

aba04a8e1d

...

node-getobject / lib / getobject.js / &lt;&gt; Jump to

Tyler Kellen initial commit

History

0 contributors

60 lines (49 sloc) 1.46 KB

...

```
1  /*
2  * getobject
3  * https://github.com/cowboy/node-getobject
4  *
5  * Copyright (c) 2013 "Cowboy" Ben Alman
6  * Licensed under the MIT license.
7  */
8
9  'use strict';
10
11  var getobject = module.exports = {};
12
13  // Split strings on dot, but only if dot isn't preceded by a backslash. Since
14  // JavaScript doesn't support lookbehinds, use a placeholder for "\", split
15  // on dot, then replace the placeholder character with a dot.
16  function getParts(str) {
17     return str.replace(/\\/\\.g, '\uffff').split('.').map(function(s) {
18         return s.replace(/\uffff/g, '.');
19     });
20 }
21
22 // Get the value of a deeply-nested property exist in an object.
23 getobject.get = function(obj, parts, create) {
24     if (typeof parts === 'string') {
25         parts = getParts(parts);
26     }
27
28     var part;
29     while (typeof obj === 'object' && obj && parts.length) {
30         part = parts.shift();
31         if (!(part in obj) && create) {
32             obj[part] = {};
33         }
34         obj = obj[part];
35     }
36
37     return obj;
38 };
39
40 // Set a deeply-nested property in an object, creating intermediary objects
41 // as we go.
42 getobject.set = function(obj, parts, value) {
43     parts = getParts(parts);
44
45     var prop = parts.pop();
46     obj = getobject.get(obj, parts, true);
47     if (obj && typeof obj === 'object') {
48         return (obj[prop] = value);
49     }
50 };
51
52 // Does a deeply-nested property exist in an object?
53 getobject.exists = function(obj, parts) {
54     parts = getParts(parts);
55
56     var prop = parts.pop();
57     obj = getobject.get(obj, parts);
58
59     return typeof obj === 'object' && obj && prop in obj;
60 };
```