Angular Training
(Intermediate to Advanced)

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Trainer Introduction

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Trainer Introduction

- 20+ Years of Experience
- Educational Background : MCA
- Experience with Software Development and Training on different technologies.
- As SDET2 : VS 2008, IE8, Bing, MS Dynamics
- MCTS certified
- Key Skills : UI Technologies, MEAN/MERN, DotNet Fullstack
- Given valued added trainings for corporate clients.
- Recent Clients :   Ness Tech, Deloitte, Cognizant,Godrej, Pega Systems, FIS Global, Aptean etc..

Training Schedule

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Schedule for Angular Training

Day#	Date	Topic
Day-1	**16-Nov-2023**	**Custom Pipes in Angular**
Day-2	17-Nov-2023	Parent-Child Communication
Day-3	20-Nov-2023	Custom Directives in Angular
Day-4	21-Nov-2023	Working with Reactive Forms
Day-5	22-Nov-2023	Dependency Injection and Services in Angular
Day-6	23-Nov-2023	Http Client – Server calls in Angular
Day-7	24-Nov-2023	Routing and Security in Angular
Day-8	27-Nov-2023	Unit Testing in Angular

Duration : 8 days (2hours per day) ;   2pm to 4 pm;   16th Nov – 27th Nov

Day1
Custom Pipes in Angular

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Index – Day1

- What is CustomPipe?
- PipeTransform Interface
- Custom Pipe : Scalar, Collection Based
- Parameterized Custom Pipes

Recapping Angular

- Angular Framework
- Features
- Angular CLI
- TypeScript
- Components
- Templates

What is pipes in Angular?

- Used to transform the data in the template itself.
- Applied in presentation logic (view) so that we can display formatted results to end-user.
- To convert component data in customized format
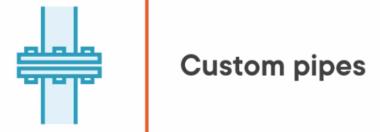- Use Pipe symbol (|) -- to apply

Transforming Data with Pipes

Transform bound properties before display

Built-in pipes: date, number, decimal, percent, currency, json, etc.

Custom pipes

```
{{ sname | uppercase}}
{{item.ename | titlecase}}
{{item.salary | currency}}
```

```
<tr *ngFor="let item of deptsArray">
  <td> {{item.dname |  uppercaase }} </td>
</tr>
```

Custom Pipe

Narasimha

Sr. IT Trainer/Consultant

Custom Pipes in Angular

- Creating pipes for custom data transformations
- To encapsulate transformations that are not provided with the built-in pipes
- Used to develop our own pipes functionality.
- Define require functionality in a separate class with @Pipe decorator.

Steps to Create Custom Pipes

1) Create a Pipe Class and decorate it with the decorator **@Pipe.**

2) Supply a **name** property to be used as template code name.

3) Register your Pipe in the module under declarations.

4) Finally, implement **PipeTransform** and write transformation logic.

5) Use/Apply your pipe in the template (HTML) using '**|**'

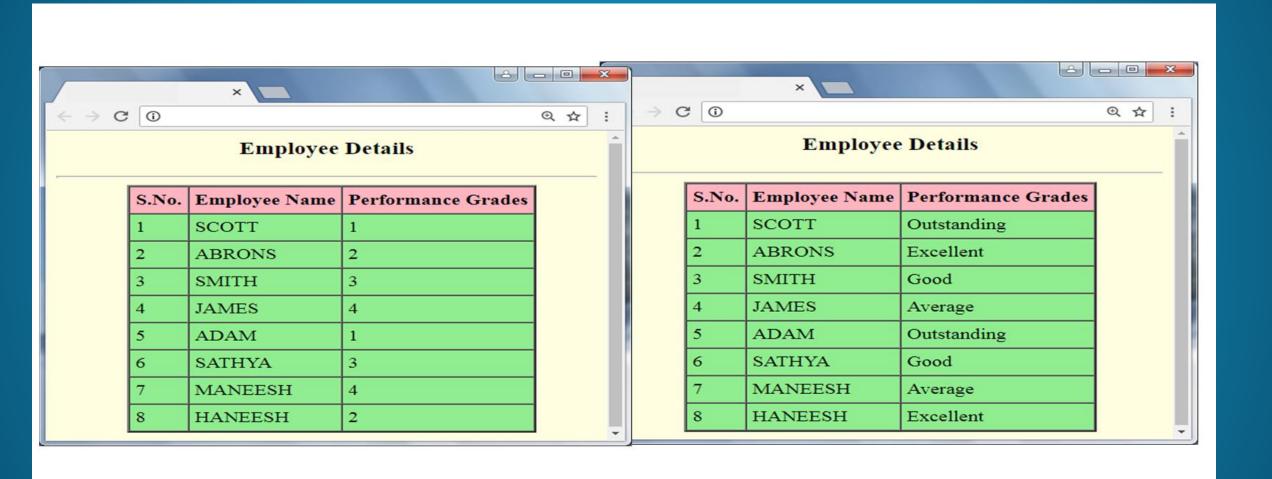6) Also, add Custom arguments if you want to.

Steps to Create Custom Pipes

1. **Define custom pipe class**

    - **@Pipe() decorator**

    - **PipeTransform interface**

        - **transform() method**

2. **Register pipe in app module**

    - **app.module.ts**

    - **declarations**

3. **Use custom pipe in template**
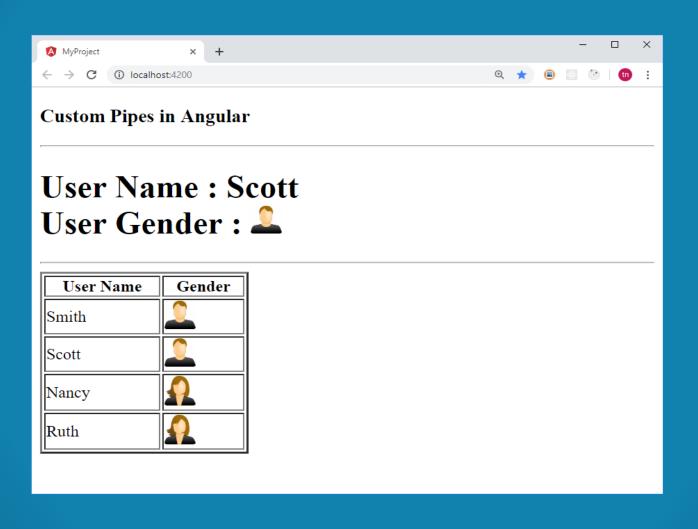
    variable | pipename

Custom Pipes in Angular

```typescript
import {Pipe, PipeTransform} from "@angular/core";

@Pipe({ name: "pipename" })
export class PipeClass implements PipeTransform
{
    public transform() : ReturnType
        {
                // Logic
        }
}
```
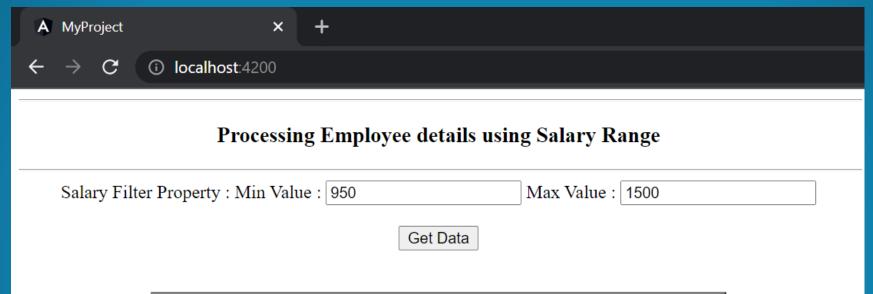
Assignment-1   (Custom Pipes)

Assignment-2　(Custom Pipes)

Assignment-2 (Custom Pipes)

Processing Employee details using Salary Range

Salary Filter Property : Min Value : `950`   Max Value : `1500`

Get Data

Emp Number	Emp Name	Emp Job	Emp Salary	Emp Deptno
6521	WARD	ANALYST	1250	30
7654	MARTIN	SALESMAN	1250	30
7844	TURNER	SALESMAN	1500	30
7876	ADAMS	CLERK	1100	20
8900	JAMES	CLERK	950	30
8844	TURNER	SALESMAN	1500	30
8876	ADAMS	CLERK	1100	20
9900	JAMES	CLERK	950	30

Parametrized Custom Pipes

Narasimha

Sr. IT Trainer/Consultant

Custom Pipes with parameters

emp-list.component.html

```
<tr *ngFor="let item of empsArray | filterpipe: 10 ">
<tr *ngFor="let item of empsArray | filterpipe: 20 ">
```

filter.pipe.ts

```
export class FilterPipe implements PipeTransform {
 transform(inputData:any[], dno:number): any[]    {

                   .....
  }
}
```