What does @Component decorator exactly do?

6

3

From official docs we know that

> Component decorator allows you to mark a class as an Angular component and provide additional metadata that determines how the component should be processed, instantiated and used at runtime.

But I would like to go deeper and understand what **Component decorator** really does except providing additional metadata.

I dived into source code and found that all decorators are created with help of makeDecorator function. And here I got lost. Where is the difference for example for Component and ngModule decorators? Are they doing the same thing? Don't think so.

Like an answer it would be great to have step by step description what I should do to recreate Component Decorator without makeDecorator function.

UPD: and, Yes, of course, I know how TypeScript Decorators work.

angular angular2-components angular2-decorators

edited Dec 6 '16 at 8:40

asked Dec 5 '16 at 21:42

Stepan Suvorov
12.9k 18 73 146

1 Answer

8

> But I would like to go deeper and understand what Component decorator really does except providing additional metadata.

Why does it need to be anything more? It's said that the argument passed to the `@Component` decorator function is the component

```
npm install --save reflect-metadata
```

2. Create the decorator function[1]

```
import 'reflect-metadata';

interface MyComponentMetadata {
  template?: string;
  selector?: string;
}

function MyComponent(metadata: MyComponentMetadata) {
  return function(ctor: Function) {
    // add metadata to constructor
    Reflect.defineMetadata('annotations', metadata, ctor);
  }
}
```

3. Use it

```
@MyComponent({
  selector: 'component',
  template: '<hello></hello>'
})
class HelloComponent {
}
```

4. Now when you need to get the metadata, just do

```
let metadata = Reflect.getMetadata('annotations', HelloComponent);
```

The purpose of the metadata is to provide information for Angular to know what to do with the class. So the decorator doesn't really need to be anything more than just a metadata provider. Angular decides what to do with the metadata, not the decorator function.

1 - See [TypeScript documentation on decorators]

FYI: In my `@angular/cli` @1.2.1 project i had to import `reflect-metadata` at the top of the `main.ts` file to make the compiler aware of the `defineMetadata` and `getMetadata` functions. Doing the import in my decorators source file caused the compiler to complain about missing `@NgModule` decorators. – TekTimmy Jul 27 '17 at 7:58