

43. Redux

Цель:

Познакомиться со следующими понятиями redux:

- action type, action creator
- reducer
- store

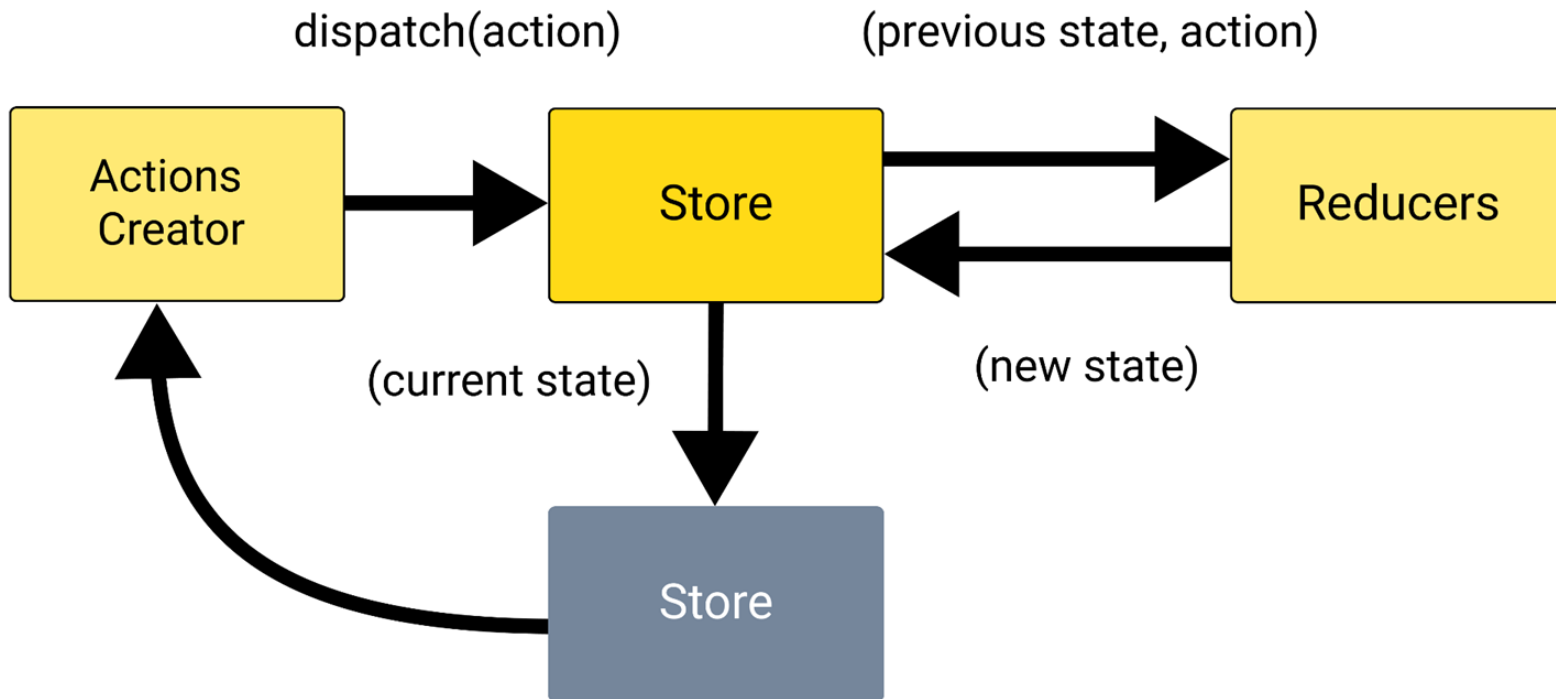
Примеры данных в redux:

- ответы сервера
- данные, созданные локально, но еще не сохраненные на сервере
- UI-состояния
- выделенный таб
- показанное модальное окно
- нумерация страниц и т.д

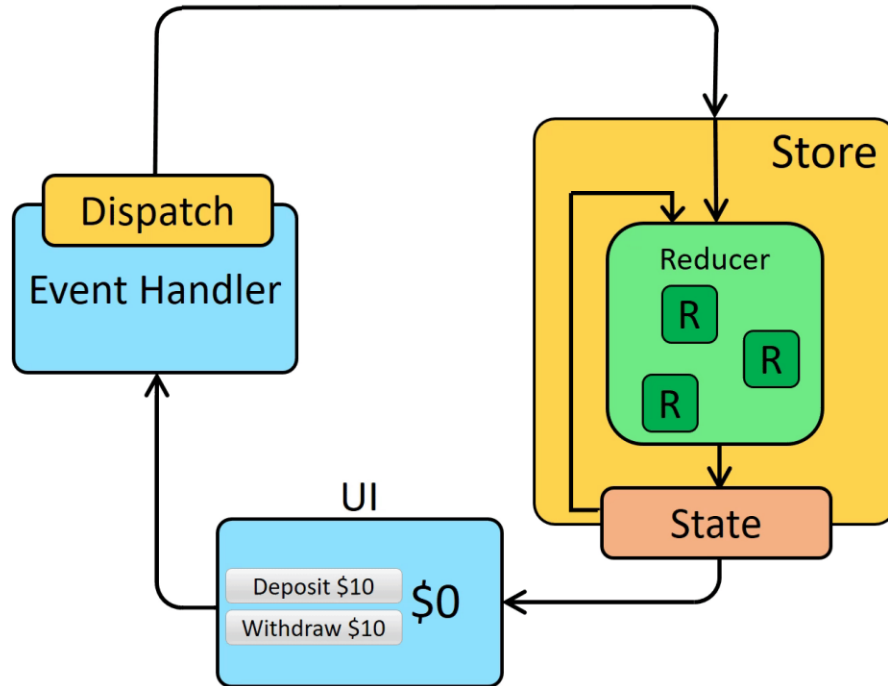
Три принципа redux:

- 1) **Единственный источник правды.** Состояние всего вашего приложения сохранено в дереве объектов внутри одного **стора**.
- 2) **Состояние только для чтения.** Единственный способ изменить состояние — это применить экшен — объект, который описывает, что случится.
- 3) **Мутации написаны, как чистые функции.** Для определения того, как дерево состояния будет трансформировано экшенами, вы пишете чистые редюсеры.

Redux Data Flow:



Redux Data Flow:



Action:

Экшены — это структуры, которые передают данные из вашего приложения в стор.

Они являются *единственными* источниками информации для стора.

Вы отправляете их в стор, используя метод `store.dispatch()` или хук `useDispatch`.

```
const ADD_USERNAME = 'ADD_USERNAME'

{
  type: ADD_USERNAME,
  text: 'Jon Snow'
}
```

Dispatch action:

Xyk useDispatch:

```
const dispatch = useDispatch();
```

```
dispatch(addUsername(username));  
dispatch(addProduct(product));
```


Reducers:

Редюсер (reducer) — это чистая функция, которая принимает предыдущее состояние и экшен (state и action) и возвращает следующее состояние (новую версию предыдущего).

(previousState, action) => newState

Reducers:

Список того, что **никогда** нельзя делать в редюсере:

- Непосредственно изменять то, что пришло в аргументах функции;
- Выполнять какие-либо сайд-эффекты: обращаться к API или осуществлять переход по роутам;
- Вызывать не чистые функции, например `Date.now()` или `Math.random()`.

Store:|

Стор - это не класс.

Это просто **объект** с некоторыми методами в нем.

Стор содержит все дерево состояний вашего приложения.

Единственный способ поменять состояние внутри него - это направить(dispatch) action на него.

Xyk useSelector:

```
import React from 'react';
import { useSelector } from 'react-redux';
import { createSelector } from 'reselect';

const selectNumActiveUsers = createSelector(
  (state) => state.users,
  (users) => users.filter((user) => user.active).length
);

export const UsersList = () => {
  const numActiveUsers = useSelector(selectNumActiveUsers);
  return <div>{numActiveUsers}</div>;
};
```

Хук useStore:

useStore - получение store целиком.

```
const { dispatch, getState, subscribe } = useStore();
```