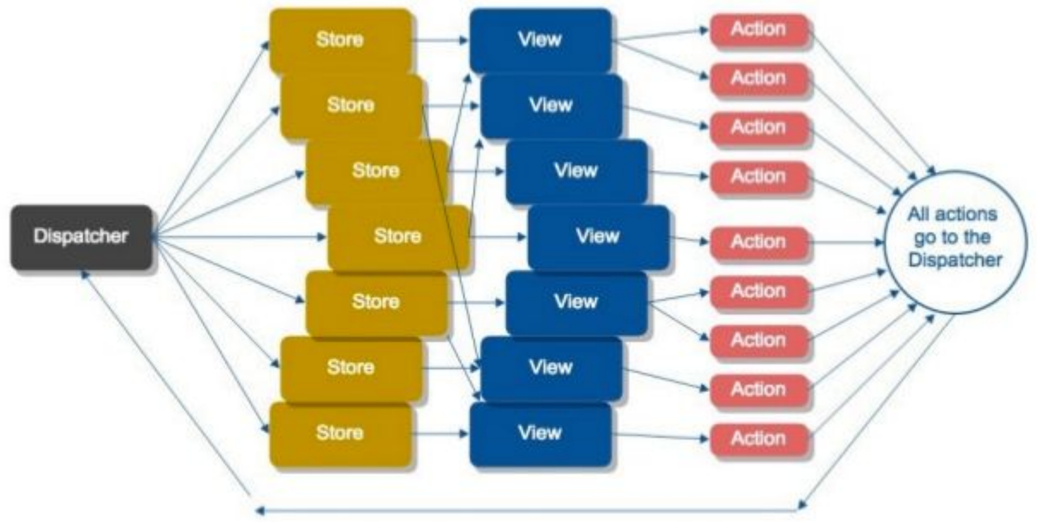


_MODEL VIEW CONTROLLER ISSUES



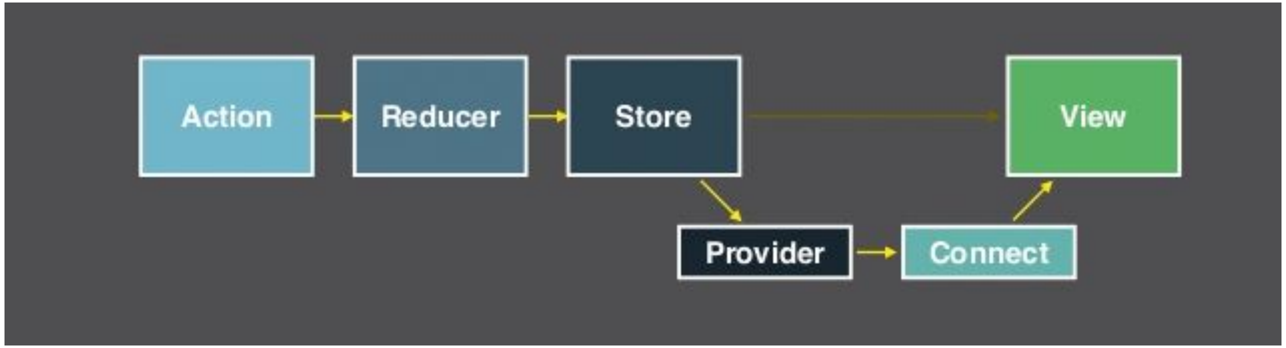
_FLUX

_PRZEPŁYW JEDNOKIERUNKOWY



_REDUX

_REDUX



_PODŁĄCZENIE REDUXA DO APLIKACJI

```
import {Provider} from 'react-redux';
import configureStore from './src/store/configureStore';

const store = configureStore();
const reactRoot = document.getElementById('react-root');

ReactDom.render(
  <Provider store={store}>
     <App/>
  </Provider>
  , reactRoot
);
```

_KONFIGURACJA STORE’A

```
import {createStore, applyMiddleware} from 'redux';
import thunk from 'redux-thunk';
import createLogger from 'redux-logger';


export default function configureStore(initialState) {
  return createStore(
     reducers,
     initialState,
     applyMiddleware(thunk, createLogger())
  );
}
```

_AKCJA

```
export const SOME_ACTION = 'SOME_ACTION';

export function someAction(value) {
  return {
     type: SOME_ACTION,
     payload: value
  };
}
```

_AKCJA ASYNCHRONICZNA

```
export const SOME_ACTION_STARTED = 'SOME_ACTION_STARTED';
export const SOME_ACTION_FINISHED = 'SOME_ACTION_FINISHED';

export function someAction() {
  return (dispatch) => {
     dispatch({
        type: SOME_ACTION_STARTED
     });

     setTimeout(() => {
        dispatch({
           type: SOME_ACTION_FINISHED,
           payload: 'some-result'
        })
     }, 5000);
  };
}
```

_REDUCER

```
import {SOME_ACTION} from '../actions';

const initialState = {
  someProp: ''
};

export default function charactersList(state = initialState, action = null) {
  switch (action.type) {
    case SOME_ACTION:
      return Object.assign({}, state, {
        someProp: action.payload
      });
    default:
      return state;
  }
}
```

_PODŁĄCZANIE KOMPONENTU DO STORE’A

```
import React from 'react';
import {connect} from 'react-redux';
import {someAction} from '../../actions';

class App extends React.Component {
  componentWillMount() {
     this.props.someAction('some value');
  }

  render() {
     return <p>{this.props.someProp}</p>
  }
}

function mapStateToProps(state) {
  return {
     someProp: state.somethingFromStore.someProp
  };
}

export default connect(
  mapStateToProps,
  {someAction}
)(App);
```