



Что такое Gulp?

06 DECEMBER 2017

Для **увеличения скорости загрузки сайта**, его **оптимизации** и **сокращения времени на разработку** требуется производить определенный набор действий. Например, минимизировать созданные css- и js-файлы, оптимизировать добавленные картинки, настраивать автоматическую синхронизацию редактора с браузером и др. В простейшем случае вы можете произвести вышеописанный набор действий самостоятельно, воспользовавшись онлайн-tools. Но также существует специальный инструмент сборки, который сделает все это за вас. Его имя – **GULP**.

С помощью добавляемых плагинов Gulp позволяет произвести все рутинные операции самостоятельно. Все, что требуется, – это указать верную последовательность действий в `gulpfile`.

Подключение Gulp

Конечно же, для начала нам потребуется Gulp установить. Делается это путем выполнения двух простых команд из консоли:

```
// сначала ставим gulp глобально для всего компьютера
npm install -g gulp

// переходим в нужную папку и инициализируем проект
npm init

// устанавливаем конкретно для нашего приложения
npm install --save-dev gulp
```

Начало работы

Предположим, что основной файл js-скрипта находится по адресу `js/main.js`. Предположим, что этот файл чрезмерно велик, и мы

горим идеей уменьшить его до приемлемых размеров. Помощь нам в этом окажет плагин `gulp-uglify`.

В первую очередь, устанавливаем плагин в проект:

```
npm install gulp-uglify --save-dev
```

Во вторую очередь, создаем в основной директории проекта `gulpfile.js` со следующим кодом:

```
// в этой секции подключаем все необходимые плагины
var gulp = require('gulp'),
    uglify = require('gulp-uglify');

// здесь вызываем необходимую задачу
gulp.task('scripts', function () {
    gulp.src('js/main.js')
        .pipe(uglify())
        .pipe(gulp.dest('build'));
});
```

В третью очередь, выполняем в консоли команду `gulp scripts`.

И – УРА: в папке `build` оказывается минимизированный файл.

Объяснение причин

Основная структурная единица *Gulp* – это **task**. Мы можем задавать любое количество задач, каждый из которых выполняет конкретную задачу. Мы можем объединять задачи друг в друга. Можем вызывать их в зависимости от входящих параметров.

Задача – это законченное действие. В приведенном выше случае мы говорим:

1. Возьми файл `js/main.js`.
2. Прими для него плагин `uglify`.
3. И положи в папку `build`.

Задача всегда имеет имя. У нас это – `scripts`.

Скользкие моменты

1. Если мы хотим включить внутрь главного таска выполнение последовательности других, то нужно придерживаться следующего синтаксиса:

```
// Таск называется build и включает в себя таски css и js
gulp.task('build', ['css', 'scripts']);
```

2. Все **таски выполняются асинхронно**, т.е. одновременно. Если мы этого не хотим, то можно определить последовательность действий:

```
// Сначала выполним все задачи из scripts, а затем только приступим к
gulp.task('css', ['scripts'], function() {
    ...
});
```

3. Фундаментальным является понимание **потоков**. Это означает, что мы можем передать данные из одной функции в другую в рамках одного таска. Записывается через `pipe`. Рассмотрим пример:

```
gulp.task('scripts', function() {
    return gulp.src('js/main.js')
        .pipe(uglify())
        .pipe(concat('app.js'))
        .pipe(gulp.dest('build'));
});
```

Итак, сначала мы применяем плагин `uglify`, затем для полученного результата вызываем плагин `concat`, а после складываем все получившееся в папку `build`. Т.е. действия выполняем последовательно, применяя в качестве входных данных результат вызова предыдущей функции.

Домашнее задание

Давайте потренируемся работать с **Gulp**. Выполните следующие шаги:

1. Начните работу с *Gulp*. Выполните пункты 1–2–3 из данной статьи.

2. Подключите плагины `gulp-concat`, `gulp-imagemin`, `gulp-svg2png`, `gulp-svgmin`, поймите, за что они отвечают.
3. Создайте три таска: первый - `scripts`, второй - `images`, третий - `svg`. В первом с помощью потоков запустите таски `uglify`, `concat`. Во втором - обработайте изображения с помощью таска `gulp-imagemin`. В третьем - в нужной последовательности через потоки подключите `gulp-svg2png`, `gulp-svgmin`.
4. Создайте команду `gulp default`, в которой будут фигурировать все созданный выше таски.
5. Запустите ее и проверьте полученный результат!

Комментарии Сообщество

 Политика конфиденциальности

 Войти ▾

 Рекомендовать

Лучшее в начале ▾

Начать обсуждение...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS 

Имя

@Afelua

Read [more posts](#) by this author.

 Moscow

Share this post