

# 前端自动化工具Gulp

设计部 黄卉 2016.09.22 - 2016.10.27

# 主要内容

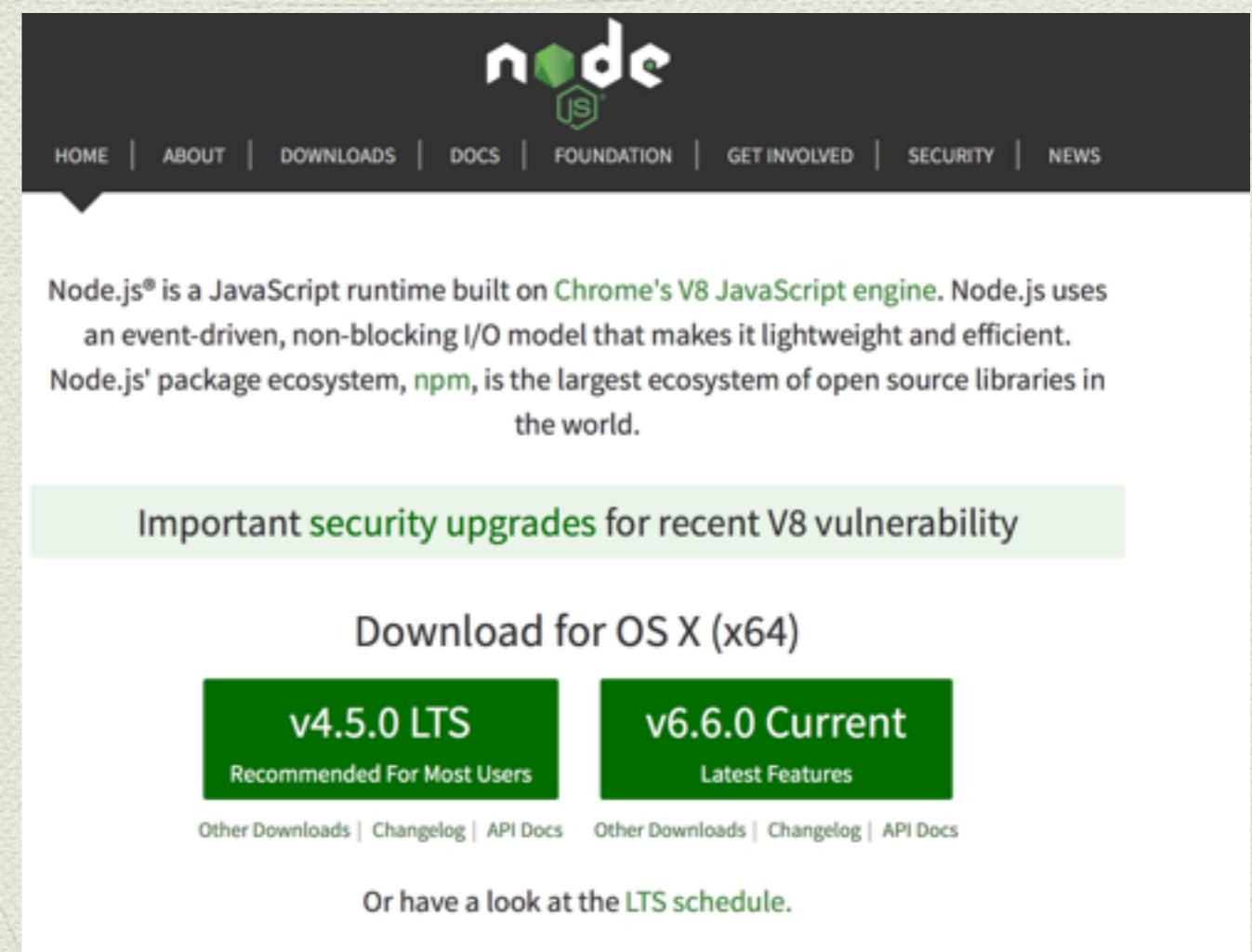
- ◆ 前期准备
- ◆ 项目配置文件
- ◆ Gulp基本使用用法
- ◆ Gulp常见插件

# 一、前期准备

- ◆ 1.1 安装node和npm
- ◆ 1.2 npm用法
- ◆ 1.3 用cnpm替代npm

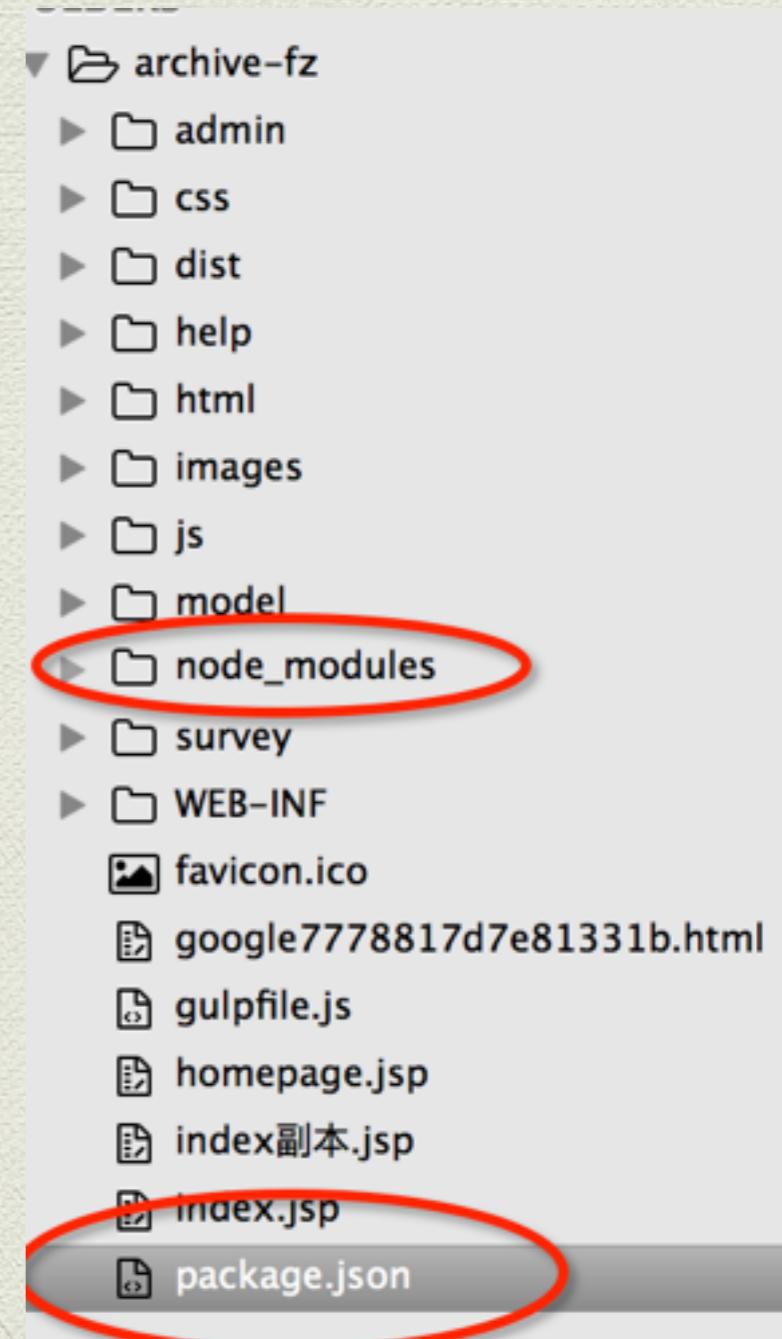
# 1.1 安装node和npm

- ◆ 安装node， gulp是基于nodejs的。
- ◆ 安装链接：<https://nodejs.org/en/>
- ◆ 安装npm：node的包管理工具，用于node插件管理（包括安装、卸载、管理依赖等）

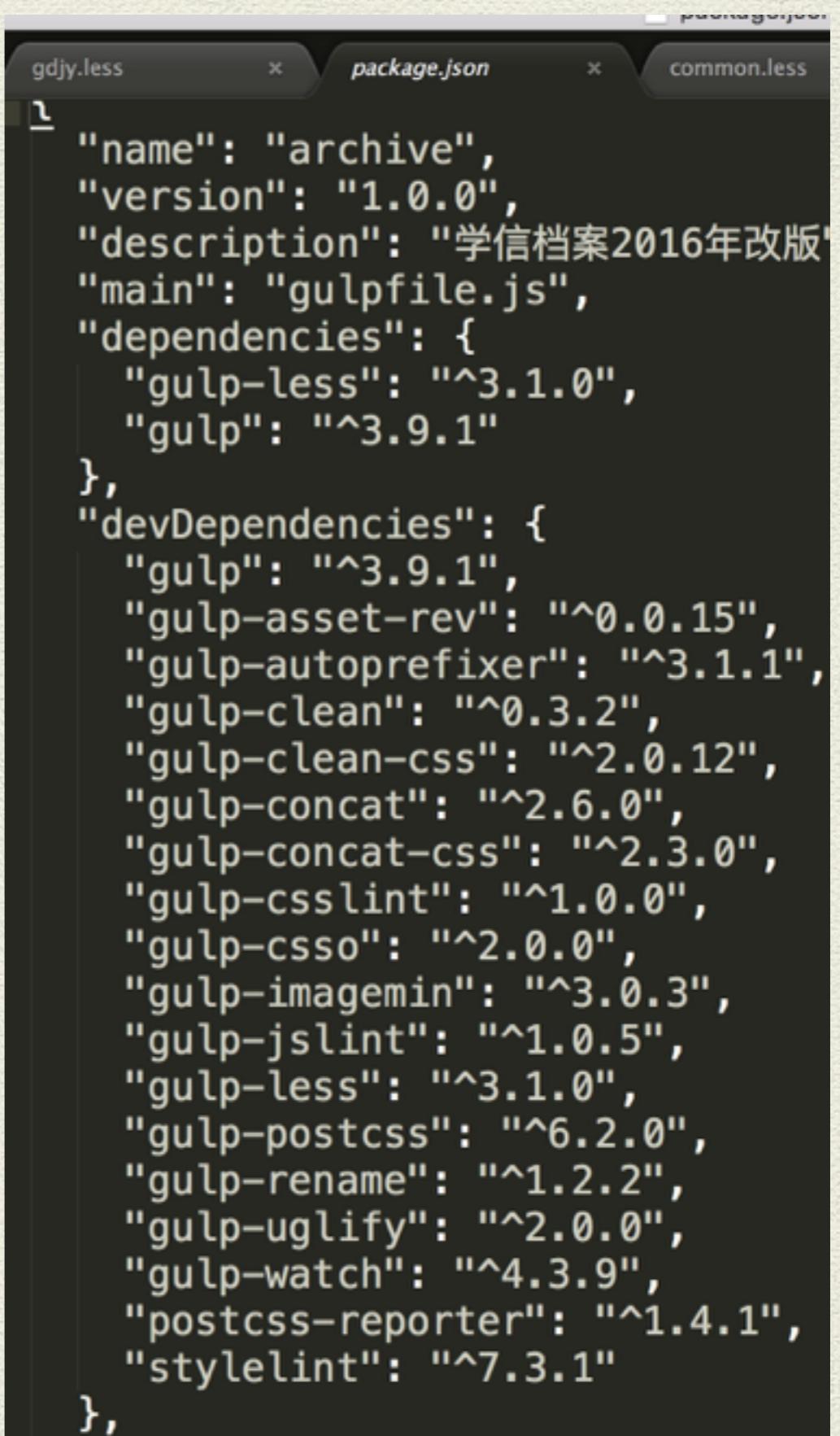


# 1.2 npm用法

- ◆ 用法: `npm install <name> [-g][--save-dev]`
- ◆ 示例:
  - ◆ `npm install gulp -g`
  - ◆ **`npm install gulp-less --save-dev`**
- ◆ name: node插件名称
- ◆ -g: 全局安装, 安装在本地, 并写入系统环境变量中。在任务位置调用node插件。不加则为局部安装, 安装在当前定位目录的`node_modules`文件夹下, 通过`require('node插件名')`调用。



- ◆ -- save: 将保存配置信息至 package.json (package.json是node项目的配置文件)；
- ◆ -dev: 将保存至 package.json的 devDependencies节点，不指定-dev将保存至 dependencies节点
- ◆ package.json两种方法生成，如何生成，有何作用？



```
gdjy.less          package.json          common.less
{
  "name": "archive",
  "version": "1.0.0",
  "description": "学信档案2016年改版",
  "main": "gulpfile.js",
  "dependencies": {
    "gulp-less": "^3.1.0",
    "gulp": "^3.9.1"
  },
  "devDependencies": {
    "gulp": "^3.9.1",
    "gulp-asset-rev": "^0.0.15",
    "gulp-autoprefixer": "^3.1.1",
    "gulp-clean": "^0.3.2",
    "gulp-clean-css": "^2.0.12",
    "gulp-concat": "^2.6.0",
    "gulp-concat-css": "^2.3.0",
    "gulp-csslint": "^1.0.0",
    "gulp-csso": "^2.0.0",
    "gulp-imagemin": "^3.0.3",
    "gulp-jslint": "^1.0.5",
    "gulp-less": "^3.1.0",
    "gulp-postcss": "^6.2.0",
    "gulp-rename": "^1.2.2",
    "gulp-uglify": "^2.0.0",
    "gulp-watch": "^4.3.9",
    "postcss-reporter": "^1.4.1",
    "stylelint": "^7.3.1"
  }
}
```

# 1.3 用cnpm代替npm

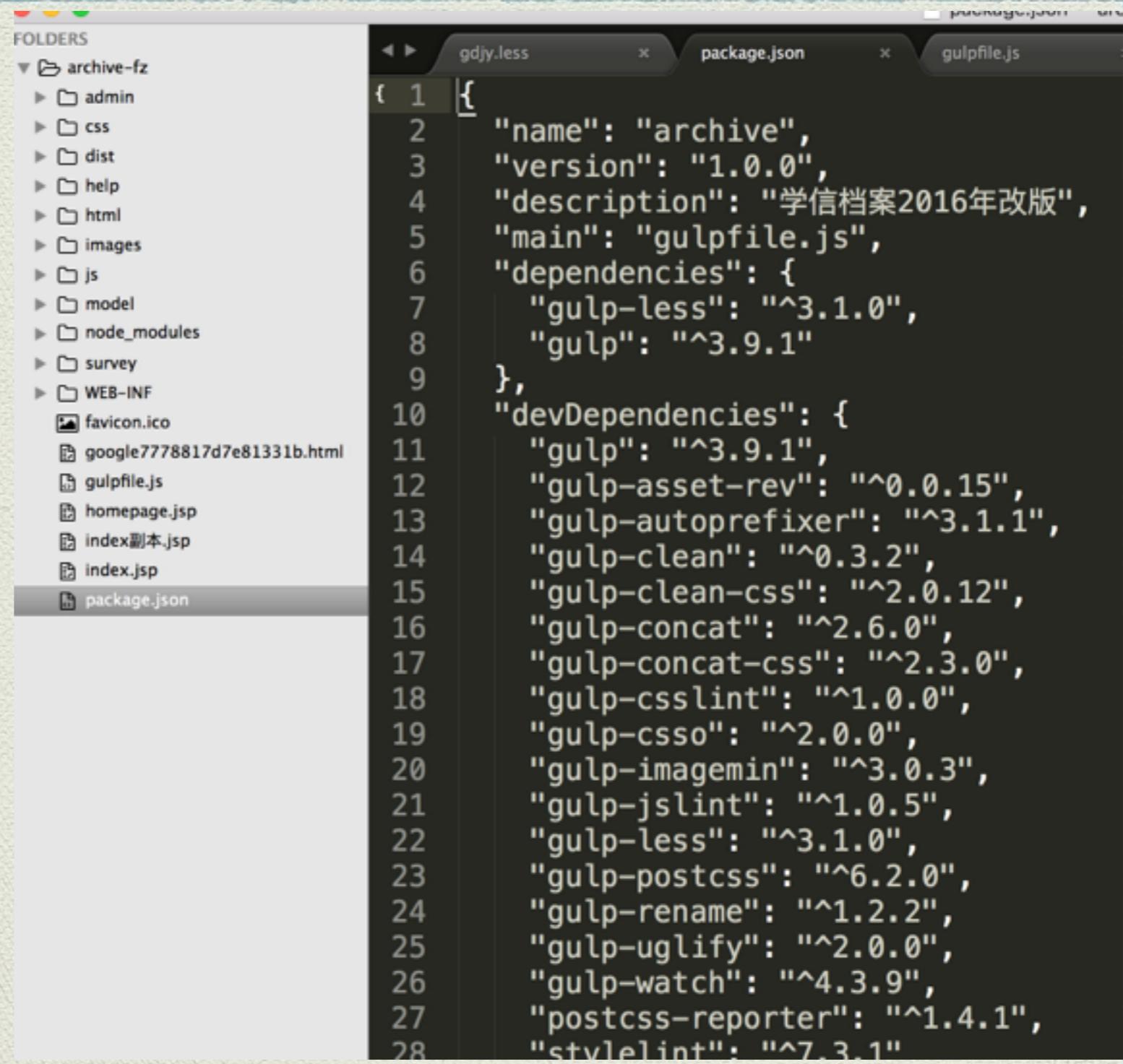
- ◆ cnpm速度快
- ◆ 安装方法： npm install cnpm -g —registry=<https://registry.npm.taobao.org>
- ◆ cnpm跟npm用法完全一致，只是在执行命令时将npm改为cnpm

## 二、项目配置文件

- ◆ 2.1 项目配置文件package.json
- ◆ 2.2 项目配置文件gulpfile.js

# 2.1 项目配置文件package.json

◆ package.json文件是基于nodejs项目必不可少的配置文件，放在根目录下的普通json文件



```
{ 1 {
 2   "name": "archive",
 3   "version": "1.0.0",
 4   "description": "学信档案2016年改版",
 5   "main": "gulpfile.js",
 6   "dependencies": {
 7     "gulp-less": "^3.1.0",
 8     "gulp": "^3.9.1"
 9   },
10   "devDependencies": {
11     "gulp": "^3.9.1",
12     "gulp-asset-rev": "^0.0.15",
13     "gulp-autoprefixer": "^3.1.1",
14     "gulp-clean": "^0.3.2",
15     "gulp-clean-css": "^2.0.12",
16     "gulp-concat": "^2.6.0",
17     "gulp-concat-css": "^2.3.0",
18     "gulp-csslint": "^1.0.0",
19     "gulp-cssso": "^2.0.0",
20     "gulp-imagemin": "^3.0.3",
21     "gulp-jslint": "^1.0.5",
22     "gulp-less": "^3.1.0",
23     "gulp-postcss": "^6.2.0",
24     "gulp-rename": "^1.2.2",
25     "gulp-uglify": "^2.0.0",
26     "gulp-watch": "^4.3.9",
27     "postcss-reporter": "^1.4.1",
28     "stylelint": "^7.3.1"
29 }
```

- ◆ package.json生成方法：
  - ◆ 初次生成，可通过命令行直接生成该配置文件： **npm init**
  - ◆ 如果已有package.json文件，则可以通过命令行将devDependencies配置中的相关依赖插件自动生成到node\_modules目录下。自动生成命令： **npm install --save-dev**

# 2.2 项目配置文件gulpfile.js

- ◆ gulpfile.js是gulp项目的配置文件，是位于项目根目录的普通js文件（其实将gulpfile.js放入其他文件夹下亦可）

The screenshot shows a code editor with several tabs at the top: gdjy.less, package.json, gulpfile.js (which is the active tab), common.less, and index.jsp. On the left, there's a sidebar titled 'FOLDERS' showing the project structure:

```
gdjy
  |- archive-fz
    |- admin
    |- css
    |- dist
    |- help
    |- html
    |- images
    |- js
    |- model
    |- node_modules
    |- survey
    |- WEB-INF
      |- favicon.ico
      |- google7778817d7e81331b.html
  |- gulpfile.js
  |- homepage.jsp
  |- index副本.jsp
  |- index.jsp
  |- package.json
```

The main pane displays the contents of the gulpfile.js file:

```
1  // gulp配置文件，整合css/js/images等自动化插件
2  * @description gulp配置文件，整合css/js/images等自动化插件
3  * @author huanghui
4  * @date 20160808
5  */
6  var gulp = require('gulp'), //本地安装gulp所用到的地方
7    less = require('gulp-less'), //less文件转为css
8    autoprefixer = require('gulp-autoprefixer'), //css添加厂商前缀
9    csllint = require('gulp-csllint'), //检查css语法
10   rename = require('gulp-rename'), //重命名
11   cssmin = require('gulp-clean-css'), //压缩css
12   concatCss = require('gulp-concat-css'),
13   watch = require('gulp-watch'), //即时编译
14   assetRev = require('gulp-asset-rev'), //添加版本号
15   concat = require('gulp-concat'), //js合并
16   uglify = require('gulp-uglify'), //js压缩
17   clean = require('gulp-clean'), //删除文件，做操作前先删除
18   imgmin = require('gulp-imagemin'); //图片压缩
19
20 var ArrAll = {
21   commonLessDir : 'css/common', //需要解析的less文件目录
22   customLessDir: 'css/custom',
23   cssDir : 'src/css', //需要解析的less文件目录
24   cssRes : 'dist/css', //生成的css目录
25   cssMin : 'dist/css' , //生成的min.css目录
26   jsSrc : 'src/js', //js目录
27   jsRes : 'dist/js', //js目录
```

# 三、Gulp基本使用方法

- ◆ 3.1 Gulp安装步骤
- ◆ 3.2 Gulp语法
- ◆ 3.3 Gulp具体用法
- ◆ 3.3 Gulp插件的卸载

# 3.1 Gulp安装步骤

- ◆ 第一步：全局安装Gulp：(sudo) npm install --global gulp
- ◆ 第二步：作为项目的开发依赖安装： npm install --save-dev gulp
- ◆ 第三步：在项目根目录下创建一个名为**gulpfile.js**的文件，然后建立需要执行的任务（手动建），例如default：

```
var gulp = require('gulp');
gulp.task('default',
function() {
  // 将需要执行的任务代码放在这
});
```

- ◆ 第四步：运行gulp： gulp (gulpname)

## 3.2 Gulp语法

- ◆ `gulp.task(name[, deps], fn)` 定义任务。name：任务名称，deps：依赖任务名称，fn：回调函数
- ◆ `gulp.src(globs[, options])` 执行任务处理的文件。glob：处理的文件路径(字符串或者字符串数组)
- ◆ `gulp.pipe(fn)`，执行的模块。fn：调用的模块。
- ◆ `gulp.dest(path[, options])` 处理完后文件生成路径

### 3.3 Gulp具体用法

```
//导入工具包 require('node_modules里对应模块')
var gulp = require('gulp'), //本地安装gulp所用到的地方
    less = require('gulp-less');

//定义一个testLess任务 (自定义任务名称)
gulp.task('testLess', function () {
    gulp.src('src/less/index.less') //该任务针对的文件
        .pipe(less()) //该任务调用的模块
        .pipe(gulp.dest('src/css')); //将会在src/css下生成
index.css
});

gulp.task('default',['testLess', 'elseTask']); //定义默认任务
elseTask为其他任务，该示例没有定义elseTask任务
```

## 3.4 Gulp插件的卸载

- ◆ 使用npm卸载插件： `npm uninstall <name> -g` , 不要直接删除本地插件包
- ◆ 删除全部插件： `npm uninstall gulp-less gulp-uglify gulp-concat .....`
- ◆ 借助rimraf删除全部插件：
  - ◆ 安装rimraf: `npm install rimraf -g`
  - ◆ 用法: `rimraf node_modules`

# 四、 Gulp常用插件

- ◆ 4.1 公用插件
- ◆ 4.2 css相关插件
- ◆ 4.2 js相关插件
- ◆ 4.3 images相关插件

# 4.1 公用插件

- ◆ gulp-rename：重命名
- ◆ gulp-watch：及时更改文件，执行gulp操作，不用手动执行
- ◆ gulp-asset-rev：加入版本号，例如：“a.js”改成“a-md5编码.js”。（修改下源码，让远程链接也可以修改版本号，同时版本号所在位置也进行了修改。“a.js”改成“a.js?v=1477569249589”）
- ◆ gulp-clean：删除目录

- ◆ 如何修改jsp页面静态资源的版本号?
  - ◆ 修改gulp-asset-rev插件中的index.js
  - ◆ 远程链接和本地连接
  - ◆ 在gulpfile.js配置文件中写gulp方法
  - ◆ 在node下面执行： gulp 自定义gulp方法

```
//index页面
gulp.task('indexRev',function() {
    gulp.src('index.jsp')
        .pipe(assetRev())
        .pipe(gulp.dest(''));
});
ArrAll.layoutSrc = 'WEB-INF/view/jsp/archives/layout';
//layout页面
gulp.task('layoutRev',function() {
    gulp.src(ArrAll.layoutSrc+'mainLayout_my.jsp')
        .pipe(assetRev())
        .pipe(gulp.dest(ArrAll.layoutSrc));
});
```

## ◆ 执行： gulp layoutRev后的效果如下：

```
archive-fz gulp layoutRev
[11:40:54] Using gulpfile ~/Documents/workspace/chsi/archive-fz/gulpfile.js
[11:40:54] Starting 'layoutRev'...
[11:40:54] Finished 'layoutRev' after 7.04 ms
远程连接更改: http://t1.chei.com.cn/common/jquery/1.9.1/jquery.min.js?v=14775396
54666
远程连接更改: https://t1.chei.com.cn/common/bootstrap3.3.5/js/bootstrap.min.js?v
=1477539654666
远程连接更改: https://t1.chei.com.cn/common/ch/browser/es5.min.js?v=147753965466
6
远程连接更改: https://t1.chei.com.cn/common/bootstrap3.3.5/css/bootstrap.min.css
?v=1477539654667
远程连接更改: https://t1.chei.com.cn/common/ch/iconfont.css?v=1477539654667
archive-fz |
```

```
header-new.jsp      *   index1.jsp      *   gulpfile.js      *   mainLayout_my.jsp      *   index.js      *   index.jsp      *   index
<link rel="icon" href="<%=_request.getContextPath()%>/favicon.ico" type="image/
x-icon" />
<!--新的样式-->
<link rel="stylesheet" href="https://t1.chei.com.cn/common/bootstrap3.3.5/css/
bootstrap.min.css?v=1477539654667" />
<link rel="stylesheet" href="https://t1.chei.com.cn/common/ch/
iconfont.css?v=1477539654667" />
<link rel="stylesheet" href="/archive/css/custom/gdjiy.css"/>
<script type="text/javascript" src="http://t1.chei.com.cn/common/jquery/1.9.1/
jquery.min.js?v=1477539654666"></script>
<script src="https://t1.chei.com.cn/common/bootstrap3.3.5/js/
bootstrap.min.js?v=1477539654666"></script>
<jsp:include page="/WEB-INF/view/jsp/archives/common/ga.jsp" flush="true" />
</head>
<body>
```

- ◆ gulp-asset-rev 添加版本号，并修改远程连接给其加入版本号。

The screenshot shows a code editor with several tabs at the top: gulpfile.js, index.js, index.jsp, and index.html. The left sidebar shows a file tree:

- node\_modules
  - .bin
  - .npminstall
  - gulp
  - gulp-asset-rev
    - node\_modules
    - .npminstall.done
    - index.js
    - package.json
    - README.md
  - gulp-autoprefixer

The main editor area displays the content of index.js:

```
// remote resource
if (/^https?:\/\/\//.test(src)) {
    //return str;
    /**
     * @description 修改远程图片版本号，如果已存在则替换
     * @modify huangh@chsi.com.cn
     * @date 20160804
     */
    var date = new Date().getTime();
    var hasVer = src.indexOf('?v=');
    if(hasVer>-1){
        src = src.substring(0,hasVer);
    }
    src = src + "?v=" + date;
    console.log('远程连接更改: '+src);
    return tag + "''" + src + "''";
}
```

```
sp x index1.jsp x gulpfile.js x index.js x index.jsp x index.less x
if (fs.existsSync(assetPath)) {

    var buf = fs.readFileSync(assetPath);

    var md5 = createHash(buf, options.hashLen || 7);
    /**
     * @description 更改文件版本号输出方式
     * @modify      huangh@chsi.com.cn
     * @date        20160804
     */
    //var verStr = (options.verConnecter || "-") + md5;
    //src = src.replace(verStr, '').replace(/(\.[^\.]+)$/, verStr + "$1");
    var verStr = (options.verConnecter || "") + md5;
    var date = new Date().getTime();
    console.log('本地连接更改: '+src);
    src = src + "?v=" + date;
} else {
    return str;
}
```

- ◆ 本地连接更改输入方式，改成“a.js”-»  
“a.js?V=date”

## 4.2 css相关插件

- ◆ gulp-less：将less文件编译成css，当有less文件发生改变自动编译less，并保证less语法错误或出现异常时能正常工作并提示错误
- ◆ gulp-autoprefixer：根据设置浏览器版本自动处理浏览器前缀
- ◆ gulp-clean-css：压缩css文件，减小文件大小，并给引用url添加版本号避免缓存
- ◆ gulp-csslint/stylelint，检查css语法

```
gulp.task('cusionLess', function () {
  gulp.src(ArrAll.customLessDir+'/*.less') //该任务针对的文件和不对其进行处理的文件
    .pipe(less()) //将less转化为css
    .pipe(autoprefixer({
      browsers: ['Android 2.3', 'Android >= 4', 'iOS >= 6',
        'Explorer >= 6', 'Chrome >= 20', 'Firefox >= 24', 'Opera >= 12'],
      cascade: true, //是否美化属性值 默认: true
      remove:true //是否去掉不必要的前缀 默认: true
    }))
    .pipe(assetRev()) //给css中的链接添加版本号
    .pipe(gulp.dest(ArrAll.customLessDir)) //生成css文件, less放到css源目录中
    .pipe(rename({ suffix: '.min' })) //重新命名, 添加后缀名
    .pipe(cssmin()) //压缩css
    .pipe(gulp.dest(ArrAll.cssDir)); //生成压缩文件
});
```

## ◆ 实例 - Less解析

```
gulp.task('concatCss', function(){
  gulp.src(ArrAll.cssDir+'/*.css')
    .pipe(concatCss('archive.css'))
    .pipe(autoprefixer({
      browsers: ['Android 2.3', 'Android >= 4', 'iOS >= 6',
      'Explorer >= 6', 'Chrome >= 20', 'Firefox >= 24', 'Opera >= 12'],
      cascade: true, //是否美化属性值 默认: true
      remove: true //是否去掉不必要的前缀 默认: true
    }))
    .pipe(assetRev()) //给css中的链接添加版本号
    .pipe(gulp.dest(ArrAll.cssRes))
    .pipe(rename({ suffix: '.min' })) //将文件名后缀改为.min
    .pipe(cssmin())
    .pipe(gulp.dest(ArrAll.cssMin));
});
```

## ◆ 实例 - CSS合并

# 4.3 js相关插件

- ◆ gulp-concat： 合并javascript文件， 减少网络请求
- ◆ gulp-uglify： 压缩javascript文件， 减小文件大小
- ◆ gulp-jslint， js代码检

```
gulp.task('jsConcat',function(){
  gulp.src(ArrAll.jsSrc+'*.js')
    .pipe(concat('all.js'))
    .pipe(gulp.dest(ArrAll.jsRes))
    .pipe(rename({ suffix: '.min' }))
    .pipe(uglify())
    .pipe(gulp.dest(ArrAll.jsMin));
});
```

# 4.4 images相关插件

- ◆ gulp-imagemin压缩图片文件（包括PNG、JPEG、GIF和SVG图片）
- ◆ gulp-css-spriter：将多张图自动合成雪碧图（待测试）

```
//图片处理
gulp.task('imgmin',function(){
  gulp.src(ArrAll.imgSrc+/*.{png,jpg,gif,ico}')
    .pipe(imgmin({
      optimizationLevel: 5, //类型: Number 默认: 3 取值范围: 0-7 (优化等级)
      progressive: true, //类型: Boolean 默认: false 无损压缩jpg图片
      interlaced: true, //类型: Boolean 默认: false 隔行扫描gif进行渲染
      multipass: true //类型: Boolean 默认: false 多次优化svg直到完全优化
    }))
    .pipe(gulp.dest(ArrAll.imgMin))
});
```

# 参考文档

- ◆ Gulp官网: <http://www.gulpjs.com.cn/>
- ◆ Gulp入门: <http://www.ydcss.com/>

The end ! Thanks !

- 设计部 黄卉