How to remove elements/nodes from angular.js array

I am trying to remove elements from the array `$scope.items` so that items are removed in the view `ng-repeat="item in items"`

Just for demonstrative purposes here is some code:

```
for(i=0;i<$scope.items.length;i++){
    if($scope.items[i].name == 'ted'){
      $scope.items.shift();
    }
}
```

I want to remove the 1st element from the view if there is the name ted right? It works fine, but the view reloads all the elements. Because all the array keys have shifted. This is creating unnecessary lag in the mobile app I am creating..

Anyone have an solutions to this problem?

angularjs scope angularjs-ng-repeat

edited Nov 17 '15 at 15:51 Dan Beaulieu **9,137** 9 58 87

asked Aug 18 '13 at 19:50 TheNickyYo **1,131** 4 12 28

I've used splice successfully to modify an array that is used in ng-repeat with no weird side effects. – BoxerBucks Aug 18 '13 at 19:57

Looks like the items is an array of array, or you can't call items[i].shift(); – zsong Aug 18 '13 at 20:12

Hi, thanks for the replies. Sorry, there was a typos in the code of my question, i've just updated it. – TheNickyYo Aug 18 '13 at 20:21

Then why you remove the first element from the array rather than the element at the position i? – zsong Aug 18 '13 at 20:25

3 rows will solve you the problem, just add $filter in controller – Maxim Shoustin Aug 18 '13 at 21:25

12 Answers

There is no rocket science in deleting items from array. To delete items from any array you need to use `splice` : `$scope.items.splice(index, 1);` . Here is an example:

HTML

```
<!DOCTYPE html>
<html data-ng-app="demo">
  <head>
    <script data-require="angular.js@1.1.5" data-semver="1.1.5"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.1.5/angular.js"></script>
    <link rel="stylesheet" href="style.css" />
    <script src="script.js"></script>
  </head>
  <body>
    <div data-ng-controller="DemoController">
      <ul>
        <li data-ng-repeat="item in items">
          {{item}}
          <button data-ng-click="removeItem($index)">Remove</button>
        </li>
      </ul>
      <input data-ng-model="newItem"><button data-ng-click="addItem(newItem)">Add</button>
    </div>
  </body>
</html>
```

JavaScript

```
"use strict";

var demo = angular.module("demo", []);

function DemoController($scope){
  $scope.items = [
    "potatoes",
    "tomatoes",
    "flour",
    "sugar",
    "salt"
  ];

  $scope.addItem = function(item){
    $scope.items.push(item);
    $scope.newItem = null;
  }

  $scope.removeItem = function(index){
    $scope.items.splice(index, 1);
  }
}
```

answered Aug 18 '13 at 20:25

madhead
8,622 5 49 90

What is the meaning of "1" in (index, 1) – ShibinRagh Sep 15 '15 at 11:31

@ShibinRagh, read the doc: developer.mozilla.org/en/docs/Web/JavaScript/Reference/… – madhead Sep 15 '15 at 13:00

18 Beware that `$index` does not necessarily match `$scope.items` indexes. Especially if you use `orderBy` or `filter`. I think using `$index` is not scalable and may be dangerous. – Antoine Pinsard Sep 18 '15 at 9:45

@AntoinePinsard what do you recommend instead? – stackPusher Feb 2 at 4:26

1 @stackPusher, if I remember correctly I used bpaul's solution stackoverflow.com/a/22160628/1529346 – Antoine Pinsard Feb 2 at 9:07

For anyone returning to this question. The correct "Angular Way" to remove items from an array is with $filter. Just inject $filter into your controller and do the following:

```
$scope.items = $filter('filter')($scope.items, {name: '!ted'})
```

You don't need to load any additional libraries or resort to Javascript primitives.

answered Mar 4 '14 at 0:42

bpaul
1,644 1 10 17

Would this work if you have an element with the name "ted" and "teddy", and just want to delete the "ted" element? – Jervelund Oct 3 '14 at 20:41

1 @Jervelund yes you can by using true at the end: `$filter('filter')($scope.items, {name: '!ted'}, true)` , see docs.angularjs.org/api/ng/filter/filter – Betty St Oct 7 '14 at 12:09

4 I think this answer should probably be the accepted one because of its angular-ness. – MrBoJangles Jan 13 '15 at 23:24

Is it possible to use filter to remove an element containing a specific numeric id? `$filter('filter')($scope.items, {id: 42}, true);` works for finding the element, but I haven't found a way to negate it, `{id: "!42"}` doesn't work since it's a string instead. – Oscar Feb 16 '15 at 8:46

24 This works: `$filter('filter')($scope.items, function(value, index) {return value.id !== 42;});` – Oscar Feb 16 '15 at 10:44

You can use plain javascript - Array.prototype.filter()

```
$scope.items = $scope.items.filter(function(item) {
    return item.name !== 'ted';
});
```

answered Feb 23 '14 at 18:42

Bogdan D
2,072 16 25

Because when you do `shift()` on an array, it changes the length of the array. So the for loop will be messed up. You can loop through *from end to front* to avoid this problem.

Btw, I assume you try to remove the element at the position **i** rather than the first element of the array. (`$scope.items.shift();` in your code will remove the first element of the array)

```
for(var i = $scope.items.length - 1; i >= 0; i--){
    if($scope.items[i].name == 'ted'){
```

```
        $scope.items.splice(i,1);
    }
}
```

edited Aug 18 '13 at 20:34

answered Aug 18 '13 at 20:26

zsong
40.4k 17 113 162

Here is `filter` with Underscore library might help you, we remove item with name "ted"

```
$scope.items = _.filter($scope.items, function(item) {
    return !(item.name == 'ted');
 });
```

edited Mar 3 '14 at 5:49

answered Aug 18 '13 at 20:39

Maxim Shoustin
49.2k 16 149 176

2 This is the "angular" way! Of course it can be improved: `$scope.items = $filter('filter').filter($scope.items, {name: '!ted'})` – bpaul Feb 4 '14 at 0:29

3 Nice improvement from @bpaul but looks like there is an error, it should be: `$scope.items = $filter('filter')($scope.items, {name: '!ted'})` – Eugene Mar 3 '14 at 0:31

1 @Eugene not at all. I used Underscore library, added link. – Maxim Shoustin Mar 3 '14 at 5:50

Thanks @Eugene, that was a typo. I should test code before I post! Also, this should be an answer as it does not need underscore (or any other lib) and is idiomatic angular. I'll add it. – bpaul Mar 4 '14 at 0:31

Just a slight expansion on the 'angular' solution. I wanted to exclude an item based on it's numeric id, so the ! approach doesn't work. The more general solution which should work for { name: 'ted' } or { id: 42 } is:

```
mycollection = $filter('filter')(myCollection, { id: theId }, function (obj, test) {
                                                    return obj !== test; });
```

answered May 22 '14 at 15:33

Mark Farmiloe
188 1 7

My solution to this (which hasn't caused any performance issues):

1. Extend the array object with a method remove (i'm sure you will need it more than just one time):

```
Array.prototype.remove = function(from, to) {
  var rest = this.slice((to || from) + 1 || this.length);
  this.length = from < 0 ? this.length + from : from;
  return this.push.apply(this, rest);
};
```

I'm using it in all of my projects and credits go to John Resig John Resig's Site

2. Using forEach and a basic check:

```
$scope.items.forEach(function(element, index, array){
          if(element.name === 'ted'){
              $scope.items.remove(index);
          }
        });
```

At the end the $digest will be fired in angularjs and my UI is updated immediately without any recognizable lag.

answered Oct 8 '14 at 9:35

Fer To
830 7 16

Is it really safe to remove an element while iterating over the list? – Volte Oct 8 '15 at 4:19

Generally, you should be careful about doing this. I would say using .forEach and not a for loop makes it safe as MDN developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/… says: "forEach() executes the provided callback once for each element present in the array in ascending order. It is not invoked for index properties that have been deleted or are uninitialised (i.e. on sparse arrays)." So deleted items after visit / or before visit are just "ignored" which imho makes it safe for this case. – Fer To Oct 8 '15 at 10:28

I liked the solution provided by @madhead

However the problem I had is that it wouldn't work for a sorted list so instead of passing the index to the delete function I passed the item and then got the index via indexof

e.g.:

```
var index = $scope.items.indexOf(item);
$scope.items.splice(index, 1);
```

An updated version of madheads example is below: link to example

HTML

```
<!DOCTYPE html>
<html data-ng-app="demo">
  <head>
    <script data-require="angular.js@1.1.5" data-semver="1.1.5"
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.1.5/angular.js"></script>
    <link rel="stylesheet" href="style.css" />
    <script src="script.js"></script>
  </head>
  <body>
    <div data-ng-controller="DemoController">
      <ul>
        <li data-ng-repeat="item in items|orderBy:'toString()'">
          {{item}}
          <button data-ng-click="removeItem(item)">Remove</button>
        </li>
      </ul>
      <input data-ng-model="newItem"><button data-ng-click="addItem(newItem)">Add</button>
    </div>
  </body>
</html>
```

JavaScript

```
"use strict";

var demo = angular.module("demo", []);

function DemoController($scope){
  $scope.items = [
    "potatoes",
    "tomatoes",
    "flour",
    "sugar",
    "salt"
  ];

  $scope.addItem = function(item){
    $scope.items.push(item);
    $scope.newItem = null;
  }

  $scope.removeItem = function(item){
    var index = $scope.items.indexOf(item);
    $scope.items.splice(index, 1);
  }
}
```

answered Nov 5 '16 at 1:41

TrtlBoy
309 4 9

If you have any function associated to list ,when you make the splice function, the association is deleted too. My solution:

```
$scope.remove = function() {
    var oldList = $scope.items;
    $scope.items = [];

    angular.forEach(oldList, function(x) {
        if (! x.done) $scope.items.push( { [ DATA OF EACH ITEM USING oldList(x) ] });
    });
};
```

The list param is named **items**. The param **x.done** indicate if the item will be deleted. Hope help you. Greetings.

answered Sep 4 '14 at 12:28

Drako
374 2 10

Using the indexOf function was not cutting it on my collection of REST resources.

I had to create a function that retrieves the array index of a resource sitting in a collection of resources:

```
factory.getResourceIndex = function(resources, resource) {
  var index = -1;
  for (var i = 0; i < resources.length; i++) {
    if (resources[i].id == resource.id) {
```

```
      index = i;
    }
  }
  return index;
}

$scope.unassignedTeams.splice(CommonService.getResourceIndex($scope.unassignedTeams,
data), 1);
```

answered Sep 17 '14 at 7:43

Stephane
1,300 6 35 56

My solution was quite straight forward

```
app.controller('TaskController', function($scope) {
 $scope.items = tasks;

    $scope.addTask = function(task) {
        task.created = Date.now();
        $scope.items.push(task);
        console.log($scope.items);
    };

    $scope.removeItem = function(item) {
        // item is the index value which is obtained using $index in ng-repeat
        $scope.items.splice(item, 1);
    }
});
```

answered Mar 23 '16 at 6:13

Bastin Robin
596 7 16

My items have unique id's. I am deleting one by filtering the model with angulars `$filter` service:

```
var myModel = [{id:12345, ...},{},{},...,{}];
...
// working within the item
function doSthWithItem(item){
...
  myModel = $filter('filter')(myModel, function(value, index)
    {return value.id !== item.id;}
  );
}
```

As id you could also use the $$hashKey property of your model items:

```
$$hashKey:"object:91"
```

answered Mar 2 at 9:19

Ruwen
533 4 9
