

前端构建工具gulp入门教程

时间 2013-12-30 15:00:01 [SegmentFault](#)

原文

本文假设你之前没有用过任何任务脚本（task runner）和命令行工具，一步步教你上手Gulp。不要怕，它其实很简单，我会分为五步向你介绍gulp并帮助你完成一些惊人的事情。那就直接开始吧。

第一步：安装Node

首先，最基本也最重要的是，我们需要搭建node环境。访问 <http://node.js.org>，然后点击大大的绿色的 `install` 按钮，下载完成后直接运行程序，就一切准备就绪。`npm`会随着安装包一起安装，稍后会用到它。

第二步：使用命令行

也许现在你还不是很了解什么是命令行——OSX中的终端（Terminal），windows中的命令提示符（Command Prompt），但很快你就会知道。它看起来没那么简单，但一旦掌握了它的窍门，就可以很方便的执行很多命令行程序，比如Sass，Yeoman和Git等，这些都是非常有用的工具。

如果你很熟悉命令行，直接跳到步骤四。

为了确保Node已经正确安装，我们执行几个简单的命令。

```
node -v
```

回车（Enter），如果正确安装的话，你会看到所安装的Node的版本号，接下来看看npm。

```
npm -v
```

这同样能得到npm的版本号。

如果这两行命令没有得到返回，可能node就没有安装正确，尝试重启下命令行工具，如果还不行的话，只能回到第一步进行重装。

第三步：定位到项目

现在，我们已经大致了解了命令行并且知道如何简单使用它，接下来只需要两个简单的命令就能定位到文件目录并看看目录里都有些什么文件。

1. `cd`，定位到目录
2. `ls`，列出文件列表

建议多敲敲这两个命令，了解文件系统并知道文件都在哪里。

习惯使用了这两个命令后，就要进入我们的项目目录，这个目录各不相同，举个例子，这是我进入我项目目录的命令：

```
cd /Applications/XAMPP/xamppfiles/htdocs/my-project
```

成功进入项目目录后，我们开始安装gulp。

第四步：安装gulp

我们已经知道如何使用命令行，现在尝试点新的东西，认识npm然后安装gulp。

NPM是基于命令行的node包管理工具，它可以将node的程序模块安装到项目中，在它的 [官网](#) 中可以查看和搜索所有可用的程序模块。

```
sudo npm install -g gulp
```

1. sudo是以管理员身份执行命令，一般会要求输入电脑密码
2. npm是安装node模块的工具，执行install命令
3. -g表示在全局环境安装，以便任何项目都能使用它
4. 最后，gulp是将要安装的node模块的名字

运行时注意查看命令行有没有错误信息，安装完成后，你可以使用下面的命令查看gulp的版本号以确保gulp已经被正确安装。

```
gulp -v
```

接下来，我们需要将gulp安装到项目本地

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

这里，我们使用 `--save-dev` 来更新package.json文件，更新 `devDependencies` 值，以表明项目需要依赖gulp。

`Dependencies` 可以向其他参与项目的人指明项目在开发环境和生产环境中的node模块依赖关系，想要更加深入的了解它可以看看 [package.json文档](#)。

第五步：新建Gulpfile文件，运行gulp

安装好gulp后我们需要告诉它要为我们执行哪些任务，首先，我们自己需要弄清楚项目需要哪些任务。

- 检查Javascript
- 编译Sass（或Less之类的）文件
- 合并Javascript
- 压缩并重命名合并后的Javascript

安装依赖

```
npm install gulp-jshint gulp-sass gulp-concat gulp-uglify gulp-rename --save-dev
```

提醒下，如果以上命令提示权限错误，需要添加 `sudo` 再次尝试。

新建gulpfile文件

现在，组件都安装完毕，我们需要新建gulpfile文件以指定gulp需要为我们完成什么任务。

gulp只有五个方法：`task`，`run`，`watch`，`src`，和`dest`，在项目根目录新建一个js文件并命名为 `gulpfile.js`，把下面的代码粘贴进去：

gulpfile.js

```
// 引入 gulp
var gulp = require('gulp');

// 引入组件
var jshint = require('gulp-jshint');
var sass = require('gulp-sass');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');

// 检查脚本
gulp.task('lint', function() {
  gulp.src('./js/*.js')
    .pipe(jshint())
    .pipe(jshint.reporter('default'));
});

// 编译Sass
gulp.task('sass', function() {
  gulp.src('./scss/*.scss')
    .pipe(sass())
    .pipe(gulp.dest('./css'));
});

// 合并，压缩文件
```

```
gulp.task('scripts', function() {
  gulp.src('./js/*.js')
    .pipe(concat('all.js'))
    .pipe(gulp.dest('./dist'))
    .pipe(rename('all.min.js'))
    .pipe(uglify())
    .pipe(gulp.dest('./dist'));
});

// 默认任务
gulp.task('default', function() {
  gulp.run('lint', 'sass', 'scripts');

  // 监听文件变化
  gulp.watch('./js/*.js', function() {
    gulp.run('lint', 'sass', 'scripts');
  });
});
```

现在，分段解释下这段代码。

引入组件

```
var gulp = require('gulp');

var jshint = require('gulp-jshint');
var sass = require('gulp-sass');
var concat = require('gulp-concat');
var uglify = require('gulp-uglify');
var rename = require('gulp-rename');
```

这一步，我们引入了核心的gulp和其他依赖组件，接下来，分开创建lint, sass, scripts 和 default 这四个不同的任务。

Lint任务

```
gulp.task('lint', function() {
  gulp.src('./js/*.js')
    .pipe(jshint())
    .pipe(jshint.reporter('default'));
});
```

Link任务会检查 `js/` 目录下得js文件有没有报错或警告。

Sass任务

```
gulp.task('sass', function() {  
  gulp.src('./scss/*.scss')  
    .pipe(sass())  
    .pipe(gulp.dest('./css'));  
});
```

Sass任务会编译 `scss/` 目录下的scss文件，并把编译完成的css文件保存到 `/css` 目录中。

Scripts 任务

```
gulp.task('scripts', function() {  
  gulp.src('./js/*.js')  
    .pipe(concat('all.js'))  
    .pipe(gulp.dest('./dist'))  
    .pipe(rename('all.min.js'))  
    .pipe(uglify())  
    .pipe(gulp.dest('./dist'));  
});
```

scripts任务会合并 `js/` 目录下得所有得js文件并输出到 `dist/` 目录，然后gulp会重命名、压缩合并的文件，也输出到 `dist/` 目录。

default任务

```
gulp.task('default', function() {  
  gulp.run('lint', 'sass', 'scripts');  
  gulp.watch('./js/*.js', function() {  
    gulp.run('lint', 'sass', 'scripts');  
  });  
});
```

这时，我们创建了一个基于其他任务的default任务。使用 `.run()` 方法关联和运行我们上面定义的任务，使用 `.watch()` 方法去监听指定目录的文件变化，当有文件变化时，会运行回调定义的其他任务。

现在，回到命令行，可以直接运行gulp任务了。

```
gulp
```

这将执行定义的default任务，换言之，这和以下的命令式同一个意思

```
gulp default
```

当然，我们可以运行在gulpfile.js中定义的任何任务，比如，现在运行sass任务：

```
gulp sass
```

(Kimi：哇塞，酷比了哎~)

结束语

现在已经做到了设置gulp任务然后运行他们，现在再回顾下之前学习的。

1. 学习了安装Node环境
2. 学习了简单使用命令行
3. 学习了用命令行进入项目目录
4. 学习了使用npm和安装gulp
5. 学习了如何运行gulp任务

另外，有一些参考资源供进一步学习：

分享

收藏

纠错

极光推送 - 最专业的推送服务平台