Web Tools: Task Runners: Gulp

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Gulp

- Gulp: Task runner based on code over configuration approach
  - Uses Node.js streams to build complex pipelines, avoiding creation of intermediate files
  - Plugins are simple and do one thing well
  - Tasks are executed with maximum concurrency possible

Installing Gulp

- Installing Gulp
  - Gulp Command Line tool (install globally)

    npm install –g gulp   (use sudo on OSX, Linux)

- Install Grunt locally

  npm install gulp –save-dev

    - need to setup *package.json* file

Gulpfile.js

- Install gulp plugins:

  npm install

  gulp-jshint jshint-stylish

  gulp-imagemin gulp-concat gulp-uglify gulp-minify-css gulp-usemin

  gulp-cache gulp-rev gulp-rename gulp-notify

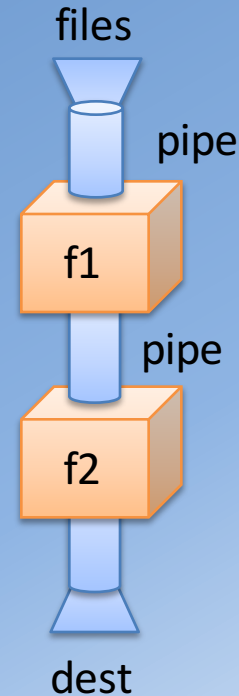 browser-sync del

  --save-dev

Load in the Plugins

```
var gulp = require('gulp'),
    jshint = require('gulp-jshint'),
    stylish = require('jshint-stylish'),
    minifycss = require('gulp-minify-css'),
    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');
```

Can also use gulp-load-plugins to load all plugins specified in package.json

Gulp Streams

- gulp.src(): function that takes file globs and creates a stream of objects that represent the files

- pipe(): Allows the stream to be piped through a function

- gulp.dest(): specifies the destination of the changed files

files

pipe

f1

pipe

f2

dest

JSHint

- Defining the JSHint Task:

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

Imagemin

- Defining the Image Minification Task

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

Cleaning Up

- Makes use of the Node module named del:

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

usemin

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

BrowserSync

```
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);
});
```

Watch

- Watch keeps a watch on the files:

```
gulp.task('watch', ['browser-sync'], function() {

    gulp.watch('{app/scripts/**/*.js,app/styles/**/*.css,app/**/*.html}',
        ['usemin']);

    gulp.watch('app/images/**/*', ['imagemin']);
});
```

Default Task

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

Exercise: Web Tools: Gulp

- 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