

# Exercise (Instructions): Web Tools: Gulp

## Objectives and Outcomes

In this exercise, you will learn to use Gulp, the task runner. You will install Gulp CLI and install Gulp plugins using NPM. Thereafter you will configure a Gulp file with a set of tasks to build and serve your web project. At the end of this exercise, you will be able to:

- Install Gulp CLI and Gulp plugins in your project
- Configure a Gulp file with a set of tasks to build a web project from a source, and serve the built project using a server.

## Clean node\_modules Folder

- Go to the *node\_modules* folder in *conFusion*, and delete all the folders/files there. We will not be using the Grunt modules existing there for this exercise.

## Initialize package.json File

- Next, update the *package.json* file in the *conFusion* folder with the following content:

```
1 {  
2   "name": "conFusion",  
3   "private": true,  
4   "devDependencies": {  
5  
6   },  
7   "engines": {  
8     "node": ">=0.10.0"  
9   }  
10 }
```

## Installing Gulp

**Note:** You should already have Node and NPM installed on your computer before you proceed further. Also, those using OSX or Linux should use **sudo** while installing **global** packages in node (when you use the **-g** flag).

- At the command prompt, type the following to install Gulp command-line interface (CLI) globally:

```
1 npm install -g gulp
```

This will install the Gulp globally so that you can use it in all projects.

- Next install Gulp to use within your project. To do this, go to the *conFusion* folder and type the following at the prompt:

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

This will install local per-project Gulp to use within your project.

## Install Gulp Plugins

- Install all the Gulp plugins that you will need for this exercise. To do this, type the following at the command prompt:

```
1  npm install jshint gulp-jshint jshint-stylish gulp-imagemin gulp-concat gulp
   -uglify gulp-minify-css gulp-usemin gulp-cache gulp-changed gulp-rev gulp
   -rename gulp-notify browser-sync del --save-dev
2
```

## Creating a Gulp File

- Next you need to create a Gulp file containing the tasks to be run when you use Gulp. To do this, create a file named *gulpfile.js* in the *conFusion* folder.

## Loading Gulp Plugins

- Load in all the Gulp plugins by including the following code in the Gulp file:

```
1  var gulp = require('gulp'),
2      minifycss = require('gulp-minify-css'),
3      jshint = require('gulp-jshint'),
4      stylish = require('jshint-stylish'),
5      uglify = require('gulp-uglify'),
6      usemin = require('gulp-usemin'),
7      imagemin = require('gulp-imagemin'),
8      rename = require('gulp-rename'),
9      concat = require('gulp-concat'),
10     notify = require('gulp-notify'),
11     cache = require('gulp-cache'),
12     changed = require('gulp-changed'),
13     rev = require('gulp-rev'),
14     browserSync = require('browser-sync'),
15     del = require('del');
```

## Adding Gulp Tasks

- Next, we will add the code for the JSHint task, the Clean task and the default task as follows:

```
1  gulp.task('jshint', function() {
2      return gulp.src('app/scripts/**/*.js')
3          .pipe(jshint())
4          .pipe(jshint.reporter(stylish));
5  });
6
7  // Clean
8  gulp.task('clean', function() {
9      return del(['dist']);
10 });
11
12 // Default task
13 gulp.task('default', ['clean'], function() {
14     gulp.start('usemin', 'imagemin', 'copyfonts');
15 });
16
```

- Next, paste in the code for the usemin, imagemin and copyfonts tasks:

```

1  gulp.task('usemin',['jshint'], function () {
2    return gulp.src('./app/menu.html')
3      .pipe(usemin({
4        css:[minifycss(),rev()],
5        js: [uglify(),rev()]
6      }))
7      .pipe(gulp.dest('dist/'));
8  });
9
10 // Images
11 gulp.task('imagemin', function() {
12   return del(['dist/images'], gulp.src('app/images/**/*.')
13     .pipe(cache(imagemin({ optimizationLevel: 3, progressive: true, interlaced:
14       true })))
15     .pipe(gulp.dest('dist/images'))
16     .pipe(notify({ message: 'Images task complete' })));
17 });
18
19 gulp.task('copyfonts', ['clean'], function() {
20   gulp.src('./bower_components/font-awesome/fonts/**/*.{ttf,woff,eof,svg}*')
21     .pipe(gulp.dest('./dist/fonts'));
22   gulp.src('./bower_components/bootstrap/dist/fonts/**/*.{ttf,woff,eof,svg}*')
23     .pipe(gulp.dest('./dist/fonts'));
24 });

```

- Finally, we add the code for the watch and browserSync tasks:

```

1  // Watch
2  gulp.task('watch', ['browser-sync'], function() {
3    // Watch .js files
4    gulp.watch(['app/scripts/**/*.js',app/styles/**/*.css,app/**/*.html'],
5      ['usemin']);
6    // Watch image files
7    gulp.watch('app/images/**/*.', ['imagemin']);
8  });
9
10 gulp.task('browser-sync', ['default'], function () {
11   var files = [
12     'app/**/*.html',
13     'app/styles/**/*.css',
14     'app/images/**/*.png',
15     'app/scripts/**/*.js',
16     'dist/**/*'
17   ];
18
19   browserSync.init(files, {
20     server: {
21       baseDir: "dist",
22       index: "menu.html"
23     }
24   });
25   // Watch any files in dist/, reload on change
26   gulp.watch(['dist/**']).on('change', browserSync.reload);
27 });

```

- Save the Gulp file

## Running the Gulp Tasks

- At the command prompt, if you type *gulp* it will run the default task:

```
1  gulp
```

## Using BrowserSync and Watch

- We configured the BrowserSync and the Watch tasks in the Gulp file. To use them, type the following at the command prompt:

```
1      gulp watch
```

You may need to reload the page in the browser.

- You can edit the *menu.html* file in the app folder and see the watch task and BrowserSync action in reloading the updated page.

## Conclusions

In this exercise, you learnt to use Gulp, install Gulp plugins, configure the gulpfile.js and then use Gulp to automate the web development tasks.

Mark as completed

