Injecting a service into an
in Angular

Matheus CAS Follow
Jan 30, 2017 · 1 min read

Let's say that you have a Service1 and Service2 in an regular Angular application — nothing fancy at all. Let's say now that this Service2 depends on Service1 and right way you would write something like:

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

@Injectable()
export class Service1 {

  constructor() { }

  doSomethingFromService1(){
    console.log('service 1 just did something');
  }

}
```

service1.service.ts hosted with ♡ by GitHub view raw

And in Service2:

```
import { Service1 } from './service1.service';
import { Injectable } from '@angular/core';

@Injectable()
export class Service2 {

  constructor(private service1: Service1) { }

  do(){
    this.service1.doSomethingFromService1();
    console.log('after service 1 function');
  }

}
```

service2.service.ts hosted with ♡ by GitHub view raw

So far, you won't see any errors in your browser's console. But at the time that you inject Service2 into a component,

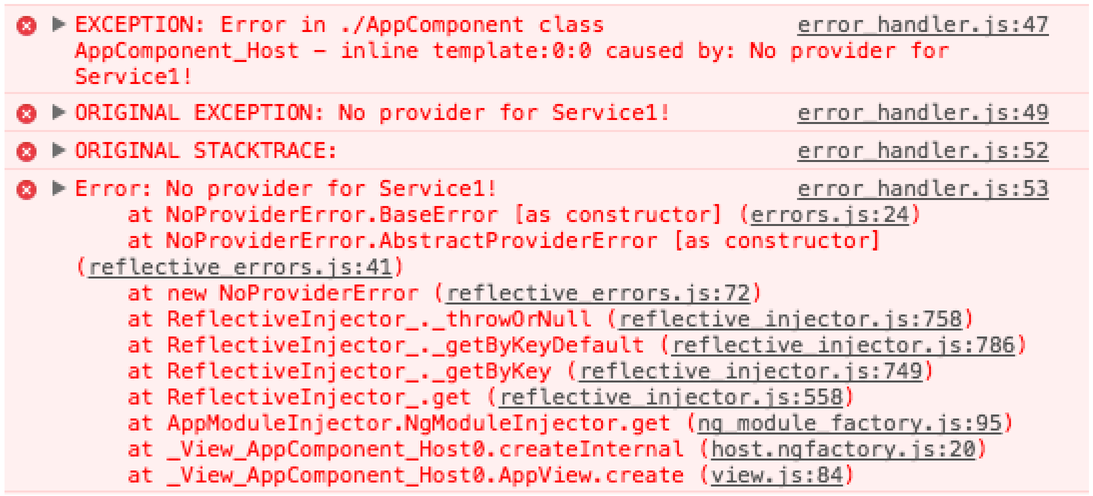
```
import { Service2 } from './service2.service';
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html'
  styleUrls: ['./app.component.css'],
  providers: [Service2]
})
export class AppComponent {
  title = 'app works!';

  constructor(private service2: Service2){}
}
```

app.component.ts hosted with ♡ by GitHub view raw

you'll see something like this:



```
▶ EXCEPTION: Error in ./AppComponent class                    error_handler.js:47
  AppComponent_Host - inline template:0:0 caused by: No provider for
  Service1!
▶ ORIGINAL EXCEPTION: No provider for Service1!               error_handler.js:49
▶ ORIGINAL STACKTRACE:                                        error_handler.js:52
▶ Error: No provider for Service1!                            error_handler.js:53
      at NoProviderError.BaseError [as constructor] (errors.js:24)
      at NoProviderError.AbstractProviderError [as constructor]
  (reflective_errors.js:41)
      at new NoProviderError (reflective_errors.js:72)
      at ReflectiveInjector_._throwOrNull (reflective_injector.js:758)
      at ReflectiveInjector_._getByKeyDefault (reflective_injector.js:786)
      at ReflectiveInjector_._getByKey (reflective_injector.js:749)
      at ReflectiveInjector_.get (reflective_injector.js:558)
      at AppModuleInjector.NgModuleInjector.get (ng_module_factory.js:95)
      at _View_AppComponent_Host0.createInternal (host.ngfactory.js:20)
      at _View_AppComponent_Host0.AppView.create (view.js:84)
```

No provider for Service1

So, whats happening here?

1 — Angular is instantiating Service2 because we injected it into AppComponent and declared it as a provider.

2 — To complete this task, Angular will check th
and, in this case, it is Service1. But, how Angular
is a provider to Service2? There is no `providers` i
instantiate Service1 manually it is strongly not recommended.

> *So what we do?*

> We must tell Angular to instantiate Service1 to be available (instatiated) before Service2.

Hence, we declare it as a provider into our module. In this small example application, we have only one module, the `app.module.ts`.

```
import { Service1 } from './service1.service';
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';

import { AppComponent } from './app.component';
import { Component1Component } from './component1/component

@NgModule({
  declarations: [
    AppComponent,
    Component1Component
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpModule
  ],
  providers: [Service1],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

app.module.ts hosted with ♡ by GitHub view raw

Đăng nhập vào medium.com bằng Google

Hiện Doãn
toilati123vn@gmail.com

TIẾP TỤC VỚI HIỆN

Now, Service1 is a singleton provider to the entire application and it is already instantiated. After that, Angular won't complain anymore about Service1. :)

See ya.

JavaScript Angular

Discover Medium

Welcome to a place where words matter. On Medium, smart voices and original ideas take center stage - with no ads in sight. Watch

Make Medium yours

Follow all the topics you care about, we'll deliver the best stories for you t homepage and inbox. Explore

Đăng nhập vào medium.com bằng Google

Hiện Doãn
toilati123vn@gmail.com

TIẾP TỤC VỚI HIỆN

About Help Legal