

Автоматизация: как делать полноценный сайт без участия бэкендера

История о том, как ленивые программисты делают свою работу

Часть 1.

**Предыстория и знакомство
с героями**

Front-End Developer

Василий



php

MySQL

Frameworks

Server



Back-End Developer

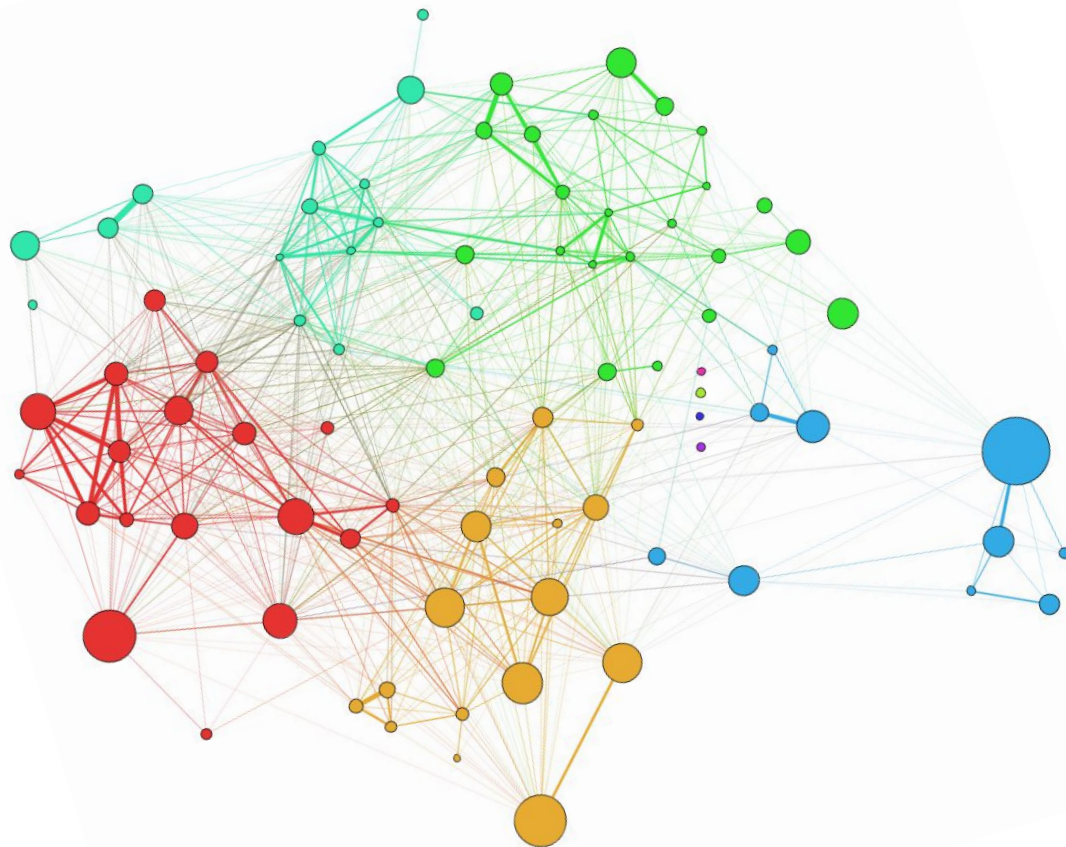
Петр



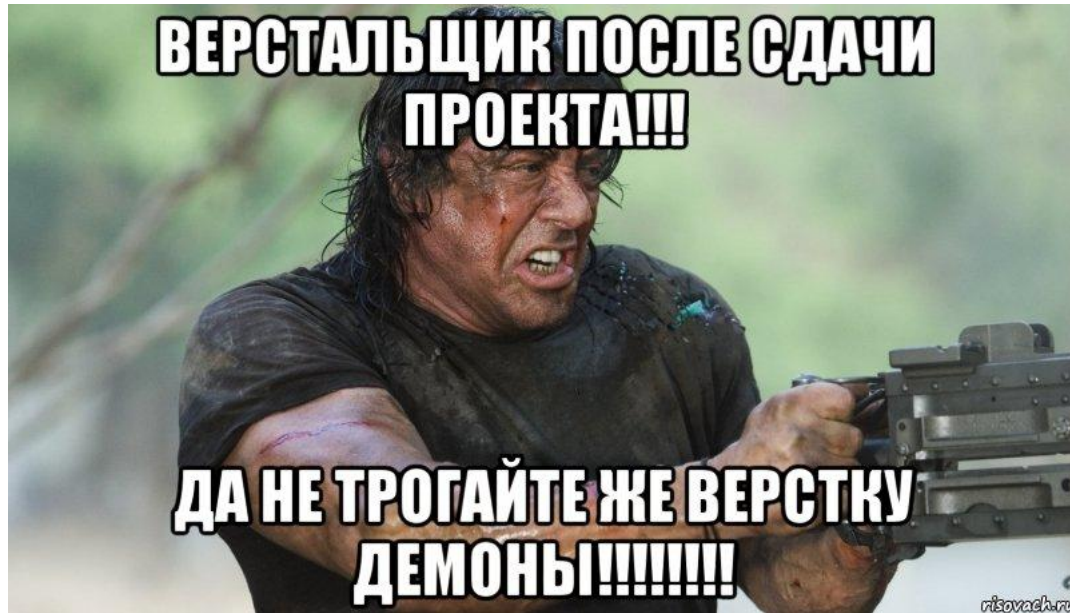
Стандартный проект: разработка сайта



В реальной жизни



Боль и слезы всех верстальщиков





Что может Вася

Автоматизация!

- Стандартная методология наименования элементов
- Препроцессор для CSS
- Шаблонизатор для HTML
- Сборщик проектов для того, чтобы это собирать



Что Вася выбрал

1. **Gulp** как сборщик проектов
2. **БЭМ** как методология наименования элементов
3. **Stylus** как препроцессор для CSS
4. **Jade** - шаблонизатор HTML

Сборщик проектов



Методология именования элементов



Препроцессор CSS



stylus

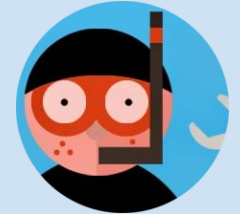
Шаблонизатор HTML





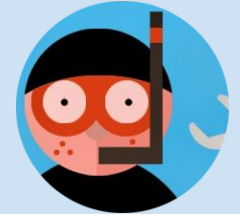
Выбор технологий для бэкенда

- Независимая структура каталогов для шаблонов
- Компонентный подход
- Несложный роутинг
- Удобный php-шаблонизатор



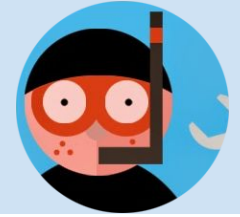
Система управления контентом **October CMS**

OCTOBER 

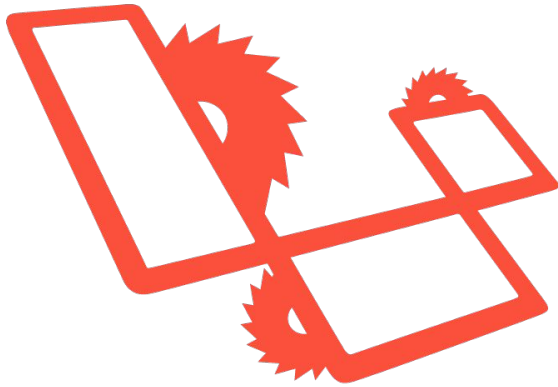


Система управления контентом **October CMS**





Шаблонизаторы в October CMS





Что может Петя

Автоматизация!

- Получать верстку с определенной структурой каталогов
- Автоматически:
 - Вырезать части кода и/или заменять их
 - Выносить части кода в отдельные файлы
 - Удалять "рыбу" из верстки



Что сделал Петя

- Берет верстку с типовой структурой каталогов (спасибо, Gulp!)
- Написал php-парсер, который:
 - подставляет директивы для php-шаблонизатора
 - вырезает код и вставляет его в отдельный файл
 - подставляет значения в атрибуты тегов
 - удаляет “рыбный” контент



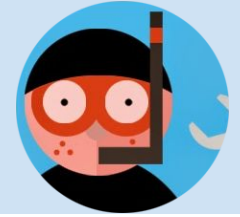
Как это работает

```
1 <head>
2   1 <!--bl:put @section('head')-->
3     <meta charset="utf-8"/>
4     <title>Title</title>
5     <link rel="stylesheet" data-bl-href="/common.css"
      href="../../common.css"/>
6     <!--bl:put @show-->
7 </head>
```



Как это работает

```
1 <head>
2   1 @section('head')
3     <meta charset="utf-8"/>
4     <title>Title</title>
5     <link rel="stylesheet" href="/common.css"/> 2
6     @show
7 </head>
8
```



Запуск скрипта из консоли

```
php artisan JtoB
```



Что же еще можно улучшить?



Автоматизация!

- Заставить Васю написать плагин для сборщика проектов

Часть 2.

Как Вася писал плагин для Gulp



Подготовка: базовые инструменты

- Node.js
- Gulp

```
npm install -g gulp
```


Краткий гайдлайн

- Плагин всегда принимает и отдает набор Vinyl объектов

Vinyl file object — просто файл. В себе хранит название файла и содержимое (Buffer или Stream)

- Плагин должен выполнять только 1 задачу
- Не писать плагин, который уже существует



Инициализация проекта

```
npm init
```



Подключение модулей

```
npm install --save through2 vinyl
```



Главный файл плагина - index.js

```
1 // through2 is a thin wrapper around node transform streams
2 var through = require('through2');
3 var File = require('vinyl');
4 var HtmlToBlade = require('plugin.js');
5
6 // Consts
7 const PLUGIN_NAME = 'gulp-html-to-blade';
8
9 // Plugin level function(dealing with files)
10 function gulpHtmlToBlade(options) {
11   var arSections = [];
12
13   return through.obj(function(file, enc, cb) {
14     if (file.isNull()) {
15       return cb(null, file);
16     }
17     if (file.isBuffer()) {
18       HtmlToBlade.scanTemplate(file, options, arSections,
19         function(template) {
20           file = new File({
21             path: template.filename ? template.filename + '.html' : file.path.replace(file.base, ''),
22             contents: new Buffer(template.content)
23           });
24           cb(null, file);
25         });
26     } else {
27       cb(null, file);
28     }
29   });
30 }
31
32 // Exporting the plugin main function
33 module.exports = gulpHtmlToBlade;
```



Зависимости и структура - index.js

```
1 // through2 is a thin wrapper around node transform streams
2 var through = require('through2');
3 var File = require('vinyl');
4 var HtmlToBlade = require('plugin.js');
5
6 // Consts
7 const PLUGIN_NAME = 'gulp-html-to-blade';
8
9 // Plugin level function(dealing with files)
10 function gulpHtmlToBlade(options) {
29 }
30
31 // Exporting the plugin main function
32 module.exports = gulpHtmlToBlade;
```



Использование through2 - index.js

```
10 function gulpHtmlToBlade(options) {  
11     var arSections = [];  
12     return through.obj(function(file, enc, cb) { 1  
13         if (file.isNull()) {  
14             return cb(null, file);  
15         }  
16         if (file.isBuffer()) {  
17             //do magic 2  
18             HtmlToBlade.scanTemplate(file, options, arSections,  
24                 function(template) {  
25             } else {  
26                 cb(null, file);  
27             }  
28         });  
29     }  
}
```



Создание Vinyl-файла - index.js

```
17 //do magic
18 HtmlToBlade.scanTemplate(file, options, arSections,
    function(template) {
19     file = new File({
20         path: template.filename,
21         contents: new Buffer(template.content)
22     });
23     cb(null, file);
24 });
```

Подключение в Gulp - gulpfile.js

```
1  'use strict';
2
3  var gulp = require('gulp');
4  var htmltoblade = require('../index');
5  var ext_replace = require('gulp-ext-replace');
6  var dest = 'dist';
7
8  gulp.task('htmltoblade', function() {
9      return gulp.src('./*.html')
10         .pipe(htmltoblade({
11             'parentName': 'main',
12             'clear': ['content'],
13             'dest': dest + '/',
14             'parentFolder': 'demo/'
15         }))
16         .pipe(ext_replace('.blade.php'))
17         .pipe(gulp.dest(dest));
18 });
```


Финальная структура папок



demo



node_modules



index.js

Тип: Файл "JS"



package.json

Тип: Файл "JSON"

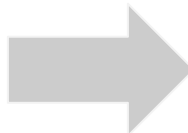


plugin.js

Тип: Файл "JS"

Результат

```
▼ public
  404.html
  500.html
  about.html
  index.html
  main.html
  map.html
  problem.html
  register-correct.html
  repair.html
  results.html
```



```
▼ dist
  ▼ errors
    404.blade.php
    500.blade.php
  ▼ partials
    ▼ form
      add_problem.blade.php
      add_problem_finish.blade.php
      add_recover_finish.blade.php
      attention.blade.php
      change_pass.blade.php
      froce-login.blade.php
      login.blade.php
      recovery.blade.php
      register.blade.php
      register_finish.blade.php
      result_problem.blade.php
    ▼ user
      create_verification.blade.php
      reset_password.blade.php
      about.blade.php
      index.blade.php
      main.blade.php
      map.blade.php
      problem.blade.php
      results.blade.php
```



История успеха



- Увеличилась скорость разработки
- Улучшилось качество кода
- Всегда актуальная верстка в php-шаблонах

Спасибо за внимание



Александра Шинкевич

[@neesoglasnaja](https://twitter.com/neesoglasnaja)

ash-web.by

Github проекта:

[lovata/gulp-html-to-blade](https://github.com/lovata/gulp-html-to-blade)

[Node.js](https://nodejs.org/)

[Gulp](https://gulpjs.com/)

[Официальный гайдлайн](#)