Gulp教程

使用Gulp、bower、构建Angularjs项目详细教程，附构建成功源码

亲爱哒，接下来由我带你了解一下Gulp构建工具，要了解一个新知，当然是先去官网啦！那我们就去看下吧：

Gulp官网地址：<https://www.gulpjs.com.cn/>

1. Gulp是什么？

Gulp是一个自动化构建工具，用于增强你的工作流程！简单地说，Gulp可用于代码打包，压缩，混淆，转化，等等一系列自动化构建。

特点：简单易用，构建快速，用最少的API完成更高质量的工作，详细请看官网介绍；

1. 如何使用Gulp工具？

我们根据官网的指导一步两步，一步两步，一步两步是爪牙，是魔鬼的步伐~~~

2.1 入门指南

1. 全局安装 gulp：

$ npm install --global gulp

2. 作为项目的开发依赖（devDependencies）安装：

$ npm install --save-dev gulp

3. 在项目根目录下创建一个名为 gulpfile.js 的文件：

var gulp = require('gulp');

gulp.task('default', function() {

// 将你的默认的任务代码放在这

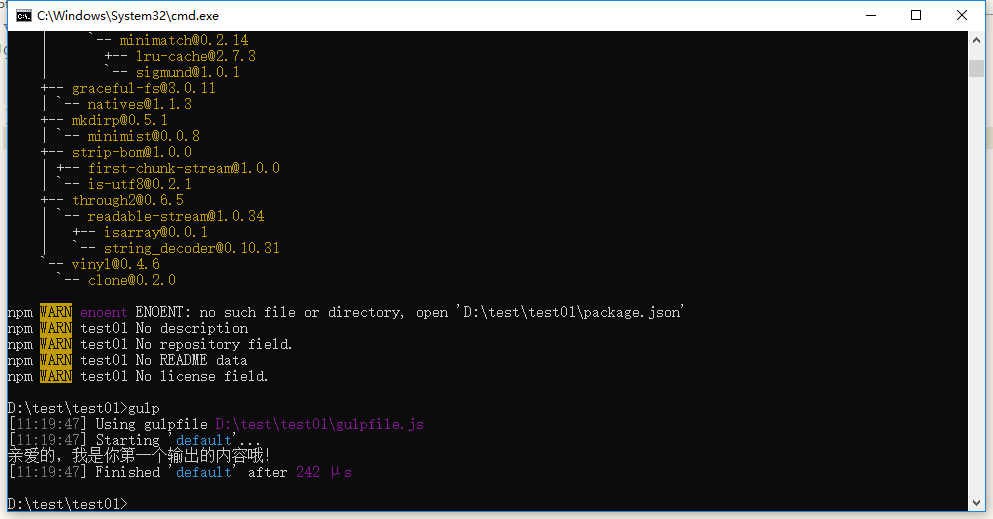
});

4. 运行 gulp：

$ gulp

默认的名为 default 的任务（task）将会被运行，在这里，这个任务并未做任何事情。

下面我们就按照实例自己创建一个项目吧：



Tips: 运行 gulp 命令，启动的是名为 ‘default’ 的任务流；

附件：



2.2 API文档

（1） gulp.src(globs[, options])

输出（Emits）符合所提供的匹配模式（glob）或者匹配模式的数组（array of globs）的文件。 将返回一个 [Vinyl files](https://github.com/wearefractal/vinyl-fs) 的 [stream](http://nodejs.org/api/stream.html) 它可以被 [piped](http://nodejs.org/api/stream.html#stream_readable_pipe_destination_options) 到别的插件中。

gulp.src('client/js/\*\*/\*.js') // 匹配 'client/js/somedir/somefile.js' 并且将 `base` 解析为 `client/js/`

.pipe(minify())

.pipe(gulp.dest('build')); // 写入 'build/somedir/somefile.js'

gulp.src('client/js/\*\*/\*.js', { base: 'client' })

.pipe(minify())

.pipe(gulp.dest('build')); // 写入 'build/js/somedir/somefile.js'

（2）gulp.task(name[, deps], fn)

定义一个使用 [Orchestrator](https://github.com/robrich/orchestrator) 实现的任务（task）

gulp.task('mytask', ['array', 'of', 'task', 'names'], function() {

// 做一些事

});

（3）gulp.watch(glob [, opts], tasks) 或 gulp.watch(glob [, opts, cb])

监视文件，并且可以在文件发生改动时候做一些事情。它总会返回一个 EventEmitter 来发射（emit） change 事件。

var watcher = gulp.watch('js/\*\*/\*.js', ['uglify','reload']);

watcher.on('change', function(event) {

console.log('File ' + event.path + ' was ' + event.type + ', running tasks...');

});

gulp.watch(glob[, opts, cb])

gulp.watch('js/\*\*/\*.js', function(event) {

console.log('File ' + event.path + ' was ' + event.type + ', running tasks...');

});

（4）gulp.dest(path[, options])

能被 pipe 进来，并且将会写文件。并且重新输出（emits）所有数据，因此你可以将它 pipe 到多个文件夹。如果某文件夹不存在，将会自动创建它。

1. 使用Gulp工具构建第一个工程项目：

要求：

需要压缩html，less，css，js，图片（各种格式的），并且需要代码混淆（防止反编译），热更新（实时刷新页面），gulp出问题提示代码报错行数,使用bower工具安装依赖

详细步骤：

1. 创建文件夹： Gulp\_myFirst 目录下创建 gulpfile.js
2. 运行命令： npm init 初始化项目，会生成package.json文件

bower init 会生成bower.json文件

1. 在该目录下运行命令：npm install --save-dev gulp 目录下会生成node\_module文件夹，
2. 根据需求，安装所需要的插件：

不禁想起《童年》里的一句： “……福利社里面什么都有，就是口袋里没有半毛钱……”，口袋里没钱不要紧，这里的资源直接npm就可以；

npm 官网： <https://www.npmjs.com/>

表3-1 项目所需一来插件及插件功能简介列表

|  |  |
| --- | --- |
| gulp-htmlmin | 压缩html文件 |
| gulp-less | 将less编译为css；注意version 4.0.0版本在gulp时，凡是有@import的less文件时，被引入的文件发生改变时，主文件不会自动编译css，所以本项目采用的version3.1.0 |
| gulp-minify-css | 压缩CSS，压缩成一行 |
| gulp-sourcemaps | 当less有各种引入关系时，编译后不容易找到对应less文件，所以需要生成sourcemap文件，方便修改 |
| gulp-jshint | 检查JS语法错误 |
| gulp-concat | 合并文件 |
| babel-core  gulp-babel  babel-preset-es2015 | ES6转ES5 |
| gulp-uglify | 压缩js文件， |
| gulp-imagemin | 压缩图片，使得文件体积变小 |
| gulp-connect | 建立一个本地服务器，connect.reload()，可以实现实时刷新 |
| open | Gulp完后，自动在浏览器打开一个网址 |
| gulp-plumber | 查找gulp时出错原因，并且打印出来 |
| gulp-clean | 删除文件或者文件夹，一般是在gulp时，删掉上次gulp的 |
| gulp-load-plugins | 按需加载插件，$. 唤醒函数 |
| express | 这个我也不好说，可用于发送请求 |
| http-proxy-middleware | 代理，可解决跨域问题 |
|  |  |
| 更多好玩的插件，尽在NPM官网 | |

我列出的这些，一般在项目上会经常用到，所以我都npm安装一下：

这么多插件，装起来需要些时间的，亲爱的，稍等下哦~~~

npm install --save-dev gulp-htmlmin gulp-less gulp-minify-css gulp-sourcemaps gulp-jshint gulp-concat babel-core gulp-babel babel-preset-es2015 gulp-uglify gulp-imagemin gulp-connect open gulp-plumber gulp-clean gulp-load-plugins

安装bower 依赖插件

bower install angular angular-animate angular-cookies angular-ui-router angular-validation jquery bootstrap echarts --save-dev

我2分钟就好了，嗯网速还不错；你呢？

1. 开始配置gulpfile.js文件

在这里我就不详细说明了，mydear，看我gulpfile.js 文件，里面的注释写的很详细的。

还有一点得说下，就是ES6转ES5的时候，需要配置一个文件，名为：

.babelrc 这个文件无文件名，只有后缀，一般新建一个txt文件，然后改成.babelrc你会发现windows不让你改，教你一招啊？

cmd运行： type NUL > .babelrc

然后 .babelrc 内容

{

"presets": ["es2015"]

}

1. 新建文件夹： src 改目录下，方式项目源码 html，css，js，img等文件
2. 在src写入你的文件后，cmd运行： gulp 或者 gulp serve 可启动项目；

gulp： 会清除之前dist目录下的文件

gulp serve： 不会清除之前dist目录下的文件

附件：

