

# Unexpected value 'AnyComponent' declared by the module 'AppModule'

Asked 2 years, 11 months ago   Active 1 year, 9 months ago   Viewed 28k times

▲  
21 I'm using Angular2 and I got this problem when trying to use two classes in the same Typescript file.  
At compile time doesn't give me any error but when I try to execute the page the console.log is giving this error:

▼  
★  
1

```
Error: BaseException@http://www.my.app/panel-  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:5116:27  
  
CompileMetadataResolver</CompileMetadataResolver.prototype.getNgModuleMetadata/<@http://www.  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:13274:35  
  
CompileMetadataResolver</CompileMetadataResolver.prototype.getNgModuleMetadata@http://www.  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:13261:21  
  
RuntimeCompiler</RuntimeCompiler.prototype._compileComponents@http://www.my.app/panel-  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:15845:28  
  
RuntimeCompiler</RuntimeCompiler.prototype._compileModuleAndComponents@http://www.my.app/p  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:15769:36  
  
RuntimeCompiler</RuntimeCompiler.prototype.compileModuleAsync@http://www.my.app/panel-  
module/node_modules/@angular/compiler//bundles/compiler.umd.js:15746:20  
  
PlatformRef_</PlatformRef_.prototype._bootstrapModuleWithZone@http://www.my.app/panel-  
module/node_modules/@angular/core//bundles/core.umd.js:9991:20  
PlatformRef_</PlatformRef_.prototype.bootstrapModule@http://www.my.app/panel-  
module/node_modules/@angular/core//bundles/core.umd.js:9984:20  
@http://www.my.app/panel-module/app/main.js:4:1  
@http://www.my.app/panel-module/app/main.js:1:31  
@http://www.my.app/panel-module/app/main.js:1:2  
Zone</ZoneDelegate</ZoneDelegate.prototype.invoke@http://www.my.app/panel-  
module/node_modules/zone.js/dist/zone.js:332:20  
Zone</Zone</Zone.prototype.run@http://www.my.app/panel-  
module/node_modules/zone.js/dist/zone.js:225:25  
scheduleResolveOrReject/<@http://www.my.app/panel-  
module/node_modules/zone.js/dist/zone.js:586:53  
Zone</ZoneDelegate</ZoneDelegate.prototype.invokeTask@http://www.my.app/panel-  
module/node_modules/zone.js/dist/zone.js:365:24  
Zone</Zone</Zone.prototype.runTask@http://www.my.app/panel-  
module/node_modules/zone.js/dist/zone.js:265:29  
drainMicroTaskQueue@http://www.my.app/panel-
```

```
module/node_modules/zone.js/dist/zone.js:491:26
F/</g@http://www.my.app/panel-module/node_modules/core-js/client/shim.min.js:8:10016
F/<@http://www.my.app/panel-module/node_modules/core-js/client/shim.min.js:8:10138
a.exports/k@http://www.my.app/panel-module/node_modules/core-
js/client/shim.min.js:8:14293

Evaluating http://www.my.app/panel-module/app/main.js
Error loading http://www.my.app/panel-module/app/main.js
```

Below is my component typescript file.

// MyComponent.ts

```
import { Component, OnInit } from '@angular/core';

@Component({
  selector: 'my-app',
  templateUrl: '/path/to/view',
})

export class MyObject {
  id: number;
}

export class MyComponent {
  obj: MyObject;

  // unnecessary code
}
```



asked Aug 17 '16 at 15:56



[Herlon Aguiar](#)

404 1 7 16

### 3 Answers

36



You have to change the order of your classes, then there is no problem to have multiple classes in the same file. Cause in your case that decorator `@Component` is now being used for your class `MyObject` instead of `MyComponent` !

The decorator `@Component` needs to be directly in front of your Component definition!!

```
import { Component, OnInit } from '@angular/core';
```

```
export class MyObject1 {  
  id: number;  
}
```

```
@Component({  
  selector: 'my-app',  
  templateUrl: '/path/to/view',  
})
```

```
export class MyComponent {  
  obj: MyObject;
```

```
  // unnecessary code  
}
```

```
export class MyObject2 {  
  id: number;  
}
```

answered Aug 17 '16 at 16:02



[mxii](#)

12k

2

34

44

Thanks, I fixed on the other way but I didn't know why I needed to do that. – [Herlon Aguiar](#) Aug 18 '16 at 0:23

- 2 because that decorator `@Component` needs to be in front of your component. without this decorator, your component isn't a real component.. otherwise it's just a class.. – [mxii](#) Aug 18 '16 at 4:51

thanks, so the component should defined first before used in the module. – [Adi Prasetyo](#) Nov 2 '16 at 3:22

I had the reverse problem; I wanted to define a `Directive` but was using the `Component` decorator. Thanks! – [The Red Pea](#) Apr 29 '18 at 3:05

Here is how I fixed it:

2

## Put the MyObject into a separated file and import him.

// MyObject.ts

```
export class MyObject {  
  id: number;  
}
```

// MyComponent.ts

```
import { Component, OnInit } from '@angular/core';  
import { MyObject } from 'path/to/MyObject';  
  
@Component({  
  selector: 'my-app',  
  templateUrl: '/path/to/view',  
})  
  
export class MyComponent {  
  obj: MyObject;  
  
  // unnecessary code  
}
```

answered Aug 17 '16 at 15:56



[Herlon Aguiar](#)

404 1 7 16

---

Why did you decide to do that instead of just place your `myObject` before your component, as @mxii suggested? – [neoswf](#) Jan 28 '17 at 18:41

---

I did that before @mxii post the answer. I didn't know what the right answer but this worked to me. – [Herlon Aguiar](#) Jan 30 '17 at 1:15

---

cool man @herlon-aguiar – [neoswf](#) Jan 30 '17 at 13:25

---

FYI, my case i just had some js logic where it could'nd be

0

```
<my-tag [HTML]="sourcesHtml ? sourcesHtml[i] : ''">
```

changed to

```
<my-tag [HTML]="sourcesHtml[i]">
```

and error was removed

answered Oct 4 '17 at 8:01



[bresleveloper](#)

4,244 3 24 42