Attractor Software

Оптимизация React приложений

https://github.com/teimurjan/react-optimization-presentation

- Размер bundle
- Вызовы render
- Использование redux

ExtractTextPlugin

```
const ExtractTextPlugin = require('extract-text-webpack-plugin');

module.exports = {
    module: {
        rules: [
            {
                oneOf: [{
                    test: /\.css$/,
                    loader: ExtractTextPlugin.extract({/* your loader here */})
                }]
            }
        ]
    }
}
```

Минимизация CSS

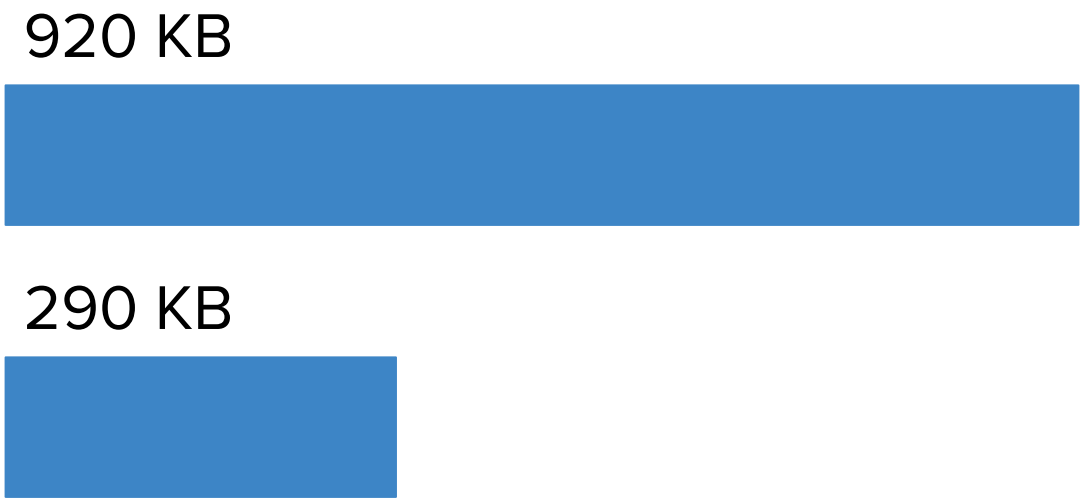
```
loader: ExtractTextPlugin.extract({
    use: [{
        loader: require.resolve('css-loader'),
        options: {
            minimize: true
        }
    }]
})
```

*.css -= 45 KB (~25%)

NODE_ENV = production

```
const webpack = require('webpack');

module.exports = {
    plugins: [
        new webpack.DefinePlugin({
            'process.env': {
                NODE_ENV: JSON.stringify('production')
            }
        })
    ]
}
```

*.js -= 630 KB (~69%)



UglifyJsPlugin

```
const webpack = require('webpack');

module.exports = {
    plugins: [
        new webpack.optimize.UglifyJsPlugin({
            /* settings */
        })
    ]
}
```

*.js -= 136 KB (~47%)

290 KB

154 KB

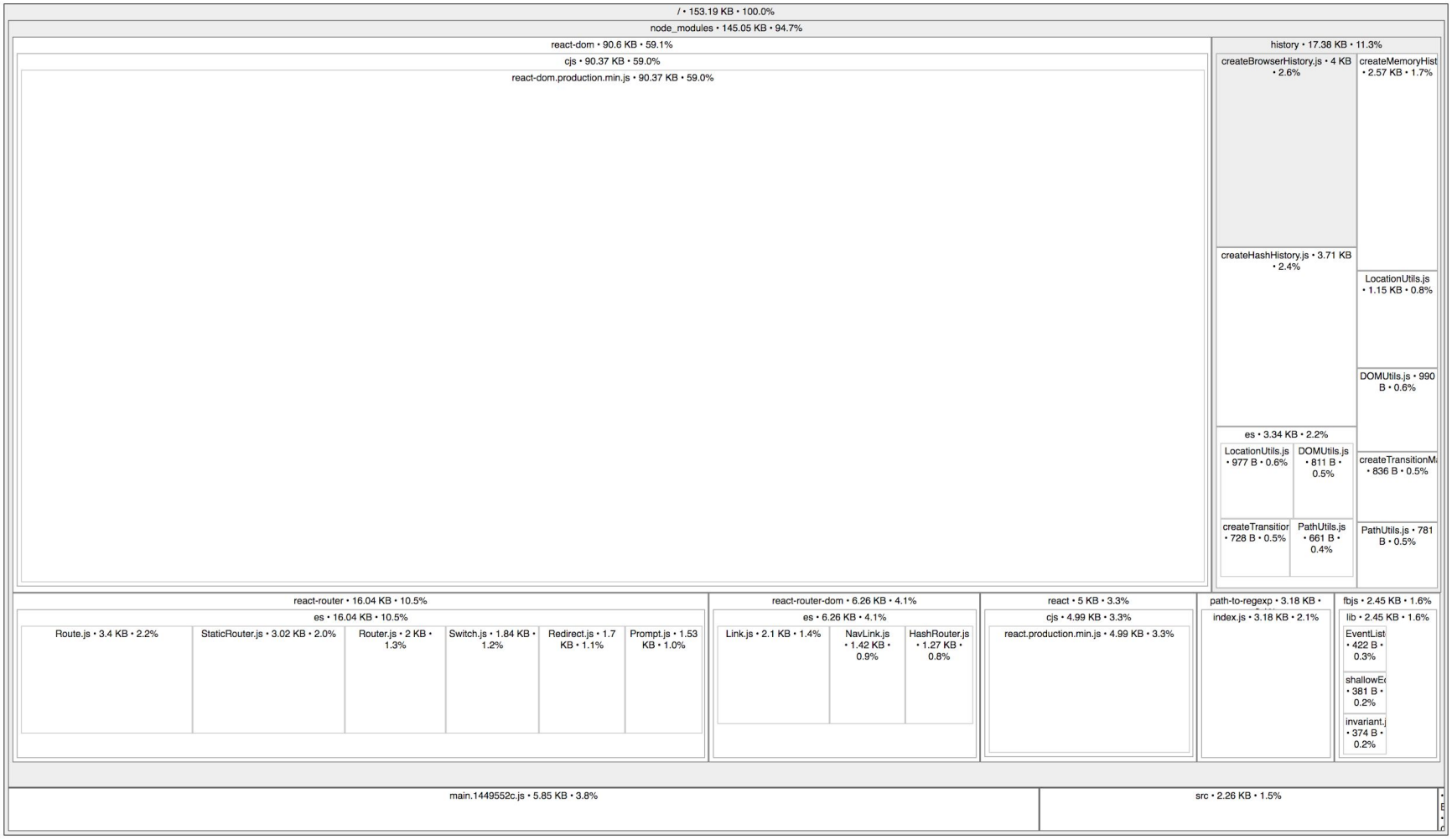
Другие методы

- Убрать prop types из bundle - babel-plugin-transform-react-remove-prop-types
- Установить ограничения по поддержке браузеров для babel

Анализ bundle с помощью

- source-map-explorer

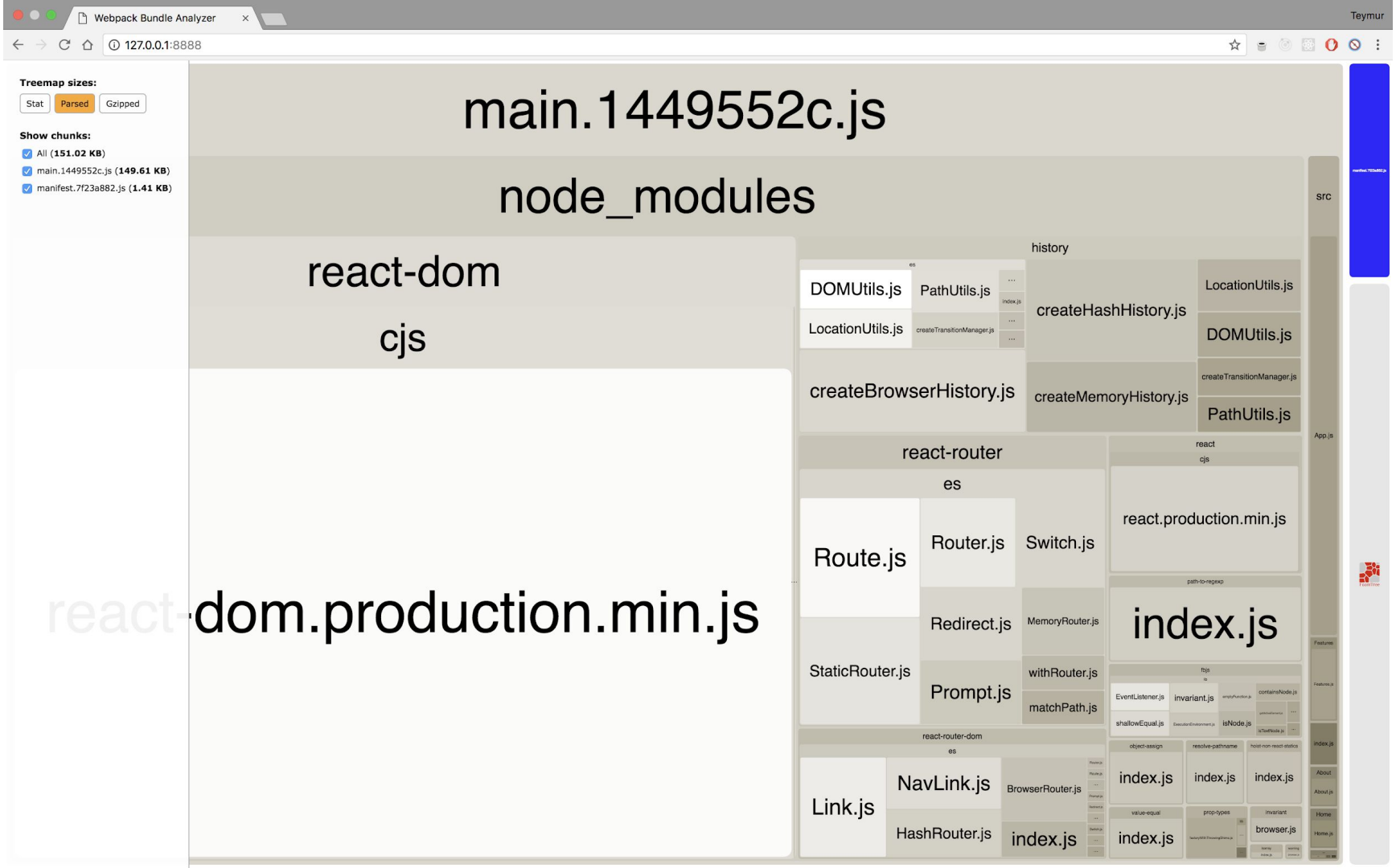
```
npm install -g source-map-explorer
source-map-explorer bundle.js bundle.js.map
```


/ • 153.19 KB • 100.0%
node_modules • 145.05 KB • 94.7%
react-dom • 90.6 KB • 59.1%
cjs • 90.37 KB • 59.0%
react-dom.production.min.js • 90.37 KB • 59.0%
history • 17.38 KB • 11.3%
createBrowserHistory.js • 4 KB • 2.6%
createMemoryHist • 2.57 KB • 1.7%
createHashHistory.js • 3.71 KB • 2.4%
LocationUtils.js • 1.15 KB • 0.8%
DOMUtils.js • 990 B • 0.6%
es • 3.34 KB • 2.2%
LocationUtils.js • 977 B • 0.6%
DOMUtils.js • 811 B • 0.5%
createTransitionM • 836 B • 0.5%
createTransitior • 728 B • 0.5%
PathUtils.js • 661 B • 0.4%
PathUtils.js • 781 B • 0.5%
react-router • 16.04 KB • 10.5%
es • 16.04 KB • 10.5%
Route.js • 3.4 KB • 2.2%
StaticRouter.js • 3.02 KB • 2.0%
Router.js • 2 KB • 1.3%
Switch.js • 1.84 KB • 1.2%
Redirect.js • 1.7 KB • 1.1%
Prompt.js • 1.53 KB • 1.0%
react-router-dom • 6.26 KB • 4.1%
es • 6.26 KB • 4.1%
Link.js • 2.1 KB • 1.4%
NavLink.js • 1.42 KB • 0.9%
HashRouter.js • 1.27 KB • 0.8%
react • 5 KB • 3.3%
cjs • 4.99 KB • 3.3%
react.production.min.js • 4.99 KB • 3.3%
path-to-regexp • 3.18 KB •
index.js • 3.18 KB • 2.1%
fbjs • 2.45 KB • 1.6%
lib • 2.45 KB • 1.6%
EventList • 422 B • 0.3%
shallowE • 381 B • 0.2%
invariant.j • 374 B • 0.2%
main.1449552c.js • 5.85 KB • 3.8%
src • 2.26 KB • 1.5%

А также с помощью

- webpack-bundle-analyzer

```
const BundleAnalyzerPlugin = require('webpack-bundle-analyzer').BundleAnalyzerPlugin;

module.exports = {
    plugins: [
        new BundleAnalyzerPlugin()
    ]
}
```


Webpack Bundle Analyzer
Teymur
127.0.0.1:8888
Treemap sizes:
Stat
Parsed
Gzipped
Show chunks:
All (151.02 KB)
main.1449552c.js (149.61 KB)
manifest.7f23a882.js (1.41 KB)
main.1449552c.js
node_modules
react-dom
cjs
react-dom.production.min.js
history
es
DOMUtils.js
PathUtils.js
LocationUtils.js
createTransitionManager.js
createHashHistory.js
createBrowserHistory.js
createMemoryHistory.js
react-router
es
Route.js
Router.js
Switch.js
Redirect.js
MemoryRouter.js
StaticRouter.js
Prompt.js
withRouter.js
matchPath.js
react-router-dom
es
Link.js
NavLink.js
BrowserRouter.js
HashRouter.js
index.js
react
cjs
react.production.min.js
path-to-regexp
index.js
fbjs
EventListener.js
invariant.js
containsNode.js
shallowEqual.js
isNode.js
object-assign
index.js
resolve-pathname
index.js
hoist-non-react-statics
index.js
value-equal
index.js
prop-types
invariant
browser.js
src
App.js
Features
index.js
About
About.js
Home
Home.js


DIVIDE
CONQUER

Делим bundle на vendor и client

```
const webpack = require('webpack');

module.exports = {
    entry: {
        client: './src/index.js',
        vendor: ['react', 'react-dom', 'react-router', 'react-router-dom']
    },
    plugins: [
        new webpack
            .optimize
            .CommonsChunkPlugin({name: 'vendor', filename: 'vendor.[chunkhash].js'})
    ]
};
```

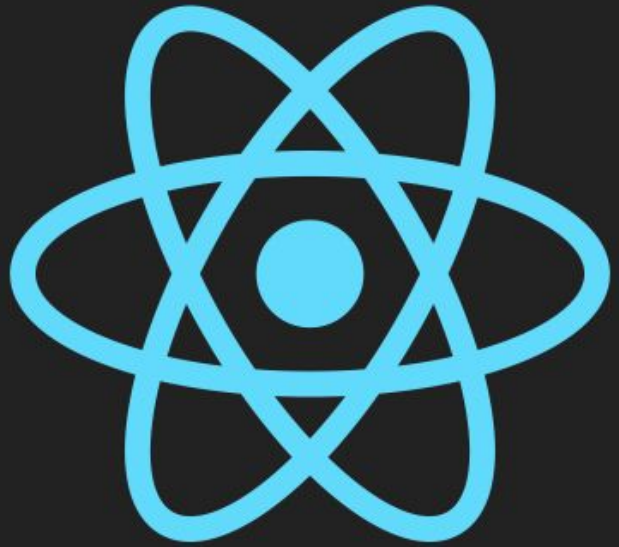
Делим client на чанки

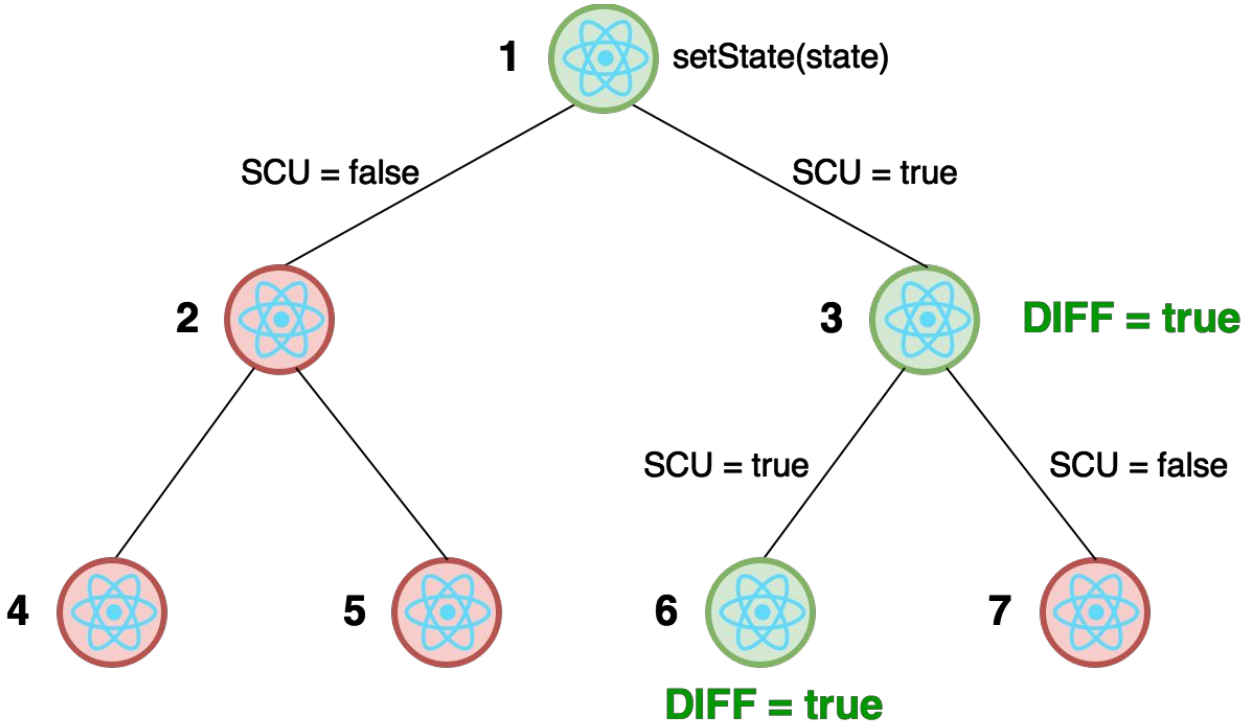
```
const webpack = require('webpack');

module.exports = {
    output: {
        path: buildPath,
        filename: '[name].[chunkhash:8].js',
        chunkFilename: '[name].[chunkhash:8].chunk.js'
    }
};
```

```jsx
// Home.js
import React from 'react';

export default () => (
    <div className="wrapper">
        <h2>Home</h2>
    </div>
);
```

```
// HomeContainer.js
import Loadable from 'react-loadable';
import Loading from '../Common/Loading';

export default Loadable({
    loader: () => import ('./Home' /* webpackChunkName: 'home' */),
    loading: Loading
})
```

Reconciliation





- RERENDERED

- NOT RERENDERED

SCU - shouldComponentUpdate

React.PureComponent

```
import React from 'react';

export default class extends React.PureComponent {
    shouldComponentUpdate(nextProps, nextState) {
        return !(
            shallowEqual(nextProps, this.props) &&
            shallowEqual(nextState, this.state)
        );
    }
}
```

Проблемы React.PureComponent

- Arrow-функции и .bind(this)
- Создание и модификация массивов и объектов

```
class TodoList extends React.Component {
    render() {
        return (
            <ul>
                {this.props.todos.map(todo => (
                    <Todo
                        todo={todo}
                        onClick={() => this.props.onTodoClick(todo.id)}
                    />
                ))}
            </ul>
        );
    }
}
```

```
class TodoList extends React.Component {
    render() {
        return (
            <ul>
                {this.props.todos.map(todo ⇒ (
                    <Todo
                        todo={todo}
                        onClick={this.props.onTodoClick}
                    />
                ))}
            </ul>
        );
    }
}
```

```
class Todo extends React.PureComponent {
    handleClick = e ⇒ {
        this.props.onClick(this.props.todo.id);
    }

    render() {
        return (
            <li className="todo" onClick={this.handleClick}>
                {this.props.todo}
            </li>
        )
    }
}
```

```
render() {
    return (
        <ul>
            {this.props.todos.map(todo => (
                <Todo
                    todo={todo}
                    onClick={this.props.handleClick.bind(this)}
                />
            ))}
        </ul>
    );
}
```

```
constructor(props) {
    super(props);
    this.handleClick = this.handleClick.bind(this);
}

render() {
    return (
        <ul>
            {this.props.todos.map(todo => (
                <Todo
                    todo={todo}
                    onClick={this.props.handleClick}
                />
            ))}
        </ul>
    );
}
```

```
handleClick = (todoId) ⇒ {
    this.setState({
        [todoId]: {clicked: true}
    });
}
```

```
const arr1 = [1, 2, 3];
const arr2 = [1, 2, 3];

arr1 === arr2 // false

const obj1 = {foo: 'bar'};
const obj2 = {foo: 'bar'};

obj1 === obj2 // false
```

```
render() {
    return (
        <TodoList
            options={{
                wrap: false,
                maximizeOnFocus: true
            }}
        />
    );
}
```

```
const TODO_LIST_OPTIONS = {
    wrap: false,
    maximizeOnFocus: true
};

export default class extends React.Component {
    render() {
        return (
            <TodoList
                options={TODO_LIST_OPTIONS}
            />
        );
    }
}
```

Использование индекса элемента как key

```
const elements = [
    {type: 'div', key: 0, textContent: 'Container #0'},
    {type: 'div', key: 1, textContent: 'Container #1'},
    {type: 'div', key: 2, textContent: 'Container #2'},
    {type: 'div', key: 3, textContent: 'Container #3'},
];

// Удалим элемент под индкесом 1
const elements = [
    {type: 'div', key: 0, textContent: 'Container #0'},
    // Render будет бесполезно вызван для всех элементов ниже
    {type: 'div', key: 2, textContent: 'Container #2'},
    {type: 'div', key: 3, textContent: 'Container #3'},
];
```

Решение проблемы

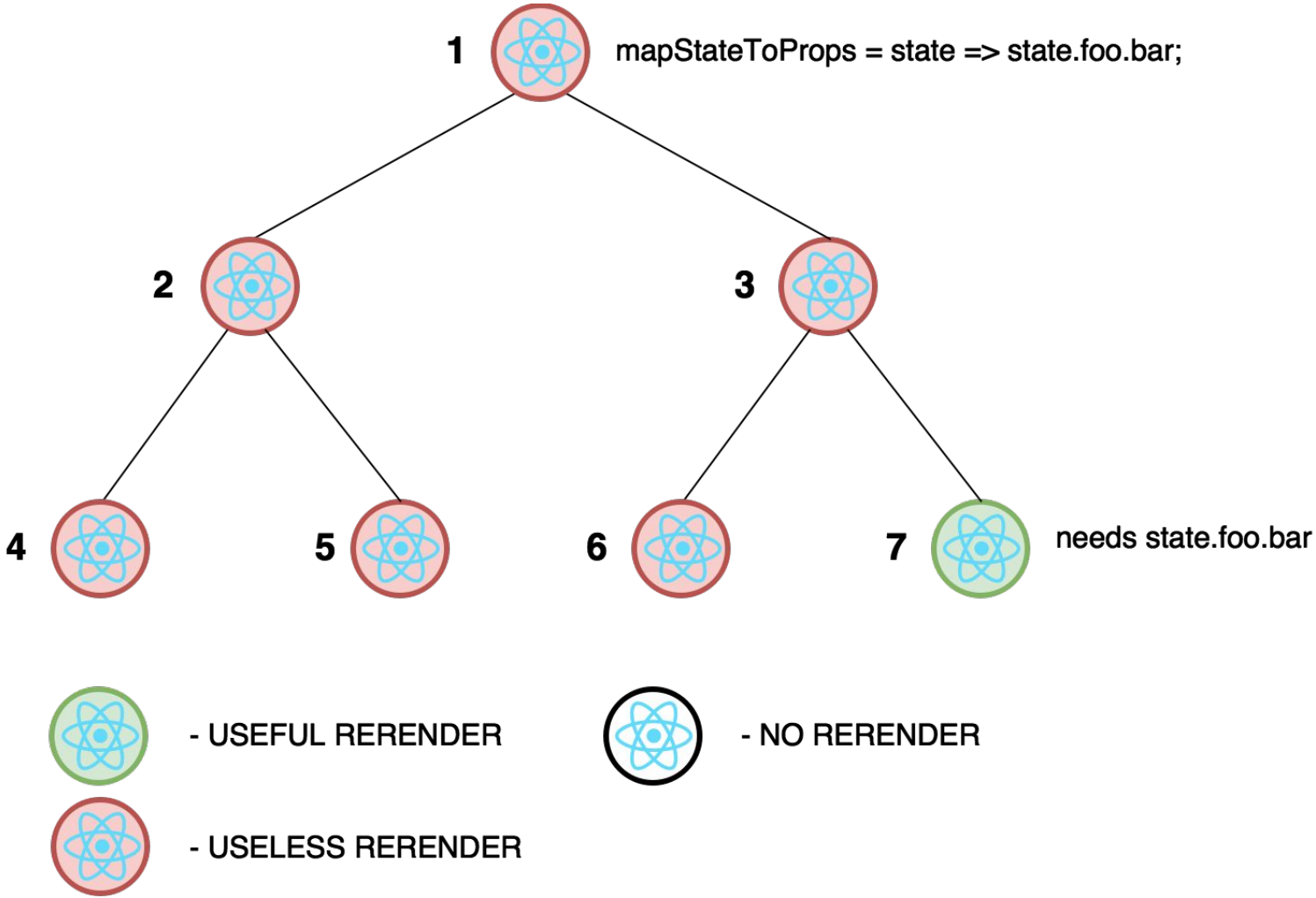
- Использование существующих стабильных ключей
- Генерация стабильных ключей во время mount (И ни в коем случае во время render!)

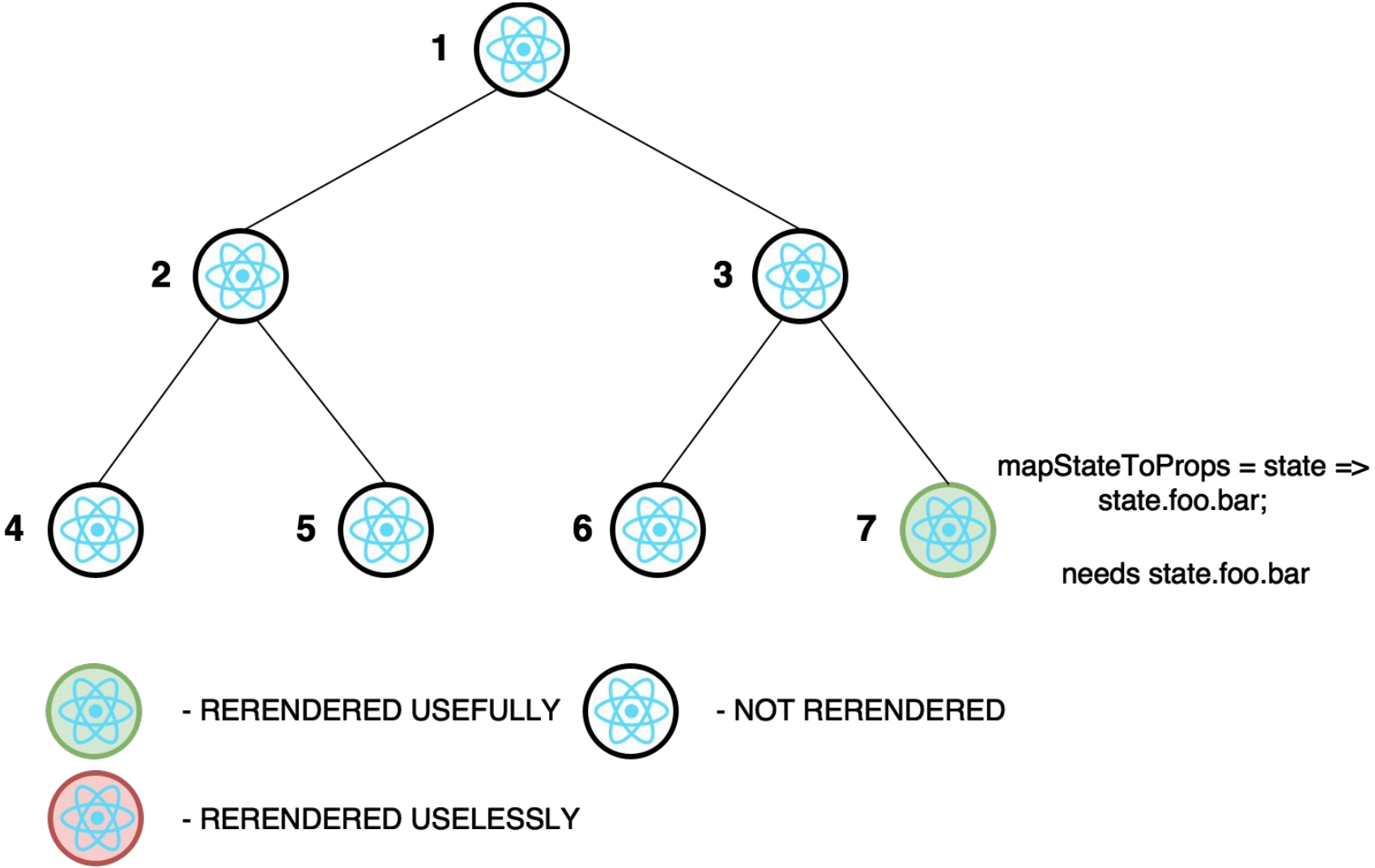
Пасхалочка

```
import React from 'react';

const TodoFactory = ({ todo, onClick }) ⇒ (
    <li className="todo" onClick={onClick}>
        {todo.title}
    </li>
);

export default ({ todos }) ⇒ (
    <ul>
        {todos.map(todo ⇒ TodoFactory({ todo, onClick: console.log }))}
    </ul>
)
```

benchmark: https://github.com/missive/functional-components-benchmark

Производительность Redux


1
mapStateToProps = state => state.foo.bar;
2
3
4
5
6
7
needs state.foo.bar
- USEFUL RERENDER
- NO RERENDER
- USELESS RERENDER


1
2
3
4
5
6
7
mapStateToProps = state =>
state.foo.bar;
needs state.foo.bar
- RERENDERED USEFULLY
- NOT RERENDERED
- RERENDERED USELESSLY