

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 *       fonts: {
16 *         files: [
17 *           {
18 *             //for font-awesome
19 *             expand: true,
20 *             dot: true,
21 *             cwd: 'node_modules/font-awesome',
22 *             src: ['fonts/*.*'],
23 *             dest: 'dist'
24 *           }]
25 *         },
26 *       },
27 *     clean: {
28 *       build: {
29 *         src: ['dist/']
30 *       }
31 *     }
32 }
```

- 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 ,
3 *     imagemin: {
4 *       dynamic: {
5 *         files: [
6 *           {
7 *             expand: true,          // Enable dynamic expansion
8 *             cwd: './',            // Src matches are relative to this path
9 *             src: ['img*.{png,jpg,gif}'], // Actual patterns to match
10 *             dest: 'dist/'          // Destination path prefix
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 *             css: ['cssmin'],
12 *             js:['uglify']
13 *           },
14 *           post: {
15 *             css: [
16 *               name: 'cssmin',
17 *               createConfig: function (context, block) {
18 *                 var generated = context.options.generated;
19 *                 generated.options = {
20 *                   keepSpecialComments: 0, rebase: false
21 *                 };
22 *               }
23 *             ]
24 *           }
25 *         }
26 *       },
27 *       concat: {
28 *         options: {
29 *           separator: ';'
30 *         },
31 *         // dist configuration is provided by useminPrepare
32 *         dist: {}
33 *       },
34 *       uglify: {
35 *         // dist configuration is provided by useminPrepare
36 *         dist: {}
37 *       },
38 *       cssmin: {
39 *         dist: {}
40 *       },
41 *       filerev: {
42 *         options: {
43 *           encoding: 'utf8',
44 *           algorithm: 'md5',
45 *           length: 20
46 *         },
47 *         release: {
48 *           // filerev:release hashes(md5) all assets (images, js and css )
49 *           // in dist directory
50 *           files: [
51 *             {
52 *               src: [
53 *                 'dist/js/*.js',
54 *                 'dist/css/*.css',
55 *               ]
56 *             }
57 *           ],
58 *           // Usemin
59 *           // Replaces all assets with their revved version in html and css files.
60 *           // options.assetDirs contains the directories for finding the assets
61 *           // according to their relative paths
62 *           usemin: [
63 *             html: ['dist/contactus.html','dist/aboutus.html','dist/index.html'],
64 *             options: {
65 *               assetsDirs: ['dist', 'dist/css','dist/js']
66 *             }
67 *           ],
68 *           htmlmin: [
69 *             dist: {                                // Task
70 *               options: {                          // Target options
71 *                 collapseWhitespace: true
72 *               },
73 *               files: {                            // Dictionary of
74 *                 'dist/index.html': 'dist/index.html', // 'destination':
75 *                 'source'
76 *                 'dist/contactus.html': 'dist/contactus.html',
77 *                 'dist/aboutus.html': 'dist/aboutus.html',
78 *               }
79 *             }
80 *           ],
81 *           options: {                                // Task
82 *             dist: {                                // Target
83 *               options: {                          // Target options
84 *                 collapseWhitespace: true
85 *               },
86 *               files: {                            // Dictionary of
87 *                 'dist/index.html': 'dist/index.html', // 'destination':
88 *                 'source'
89 *                 'dist/contactus.html': 'dist/contactus.html',
90 *                 'dist/aboutus.html': 'dist/aboutus.html',
91 *               }
92 *             }
93 *           }
94 *         }
95 *       }
96 *     }
97 }
```

- 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.

Complete

[Go to next item](#)