**Index.js**

**state.js**

**App**

**Browser Router**

**Profile**

**Dialogs**

**News**

**Music**

**Settings**

**MyPosts**

**ProfileInfo**

**Post**

**Post**

**Post**

**Messs**

**UserDialog**

**Очерёдность вызовов. Логика**

**profileReduser.js**

getUser=(userId)=>{return (dispatch)=>{

profileAPI.getUserProfile(userId).then(data => {

                dispatch(setUserProfile(data)); }); }}

setUserProfile=(profile)=>({ type: SET\_USER\_PROFILE, profile });

case SET\_USER\_PROFILE: return{ ...state, profile: action.profile}

initialState = { profile: null }

**api.js**

profileAPI = { getUserProfile(userId).get(`profile/` + userId) }

profileAPI = { getUserStatus(userId).get(`profile/status/` + userId) }

profileAPI = { updateUserStatus(status).put(`profile/`, { statu s: status })} }

**profileContainer.jsx**

componetDidMount() {

let userId = this.props.match.params.userId;

this.props.getUser(userId);}

mapStateToProps = (state) => {

  return { profile: state.profilePage.profile}

compose(connect(…, { getUser }