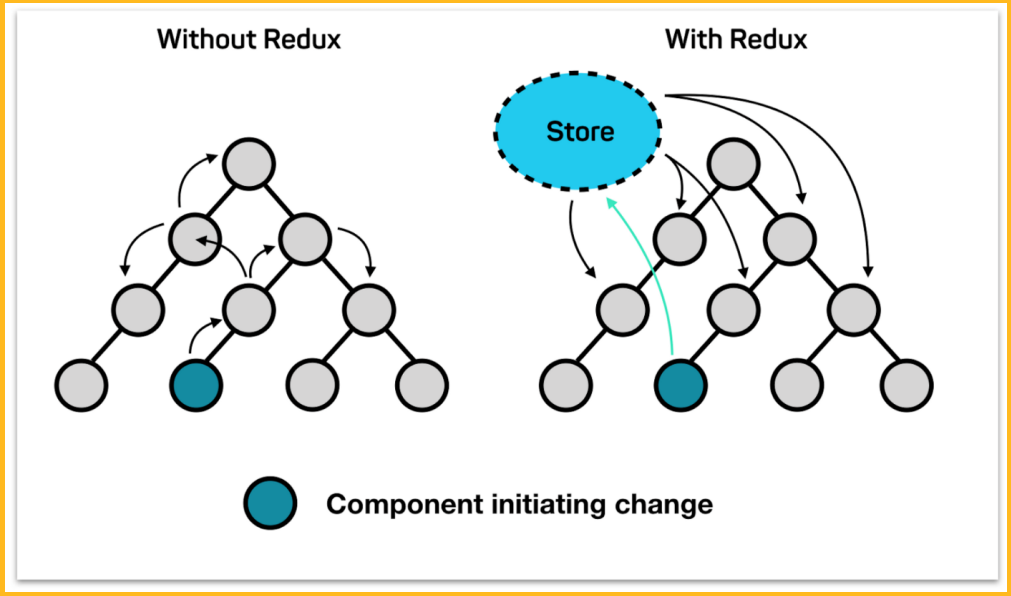
FRONTEND. REACT. ЛЕКЦІЯ 10

REDUX, REDUX TOOLKIT

ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

BEER

ЯКУ ПРОБЛЕМУ ВИРІШУЄ REDUX?



МЕНЕДЖЕРИ СТАНУ

МЕНЕДЖЕРИ СТАНУ

- Redux

МЕНЕДЖЕРИ СТАНУ

- Redux
- MobX

МЕНЕДЖЕРИ СТАНУ

- Redux
- MobX
- Recoil

REDUX

REDUX

- Автор - Ден Абрамов

REDUX

- Автор - Ден Абрамов
- Бібліотека для організації архітектури застосунку

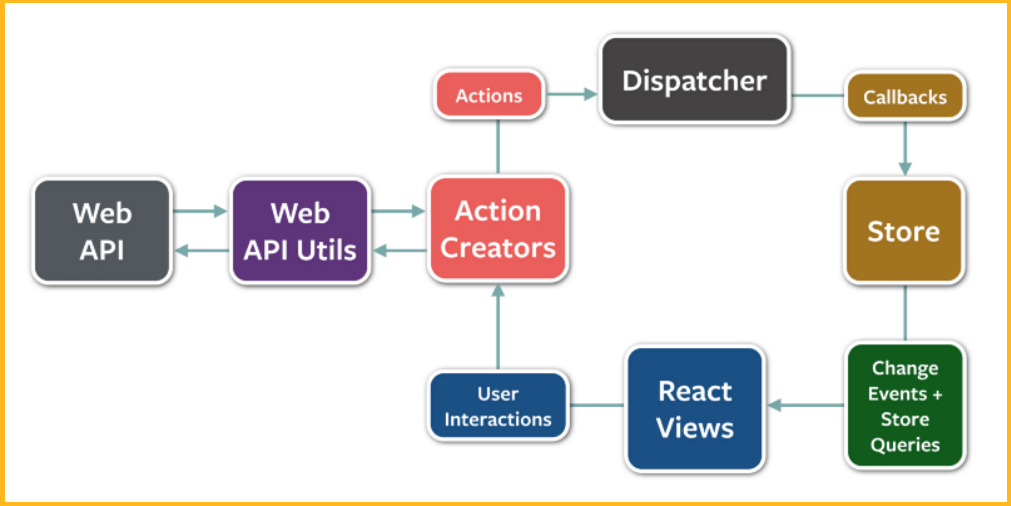
REDUX

- Автор - Ден Абрамов
- Бібліотека для організації архітектури застосунку
- Однонаправлений потік даних

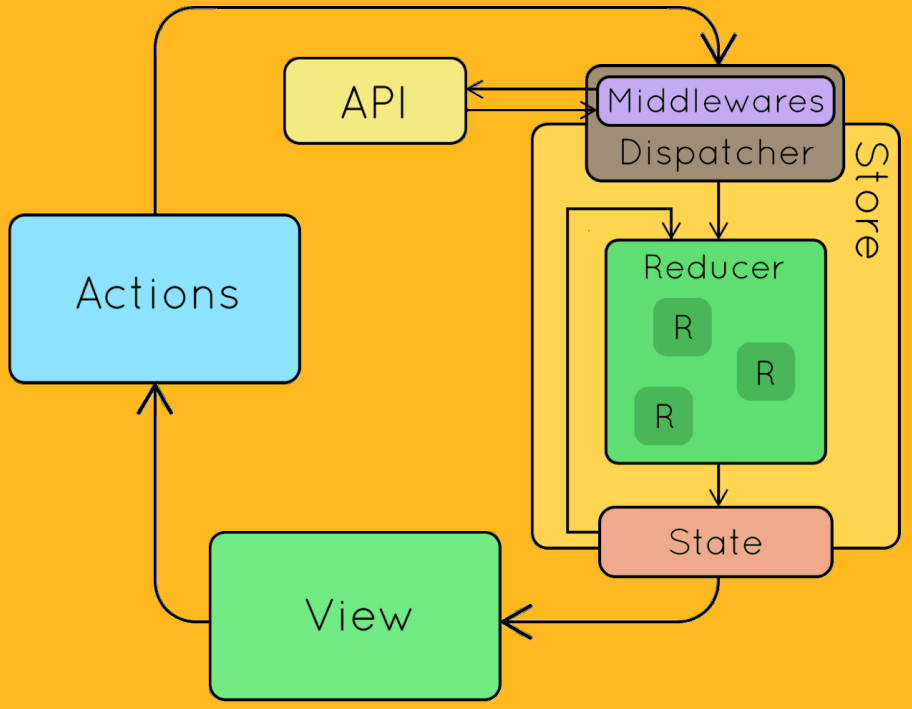
REDUX

- Автор - Ден Абрамов
- Бібліотека для організації архітектури застосунку
- Однонаправлений потік даних
- Іммутабельний стан

FLUX АРХІТЕКТУРА

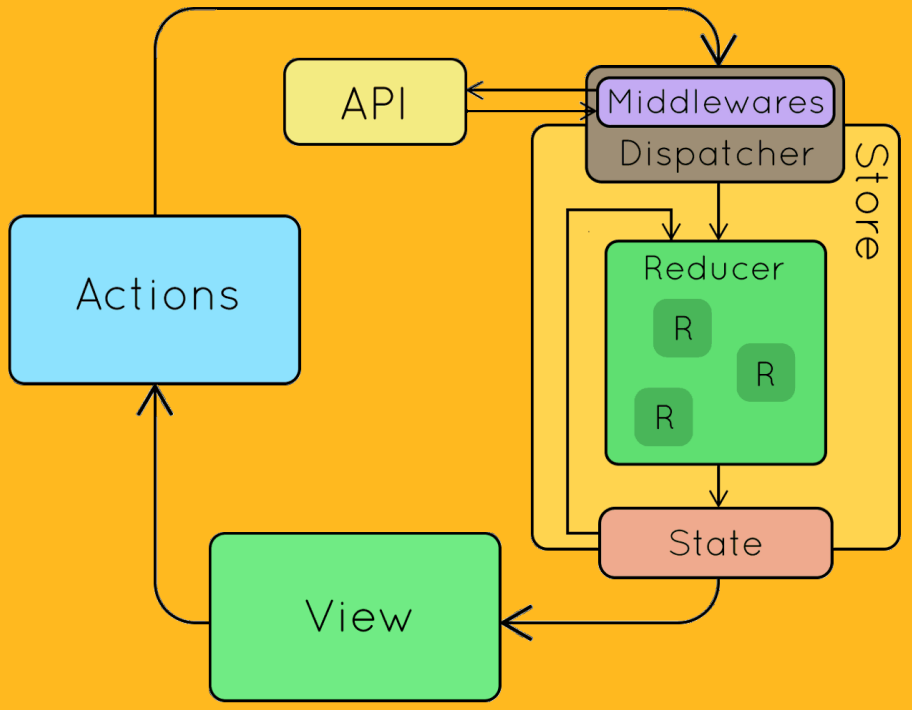


REDUX

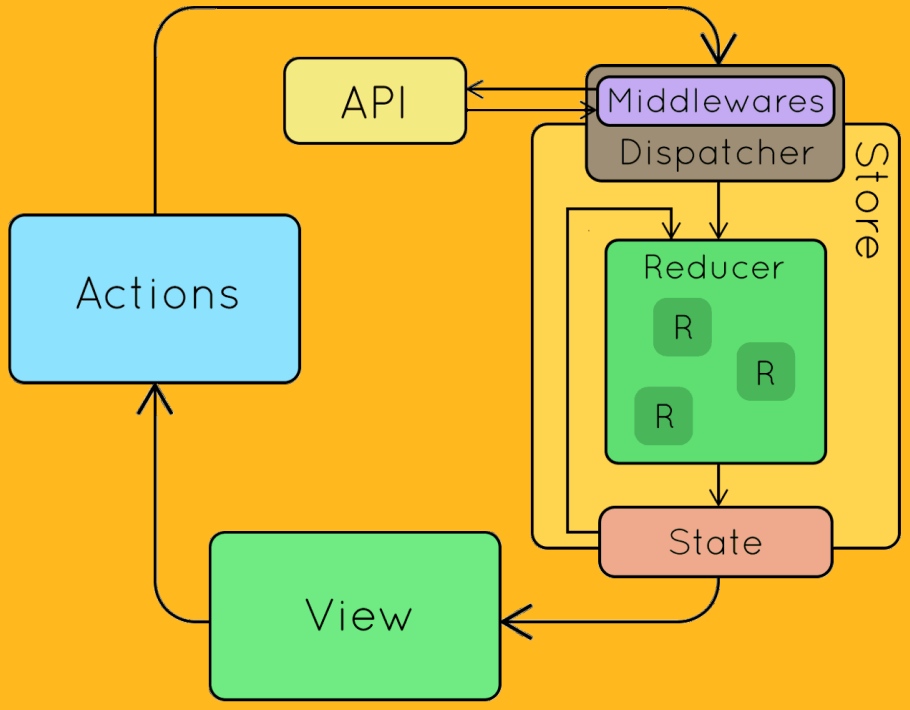

API
Middlewares
Dispatcher
Store
Actions
Reducer
R
R
R
State
View

STATE. СТАН.

Місце, де зберігається весь стан застосунку.

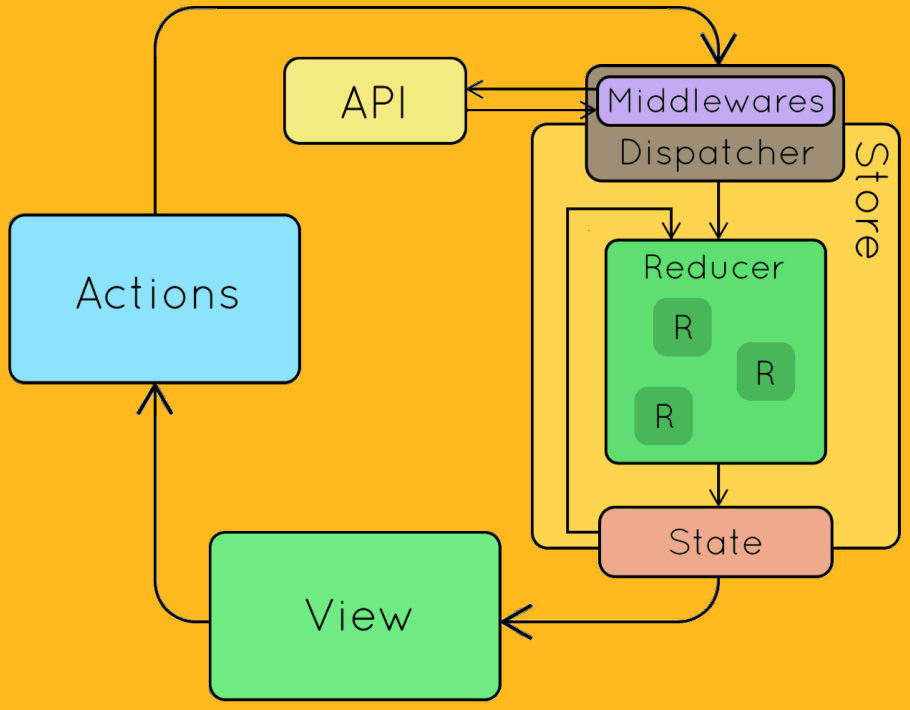


ACTION. ДІЯ.


API
Middlewares
Dispatcher
Store
Reducer
R
R
R
Actions
State
View

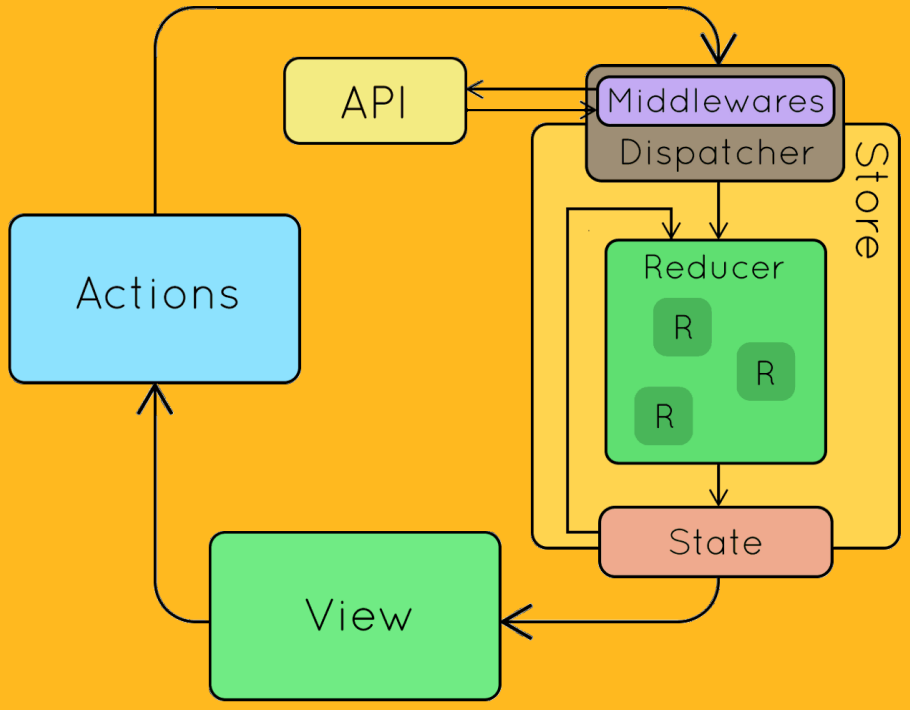
ACTION. ДІЯ.

- Стан неможливо змінювати напряму



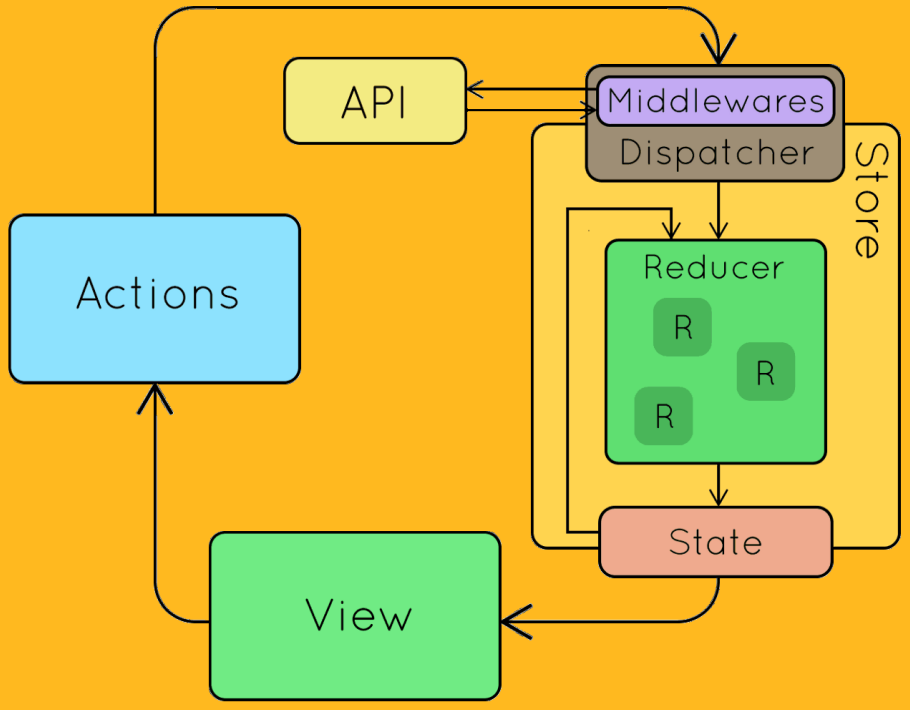
ACTION. ДІЯ.

- Стан неможливо змінювати напряму
- Стан необхідно змінювати через дію

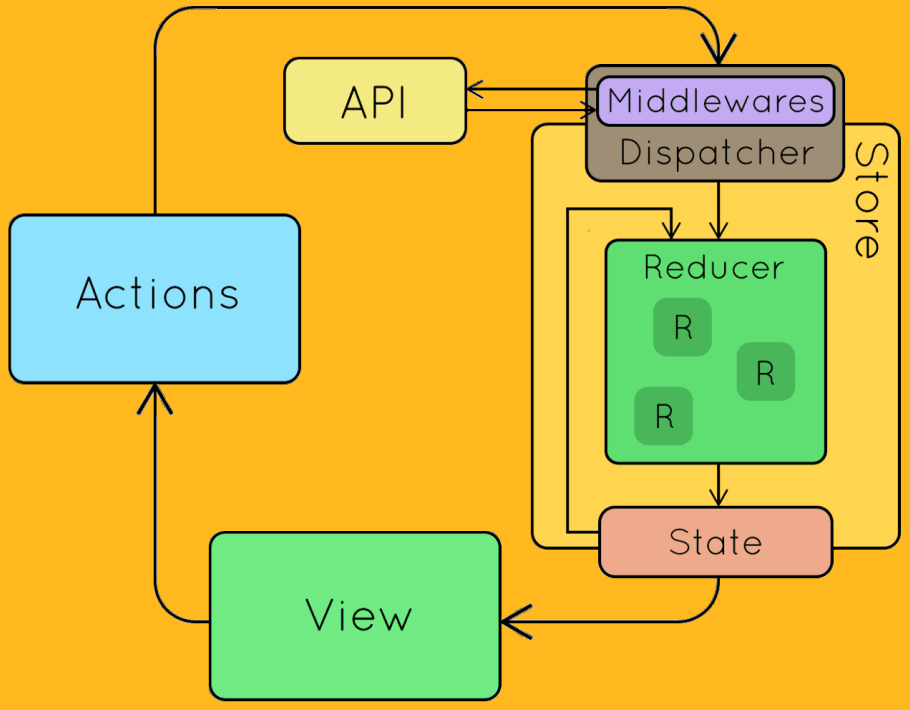


ACTION. ДІЯ.

- Стан неможливо змінювати напряму
- Стан необхідно змінювати через дію
- Те, "як" ми змінюємо дані

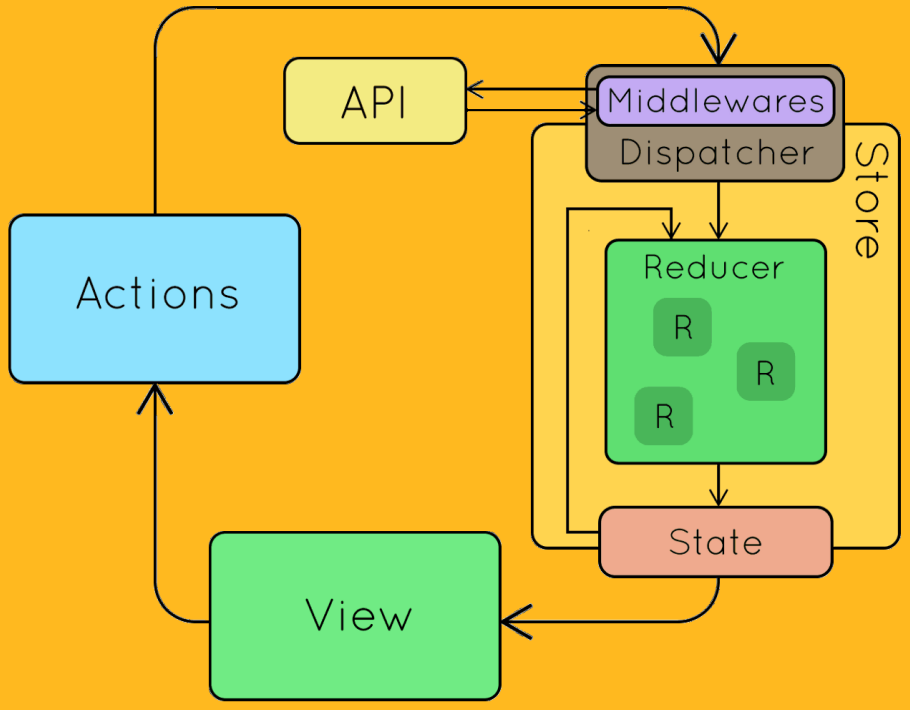


DISPATCH. ДІСПЕТЧЕР.



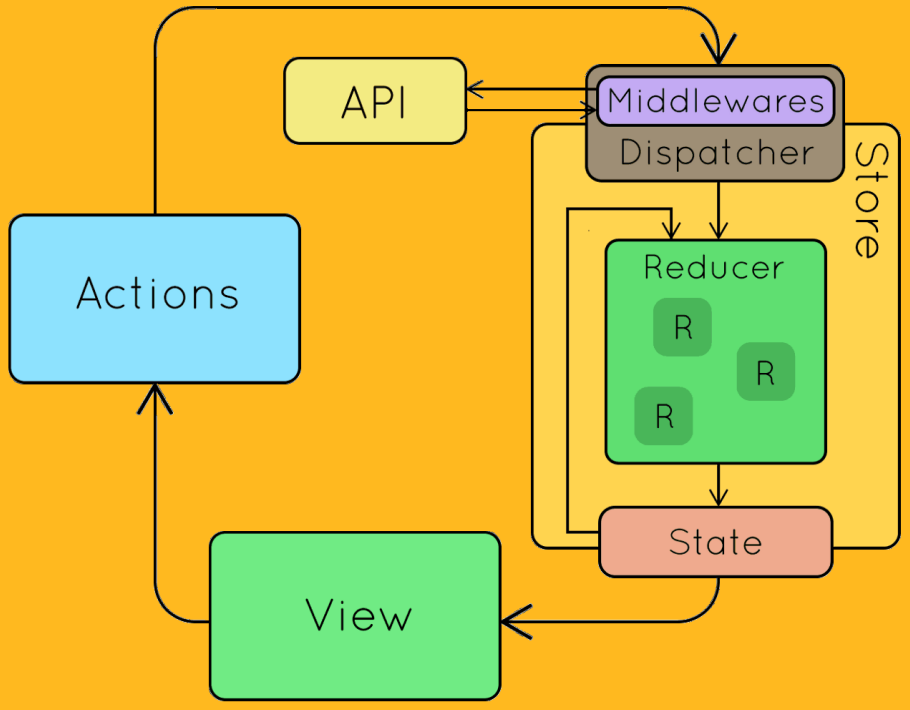
DISPATCH. ДІСПЕТЧЕР.

- Звертаємось до діспетчера



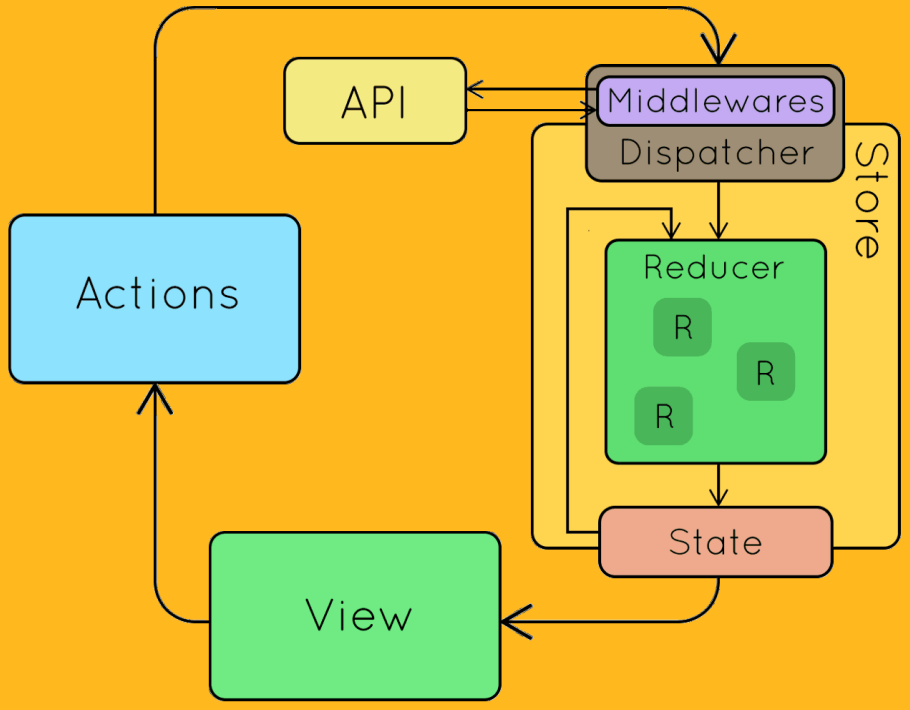
DISPATCH. ДІСПЕТЧЕР.

- Звертаємось до діспетчера
- Передаємо дію

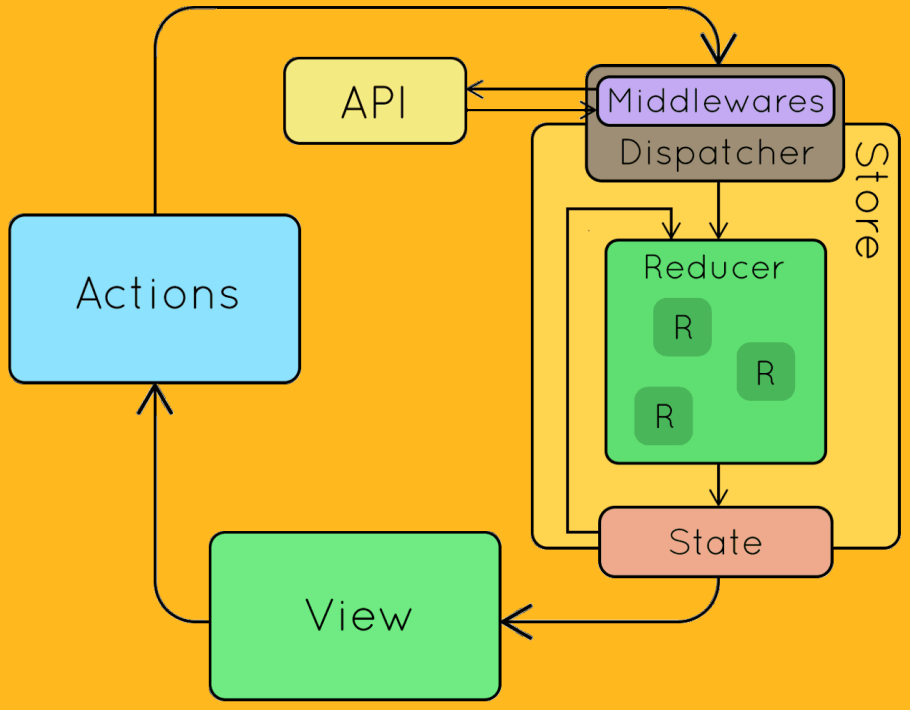


DISPATCH. ДІСПЕТЧЕР.

- Звертаємось до діспетчера
- Передаємо дію
- Той, "хто" ми змінюємо дані

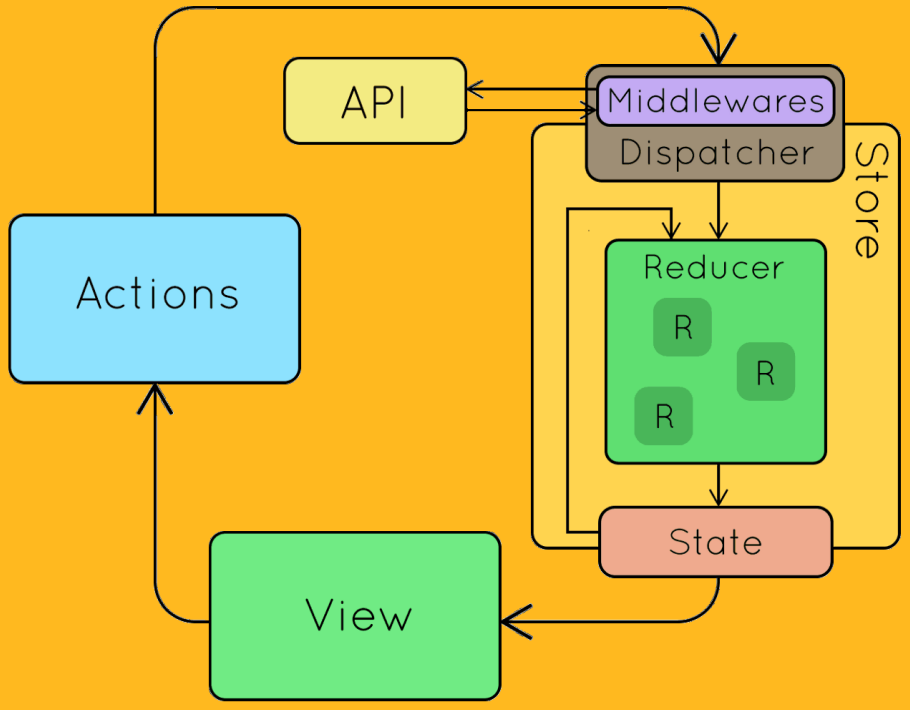


REDUXER. РЕДЬЮСЕР.



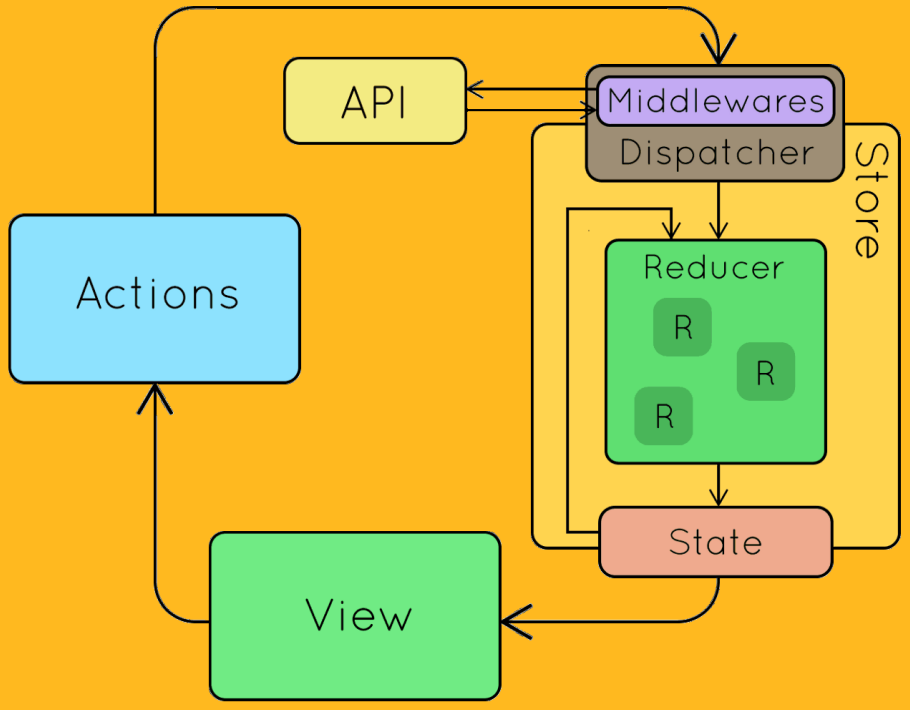
REDUXER. РЕДЬЮСЕР.

- Система, через яку проходить взаємодія



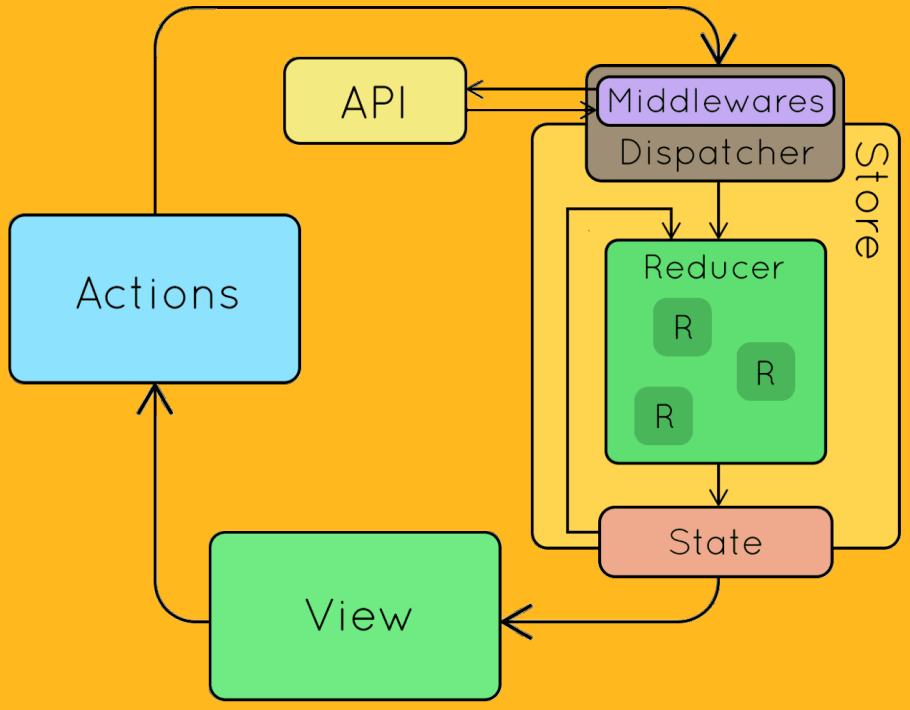
REDUXER. РЕДЬЮСЕР.

- Система, через яку проходить взаємодія
- Знаходиться вся логіка



REDUXER. РЕДЬЮСЕР.

- Система, через яку проходить взаємодія
- Знаходиться вся логіка
- Там, "де" працюють з даними



ВСТАНОВЛЕННЯ

```
npm i redux react-redux
```

STORE

```
1  import React from 'react';
2  import ReactDOM from 'react-dom/client';
3  import './index.css';
4  import App from './App';
5  import reportWebVitals from './reportWebVitals';
6  import { createStore } from 'redux';
7  import { Provider } from 'react-redux';
8
9
10 const root = ReactDOM.createRoot(document.getElementById('root'));
11 const defaultState = {
12         counter: 0
13 }
14 const counterReducer = (state = defaultState, action) => {
15         switch (action.type) {
16                 case "INCREASE_COUNTER":
17                         return {...state, counter: state.counter + Number(acti
18                 case "DECREASE_COUNTER":
```

STORE

```
17                          return {...state, counter: state.counter + Number(acti
18                  case "DECREASE_COUNTER":
19                          return {...state, counter: state.counter - Number(acti
20                  default:
21                          return state;
22                  }
23          }
24  }
25  const store = createStore(reducer);
26
27  root.render(
28  <react.strictmode>
29          <provider store="{store}">
30                  <app>
31          </app></provider>
32  </react.strictmode>
33  );
```

STORE

```
17                          return {...state, counter: state.counter + Number(acti
18                  case "DECREASE_COUNTER":
19                          return {...state, counter: state.counter - Number(acti
20                  default:
21                          return state;
22                  }
23          }
24  }
25  const store = createStore(reducer);
26
27  root.render(
28  <react.strictmode>
29          <provider store="{store}">
30                  <app>
31          </app></provider>
32  </react.strictmode>
33  );
```

STORE

```
10 const root = ReactDOM.createRoot(document.getElementById('root'));
11 const defaultState = {
12         counter: 0
13 }
14 const counterReducer = (state = defaultState, action) => {
15         switch (action.type) {
16                 case "INCREASE_COUNTER":
17                         return {...state, counter: state.counter + Number(acti
18                 case "DECREASE_COUNTER":
19                         return {...state, counter: state.counter - Number(acti
20                 default:
21                         return state;
22                 }
23         }
24 }
25 const store = createStore(reducer);
26
27 root.render(
```

STORE

```
import App from './App';
import reportWebVitals from './reportWebVitals';
import { createStore } from 'redux';
import { Provider } from 'react-redux';


const root = ReactDOM.createRoot(document.getElementById('root'));
const defaultState = {
        counter: 0
}
const counterReducer = (state = defaultState, action) => {
        switch (action.type) {
                case "INCREASE_COUNTER":
                        return {...state, counter: state.counter + Number(acti
                case "DECREASE_COUNTER":
                        return {...state, counter: state.counter - Number(acti
                default:
                        return state;
```

STORE

```
10 const root = ReactDOM.createRoot(document.getElementById('root'));
11 const defaultState = {
12         counter: 0
13 }
14 const counterReducer = (state = defaultState, action) => {
15         switch (action.type) {
16                 case "INCREASE_COUNTER":
17                         return {...state, counter: state.counter + Number(acti
18                 case "DECREASE_COUNTER":
19                         return {...state, counter: state.counter - Number(acti
20                 default:
21                         return state;
22                 }
23         }
24 }
25 const store = createStore(reducer);
26
27 root.render(
```

STORE

```
17                          return {...state, counter: state.counter + Number(acti
18                  case "DECREASE_COUNTER":
19                          return {...state, counter: state.counter - Number(acti
20                  default:
21                          return state;
22                  }
23          }
24  }
25  const store = createStore(reducer);
26
27  root.render(
28  <react.strictmode>
29          <provider store="{store}">
30                  <app>
31          </app></provider>
32  </react.strictmode>
33  );
```

STORE

```
17                          return {...state, counter: state.counter + Number(acti
18                  case "DECREASE_COUNTER":
19                          return {...state, counter: state.counter - Number(acti
20                  default:
21                          return state;
22                  }
23          }
24  }
25  const store = createStore(reducer);
26
27  root.render(
28  <react.strictmode>
29          <provider store="{store}">
30                  <app>
31          </app></provider>
32  </react.strictmode>
33  );
```

STORE

```
14 const counterReducer = (state = defaultState, action) => {
15         switch (action.type) {
16                 case "INCREASE_COUNTER":
17                         return {...state, counter: state.counter + Number(acti
18                 case "DECREASE_COUNTER":
19                         return {...state, counter: state.counter - Number(acti
20                 default:
21                         return state;
22                 }
23         }
24 }
25 const store = createStore(reducer);
26
27 root.render(
28 <react.strictmode>
29         <provider store="{store}">
30                 <app>
31         </app></provider>
```

STORE

```
                case "INCREASE_COUNTER":
                        return {...state, counter: state.counter + Number(acti
                case "DECREASE_COUNTER":
                        return {...state, counter: state.counter - Number(acti
                default:
                        return state;
                }
        }
}
const store = createStore(reducer);

root.render(
<react.strictmode>
        <provider store="{store}">
                <app>
        </app></provider>
</react.strictmode>
);
```

ВИКОРИСТАННЯ. USEDISPATCH. USESELECTOR

```
import { useDispatch, useSelector } from "react-redux";

const Lesson10 = () => {
        const dispatch = useDispatch();
        const counter = useSelector(state => state.counter);
        const increase = () => {
                dispatch({type: "INCREASE_COUNTER", payload: 5})
        }
        const decrease = () => {
                dispatch({type: "DECREASE_COUNTER", payload: 5})
        }
        return (
                <div>
                        Counter: {counter}
                        <button onclick="{increase}">Increase counter</button>
                        <button onclick="{decrease}">Decrease counter</button>
                </div>
        )
```

ВИКОРИСТАННЯ. USEDISPATCH. USESELECTOR

```
import { useDispatch, useSelector } from "react-redux";

const Lesson10 = () => {
        const dispatch = useDispatch();
        const counter = useSelector(state => state.counter);
        const increase = () => {
                dispatch({type: "INCREASE_COUNTER", payload: 5})
        }
        const decrease = () => {
                dispatch({type: "DECREASE_COUNTER", payload: 5})
        }
        return (
                <div>
                        Counter: {counter}
                        <button onclick="{increase}">Increase counter</button>
                        <button onclick="{decrease}">Decrease counter</button>
                </div>
        )
```

ВИКОРИСТАННЯ. USEDISPATCH. USESELECTOR

```
1  import { useDispatch, useSelector } from "react-redux";
2
3  const Lesson10 = () => {
4          const dispatch = useDispatch();
5          const counter = useSelector(state => state.counter);
6          const increase = () => {
7                  dispatch({type: "INCREASE_COUNTER", payload: 5})
8          }
9          const decrease = () => {
10                 dispatch({type: "DECREASE_COUNTER", payload: 5})
11         }
12         return (
13                 <div>
14                         Counter: {counter}
15                         <button onclick="{increase}">Increase counter</button>
16                         <button onclick="{decrease}">Decrease counter</button>
17                 </div>
18         )
```

ВИКОРИСТАННЯ. USEDISPATCH. USESELECTOR

```
3  const Lesson10 = () => {
4          const dispatch = useDispatch();
5          const counter = useSelector(state => state.counter);
6          const increase = () => {
7                  dispatch({type: "INCREASE_COUNTER", payload: 5})
8          }
9          const decrease = () => {
10                 dispatch({type: "DECREASE_COUNTER", payload: 5})
11         }
12         return (
13                 <div>
14                         Counter: {counter}
15                         <button onclick="{increase}">Increase counter</button>
16                         <button onclick="{decrease}">Decrease counter</button>
17                 </div>
18         )
19 }
20 export { Lesson10 }
```

ВИКОРИСТАННЯ. USEDISPATCH. USESELECTOR

```
import { useDispatch, useSelector } from "react-redux";

const Lesson10 = () => {
        const dispatch = useDispatch();
        const counter = useSelector(state => state.counter);
        const increase = () => {
                dispatch({type: "INCREASE_COUNTER", payload: 5})
        }
        const decrease = () => {
                dispatch({type: "DECREASE_COUNTER", payload: 5})
        }
        return (
                <div>
                        Counter: {counter}
                        <button onclick="{increase}">Increase counter</button>
                        <button onclick="{decrease}">Decrease counter</button>
                </div>
        )
```

REDUX HOOKS

https://react-redux.js.org/api/hooks

On this page

Hooks

React's new "hooks" APIs give function components the ability to use local component state, execute side effects, and more. React also lets us write custom hooks, which let us extract reusable hooks to add our own behavior on top of React's built-in hooks.

React Redux includes its own custom hook APIs, which allow your React components to subscribe to the Redux store and dispatch actions.

> **TIP**
>
> **We recommend using the React-Redux hooks API as the default approach in your React components.**
>
> The existing `connect` API still works and will continue to be supported, but the hooks API is simpler and works better with TypeScript.

These hooks were first added in v7.1.0.

СТРУКТУРА

СТРУКТУРА

- Тека store

СТРУКТУРА

- Тека store
- Всередині index.js

СТРУКТУРА

- Тека store
- Всередині index.js
- Тека reducers

СТРУКТУРА

- Тека store
- Всередині index.js
- Тека reducers
- Тека actions

COMBINEREDUCER

Поєднання декількох редьюсерів в одне ціле

```
import { combineReducers, createStore } from "redux";
import { counterReducer } from "./reducers/counterReducer";
import { usersReducer } from "./reducers/usersReducer";

const rootReducer = combineReducers({
        counter: counterReducer,
        users: usersReducer
})

export const store = createStore(rootReducer);
```

DEVTOOLS

https://chrome.google.com/webstore/detail/redux-devtools/lmhkpmbekcpmknklioeibfkpmmfibljd?hl=ru

DEVTOOLS

```
npm i @redux-devtools/extension
```

```
1  import { combineReducers, createStore } from "redux";
2  import { counterReducer } from "./reducers/counterReducer";
3  import { usersReducer } from "./reducers/usersReducer";
4  import { composeWithDevTools } from "redux-devtools-extension";
5
6  const rootReducer = combineReducers({
7          counter: counterReducer,
8          users: usersReducer
9  })
10
11 export const store = createStore(rootReducer, composeWithDevTools());
```

DEVTOOLS

```
npm i @redux-devtools/extension
```

```
import { combineReducers, createStore } from "redux";
import { counterReducer } from "./reducers/counterReducer";
import { usersReducer } from "./reducers/usersReducer";
import { composeWithDevTools } from "redux-devtools-extension";

const rootReducer = combineReducers({
        counter: counterReducer,
        users: usersReducer
})

export const store = createStore(rootReducer, composeWithDevTools());
```

ACTION CREATORS

```
export const addUserAction = (payload) => ({type: ADD_USER, payload});
export const removeUserAction = (payload) => ({type: REMOVE_USER, payload});
```

```
const removeUser = (id) => {
        dispatch(removeUserAction(id));
}
```

REDUX THUNK

```
npm i redux-thunk
```

ПІДКЛЮЧЕННЯ

```
export const store = createStore(rootReducer, composeWithDevTools(applyMiddleware(
```

АСИНХРОННА ЛОГІКА

```
import { addUsersAction } from "../reducers/usersReducer"

export const fetchUsers = () => {
        return (dispatch) => {
                fetch('https://jsonplaceholder.typicode.com/users')
                        .then(response => response.json())
                        .then(json => dispatch(addUsersAction(json)))
        }
}
```

REDUX SAGA

REDUX SAGA

- Комплексне рішення

REDUX SAGA

- Комплексне рішення
- Побудоване на генераторах

REDUX TOOLKIT

```
npm i @reduxjs/toolkit
```

REDUX TOOLKIT

```
npm i @reduxjs/toolkit
```

- Надбудова над редакс

REDUX TOOLKIT

```
npm i @reduxjs/toolkit
```

- Надбудова над редакс
- Автори - розробники редакс

REDUX TOOLKIT

```
npm i @reduxjs/toolkit
```

- Надбудова над редакс
- Автори - розробники редакс
- Вбудований відразу DevTools, redux

REDUX TOOLKIT

```
npm i @reduxjs/toolkit
```

- Надбудова над редакс
- Автори - розробники редакс
- Вбудований відразу DevTools, redux
- Емулює мутації

CREATEREDUCER, CREATEACTION

```
import { createAction, createReducer } from "@reduxjs/toolkit"

const defaultState = {
        counter: 0
}

const increase = createAction("INCREASE_COUNTER")
const decrease = createAction("DECREASE_COUNTER")

export default createReducer(defaultState, {
        [increase]: (state, action) => state.counter = state.counter + action.payl
        [decrease]: (state, action) => state.counter = state.counter - action.payl
})
```

CREATESLICE

```
import { createSlice } from '@reduxjs/toolkit';

const toolkitSlice = createSlice({
        name: "toolkit",
        initialState: {
                counter: 0
        },
        reducers: {
                increment(state) {
                        console.log('test')
                        state.counter = state.counter + 1
                },
                decrement(state) {
                        state.counter = state.counter - 1
                }
        }

});
```

CREATEASYNCTHUNK

ɯo? ɯo?