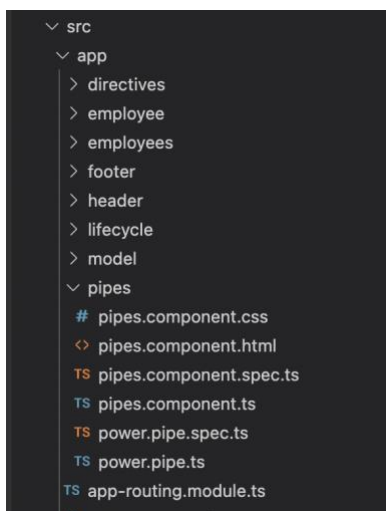


STEP 10: Pipes

1. Add Pipes component by executing the below command from the angulardemo folder:
ng g c pipes
2. Add a custom pipe **from within the pipes folder** by executing the below command:
ng g p power

```
• (base) Manishs-MacBook-Pro:angulardemo Shalini$ ng g c pipes
Node.js version v17.5.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production.
For more information, please see https://nodejs.org/en/about/releases/.
CREATE src/app/pipes/pipes.component.css (0 bytes)
CREATE src/app/pipes/pipes.component.html (20 bytes)
CREATE src/app/pipes/pipes.component.spec.ts (592 bytes)
CREATE src/app/pipes/pipes.component.ts (271 bytes)
UPDATE src/app/app.module.ts (1236 bytes)
• (base) Manishs-MacBook-Pro:angulardemo Shalini$ cd src/app/pipes/
• (base) Manishs-MacBook-Pro:angulardemo Shalini$ ng g p power
Node.js version v17.5.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production.
For more information, please see https://nodejs.org/en/about/releases/.
CREATE src/app/pipes/power.pipe.spec.ts (183 bytes)
CREATE src/app/pipes/power.pipe.ts (215 bytes)
UPDATE src/app/app.module.ts (1299 bytes)
```

3. Folder structure looks as follows:



4. Comment out the previous tags from app.component.html file and add <app-pipes> tag
5. There are various built-in pipes available which can be used to format the output. Lets have a look at various in-built pipes
6. Add the below in pipes.component.html file:
<h3>Pipes Demo</h3>
7. **Decimal Pipe:**
<p ngNonBindable>{{ 3.14659265 | number: '3.1-2' }}</p>
<p>{{ 3.14659265 | number: '3.1-2' }}</p>

For setting the Locale, add the following in app.module.ts
import { [registerLocaleData](#) } from '@angular/common';
import localeFr from '@angular/common/locales/fr';
// the second parameter 'fr' is [optional](#)
[registerLocaleData](#)(localeFr, 'fr');

```
<p ngNonBindable>{{ 3.14159265 | number: '3.1-2' : 'fr' }}</p>
<p>{{ 3.14159265 | number: '3.1-2' : 'fr' }}</p>
```

8. Inject Locale in constructor of pipes.component.ts as follows:
constructor(@Inject(LOCALE_ID) public locale: string) { }

9. Currency Pipe

```
<p>{{ 1234.56 | currency:'GBP':false }}</p>
<p>{{ 1234.56 | currency:"GBP":true }}</p>
<p>{{ locale }}</p>
```

10. Date Pipe:

Add below property in pipes.component.ts :

```
bdate = new Date()
```

Add below in pipes.component.html

```
<p>{{ bdate }}</p>
<p>{{ bdate | date:'short' }}</p>
<p>{{ bdate | date:'long' }}</p>
<p>{{ bdate | date:'medium' }}</p>
```

<https://angular.dev/api/common/DatePipe?tab=usage-notes>

11. JSON Pipe:

Add below property in pipes.component.ts :

```
object: Object = {foo: 'bar', baz: 'qux', nested: {xyz: 3, numbers: [1, 2, 3, 4, 5]}};
```

Add below in pipes.component.html

```
<p>Without JSON pipe:</p>
<pre>{{ object }}</pre>
<p>With JSON pipe:</p>
<pre>{{ object | json }}</pre>
```

12. Lower ,Upper, title case pipe

Example :

```
<p>{{ 'ASIM' | lowercase }}</p>
```

Uppercase : Transforms a string to uppercase

Example :

```
<p>{{ 'asim' | uppercase }}</p>
```

```
<h1>Purchases from {{ company | titlecase }}</h1>
```

where company is a property in component

13. Percent Pipe

```
<p ngNonBindable>{{ 0.123456 | percent }}</p>
<p>{{ 0.123456 | percent }}</p> //12.346%
<p ngNonBindable>{{ 0.123456 | percent: '2.1-2' }}</p>
<p>{{ 0.123456 | percent: '2.1-2' }}</p> //12.35%
<p ngNonBindable>{{ 42 | percent: '10.4-4' }}</p>
<p>{{ 0.123456 | percent : "10.4-4" }}</p>
//0,000,000,012.3456%
```

14. Slice Pipe

```
<p ngNonBindable>{{ [1,2,3,4,5,6] | slice:1:3 }}</p>
<p>{{ [1,2,3,4,5,6] | slice:1:3 }}</p> //2,3
<p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2 }}</p>
<p>{{ [1,2,3,4,5,6] | slice:2 }}</p> //3,4,5,6
<p ngNonBindable>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p>
<p>{{ [1,2,3,4,5,6] | slice:2:-1 }}</p> //3,4,5
<pre ngNonBindable><ul>
<li *ngFor='v of [1,2,3,4,5,6] | slice:2:-1'>
    {{ v }}
</li></ul>
</pre>
<ul>
```

// 3 4 5

```

<li *ngFor="let v of [1,2,3,4,5,6] | slice:2:-1">
  {{v}}
</li>
</ul>

```

15. Async Pipe:

```

x: Promise<string> | null = Promise.resolve("hello world");
OR

```

```

import { Observable, of } from "rxjs";
x: Observable<string> | null = of("hello world");

```

```

<p>{{x | async}}</p>

```

16. Combine multiple pipes:

```

<p>The event will occur at {{ scheduledOn | date | uppercase }}.</p>

```

```

scheduledOn = new Date() [ In the component ts file ]

```

Above will convert date to the format and convert date to uppercase

<https://angular.dev/guide/templates/pipes>

17. Custom Pipe

- a. The naming convention for custom pipes consists of two conventions:
 name - camelCase is recommended. Do not use hyphens.
 class name - PascalCase version of the name with Pipe appended to the end
- b. By default, all pipes are considered pure, which means that it only executes when a primitive input value (such as a String, Number, Boolean, or Symbol) or a changed object reference (such as Array, Object, Function, or Date). Pure pipes offer a performance advantage because Angular can avoid calling the transformation function if the passed value has not changed.
add pure:false to create impure pipe
- c. Modify transform function of power.pipe.ts:

```

transform(value: number, ...args: number[]): unknown {
  console.log(value)
  console.log(args)
  let power = 0;
  if(args.length == 0)
    power = 2;
  else
    power = args[0]
  return value** power ;
}

```

Add below in pipes.component.html

```

<p ngNonBindable>{{ 20 | power }}</p>
<p>20<sup>4</sup> : {{ 20 | power:5 }}</p>

```