В этом уроке мы с вами подготовим данные для категорий и фильтра товаров по категориям. Для начала нам нужно добавить роут для категории.

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}/>

</Router>

</Provider>,

document.getElementById('root')

);

Мы добавили роут для конкретной категории и будем на ней использовать все тот же компонент Phones. Теперь нам необходимо зафетчить категории при загрузке приложения.

Так как эти данные нам нужны только в компоненте Phones, то давайте там их и фетчить

componentDidMount () {

this.props.fetchPhones()

this.props.fetchCategories()

}

Теперь импортируем екшен и добавим его в mapDispatchToProps.

import {

fetchPhones,

loadMorePhones,

addPhoneToBasket,

fetchCategories

} from 'actions'

const mapDispatchToProps = {

fetchPhones,

loadMorePhones,

addPhoneToBasket,

fetchCategories

}

Опишем actionTypes.

export const FETCH\_CATEGORIES\_START = 'FETCH\_CATEGORIES\_START'

export const FETCH\_CATEGORIES\_SUCCESS = 'FETCH\_CATEGORIES\_SUCCESS'

export const FETCH\_CATEGORIES\_FAILURE = 'FETCH\_CATEGORIES\_FAILURE'

Импортируем екшен тайпы и опишем наш екшен

import {

FETCH\_PHONES\_START,

FETCH\_PHONES\_SUCCESS,

FETCH\_PHONES\_FAILURE,

LOAD\_MORE\_PHONES\_START,

LOAD\_MORE\_PHONES\_SUCCESS,

LOAD\_MORE\_PHONES\_FAILURE,

FETCH\_PHONE\_BY\_ID\_START,

FETCH\_PHONE\_BY\_ID\_SUCCESS,

FETCH\_PHONE\_BY\_ID\_FAILURE,

ADD\_PHONE\_TO\_BASKET,

SEARCH\_PHONE,

FETCH\_CATEGORIES\_START,

FETCH\_CATEGORIES\_SUCCESS,

FETCH\_CATEGORIES\_FAILURE,

} from 'actionTypes'

export const fetchCategories = () => async dispatch => {

dispatch({type: FETCH\_CATEGORIES\_START})

try {

const phones = await fetchCategoriesApi()

dispatch({

type: FETCH\_CATEGORIES\_SUCCESS,

payload: phones

})

} catch (err) {

dispatch({

type: FETCH\_CATEGORIES\_FAILURE,

payload: err,

error: true

})

}

}

Теперь импортируем fetchCategoriesApi

import {

fetchPhones as fetchPhonesApi,

loadMorePhones as loadMorePhonesApi,

fetchPhoneById as fetchPhoneByIdApi,

fetchCategories as fetchCategoriesApi,

} from 'api'

Создадим файл mockCategories в api с моковыми данными наших категорий.

export default [

{

id: "1",

name: 'Apple'

},

{

id: "2",

name: 'Samsung'

},

{

id: "3",

name: 'Htc'

},

{

id: "4",

name: 'Lenovo'

},

{

id: "5",

name: 'Microsoft'

}

]

Теперь опишем метод fetchCategories в api и експортируем моковые данные.

import categories from './mockCategories'

export const fetchCategories = async () => {

return new Promise((resolve, reject) => {

resolve(categories)

})

}

Если мы посмотрим в браузер, то у нас выстреливает два екшена FETCH\_CATEGORIES\_START и FETCH\_CATEGORIES\_SUCCESS.

Теперь пора создать редьюсер под категории. Он будет абсолютно такой же как был редьюсер phones.

import R from 'ramda'

import {

FETCH\_CATEGORIES\_SUCCESS

} from 'actionTypes'

const initialState = {}

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

switch (type) {

case FETCH\_CATEGORIES\_SUCCESS:

const newValues = R.indexBy(R.prop('id'), payload)

return R.merge(state, newValues)

default:

return state

}

}

И добавим этот 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'

import categories from './categories'

export default combineReducers({

routing: routerReducer,

phones,

phonesPage,

phonePage,

basket,

categories

})

Если мы посмотрим в браузер, то у нас наши категории записались в редьюсер categories.

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