

#angular: What are components in angular?

In this tutorial, I would like talk about:

- Basics of components ?
- Structure of angular component ?
- How to register a component in a Module file ?
- How to write component.ts file ?

## Basics of components ?

- ★ Components are building blocks of an angular application.
- ★ They control the “views” part of an application. It is a bit off topic but an important concept to understand, there are different presentation patterns for any application we develop. Such as MVC = Model View Controller Pattern.
- ★ We use angular cli command “generate” to create angular artifacts. To create angular component we will use:  

```
ng generate <type> <name>
```

```
ng generate component hero
```

-> Now an angular component has been created in your application.

## Structure of angular component ?

- ★ Once a component is created, let's see the structure of the angular component and files included in it.
- ★ There are 4 files included by default:
  - Hero.component.ts -> it contains the data and presentation logic of the component
  - Hero.component.html -> it template file containing the view of the component.
  - Hero.component.css -> it provides styles for your html template file
  - Hero.component.spec.ts -> it contains the units for your component.
- ★ Component keyword is added in each file name by default.

## How to register a component in a Module file ?

- ★ Actually, angular applications consist of different modules.
- ★ So, angular framework will not know about your component unless you tell the angular framework that you have created this component and you want to use it in your app.

- ★ So, to let angular framework know about the component you created, you need to register that component in module.ts file.
- ★ If you have created a component using cli command, as we discussed in last section, then angular cli automatically registers a component upon creation by adding it to the declaration property of the main application module.
- ★ If you open the app.module.ts file, it looks like below:

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { AppComponent } from './app.component';
import { HeroComponent } from './hero/hero.component';
@NgModule({
  declarations: [
    AppComponent,
    HeroComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

### How to write component.ts file ?

- ★ Hero.component.ts file looks like below:

```
@Component({
  selector: 'app-hero',
  templateUrl: './hero.component.html',
  styleUrls: ['./hero.component.css']
})
export class HeroComponent implements OnInit {
  constructor() { }
  ngOnInit(): void { }
}
```

- ★ Actually, it consists of 2 major parts:

- Decorator -> @Component
- Class -> HeroComponent
- ★ So, the angular framework needs something to let it know that the class I am writing is a component. To let angular know about my component class, I need to write a decorator @Component.
- ★ What is a decorator ? -> it is a typescript concept. The decorator is used to pass a metadata to angular so that it knows how to create a specific artifact.
- ★ In above code, the metadata of the @component is a plain object with specific properties:

```
{
  selector: 'app-hero',
  templateUrl: './hero.component.html',
  styleUrls: ['./hero.component.css']
}
```
- ★ selector -> It defines the name of the component to be identified in an HTML template. It tells Angular where to create the component when it finds the corresponding tag in HTML.
- ★ templateUrl -> it is the path of the component template file. You can also pass inline templates by using template property.
- ★ styleUrls -> it is a path to component style files. This option is an array, it can contain multiple files. You can also provide inline styles by using styles property.

I hope it helped you. I would really appreciate your feedback. How can I improve these tutorials?

follow and add Rizwan Mushtaq Dhudhaal  in your friend list