014 Gulp - jshint & .jshintrc

Gulp Kata List – Blog Page

[Gulp Kata List on GitHub](https://github.com/robertdunaway/katas-gulp)

# Duration

10 minutes

# Brief

Working with jshint and configuration with .jshintrc.

### For More Information

BING/GOOGLE: “Gulp ”

Book: <http://www.amazon.com/Gulp-Quick-guide-getting-running/dp/151468960X>

# Instructions

Get tutorial folder or the entire katas-angular2 repo.

Open the [before/\*.sln] file and execute the kata.

Feel free to execute this kata multiple times because repetition creates motor memory.

# Github

* Before (start kata with this)
  + https://github.com/robertdunaway/katas-gulp/tree/master/014%20Gulp%20-%20jshint%20%26%20.jshintrc/before
* After
  + https://github.com/robertdunaway/katas-gulp/tree/master/014%20Gulp%20-%20jshint%20%26%20.jshintrc/after

# Kata

In this kata you have a solution with several JavaScript files in the src/js folder. After copying these files to wwwroot execute a JavaScript linter named jshint against the JavaScript files the style the report with jshint-stylish and export the results to a reporter named gulp-jshint-html-reporter.

# Review

First install the required npm plugins.

npm install gulp-jshint --save-dev

npm install jshint-stylish --save-dev

npm install gulp-jshint-html-reporter --save-dev

Reference these plugins.

, jshint = require('gulp-jshint')

, stylish = require('jshint-stylish')

, jshinthtmlreporter = require('gulp-jshint-html-reporter')

Add the jshint task

gulp.task('jshint', function () {

return gulp.src(['./src/\*\*/(\*!.min.js)\*.js'])

.pipe(jshint('.jshintrc'))

.pipe(jshint.reporter(stylish))

.pipe(jshint.reporter('gulp-jshint-html-reporter', { filename: 'jshint-output.html' }))

;

});

Before this can work we need to add the .jshintrc file. This file holds our preferences for jshint. Creating the .jshintrc file is a bit challenging on Windows.

### Creating .jshintrc

This is tricky because Windows doesn’t permit file names with a leading period, “.jshintrc”.

Create a file with any name and then rename it to “.jshintrc”. For some reason Windows doesn’t check an already created file for this.

Now add the following configuration to the .jshintrc file. jshint will use this to drive what is enforced and what is not.

{

"bitwise": true,

"camelcase": true,

"curly": true,

"eqeqeq": true,

"es3": false,

"forin": true,

"freeze": true,

"immed": true,

"indent": 4,

"latedef": "nofunc",

"newcap": true,

"noarg": true,

"noempty": true,

"nonbsp": true,

"nonew": true,

"plusplus": false,

"quotmark": "single",

"undef": true,

"unused": false,

"strict": false,

"maxparams": 10,

"maxdepth": 5,

"maxstatements": 40,

"maxcomplexity": 8,

"maxlen": 160,

"asi": false,

"boss": false,

"debug": false,

"eqnull": true,

"esnext": false,

"evil": false,

"expr": false,

"funcscope": false,

"globalstrict": false,

"iterator": false,

"lastsemic": false,

"laxbreak": false,

"laxcomma": false,

"loopfunc": true,

"maxerr": false,

"moz": false,

"multistr": false,

"notypeof": false,

"proto": false,

"scripturl": false,

"shadow": false,

"sub": true,

"supernew": false,

"validthis": false,

"noyield": false,

"browser": true,

"node": true,

"globals": {

"angular": false,

"$": false

}

}

Execute the jshint task.

Either the jshint task runs to completion, generating and output file, “jshint-output.html”, or you receive the error “ERROR: Can’t parse config file: .jshintrc”.

Addressing this error. If you created your .jshintrc file in Visual Studio then there is a good chance the file was created as UTF-8 with a prefix BOM (byte order mark).

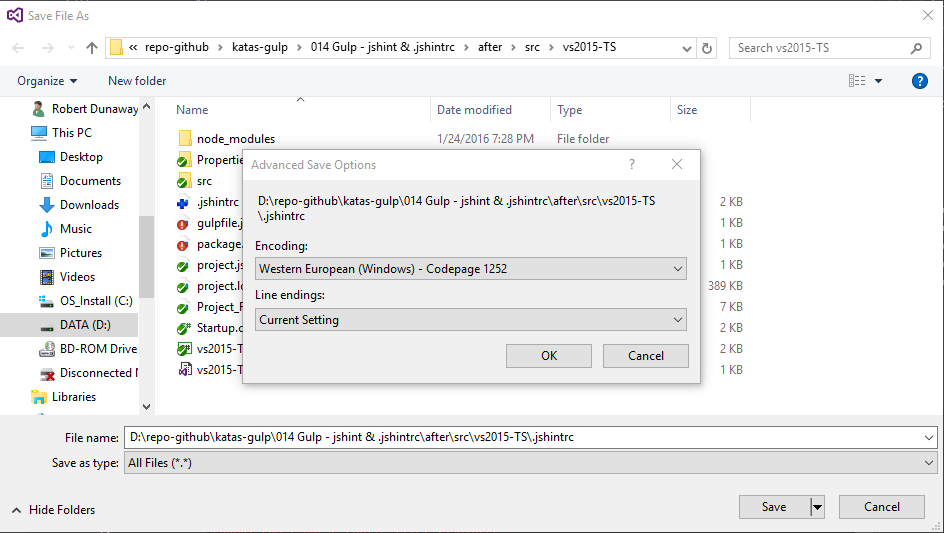
To correct this:

Select the .jshintrc file. Now select File 🡪 Save .jshintrc As

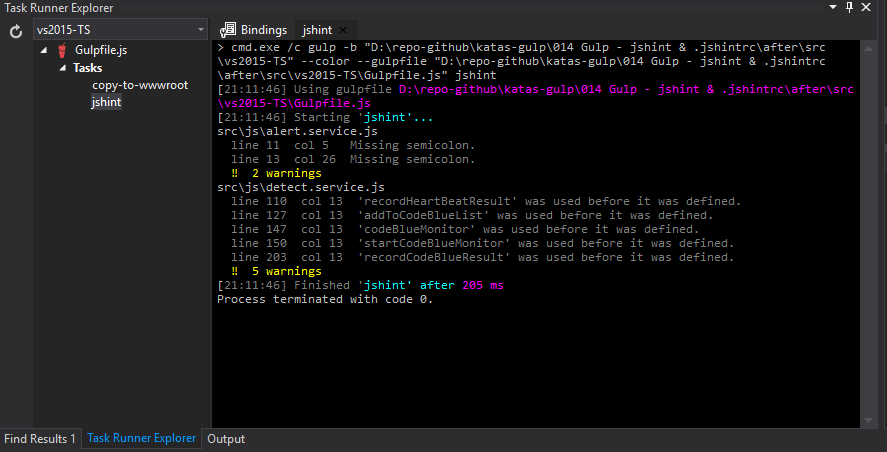
You’ll notice an arrow on the Save button. Press this and select “Save with encoding” and answer “yes” to the dialog box.

Change the “Encoding” to: “Western European (Windows) – Codepage 1252”

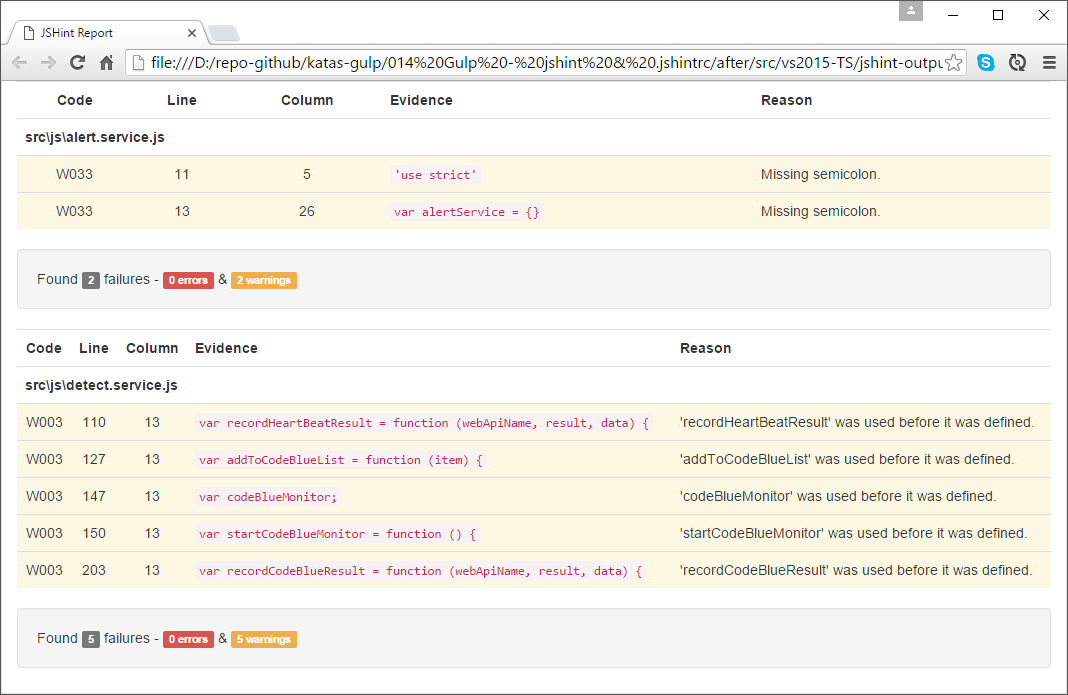
Press OK.



Run the jshint task and you will see this.



A jshint-output.html file is created if there are any problems.



# Next

Take a few minutes and imagine more examples.