

Using lodash push to an array only if value doesn't exist?

Asked 3 years, 4 months ago Active 5 months ago Viewed 64k times



I'm trying to make an array that if a value doesn't exist then it is added but however if the value is there I would like to remove that value from the array as well.

50



Feels like Lodash should be able to do something like this.



I'm interested in your best practises suggestions.

3

Also it is worth pointing out that I am using Angular.js

*** Update ***

```
if (!_.includes(scope.index, val)) {  
    scope.index.push(val);  
} else {  
    _.remove(scope.index, val);  
}
```

javascript

arrays

angularjs

lodash

edited Aug 15 '17 at 12:36



alexmac

13.8k

6

37

56

asked May 6 '16 at 13:33



Max Lynn

753

2

13

30

sample input and output would help. Plus what you have tried. – [Davin Tryon](#) May 6 '16 at 13:36

In modern JavaScript environments, it would be much better to use a Set than an array. – [Pointy](#) May 6 '16 at 13:37

8 Answers

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

60



```
var s = new Set();

// Adding alues
s.add('hello');
s.add('world');
s.add('hello'); // already exists

// Removing values
s.delete('world');

var array = Array.from(s);
```

Or if you want to keep using regular Arrays

```
function add(array, value) {
  if (array.indexOf(value) === -1) {
    array.push(value);
  }
}

function remove(array, value) {
  var index = array.indexOf(value);
  if (index !== -1) {
    array.splice(index, 1);
  }
}
```

Using vanilla JS over Lodash is a good practice. It removes a dependency, forces you to understand your code, and often is more performant.

edited Sep 15 '16 at 19:31

answered May 6 '16 at 13:38



floribon


17.3k 4 46 63

What is the support for using set? – Max Lynn May 6 '16 at 13:45

It's been around for a little while. IE11 though. developer.mozilla.org/fr/docs/Web/JavaScript/Reference/... – floribon May 6 '16 at 13:57

- 1 Looks like this like of code within the add() function is wrong: **array.indexOf(value) !== -1**, should be **array.indexOf(value) === -1** instead. Current code logic is: if item already exists - add. – Pjotr Sep 15 '16 at 12:50

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

the code in question is a bottleneck. Optimizing non-bottlenecks does not improve your app's performance and can actually have a negative overall impact since every minute spent optimizing a non-bottleneck is a minute NOT spent on something that matters. In general, it is better to use already-written & debugged code (when *profiling* does not direct you otherwise). – [Troy](#) Aug 31 '17 at 18:14 

You can use `_.union`

52

```
_.union(scope.index, [val]);
```

answered Jan 11 '17 at 9:24



[Afzal Hossain](#)

1,860 14 11

1 Bravo, this is awesome. – [crazy_phage](#) Dec 17 '18 at 9:54

Perhaps `_.pull()` can help:

5

```
var _ = require('lodash');

function knock(arr, val){
  if(arr.length === _.pull(arr, val).length){
    arr.push(val);
  }
  return arr;
}
```

Mutates the existing array, removes duplicates as well:

```
> var arr = [1,2,3,4,4,5];

> knock(arr,4);
[ 1, 2, 3, 5 ]

> knock(arr,6);
[ 1, 2, 3, 5, 6, 1 ]
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

edited May 6 '16 at 14:06

answered May 6 '16 at 14:00



S.D.

25.7k

1

66

115

Use `includes` function to check that item is exists in array, and `remove` to delete existing item.

2

```
function addOrRemove(arr, val) {  
  if (!_.includes(arr, val)) {  
    arr.push(val);  
  } else {  
    _.remove(arr, item => item === val);  
  }  
  console.log(arr);  
}
```

```
var arr = [1, 2, 3];  
addOrRemove(arr, 1); // arr = [2, 3]  
addOrRemove(arr, 4); // arr = [2, 3, 4]  
addOrRemove(arr, 2); // arr = [3, 4]
```

```
<script src="https://raw.githubusercontent.com/lodash/lodash/4.11.2/dist/lodash.min.js">  
</script>
```

Run code snippet

[Expand snippet](#)

edited May 6 '16 at 18:44

answered May 6 '16 at 13:37



alexmac

13.8k

6

37

56

I'm liking this answer alex but I tried it and it doesn't seem to be working for me. — [Max Lynn](#) May 6 '16 at 14:01

It seems to add very well but doesn't remove the value if the function is triggered again. I've updated my question with a code snippet — [Max Lynn](#) May 6 '16 at 14:02

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



This single liner should do the job. If element to be inserted does not exist, it inserts the element and returns the length of the resulting array. If element exists in the array it deletes the element and returns the deleted element in a separate array.

```
var arr = [1,2,3,4,5],
    aod = (a,e,i=0) => !!~(i = a.indexOf(e)) ? a.splice(i,1) : a.push(e);

document.write("<pre>" + JSON.stringify(aod(arr,6)) + JSON.stringify(arr) + "</pre>");
document.write("<pre>" + JSON.stringify(aod(arr,6)) + JSON.stringify(arr) + "</pre>");
```

[Run code snippet](#)
[Expand snippet](#)

Well actually i hate push since it returns the resulting array length value which is most of the time useless. I would prefer to have a reference to the resulting array to be returned so that you can chain the functions. Accordingly a simple way to achieve it is;

```
var arr = [1,2,3,4,5],
    aod = (a,e,i=0) => !!~(i = a.indexOf(e)) ? a.splice(i,1) : (a.push(e),a);

document.write("<pre>" + JSON.stringify(aod(arr,6)) + JSON.stringify(arr) + "</pre>");
document.write("<pre>" + JSON.stringify(aod(arr,6)) + JSON.stringify(arr) + "</pre>");
```

[Run code snippet](#)
[Expand snippet](#)

So now this is reasonably chainable.

edited May 6 '16 at 15:42

answered May 6 '16 at 14:20



Redu

13.9k

2

26

40



0

```
const addUniq = (array, value) => array.includes(value)
  ? array.length
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

answered Jun 14 '18 at 20:07



Rich Howell

1

The simplest way to do this is use `_.isEmpty` and `_.remove` Lodash functions:

```
0  if (_.isEmpty(_.remove(array, value)) {  
    array.push(value);  
  }
```

After remove function will be return removed values or an empty array, and if return an empty array then we will add a new value.

answered Sep 27 '18 at 11:09



Ilya Lagoshny

45 7

In this case you can use 'concat' to push and 'uniq' to validate unique values:

example:

```
var ar = [1, 2, 3, 1, 5, 2, 4]  
ar = _.uniq(_.concat(ar, 9))  
//[1, 2, 3, 5, 4, 9]
```

e.g.: <https://codepen.io/dieterich/pen/xeZNJY>

ref.: <https://lodash.com/docs/4.17.11#uniq>

edited Apr 4 at 13:03

answered Sep 4 '18 at 20:20



Darlan Dieterich

488 8 19

concat doesn't care about unique values – [mjoyce91](#) Mar 8 at 18:47

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

