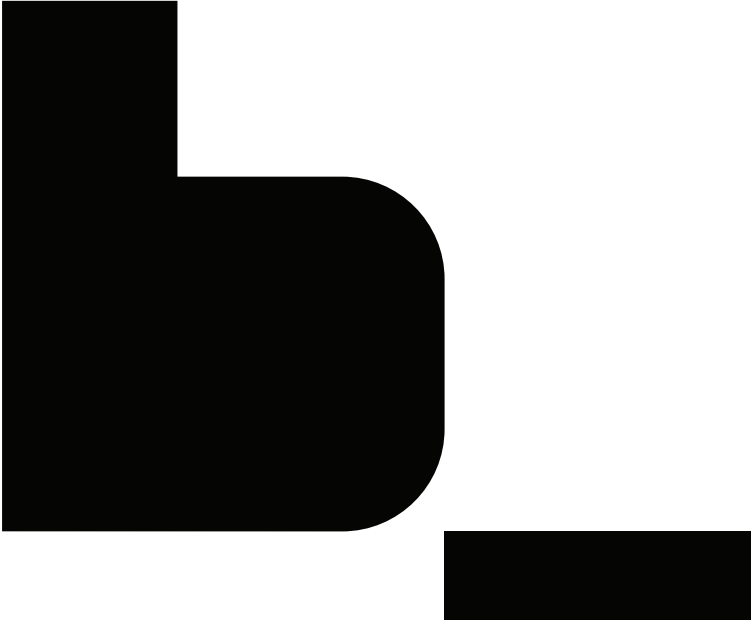




Генератор БЭМ- проектов на Yeoman

Евгений Гаврюшин, группа разработки БЭМ,
ВЕМир в Москве, 17 мая 2014

Обо мне

- › gavryushin@yandex-team.ru
- › github.com/eGavr
- › Младший разработчик в Яндексе
- › bit.ly/bemur-msk-gemini



bit.ly/bemup-msk-yo



github.com/bem/generator-bem-stub

Сборка ВЕМ–проектов

- › Указываем какие уровни переопределения есть
- › Указываем какие бандлы мы хотим собрать
- › Указываем с помощью каких технологий собирать эти бандлы

Конфиг для enb

```
module.exports = function(config) {  
  config.nodes('*.bundles/*', function(nodeConfig) {  
    nodeConfig.addTechs([  
      [ require('enb/techs/file-provider'), { target: '?bemjson.js' } ],  
      [ require('enb/techs/files') ],  
      [ require('enb/techs/deps') ],  
      [ require('enb/techs/bemdecl-from-bemjson') ],  
      [ require('enb-bemxjst/techs/bemtree-old') ],  
      [ require('enb-diverse-js/techs/node-js') ],  
      [ require('enb-diverse-js/techs/browser-js'), { target: '?js' } ],  
      [ require('enb-roole/techs/css-roole'), { target: '?noprefix.css' } ],  
      [ require('enb-bemxjst/techs/bemhtml-old') ],  
      [ require('enb/techs/html-from-bemjson') ]  
    ]);  
  
    nodeConfig.addTarget([  
      '?min.css',  
      '?bemtree.js',  
      '?node.js',  
      '?min.js',  
      '?bemhtml.js',  
      '?html'  
    ]);  
  });  
};
```

Конфиг для enb

```
config.nodes('touch-pad.bundles/*', function(nodeConfig) {
  nodeConfig.addTechs([
    [ require('enb/techs/levels'), { levels: getTouchPads(config) } ],
    [ require('enb-autoprefixer/techs/css-autoprefixer'), {
      browserSupport: [ 'android 4', 'ios 5' ],
      sourceTarget: '?.noprefix.css'
    }
  ]);
});

config.nodes('touch-phone.bundles/*', function(nodeConfig) {
  nodeConfig.addTechs([
    [ require('enb/techs/levels'), { levels: getTouchPhones(config) } ],
    [ require('enb-autoprefixer/techs/css-autoprefixer'), {
      browserSupport: [ 'android 4', 'ios 6', 'ie 10' ],
      sourceTarget: '?.noprefix.css'
    }
  ]);
});
```

Конфиг для enb

```
config.mode('development', function(modeConfig) {
  config.nodes('*.bundles/*', function(nodeConfig) {
    nodeConfig.addTechs([
      [ require('enb/techs/file-copy'), { sourceTarget: '?.css', destTarget: '?min.css' } ],
      [ require('enb/techs/file-copy'), { sourceTarget: '?js', destTarget: '?min.js' } ]
    ]);
  });
});

config.mode('production', function(modeConfig) {
  config.nodes('*.bundles/*', function(nodeConfig) {
    nodeConfig.addTechs([
      [ require('enb/techs/borschik'), { sourceTarget: '?css', destTarget: '?min.css' } ],
      [ require('enb/techs/borschik'), { sourceTarget: '?js', destTarget: '?min.js' } ]
    ]);
  });
});

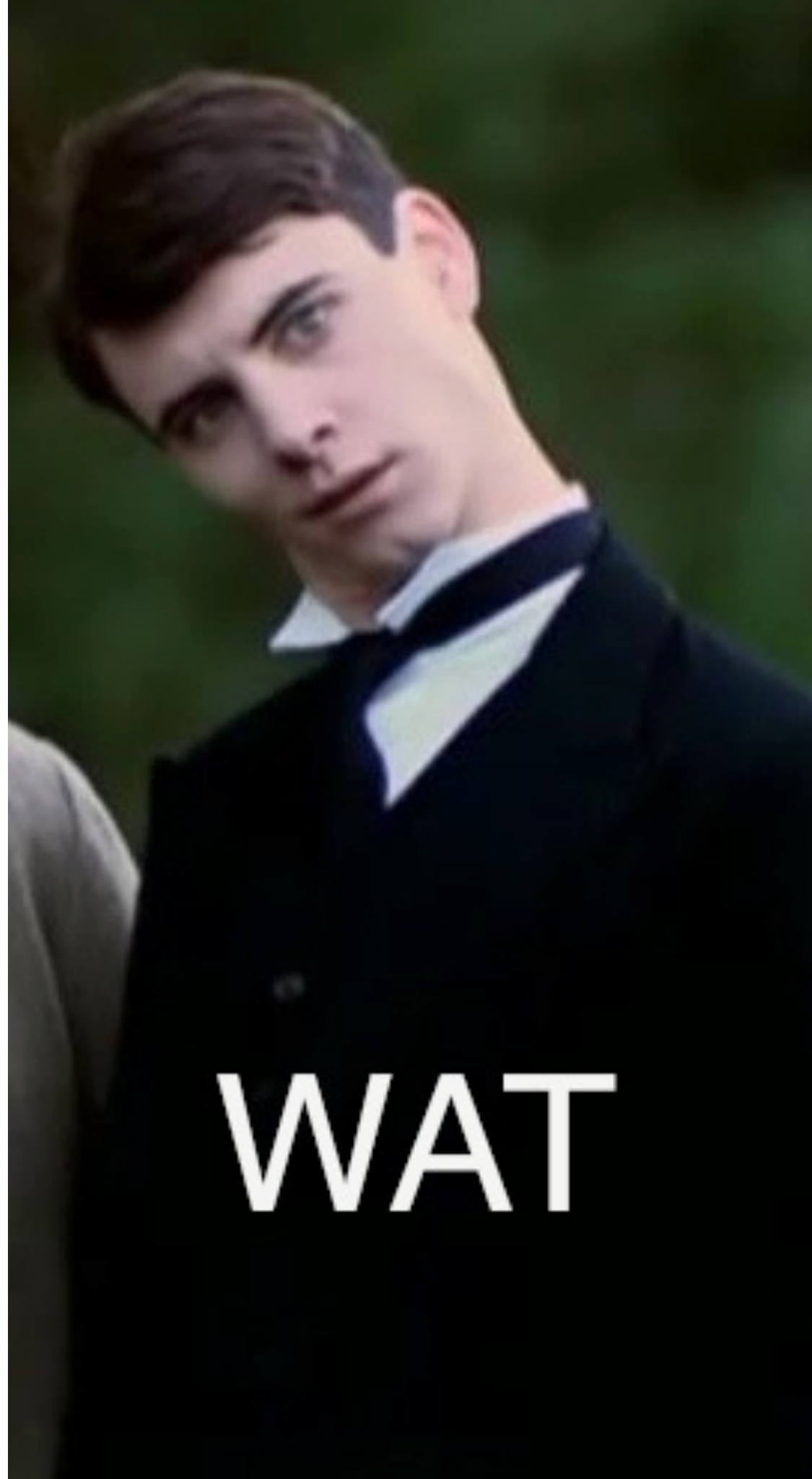
};
```

Конфиг для enb

```
function getTouchPads(config) {  
  return [  
    { path: 'libs/bem-core/common.blocks', check: false },  
    { path: 'libs/bem-core/touch.blocks', check: false },  
    { path: 'libs/bem-components/common.blocks', check: false },  
    { path: 'libs/bem-components/design/common.blocks', check: false },  
    { path: 'libs/bem-components/touch.blocks', check: false },  
    { path: 'libs/bem-components/design/touch.blocks', check: false },  
    { path: 'libs/bem-components/design/touch-pad.blocks', check: false },  
    'common.blocks',  
    'touch.blocks',  
    'touch-pad.blocks'  
  ].map(function(level) {  
    return config.resolvePath(level);  
  });  
}
```

Конфиг для enb

```
function getTouchPhones(config) {  
  return [  
    { path: 'libs/bem-core/common.blocks', check: false },  
    { path: 'libs/bem-core/touch.blocks', check: false },  
    { path: 'libs/bem-components/common.blocks', check: false },  
    { path: 'libs/bem-components/design/common.blocks', check: false },  
    { path: 'libs/bem-components/touch.blocks', check: false },  
    { path: 'libs/bem-components/design/touch.blocks', check: false },  
    { path: 'libs/bem-components/design/touch-phone.blocks', check: false },  
    'common.blocks',  
    'touch.blocks',  
    'touch-phone.blocks'  
  ].map(function(level) {  
    return config.resolvePath(level);  
  });  
}
```



WAT

Конфиг для bem-tools

```
/* global MAKE:false */

// process.env.YENV = 'production';

var PATH = require('path');

require('bem-tools-autoprefixer').extendMake(MAKE);

MAKE.decl('Arch', {

  blocksLevelsRegexp : /^.+?\\.blocks/,
  bundlesLevelsRegexp : /^.+?\\.bundles$/

});

MAKE.decl('BundleNode', {

  getTechs: function() {

    return [
      'bemjson.js',
      'bemdecl.js',
      'deps.js',
      'roole',
      'css',
      'bemt看ree',
      'bemhtml',
      'node.js',
      'browser.js+bemhtml',
      'html'
    ];

  },

}
```

Конфиг для bem-tools

```
getLevelsMap : function() {  
  return {  
    'touch-pad': [  
      'libs/bem-core/common.blocks',  
      'libs/bem-core/touch.blocks',  
      'libs/bem-core/touch-pad.blocks',  
      'libs/bem-components/common.blocks',  
      'libs/bem-components/design/common.blocks',  
      'libs/bem-components/touch.blocks',  
      'libs/bem-components/design/touch.blocks',  
      'libs/bem-components/touch-pad.blocks',  
      'libs/bem-components/design/touch-pad.blocks',  
      'common.blocks',  
      'touch.blocks',  
      'touch-pad.blocks'  
    ],  
    'touch-phone': [  
      'libs/bem-core/common.blocks',  
      'libs/bem-core/touch.blocks',  
      'libs/bem-core/touch-phone.blocks',  
      'libs/bem-components/common.blocks',  
      'libs/bem-components/design/common.blocks',  
      'libs/bem-components/touch.blocks',  
      'libs/bem-components/design/touch.blocks',  
      'libs/bem-components/touch-phone.blocks',  
      'libs/bem-components/design/touch-phone.blocks',  
      'common.blocks',  
      'touch.blocks',  
      'touch-phone.blocks'  
    ]  
  };  
},
```

Конфиг для bem-tools

```
getLevels : function() {
  var resolve = PATH.resolve.bind(PATH, this.root),
      buildLevel = this.getLevelPath().split('.')[0],
      levels = this.getLevelsMap()[buildLevel] || [];

  return levels
    .map(function(path) { return resolve(path); })
    .concat(resolve(PATH.dirname(this.getNodePrefix()), 'blocks'));
},

'create-css-node' : function(tech, bundleNode, magicNode) {
  var source = this.getBundlePath('roole');
  if(this.ctx.arch.hasNode(source)) {
    return this.createAutoprefixerNode(tech, this.ctx.arch.getNode(source), bundleNode, magicNode);
  }
}

});
```

Конфиг для bem-tools

```
var PATH = require('path'),
    environ = require('bem-environ'),
    getTechResolver = environ.getTechResolver,

    PRJ_ROOT = environ.PRJ_ROOT,
    PRJ_TECHS = PATH.resolve(PRJ_ROOT, '.bem/techs'),
    BEMCORE_TECHS = environ.getLibPath('bem-core', '.bem/techs');

exports.getTechs = function() {
  var techs = {
    'bemdecl.js'      : 'v2/bemdecl.js',
    'deps.js'         : 'v2/deps.js',
    'roole'           : 'v2/roole',
    'css'             : 'v2/css',
    'js'              : 'v2/js-i'
  };

  // use techs from project (.bem/techs)
  ['bemjson.js'].forEach(getTechResolver(techs, PRJ_TECHS));

  // use techs from bem-core library
  ['bemtree', 'bemhtml', 'node.js', 'vanilla.js', 'browser.js'].forEach(getTechResolver(techs, BEMCORE_TECHS));

  return techs;
};

exports.defaultTechs = ['roole', 'bemhtml', 'browser.js'];
```

Конфиг для bem-tools

```
var PATH = require('path'),
    environ = require('bem-environ'),
    getTechResolver = environ.getTechResolver,

    BEMCORE_TECHS = environ.getLibPath('bem-core', '.bem/techs');

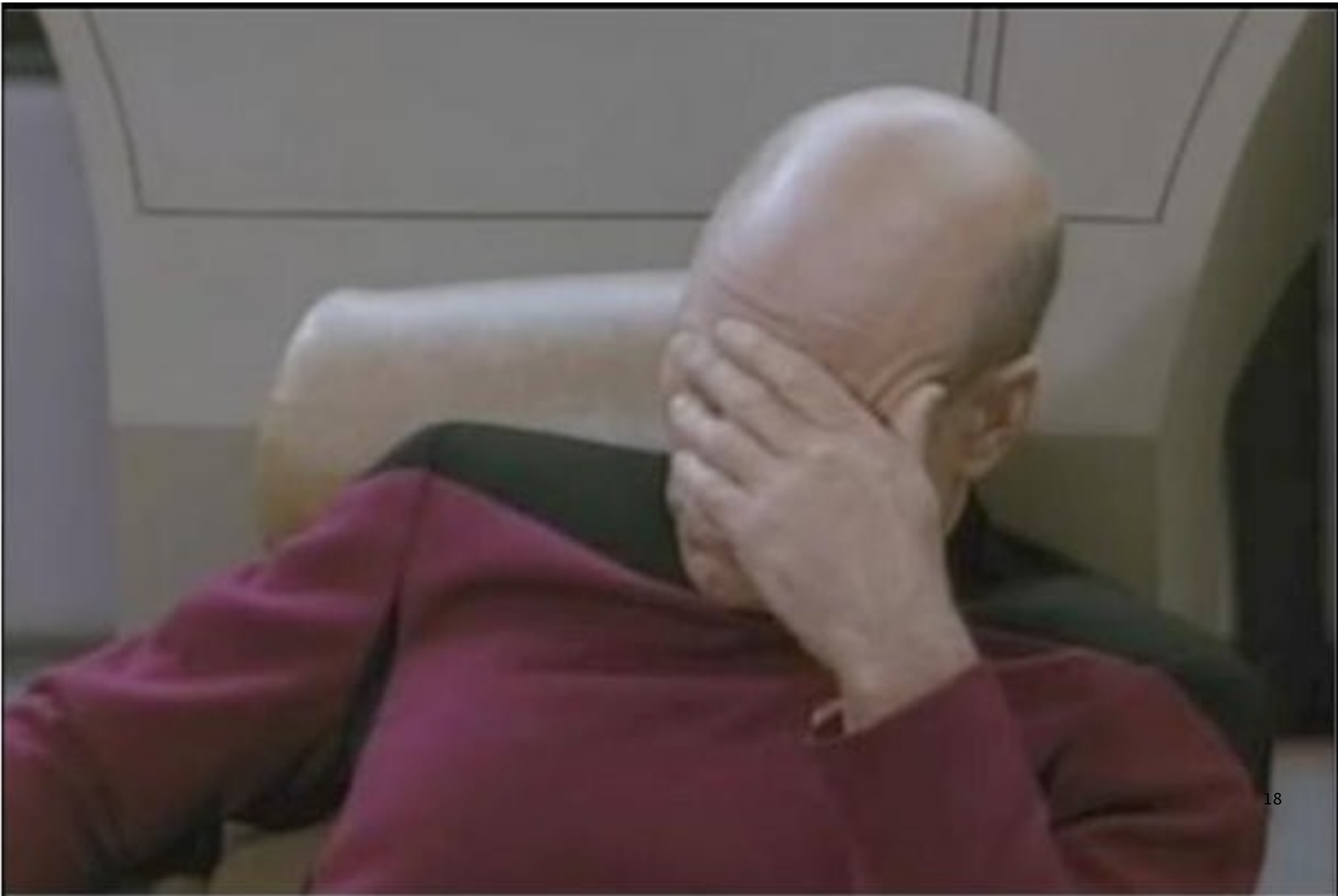
exports.baseLevelPath = require.resolve('./blocks');

exports.getTechs = function() {
    var techs = this.__base();

    // Use techs from lib bem-core
    ['browser.js+bemhtml', 'html'].forEach(getTechResolver(techs, BEMCORE_TECHS));

    return techs;
};

// Create bundles in bemjson.js tech
exports.defaultTechs = ['bemjson.js'];
```



Что мы захотели сделать?

- › Уменьшить время начала разработки нового проекта
- › Конфигурировать сборку без погружения в API

generator-bem-stub



yeoman.io



- › `npm install -g yo`
- › `npm install -g generator-generator`
- › `yo generator`
- › Отвечаем на вопросы
- › Получаем базу генератора

yeoman.io

Давайте что-нибудь
сгенерим!

Евгений Гаврюшин

Яндекс



eGavr

gavryushin@yandex-team.ru

Ваш звонок очень важен для нас:

<http://bit.ly/msk-bemup-feedback>