

- By Deepa Chawasia

What are Modules In Angular?

Angular doesn't scan automatically all your components, directives etc.

You have to tell Angular what component, directives, service your project have. Or you are using.

Modules does that Job for you!

Module tell Angular list of all components, directives or services we are using.

It is basically a place where you group together all your components, directives, services.

@NgModule ({})

declaration: [AppComponent,
XComponent,
YComponent]

imports: [AppRoutingModule,
BrowserModule, ...]

providers: [], (services goes here)

bootstrap: [AppComponent],

entryComponents: []

})

export class AppModule { }

1. `@NgModule` - Angular analyzes `ngModule`s to understand application and its features.

You can't use a feature without including it in `@NgModule`.

2. Imports - Imports are used for importing other modules like `FormsModule`, `RoutingModule` etc. and their features into our main Module.

3. Providers - It defines all services we are providing in our angular app.

All services declare in your app must be inside the Providers of APPMODULE

OR

You have one other method to import your services throughout root of application

`@Injectable({
 providedIn: 'root'
})`

If you use `@Injectable`, You don't have to declare it in providers of Module.

Remember `@Injectable (S)`
provided In : 'root'
3)

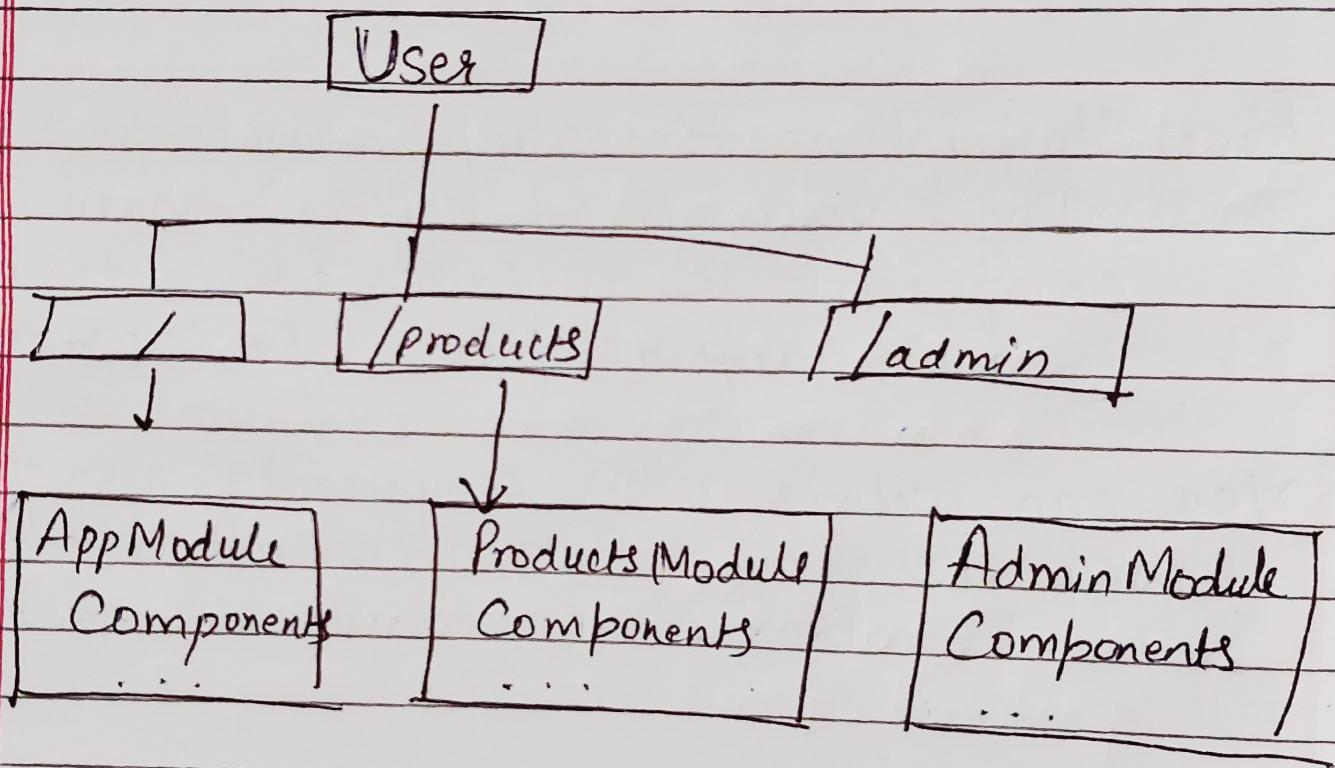
Will come above every service in Service Component.

`Bootstrap[]` - Bootstrap is important for starting your app. It defines which component is available straight in your app at first. (i.e app-root)

You can include other components also that's why it's array. But it is not likely to add other components generally

Lazy Loading In Angular

Consider Our application has 3 routes,
Every routes has a module
associated with it that contains
their respective component, services
and directives.



Here you can see we have 3 routes

- Root
- Products
- Admin

Now When user visit root route then
we must be loading the respective
modules, and then we load
other modules when required.

But does it really happen ??

The answer is NO.

Angular actually load all the modules whenever we visit any page

It's like you ordered starter, maincourse and desert

And chef is saying No I will make all starter, maincourse, desert first then start serving.

You definitely want atleast your starter, first, and ask chef to make rest when it's required.

So Lazy Loading does the same for you.

In Lazy Loading the modules are loaded only if the user hit the route of that particular Module.

Advantages of Lazy Loading In Angular

- * Initially we load a smaller bundle and module for our root route.
- * We load more modules / bundle as we hit route for following.
- * So our first page loads faster, and it doesn't take much time becoz bundles are comparatively less to download now.
- * Basically your application size shrinks and it becomes faster.

Good Optimization Can be achieved with Lazy Loading .

→ Notes to add

- * you can read about Preloading Strategy
- * dynamic importing of Components and render using Component Factory for better Angular Build Optimization