No Provider for CustomPipe - angular 4

Asked 2 years ago Active 5 months ago Viewed 14k times

27

2

I've a custom decimal format pipe that uses angular Decimal pipe inturn. This pipe is a part of the shared module. I'm use this in feature module and getting no provider error when running the application.

Please ignore if there are any typo.

./src/pipes/custom.pipe.ts

```
import { DecimalPipe } from '@angular/common';
..
@Pipe({
    name: 'customDecimalPipe'
})
...
export class CustomPipe {
constructor(public decimalPipe: DecimalPipe) {}
transform(value: any, format: any) {
    ...
}
```

./modules/shared.module.ts

```
import  { CustomPipe } from '../pipes/custom.pipe';

...

@NgModule({
  imports:      [ .. ],
  declarations: [ CustomPipe ],
  exports:    [ CustomPipe ]
})
export class SharedModule { }
```

I inject the custom pipe in one of the components and call transform method to get the transformed values. The shared module is imported in he feature module asked Sep 19 '17 at 12:02

mperle
731 4 11 22

Possible duplicate of [Angular 2/4 use pipes in services and components](#) – alexKhymenko Sep 19 '17 at 12:11

3 Answers

57

If you want to use pipe's `transform()` method in component, you also need to add `CustomPipe` to module's providers:

```
import  { CustomPipe } from '../pipes/custom.pipe';

...

@NgModule({
  imports:      [ .. ],
  declarations: [ CustomPipe ],
  exports:     [ CustomPipe ],
  providers:    [ CustomPipe ]
})
export class SharedModule { }
```

answered Sep 19 '17 at 12:07

Stefan Svrkota
30.6k 5 75 76

1 Thank you for your answer. so that means the Pipe becomes an injectable so it has to be provided. am i right.? – mperle Sep 19 '17 at 12:27

@mperle Exactly. – Stefan Svrkota Sep 19 '17 at 12:28

But, does that deviate form angular's shared module definition. ? – mperle Sep 19 '17 at 13:18

great answer! Thanks. – Mattijs Jan 25 '18 at 2:31

@mperle you can add the `CustomPipe` to the *ServiceModule*(if you have a dedicated one) or any feature module instead of the shared one. – Faruq

7

Apart from adding the `CustomPipe` to the module's provider list, an alternative is to add to the component's providers. This can be helpful if your custom pipe is used in only a few components.

```
import  { CustomPipe } from '../pipes/custom.pipe';
...
@Component({
    templateUrl: './some.component.html',
    ...
    providers: [CustomPipe]
})
export class SomeComponent{
    ...
}
```

Hope this helps.

answered May 16 '18 at 16:53
Faruq
752 5 19

1

You could also make the pipe Injectable (the same way it is done with the services you create using the cli):

```
    import { Injectable, Pipe, PipeTransform } from '@angular/core';

    @Pipe({
      name: 'customDecimalPipe'
    })
    @Injectable({
      providedIn: 'root'
    })
    export class CustomPipe extends PipeTransform {
      ...
    }
```

answered Apr 27 at 18:53
José Antonio Postigo