

# Как JavaScript может не испортить ваши тесты?

Кто?

@vvscod



О чем?

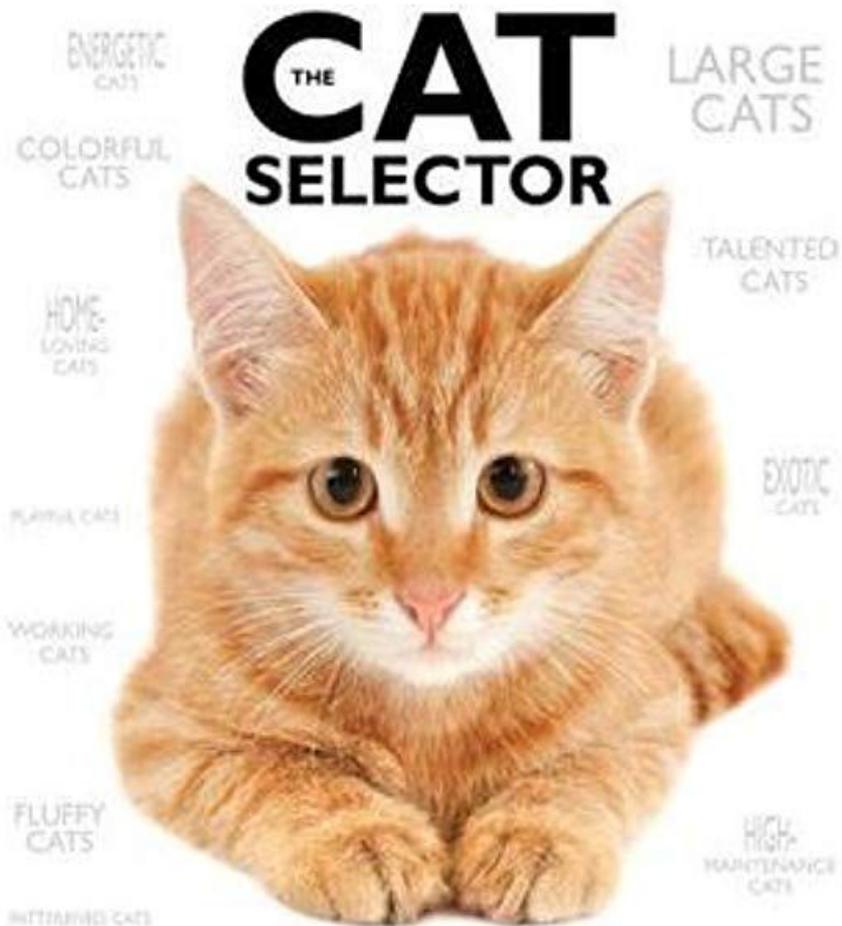


## driver.exec\_script

```
driver.execute_script(  
    "document.getElementsByName('username')[0].click();" )
```

```
result = driver.execute_async_script("""  
    var done = arguments[0];  
    setTimeout(() => done(2), 1000);  
    """)
```

# Селекторы



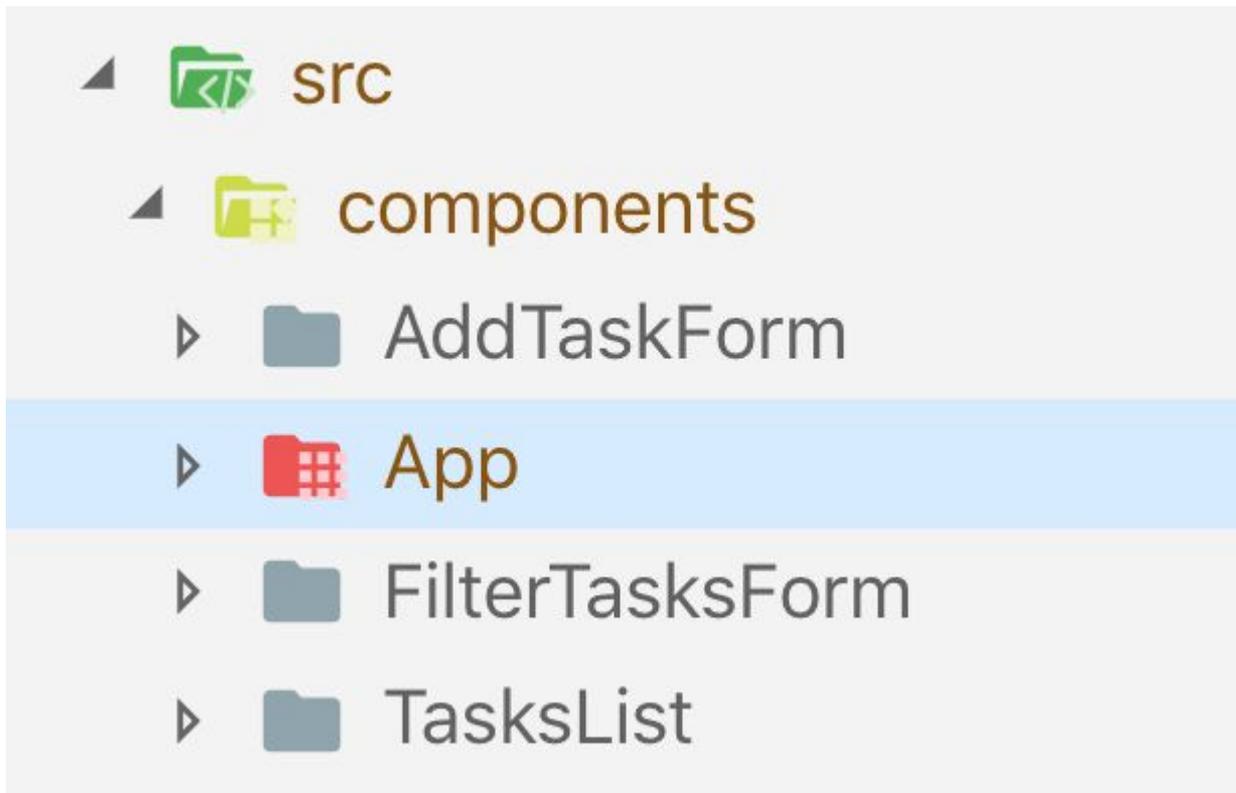
**HOW TO CHOOSE THE RIGHT CAT FOR YOU**

David Alderton

## Селекторы - частый случай

```
<form>
  <div class="wrapper">
    <label>Name</label>
    <input class="input" placeholder="Enter name" />
  </div>
  <div class="wrapper">
    <label>Phone</label>
    <input type="text" class="input" placeholder="Phone" />
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

## Селекторы - компонентный подход



## Компоненты - что внутри

```
// ...
import './styles.css';
// ...

render() {
  return <form>
    <div className="wrapper">
      <label>Name</label>
      <input className="input" placeholder="Enter name" />
    </div>
    <div className="wrapper">
      <label>Phone</label>
      <input type="text" className="input" placeholder="Phone" />
    </div>
    <button type="submit" className="btn btn-primary">Submit</button>
  </form>
}
```

## Компоненты - желаемый результат

```
<form>
  <div class="wrapper">
    <label>Name</label>
    <input class="input qa--user-name-input" placeholder="Enter name" />
  </div>
  <div class="wrapper">
    <label>Phone</label>
    <input type="text" class="input qa--user-phone-input" placeholder="Phone" />
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

## Селекторы - BEM

```
<form>
  <div class="UserDetailForm__wrapper">
    <label>Name</label>
    <input class="UserDetailsForm__input" placeholder="Enter name" />
  </div>
  <div class="UserDetailForm__wrapper">
    <label>Phone</label>
    <input type="text" class="UserDetailsForm__input"
      placeholder="Phone" />
  </div>
  <button
    type="submit"
    class="UserDetailForm__submit-btn"
  >Submit</button>
</form>
```

# Селекторы - CSS Modules

```
<form>
  <div class="UserDetailForm__wrapper__2x0XA">
    <label>Name</label>
    <input
      class="UserDetailForm__input__2xxteZt"
      placeholder="Enter name" />
  </div>
  <div class="UserDetailForm__wrapper__2x0XA">
    <label>Phone</label>
    <input type="text" class="UserDetailForm__input__2xxteZt"
      placeholder="Phone" />
  </div>
  <button
    type="submit"
    class="UserDetailForm__button__3TplV"
  >Submit</button>
</form>
```

## Селекторы - CSS Modules внутри

```
/* import './styles.css'; */  
| import styles from './UserDetailsForm.module.css';  
  
// ...  
  
/* <div className="wrapper"> */  
| <div className={styles.wrapper}>
```

## Селекторы - Component Data Attribute

```
<form data-component="UserDetailsForm">
  <div class="wrapper">
    <label>Name</label>
    <input class="input" placeholder="Enter name" />
  </div>
  <div class="wrapper">
    <label>Phone</label>
    <input type="text" class="input" placeholder="Phone" />
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

## Селекторы - Component Data Attribute - установка

```
// package.json
"devDependencies": {
  "@vvscod/babel-plugin-react-component-data-attribute": "0.5.0"
},
"babel": {
  "plugins": [
    "@vvscod/react-component-data-attribute"
  ]
}
```

# Селекторы - Component Data Attribute - установка

## Installation

---

```
# yarn
yarn add --dev babel-plugin-react-component-data-attribute

# npm
npm install --save-dev babel-plugin-react-component-data-attribute
```

## Usage

---

Via `.babelrc` (Recommended)

`.babelrc`

```
{
  "plugins": ["react-component-data-attribute"]
}
```

## Селекторы - типичный список

```
<ul>  
  <li>Monday</li>  
  <li>Tuesday</li>  
  <li>Wednesday</li>  
  <li>Thursday</li>  
  <li>Friday</li>  
  <li>Saturday</li>  
  <li>Sunday</li>  
</ul>
```

## Селекторы - типичный список внутри

```
render() {  
  return (  
    <ul>  
      {weekDays.map((el) => <li key={el.id}>  
        {el.name}  
      </li>)}  
    </ul>  
  )  
}
```

## Селекторы - ключ к оптимизациям

⊗ ▶ Warning: Each child in a list preview-24934f2...js:1 should have a unique "key" prop.

Check the render method of `WeekDaysList`. See <https://fb.me/react-warning-keys> for more information.  
in li (created by WeekDaysList)  
in WeekDaysList

## Селекторы - помощь с ключами

```
const key = el => ({
  key: el.id,
  'data-key': el.id,
});

// ...

/* weekdays.map((el) => <li key={el.id}> */
weekdays.map((el) => <li {...key(el)}>
```

## Селекторы - data-key

```
<ul>  
  <li data-key="1">Monday</li>  
  <li data-key="2">Tuesday</li>  
  <li data-key="3">Wednesday</li>  
  <li data-key="4">Thursday</li>  
  <li data-key="5">Friday</li>  
  <li data-key="6">Saturday</li>  
  <li data-key="7">Sunday</li>  
</ul>
```



Есть задача? Реши ее

На моей машине все работает

ㄟ(っ)ㄟ

**IT WORKS**

*on my machine*

## Отображение версии - API

```
{  
  "status": "OK",  
  "sdate": "11:09:08 PM 5/7/2019",  
  "gshort": "031d39b"  
}
```

## Отображение версии - meta-tag

```
<meta name="gshort" content="031d39b">  
<meta name="ddate" content="2019-04-04T15:16:07.00Z">
```

## Отображение версии - результаты

- Знаем с какой версией работаем
- Легко предоставить фикс

## Отображение версии - как путь к assets

```
<link rel="stylesheet" href="/031d39b/assets/vendor.css ">  
<link rel="stylesheet" href="/031d39b/assets/project.css">  
<script src="/031d39b/assets/vendor.js"></script>  
<script src="/031d39b/assets/project.js "></script>
```

## Изменение версии

```
var buildMatch = window.location.href.match(/build(\w+)/) ||
  document.cookie.match(/build=([a-z0-9]+)/);
var shouldReplace = !!buildMatch;

if (shouldReplace) {
  var replaceRegexp = /(\w+)\//assets/;
  var assetsPath = buildMatch.pop() + "/assets";

  document.querySelectorAll('script[src]').forEach(function (el) {
    el.src = el.src.replace(replaceRegexp, assetsPath);
  });
}
```

## Изменение версии - def-атрибуты

```
<link def-rel="stylesheet" def-href="/031d39b/assets/project.css">  
<script def-src="/031d39b/assets/project.js" type="def/script"></script>
```

## Изменение версии - передача версии

- URL-параметр
- Cookie / Storage
- Смешать

## Изменение версии

- не нужны дополнительные окружения для FE
- разработчик легко может показать результат своей работы

## Настройки приложения

- Подмена адресов API
- Управление анимациями
- Конфигурация сторонних сервисов

## Настройки приложения - конфиг

```
export const API_DELAY = 500;

export const STORAGE_PREFIX = 'heisenbug';

export const PRIORITIES = {
  1: 'high',
  2: 'medium',
  3: 'low',
};
```

## Настройки приложения - конфиг

```
import * as config from './config';

const getKey = () => {
  const key = `${config.STORAGE_PREFIX}__tasks`;
  return key;
}
```

## Настройки приложения - конфиг

```
f = n(10),  
h = { 1: "high", 2: "medium", 3: "low" },  
b = [],  
E = function () {  
  return "".concat("heisenbug", "__tasks");  
},  
O = n(13),
```

## Настройки приложения - конфиг объект

```
const API_DELAY = 500;

const STORAGE_PREFIX = 'heisenbug';

const PRIORITIES = {
  1: 'high',
  2: 'medium',
  3: 'low',
};

const config = {
  API_DELAY,
  STORAGE_PREFIX,
  PRIORITIES,
}

export default config;
```

## Настройки приложения - конфиг объект

```
// import * as config from './config';  
import config from './config';  
  
const getKey = () => {  
  const key = `${config.STORAGE_PREFIX}__tasks`;  
  return key;  
}
```

## Настройки приложения - конфиг объект

```
E = [],  
b = function () {  
  var e = "".concat(h.STORAGE_PREFIX, "__tasks");  
  return e;  
},  
O' = n(13),
```

## Настройки приложения - доступ для тестов

```
const config = {
  API_DELAY,
  STORAGE_PREFIX,
  PRIORITIES,
}

const isTestSession = localStorage.getItem('isTestRunning');
if (isTestSession) {
  window._testSession = window._testSession || {};
  window._testSession.config = config;
}

export default config;
```

## Настройки приложения - setup first

```
import config from './config';  
  
let list = [];  
  
const { STORAGE_PREFIX } = config;  
  
const getKey = () => {  
  const key = `${STORAGE_PREFIX}__tasks`;  
  return key;  
}
```

## Настройки приложения - setup first

```
E = [],  
  b = h.STORAGE_PREFIX,  
  0 = function () {  
    | return "".concat(b, "__tasks");  
  },  
  y = n(13),
```

## Кстати об анимациях

```
.no-animations *,
.no-animations *:before,
.no-animations *:after {
  /* disable transitions */
  -webkit-transition-property: none !important;
  transition-property: none !important;

  /* disable transforms */
  -webkit-transform: none !important;
  transform: none !important;

  /* disable animations */
  -webkit-animation: none !important;
  animation: none !important;
}
```

## Старт приложения

```
ReactDOM.render(  
  <Provider store={store}>  
    <App />  
  </Provider>,  
  document.getElementById( 'root' )  
)
```

## Отложенный старт приложения

```
function start() {  
  ReactDOM.render(  
    <Provider store={store}>  
      <App />  
    </Provider>,  
    document.getElementById('root')  
  )  
}
```

## Отложенный старт приложения

```
const isTestSession = sessionStorage.getItem('testSession');
if (!isTestSession) {
  start();
} else {
  const timer = setInterval(() => {
    if (window._testSession && window._testSession.startApp) {
      clearInterval(timer);
      start();
    }
  }, 500);
}
```

```
driver.exec_script('window._testSession.startApp = true');
```

## Тестовая сессия - настройка окружения

- использование storage для передачи данных и настроек
- создание “*посадочной страницы*” для настроек

## Логика на клиенте

- Связанная с логикой клиента
- Serverless

## Логика на клиенте - где и как

- State management library
- Компоненты

# Redux

- React
- Angular
- Vue
- Ember

## Redux - state

```
{  
  items: [],  
  filter: '',  
  showCompleted: false,  
}
```

## Redux - reducer

```
function reducer(state = DEFAULT_STATE, action) {  
  switch (action.type) {  
    // case X: return newState;  
    default:  
      return state;  
  }  
}
```

## Redux - reducer. Установка начального состояния

```
const lsValue = localStorage.getItem('DEFAULT_STATE');  
const DEFAULT_STATE = lsValue ? JSON.parse(lsValue) : {  
  items: [],  
  filter: '',  
  showCompleted: false,  
}
```

## Redux - action

```
{  
  type: 'ADD_ITEM',  
  payload: {  
    title: 'New task',  
    priority: 3  
  }  
}
```

## Redux - action creator

```
function addItem({ title, priority }) {  
  return {  
    type: 'ADD_ITEM',  
    payload: {  
      title,  
      priority  
    }  
  }  
}
```

## Redux - action creator as dispatch wrapper

```
function addItem({ title, priority }) {  
  return function (dispatch) {  
    // some logic here  
    dispatch({  
      type: 'ADD_ITEM',  
      payload: {  
        title,  
        priority  
      }  
    });  
  };  
}
```

## Redux - middleware

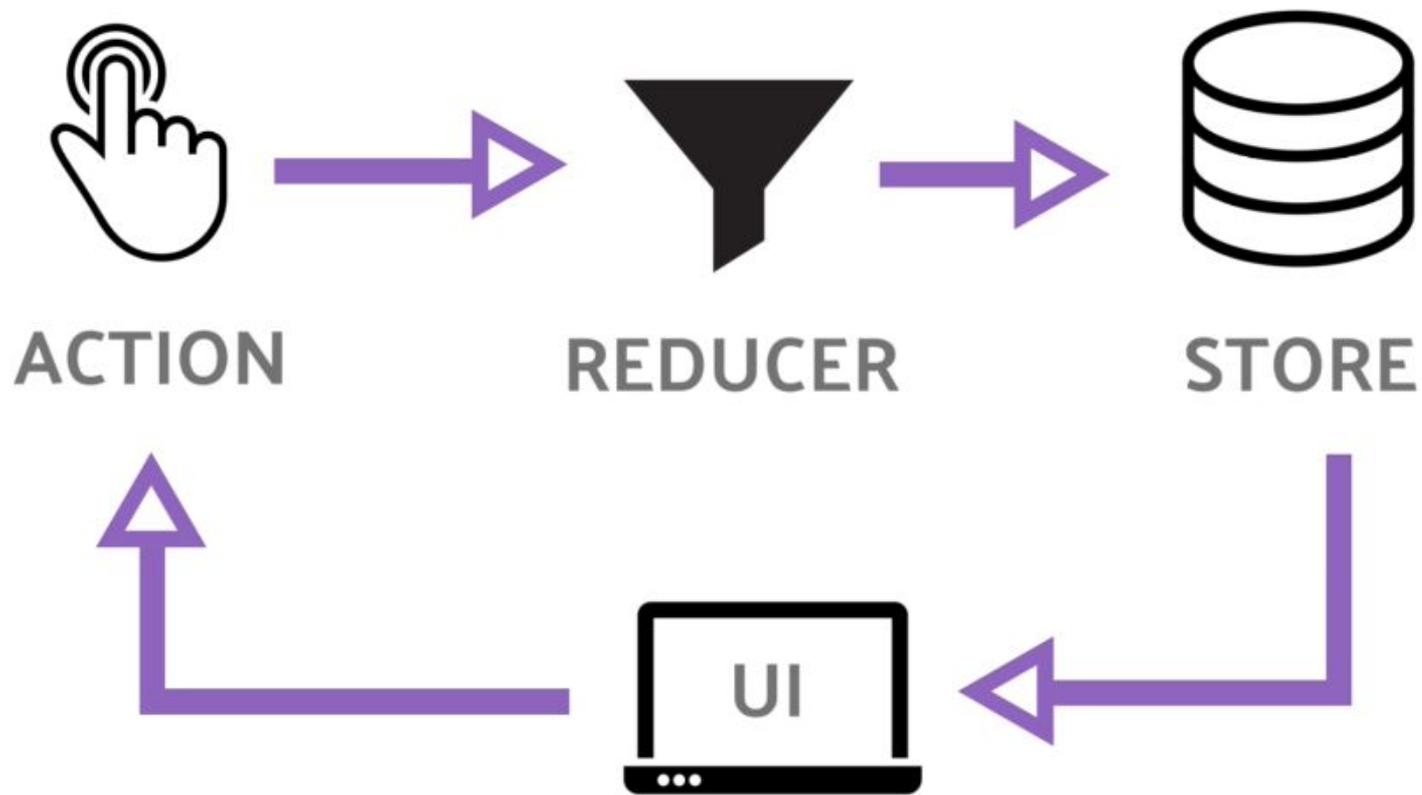
```
// middleware = store => next => action => next(action)
function middleware(store) {
  return function (next) {
    return function (action) {
      next(action)
    }
  }
}
```

## Redux - store

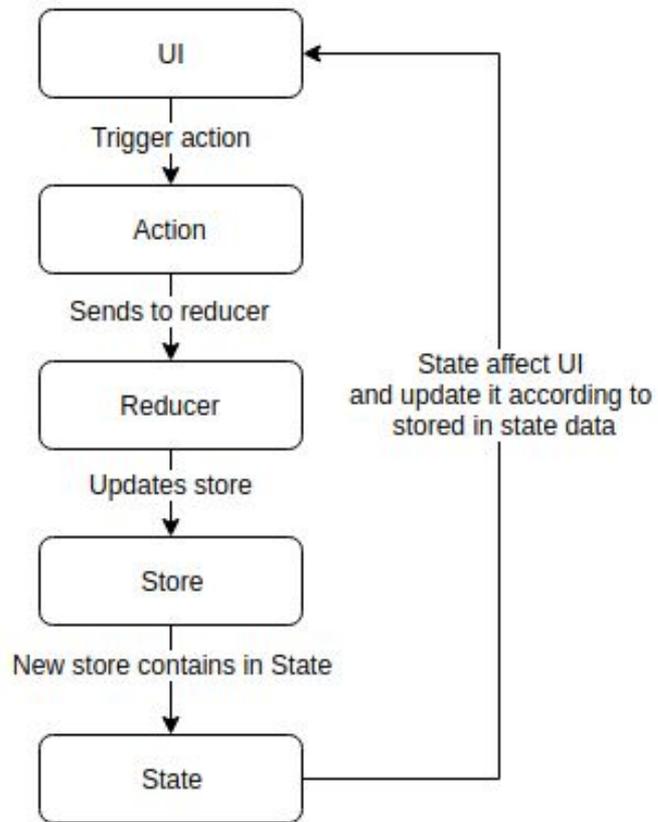
```
const store = createStore(  
  reducer,  
  composeEnhancers(applyMiddleware(...middlewares))  
);
```

- .dispatch()
- .getState()
- .subscribe()

## Redux - синхронный



# Redux - синхронный



## Redux - асинхронность в компоненте

```
onClick = () => api.someAsyncOperation()  
  .then((data) => this.props.dispatch(  
    addItem(data)  
  ));
```

```
onClick = () => api.someAsyncOperation()  
  .then((data) => this.props.addItem(data));
```

## Redux - асинхронность в action creator

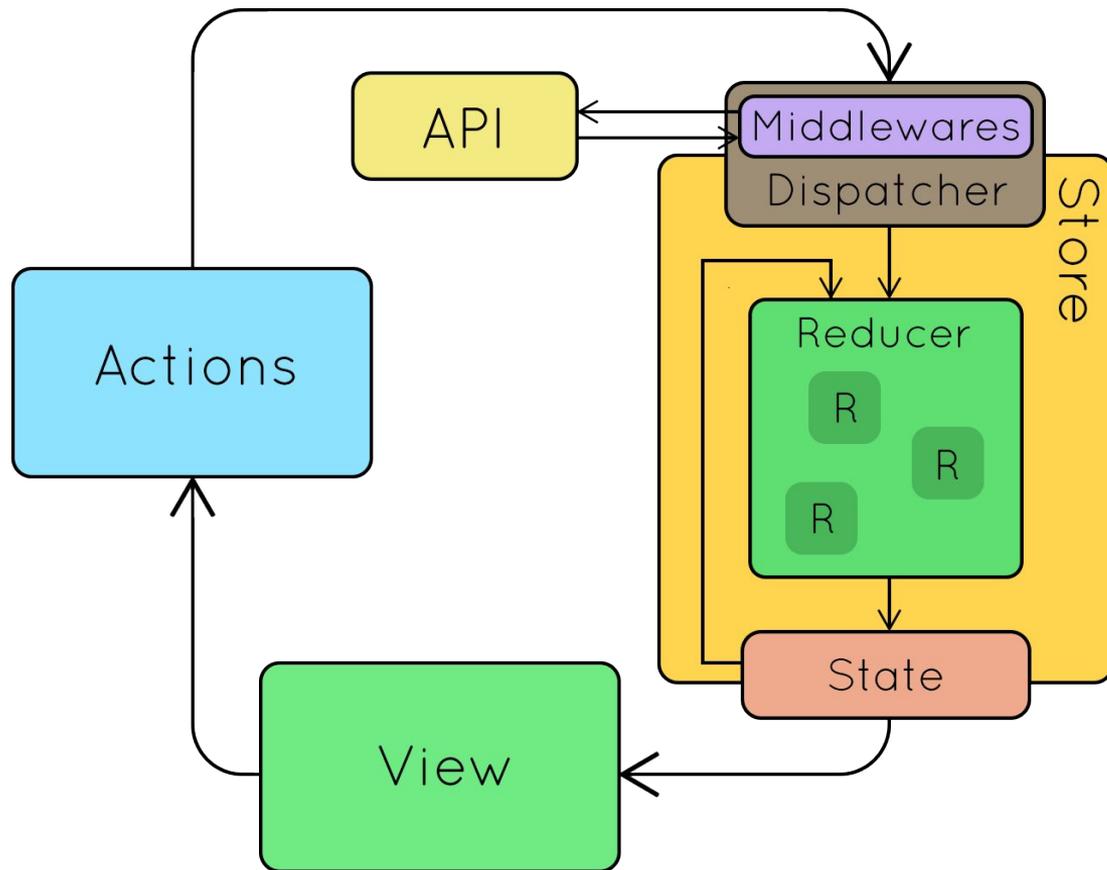
```
const addItem = () => dispatch => api.someAsyncOperation()  
  .then(data => dispatch({  
    type: 'ADD_ITEM',  
    payload: data  
  }));
```

```
onClick = () => this.props.dispatch(addItem());  
// or  
onClick = () => this.props.addItem();
```

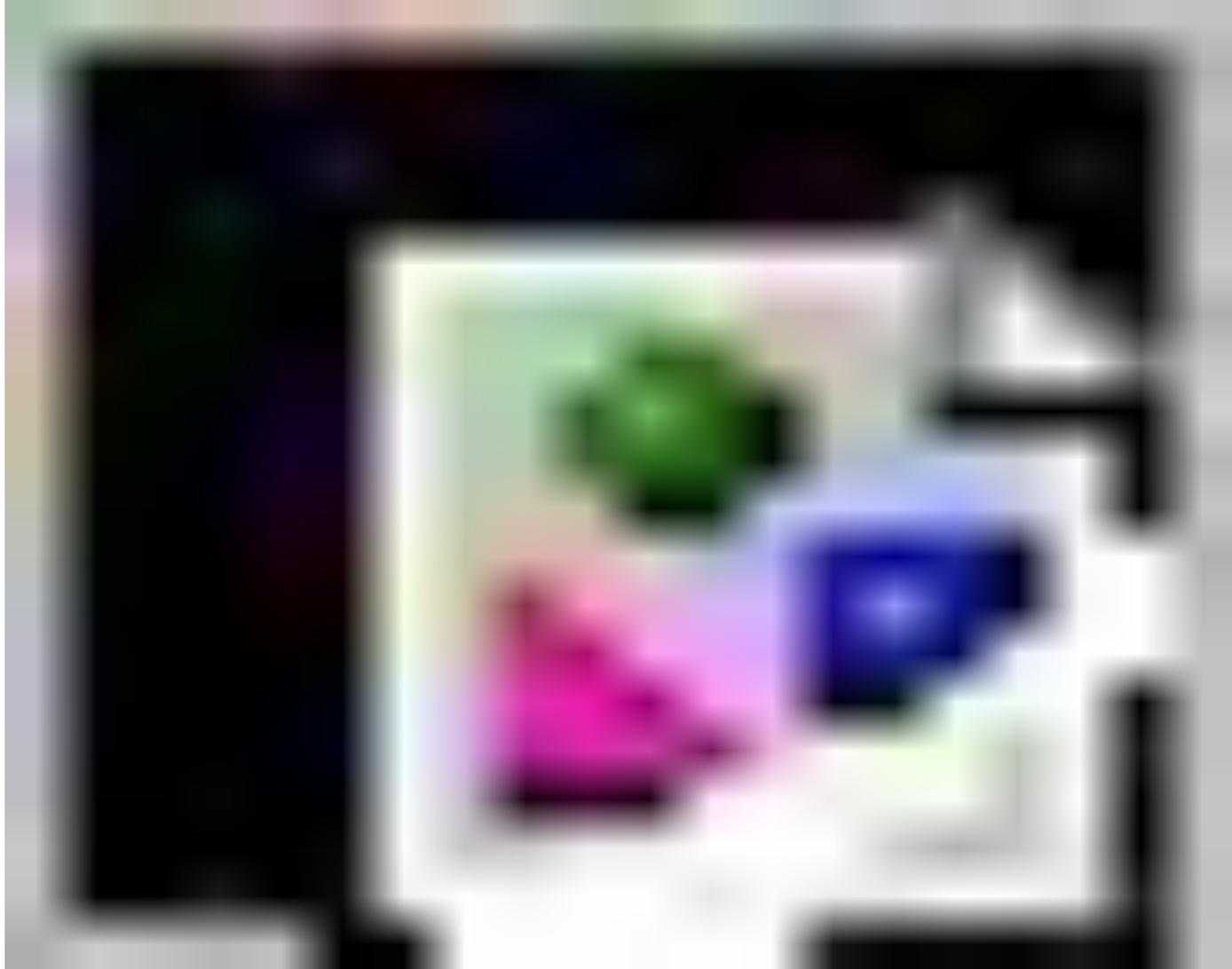
## Redux - асинхронность в middleware

```
const middleware = store => next => action => {  
  if (action.type === 'INITIATE_ADD_ITEM') {  
    api.someAsyncOperation(action.payload)  
      .then((data) => store.dispatch({  
        type: 'ADD_ITEM',  
        payload: data,  
      }));  
  }  
  next(action);  
}
```

# Redux - схема асинхронности



## Redux - Dev tools



## Redux - expose api

```
import store from './redux/store';
import * as actions from './redux/actions';

const isTestSession = localStorage.getItem('isTestRunning');
if (isTestSession) {
  window._testSession = window._testSession || {};
  window._testSession.store = store;
  window._testSession.actions = actions;
}
```

## Redux - example

```
window._testSession.store.dispatch(  
  window._testSession.actions.addTask({  
    title: 'XXX',  
    priority: 1  
  })  
)
```

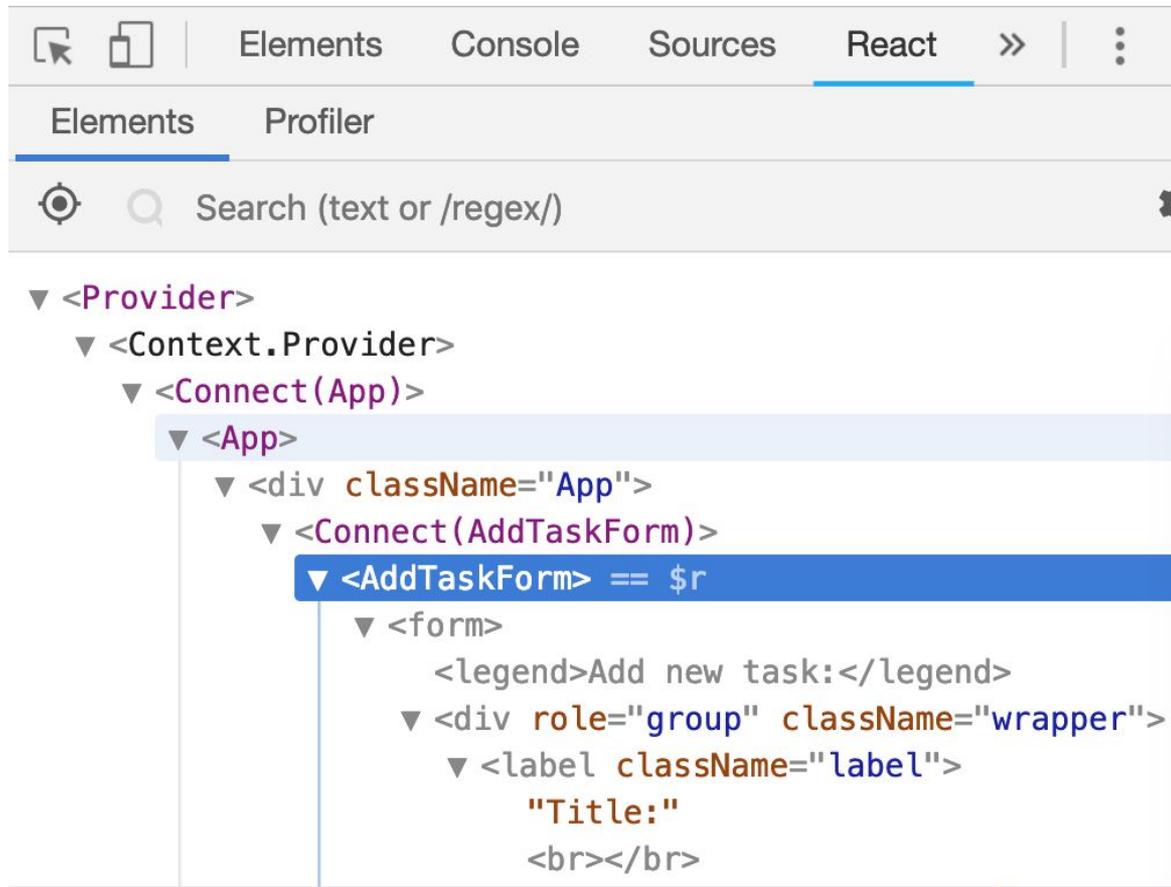
## Состояние в компоненте

```
class Form extends React.Component {  
  state = {  
    isOpen: false,  
    name: ' ',  
    items: []  
  }  
  // ...  
}
```

## Состояние в компоненте - изменение состояния

```
onClick = () => this.setState({  
  isOpen: true,  
  name: 'vscode'  
})
```

# React - dev tools



The screenshot shows the React DevTools interface with the React tab selected. The component tree is expanded to show the following structure:

```
▼ <Provider>  
  ▼ <Context.Provider>  
    ▼ <Connect(App)>  
      ▼ <App>  
        ▼ <div className="App">  
          ▼ <Connect(AddTaskForm)>  
            ▼ <AddTaskForm> == $r  
              ▼ <form>  
                <legend>Add new task:</legend>  
                ▼ <div role="group" className="wrapper">  
                  ▼ <label className="label">  
                    "Title:"  
                    <br></br>
```

# React Dev tools

The screenshot shows the React DevTools component inspector. The top section displays the component tree for `<AddTaskForm>`. The tree structure is as follows:

- `<AddTaskForm> == $r`
  - `<form>`
    - `<legend>Add new task:</leg`
    - `<div role="group" classNar`
      - `<label className="label'`
        - `"Title:"`
        - `<br></br>`

Below the component tree, a breadcrumb trail shows the component path: `xt.Provider` `Connect(App)` `App` `div` `Connect(AddTaskFo`.

The **Props** section shows:

- `onSubmit: fn()`

The **State** section shows:

- `priority: 3`
- `title: ""`

# Доступ к экземплярам компонентов

Here's what I use: (updated to work for React <16 and 16+)

```
window.FindReact = function(dom) {
  let key = Object.keys(dom).find(key=>key.startsWith("__reactInternalInstance$"));
  let internalInstance = dom[key];
  if (internalInstance == null) return null;

  if (internalInstance.return) { // react 16+
    return internalInstance._debugOwner
      ? internalInstance._debugOwner.stateNode
      : internalInstance.return.stateNode;
  } else { // react <16
    return internalInstance._currentElement._owner._instance;
  }
}
```

## Состояние в компоненте - изменение из тестов

```
FindReact(document.querySelector('form')).setState({  
  title: 'vvscode'  
})
```

# Скрипты мониторинга



**Catch errors before  
your users do**

Resolve errors in minutes, and  
deploy with confidence



**Fewer bugs.  
More shipped code.**

Understand the errors that affect your  
users. Detect the root causes automatically.  
Give your developers 10 years more  
experience overnight.

## Скрипты мониторинга - Контекст и тело отчета

- Stacktrace ошибки
- Окружение (браузер, разрешение экрана и тп)
- Sourcemaps
- “Контекст”  
(история действий, данные о сессии и тп)

## Скрипты мониторинга - применение для тестов

```
driver.execute_script(  
    "throw new Error('TC-XXX User Auth: Captcha not displayed')"  
)
```

**Давайте дружить**

