

Webpack

Новая жизнь angular приложения

Василий Малыгин

QuantumSoft

Angular

Также известен, как:

- Супер-героический фреймворк для веб-приложений
- HTML enhanced for web apps
- JavaScript MVW фреймворк

Основные фичи

- Шаблонизатор
- Система управления зависимостями

Зависимости: DI

```
1
2
3 angular.module('ModuleName').service('CoolService', function () {
4
5     return {
6         pureFunction: pureFunction
7     };
8
9     function pureFunction(param) {
10         return param ? true : false;
11     }
12
13 });
14
```

```
1
2
3 angular.module('ModuleName').controller('ControllerName', function (CoolService) {
4
5     console.log(CoolService.pureFunction())
6
7 });
8
```

Зависимости: templateUrl

```
1
2
3 angular.module('ModuleName', []).config(function ($stateProvider) {
4
5     $stateProvider
6     .state('coolPage', {
7         url: '/cool',
8         controller: 'CoolController',
9         templateUrl: 'views/cool.html'
10    });
11
12 });
```

Проблемы

Ангуляр

1. Vendor lock
2. Опечатки в названии сервисов
3. Тестирование + templateUrl

Билд

1. Огромный index.html
2. Разбиение по типу файла
3. Обработка темплейтов

Webpack

- Сборщик, который умеет все.
- Grunt нового поколения.
- Development-server.
- Самый простой способ для добавления ES6 в Ваш стек.

A1: vendor lock (до)

```
1
2
3 angular.module('ModuleName').service('CoolService', function () {
4
5     return {
6         pureFunction: pureFunction
7     };
8
9     function pureFunction(param) {
10         return param ? true : false;
11     }
12
13 });
```

```
1
2
3 angular.module('ModuleName').controller('CoolController', function (CoolService) {
4
5     console.log(CoolService.pureFunction())
6
7 });
```

A1: vendor lock (после)

```
2  
3 export default {  
4   pureFunction: pureFunction  
5 };  
6  
7 function pureFunction(param) {  
8   return param ? true : false;  
9 }  
10
```

```
1 import CoolService from './cool-service.js';  
2  
3 angular.module('ModuleName').controller('CoolController', function () {  
4  
5   console.log(CoolService.pureFunction())  
6  
7 });  
8
```


A2: очепятки (до)

```
1  import CoolService from './cool-service.js';
2
3  angular.module('ModuleName').controller('CoolController', function () {
4
5      console.log(CoolService.pureFunction())
6
7  });
```

```
1
2
3  angular.module('ModuleName', []).config(function ($stateProvider) {
4
5      $stateProvider
6
7          .state('coolPage', {
8              url: '/cool',
9              controller: 'CoolController',
10             templateUrl: 'views/cool.html'
11         });
12  });
```

A2: очепятки (после)

```
1
2 import CoolService from './cool-service.js';
3
4 export default function CoolController() {
5
6     console.log(CoolService.pureFunction())
7
8 };
9
```

```
1
2 angular.module('ModuleName', []).config(function ($stateProvider) {
3
4     $stateProvider
5     .state('coolPage', {
6         url: '/cool',
7         controller: require('pages/cool.controller'),
8         template: require('pages/cool.html')
9     });
10
11 });
12
```

A3: templateUrl

```
1
2 describe('ValidationService.validateDataImport', function () {
3
4     beforeEach(module('ModuleName'));
5
6     var ValidationService;
7
8     beforeEach(inject(function(_ValidationService_, _$httpBackend_) {
9         ValidationService = _ValidationService_;
10
11
12         _$httpBackend_.whenGET('/api/v1/something').respond({});
13     }));
14
15     it('should test some important thing', function () {
16         //...
17     });
18
19 });
20
```

```
at /home/zzz/ws/supplyshift/client-angular/app/external/angular-mocks/angular-mocks.js
at /home/zzz/ws/supplyshift/client-angular/test/unit/spec/services/SupportService/SupportService.spec.js:10:13
PhantomJS 1.9.8 (Linux 0.0.0) ValidationService.validateDataImport should hydrate results FAILED
Error: Unexpected request: GET views/splash.html
No more request expected
```

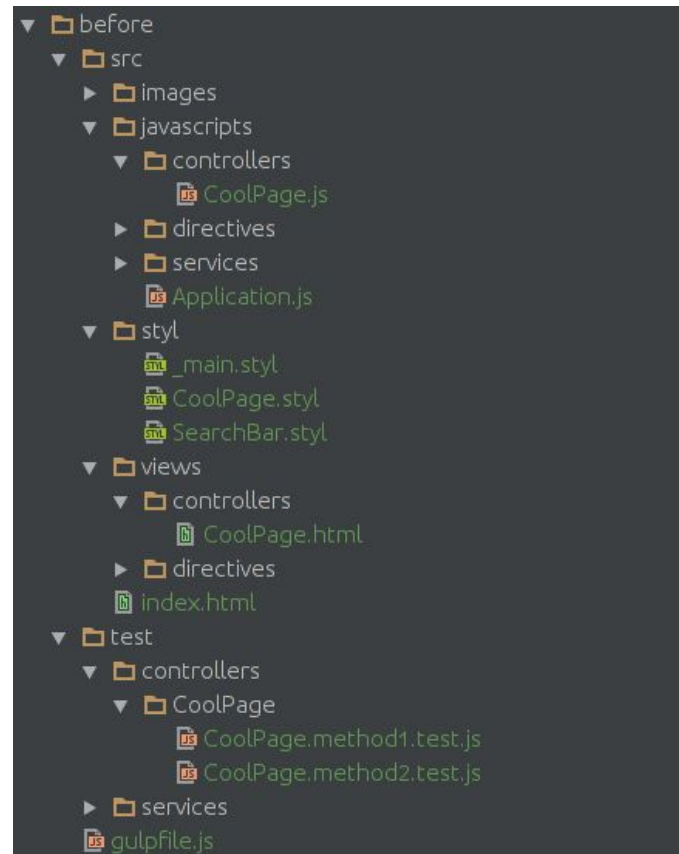
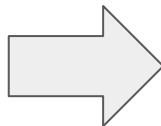
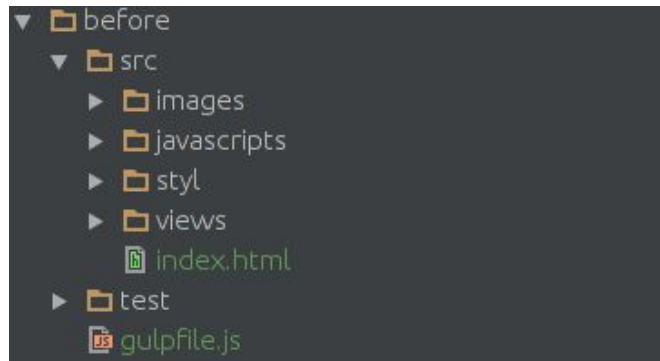
Б1: index.html (до)

```
1  <!doctype html>
2  <html id="ng-app" >
3  <head>
4      <meta charset="utf-8">
5      <title>Title</title>
6      <link href="images/favicon.ico?v=1" rel=icon type="image/png">
7      <!-- build:css css/main.css -->
8      <link rel="stylesheet" href="css/main.css">
9      <!-- endbuild -->
10 </head>
11 <body>
12     <div class="wrapper" ui-view></div>
13     <!-- build:vendorJs javascripts/net.supplyshift.externals.js -->
14     <!-- ... -->
41     <!-- endbuild -->
42
43     <!-- build:js javascripts/net.supplyshift.modules.js -->
44     <!-- ... -->
324     <!-- endbuild -->
325
326     <!-- build:js javascripts/net.supplyshift.templates.js -->
327     <!-- inject:partials -->
328     <!-- endinject -->
329     <!-- endbuild -->
330 </body>
331 </html>
332
```

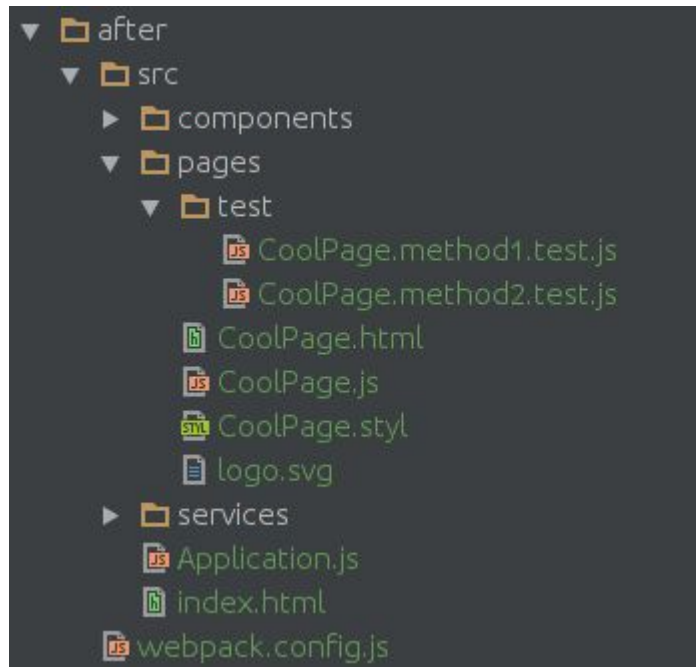
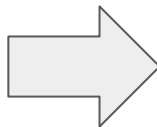
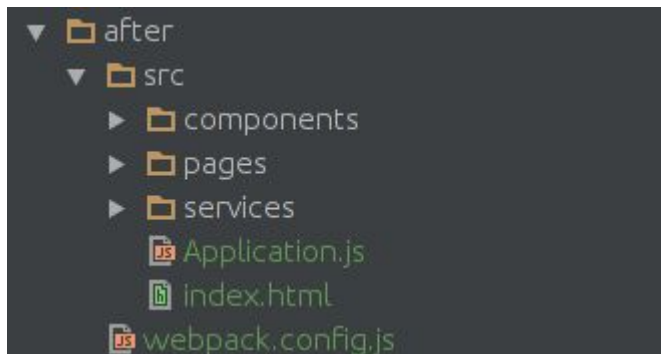
Б1: index.html (после)

```
1  <!doctype html>
2  <html id="ng-app">
3  <head>
4      <meta charset="utf-8">
5      <title>Title</title>
6      <link href="{%=o.htmlWebpackPlugin.files.favicon%}" rel="icon" type="image/png">
7
8      {% for (var css in o.htmlWebpackPlugin.files.css) { %}
9      <link href="{%=o.htmlWebpackPlugin.files.css[css] %}" rel="stylesheet">
10     {% } %}
11 </head>
12 <body>
13     <div class="wrapper" ui-view></div>
14     <script src="{%=o.htmlWebpackPlugin.files.chunks.vendor.entry %}"></script>
15     <script src="{%=o.htmlWebpackPlugin.files.chunks.bundle.entry %}"></script>
16 </body>
17 </html>
18
```

Б2: структура проекта (до)



Б2: структура проекта (после)



Б3: темплейты (до)

```
137 gulp.task('partials', ['minify-html'], function () {  
138   return gulp.src('.tmp/views/**/*.html')  
139     .pipe(ngHtml2Js({  
140       moduleName: 'ssTemplates',  
141       prefix: 'views/'  
142     })))  
143     .pipe(gulp.dest('.tmp/partials'));  
144 }
```

```
373  
374 <!-- build:js javascripts/net.supplyshift.templates.js -->  
375 <!-- inject:partials -->  
376 <!-- angular templates will be automatically converted in js and inserted here -->  
377 <!-- endinject -->  
378 <!-- endbuild -->  
379
```


БЗ: темплейты (после)

```
1
2 angular.module('ModuleName', []).config(function ($stateProvider) {
3
4     $stateProvider
5         .state('coolPage', {
6             url: '/cool',
7             controller: require('pages/cool.controller'),
8             template: require('pages/cool.html')
9         });
10
11 });
12
```

webpack.config.js

```
1  import webpack from 'webpack';
2  import HtmlWebpackPlugin from 'html-webpack-plugin';
3
4  export default {
5    context: __dirname + '/src',
6    entry: './Application.js',
7    output: {
8      path: __dirname + '/dist/',
9      filename: '[name].[hash].js',
10     publicPath: '/assets/'
11   },
12
13   module: {
14     loaders: [
15       { test: /\.js$/, loader: 'ng-annotate!babel' },
16       { test: /\.styl$/, loader: 'style!css!autoprefixer?browsers=last 2 version!styl' },
17       { test: /\.html$/, loader: 'html' },
18       { test: /\.svg$/, loader: 'url?limit=8192' }
19     ]
20   },
21
22   plugins: [
23     new HtmlWebpackPlugin({
24       template: 'src/index.html'
25     }),
26     new webpack.optimize.UglifyJsPlugin()
27   ]
28 }
```

Webpack - сборщик, который:

- + Позволяет делать код более модульным.
- + Позволяет меньше завязываться на фреймворк.
- + Позволяет снизить количество ошибок в runtime.
- + Позволяет писать на ES6!