

Error: "Encountered undefined provider! Usually this means you have a circular dependencies"

Asked 2 years, 4 months ago Active 1 month ago Viewed 52k times



46



8

Here's a somewhat useless error I'm getting in my Angular / TypeScript application. Until someone makes the error message better, what can we do about this? What are the most likely situations that cause this to happen?

Uncaught Error: Encountered undefined provider! Usually this means you have a circular dependencies (might be caused by using 'barrel' index.ts files.

```
at Object.syntaxError
at eval      at Array.forEach (native) [<root>]
at CompileMetadataResolver._getProvidersMetadata
at CompileMetadataResolver.getNgModuleMetadata
at CompileMetadataResolver.getNgModuleSummary
at eval
...
```



edited May 18 at 3:04



[risingTide](#)

683 4 12 41

asked Mar 24 '17 at 20:15



[Alexander Taylor](#)

7,718 8 45 64

- 1 I wish the error would mention the provider in question! In a non-trivial app, there are a lot of providers to narrow down. – [Stan Kurdziel](#) Jul 9 '18 at 22:30
- 2 I don't think it could provide the exact name, since the very problem it's complaining about is it's just getting undefined somewhere in its providers, e.g.: [Foo1, Foo2, undefined, Foo3]. But it could give the index of the entry that's undefined or line number or something. I agree. – [Alexander Taylor](#) Jul 11 '18 at 1:27

12 Answers



It is very hard to tell from the error message which provider causes this issue. The way I managed to debug this is the following:

32

- I went into the `node_modules@angular\compiler\bundles\compiler.umd.js` file
- I found the line where it says: "Encountered undefined provider! Usually this means you have a circular dependencies. This might be caused by using 'barrel' index.ts files."
- One line before I added `console.log('type', type);` in order to see in which file is the undefined provider (You can also console log other relevant variables there).
- In the relevant file I found the 'barrel' import that caused the issue, and replaced it with exact file path import.

edited May 3 at 20:59



Kris Hollenbeck

13k 18 54 90

answered Jul 12 '18 at 11:24



Ophir Bushinsky

796 7 11

- 1 while "cave man like" (surely the tool could output *something* if we ourselves can make this simple edit to get some context), this suggestion does at least let us get some info re the culprit. – [Neil Clayton](#) Jul 30 '18 at 20:18
 - 2 I added a `console.log(looking for ${provider}, ${debugInfo});` The provider might be undefined, but the debugInfo gives on the context of the failure. – [Neil Clayton](#) Jul 30 '18 at 21:20
- I had this error, and it turned out it was a problem with a barrel file. Once I stopped importing the barrel file and using it's exports and started to import those things directly, it worked. Took me two days though. Upgrading from 5 => 6 is quite a bit of work. – [frosty](#) Sep 6 '18 at 19:30
- I used the above code from Neil `console.log(`looking for ${provider}, ${debugInfo}`);` and added the `--verbose` flag to my `ng build --prod` command and found it fairly helpful. – [Jeph](#) Sep 14 '18 at 17:48
- 1 Instead of manipulating the angular bundle files, one may also put a breakpoint in the corresponding line in `compiler.js` file that includes the error message using browser dev tools. Once the breakpoint is hit, I investigated the call stack and found the `providers` array, which based on the types of the other imports helped me to trace down the faulty provider. – [Capricorn](#) Nov 1 '18 at 23:14

One possibility is trying to declare a service and module in the same file, and declaring the module before the service:

21

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

@NgModule({providers: [FooService]}) // WRONG: used before declared
export class FooModule {
}

@Injectable()
export class FooService {
}
```

You can fix this by declaring the service first, or you can use [forwardRef](#) like this:

```
import {forwardRef, Injectable, NgModule} from '@angular/core';

@NgModule({providers: [forwardRef(() => FooService)]})
export class FooModule {
}

@Injectable()
export class FooService {
}
```

answered Mar 24 '17 at 20:15



Alexander Taylor

7,718 8 45 64

Works perfectly! Is there a scenario where creating a forwardRef like this would be a bad idea? – [cjsimon](#) Apr 27 '17 at 2:32

-
- 1 I don't think forwardRef is ever dangerous, just unnecessary if your symbols are declared before they're used. Sometimes it's better to put things in separate files though, but small/trivial module declarations aren't always worth the extra file. – [Alexander Taylor](#) Apr 27 '17 at 6:50

I have my components in separate files. Would it be enough to declare the provider module first, or does it have to be initialized on the page before the other module? – [cjsimon](#) Apr 27 '17 at 21:07

-
- 1 If you're using TypeScript, and you have a file that imports another file, I think the compiler will cause everything to be loaded in the correct order. Sorry I just saw your comment; if you got this error any my solution didn't fix it, please post another solution :) – [Alexander Taylor](#) May 3 '17 at 16:38

Your solution works for me. I was just wondering if the provider module had to be initialized before use, though it seems that simply declaring it is enough to get it working, as you have shown in your answer – [cjsimon](#) May 4 '17 at 21:35



3



I was running into this while using ng-packagr to package a library then importing it into another library. What ended up being my problem was indeed the 'barrel' index.ts imports.

This was making it break

```
import { Activate, Another, Data } from './services
@NgModule({ providers: [ Activate, Another, Data ]})
```

where in the services folder I had one index.ts file that was exporting all of the services.

This fixed it:

```
import { Activate } from './services/activate.service.ts'
import { Another } from './services/another.service.ts'
import { Data } from './services/data.service.ts'
@NgModule({ providers: [ Activate, Another, Data ]})
```

answered Dec 19 '18 at 22:18



Zachary Hale

61 5



Got this error running --prod.

2

You can't import things like that:



```
import { MyService } from '.';
```

You should use the full path

```
import { MyService } from './my.service'
```

answered Oct 1 '18 at 22:23



Jaime Yule

592 1 7 19



Check if the module can find the service you have mentioned.

1

In my case I was exporting a `guard` from my `guards` folder. this folder contained an `index.ts` file. There were two more files in this `guards` folder `auth.guard.ts` and `company.guard.ts` . Ideally I **should have exported** these files in the index as follows:



contents of **guards/index.ts**

```
export * from './auth.guard';
export * from './company.guard'; // forgot this
```

But I forgot to include the line above that exports from `company.guard.ts` . This was creating problem.

answered Jul 5 '18 at 5:57



[Ramesh Pareek](#)

804 1 14 39



For me just helped to restart the NG SERVE

1



answered Jun 18 at 8:15



[Václav Mikeska](#)

55 1 5



I got this error when missing an import for an override of an angular class. I imagine an incorrect import may cause the error in other ways also.

0

In my case I had no import statement for `File` and it defaulted to a `File` interface which wasn't what I wanted. Adding `import { File } from "@ionic-native/file"` fixed the problem.



answered Jul 9 '18 at 22:28



[Stan Kurdziel](#)

3,982 1 30 35



I also console logged the value ***right before the error message*** statement in `node_modules\@angular\compiler\bundles\compiler.umd.js` file.

0

And checked that ***Document interface*** was there in **providers array** of a **component** which was the root cause.



I **removed it to fix** this issue.

answered Sep 11 '18 at 5:33



[Simran kaur](#)

141 1 6

in my case i changes this

0

```
@Injectable()  
export class LocationTracker {  
}
```

to

```
@Injectable()  
export class LocationTrackerProvider {  
}
```

answered Nov 24 '18 at 6:38



RBT

368 1 15

Sometime this issue occurred because of some dependency in third party api used in angular app. I faced same issue and resolved it using following steps:

0

1. Removed package.lock.json file
2. Deleted node_modules folder
3. Again run npm install
4. Run "ng build --prod" These steps will resolve your issue.

answered Feb 15 at 7:02



Nivedita

1

didn't work for me. – [Vladimir Despotovic](#) Jun 20 at 15:46

in my case I just deleted the @Injectable() decorator from my service(cause it didn't need any services to be injected in it)

answered Jun 19 at 11:39



Sadra Rahmani

1 3



0



```
@Component({
  selector: "app-dispatching-history",
  templateUrl: "./dispatching-history.component.html",
  styleUrls: ["./dispatching-history.component.css"],
  providers: [RecommendationService, Location, { provide: HashLocationStrategy,
    useClass: HashLocationStrategy }]}
})
```

I just added `Location, { provide: HashLocationStrategy, useClass: HashLocationStrategy }` as a provider in one of my component and didn't do any other necessary changes in files like `app.module.ts`. I don't need it anymore, so I simply remove it. And the command `ng build -c deploy --build-optimizer --aot --prod --sourceMap` works again.

answered Jul 2 at 4:04



heinels

1

protected by [mu 無](#) Jul 29 at 16:26

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

