ViewVC

Diff of /app/gsunit-test/src/test-refactor.js

🐾 Parent Directory | 📰 Revision Log | 📄 Patch

	revision 1.5, Thu Feb 18 19:59:36 2021 UTC	revision 1.6, Thu Feb 18 20:04:39 2021 UTC
#	**Line 1**	**Line 1**
1	/*	/*
2	$Header$	$Header$
3	BEFORE	AFTER
4		
5		wc test-refactor-after.js test-refactor-before.js
6		lines words chars
7		127 539 4821 test-refactor-before.js
8		141 574 5177 test-refactor-after.js
9		+11% +6% +7%
10	*/	*/
11	'use strict';	'use strict';
12		
#	**Line 86**	**Line 92**
92	} // addTestFolder	} // addTestFolder
93		
94	_walkStructure(pArray, pFolder, pFolderName) {	_walkStructure(pArray, pFolder, pFolderName) {
95	for (let tEl of pArray) {	for (let tEl of pArray)
96	if (Array.isArray(tEl)) {	if (Array.isArray(tEl))
97	this._walkStructure(this, tEl, pFolder, pFolderName);	this._walkStructure(tEl, pFolder, pFolderName);
98	} else {	else
99	if (tEl.name === '')	({ pFolderName, pFolder } = this._processElement(tEl, pFolderName, pFolder));
100	throw new Error('Internal Error: missing name.');	} // _walkStructure
101	if (tEl.type == 'file') {	
102	console.info('Create file: "' + tEl.name + '" in "' + pFolderName + '"');	_processElement(pEl, pFolderName, pFolder) {
103	if (tEl.parent != undefined && tEl.parent !== pFolderName)	elParentMatches(pEl, pFolderName);
104	throw new SyntaxError('Bad structure. Expected: "' + tEl.parent + '"');	if (pEl.type == 'file')
105	pFolder.createFile(tEl.name, 'content');	return ({ pFolderName, pFolder } = this._createFile(pEl, pFolderName, pFolder));
106	} else if (tEl.type == 'folder') {	if (pEl.type == 'folder')
107	console.info('Create folder: "' + tEl.name + '" in "' + pFolderName +	return ({ pFolderName, pFolder } = this._createFolder(pEl,

```
          "");                                                      pFolderName, pFolder));
          if (tEl.parent != undefined && tEl.parent !== pFolderName)
            throw new SyntaxError('Bad structure. Expected: "' + tEl.parent + '"');
          pFolder = pFolder.createFolder(tEl.name);
          pFolderName = tEl.name;
        } else {
108       throw new SyntaxError('Invalid type.');                   throw new SyntaxError('Invalid type.');
109     }                                                        }
110
111                                                            _createFolder(pEl, pFolderName, pFolder) {
112                                                              console.info('Create folder: "' + pEl.name + '" in "' + pFolderName + '"');
113                                                              pFolder = pFolder.createFolder(pEl.name);
114                                                              pFolderName = pEl.name;
115                                                              return { pFolderName, pFolder };
116                                                            }
117
118                                                            _createFile(pEl, pFolderName, pFolder) {
119                                                              console.info('Create file: "' + pEl.name + '" in "' + pFolderName + '"');
120                                                              pFolder.createFile(pEl.name, 'content');
121                                                              return { pFolderName, pFolder };
122       }                                                        }
      };
      } // _walkStructure
123
124   delTestFolder() {                                          delTestFolder() {
125     if (!this.exists) {                                        if (!this.exists) {
#     Line 125                                                 Line 136
136     this.ss.toast('Moved folder ' + this.name + ' to trash.', 'Notice', 30);     this.ss.toast('Moved folder ' + this.name + ' to trash.', 'Notice', 30);
137     } // delTestFolder                                         } // delTestFolder
138   } // TestSetup                                              } // TestSetup
139
140                                                            function elParentMatches(pEl, pFolderName) {
141                                                              if (pEl.name === '')
142                                                                throw new Error('Internal Error: missing name.');
143                                                              if (pEl.type === '')
144                                                                throw new Error('Internal Error: missing type.');
145                                                              if (pEl.parent != undefined && pEl.parent !== pFolderName)
146                                                                throw new SyntaxError('Bad structure. Expected: "' + pEl.parent + '"');
147                                                            }
```

Colored Diff ⌄ | Show   Legend:

Removed from v.1.5	
changed lines	
	Added in v.1.6