В этом уроке мы с вами подключим к проекту react-router, redux и выведем главную страницу.

Для начала небольшой нюанс. Мы хотим все файлы импортировать не с относительными путями как

import module from '../../module'

а как node модули.

import module from 'module'

и проще всего это сделать положив в папку src папку node\_modules, куда мы будем класть все файлы проекта. Это самый простой вариант, который работает на всех операционных системах.

Давайте для начала создадим redux store

import './main.css'

import {createStore} from 'redux'

const store = createStore()

Теперь в createStore нам нужно передать рутовый редьюсер нашего проекта. Все наши редьюсеры будут лежать в папке src/node\_modules/reducers

Давайте создадим папку и index файл. Здесь у нас будет происходить combineReducers и мы их тут будем експортировать и передавать на вход в store. Если вы не помните, что такое combineReducers, то можете посмотреть в моем видео ["combineReducers в Redux"](http://monsterlessons.com/project/lessons/reduxjs-combinereducers).

Давайте создадим один из редьюсеров, который понадобится нам позже. Так как у нас магазин телефонов, то у нас будет сущность телефоны.

Давайте добавим редьюсер phones.

const initialState = {}

export default (state = initialState) => {

return state

}

Вот у нас пустой редьюсер с initialState в виде обьекта. Теперь давайте импортируем его в рутовом редьюсере и вернем все наши редьюсеры с помощью combineReducers.

import {combineReducers} from 'redux'

import phones from './phones'

export default combineReducers({

phones

})

Напомню, что combineReducers позволяет комбинировать редьюсеры в один. И теперь мы можем просто добавлять сколько угодно новых редьюсеров в этой папке.

Теперь давайте передадим их в наш store.

import './main.css'

import {createStore} from 'redux'

import reducers from 'reducers'

const store = createStore(reducers)

Обратите внимание, что из-за структуры с папкой node\_modules мы можем обращатся к нашим редьюсерам напрямую без абсолютных или относительных путей.

Теперь нам нужно передать middleware в наш store вторым параметром. Из middleware нам понадобятся redux-devtools, для удобного дебага и redux-thunk. Если вы не помните что это и зачем нужно, то я оставлю ссылки в тексте видео.

[Redux-devtools](http://monsterlessons.com/project/lessons/reduxjs-redux-devtools)

[Redux-thunk](http://monsterlessons.com/project/lessons/reduxjs-asinhronnye-eksheny-s-pomoshyu-redux-thunk)

Импортируем redux-thunk и redux-devtools-extension, а также applyMiddleware из redux.

import './main.css'

import {createStore, applyMiddleware} from 'redux'

import {composeWithDevTools} from 'redux-devtools-extension'

import thunk from 'redux-thunk'

import reducers from 'reducers'

const store = createStore(reducers, composeWithDevTools(

applyMiddleware(thunk)

))

Теперь давайте подключим react-router-redux. Ссылка на видео зачем он нужен опять же в описании.

[React-router-redux](http://monsterlessons.com/project/lessons/reduxjs-routing-s-pomoshyu-react-router-redux)

Для этого создадим history, который потом будем прокидывать в react-router.

import './main.css'

import {createStore, applyMiddleware} from 'redux'

import {composeWithDevTools} from 'redux-devtools-extension'

import thunk from 'redux-thunk'

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

import {browserHistory} from 'react-router'

import reducers from 'reducers'

const store = createStore(reducers, composeWithDevTools(

applyMiddleware(thunk)

))

const history = syncHistoryWithStore(browserHistory, store)

Так как мы добавили react-router-redux, то мы должны добавить и его редьюсер, чтобы он мог работать

import {combineReducers} from 'redux'

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

import phones from './phones'

export default combineReducers({

routing: routerReducer,

phones

})

Теперь давайте напишем роутер и передадим туда store и history. Но для начала нам нужно импортировать React и ReactDOM.

import './main.css'

import React from 'react'

import ReactDOM from 'react-dom'

import {createStore, applyMiddleware} from 'redux'

import {composeWithDevTools} from 'redux-devtools-extension'

import thunk from 'redux-thunk'

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

import {Router, Route, browserHistory} from 'react-router'

import {Provider} from 'react-redux'

import reducers from 'reducers'

const store = createStore(reducers, composeWithDevTools(

applyMiddleware(thunk)

))

const history = syncHistoryWithStore(browserHistory, store)

ReactDOM.render(

<Provider store={store}>

<Router history={history}>

</Router>

</Provider>,

document.getElementById('root')

);

Мы создали роутер и передали в него history. Также предварительно мы обернули роутер в Provider, чтобы прокинуть store.

Теперь давайте поговорим о роутах какие нам нужны. Мы хотим создать Layout, который будет использован как стандартная обертка вокруг любой страницы, содержащая в себе место для контента и сайтбар. Также давайте добавим роут для страницы с списком телефонов.

<Router history={history}>

<Route component={Layout}>

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

</Route>

</Router>

Теперь нам нужно добавить компоненты Layout и Phones. Все компоненты страниц мы будем хранить в папке containers.

Создадим компонент Layout

import React from 'react'

const Layout = ({children}) => (

<div>

{children}

</div>

)

export default Layout

и компонент Phones

import React from 'react'

const Phones = () => (

<div>

Phones

</div>

)

export default Phones

Если мы посмотрим в браузер, то увидим, что у нас вывелось слово Phones, а в вкладке Redux-devtools доступен просмотр состояния. Это значит, что настройка проекта успешно завершена.

Эту заготовку можно копировать из проекта в проект, так как она меняется лишь минимально.

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