

# [ Lesson 4 ]

## [Что мы рассмотрим:]

- Что такое Redux и какие задачи он решает
- Основные принципы Redux
- Установка и использование Redux в ваших проектах

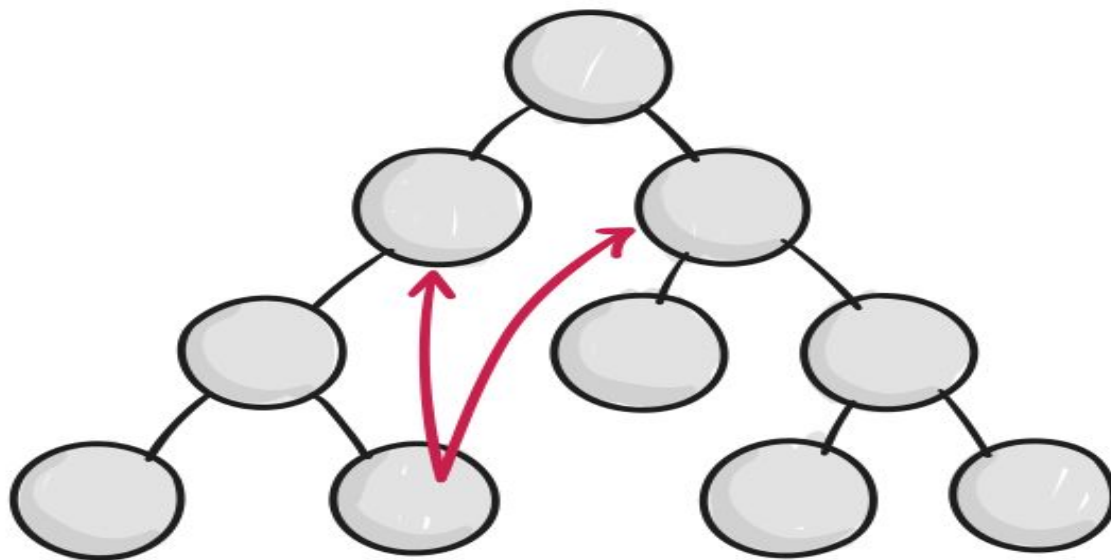
## [Что такое Redux]



# Redux

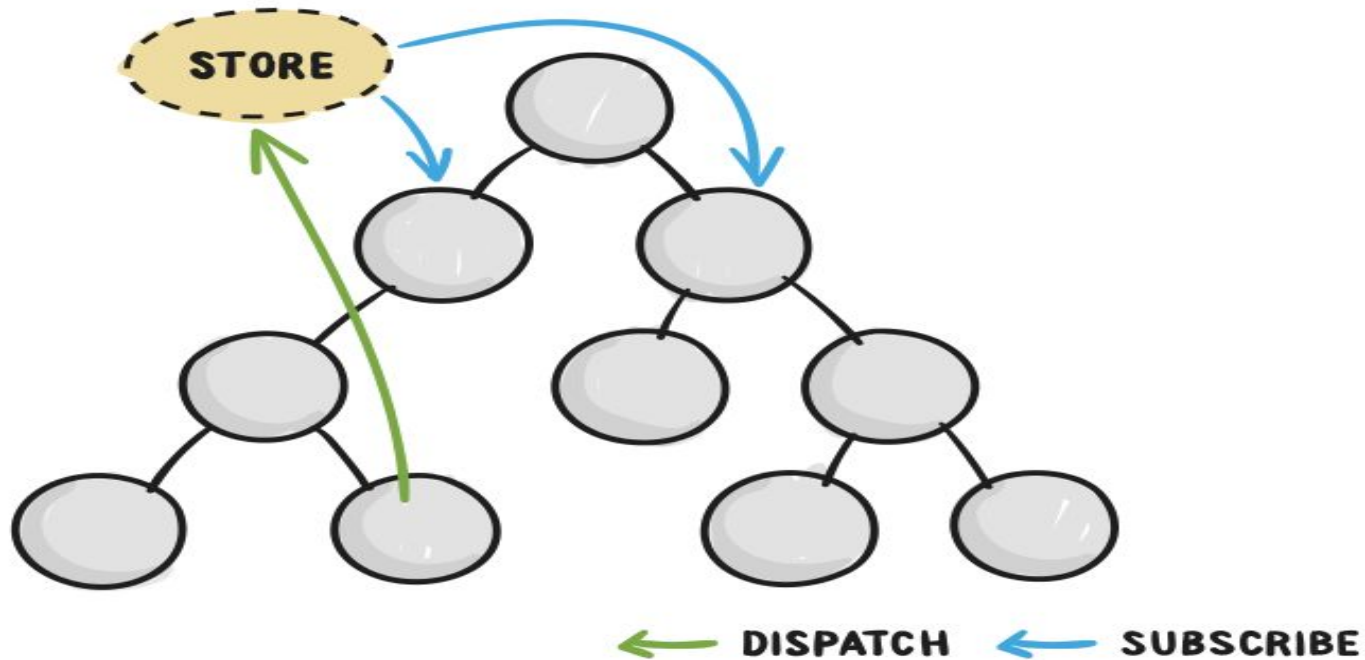
**Redux** — это инструмент управления как состоянием данных, так и состоянием интерфейса в JavaScript-приложениях.

# [Redux]



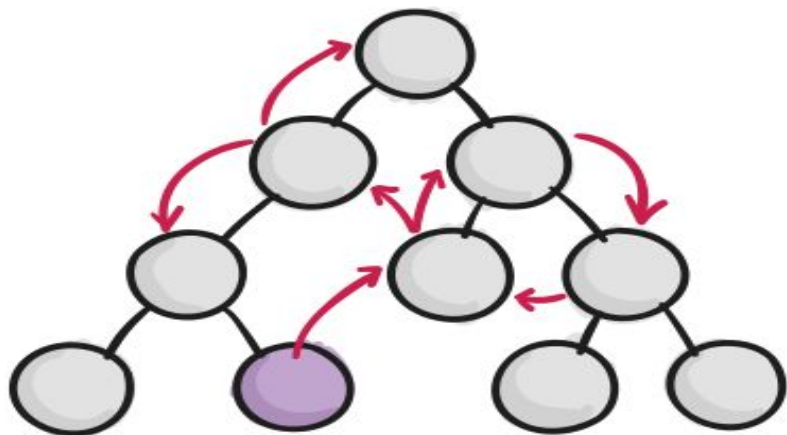
**POOR PRACTICE WHEN COMPONENTS TRY TO  
COMMUNICATE DIRECTLY**

# [Redux]

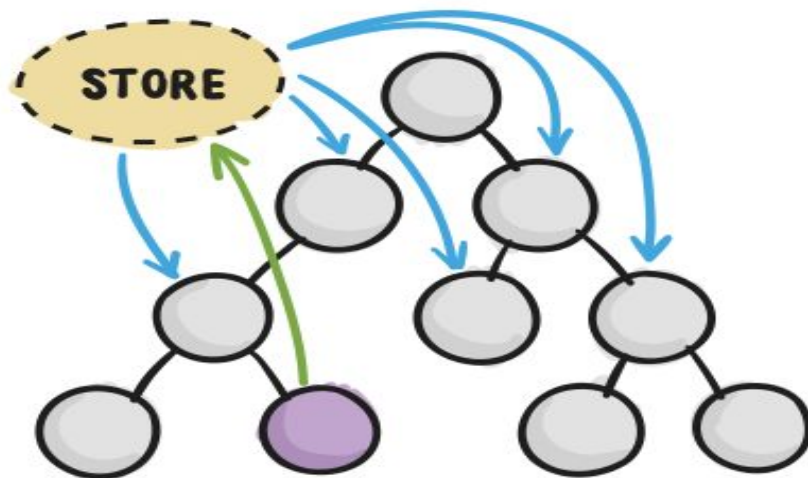


# [Redux]

## WITHOUT REDUX



## WITH REDUX



 COMPONENT INITIATING CHANGE

# [Redux]

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

## **./src/index.js**

```
import { Provider } from 'react-redux'  
import store from "../store";  
  
...  
  
ReactDOM.render(  
  <Provider store={store}>  
    <BrowserRouter><App /></BrowserRouter>  
  </Provider>  
  , document.getElementById('root'));
```

# [Redux]

## ./src/store.js

```
import {createStore} from 'redux'  
import rootReducer from './reducers/index'  
  
const store = createStore(rootReducer)  
  
export default store;
```



# [Redux]

## ./src/reducers/index.js

```
import {combineReducers} from 'redux'  
  
const rootReducer = combineReducers ({  
  
})  
  
export default rootReducer;
```

# [Redux]

## ./src/reducers/mails/index.js

```
const initialState = {
  mailList: { getted: [], sent: [] },
}

function mails (state = initialState, action) {
  switch (action.type){
    case 'SENT_EMAIL': //что делаем при отправке письма
      let updatedSentMailList = [...this.state.mailList.sent];
      updatedSentMailList.push(action.payload)
      return {...state, mailList: {...this.state.mailList, sent: [...updatedSentMailList]}}
    default:
      return {...state}
  }
}

export default mails
```

# [Redux]

## ./src/components/MailList/index.js

```
import {connect} from 'react-redux';  
...  
const mapStateToProps = (state) => {  
  return {  
    // из стора мы берем переменную mailList  
    // и импортируем в наш компонент под именем  
    // mailList в пропсы  
    mails: state.mails.mailList,  
  }  
}  
  
export default connect(mapStateToProps)(MailList)
```

# [Redux]

## ./src/actions/mails.js

```
export const DEL_EMAIL= 'DEL_EMAIL'
```

## ./src/reducers/mails/index.js

```
import { DEL_EMAIL } from '../actions/mails'
...
function mails (state = initialState, action) {
  switch (action.type){
    ...
    case DEL_EMAIL:
      return {...state, mailList: action.payload}
    ...
  }
}
```

# [Redux]

## ./src/components/MailList/index.js

```
const mapDispatchToProps = (dispatch) => {
  return {
    delEmail: (delID, mailList) => {
      let newMailList = {...mailList }
      for(let key in newMailList){
        newMailList[key] = newMailList[key].filter( (item) => {
          return item.id !== delID ? item : null
        })
      }
      dispatch({ type: DEL_EMAIL, payload: {mailList: newMailList} })
    }
  }
}

export default connect(mapStateToProps, mapDispatchToProps)(MailList)
```