Changing the case of JavaScript object keys

Asked 4 years, 10 months ago Active 1 year, 5 months ago Viewed 5k times

4

I have following object.

```
var obj = [{
  Address1: "dd",
  Address2: "qww",
  BankAccNo: "44",
  BankBranchCode: "44",
  BloodGrp: "A+"
},
{
  Address1: "dd",
  Address2: "qww",
  BankAccNo: "44",
  BankBranchCode: "44",
  BloodGrp: "A+"
}];
```

How can I make all of the keys uppercase?

I want to be able to access values like this : - `obj[0].ADDRESS1`

javascript

edited Oct 29 '14 at 11:39
nbrooks
16.5k 4 41 57

asked Oct 29 '14 at 11:29
Anup
5,164 13 46 105

1 Why not just access them as they are? – nnnnnn Oct 29 '14 at 11:33

@nnnnnn maybe because of consistency with some DB name fields, just guessing – A. Wolff Oct 29 '14 at 11:40

Yes...I am matching this keys with values stored in DB, which are in Uppercase.! – Anup Oct 29 '14 at 11:43

8

✓

```
obj = obj.map( function( item ){
    for(var key in item){
        var upper = key.toUpperCase();
        // check if it already wasn't uppercase
        if( upper !== key ){
            item[ upper ] = item[key];
            delete item[key];
        }
    }
    return item;
});
```

http://jsfiddle.net/07xortqy/

edited Oct 29 '14 at 12:00

answered Oct 29 '14 at 11:34
pawel
23.4k 5 40 44

This won't work if the key is already upper case. And I'm not sure how it will work on every browser because you change an object while looping on it with a `for in` – Sebastien C. Oct 29 '14 at 11:50

@SebastienC. thanks, added a safeguard for existing uppercase keys. – pawel Oct 29 '14 at 12:01

6

1. Loop over all the properties in the object (with `for in`)
2. Use `.toUpperCase()` to get the uppercase version of the property name
3. Copy the value from the original property to the uppercase version
4. `delete` the original property

answered Oct 29 '14 at 11:31
Quentin
686k 79 929 1083

1

For anyone looking for a solution working with objects, arrays, and nested objects or arrays:

```
// rename function depending on your needs
const capitalizeKeys = (obj) => {
  const isObject = o => Object.prototype.toString.apply(o) === '[object Object]'
  const isArray = o => Object.prototype.toString.apply(o) === '[object Array]'

  let transformedObj = isArray(obj) ? [] : {}

  for (let key in obj) {
    // replace the following with any transform function
    const transformedKey = key.replace(/^\w/, (c, _) => c.toUpperCase())

    if (isObject(obj[key]) || isArray(obj[key])) {
      transformedObj[transformedKey] = capitalizeKeys(obj[key])
    } else {
      transformedObj[transformedKey] = obj[key]
    }
  }
  return transformedObj
}

const t = {
  test1: 'hello',
  test2: {
    aa: 0,
    bb: '1',
    cc: [ 3, '4', 'world']
  },
  test3: [{
      aa: 5,
      bb: '6'
    }, {
      cc: [ 'hello', 'world', 7 ]
    }
  ]
}

console.log(JSON.stringify(capitalizeKeys(t)))
```

Run code snippet   Expand snippet

answered Apr 4 '18 at 15:36
Lorine
56 5

0

```
$.each(obj, function(i, parent) {
  $.each(parent, function(key, record) {
    parent[ key.toUpperCase() ] = record[key]; //rename key
    delete parent[key]; //delete old key
  });
});
```

answered Oct 29 '14 at 11:51
Zainul Abdin
810 4 10

Please add explanation to your answer. Just code lines do not constitute a good answer. – user1803551 Oct 29 '14 at 12:45

first iterate array then iterate keys, inside second loop rename key and delete old one. – Zainul Abdin Oct 29 '14 at 13:10

1 Edit your question instead of adding a comment. – user1803551 Oct 29 '14 at 13:20

0

```
let obj = [
{ Address1: "dd",Address2: 'qww',BankAccNo: 44,BankBranchCode: 44,BloodGrp: 'A+' },
{ Address1: "dd",Address2: 'qww',BankAccNo: 44,BankBranchCode: 44,BloodGrp: 'A+' }
];

const uppercaseKeys = (elem) => {
  let newObject = {}

  Object.keys(elem).reduce( (acc, key, allKeys) => {
    acc[key.toUpperCase()] = elem[key]
    delete elem[key]
    return acc
  }, elem)
```

```
obj.forEach( o => uppercaseKeys )
console.log(obj)
```

answered Feb 13 '18 at 8:30

kitimenpolku

1,261 3 25 40

-2

You could just loop through them and add new entries?

```
for (index in obj) {
  for (key in obj[index]) {
     obj[index][key.toUpperCase()] = obj[key];
  }
}
```

answered Oct 29 '14 at 11:34

Evan Knowles

5,986 1 25 60

Using `for..in` to iterate over arrays is frowned upon. And now you have both upper- an lower-case keys, which may or may not be what OP wanted. – pawel Oct 29 '14 at 11:36