✕ Lessons

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:

```
{
  "name": "conFusion",
  "private": true,
  "devDependencies": {

  },
  "engines": {
    "node": ">=0.10.0"
  }
}
```

Installing Gulp

Note: You should already have Node (http://nodejs.org/) 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:

```
     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:

```
     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:

```
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-re
name gulp-notify  browser-sync del --save-dev
```

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:

```
var gulp = require('gulp'),
    minifycss = require('gulp-minify-css'),
    jshint = require('gulp-jshint'),
    stylish = require('jshint-stylish'),
    uglify = require('gulp-uglify'),
    usemin = require('gulp-usemin'),
    imagemin = require('gulp-imagemin'),
    rename = require('gulp-rename'),
    concat = require('gulp-concat'),
    notify = require('gulp-notify'),
    cache = require('gulp-cache'),
    changed = require('gulp-changed'),
    rev = require('gulp-rev'),
    browserSync = require('browser-sync'),
    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:

```
gulp.task('jshint', function() {
  return gulp.src('app/scripts/**/*.js')
  .pipe(jshint())
  .pipe(jshint.reporter(stylish));
});

// Clean
gulp.task('clean', function() {
    return del(['dist']);
});

// Default task
gulp.task('default', ['clean'], function() {
    gulp.start('usemin', 'imagemin','copyfonts');
});
```

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

```
gulp.task('usemin',['jshint'], function () {
  return gulp.src('./app/menu.html')
      .pipe(usemin({
        css:[minifycss(),rev()],
        js: [uglify(),rev()]
      }))
      .pipe(gulp.dest('dist/'));
});

// Images
gulp.task('imagemin', function() {
  return del(['dist/images']), gulp.src('app/images/**/*')
    .pipe(cache(imagemin({ optimizationLevel: 3, progressive: true, interlaced: true })))
    .pipe(gulp.dest('dist/images'))
    .pipe(notify({ message: 'Images task complete' }));
});

gulp.task('copyfonts', ['clean'], function() {
   gulp.src('./bower_components/font-awesome/fonts/**/*.{ttf,woff,eof,svg}*')
   .pipe(gulp.dest('./dist/fonts'));
   gulp.src('./bower_components/bootstrap/dist/fonts/**/*.{ttf,woff,eof,svg}*')
   .pipe(gulp.dest('./dist/fonts'));
});
```

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

```
// Watch
gulp.task('watch', ['browser-sync'], function() {
  // Watch .js files
  gulp.watch('{app/scripts/**/*.js,app/styles/**/*.css,app/**/*.html}', ['usemin']);
      // Watch image files
  gulp.watch('app/images/**/*', ['imagemin']);

});

gulp.task('browser-sync', ['default'], function () {
   var files = [
      'app/**/*.html',
      'app/styles/**/*.css',
      'app/images/**/*.png',
      'app/scripts/**/*.js',
      'dist/**/*'
   ];

   browserSync.init(files, {
      server: {
         baseDir: "dist",
         index: "menu.html"
      }
   });
        // Watch any files in dist/, reload on change
  gulp.watch(['dist/**']).on('change', browserSync.reload);
    });
```

- Save the Gulp file

Running the Gulp Tasks

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

```
     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:

```
     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.