

Web Standards Days

Прототип сайта: разработка и развёртывание

[Владимир Кузнецов](#), Engine6

**Прототип сайта:
разработка и
развёртывание**

Одно из правил в нашей компании

Показывать заказчику части проекта по мере готовности.

Задачи

- командная работа над прототипом;
- простая процедура развёртывания.

Сборка проекта

Хоть нам и нужно в конечном итоге получить статические HTML-, CSS- и JS-файлы, но «собирать» их можно из совершенно других форматов.

Сборка JavaScript

Отдельные компоненты «склеиваются» в один файл, который потом обрабатывается минификатором.

Сборка CSS

Файлы стилей генерируются с помощью препроцессоров
LESS, Stylus и т.п.

Сборка HTML

Файлы с разметкой генерируются шаблонизатором из отдельных страниц, повторно используемых фрагментов и раскладок.

**Наши первые
шаги**

Assemble

Генератор статических сайтов на Node.js

<http://assemble.io>

Есть [шаблон для Grunt](#) и [генератор для Yeoman](#).

Assemble

- запускается как Grunt-задача;
- использует шаблоны Handlebars (или любые другие);
- содержит набор готовых хелперов (легко написать свой);
- поддерживает раскладки любой вложенности;
- позволяет повторно использовать фрагменты страниц и выносить данные из разметки.

```
1 ---  
2 title: Useful helpers  
3 ---  
  
4 <article class="article">  
5   <h1 class="article__title">{{title}}</h1>  
6   <div class="article__body">  
7     <p>Here is the list of useful helpers. For more information please  
8       visit <a href="http://assemble.io/helpers/">Assemble documentation</a>  
9     </p>  
10    <h2 id="block-helpers">Block helpers</h2>  
11    <ul>  
12      <li><strong>#{each object}</strong> – iterate over  
13        specified object (private variables inside block: <code>@index</code>, <code>@key</code>)</li>  
14      <li><strong>#{if condition}</strong><br>  
15        <strong>#{unless condition}</strong><br>  
16        <strong>#{is arg1 arg2}</strong><br>  
17        <strong>{#isnt arg1 arg2}</strong><br>  
18        <strong>#{gt arg1 arg2}</strong><br>
```

```
1 ---
2 title: Useful helpers
3 ---
4
```

```
5 <article class="article">
```

```
6   <h1 class="article_title">{{title}}</h1>
```

```
7   <div class="article_body">
```

```
8     <p>Here is the list of useful helpers. For more information please
visit <a href="http://assemble.io/helpers/">Assemble documentation</a>
</p>
```

```
9     <h2 id="block-helpers">Block helpers</h2>
```

```
10    <ul>
```

```
11      <li><strong>#{each object}</strong> - iterate over
specified object (private variables inside block: <code>@index</code>,
<code>@key</code>)</li>
```

```
12      <li><strong>#{if condition}</strong><br>
```

```
13        <strong>#{unless condition}</strong><br>
```

```
14        <strong>#{is arg1 arg2}</strong><br>
```

```
15        <strong>#{isnt arg1 arg2}</strong><br>
```

```
16        <strong>#{gt arg1 arg2}</strong><br>
```

```
17        <strong>#{gte arg1 arg2}</strong><br>
```

```
18        <strong>#{lt arg1 arg2}</strong><br>
```

Мета-данные страницы

YAML front matter

- layout
- published
- category, categories и tags

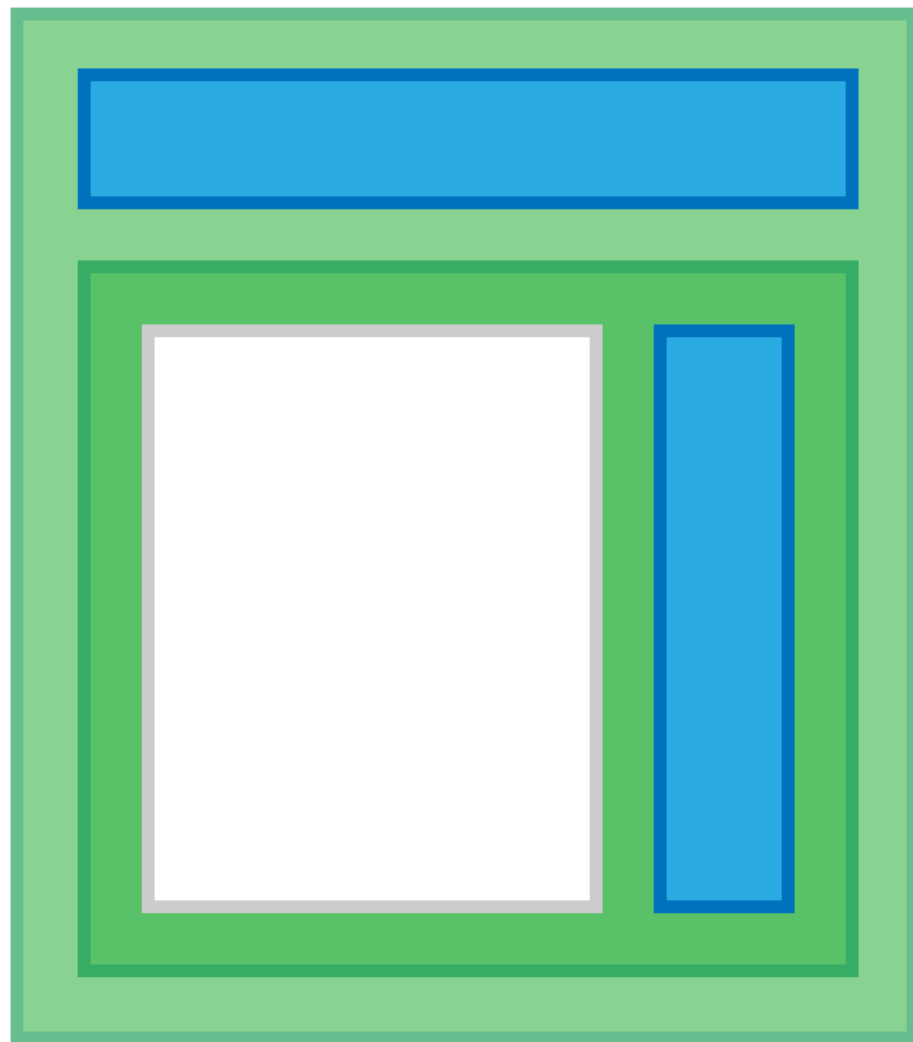
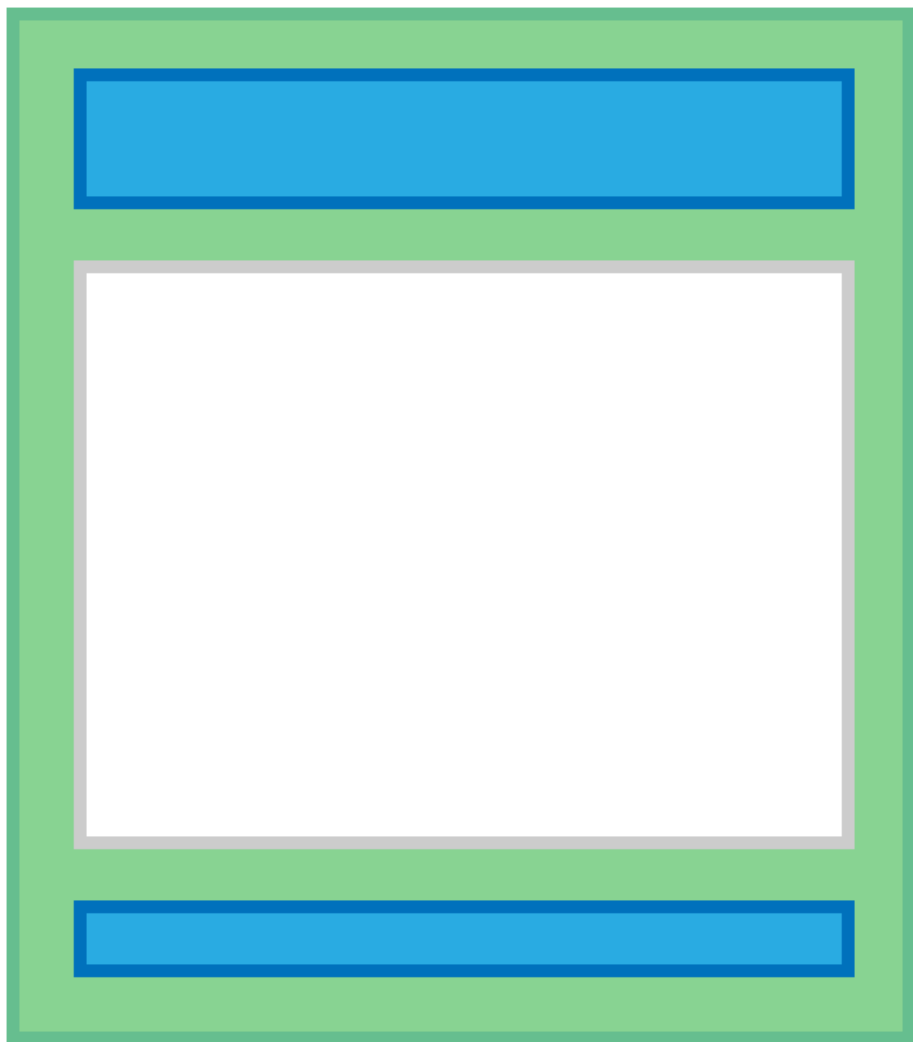
```
1 ---
2 title: Page number three
3 layout: two-columns.hbs
4 products:
5 - { id: 1, name: "Lorem ipsum dolor sit amet, consectetur adipiscing elit",
6   type: "seminar", start: "10:00", end: "10:30" }
7 - { id: 2, name: "Fusce dignissim eleifend quam, aliquam hendrerit nisi",
8   type: "seminar", start: "10:30", end: "11:10" }
9 - { id: 3, name: "Praesent quis felis imperdiet sapien molestie cursus.",
10  type: "seminar", start: "11:10", end: "12:00" }
11 - { id: 4, name: "Ut vel sapien tellus. Sed in dictum lacus.",
12  type: "masterclass", start: "13:00", end: "13:45" }
13 - { id: 5, name: "Cum sociis natoque penatibus et magnis dis parturient",
14  type: "masterclass", start: "13:45", end: "14:30" }
15 - { id: 6, name: "Curabitur porttitor, arcu ac viverra aliquet, nisi quam",
16  type: "masterclass", start: "14:30", end: "15:00" }
17 ---
18
19 <article class="article">
20   <h1 class="article__title">{{title}}</h1>
21   <div class="article__body">
22     <p>
```

```
1 ---
2 title: Page number three
3 layout: two-columns.hbs
4 products:
5 - { id: 1, name: "Lorem ipsum dolor sit amet, consectetur adipiscing elit",
6   | type: "seminar", start: "10:00", end: "10:30" }
7 - { id: 2, name: "Fusce dignissim eleifend quam, aliquam hendrerit nisi",
8   | type: "seminar", start: "10:30", end: "11:10" }
9 - { id: 3, name: "Praesent quis felis imperdiet sapien molestie cursus.",
10  | type: "seminar", start: "11:10", end: "12:00" }
11 - { id: 4, name: "Ut vel sapien tellus. Sed in dictum lacus.",
12  | type: "masterclass", start: "13:00", end: "13:45" }
13 - { id: 5, name: "Cum sociis natoque penatibus et magnis dis parturient",
14  | type: "masterclass", start: "13:45", end: "14:30" }
15 - { id: 6, name: "Curabitur porttitor, arcu ac viverra aliquet, nisi qu",
16  | type: "masterclass", start: "14:30", end: "15:00" }
17 ---
18
19 <article class="article">
20   <h1 class="article__title">{{title}}</h1>
21   <div class="article__body">
22     <n>
```

Мета-данные страницы

Шаблон страницы получает данные из:

- YAML front matter
- Gruntfile.js
- **внешних JSON- или YAML-файлов**



```
1 <!doctype html>
2 <html lang="{{default siteConfig.lang 'en'}}" class="no-js">
3 <head>
4     <meta charset="utf-8">
5     <title>{{default title basename}} :: {{siteConfig.title}}</title>
6     <link rel="stylesheet" href="/assets/styles/all.css?v={{now "%d%H%M%S"}}">
7 </head>
8 <body class="page page_{{lowercase basename}}">
9     <!-- header -->
10    {{> header}}
11    <!-- /header -->
12
13    <!-- page content -->
14    {{> body}}
15    <!-- /page content -->
16
17 <div class="page__bottom">
18     <!-- footer -->
19     {{> footer}}
20     <!-- /footer -->
21 </div>
22
```

```
1 <!doctype html>
2 <html lang="{{default siteConfig.lang 'en'}}" class="no-js">
3 <head>
4     <meta charset="utf-8">
5     <title>{{default title basename}} :: {{siteConfig.title}}</title>
6     <link rel="stylesheet" href="/assets/styles/all.css?v={{now "%d%H%M%S"}}">
7 </head>
8 <body class="page page_{{lowercase basename}}">
9     <!-- header -->
10    {{> header}}
11    <!-- /header -->
12
13    <!-- page content -->
14    {{> body}}
15    <!-- /page content -->
16
17 <div class="page__bottom">
18     <!-- footer -->
19     {{> footer}}
20     <!-- /footer -->
21 </div>
22
```

```
1 ---
2 layout: setting.hbs
3 ---
4
5 <div class="page__content">
6   <div class="page__main-column">
7     {{> body}}
8   </div>
9   <div class="page__sidebar">
10    {{> sidebar}}
11  </div>
12 </div>
```

```
1 ---
2 layout: setting.hbs
3 ---
4
5 <div class="page__content">
6   <div class="page__main-column">
7     {{> body}}
8   </div>
9   <div class="page__sidebar">
10     {{> sidebar}}
11   </div>
12 </div>
```

Встроенные переменные

- `basename` или `pagename`
- `filename`
- `dirname`
- `ext`

Helpers

Блочные: `{{#each}}`, `{{#if}}` и т.п.

01. `{{#is basename "index"}}`

02. `<span>Site name</span>`

03. `{{else}}`

04. `<a href="/">Site name</a>`

05. `{{/is}}`

Helpers

Функции: `{{default}}`, `{{formatDate}}` и т.п.

01. `<title>{{default title basename}}</title>`

02. `<script src="/all.js?v={{now "%d%H%M%S"}}"></script>`

Helpers

```
01. module.exports.register =  
02.   function (Handlebars, options) {  
03.     Handlebars.registerHelper("foo", function (str) {  
04.       return "---=== " + str + " ===---";  
05.     });  
06.   };
```

Partials

Вынесем в отдельные файлы повторяющиеся фрагменты кода.

```
7 </head>
8 <body class="page page_{{lowercase basena
9     <!-- header -->
10     {{> header}}
11     <!-- /header -->
12
13     <!-- page content -->
14     {{> body}}
15     <!-- /page content -->
16
```

Partials

```
{{> sidebar}}
```

ВЫЗОВ В КОНТЕКСТЕ СТРАНИЦЫ

```
{{> sidebar context1}}
```

ВЫЗОВ В КОНТЕКСТЕ ОБЪЕКТА **context1**

```
{{#context2}} {{> sidebar}} {{/context2}}
```

ВЫЗОВ В КОНТЕКСТЕ ОБЪЕКТА **context2**

Запуск задач с помощью Grunt

\$ `grunt`

одноразовый запуск сборки всего проекта

\$ `grunt watch`

запуск сборки и отслеживание изменения в файлах

\$ `grunt server`

запуск HTTP-сервера и отслеживание изменений с livereload

Пример Gruntfile.js

<https://github.com/doneright/assemble>

- сборка HTML из шаблонов;
- компиляция LESS-файлов;
- склеивание и минификация скриптов;
- watcher, сервер и livereload.

Git

Git

- Сторонние сервисы ([GitHub](#), [CodeBase](#) и др.)
- Свой Git-сервер (Gitosis)

**Git работает
без сервера!**

Автоматическое развёртывание

Если **GitHub** и VPS, то настраиваем «WebHook»

Available Service Hooks

WebHook URLs (1) ✓

ActiveCollab

Acunote

AgileBench

AgileZen

AmazonSNS

Aplary

Apolo

AppHarbor

Apropos

WebHook URLs

URL (remove)

http://my-awesome-project.com:8000/hook/

URL (remove)

[Add another webhook URL](#)

Test Hook

Update settings

```
1 var express = require('express');
2 var app = express();
3
4 app.use(express.urlencoded());
5 app.use(express.json());
6
7 app.post('/hook/', function (req, res) {
8     var payload = JSON.parse(req.param('payload'));
9     console.log('=====');
10    console.log('Repository: %s', payload.repository.name);
11    console.log('Branch: %s', payload.ref);
12    console.log('=====\\n');
13
14    var body = 'OK\\n';
15    res.setHeader('Content-Type', 'text/plain');
16    res.setHeader('Content-Length', body.length);
17    res.end(body);
18 });
19
20 app.listen(8000);
21
22
```



hook-server — node — 80×11



```
slim:hook-server mista_k$ node index.js
```



hook-tester — bash — 80×9



```
slim:hook-tester mista_k$ echo "*" >> 1.txt && git commit -am 'update' && git push origin master
```

Автоматическое развёртывание

Если **собственный сервер**,
то настраиваем `post-receive` или `update` хук.

Автоматическое развёртывание

Если к хостингу есть **доступ только по FTP**,
то выручит скрипт [git-ftp](#).

Файлы добавляются, удаляются и обновляются при условии,
что они находятся под контролем версий.

Демо



ftp-upload — bash — 80x22



```
slim:ftp-upload mista_k$
```



ftp-upload — bash — 80x22



```
slim:ftp-upload mista_k$
```




ftp-upload — bash — 80x22



```
slim:ftp-upload mista_k$
```



ftp-upload — bash — 80x22



```
slim:ftp-upload mista_k$
```

Заключение

- Узнали как собрать статический сайт из страниц, повторно используемых фрагментов и раскладок.
- Узнали как быстро и просто выложить собранный сайт на сервер.

Слайды презентации: bit.ly/wsd-2013-ci

Спасибо за внимание

Владимир Кузнецов

Twitter: [@mista_k](https://twitter.com/mista_k)

Блог: noteskeeper.ru

Слайды презентации: bit.ly/wsd-2013-ci