Angular Training (Intermediate to Advanced)

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..

Schedule for React JS Training

Day1: Custom Pipes in Angular

Day2 : Parent-Child Communication

Day3 : Custom Directives in Angular

Day4 : Working with Reactive Forms

Day5 : Dependency Injection and Services in Angular

Day6: RxJS Library – Observables

Day7: Http Client – Server calls in Angular

Day8: Routing and Security in Angular

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

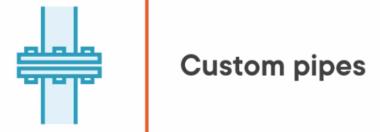
Transforming Data with Pipes



Transform bound properties before display



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



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

```
    {(item.dname | uppercaase )}
```



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 your pipe in the HTML using '|'
- 6) Also, add Custom arguments if you want to.

Steps to Create Custom Pipes

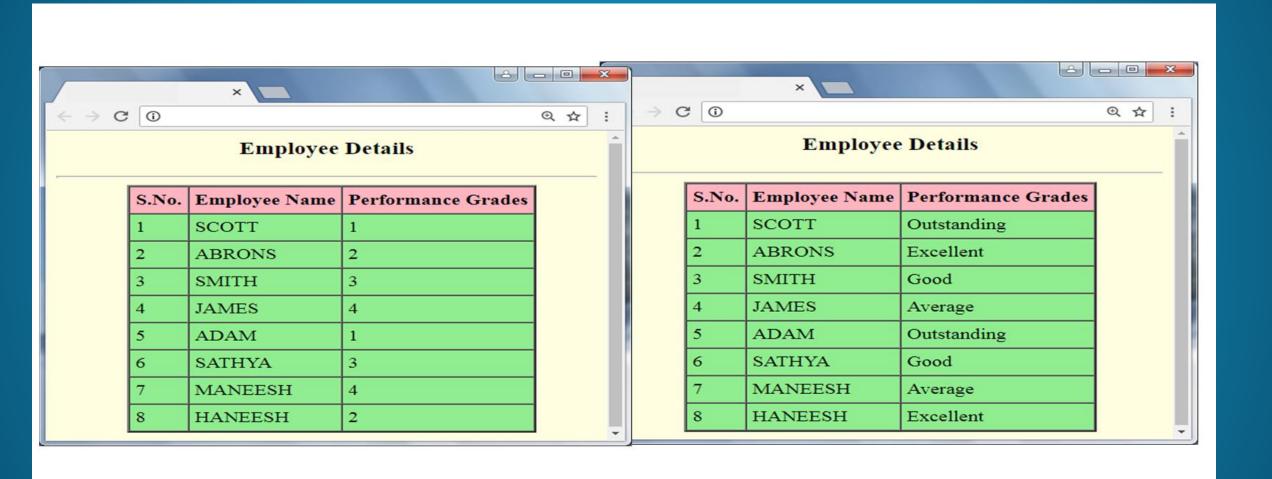
- 1. Define custom pipe class
 - @Pipe() decorator
 - PipeTransform interface
 - transform() method
- 2. Register pipe in app module
 - app.module.ts
 - declarations
- 3. Use custom pipe in template

variable | pipename

Custom Pipes in Angular

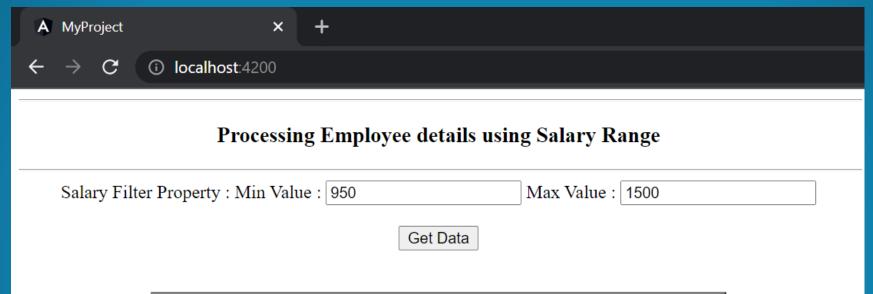
```
import {Pipe, PipeTransform} from "@angular/core";
@Pipe({ name: "pipename" })
export class PipeClass implements PipeTransform
  public transform() : ReturnType
            // Logic
```

Assignment-1 (Custom Pipes)



Practical HandsOns

Assignment-2 (Custom Pipes)



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

filter.pipe.ts





