

8 Features of ng-repeat

Ben Garvey | September 2, 2015

THE DATAPOINT ► ENGINEERING ► 8 FEATURES OF NG-REPEAT

Share 

ng-repeat is my favorite of all the core **AngularJS** directives. A huge amount of front end programming time goes into displaying lists of things in helpful ways and ng-repeat helps make that easier, but it's also filled to the brim with gotchas. Even worse, when something doesn't work it fails silently and you're left to figure out why your list isn't sorting or filtering correctly.

Because of these gotchas, I put together a presentation last year for our Angular-Philly meetup titled, **14 Slides on ng-repeat**. I frequently refer back to it to check my syntax so I figured it might be a useful blog post. Here are 8 features of ng-repeat!

Plain ng-repeat

Model

```
1 | $scope.names = ['Ben', 'Nate', 'Austin', 'Yiannis', 'Matt M', 'Matt B']
```

Template

```
1 | <ul>
2 |   <li ng-repeat="name in names">{{name}}</li>
3 | </ul>
```

View

Ben

Nate

Austin

Yiannis

Matt M

Matt B

Very simple. We have an array of strings and we're creating an unordered list and populating each list item with a string from the names array. The data is bound to `$scope` and Angular keeps the model and the view consistent.

track by

Model

```
1 | $scope.names = ['Ben', 'Nate', 'Austin', 'Yiannis', 'Matt', 'Matt'];
```

Template

```
1 <ul>
2   <li ng-repeat="name in names track by $index">{{name}}</li>
3 </ul>
```

View

Ben

Nate

Austin

Yiannis

Matt

Matt

Around Angular 1.3 they started requiring **track by** for any list that may include duplicate values. Using track by also speeds up list changes significantly. If you don't use **track by** in this case, you get the error: **Duplicates in a repeater are not allowed**

filter

Model

```
1 $scope.names = ['Ben', 'Nate', 'Austin', 'Yiannis', 'Matt M', 'Matt B']
```

Template

```
1 <ul>
2   <input type="text" ng-model="searchBox">
3   <li ng-repeat="name in names | filter:searchBox track by $index">
4     {{name}}
5   </li>
6 </ul>
```

View

Matt

Matt M

Matt B

By binding the filter to the searchBox, you can create an easy way to find the list item you want. A common mistake is to put the filter after the track by like this.

name in names track by \$index | filter:searchBox. Don't do it!

object

Model

```
1 | $scope.statuses = {"Ben": true, "Nate": true, "Austin": false};
```

Template

```
1 | <ul>
2 |   <li ng-repeat="(key, value) in statuses">
3 |     {{key}} {{value}}
4 |   </li>
5 | </ul>
```

View

Ben true

Nate true

Austin false

Yes, you can iterate over objects with ng-repeat, but don't do it! You'll regret it. All the cool stuff like sorting and filtering either

doesn't work or gets way more complicated. If you're iterating over an object with ng-repeat, you've already screwed up.

array of objects

Model

```
1 $scope.users = [  
2   {"name": "Ben", "status": true},  
3   {"name": "Nate", "status": true},  
4   {"name": "Austin", "status": false}  
5 ];
```

Template

```
1 <ul>  
2   <input type="text" ng-model="searchBox"/>  
3   <li ng-repeat="user in users | filter:searchBox:user.name">  
4     {{user.name}} {{user.status}}  
5   </li>  
6 </ul>
```

View

Ben true

stacked filters

Model

```
1 $scope.users = [  
2   {"name": "Ben", "status": true},  
3   {"name": "Nate", "status": true},  
4   {"name": "Austin", "status": false}  
5 ];
```

Template

```
1 <ul>
2   <input type="text" ng-model="searchBox"/>
3   <li ng-repeat="user in users | filter:searchBox:user.name | filter:tr
4     {{user.name}} {{user.status}}
5   </li>
6 </ul>
```

View

Ben true

Nate true

nested loops

Model

```
1 $scope.statuses = ["away", "ready", "busy"];
2 $scope.users = [
3   {"name": "Ben", "status": "ready"},
4   {"name": "Nate", "status": "ready"},
5   {"name": "Austin", "status": "busy"}
6 ];
```

Template

```
1 <ul>
2   <li><input type="text" ng-model="searchBox"/></li>
3   <li ng-repeat="user in users | filter:searchBox:user.name">
4     {{user.name}}
5     <select ng-model="user.status" ng-options="status for sta
6   </li>
7 </ul>
```

View

Nate ready ▾

Ben ready ▾

Austin busy ▾

order by

Model

```
1 $scope.statuses = ["away", "ready", "busy"];
2 $scope.users = [
3   {"name": "Ben", "status": "ready"},
4   {"name": "Nate", "status": "ready"},
5   {"name": "Austin", "status": "busy"},
6 ];
```

Template

```
1 <ul>
2   <li><input type="text" ng-model="searchBox"/></li>
3   <li ng-repeat="user in users | filter:searchBox:user.name | o
4     {{user.name}}
5     <select ng-model="user.status" ng-options="status for sta
6   </li>
7 </ul>
```

View

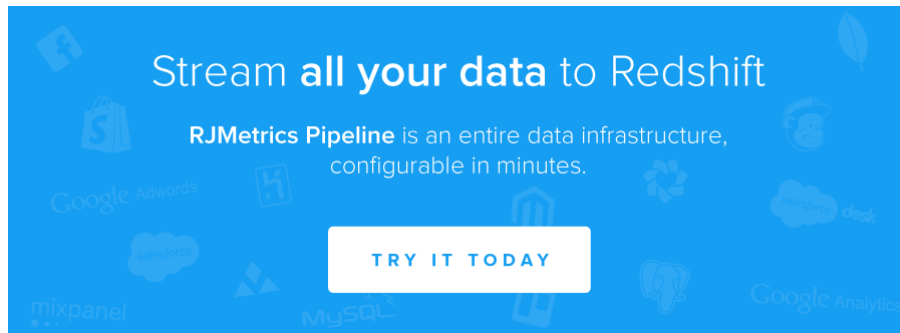
Austin busy ▾

Ben ready ▾

Nate ready ▾

Don't forget the orderBy value has to be a string. Don't do this!

`ng-repeat="user in users | filter:searchBox:user.name | orderBy:name"`

A blue rectangular banner with white text and various data source logos. The text reads: "Stream all your data to Redshift" in large font, followed by "RJMetrics Pipeline is an entire data infrastructure, configurable in minutes." in smaller font. At the bottom center is a white button with the text "TRY IT TODAY". Logos for Google AdWords, Google Analytics, mixpanel, and MySQL are visible in the background.

Stream **all your data** to Redshift

RJMetrics Pipeline is an entire data infrastructure, configurable in minutes.

TRY IT TODAY

ABOUT THE AUTHOR



Ben Garvey

Ben Garvey is a software engineer at RJMetrics and you can follow him on twitter @bengarvey

Follow @bengarvey 2,269 followers

YOU MIGHT ALSO ENJOY

The Five Indicators of Breakout Ecommerce Growth

Janessa Lantz | February 4, 2015

Introducing: The RJMetrics Analysis Library

Jim Roddy | February 9, 2017

What You Missed: The Growth-Hacking Skill No One's Talking About

Anna Kegler | January 23, 2015

Understanding your Magento Data Series: The Balancing Act of Returns

Chris Schmid | February 16, 2017

Customer Success and the Evolution of SaaS Metrics: Churn, LTV, and CAC

Karan Mehandru | October 15, 2014

Some Downright Fancy SQL Visualizations

Akash Agrawal | February 23, 2017

8 Comments

RJMetrics Ecommerce Blog

 Login ▾

 Recommend 3

 Tweet

 Share

Sort by Best ▾



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Name



Hafiz Muhammad Muazzam • 4 months ago

I have following code ``<th style="min-width:10px;" ng-repeat="(key,value) in keys"> {{key}} {{value}} </th>``

But when I want put filter as

`ng-repeat="(key,value) in keys | filter;searchbox track by $index"`

the values of `{{key}}` and `{{value}}` gets blank even I don't put anything inside searchbox

^ | v • Reply • Share ›



Padmapriya Vishnuvardhan • 2 years ago

Hi,

I have key,value in list of ng-repeat, where key is holding some special character like \$advance, so the ng-repeat is not looping through that particular object. Is there any way to handle this situation?

^ | v • Reply • Share ›



srikanth reddy • 2 years ago

As you said , "ng-repeat with objects filters will not work & will work for array of objects" is this point correct which I stated from your article ? Because I can see in array of objects you had provided the filter as `<li ng-repeat=""user" in="" users="" |="" filter:searchbox:user.name=""="">`. Please confirm it ?

^ | v • Reply • Share ›



eric • 2 years ago

i noticed some ng-repeat using this "product in results = products" .. any idea why they did that? I still couldn't figure out the reasons as inside the loop elements, they are using `{{product.name}}` .. how does the results come into this context? Thanks.

^ | v • Reply • Share ›



Ben Garvey ➔ eric • 2 years ago

eric,

I'd have to see the full context to be sure, but it looks like they are using the expression to set the value of results in addition to using it as an iterator. It would make more sense to me if it was something where results was the value and they were setting the value of products for readability.

If you change it to ``product in products`` does the code still function?

35 ^ | v • Reply • Share ›

35 ^ | v • Reply • Share ›



Bala Abhinav • 3 years ago

You have mentioned that using key,value in ng-repeat is bad. But I have a situation where there is a json object with atleast 200 key, value pairs. What would you suggest is a better way to display them and make sure that there is two way data binding?

^ | v • Reply • Share ›



Ben Garvey ➔ Bala Abhinav • 3 years ago

If you have a simple situation, repeating over object keys is fine. The problem is that you lose access to some of the great features of ng-repeat. I have been in your situation before and to get around it, I have created a new array in the parent directive from the object keys.

^ | v • Reply • Share ›

DATA JOURNALISM

APPLICATIONS OF DATA

THINKING IN DATA

ABOUT US

CONNECT WITH US

