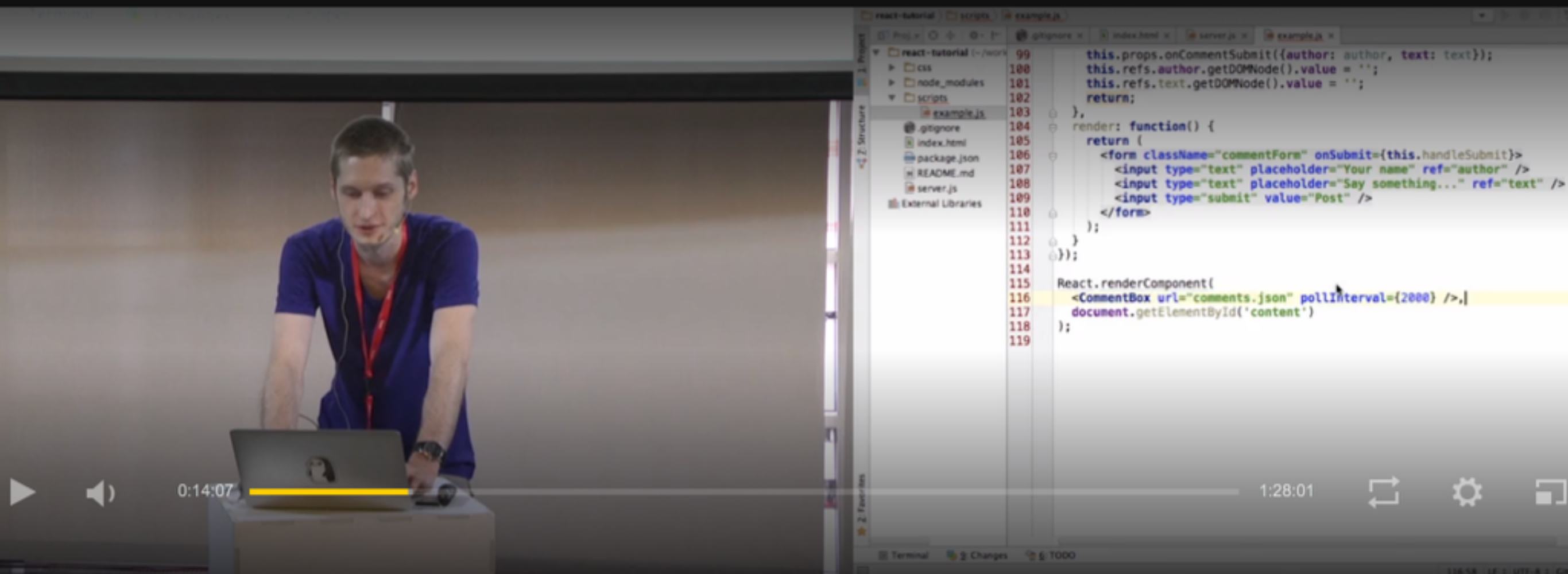


ReactJS и немного БЭМ

@jchouse_ua



БЭМ



Реактивные интерфейсы с React.js и БЭМ: мастер-класс

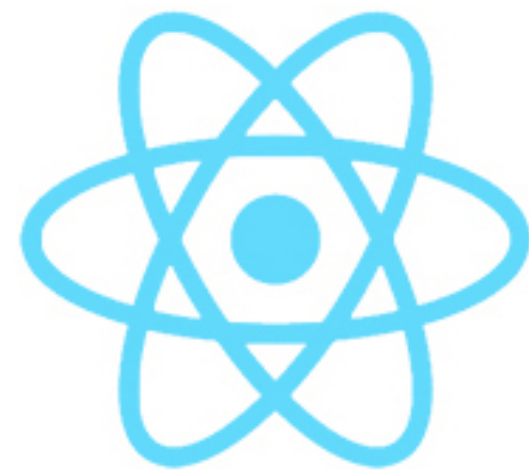
Техноло

БЭМ

БЭМ (Блок, Элемент, Модификатор) – технология разра

MoscowJS 14 — Major Release · 28 августа 2014

Вы наверняка знакомы с React, простым JavaScript-фреймворком от Facebook. Он служит для создания веб-интерфейсов с переиспользуемыми компонентами. На мастер-классе мы



React

Code

```
// tutorial3.js
var CommentBox = React.createClass({
  render: function() {
    return (
      <div className="commentBox">
        <h1>Comments</h1>
        <CommentList />
        <CommentForm />
      </div>
    );
  }
});
```

JSX

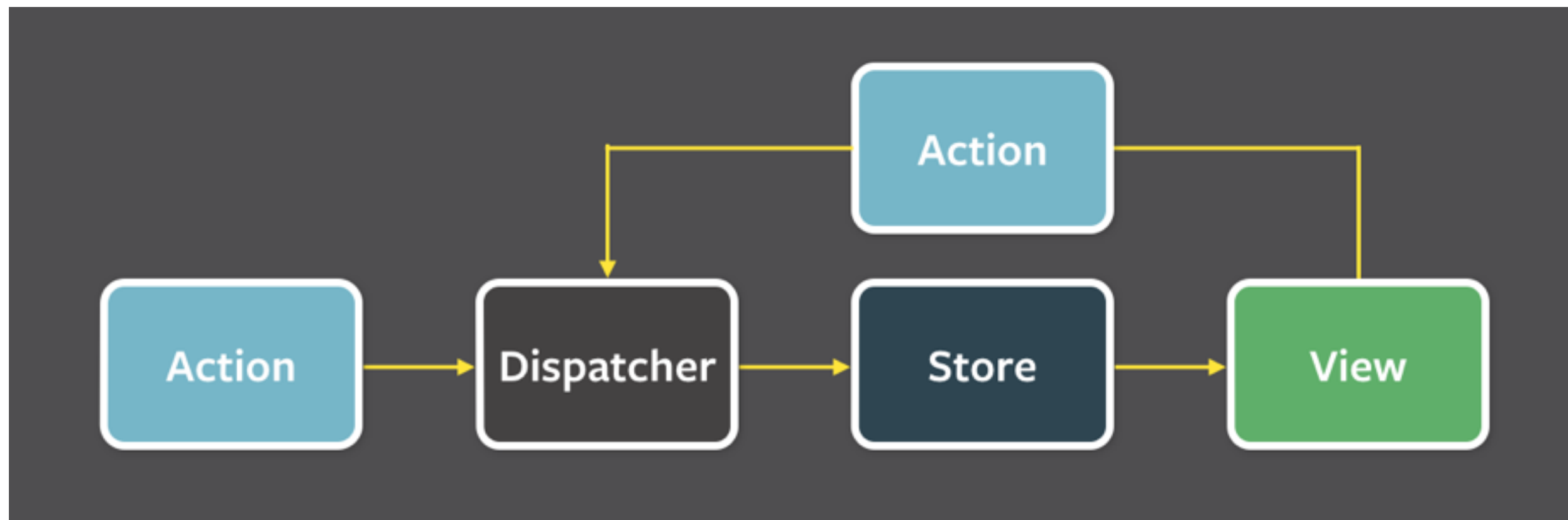
by Facebook

Минусы

- Систему событий (если вам нужны не DOM)
- XHR
- Данные
- Promises
- Роутинг
- Локализацию
- Тестирование (это не совсем правда)
- Инструкцию как собрать все это в кучу

React + Flux





Flux

как работает



Предлагаю такой план:
туда мы идем три книги,

а назад возвращаемся на орлах



GRUNT

The JavaScript Task Runner



// standalone browserify watch

```
client: {  
  src: ['src/blocks/**/*.jsx'],  
  dest: 'build/scripts/index.js',  
  options: {  
    transform: ['babelify', require('grunt-react').browserify],  
    watch: true,  
    keepAlive: true  
  }  
},
```

// working with grunt-watch – do NOT use with keepAlive above

```
watchClient: {  
  src: ['src/blocks/**/*.jsx'],  
  dest: 'build/scripts/index.js',  
  options: {  
    transform: ['babelify', require('grunt-react').browserify],  
    watch: true  
  }  
}
```

```

5  const blockName = 'button';
6
7  /**
8   * Button
9   *
10  * @property {String} text          - array of additional element
11  * @property {String} [name]        - array of additional element
12  * @property {String} [option='default'] - options [default]|reject
13  * @property {String} [mixCls]       - css classes for mix
14  * @property {Array} [children]      - array of additional element,
15  *                                   placement before text
16  * @property {Boolean} [disabled=false] - array of additional element
17  *
18  * @property {function} [clickHandler] - on click callback function
19  */
20
21  class Button extends React.Component {
22      constructor (props) {
23          super(props);
24
25          this.state = {
26              animate: {
27                  animate: false,
28                  top: 0,
29                  left: 0
30              }
31          };
32      }

```

BABEL

ПО ОДНОМУ, СУКИ



ПО ОДНОМУ

{less}

substack / browserify-handbook

Watch

106

Star

3,100

Fork

192

Code

Issues12

Pull requests26

Wiki

Pulse

Graphs

how to build modular applications with browserify

118 commits

1 branch

9 releases

18 contributors

Branch: master

New pull request

New file

Find file

HTTPS

https://github.com/subst

Download ZIP

 substack simplified hmr example further

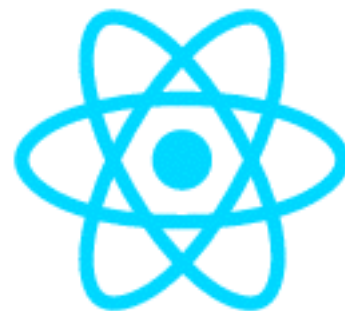
Latest commit e21ccd3 on Sep 1

bin	node packaged manuscript	2 years ago
example	simplified hmr example further	3 months ago
.npmignore	npmignore	3 months ago
LICENSE	package.json etc	2 years ago
package.json	1.8.0	3 months ago
readme.markdown	simplified hmr example further	3 months ago

github.com/substack/browserify-handbook

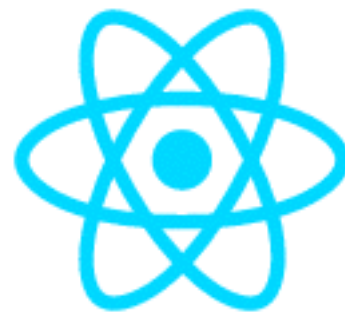
. ./ ./ ./ ./ ./

```
<IntlProvider locale='en'>  
  <Router history={createBrowserHistory()}>  
    <Route path='/' component={App}>  
      <Route path='login' component={Login} />  
      <Route path='registration'>  
        <IndexRoute component={Registration} />  
        <Route path='init/:type' component={Init} />  
      </Route>  
    </Route>  
  </Router>  
</IntlProvider>,
```



React Intl

```
<FormattedMessage  
  id='Registration.checkEmail'  
  defaultMessage='Check your email for invitation to continue'
```



React Intl


```
<Router history={createBrowserHistory()}>  
  <Route path="/" component={App}>  
    <Route path='login' component={Login} />  
    <Route path='registration'>  
      <IndexRoute component={Registration} />  
      <Route path='init/:type' component={Init} />  
    </Route>  
  </Route>  
</Router>
```



REACT/ROUTER

Запускаем!

It's Alive!

Sergeys-MacBook-Pro /Users/jchouse/projects/wsd grunt
Running "exec:set_path" (exec) task

Running "concat:less" (concat) task
File **build/index.less** created.

Running "less:dev" (less) task
File **build/index.css** created

Running "browserify:dev" (browserify) task
>> Bundle **build/scripts/index.js** created.

Running "exec:copy_static_content" (exec) task

Done, without errors.

Тестирование

Самое интересное только начинается



jasmine, mock и istanbul

“Writing tests for promises in jasmine is kind of a pain in the ass.”

– internet

import vs require

CSS

Code

```
var divStyle = {  
  color: 'white',  
  backgroundImage: 'url(' + imgUrl + ')',  
  WebkitTransition: 'all', // note the capital 'W' here  
  msTransition: 'all' // 'ms' is the only lowercase vendor prefix  
};  
  
ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode);
```

Inline Styles

Shorthand for Specifying Pixel Values in style props

When specifying a pixel value for your inline `style` prop, React automatically appends the string "px" for you after your number value, so this works:

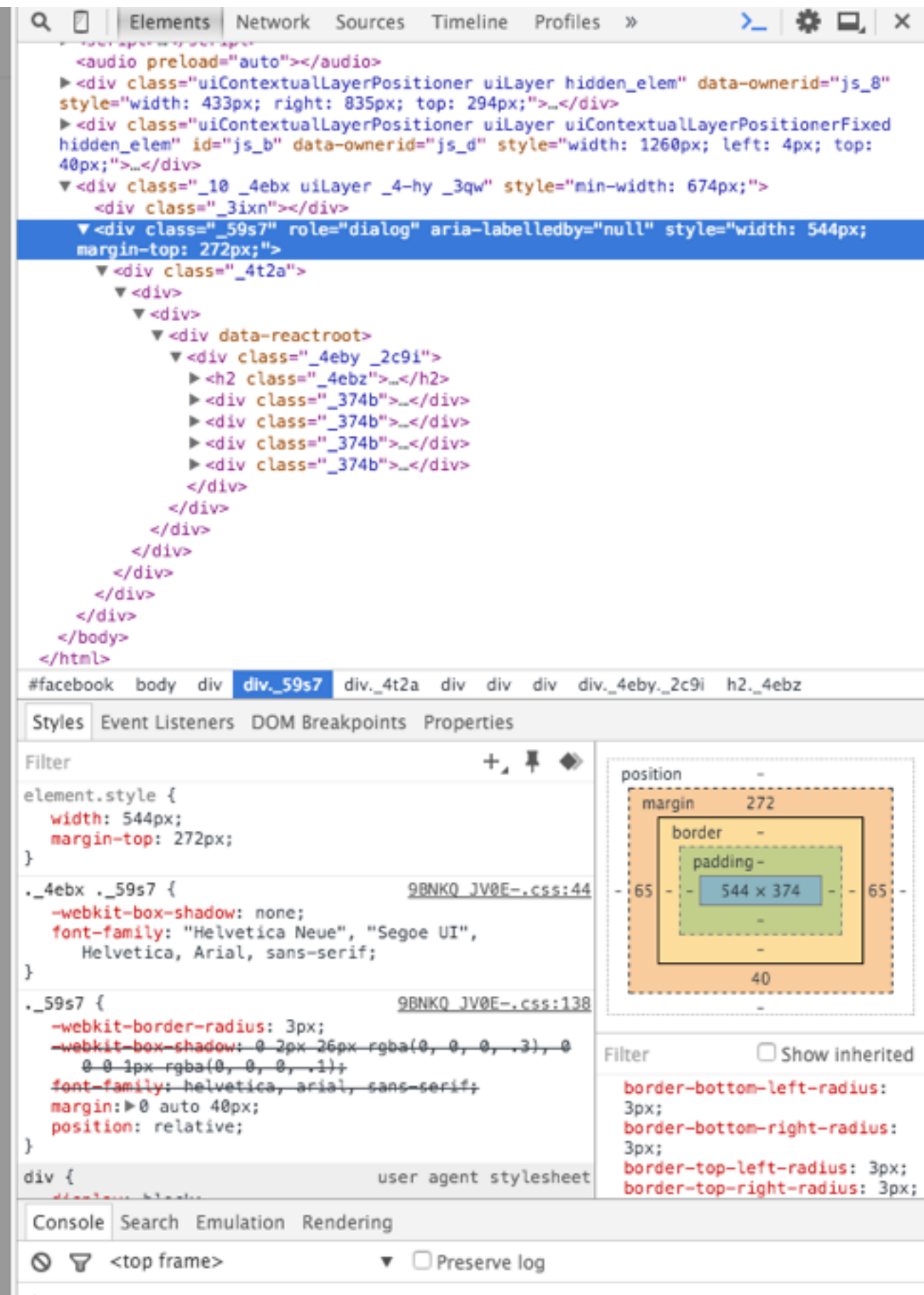
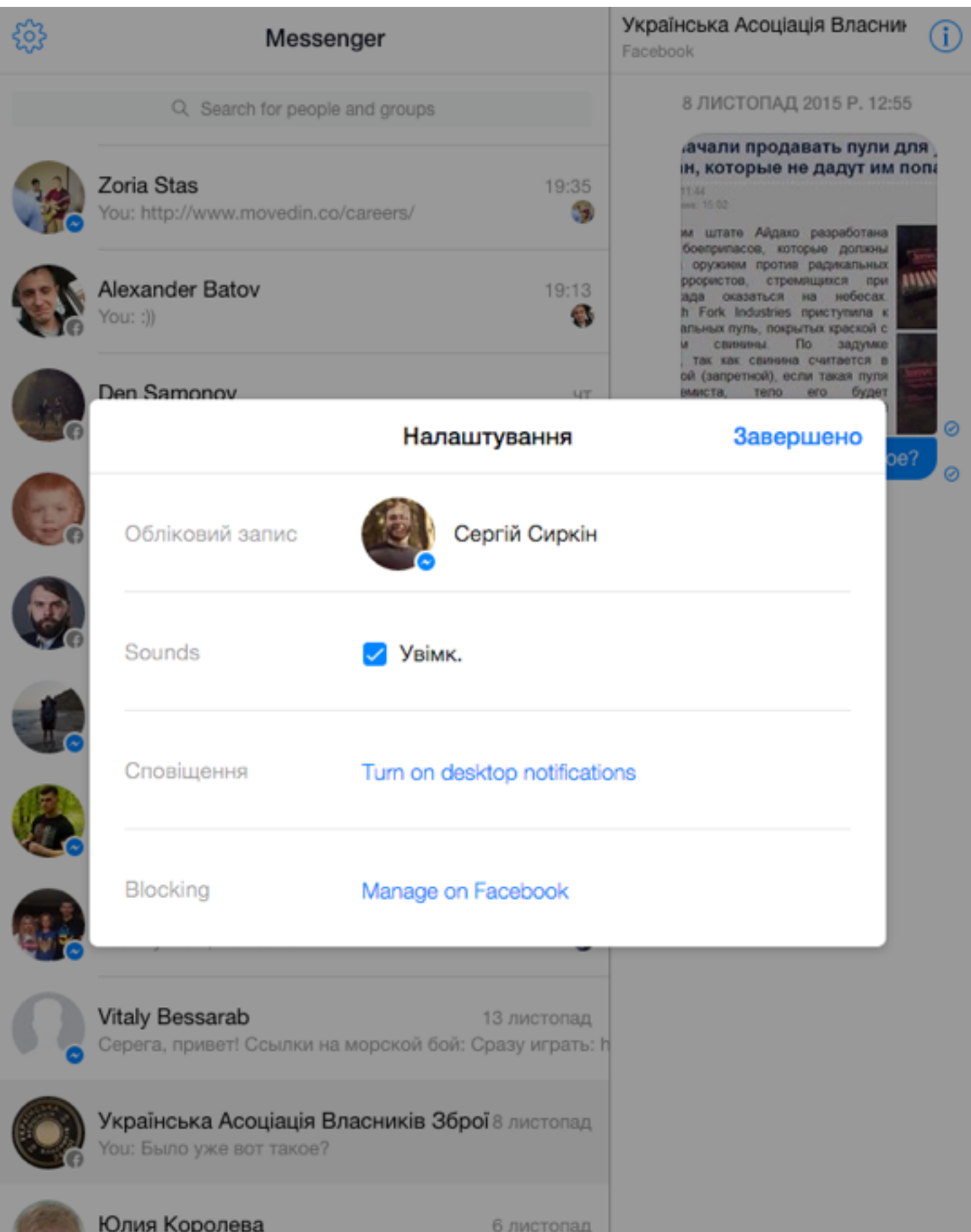
Code

```
var divStyle = {height: 10}; // rendered as "height:10px"
ReactDOM.render(<div style={divStyle}>Hello World!</div>, mountNode);
```

See [Inline Styles](#) for more info.

Sometimes you *do* want to keep the CSS properties unitless. Here's a list of properties that won't get the automatic "px" suffix:

- `animationIterationCount`
- `boxFlex`
- `boxFlexGroup`
- `boxOrdinalGroup`
- `columnCount`
- `fillOpacity`
- `flex`
- `flexGrow`
- `flexPositive`
- `flexShrink`



Впихнуть невпихуемое

```
className= 'some-block-name some-block-name--mod'
```



```
const blockName = 'some-block-name';
```

```
className={blockName + ' ' + blockName + '--mod' }
```

```
const blockName = 'some-block-name';
```

```
className={ [blockName, blockName + '--mod'].join(' ' ) }
```

```

3  class ibem extends React.Component {
4      /**
5       * Make cls for block with elem and mod
6       *
7       * @param {string} blockName    - block name
8       * @param {string} [elem]        - elem name, join with '__'
9       * @param {string} [mod]         - mod name, join with '--'
10     */
11  >   static cls (blockName, elem, mod) {
28
29     /**
30      * Make cls elem name
31      *
32      * @param {string} name          - block name
33      * @param {string} elem          - elem name, join with '__'
34     */
35  >   static elem (name, elem) {
38
39     /**
40      * Make mod name
41      *
42      * @param {string} name          - block or elem name
43      * @param {string} elem          - mod name, join with '--'
44     */
45  >   static mod (name, mod) {
48
49     render () {}
50 }

```

```
4      /**
5       * Make cls for block with elem and mod
6       *
7       * @param {string} blockName    - block name
8       * @param {string} [elem]        - elem name, join with '___'
9       * @param {string} [mod]         - mod name, join with '--'
10      */
11 > static cls (blockName, elem, mod) {
```

```
var content,  
    rowCls = ibem.cls(blockName, 'row');
```

“block-name__row”

```
var cls = ibem.cls(blockName, null, ['selected', 'visible']);
```

“block-name block-name--selected block-name--visible”

```
var cls = ibem.cls(blockName, 'elem', ['selected', 'visible']);
```

“block-name__elem block-name__elem-selected
block-name__elem-visible”

Зачем?

WordPress powers 25% of the internet

[https://github.com/
jchouse/reactjs-starter-kit](https://github.com/jchouse/reactjs-starter-kit)