

2692. Make Object Immutable Premium

Solved

Medium Hint

Write a function that takes an object `obj` and returns a new **immutable** version of this object.

An **immutable** object is an object that can't be altered and will throw an error if any attempt is made to alter it.

There are three types of error messages that can be produced from this new object.

- Attempting to modify a key on the object will result in this error message: `"Error Modifying: ${key}"`.
- Attempting to modify an index on an array will result in this error message: `"Error Modifying Index: ${index}"`.
- Attempting to call a method that mutates an array will result in this error message: `"Error Calling Method: ${methodName}"`. You may assume the only methods that can mutate an array are `['pop', 'push', 'shift', 'unshift', 'splice', 'sort', 'reverse']`.

`obj` is a valid JSON object or array, meaning it is the output of `JSON.parse()`.

Note that a string literal should be thrown, not an `Error`.

Example 1:

Input:
`obj = {
 "x": 5
}`
`fn = (obj) => {
 obj.x = 5;
 return obj.x;
}`
Output: `{"value": null, "error": "Error Modifying: x"}`
Explanation: Attempting to modify a key on an object results in a thrown error. Note that it doesn't matter that the value was set to the same value as it was before.

Example 2:

Input:
`obj = [1, 2, 3]
fn = (arr) => {
 arr[1] = {};
 return arr[2];
}`
Output: `{"value": null, "error": "Error Modifying Index: 1"}`
Explanation: Attempting to modify an array results in a thrown error.

Example 3:

Input:
`obj = {
 "arr": [1, 2, 3]
}`
`fn = (obj) => {
 obj.arr.push(4);
 return 42;
}`
Output: `{"value": null, "error": "Error Calling Method: push"}`
Explanation: Calling a method that can result in a mutation results in a thrown error.

Example 4:

Input:
`obj = {
 "x": 2,
 "y": 2
}`
`fn = (obj) => {
 return Object.keys(obj);
}`
Output: `{"value": ["x", "y"], "error": null}`
Explanation: No mutations were attempted so the function returns as normal.

Constraints:

- `obj` is a valid JSON object or array
- `2 <= JSON.stringify(obj).length <= 105`

Seen this question in a real interview before? 1/5

Yes No

Accepted 342 Submissions 538 Acceptance Rate 63.6%

Hint 1

Hint 2

Hint 3

Similar Questions

Discussion (0)

Copyright © 2024 LeetCode All rights reserved

15 0 0 0 0 0

</> Code

JavaScript Auto

```
1 /**
2  * @param {Object|Array} obj
3  * @return {Object|Array} immutable obj
4  */
5 var makeImmutable = function(obj) {
6
7     return new Proxy(obj, {
8         set(target, prop, value, receiver){
9             if (Array.isArray(target) && typeof prop === 'string'){
10                 const asNumber = Number(prop);
11                 if (asNumber !== undefined){
12                     throw `Error Modifying Index: ${prop}`;
13                 }
14             }
15             throw `Error Modifying: ${prop}`;
16         },
17         get(target, prop){
18             if (Array.isArray(target) && (
19                 prop==='pop' || prop==='push' || prop==='shift' || prop==='unshift' ||
20                 prop==='splice' || prop==='sort' || prop==='reverse'
21             )){
22                 return (...arguments) => {throw `Error Calling Method: ${prop}`};
23             }
24             if (typeof obj[prop] === 'object' && obj[prop] !== null){
25                 // call recursively
26                 return makeImmutable(obj[prop]);
27             }
28         }
29     });
30 }
```

Saved

Ln 23, Col 14

Testcase Test Result

Accepted Runtime: 60 ms

Case 1 Case 2 Case 3 Case 4

Input

`{"x":5}``(obj) => { obj.x = 5; return obj.x; }`