

# Grunt & Git 활용

---

작성자: 조성민

소속팀 : UIT 개발실

작성년월일: 2014. 8. 4

대외비

# 목차

---

## 1. Node.js & Grunt

1.1 Node.js

1.2 Grunt

1.3 Grunt 설치

1.4 Grunt 실행 환경 파일

1.5 Grunt 실행 환경 마무리

1.6 Gruntfile.js

---

# 1. Node.js & Grunt

### Node.js

1. Node.js는 Javascript를 서버에서도 사용할 수 있도록 설계가 되어 있는 서버-사이드 언어.
2. V8이라는 Javascript 엔진 위에서 동작하는 이벤트 처리 I/O 프레임워크.
3. 웹 서버와 같이 확장성 있는 네트워크 프로그램을 제작을 할 수 있도록 고안.



공식 홈페이지

<http://nodejs.org/>

설치

1. Windows, Mac, Unix 지원
2. 다운로드 페이지 : <http://nodejs.org/download/>

### Grunt란?

A task-based command line build tool for JavaScript projects.

1. Grunt는 Node.js로 만들어 졌음.
2. Task 중심의 Javascript 빌드 도구.
3. 비슷한 툴로써 Glup이 있음.

공식 홈페이지

<http://gruntjs.com/>

<http://gruntjs-kr.herokuapp.com/>

설치

1. 기본적으로 Node.js가 설치되어 있어야 함.
2. npm(Node Packaged Modules)을 통해 설치.



### Grunt(grunt-cli) 설치

```
$ npm install -g grunt-cli
```

1. npm : Node.js 설치시 기본으로 설치되는 Node.js 모듈 관리 도구
2. 설치 명령어 : **npm install <module>**
3. **grunt-cli** : 커맨드 라인에서 grunt 명령어를 사용할 수 있도록 해주는 모듈
4. **-g 옵션** : global 옵션으로 시스템의 어떤 곳에서도 grunt 명령어를 사용할 수 있도록 해주며, 최초 한번만 설치하면 더 이상 설치하지 않아도 됨.

※ Unix(Mac) 에서 -g 옵션 사용시 sudo 명령어를 통해 관리자 권한으로 실행해야 함.

```
$ sudo npm install -g grunt-cli
```



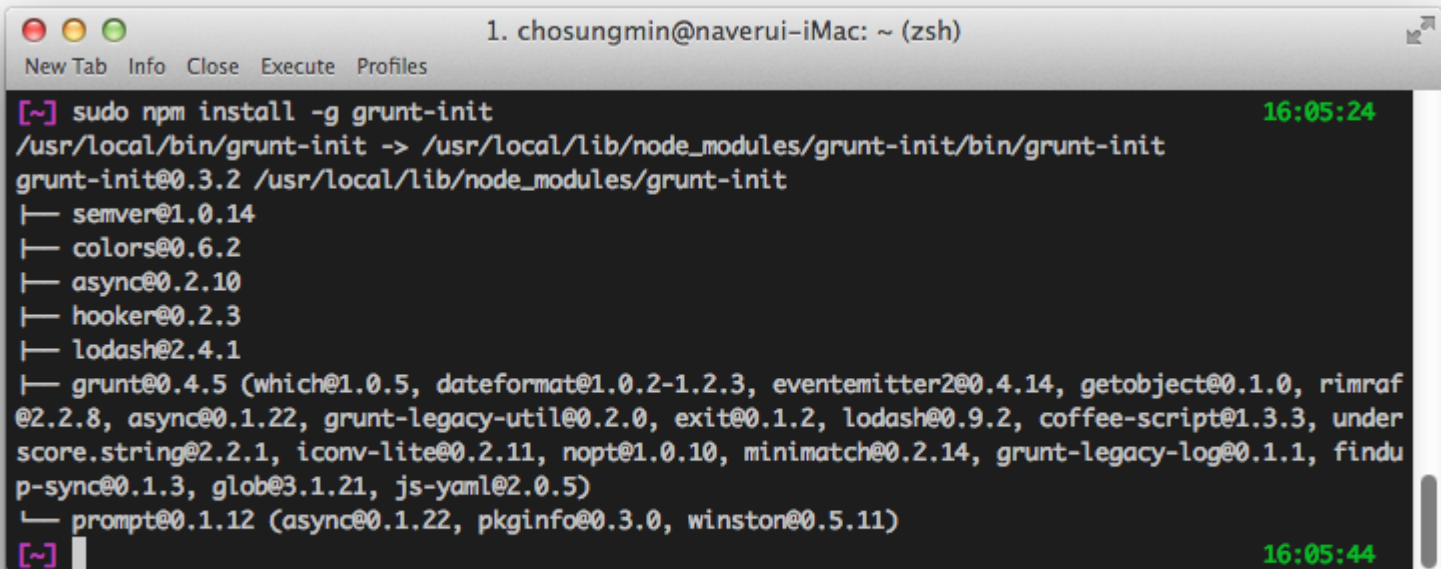
```
1. chosungmin@naverui-iMac: ~ (zsh)
New Tab Info Close Execute Profiles
[~] sudo npm install -g grunt-cli 16:02:02

Password:
/usr/local/bin/grunt -> /usr/local/lib/node_modules/grunt-cli/bin/grunt
grunt-cli@0.1.13 /usr/local/lib/node_modules/grunt-cli
├─ resolve@0.3.1
├─ nopt@1.0.10 (abbrev@1.0.5)
└─ findup-sync@0.1.3 (lodash@2.4.1, glob@3.2.11)
[~] 16:02:46
```

### Grunt(grunt-cli) 설치

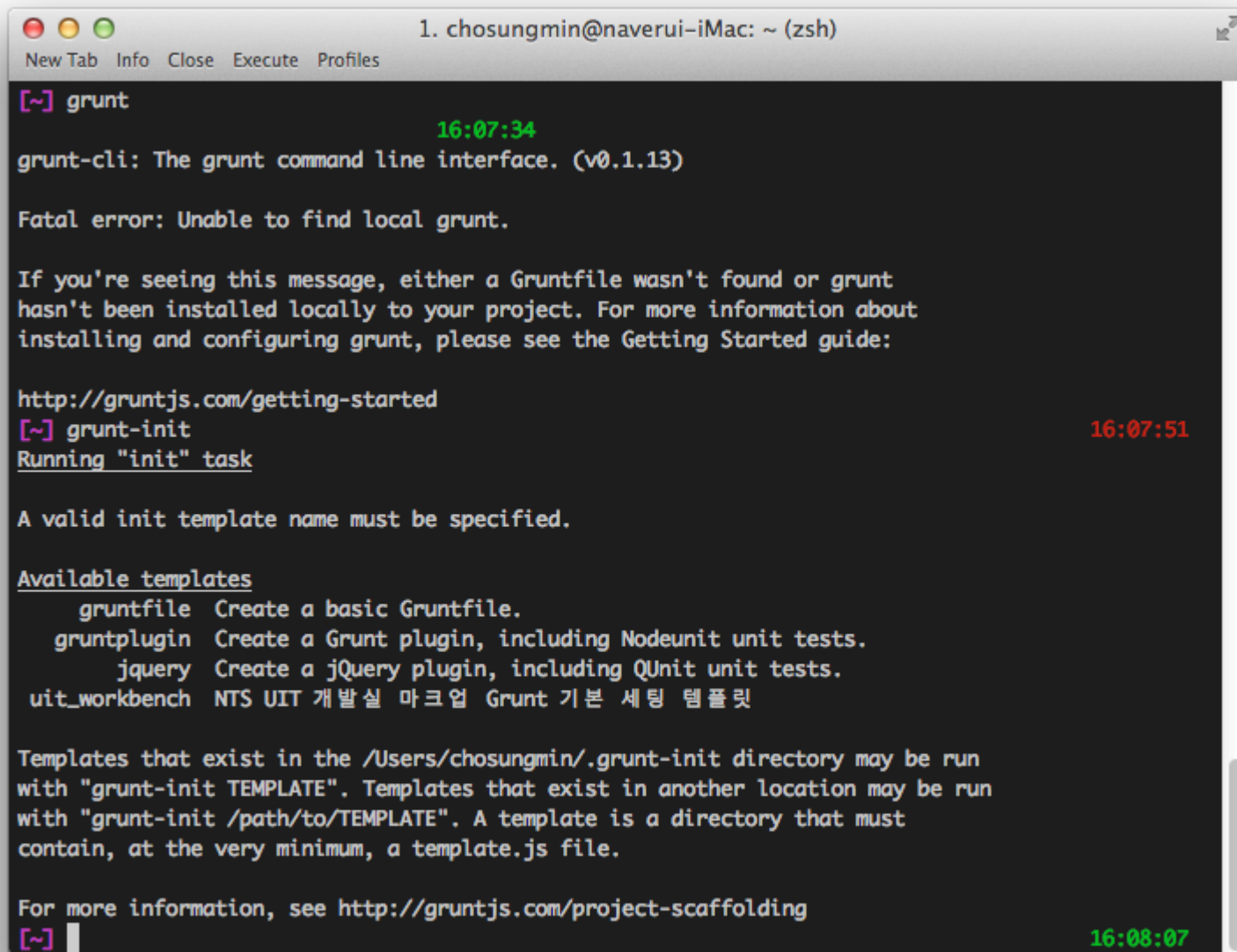
```
$ npm install -g grunt-init
```

1. **grunt-init** : grunt 템플릿을 사용할 수 있도록 해주는 모듈
2. 템플릿 : 프로젝트에 기반이 되는 파일과 구조를 프로젝트에 맞게 자동으로 만들어 줌.



```
1. chosungmin@naverui-iMac: ~ (zsh)
New Tab Info Close Execute Profiles

[~] sudo npm install -g grunt-init 16:05:24
/usr/local/bin/grunt-init -> /usr/local/lib/node_modules/grunt-init/bin/grunt-init
grunt-init@0.3.2 /usr/local/lib/node_modules/grunt-init
├─ semver@1.0.14
├─ colors@0.6.2
├─ async@0.2.10
├─ hooker@0.2.3
├─ lodash@2.4.1
├─ grunt@0.4.5 (which@1.0.5, dateformat@1.0.2-1.2.3, eventemitter2@0.4.14, getobject@0.1.0, rimraf
@2.2.8, async@0.1.22, grunt-legacy-util@0.2.0, exit@0.1.2, lodash@0.9.2, coffee-script@1.3.3, under
score.string@2.2.1, iconv-lite@0.2.11, nopt@1.0.10, minimatch@0.2.14, grunt-legacy-log@0.1.1, findu
p-sync@0.1.3, glob@3.1.21, js-yaml@2.0.5)
├─ prompt@0.1.12 (async@0.1.22, pkginfo@0.3.0, winston@0.5.11)
[~] 16:05:44
```



```
1. chosungmin@naverui-iMac: ~ (zsh)
New Tab Info Close Execute Profiles

[~] grunt
16:07:34
grunt-cli: The grunt command line interface. (v0.1.13)

Fatal error: Unable to find local grunt.

If you're seeing this message, either a Gruntfile wasn't found or grunt
hasn't been installed locally to your project. For more information about
installing and configuring grunt, please see the Getting Started guide:

http://gruntjs.com/getting-started
[~] grunt-init
16:07:51
Running "init" task

A valid init template name must be specified.

Available templates
  gruntfile  Create a basic Gruntfile.
  gruntplugin Create a Grunt plugin, including Nodeunit unit tests.
  jquery     Create a jQuery plugin, including QUnit unit tests.
  uit_workbench NTS UIT 개발실 마크업 Grunt 기본 세팅 템플릿

Templates that exist in the /Users/chosungmin/.grunt-init directory may be run
with "grunt-init TEMPLATE". Templates that exist in another location may be run
with "grunt-init /path/to/TEMPLATE". A template is a directory that must
contain, at the very minimum, a template.js file.

For more information, see http://gruntjs.com/project-scaffolding
[~] 16:08:07
```



### Grunt 실행 환경 파일

Grunt 를 이용하기 위해서는 프로젝트(작업) 폴더에 `package.json` 과 `Gruntfile.js` 라는 파일이 필요.

#### `package.json`

1. 해당 프로젝트를 npm 모듈로 npm에 퍼블리싱할 때 사용하는 메타데이터 정보를 가지고 있음.
2. 프로젝트에 필요한 grunt와 grunt 모듈은 이 파일의 devDependencies라는 항목에 나열.

```
$ npm install grunt --save-dev
```

위와 같이 명령어 입력 시 `package.json` 파일의 devDependencies 항목에 자동으로 grunt 모듈을 추가함.

3. “npm init” 명령어 실행 시 `package.json` 기본 파일 생성됨.

#### `Gruntfile.js`

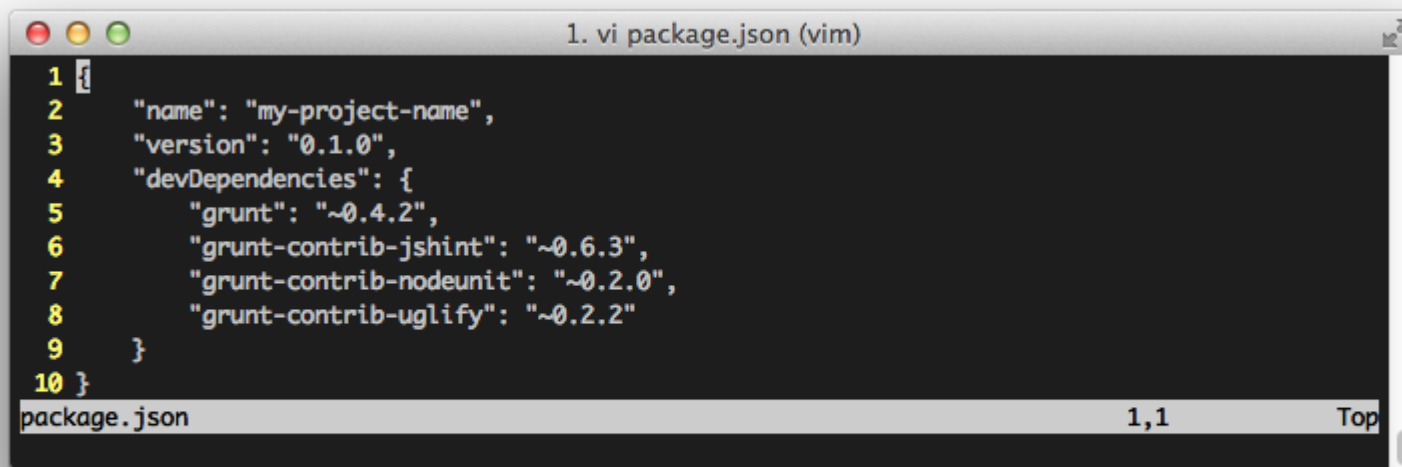
1. 이 파일의 이름은 `Gruntfile.js`이거나 `Gruntfile.coffee`이어야 함.
2. task를 설정하거나 정의하고 Grunt 플러그인을 불러오는데 사용.
3. `package.json` 과 함께 프로젝트 Root 폴더에 있어야 함.

### Grunt 실행 환경 마무리

grunt 템플릿(grunt-init 이용) 등을 이용하여 package.json, Gruntfile.js 파일을 생성했다면 아래 명령을 통해 grunt 실행 환경 구축을 마무리 함.

```
$ npm install
```

1. “npm install” 명령어에서 패키지 이름을 지정해 주지 않을 경우에는 npm이 package.json 파일을 검색하며, 파일이 존재할 시 **package.json devDependencies 항목의 패키지 정보를 읽어와 해당 패키지들을 자동으로 모두 설치해 줌.**
2. 모듈 설치는 node\_modules 폴더 아래에 설치 됨.



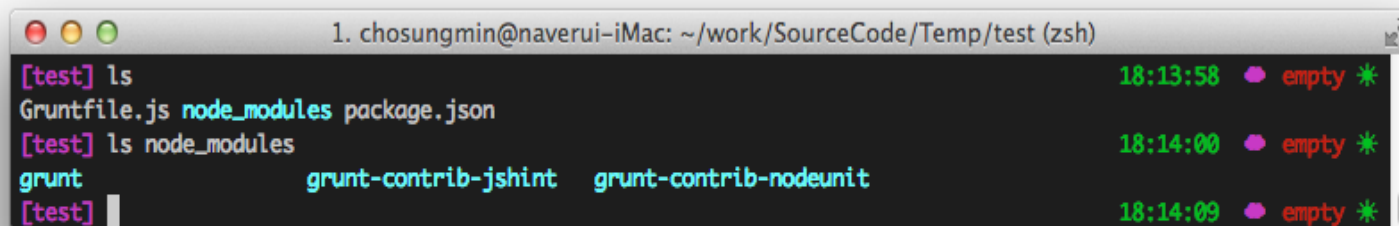
```
1 {  
2   "name": "my-project-name",  
3   "version": "0.1.0",  
4   "devDependencies": {  
5     "grunt": "~0.4.2",  
6     "grunt-contrib-jshint": "~0.6.3",  
7     "grunt-contrib-nodeunit": "~0.2.0",  
8     "grunt-contrib-uglify": "~0.2.2"  
9   }  
10 }
```

package.json 1,1 Top

package.json 소스 코드이다.

“npm install” 실행 시 “grunt, grunt-contrib-jshint, grunt-contrib-nodeunit, grunt-contrib-uglify” 모듈이 자동으로 설치 된다.

아래는 설치 후 폴더 구조이다.



```
1. chosungmin@naverui-iMac: ~/work/SourceCode/Temp/test (zsh)  
[test] ls 18:13:58 empty *  
Gruntfile.js node_modules package.json  
[test] ls node_modules 18:14:00 empty *  
grunt grunt-contrib-jshint grunt-contrib-nodeunit  
[test] 18:14:09 empty *
```

### Gruntfile.js 구성요소

1. "wrapper" 함수.
2. 프로젝트와 task의 환경설정.
3. grunt plugin과 task 로딩.
4. 사용자 정의 task

Gruntfile.js

```
module.exports = function(grunt) {  
  
  // 프로젝트 환경설정.  
  grunt.initConfig({  
    pkg: grunt.file.readJSON('package.json'),  
    uglify: {  
      options: {  
        banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n',  
      },  
      build: {  
        src: 'src/<%= pkg.name %>.js',  
        dest: 'build/<%= pkg.name %>.min.js'  
      }  
    }  
  });  
  
  // "uglify" task를 지원하는 플러그인 로드.  
  grunt.loadNpmTasks('grunt-contrib-uglify');  
  
  // Default task(s).  
  grunt.registerTask('default', ['uglify']);  
  
};
```

### “wrapper” 함수

```
module.exports = function(grunt) {  
  // grunt 관련 코드는 여기에  
};
```

1. 모든 Gruntfile(플러그인 포함)은 wrapper 함수를 기본 형태로 사용함.
2. 모든 Grunt 코드는 wrapper 함수 안쪽에 있어야 함.

### 프로젝트와 task의 환경설정

```
// 프로젝트 환경설정.
grunt.initConfig({
  pkg: grunt.file.readJSON('package.json'),
  uglify: {
    options: {
      banner: '/*! <%= pkg.name %> <%= grunt.template.today("yyyy-mm-dd") %> */\n'
    },
    build: {
      src: 'src/<%= pkg.name %>.js',
      dest: 'build/<%= pkg.name %>.min.js'
    }
  }
});
```

1. 대부분의 Grunt task는 configuration을 위한 데이터로 `grunt.initConfig` 메서드의 인자로 전달되는 객체를 사용함.  
(이를 configuraton 객체라 함)
2. 여러분의 task가 필요로하는 프로퍼티와 충돌하지 않는 이상 configuraton 객체안에는 어떤 값을 넣어도 상관없으며, 이 파일은 단순 JSON이 아닌 자바스크립트 파일이므로, 유효한 JS코드는 모두 사용가능함.  
즉, 필요한 configuration을 동적으로 생성할 수도 있음.
3. `<% %>` : 템플릿 문자열을 사용해서 config의 모든 프로퍼티를 참조할 수 있도록 해 줌.  
`<%= pkg.name %>` : “my-project-name”
4. “uglify, pkg”는 Task 이름, “banner, src, dest”는 옵션 이름, “build”는 타겟명

\$ grunt uglify:build

### grunt plugin과 task 로딩

```
// "uglify" task를 지원하는 플러그인 로드.  
grunt.loadNpmTasks('grunt-contrib-uglify');  
grunt.loadNpmTasks('xxxxxxx');  
grunt.loadNpmTasks('xxxxxxxxxxx');
```

```
.  
. .  
. .  
. .
```

1. 사용할 플러그인을 로드할 때 사용.
2. 여러 플러그인을 자동으로 로딩할때는 “load-grunt-tasks” 모듈을 이용하면 편함.  
(<https://www.npmjs.org/package/load-grunt-tasks>)

```
require('load-grunt-tasks')(grunt);
```

### 사용자 정의 task

```
// Default task(s).  
grunt.registerTask('default', ['uglify']);  
grunt.registerTask('test', ['uglify', 'csslint']);  
.  
.  
.
```

1. default task를 설정하면 기본으로 실행할 task를 하나 혹은 그 이상 지정할 수 있음.

```
$ grunt  
$ grunt default
```

커맨드 라인에 특정 task를 지정하지 않고 grunt만 입력하면 uglify task가 실행됨.

2. 배열 안에는 다수의 task를 지정할 수도 있음. (argument가 있을 수도 있고 없을 수도 있음.)



### 사용자 정의 task

```
module.exports = function(grunt) {  
  
  // default task의 기본형.  
  grunt.registerTask('default', 'Log some stuff.', function() {  
    grunt.log.write('Logging some stuff...').ok();  
  });  
  
};
```

1. 만약 프로젝트에서 Grunt 플러그인 목록에 없는 task가 필요하다면, Gruntfile에 직접 custom task를 만들 수도 있음.
2. 프로젝트 전용 task라면 Gruntfile안에 정의할 필요없이. 별도의 .js 파일로 정의하고 grunt.loadTasks 메서드로 불러 올 수도 있음.

# Thank you.