**Методические указания по теме “Gulp”**



1. [**Gulp (http://gulpjs.com)**](#h.8151c8wi0f0g)
2. [**Gulp.src()**](#h.sl6o1wxjmbxa)
3. [**Gulp.dest()**](#h.r4folwur1yoj)
4. [**Gulp.task()**](#h.cmr55690pupb)
5. [**Gulp.watch()**](#h.yt8impqfpii3)
6. [**Gulp-concat-css**](#h.imn4z1x1m9mg)
7. [**Gulp-minify-css**](#h.wvaswsh4jpdw)
8. [**Gulp-rename**](#h.ihgtmxwzj4cl)
9. [**Gulp-uglify**](#h.erpuygi5x5dt)
10. [**Gulp-autoprefixer**](#h.g41u3imew9rh)
11. [**Gulp-jade**](#h.9kmfioa9iytw)
12. [**Gulp-sass**](#h.dojh5vdn9dsq)
13. [**Gulp-uncss**](#h.snrz19bdils9)
14. [**Gulp-imagemin**](#h.ts8lstuqij89)
15. [**Del**](#h.e1rdutqwfppf)
16. [**Gulp-util**](#h.1a51pd64l5mw)
17. [**Vinyl-ftp**](#h.mreodom3fs09)
18. [**Browser-sync**](#h.35xiij8phqpo)
19. [**Gulp-useref**](#h.711o6euo7e24)
20. [**Wiredep**](#h.f0dws364z1am)
21. [**Gulp Blacklist**](#h.ln3zz62oi63y)

# Gulp (<http://gulpjs.com>)

***Gulp***- потоковый сборщик проектов на javascript. Используется для автоматизации самых разных задач.

|  |
| --- |
| * *Для работы необходим* [***nodejs***](https://nodejs.org/) *и* [***npm***](https://www.npmjs.com/) *(****npm*** *устанавливается вместе с* ***nodejs****).* * [***Оффициальный сайт***](http://gulpjs.com/) * [***Поиск плагинов***](http://gulpjs.com/plugins/) * [***Курс по gulp***](http://www.youtube.com/playlist?list=PLY4rE9dstrJwXCz1utct9b6Vub9VWQoKo) |

1) Проверка установлен ли Gulp:

|  |
| --- |
| **$** **gulp -v** |

2) Устанавливаем глобально

|  |
| --- |
| **$** **npm install --global gulp** |

*сокращенно - $ npm i -g gulp*

3) Создаем в корне проекта файл ***package.json***

|  |
| --- |
| **$** **npm init** |

Либо создать файл самостоятельно.

* **name** - название проекта
* **version** - версия проекта
* **description** - описание
* **...**

остальные поля [**ссылка**](https://docs.npmjs.com/files/package.json)

4) Устанавливаем в проект в devDependencies

|  |
| --- |
| **$** **npm install --save-dev gulp** |

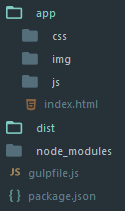
*сокращенно - $ npm i -D gulp*

5) Создаем в корне проекта файл ***gulpfile.js***

|  |
| --- |
| **var gulp = require('**gulp**');**  **gulp.task('**default**', function() {**  **// задача по умолчанию**  **});** |

6) Запускаем в консоли

|  |
| --- |
| **$** **gulp** |



**Структура:**

**app** - *рабочая директория проекта (исходники)*

**dist** - *директория сборки (генерируется сборщиком)*

**node\_modules** - *директория модулей, установленных через npm*

**package.json** - *файл с глобальными настройками проекта*

**gulpfile.js** - *файл для управления проектом*

# 

# 

# Работа с gulp глобально

После установки глобально

|  |
| --- |
| **$** **npm install --global gulp** |

Мы можешь использовать глобально установленный gulp локально

|  |
| --- |
| **$** **npm link gulp** |

|  |
| --- |
| При работе глобально необходимо следить за версиями, чтобы при обновлении версии не сломался проект |

# 

**GULP API**

# 

# 

# Gulp.src()

Возвращает поток, представляющий файлы, который уже может быть передан на вход плагинам ([**ссылка**](https://github.com/gulpjs/gulp/blob/master/docs/API.md#gulpsrcglobs-options))

**gulp.src(globs[, options])**

1. **globs** - *маска файлов (строка или массив)*
2. **options** - *объект настроек*

Пример использования при склеивании файлов:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **concatCss = require('**gulp-concat-css**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( concatCss('**fullstyle.css**') )**  **.pipe( gulp.dest('**app/css/**') );**  **});** |

Пример масок файлов:

|  |
| --- |
| * **css/style.css** — файл style.css в директории css * **css/\*.css** — все файлы с расширением .css в директории css * **js/\*\*/\*.js** — все файлы с расширением .js в директории js и всех дочерних директориях * **!js/app.js** — исключает определенный файл * **\*.+(js|css)** — все файлы в корневой директории с расширениями .js или .css |

Пример маски для всех файлов .css в директории css, кроме уже минифицированных:

|  |
| --- |
| **gulp.src(['**css/\*\*/\*.css**', '**!css/\*\*/\*.min.css**']);** |

# Gulp.dest()

Записывает в файл из потока, если папок в пути не существуют они будут созданы ([**ссылка**](https://github.com/gulpjs/gulp/blob/master/docs/API.md#gulpdestpath-options))

**gulp.dest(path[, options])**

1. **path**- *путь*
2. **options** - *объект настроек*

Пример использования при склеивании файлов:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **concatCss = require('**gulp-concat-css**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( concatCss('**fullstyle.css**') )**  **.pipe( gulp.dest('**app/css/**') );**  **});** |

# Gulp.task()

Объявляет задачу ([**ссылка**](https://github.com/gulpjs/gulp/blob/master/docs/API.md#gulptaskname-deps-fn))

**gulp.task(name[, deps], fn)**

1. **name** - *названия задачи*
2. **deps** - *массив задач которые должны быть выполнены и завершены до запуска данной*
3. **fn** - *функция выполняемая при вызове задачи*

Использоание:

|  |
| --- |
| **var gulp = require('**gulp**');**  **gulp.task('**default**', function() {**  **console.log('**default task**');**  **});** |

Использоание с **deps**:

|  |
| --- |
| **var gulp = require('**gulp**');**  **gulp.task('**default**', ['**your\_task\_1**', '**your\_task\_2**'], function() {**  **console.log('**default task**');**  **});** |

|  |
| --- |
| По умолчанию, задачи выполняются асинхронно - т.е., запускаются одновременно сразу все задачи без ожидания. Именно поэтому *deps*-задачи могут не соблюдать очередность |

Чтобы добиться очередности выполнения следует:

* *Сообщить что задача закончена*
* *Указать, что задача зависит от другой задачи*

Чтобы выполнить синхронно задачу:

1. Передать callback:

|  |
| --- |
| **gulp.task('**callback\_task**', function(callback\_func) {**  **// любая асинхронная задача**  **setTimeout(function () {**  **callback\_func();**  **}, 500);**  **});** |

1. Вернуть поток:

|  |
| --- |
| **gulp.task('**stream\_task**', function() {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( concatCss('**fullstyle.css**') )**  **.pipe( gulp.dest('**app/css/**') );**  **});** |

1. Вернуть [**промис**](https://github.com/kriskowal/q):

|  |
| --- |
| **gulp.task('**promise\_task**', function() {**  **var deferred = Q.defer();**  **// любая асинхронная задача**  **setTimeout(function () {**  **deferred.resolve();**  **}, 500);**  **return deferred.promise;**  **});** |

Выполняем задачи последовательно:

|  |
| --- |
| **var gulp = require('**gulp**');**  **// Первая задача**  **gulp.task('**first\_task**', function() { });**  **// Вторая задача**  **gulp.task('**second\_task**', ['**first\_task**'], function() {**  **// Не запустится пока не выполнится первая задача**  **});**  **gulp.task('**default**', ['**first\_task**', '**second\_task**'], function() { });** |

Gulp позволяет определить таск по умолчанию (тот самый с названием “*default*”), который будет запускаться по команде *gulp:*

|  |
| --- |
| **$ gulp** |

Для вызова конкретного таска - указываем его имя*:*

|  |
| --- |
| **$ gulp task\_name** |

# 

# Gulp.watch()

Следит за файлами, и выполняет действия в случае их изменения ([**ссылка**](https://github.com/gulpjs/gulp/blob/master/docs/API.md#gulpwatchglob--opts-tasks-or-gulpwatchglob--opts-cb))

**gulp.watch(glob[, opts], tasks)** *(вариант 1)*

1. **glob** - *файлы за которыми наблюдать (строка или массив)*
2. **opts** - *объект настроек*
3. **tasks** - массив *названий задач запускаемых при изменении файла*

Использование:

|  |
| --- |
| **var watcher = gulp.watch('**js/\*\*/\*.js**', ['**task\_1**', '**task\_2**']);**  **watcher.on('**change**', function(event) {**  **console.log('**File change - running tasks...**');**  **});** |

**gulp.watch(glob[, opts, cb])** *(вариант 2)*

1. **glob** - *файлы за которыми наблюдать (строка или массив)*
2. **opts** - *объект настроек*
3. **cb** - *callback-функция вызываемая при изменении*

Использование:

|  |
| --- |
| **var watcher = gulp.watch('**js/\*\*/\*.js**', function(event) {**  **console.log('**File change - running tasks...**');**  **});** |

* **event.type** - тип изменения *added, changed* или *deleted*
* **event.path** - путь к файлу, который вызвал событие

# 

# 

# 

**MODULES**

# 

# 

# Gulp-concat-css

Модуль для склеивания css файлов ([**ссылка**](https://www.npmjs.com/package/gulp-concat-css))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-concat-css** |

*сокращенно - $ npm i -D gulp-concat-css*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **concatCss = require('**gulp-concat-css**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( concatCss('**fullstyle.css**') )**  **.pipe( gulp.dest('**app/css/**') );**  **});** |

**concatCss(targetFile, options)**

1. **targetFile -** *относительный путь генерируемого файла*
2. **options -** *объект настроек*

# 

# 

# Gulp-minify-css

Модуль для минификации css файлов ([ссылка](https://www.npmjs.com/package/gulp-minify-css))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-minify-css** |

*сокращенно - $ npm i -D gulp-minify-css*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **concatCss = require('**gulp-concat-css**'),**  **minifyCss = require('**gulp-minifycss**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( concatCss('**fullstyle.css**') )**  **.pipe( minifyCss() )**  **.pipe( gulp.dest('**app/css/**') );**  **});** |

**minifyCss (options)**

1. **path -** *объект настроек*

# Gulp-rename

Модуль для переименования файлов ([**ссылка**](https://www.npmjs.com/package/gulp-rename))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-rename** |

*сокращенно - $ npm i -D gulp-rename*

**uglify (options)**

1. **options -** строка пути*, callback-функция, hash*

Строка:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **rename = require('**gulp-rename**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.js**')**  **.pipe( rename('**new.js**') )**  **.pipe( gulp.dest('**dist/js/**') );**  **});** |

*Callback-функция*:

|  |
| --- |
| **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.js**')**  **.pipe( rename( function (path) {**  **path.dirname += '/**js**';**  **path.basename += '**new**';**  **path.extname = '**.js**'**  **}))**  **.pipe( gulp.dest('**dist**') );**  **});** |

*Hash*:

|  |
| --- |
| **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.js**')**  **.pipe( rename( {**  **dirname : '/**js**',**  **basename : '**new**',**  **prefix : '**pre-**',**  **suffix : '**.full**',**  **extname : '**.js**'**  **} )**  **.pipe( gulp.dest('**dist**') );**  **});** |

# Gulp-uglify

Модуль для минификации js файлов ([**ссылка**](https://www.npmjs.com/package/gulp-uglify))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-uglify** |

*сокращенно - $ npm i -D gulp-uglify*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **rename = require('**gulp-rename**'),**  **uglify = require('**gulp-uglify**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.js**')**  **.pipe( uglify() )**  **.pipe( rename('**main.min.js**') )**  **.pipe( gulp.dest('**dist/js**') );**  **});** |

**uglify (options)**

1. **options -** *объект настроек*

# Gulp-autoprefixer

Модуль для управления браузерными префиксами ([**ссылка**](https://www.npmjs.com/package/gulp-autoprefixer))

Его работа основана на [**сайте**](http://caniuse.com/)

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-autoprefixer** |

*сокращенно - $ npm i -D gulp-autoprefixer*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **autoprefixer = require('**gulp-autoprefixer**');**  **gulp.task('**default**', function () {**  **return gulp.src('**app/\*\*/\*.css**')**  **.pipe( autoprefixer({**  **browsers: ['**last 2 versions**'],**  **cascade: false**  **}))**  **.pipe( gulp.dest('**dist/js**') );**  **});** |

**autoprefixer (options)**

1. **options -** *объект настроек*

# Gulp-jade

Модуль для компиляции jade файлов в html ([**ссылка**](https://www.npmjs.com/package/gulp-jade))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-jade** |

*сокращенно - $ npm i -D gulp-jade*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **jade = require('**gulp-jade**');**  **gulp.task('**jade\_task**', function () {**  **gulp.src('**app/\*\*/\*.jade**')**  **.pipe( jade({pretty: true}) )**  **.pipe( gulp.dest('**dist**') );**  **});** |

**jade (options)**

1. **options -** *объект настроек*

# Gulp-sass

Модуль для компиляции sass файлов ([**ссылка**](https://www.npmjs.com/package/gulp-sass))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-sass** |

*сокращенно - $ npm i -D gulp-sass*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **sass = require('**gulp-sass**');**  **gulp.task('**sass\_task**', function () {**  **gulp.src('**app/\*\*/\*.sass**')**  **.pipe( sass({outputStyle: '**compressed**'}) )**  **.pipe(gulp.dest('**dist/css**'));**  **});** |

**sass (options)**

1. **options -** *объект настроек*

# Gulp-uncss

Модуль для оптимизации css файлов, на основе использования в html ([**ссылка**](https://www.npmjs.com/package/gulp-uncss))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-uncss** |

*сокращенно - $ npm i -D gulp-uncss*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **uncss = require('**gulp-uncss**');**  **gulp.task('**sass\_task**', function () {**  **gulp.src('**app/\*\*/\*.css**')**  **.pipe( uncss({ html: [ '**app/index.html**' ] }) )**  **.pipe(gulp.dest('**dist/css**'));**  **});** |

**uncss (options)**

1. **options -** *объект настроек*

# Gulp-imagemin

Модуль для сжатия изображений ([**ссылка**](https://www.npmjs.com/package/gulp-imagemin))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-imagemin** |

*сокращенно - $ npm i -D gulp-imagemin*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **imagemin = require('**gulp-imagemin**');**  **gulp.task('**imagemin\_task**', function () {**  **gulp.src('**app/img/\*\*/\***')**  **.pipe( imagemin( {**  **progressive: true,**  **interlaced: true**  **} )**  **.pipe(gulp.dest('**dist/css**'));**  **) );**  **});** |

**clean (options)**

1. **options -** *объект настроек*

# 

# 

# Del

Модуль для удаления папок и файлов ([**ссылка**](https://www.npmjs.com/package/del))

Установка:

|  |
| --- |
| **$ npm install --save-dev del** |

*сокращенно - $ npm i -D del*

Использование:

|  |
| --- |
| **var del = require('**del**');**  **del(['**tmp/\*.js**', '**!tmp/unicorn.js**'], function (err, paths) {**  **console.log('**Deleted files/folders:\n**', paths.join('**\n**'));**  **});** |

**del (patterns, [options], callback)**

1. **patterns -** *строка или массив паттернов*
2. **options -** *объект настроек*
3. **callback -** *callback-функция*

# Gulp-util

Модуль утилит ([**ссылка**](https://www.npmjs.com/package/gulp-util))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-util** |

*сокращенно - $ npm i -D gulp-util*

Использование:

|  |
| --- |
| **var gutil = require('**gulp-util**');**  **gutil.log('**stuff happened**', '**Really it did**', gutil.colors.magenta('**123**'));** |

# 

# Vinyl-ftp

Модуль для работы по ftp ([**ссылка**](https://www.npmjs.com/package/vinyl-ftp))

Установка:

|  |
| --- |
| **$ npm install --save-dev vinyl-ftp** |

*сокращенно - $ npm i -D vinyl-ftp*

|  |
| --- |
| **ВНИМАНИЕ!!! Ни в коем случае не заливайте данные FTP доступа на github, это приведет к взлому вашего сайта!!!** |

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **gutil = require('**gulp-util**'),**  **ftp = require('**vinyl-ftp**');**  **gulp.task('**deploy**', function () {**  **var conn = ftp.create({**  **host: '**your\_host**',**  **user: '**your\_user**',**  **password: '**your\_password**',**  **parallel: 10,**  **log: gutil.log**  **});**  **var globs = [**  **'**dist/\*\*/\***'**  **];**  **return gulp.src(globs, { base: '**dist/**', buffer: false})**  **.pipe(conn.dest('**public\_html/**'));**  **});** |

**clean (options)**

1. **options -** *объект настроек*

# Browser-sync

Модуль для синхронизации с браузером ([**ссылка**](https://www.npmjs.com/package/browser-sync))

Установка:

|  |
| --- |
| **$ npm install --save-dev browser-sync** |

*сокращенно - $ npm i -D browser-sync*

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **browserSync = require('**browser-sync**');**  **gulp.task('**browser\_sync**', function () {**  **browserSync({**  **port: 9000,**  **open: true,**  **notify: false,**  **server: '**./app**'**  **});**  **});** |

**browserSync (options)**

1. **options -** *объект настроек*

# 

# 

# Gulp-useref

Модуль для конкатенации стилей и скриптов через парсинг спец. блоков ([**ссылка**](https://www.npmjs.com/package/gulp-useref))

Установка:

|  |
| --- |
| **$ npm install --save-dev gulp-useref** |

*сокращенно - $ npm i -D gulp-useref*

Спец. блок

|  |
| --- |
| **<html>**  **<head>**  **<!-- build:css css/combined.css -->**  **<link href="css/one.css" rel="stylesheet">**  **<link href="css/two.css" rel="stylesheet">**  **<!-- endbuild -->**  **</head>**  **<body>**  **<!-- build:js scripts/combined.js -->**  **<script type="text/javascript" src="scripts/one.js"></script>**  **<script type="text/javascript" src="scripts/two.js"></script>**  **<!-- endbuild -->**  **</body>**  **</html>** |

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **useref = require('**gulp-useref**');**  **gulp.task('**useref\_task**', function () {**  **var assets = useref.assets();**  **gulp.src('**app/\*.html**')**  **.pipe( assets )**  **.pipe( assets.restore() )**  **.pipe( useref() )**  **.pipe(gulp.dest('**dist**'));**  **});** |

**useref (options)**

1. **options -** *объект настроек*

Результат:

|  |
| --- |
| **<html>**  **<head>**  **<link href="css/combined.css" rel="stylesheet">**  **</head>**  **<body>**  **<script type="text/javascript" src="scripts/combined.js"></script>**  **</body>**  **</html>** |

# Wiredep

Модуль для автоподключения [**bower**](http://www.youtube.com/playlist?list=PLY4rE9dstrJwqwHBF4KFMIo9U0HKIR3UP) компонентов через спец. блоки ([**ссылка**](https://www.npmjs.com/package/wiredep))

Установка:

|  |
| --- |
| **$ npm install --save-dev wiredep** |

*сокращенно - $ npm i -D wiredep*

Спец. блок

|  |
| --- |
| **<html>**  **<head>**  **<!-- bower:css -->**  **<!-- endbower -->**  **</head>**  **<body>**  **<!-- bower:js -->**  **<!-- endbower -->**  **</body>**  **</html>** |

Использование:

|  |
| --- |
| **var gulp = require('**gulp**'),**  **wiredep = require('**wiredep**').stream;**  **gulp.task('**wiredep\_task**', function () {**  **gulp.src('**app/templates/\*.html**')**  **.pipe( wiredep ({**  **ignorePath: /^(\.\.\/)\*\.\./**  **}) )**  **.pipe(gulp.dest('**app/temp/templates**'));**  **});** |

**wiredep (options)**

1. **options -** *объект настроек*

Так же можно работать и с jade файлами:

|  |
| --- |
| **gulp.task('**wiredep\_jade**', function () {**  **gulp.src('**app/templates/\*.jade**')**  **.pipe( wiredep ({**  **ignorePath: /^(\.\.\/)\*\.\./**  **}) )**  **.pipe(gulp.dest('**app/temp/templates**'));**  **});** |

Пропишем в самом jade файле

|  |
| --- |
| **doctype html**  **html(lang='**ru-RU**')**  **head**  **// bower:css**  **// endbower**  **script(src='**bower/modernizr/modernizr.js**')**  **body**  **…**  **// bower:js**  **// endbower** |

modernizr.js лучше подключить вручную т.к. он должен быть в head

Wiredep автоматически определит какие файлы в какой последовательности необходимо подключать. Чтобы указать другие файлы (например чтобы подключить .min файл) необходимо в нашем bower.json файле указать следующее:

|  |
| --- |
| **{**  **“**overrides**” : {**  **“**qtip2**” : {**  **“**main**” : [**  **“**./jquery.qtip.min.js**” ,**  **“**./jquery.qtip.min.css**”**  **],**  **“**dependencies**” : {**  **“**jquery**” : “**>=1.6.0**”**  **}**  **}**  **}**  **}** |

**main** - какие файлы подключать

**dependencies** - зависимости

# 

**GULP BLACKLIST**

# 

# Gulp Blacklist

Нерекомендуемые модули ([**ссылка**](https://github.com/gulpjs/plugins/blob/master/src/blackList.json) на полный список)

* **gulp-clean** - use the `del` module
* **gulp-browserify** - use the browserify module directly
* **gulp-rimraf** - use the `del` module
* **gulp-image-optimization** - duplicate of gulp-imagemin
* **gulp-bower** - use the bower module directly
* **gulp-php** - use PHP directly through gulp-spawn or ChildProcess.spawn()
* **gulp-tinypng** - uses fs to create a temp .gulp folder
* **gulp-css** - duplicate of gulp-minify-css
* **gulp-cssmin** - duplicate of gulp-minify-css
* **gulp-clean-old** - duplicate of gulp-clean
* **gulp-if-else** - duplicate of gulp-if
* **gulp-prettify** - duplicate of gulp-html-prettify
* [**...**](https://github.com/gulpjs/plugins/blob/master/src/blackList.json)