React Redux

<https://github.com/reduxjs/react-redux>

<https://redux.js.org/basics/usage-with-react>

Redux is a predictable state container for JavaScript apps.

Встановлення

npm install --save react-redux

Using Redux with any UI layer requires [the same consistent set of steps](https://blog.isquaredsoftware.com/presentations/workshops/redux-fundamentals/ui-layer.html" \l "/4):

1. Create a Redux store
2. Subscribe to updates
3. Inside the subscription callback:
   1. Get the current store state
   2. Extract the data needed by this piece of UI
   3. Update the UI with the data
4. If necessary, render the UI with initial state
5. Respond to UI inputs by dispatching Redux actions

При розробці додатку необхідно виконати такі кроки:

1. Створити презентаційні компоненти
2. Створити сховище
3. Підписати компоненти до сховища

|  |  |
| --- | --- |
|  |  |
| Створюємо типи подій |  |
| Створюємо генератори подій |  |

Приєднання до сховища (store)

|  |  |
| --- | --- |
|  |  |
| Підключення необхідних бібліотек  import { Provider } from 'react-redux'  Підлючення сховища  import store from './redux/store'  <Provider store={ сховище }>  <компоненти />  </Provider>, | // index.js  import React from 'react'  import ReactDOM from 'react-dom'  import TodoApp from './TodoApp'  import { Provider } from 'react-redux'  import store from './redux/store'  const rootElement = document.getElementById('root')  ReactDOM.render(  <Provider store={store}>  <TodoApp />  </Provider>,  rootElement  ) |

Зчитування даних зі сховища (store)

|  |  |
| --- | --- |
|  |  |
| //Функція для зчитування даних зі state  const mapStateToProps = (state, ownProps) => ({  . . . повертаємо дані для компонентів . . .  })  //Функція для формування колбеків для компонентів  const mapDispatchToProps = {  … повертаємо об’єкт з колбеками для компонентів …  }  // До компонента додаємо можливість зчитувати дані  // та вносити зміни у сховище через actions  const під’єднаний\_компонент = connect(  mapStateToProps,  mapDispatchToProps  ) (компонент)  //Передача store до компонента  <Provider store={ сховище }>  < під’єднаний\_компонент />  </Provider>, | import React from 'react'  import {Provider, connect} from 'react-redux'  import DormatoryManager from './dormatory-manager'  import store from "../../store"  import {addBoy,deleteBoy,addGirl,deleteGirl} from '../../action-creators'  //---------------------  const mapStateToProps = (state) => {  return {  boysList:state.boysGroup.boys,  girlsList:state.girlsGroup.girls  }  }  //-----------------------  const mapDispatchToProps = dispatch => {  return {  onAddBoy:(boyName)=>dispatch( addBoy(boyName) ),  onDeleteBoy:(boyId)=>dispatch( deleteBoy(boyId)),  onAddGirl:(girlName)=>dispatch( addGirl(girlName) ),  onDeleteGirl:(girlId)=>dispatch( deleteGirl(girlId)),  }  }  //---------------------  let ConnectedDormatory = connect(  mapStateToProps,  mapDispatchToProps  )(DormatoryManager);  export default () => (  <Provider store={store}>  <ConnectedDormatory/>  </Provider>  ) |