

Grunt JSHint Plugin

Olivier Liechti & Simon Oulevay
COMEM Web Services 2016

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

How do I code quality JavaScript?



Enter JSHint



JSHint is a community-driven tool to **detect errors and potential problems** in JavaScript code and to **enforce your team's coding conventions**. It is very flexible so you can easily adjust it to your particular **coding guidelines** and the environment you expect your code to execute in.

<http://jshint.com/about/>

Another tool to install?

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Grunt to the rescue

grunt-contrib-jshint v1.0.0 build passing build passing

Validate files with JSHint

Getting Started

This plugin requires Grunt `>=0.4.0`

If you haven't used [Grunt](#) before, be sure to check out the [Getting Started](#) guide, as it explains how to create a [Gruntfile](#) as well as install and use Grunt plugins. Once you're familiar with that process, you may install this plugin with this command:

```
npm install grunt-contrib-jshint --save-dev
```

Once the plugin has been installed, it may be enabled inside your Gruntfile with this line of JavaScript:

```
grunt.loadNpmTasks('grunt-contrib-jshint');
```

Install grunt-contrib-jshint

Install the plugin:

```
$> cd /path/to/express/project  
$> npm install --save-dev grunt-contrib-jshint
```

Add a **jshint** section to your Gruntfile and **load** the plugin's tasks:

```
module.exports = function(grunt) {  
  grunt.initConfig({  
    // ... previous plugin configs  
    jshint: {  
      all: [ "Gruntfile.js", "app/**/*.js", "public/js/**/*.js", "app.js" ]  
    },  
    // ... next plugin configs  
  });  
  
  grunt.loadNpmTasks("grunt-contrib-jshint");  
}
```

Install grunt-contrib-jshint

Run the plugin with `grunt jshint`:

```
$> grunt jshint
```

```
Running "jshint:all" (jshint) task
```

```
Gruntfile.js
```

```
1 | 'use strict';  
  ^ Use the function form of "use strict".  
3 | var request = require('request');  
  ^ 'require' is not defined.  
7 |   require('time-grunt')(grunt);  
  ^ 'require' is not defined.  
9 |   require('load-grunt-tasks')(grunt);  
  ^ 'require' is not defined.  
5 | module.exports = function (grunt) {  
  ^ 'module' is not defined.  
70 |   setTimeout(function () {  
    ^ 'setTimeout' is not defined.
```

What do you mean
"require is not defined"?
It's Node.js, dude.

```
>> 6 errors in 6 files
```

```
Warning: Task "jshint:all" failed. Use --force to continue.
```

```
Aborted due to warnings.
```

Install grunt-contrib-jshint

JSHint Options

This page's content is sourced from [the JSHint project repository](#). If you spot an error, please [open an issue](#) or (better yet) [make a pull request](#)!

Enforcing options

When set to true, these options will make JSHint produce more warnings about your code.

bitwise	This option prohibits the use of bitwise operators such as <code>^</code> (XOR), <code> </code> (OR) and others. Bitwise operators are very rare in JavaScript programs and quite often <code>&</code> is simply a mistyped <code>&&</code> .
camelcase	Warning This option has been deprecated and will be removed in the next major release of JSHint. JSHint is limiting its scope to issues of code correctness. If you would like to enforce rules relating to code style, check out the JSCS project . This option allows you to force all variable names to use either camelCase style or UPPER_CASE with underscores.

<http://jshint.com/docs/options/>

Configure JSHint

Add the **node** option to the JSHint plugin's task configuration:

```
jshint: {  
  options: {  
    node: true  
  },  
  all: [ "Gruntfile.js", "app/**/*.js", "public/js/**/*.js", "app.js" ]  
}
```

Run it again:

```
$> grunt jshint  
Running "jshint:all" (jshint) task  
>> 6 files lint free.  
Done, without errors.
```

Yay!

If you still have errors, either you've written bad code,
or there are more options to configure.

Add JSHint to the default task

Add the JSHint task to your default Grunt task if you want JSHint to run every time you make a change:

```
grunt.registerTask('default', [  
  'stylus',  
  'jshint',  
  'develop',  
  'watch'  
]);
```