В этом уроке мы с вами реализуем вывод корзины на странице списка товаров и отдельного товара. Это нужно для того, чтобы пользователь всегда видел сколько товаров у него добавлено в корзину и на какую сумму.

У нас на каждом товаре есть кнопка buy now. Давайте добавим при клике екшен добавления товара в корзину.

<button

className='btn btn-primary'

onClick={() => addPhoneToBasket(phone.id)}

>

При клике мы выстреливаем метод addPhoneToBasket, в который передаем phone.id.

Мы должны его также деструктурировать из this.props

renderPhone (phone, index) {

const {addPhoneToBasket} = this.props

const shortDescription = `${R.take(60, phone.description)}...`

return (

...

)

}

Теперь давайте импортируем его из actions и добавим в mapDispatchToProps

import {

fetchPhones,

loadMorePhones,

addPhoneToBasket

} from 'actions'

const mapDispatchToProps = {

fetchPhones,

loadMorePhones,

addPhoneToBasket

}

И добавим actionType

export const ADD\_PHONE\_TO\_BASKET = 'ADD\_PHONE\_TO\_BASKET'

Импортируем новый actionType в actions и опишем этот екшен

export const addPhoneToBasket = id => dispatch => {

dispatch({

type: ADD\_PHONE\_TO\_BASKET,

payload: id

})

}

Теперь нам нужно поймать этот екшен в редьюсере. Для этого давайте создадим новый редьюсер basket, который будет хранить только idшники телефонов, которые мы добавили в корзину.

import R from 'ramda'

import {

ADD\_PHONE\_TO\_BASKET

} from 'actionTypes'

const initialState = []

export default (state = initialState, {type, payload}) => {

switch (type) {

case ADD\_PHONE\_TO\_BASKET:

return R.append(payload, state)

default:

return state

}

}

Начальный стейт у нас пустой массив, а когда срабатывает action ADD\_PHONE\_TO\_BASKET, то мы добавляем в наш массив новый id с помощью метода Ramda append.

Теперь давайте добавим редьюсер в combineReducers.

import {combineReducers} from 'redux'

import {routerReducer} from 'react-router-redux'

import phones from './phones'

import phonesPage from './phonesPage'

import phonePage from './phonePage'

import basket from './basket'

export default combineReducers({

routing: routerReducer,

phones,

phonesPage,

phonePage,

basket

})

Если мы посмотрим в браузер и кликнем на add to basket, то увидим, что в наш редьюсер корзины добавился новый id.

Давайте добавим компонент Sidebar, который будет рендериться на странице списка товаров в папку /components/sidebar/index

import React from 'react'

const Sidebar = () => {

return (

<div>

Sidebar

</div>

)

}

export default Sidebar

И подключим его в наш Layout.

import Sidebar from 'components/sidebar'

и вставим

Теперь давайте добавим новый компонент для отображения корзины. Это будет наш первый sharable компонент, который мы будем рендерить на странице товаров и отдельного товара.

import React from 'react'

import {Link} from 'react-router'

const BasketCart = () => (

<div className='cart'>

<div className='dropdown'>

<Link

to='/basket'

id='dLabel'

className='btn btn-inverse btn-block btn-lg'

>

<i className='fa fa-fa-shopping-cart' />

<span>Prices</span>

</Link>

</div>

</div>

)

export default BasketCart

Давайте импортируем его в Sidebar.

import React from 'react'

import BasketCart from 'components/basketCart'

const Sidebar = () => {

return (

<div>

<BasketCart />

</div>

)

}

export default Sidebar

Теперь мы хотим в нашем компоненте вывести Общую цену и количество айтемов. Для этого нам понадобится connect

import {connect} from 'react-redux'

const mapStateToProps = state => {

return {

totalBasketCount: getTotalBasketCount(state),

totalPrice: getTotalBasketPrice(state)

}

}

export default connect(mapStateToProps, null)(BasketCart)

У нас будет два специальный метода в selectors, который будут подсчитывать количество товаров и его общую цену.

Давайте теперь добавим эти методы.

Для того, чтобы посчитать общее количество товаров в корзине достаточно взять length от массива idшников в корзине

export const getTotalBasketCount = state => R.length(state.basket)

getTotalBasketPrice немного сложнее

export const getTotalBasketPrice = state => {

const totalPrice = R.compose(

R.sum,

R.pluck('price'),

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

)(state.basket)

return totalPrice

}

Мы находим все телефоны по idшникам в корзине и берем из каждого обьекта только цену и потом суммируем ее с помощью метода sum.

Теперь мы можем вместо слова Prices вывести

const BasketCart = ({totalBasketCount, totalPrice}) => (

<div className='cart'>

<div className='dropdown'>

<Link

to='/basket'

id='dLabel'

className='btn btn-inverse btn-block btn-lg'

>

<i className='fa fa-fa-shopping-cart' />

<span>{totalBasketCount} item(s) - ${totalPrice}</span>

</Link>

</div>

</div>

)

Если мы посмотрим в браузер, то у нас полностью работает вывод информации по количеству товаров.

Теперь давайте добавим сайдбар в страницу отдельного товара. Так как там сайдбар будет маленький, то выносить его в отдельный компонент не будем.

renderSidebar () {

const {phone, addPhoneToBasket} = this.props

return (

<div>

<p className='lead'>Quick shop</p>

<BasketCart />

<div className='form-group'>

<h1>{phone.name}</h1>

<h2>${phone.price}</h2>

</div>

<Link to='/' className='btn btn-info btn-block'>Back to store</Link>

<button

type='button'

className='btn btn-success btn-block'

onClick={() => addPhoneToBasket(phone.id)}

>

Add to cart

</button>

</div>

)

}

Нам нужно также импортировать наш екшен и react-router-redux и добавить екшен в mapDispatchToProps.

import {Link} from 'react-router'

import {fetchPhoneById, addPhoneToBasket} from 'actions'

import BasketCart from 'components/basketCart'

const mapDispatchToProps = {

fetchPhoneById,

addPhoneToBasket

}

Если мы посмотрим в браузер, то у нас вывелся наш компонент, а также короткая информация о телефоне и кнопка назад и добавить в корзину.

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