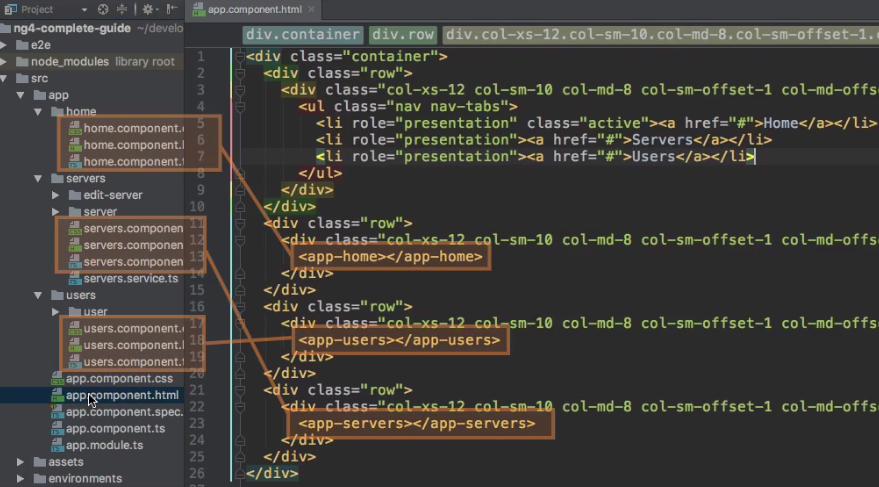
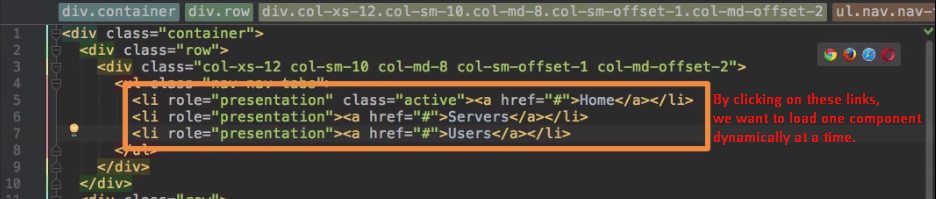
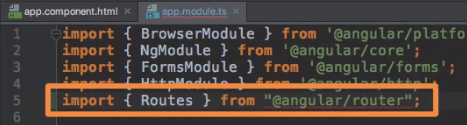
The following is the source code we will work on. Here you can see 3 major components and their sub-components. In the app component we’re loading all components. 

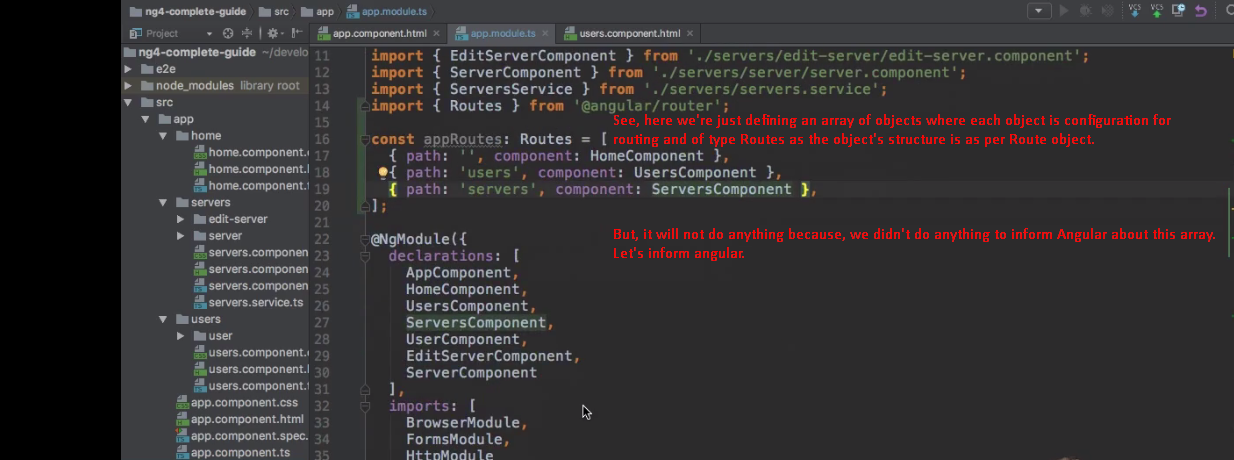
1. **Requirement**: We want to load dynamically only one at a time by clicking on the links given at lines (5,6,7) and we really load a different route in our app. These are **different routes** in our app.  
   
2. Where do we register these **routes**?
3. Let’s think of it logically.
4. These routes are responsible for our whole app. Everywhere in our application, if we enter **/users**, we want to load the users component as our main page let’s say.

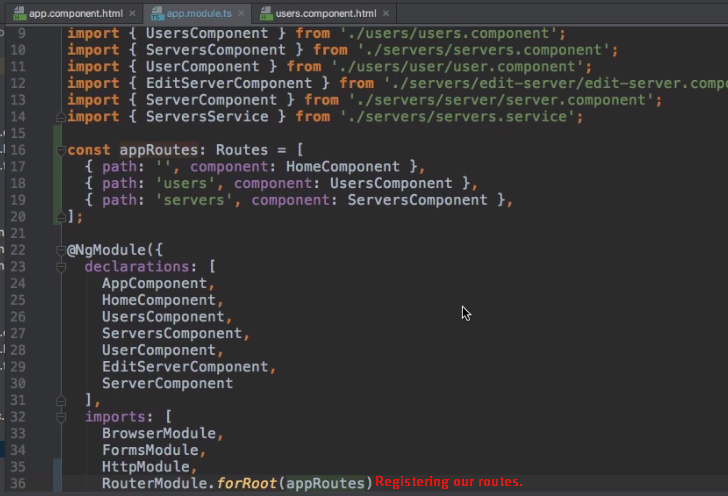
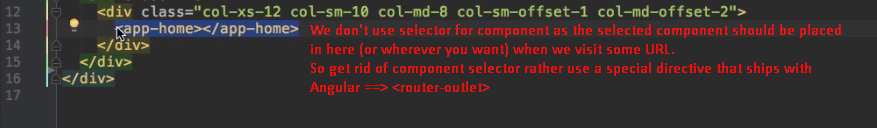
So, as this is really a core part of our app, so where should we register it?

1. Well, **App Module** maybe a good place to inform Angular about our **routes** because here is where we configure our app such as adding our all components.
2. 
3. Let’s **configure the routes**.
   1. const appRoutes: **Routes** = [{r1}, {r2 }]🡸   
      Type🡺 **Routes**🡺 To give routes some structure. (Actually this is for us so that if we provide incorrect type, we can get error at compile type as we’re using typescript).

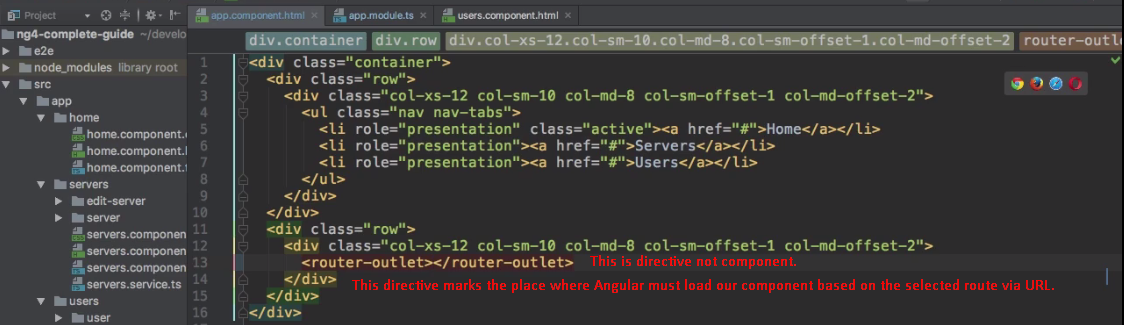
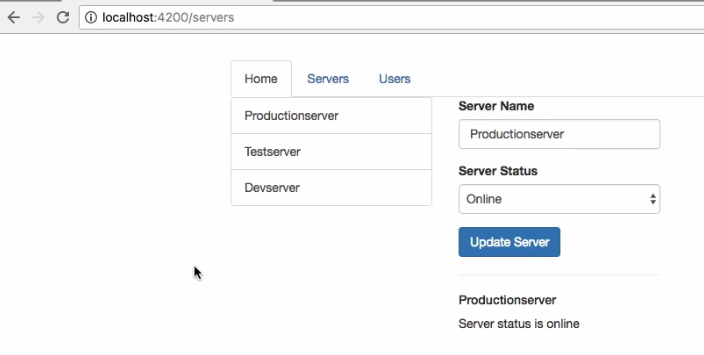
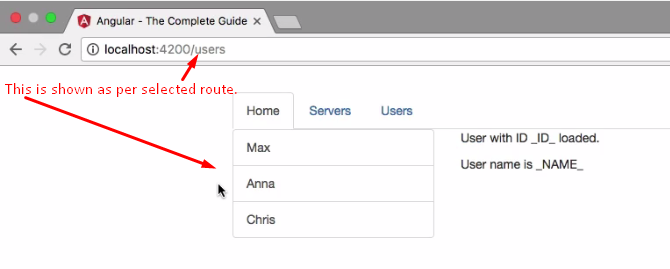
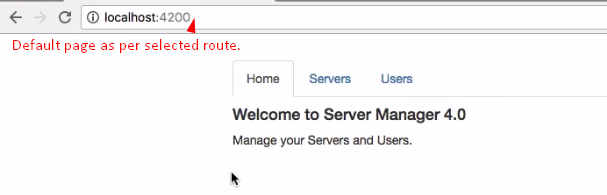
It is an array as we want to configure multiple routes.  


**NOTE**: no slash before the path.  
Now, we also need to tell what should happen when this route is reached. Till now, nothing happens as no action is attached.  
The action is actually a component.

* + 1. **path**: What is entered in the URL after the domain. Tells angular which of the routes is selected identified via the URL
       1. **Example**: path: ‘users’ 🡸 localhost:4200/users
    2. **component**: Tells Angular to load a component when a specific path is reached.  
       

1. **Route Registration:** **Now**, Angular has routing functionalities.  
   With this, now our routes are registered in angular app on this RouterModule which gives us this routing functionality. Now angular knows our routes.
2. **Now** the missing piece is someplace where Angular should render the currently selected component.
3. **Ok,** if we visit “/users”, so Angular knows we want to load the users page but how does it know where to display it.
4. 

It looks like component but it’s directive which marks the place where selected component should be loaded.

1. I
2. **<router-outlet>:**Directive  
   
3. 
4. **NOTE**: Our links (Home, Servers, Users) are not working but when we enter the URL into URL bar, and then they are working.
5. Let’s next to see how to make links work.