

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

[Support us](#) [.Net Basics](#) [C#](#) [SQL](#) [ASP.NET](#) [ADO.NET](#) [MVC](#) [Slides](#) [C# Programs](#) [Subscribe](#) [Buy DVD](#)

Angular ngFor trackBy

Suggested Videos

[Part 14 - Angular2 event binding](#) | [Text](#) | [Slides](#)

[Part 15 - Two way data binding in angular 2](#) | [Text](#) | [Slides](#)

[Part 16 - Angular ngFor directive](#) | [Text](#) | [Slides](#)

In this video we will discuss

1. Using **trackBy** with **ngFor** directive
2. How to get the index of an item in a collection
3. Identifying the first and the last elements in a collection
4. Identifying even and odd elements in a collection

Using trackBy with ngFor directive :

- ngFor directive may perform poorly with large lists
- A small change to the list like, adding a new item or removing an existing item may trigger a cascade of DOM manipulations

For example, consider this code in **employeeList.component.ts**

PRAGIM
TECHNOLOGIES
Training + Placements

Best software training and placements in marathahalli, bangalore. For further details please call 09945699393.

Complete Tutorials

[JavaScript tutorial](#)

[Bootstrap tutorial](#)

[Angular tutorial for beginners](#)

[Angular 5 Tutorial for beginners](#)

Important Videos

[The Gift of Education](#)

[Web application for your business](#)

The **constructor()** initialises the employees property with 4 employee objects
getEmployees() method returns another list of 5 employee objects (The 4 existing employees plus a new employee object)

```
import { Component } from '@angular/core';

@Component({
  selector: 'list-employee',
  templateUrl: 'app/employee/employeeList.component.html',
  styleUrls: ['app/employee/employeeList.component.css']
})
export class EmployeeListComponent {
  employees: any[];

  constructor() {
    this.employees = [
      {
        code: 'emp101', name: 'Tom', gender: 'Male',
        annualSalary: 5500, dateOfBirth: '25/6/1988'
      },
      {
        code: 'emp102', name: 'Alex', gender: 'Male',
        annualSalary: 5700.95, dateOfBirth: '9/6/1982'
      },
      {
        code: 'emp103', name: 'Mike', gender: 'Male',
        annualSalary: 5900, dateOfBirth: '12/8/1979'
      },
      {
        code: 'emp104', name: 'Mary', gender: 'Female',
        annualSalary: 6500.826, dateOfBirth: '14/10/1980'
      }
    ];
  }

  getEmployees(): void {
    this.employees = [
      {
        code: 'emp101', name: 'Tom', gender: 'Male',
        annualSalary: 5500, dateOfBirth: '25/6/1988'
      },
      {
        code: 'emp102', name: 'Alex', gender: 'Male',
        annualSalary: 5700.95, dateOfBirth: '9/6/1982'
```

How to become .NET developer

Resources available to help you

Dot Net Video Tutorials

ASP.NET Core Tutorial

Angular 6 Tutorial

Angular CRUD Tutorial

Angular CLI Tutorial

Angular 2 Tutorial

Design Patterns

SOLID Principles

ASP.NET Web API

Bootstrap

AngularJS Tutorial

jQuery Tutorial

JavaScript with ASP.NET Tutorial

JavaScript Tutorial

Charts Tutorial

LINQ

LINQ to SQL

LINQ to XML

Entity Framework

```

    },
    {
      code: 'emp103', name: 'Mike', gender: 'Male',
      annualSalary: 5900, dateOfBirth: '12/8/1979'
    },
    {
      code: 'emp104', name: 'Mary', gender: 'Female',
      annualSalary: 6500.826, dateOfBirth: '14/10/1980'
    },
    {
      code: 'emp105', name: 'Nancy', gender: 'Female',
      annualSalary: 6700.826, dateOfBirth: '15/12/1982'
    }
  ];
}
}

```

Now look at this code in **employeeList.component.html**

```

<table>
  <thead>
    <tr>
      <th>Code</th>
      <th>Name</th>
      <th>Gender</th>
      <th>Annual Salary</th>
      <th>Date of Birth</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor='let employee of employees'>
      <td>{{employee.code}}</td>
      <td>{{employee.name}}</td>
      <td>{{employee.gender}}</td>
      <td>{{employee.annualSalary}}</td>
      <td>{{employee.dateOfBirth}}</td>
    </tr>
    <tr *ngIf='!employees || employees.length==0'>
      <td colspan="5">
        No employees to display
      </td>
    </tr>
  </tbody>
</table>
<br />
<button (click)='getEmployees()'>Refresh Employees</button>

```

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Dot Net Interview Questions

Slides

Entity Framework

WCF

ASP.NET Web Services

Dot Net Basics

C#

SQL Server

ADO.NET

ASP.NET

At the moment we are not using **trackBy** with **ngFor** directive

1. When the page initially loads we see the 4 employees
2. When we click **"Refresh Employees"** button we see the fifth employee as well
3. It looks like it just added the additional row for the fifth employee. That's not true, it effectively teared down all the `<tr>` and `<td>` elements of all the employees and recreated them.
4. To confirm this launch browser developer tools by pressing F12.
5. Click on the "Elements" tab and expand the `<table>` and then `<tbody>` elements
6. At this point click the **"Refresh Employees"** button and you will notice all the `<tr>` elements are briefly highlighted indicating they are teared down and recreated.

[GridView](#)[ASP.NET MVC](#)[Visual Studio Tips and Tricks](#)

Java Video Tutorials

[Part 1 : Video | Text | Slides](#)[Part 2 : Video | Text | Slides](#)[Part 3 : Video | Text | Slides](#)

Interview Questions

[C#](#)[SQL Server](#)[Written Test](#)

Code	Name	Gender	Annual Salary	Date of Birth
emp101	Tom	Male	5500	25/6/1988
emp102	Alex	Male	5700.95	9/6/1982
emp103	Mike	Male	5900	12/8/1979
emp104	Mary	Female	6500.826	14/10/1980
emp105	Nancy	Female	6700.826	15/12/1982

Refresh Employees

This happens because Angular by default keeps track of objects using the object

references. When we click **"Refresh Employees"** button we get different object references and as a result Angular has no choice but to tear down all the old DOM elements and insert the new DOM elements.

Angular can avoid this churn with **trackBy**. The **trackBy** function takes the index and the current item as arguments and returns the unique identifier by which that item should be tracked. In our case we are tracking by Employee code. Add this method to **employeeList.component.ts**.

```
trackByEmpCode(index: number, employee: any): string {  
    return employee.code;  
}
```

Make the following change in **employeeList.component.html** : Notice along with **ngFor** we also specified **trackBy**

```
<tr *ngFor="let employee of employees; trackBy:trackByEmpCode">
```

At this point, run the application and launch developer tools. When you click **"Refresh Employees"** first time, only the the row of the fifth employee is highlighted indicating only that **<tr>** element is added. On subsequent clicks, nothing is highlighted meaning none of the **<tr>** elements are teared down or added as the employees collection has not changed. Even now we get different object references when we click **"Refresh Employees"** button, but as Angular is now tracking employee objects using the employee code instead of object references, the respective DOM elements are not affected.

How to get the index of an item in a collection : Notice in the example below, we are using the index property of the **ngFor** directive to store the index in a template input variable "i". The variable is then used in the **<td>** element where we want to display the index. We used the **let** keyword to create the template input variable "i".

The index of an element is extremely useful when creating the HTML elements dynamically. We will discuss creating HTML elements dynamically in our upcoming videos.

```
<table>
```

```

<thead>
  <tr>
    <th>Code</th>
    <th>Name</th>
    <th>Gender</th>
    <th>Annual Salary</th>
    <th>Date of Birth</th>
    <th>Index</th>
  </tr>
</thead>
<tbody>
  <tr *ngFor='let employee of employees; let i=index'>
    <td>{{employee.code}}</td>
    <td>{{employee.name}}</td>
    <td>{{employee.gender}}</td>
    <td>{{employee.annualSalary}}</td>
    <td>{{employee.dateOfBirth}}</td>
    <td>{{i}}</td>
  </tr>
  <tr *ngIf='!employees || employees.length==0'>
    <td colspan="5">
      No employees to display
    </td>
  </tr>
</tbody>
</table>

```

Notice the index of the element is displayed into the last `<td>` element

Code	Name	Gender	Annual Salary	Date of Birth	Index
emp101	Tom	Male	5500	25/6/1988	0
emp102	Alex	Male	5700.95	9/6/1982	1
emp103	Mike	Male	5900	12/8/1979	2
emp104	Mary	Female	6500.826	14/10/1980	3

Identifying the first and the last element in a collection : Use the **first** and **last** properties of the **ngFor** directive to find if an element is the first or last element respectively.

Modify the code in **employeeList.component.html** as shown below.

```
<table>
  <thead>
    <tr>
      <th>Code</th>
      <th>Name</th>
      <th>Gender</th>
      <th>Annual Salary</th>
      <th>Date of Birth</th>
      <th>Is First</th>
      <th>Is Last</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor='let employee of employees; let isFirst = first; let isLast = last'>
      <td>{{employee.code}}</td>
      <td>{{employee.name}}</td>
      <td>{{employee.gender}}</td>
      <td>{{employee.annualSalary}}</td>
      <td>{{employee.dateOfBirth}}</td>
      <td>{{isFirst}}</td>
      <td>{{isLast}}</td>
    </tr>
    <tr *ngIf='!employees || employees.length==0'>
      <td colspan="5">
        No employees to display
      </td>
    </tr>
  </tbody>
</table>
<br />
<button (click)='getEmployees()'>Refresh Employees</button>
```

The output is shown below

Code	Name	Gender	Annual Salary	Date of Birth	Is First	Is Last
emp101	Tom	Male	5500	25/6/1988	true	false
emp102	Alex	Male	5700.95	9/6/1982	false	false
emp103	Mike	Male	5900	12/8/1979	false	false
emp104	Mary	Female	6500.826	14/10/1980	false	true

Identifying even and odd element in a collection : This is similar to identifying first and last element in a collection. Instead of using first and last properties, use **even** and **odd** properties.

Here is the code in **employeeList.component.html**

```

<table>
  <thead>
    <tr>
      <th>Code</th>
      <th>Name</th>
      <th>Gender</th>
      <th>Annual Salary</th>
      <th>Date of Birth</th>
      <th>Is Even</th>
      <th>Is Odd</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor='let employee of employees; let isEven = even; let isOdd = odd'>
      <td>{{employee.code}}</td>
      <td>{{employee.name}}</td>
      <td>{{employee.gender}}</td>
      <td>{{employee.annualSalary}}</td>
      <td>{{employee.dateOfBirth}}</td>
      <td>{{isEven}}</td>
      <td>{{isOdd}}</td>
    </tr>
    <tr *ngIf='!employees || employees.length==0'>
      <td colspan="5">
        No employees to display
      </td>
    </tr>
  </tbody>
</table>

```

```
</tr>  
</tbody>  
</table>  
<br />  
<button (click)='getEmployees()'>Refresh Employees</button>
```

The output is shown below

Code	Name	Gender	Annual Salary	Date of Birth	Is Even	Is Odd
emp101	Tom	Male	5500	25/6/1988	true	false
emp102	Alex	Male	5700.95	9/6/1982	false	true
emp103	Mike	Male	5900	12/8/1979	true	false
emp104	Mary	Female	6500.826	14/10/1980	false	true

WWW.PRAGIMTECH.COM

**CLICK HERE FOR THE FULL
ANGULAR 2 TUTORIAL PLAYLIST**

facebook.com/pragimtech | twitter.com/kudvenkat

6 comments:

Subhendu July 20, 2017 at 12:01 PM



Hello sir. Thanks a lot for your videos. Can you please explain how the input variable for trackByEmpCode method is getting passed in the tr ngFor ?

Reply



Mahendra P. Verma September 7, 2017 at 4:27 PM

This comment has been removed by the author.

Reply



B.nandanam March 18, 2018 at 4:08 AM

To the trackBy method no need to pass index, i think. Current item is only needed. Please correct me if I am wrong.

Reply



Unknown April 11, 2018 at 12:57 AM

IN the above example , how is the first and third row employee "Even" , rather it should be "Odd" right ? Vice-Versa is true when you see the "Odd" ones should be "Even" in this case .

Reply

Replies

Anonymous July 25, 2018 at 5:05 AM

the index value 0 indicates even and third row index is 2 so it displays first and third row as even

Anonymous September 14, 2018 at 3:54 PM

I think it has something to do with the index. Add the index Index column along with the Even and Odd columns in the table. Not sure how zero index is even though, but... meh! :)

Reply

Enter your comment...



Comment as:

toilati123vn



Publish

Preview

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by Blogger.