В этом уроке мы с вами подготовим данные для нашей страницы корзины.

Для начала давайте добавим роут /basket в роутер. Basket роут у нас не находится в Layout, так как будет обладать отдельной разметкой

ReactDOM.render(

<Provider store={store}>

<Router history={history}>

<Route component={Layout}>

<Route path='/' component={Phones} />

<Route path='/categories/:id' component={Phones} />

</Route>

<Route path='/phones/:id' component={Phone} />

<Route path='/basket' component={Basket} />

</Router>

</Provider>,

document.getElementById('root')

);

И импортируем контейнер basket.

import Basket from 'containers/basket'

Теперь создадим компонент

import React from 'react'

const Basket = () => {

return (

<div>Basket</div>

)

}

export default Basket

Если мы посмотрим в браузер и кликнем по ссылке на корзину, то у нас отрендерится наш компонент.

Давайте теперь добавим connect и mapStateToProps

import {connect} from 'react-redux'

const mapStateToProps = state => {

return {

phones: getBasketPhonesWithCount(state),

totalPrice: getTotalBasketPrice(state)

}

}

export default connect(mapStateToProps, null)(Basket)

В mapStateToProps нам нужно найти телефоны, сгруппированные по товару и общую цену. Сейчас у нас в корзине хранится плоский массив idшников товаров, которые могут повторятся сколько угодно раз. Мы же хотим получить список телефонов, сгруппированых по id и имеющих общее количество каждого товара.

Селектор getTotalBasketPrice у нас уже есть, так как мы его писали ранее. Давайте добавим селектор getBasketPhonesWithCount.

Для начала найдем массив уникальных idшников из нашей корзины и получим все телефоны, соответствующие этим idшникам.

export const getBasketPhonesWithCount = state => {

const uniqueIds = R.uniq(state.basket)

const phones = R.compose(

R.map(id => getPhoneById(state, id))

)(uniqueIds)

return phones

}

Теперь нам нужно к каждому телефону добавить поле count. Для этого применим еще один map и внутри функции phoneWithCount добавим в каждый телефон поле count, вызвав метод phoneCount. Этот метод будет возвращать количество телефонов по id. Внутри метод phoneCount мы фильтруем все idшники в basket по текущему id и находим их количество.

export const getBasketPhonesWithCount = state => {

const phoneCount = id => R.compose(

R.length,

R.filter(basketId => R.equals(id, basketId))

)(state.basket)

const phoneWithCount = phone => R.assoc('count', phoneCount(phone.id), phone)

const uniqueIds = R.uniq(state.basket)

const phones = R.compose(

R.map(phoneWithCount),

R.map(id => getPhoneById(state, id))

)(uniqueIds)

return phones

}

Давайте добавим разметку в нашу корзину. Создадим дополнительную переменную isBasketEmpty, чтобы понимать какие куски разметки выводить если товары в корзине есть, а какие, когда корзина пустая.

const Basket = ({phones, totalPrice}) => {

const isBasketEmpty = R.isEmpty(phones)

const renderContent = () => {

return (

<div>

{isBasketEmpty && <div>Your shopping cart is empty</div>}

<div className='table-responsive'>

<table className='table-bordered table-striped table-condensed cf'>

<tbody>

{phones.map((phone, index) => (

<tr

key={index}

className='item-checout'

>

<td className='first-column-checkout'>

<img

className='img-thumbnail'

src={phone.image}

alt={phone.name}

/>

</td>

<td>{phone.name}</td>

<td>${phone.price}</td>

<td>{phone.count}</td>

<td>

<span

className='delete-cart'

/>

</td>

</tr>

))}

</tbody>

</table>

</div>

{

R.not(isBasketEmpty) &&

<div className='row'>

<div className='pull-right total-user-checkout'>

<b>Total:</b>

${totalPrice}

</div>

</div>

}

</div>

)

}

const renderSidebar = () => (

<div>

Sidebar

</div>

)

return (

<div className='view-container'>

<div className='container'>

<div className='row'>

<div className='col-md-9'>

{renderContent()}

</div>

<div className='col-md-3 btn-user-checkout'>

{renderSidebar()}

</div>

</div>

</div>

</div>

)

}

Если мы посмотрим в браузер, то у нас выводятся товары, которые мы добавили в корзину и общая цена. Если же товаров нет, то выводится сообщение, что корзина пустая. При перезагрузке страница очищается, так как данные мы храним только в памяти, а не на бекенде или в localStorage.

Если у вас что-то не получается, либо возникли вопросы и комментарии, пишите их прямо под этим видео и я обязательно на них отвечу.