Viac ťahákov nájdeš na

bart.sk/tahaky

Potrebujete pomôcť?

podpora@bart.sk | www.bart.sk

REDUX

Pokročilý manažment stavu aplikácie

- **Redux DevTools**
  Pri vývoji sa ti určite zíde výborný nástroj, s ktorým získaš prehľad o všetkom, čo sa deje v store.
- **redux-thunk**
  Použi ho ak potrebuješ dispatch-ovať asynchrónne akcie ako napríklad volania REST API.
- **redux-form**
  Prepojenie formulárových prvkov s reduxom vrátane validácie.

Pozor na referencie! Reducer by mal upraviť state len tak, že zmení referenciu objektu. Nemal by si preto používať mutable operácie ako je napríklad Array.prototype.push(). Pomôže ti knižnica **Immutable.js** alebo natívny **Spread Operator**.

STYLED COMPONENTS

Integrácia CSS priamo do JavaScript-u

CSS komponent

```
const Title = styled.h1`
  color: black;
  font-size: 14px;
  ${props => `font-size: ${props.size}px`};
`;
```

Použitie CSS komponentu

```
render() {
  return <Title size={24} />;
}
```

Theme Provider
Silný nástroj, ktorý ti pomôže udržiavať styleguide naprieč celou aplikáciou.

VYCHYTÁVKY

- **Router**
  Deklaratívne routovanie pre tvoju appku.

  ```
  <Route path="/product/:id" component={Product} />
  ```

- **create-react-app**
  Zabezpečí všetko potrebné, aby si sa ty mohol venovať vývoju. *bart.sk/create-react-app*

  ```
  npx create-react-app my-app
  ```

- **Server Side Rendering**
  Efektívne zrýchli načítanie stránky a umožní ti robiť SEO. *bart.sk/react-dom-server*
- **React Native**
  Pomocou React-u môžeš vytvoriť skutočné natívne mobilné aplikácie. *bart.sk/react-native*

UŽITOČNÉ ODKAZY

Oficiálna React dokumentácia
bart.sk/react-docs

Povinné čítanie
bart.sk/thinking-in-react

Skvelý nástroj na testovanie
bart.sk/jest

Doplnok do Chrome pre debuggovanie React aplikácií
bart.sk/react-chrome

ECMAScript 6
bart.sk/es6-tutorial

Komponent a kontajner
bart.sk/component-container

redux-thunk
bart.sk/redux-thunk

redux-form
bart.sk/redux-form

Redux
bart.sk/redux

Styled components
bart.sk/styled-components

Polished.js - knižnica funkcií pre styled-components
bart.sk/polished

REACT
ťahák

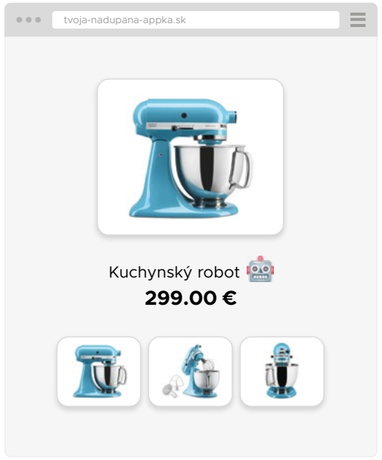
bart

bart.sk/tahaky

AKO TO FUNGUJE?

```
import React, {Component, Fragment} from 'react';
class Product extends Component {
  render() {
    const {img, name, gallery, price} = this.props;
    return (
      <Fragment>
        {img && <Image src={img.src} alt="..." />}
        {name}
        <Price amount={price.amount} currency="EUR" />
        {gallery.map(img => <Image src={img.src} alt="..." />)}
      </Fragment>
    );
  }
}
```

Ukážka kódu



Vykreslenie kódu v prehliadači

PROPS

Vstupy pre tvoj komponent

Môže to byť takmer čokoľvek. Napríklad číslo alebo iný komponent.

Názvy a typy props zadefinuj v propTypes.
Vstupy tak budeš mať pod kontrolou a kód lepšie zdokumentovaný.

```
import PropTypes from 'prop-types';
Product.propTypes = {
  name: PropTypes.string.isRequired,
  price: PropTypes.shape({
    currency: PropTypes.string,
    amount: PropTypes.number,
  }),
};
```

Rovnako môžeš zadefinovať predvolené hodnoty pre props v defaultProps.

```
Product.defaultProps = {
  price: {
    currency: "EUR",
    amount: 0.00,
  },
};
```

STATE

Objekt, ktorý predstavuje stav komponentu

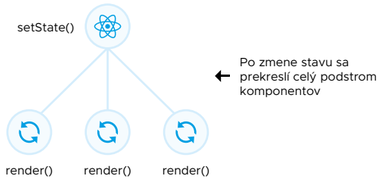
Zmenu v state vykonávaj vždy cez funkciu...

```
this.setState({isLoading: true});
```

...nikdy nie priamo cez

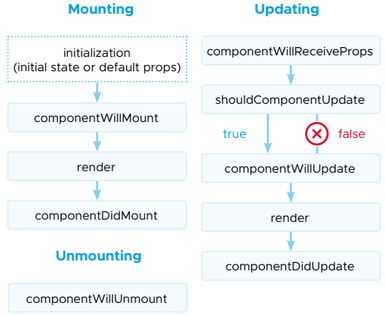
```
this.state.isLoading = true;
```

Dávaj bacha! Zmena stavu môže byť asynchrónna.



Po zmene stavu sa prekreslí celý podstrom komponentov

ŽIVOTNÝ CYKLUS KOMPONENTU



KOMPONENT A KONTAJNER

Kontajner
Kontajnery vedia všetko o dátach, napríklad odkiaľ prichádzajú a aký majú tvar. Poznajú procesy (biznis logiku) aplikácie. Transformujú dáta do priamejšej podoby pre vizuálne komponenty.

(Vizuálny, prezentačný) komponent
Komponent, ktorý rieši iba vzhľad. Obvykle neobsahuje vlastný state a pracuje iba s props, ktoré mu poskytne napr. kontajner.

```
src/
├── components/
│   ├── Product.jsx
│   ├── Input.jsx
│   └── Img.jsx
└── containers/
    ├── App.jsx
    ├── ProductDetail.jsx
    └── ProductList.jsx
    ...
```

Príklad štruktúry v projekte