edWisor

Angular - Setup the application

- Aditya Kumar
Chief Technology Officer, edwisor.com

We will cover with the following concepts

1) Planning the application
2) Modules Setup
3) Basic Components setup
4) Setup Basic routing

Planning the application is important.

User management part

- Login Page
- Signup Page

Chat management part

- Main view to load all online users and unread chat messages
- Single user chat view with option to load older chats

Modules are primarily of 3 types

1) **Root Module** - The module that is created by default. It is the entry point of the whole application.

2) **Features Module** - As your application grows in size, you may need to bifurcate it into more than just one module for better code control. That's where you will have to create Features Modules.

3) **Shared Module** - The idea behind shared modules is to organize the commonly used pieces of the application into one module and then export that module, so that other modules in the application can use it.

let's create our modules - Chat and Shared module using the command - ng g module <name>

Deciding on how to break modules can be tricky

Rule - Separate the modules based on the utility of the components and the functionality to which they belong.

Ecommerce

- User management
- Shopping management
- Orders and Purchase management

Ride hailing app

- User management
- Booking management
- Payments management

Social Network

- User management
- Feed and posts management
- Chat & message management
- Other functionalities - birthdays, events etc.

Notice the difference in the modules we generated?

1) Apart from Browser Module, all other core angular modules can be imported into the feature components.

2) BrowserModule must be imported only in the root module i.e AppModule

3) In feature modules, Angular uses CommonModule which exports the needed ngIf and ngFor directives.

4) Behind the scenes the BrowserModule actually imports and exports CommonModule and that is the reason why we are able to use those directives when we use BrowserModule

5) For these feature modules to be functional, they have to be imported into the AppModule.

A bit of difference in routing

1) In our feature module instead of using RouterModule.forRoot, we use RouterModule.forChild.

2) **forRoot** as the name suggests is for only the root module of the application. For every other module, you are supposed to use **forChild** method

3) Behind the scenes, RouterModule works like a service and like we discussed, its better to keep a single instance of a service. So, basically by not calling forRoot again, we are telling angular to keep only one instance and then by using forChild, we are telling it to add these new routes to the existing services.

4) This makes more sense as all the routes, even though belonging to different modules, are at the end part of same application.

The next steps are ...

Work on Signup and Login Components