

master

...

jscover / lib / jscover.js / <> Jump to ▾

saper Turn off console logging for newer JSCover ...

History

2 contributors

112 lines (102 sloc) | 3.13 KB

...

```

1  /*!
2   * jscover - lib/jscover.js
3   * Copyright(c) 2012 fengmk2 <fengmk2@gmail.com>
4   * MIT Licensed
5   */
6
7  "use strict";
8
9  /**
10   * Module dependencies.
11   */
12
13  var debug = require('debug')('jscover');
14  var exec = require('child_process').exec;
15  var path = require('path');
16  var ndir = require('ndir');
17  var fse = require('fs-extra');
18  var fs = require('fs');
19
20  var root = path.dirname(__dirname);
21  var JSCoverPath = process.env.JSCOVER || path.join(root, 'bin', 'JSCover-all.jar');
22  var JSCoverLogging = path.join(root, 'bin', 'logging.properties');
23  var JSCoverCommand = ['java',
24    '-Dfile.encoding=UTF-8',
25    '-Djava.util.logging.config.file="' + JSCoverLogging + '"',
26    '-jar "' + JSCoverPath + '"',
27    '-fs'].join(' ');
28
29  /**
30   * Pedding codes to support nodejs
31   * @type {String}
32   */
33  var PEDDING = "/* ***** automatically generated by jscover - do not edit *****\n";
34  if (typeof _jscoverage === 'undefined') { _jscoverage = {}; }\n
35  /* ***** end - do not edit *****\n";
36
37  module.exports = function jscover(source, target, options, callback) {
38    source = source || '';
39    target = target || '';
40    options = options || {};
41    var tmpName = '__cov__' + Date.now();
42    var tmpdir = process.env.TMPDIR || '/tmp';
43    var tmpTargetDir = path.join(tmpdir, tmpName);
44    var tmpTarget = path.join(tmpTargetDir, path.basename(target));
45
46    var cleanup = function () {
47      fse.removeSync(tmpTargetDir);
48      fse.removeSync(path.join(target, tmpName));
49    };
50
51    var cmd = JSCoverCommand;
52    if (options && options.length > 0) {
53      cmd += ' ' + options.join(' ');
54    }
55    cmd += ' --exclude=node_modules --exclude=.git/ --exclude=.svn/';
56    cmd += ' --exclude="' + tmpTarget + '" --exclude="' + target + '"';
57    cmd += ' "' + source + '" "' + tmpTarget + '"';
58    debug(cmd);
59    var child = exec(cmd, function (err, stdout, stderr) {
60      var output = '';
61      if (stdout) {
62        output += stdout;
63      }
64      if (stderr) {
65        output += stderr;
66        if (!err) {
67          err = new Error(stderr.trim());
68        }
69        err.name = 'JSCover' + err.name;
70      }
71      if (err) {
72        debug(err, stderr);
73        cleanup();
74        return callback(err, output);
75      }
76

```

```

77     var success = !stdout && !stderr;
78     if (!success) {
79         return callback(null, output);
80     }
81     ndir.walk(tmpTarget, function onDir(dirpath, items) {
82         var todir = dirpath.replace(tmpTarget, target);
83         fse.mkdirpSync(todir);
84         for (var i = 0; i < items.length; i++) {
85             var info = items[i];
86             var from = info[0];
87             var name = path.basename(from);
88             if (name === '.git' || name === '.svn' || name.indexOf('jscoverage') === 0) {
89                 continue;
90             }
91             var to = path.join(todir, path.basename(from));
92             if (info[1].isDirectory()) {
93                 fse.mkdirpSync(to);
94             } else if (info[1].isFile()) {
95                 var content = fs.readFileSync(from);
96                 if (path.extname(to).toLowerCase() === '.js') {
97                     content = PEDDING + content.toString();
98                 }
99                 fs.writeFileSync(to, content);
100             }
101         }
102     }, function end() {
103         cleanup();
104         callback();
105     }, function error(err, errPath) {
106         cleanup();
107         console.error('%s error: %s', errPath, err);
108         callback(err);
109     });
110
111 });
112 };

```