2700. Differences Between Two Objects Premium

Solved

Medium | Companies | Hint

Write a function that accepts two deeply nested objects or arrays `obj1` and `obj2` and returns a new object representing their differences.

The function should compare the properties of the two objects and identify any changes. The returned object should only contains keys where the value is different from `obj1` to `obj2`.

For each changed key, the value should be represented as an array `[obj1 value, obj2 value]`. Keys that exist in one object but not in the other should not be included in the returned object. The end result should be a deeply nested object where each leaf value is a difference array.

When comparing two arrays, the indices of the arrays are considered to be their keys.

You may assume that both objects are the output of `JSON.parse`.

Example 1:

```
Input:
obj1 = {}
obj2 = {
  "a": 1,
  "b": 2
}
Output: {}
Explanation: There were no modifications made to obj1. New keys "a" and "b" appear in obj2, but keys that are added or removed should be ignored.
```

Example 2:

```
Input:
obj1 = {
  "a": 1,
  "v": 3,
  "x": [],
  "z": {
    "a": null
  }
}
obj2 = {
  "a": 2,
  "v": 4,
  "x": [],
  "z": {
    "a": 2
  }
}
Output:
{
  "a": [1, 2],
  "v": [3, 4],
  "z": {
    "a": [null, 2]
  }
}
Explanation: The keys "a", "v", and "z" all had changes applied. "a" was changed from 1 to 2. "v" was changed from 3 to 4. "z" had a change applied to a child object. "z.a" was changed from null to 2.
```

Example 3:

```
Input:
obj1 = {
  "a": 5,
  "v": 6,
  "z": [1, 2, 4, [2, 5, 7]]
}
obj2 = {
  "a": 5,
  "v": 7,
  "z": [1, 2, 3, [1]]
}
Output:
{
  "v": [6, 7],
  "z": {
    "2": [4, 3],
    "3": {
      "0": [2, 1]
    }
  }
}
Explanation: In obj1 and obj2, the keys "v" and "z" have different assigned values. "a" is ignored because the value is unchanged. In the key "z", there is a nested array. Arrays are treated like objects where the indices are keys. There were two alterations to the the array: z[2] and z[3][0]. z[0] and z[1] were unchanged and thus not included. z[3][1] and z[3][2] were removed and thus not included.
```

Example 4:

```
Input:
obj1 = {
  "a": {"b": 1},
}
obj2 = {
  "a": [5],
}
Output:
{
  "a": [{"b": 1}, [5]]
}
Explanation: The key "a" exists in both objects. Since the two associated values have different types, they are placed in the difference array.
```

Example 5:

```
Input:
obj1 = {
  "a": [1, 2, {}],
  "b": false
}
obj2 = {
  "b": false,
  "a": [1, 2, {}]
}
Output:
{}
Explanation: Apart from a different ordering of keys, the two objects are identical so an empty object is returned.
```

Code

TypeScript • Auto

```typescript
type JSONValue = null | boolean | number | string | JSONValue[] | { [key: string]: JSONValue };
type Obj = Record<string, JSONValue> | Array<JSONValue>

function objDiff(obj1: any, obj2: any): any {

    // support variables
    let t1 = Array.isArray(obj1) ? 'array' : typeof obj1,
        t2 = Array.isArray(obj2) ? 'array' : typeof obj2;
    // base case - different types
    if (t1 !== t2) return [obj1, obj2];
    // both are objects
    if (t1 === 'object' || t1 === 'array') return Object.fromEntries(
        Object.entries(obj1).reduce((acc, [key, el1]) => {
            const el2 = obj2[key];
            // el2 does not exist or el1 === el2
            if (el2 === undefined || el1 === el2) return acc;
            // el1 and el2 are different
            const tmp = objDiff(el1, el2);
```

Saved | Ln 9, Col 35

Testcase | Test Result

Accepted Runtime: 74 ms

Case 1 | Case 2 | Case 3 | Case 4 | Case 5

Input

```
{}
```

```
{"a": 1, "b": 2}
```

Output

```
{}
```

Expected

Constraints:

- `obj1` and `obj2` are valid JSON objects or arrays
- `2 <= JSON.stringify(obj1).length <= 10^4`
- `2 <= JSON.stringify(obj2).length <= 10^4`

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

Yes No

Accepted 6.8K | Submissions 8.7K | Acceptance Rate 77.8%

Companies

Hint 1

Hint 2

Hint 3

Similar Questions

Discussion (8)

143 8

```
{}
```