

# Using Angular Pipes in Components or Services

1 Comment / 2 minutes of reading / March 9, 2023

← [KeyValue Pipe](#)

[Angular Tutorial](#)

[Component Communication](#)



In this guide let us learn how to use [Angular Pipes](#) in [components](#) & [Services](#). We usually use [Angular Pipes](#) in the template. But a pipe is nothing but a class with a method `transform`. Whether it is a built-in pipe or custom pipe, we can easily use it in an [angular component](#) or [service](#).

## Table of Contents

[Using Pipes in Components & Services](#)

[Using Date pipe in Components & Services](#)

[References](#)

## Using Pipes in Components & Services

Using pipes in your code involves three simple steps

1. Import the pipe in Module
2. Inject pipe in the constructor
3. Use the transform method of the pipe

## Using Date pipe in Components & Services

First import the [DatePipe](#) from `@angular/common`. Add it in the [Angular Provider](#) metadata providers: `[ DatePipe ],`.

```
1
2 import { BrowserModule } from '@angular/platform-browser';
3 import { NgModule } from '@angular/core';
4
5 import { DatePipe } from '@angular/common';
6
7 import { AppRoutingModule } from './app-routing.module';
8 import { AppComponent } from './app.component';
9
10 @NgModule({
11   declarations: [
12     AppComponent
13   ],
14   imports: [
15     BrowserModule,
16     AppRoutingModule
17   ],
18   providers: [
19     DatePipe
20   ],
21   bootstrap: [AppComponent]
22 })
23 export class AppModule { }
24
25
```

Open the `app.component.html` and [inject](#) the [DatePipe](#) in the constructor.

```
1  
2 constructor(private datePipe:DatePipe) {  
3 }  
4
```

You can use it in component as shown below.

```
1  
2 this.toDate = this.datePipe.transform(new Date());  
3
```

The transform method accepts the date as the first argument. You can supply additional [Parameters to DatePipe](#) like format, timezone & locale .

```
1  
2 this.toDate = this.datePipe.transform(new Date(),'dd/MM/yy HH:mm');  
3
```

The complete component code is as below.

```
1  
2 import { Component, OnInit } from '@angular/core';  
3 import { DatePipe } from '@angular/common';  
4  
5 @Component({  
6   selector: 'app-root',  
7   template: `
```

```
8     {{toDate}}
9     `
10    styleUrls: ['./app.component.css']
11  })
12  export class AppComponent implements OnInit {
13    title = 'pipesInService';
14
15    toDate
16
17    constructor(private datePipe: DatePipe) {
18    }
19
20    ngOnInit() {
21
22      this.toDate = this.datePipe.transform(new Date());
23    }
24  }
25
```

## References

1. [Transforming Data Using Pipes](#)

### Read More

1. [Angular Pipes](#)
2. [Date Pipes in Angular](#)
3. [Custom Pipes](#)
4. [Dependency Injection](#)
5. [Angular Providers](#)

[← KeyValue Pipe](#)[Angular Tutorial](#)[Component Communication](#)