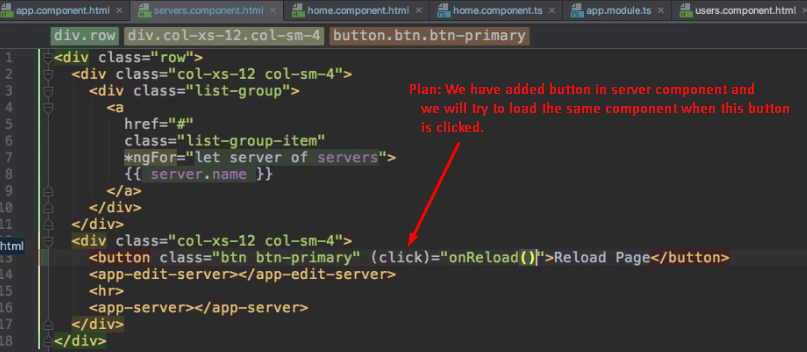
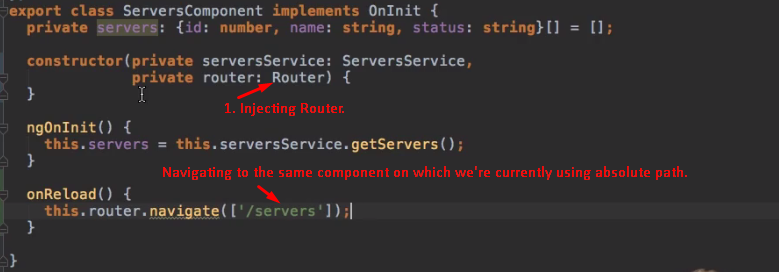
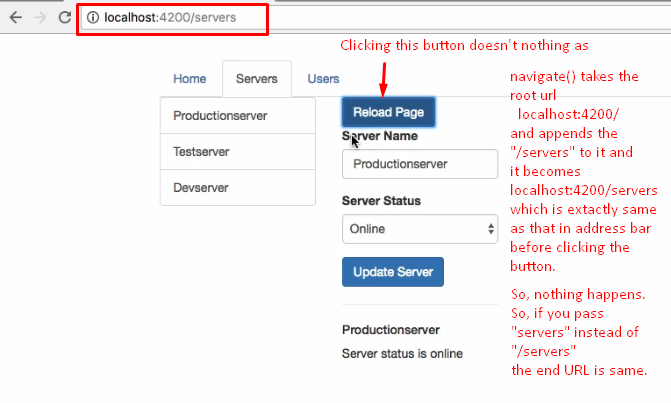
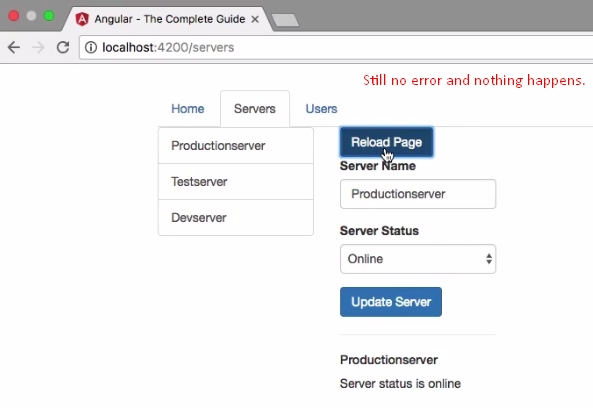
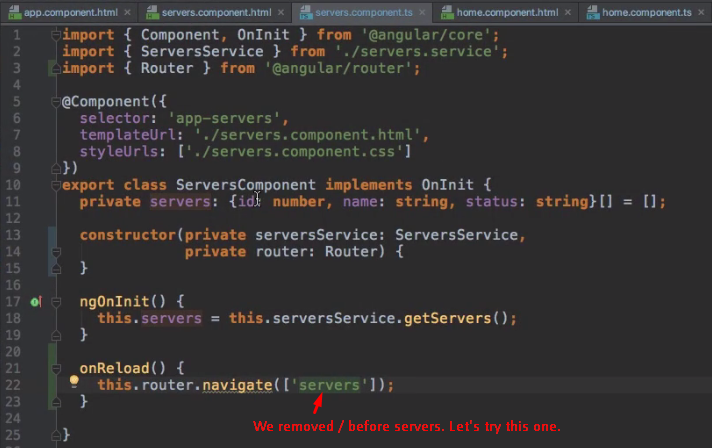
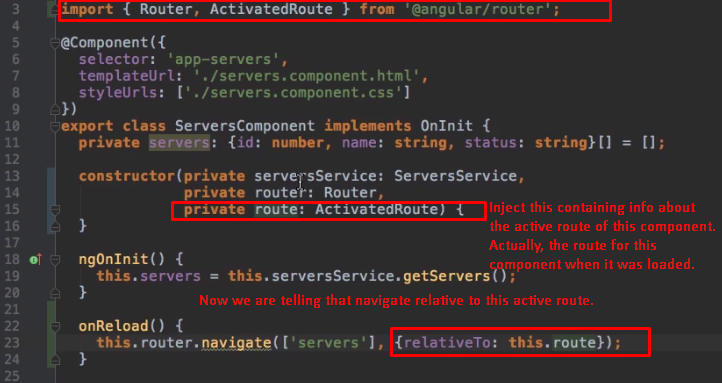
1. In last lec, we navigated relatively.   
   **Agenda**: Let’s see how to navigate using relative path.
2. 
3. Let’s give its handler to navigate to a different route.   
   
4. 
5. 
6. This is strange because we got an error when we used routerLink.   
   But why doesn’t it work when we use navigate method?
7. Unlike the **routerLink**, navigator method doesn’t know on which route you’re currently on. The **routerLink** always knows in which component it sits ( In which component’s template) and therefore it knows what the currently loaded route is.

But we can argue that here we also call this inside the component TypeScript code, so it could determine where it is in but it is just not how it works.

So, to tell this where we currently are in, we have to pass a second argument to the navigate method which is a JS object. Here we can configure this navigation action. We will come back to this later as there are more things we can add but on configuration is the **relativeToProperty** here.  
Here we define relative to which route this link should be loaded and by default, this is always the root domain.

1. We can change this default behavior. 
2. 