

Building and Deployment: Task Runners: Grunt and Gulp

Reading: Building and Deployment: Task Runners

10 min

Exercise (Instructions): Grunt Part 2

Objectives and Outcomes

In this exercise, you will continue to learn to use Grunt, the task runner. You will configure the Grunt file with a set of additional tasks to build your web project. At the end of this exercise, you will be able to:

- Configure a Grunt file with a set of tasks to build your web project from a source.

Copying the Files and Cleaning Up the Dist Folder

- Next you will install the Grunt modules to copy over files to a distribution folder named dist, and clean up the dist folder when needed. To do this, install the following Grunt modules:

```
1 npm install grunt-contrib-copy@1.0.0 --save-dev
2 npm install grunt-contrib-clean@1.1.0 --save-dev
```

- You will now add the code to perform the copying of files to the dist folder, and cleaning up the dist folder. To do this, add the following code to *Gruntfile.js*. This should be added right after the configuration of the SASS task.:

```
1 ,
2
3   copy: {
4     html: {
5       files: [
6         {
7           //for html
8           expand: true,
9           dot: true,
10          cwd: './',
11          src: ['*.html'],
12          dest: 'dist'
13        }
14      ]
15    },
16    fonts: {
17      files: [
18        {
19          //for font-awesome
20          expand: true,
21          dot: true,
22          cwd: 'node_modules/font-awesome',
23          src: ['fonts/*.'],
24          dest: 'dist'
25        }
26      ]
27    },
28    clean: [
29      build: {
30        src: [ 'dist/' ]
31      }
32    ]
33  }
```

- Remember to add the comma after the end of the SASS task.

Compressing and Minifying Images

- Next we install the grunt-contrib-imagemin module and use it to process the images. To install this module type at the prompt:

```
1 npm install grunt-contrib-imagemin@2.0.1 --save-dev
```

- Then, configure the imagemin task as shown below in the Gruntfile:

```
1 ,
2   imagemin: [
3     dynamic: {
4       files: [
5         {
6           expand: true,           // Enable dynamic expansion
7           cwd: './',             // Src matches are relative to
8           src: ['img/*.{png,jpg,gif}'], // Actual patterns to match
9           dest: 'dist/'          // Destination path prefix
10        }
11      ]
12    }
13  ]
```

Preparing the Distribution Folder and Files

- We are now going to use the Grunt *usemin* module together with *concat*, *cssmin*, *uglify* and *filerev* to prepare the distribution folder. To do this, install the following Grunt modules:

```
1 npm install grunt-contrib-concat@1.0.1 --save-dev
2 npm install grunt-contrib-cssmin@2.2.1 --save-dev
3 npm install grunt-contrib-htmlmin@2.4.0 --save-dev
4 npm install grunt-contrib-uglify@3.3.0 --save-dev
5 npm install grunt-filerev@2.3.1 --save-dev
6 npm install grunt-usemin@3.1.1 --save-dev
```

- Next, update the task configuration within the Gruntfile.js with the following additional code to introduce the new tasks:

```
1 ,
2
3   useminPrepare: {
4     foo: {
5       dest: 'dist',
6       src: ['contactus.html','aboutus.html','index.html']
7     },
8     options: {
9       flow: {
10        steps: [
11          { type: 'cssmin' },
12          { type: 'uglify' }
13        ],
14        post: {
15          css: [
16            {
17              name: 'cssmin',
18              createConfig: function (context, block) {
19                var generated = context.options.generated;
20                generated.options = {
21                  keepSpecialComments: 0, rebase: false
22                };
23              }
24            }
25          ]
26        }
27      }
28    },
29    // Concat
30    concat: {
31      options: {
32        separator: ';'
33      },
34      // dist configuration is provided by useminPrepare
35      dist: {}
36    },
37    // Uglify
38    uglify: {
39      // dist configuration is provided by useminPrepare
40      dist: {}
41    },
42    cssmin: {
43      dist: {}
44    },
45    // Filerev
46    filerev: {
47      options: {
48        encoding: 'utf8',
49        algorithm: 'md5',
50        length: 20
51      },
52      release: {
53        // filerev:release hashes(md5) all assets (images, js and css )
54        // in dist directory
55        files: [
56          {
57            src: [
58              'dist/js/*.js',
59              'dist/css/*.css',
60            ]
61          }
62        ]
63      }
64    },
65    // Usemin
66    // Replaces all assets with their revved version in html and css files.
67    // options.assetDirs contains the directories for finding the assets
68    // according to their relative paths
69    usemin: {
70      html: ['dist/contactus.html','dist/aboutus.html','dist/index.html'],
71      options: {
72        assetDirs: ['dist', 'dist/css','dist/js']
73      }
74    },
75    htmlmin: {
76      dist: {} // Task
77      options: { // Target
78        collapseWhitespace: true // Target options
79      },
80      files: { // Dictionary of
81        'dist/index.html': 'dist/index.html', // 'destination':
82        'source':
83        'dist/contactus.html': 'dist/contactus.html',
84        'dist/aboutus.html': 'dist/aboutus.html',
85      }
86    }
87  }
```

- Next, update the jit-grunt configuration as follows, to inform it that useminPrepare task depends on the usemin package:

```
1 = require('jit-grunt')(grunt, {
2   useminPrepare: 'grunt-usemin'
3 });
```

- Next, update the Grunt build task as follows:

```
1   grunt.registerTask('build', [
2     'clean',
3     'copy',
4     'imagemin',
5     'useminPrepare',
6     'concat',
7     'cssmin',
8     'uglify',
9     'filerev',
10    'usemin',
11    'htmlmin'
12  ]);
```

- Now if you run Grunt, it will create a dist folder with the files structured correctly to be distributed to a server to host your website. To do this, type the following at the prompt:

```
1 grunt build
```

Conclusions

In this exercise you have learnt how to configure a Grunt file to perform several tasks. You were able to build a distribution folder for your web project.