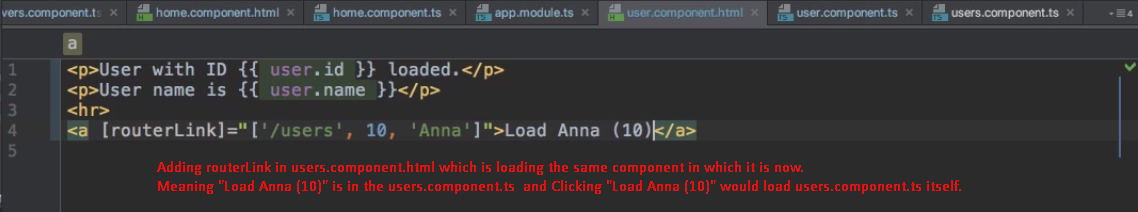
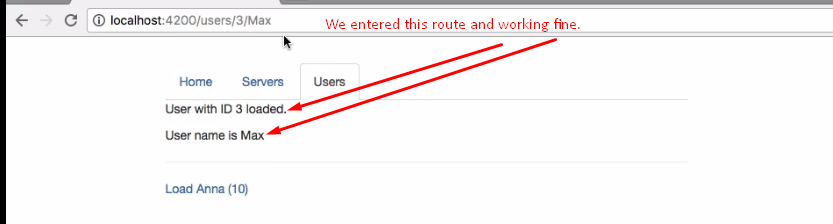
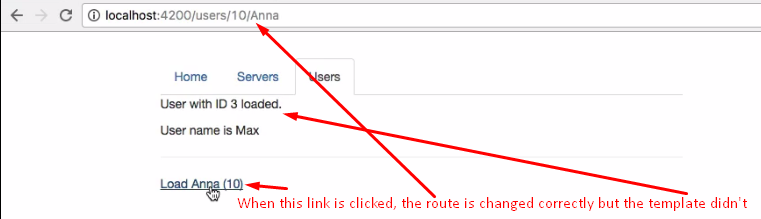
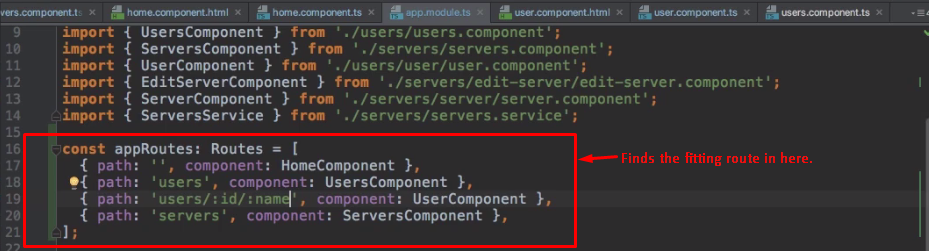
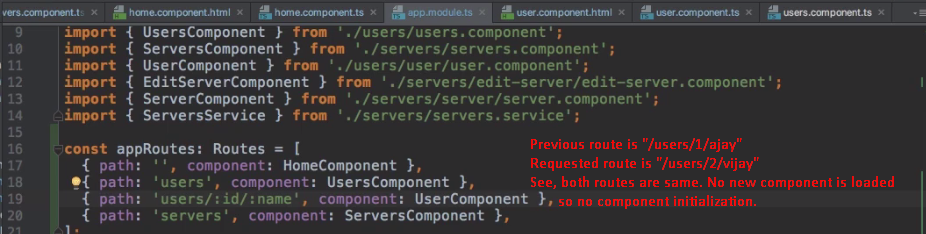
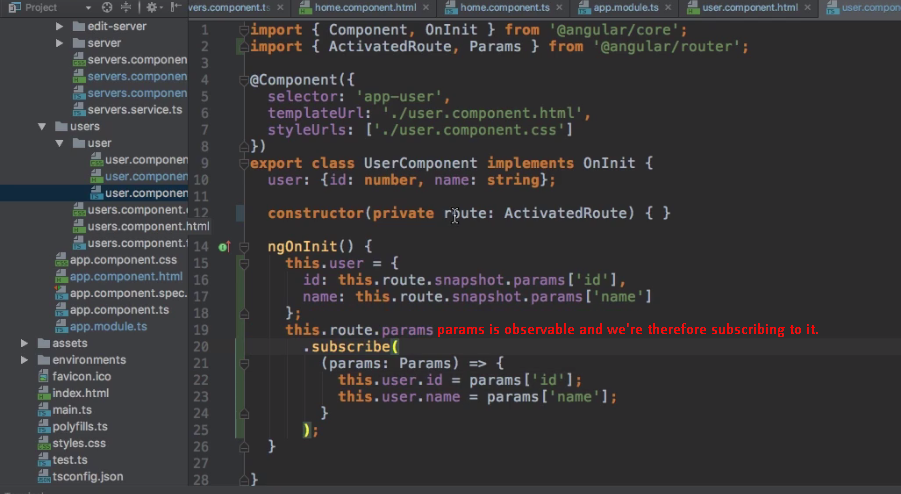
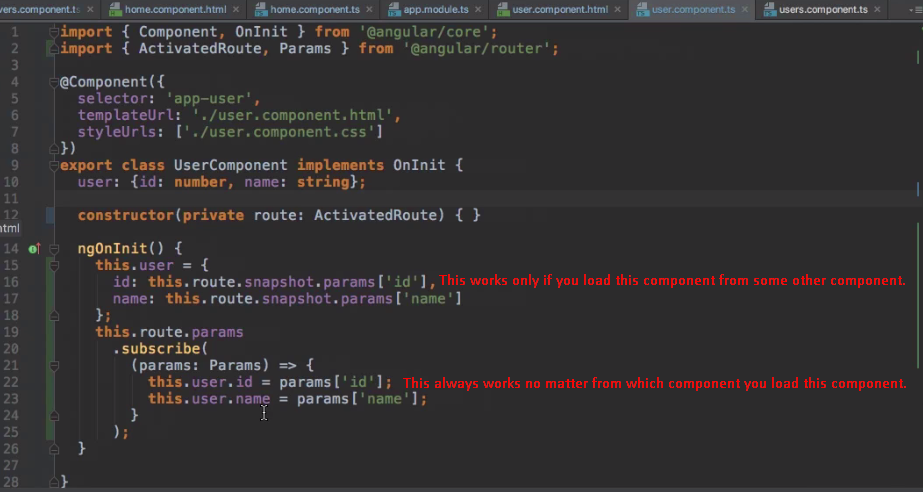
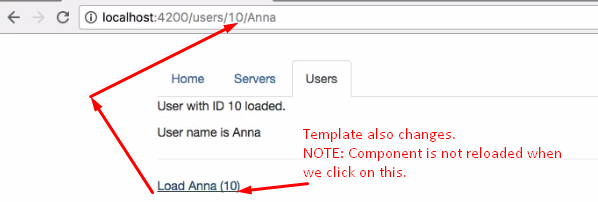
1. In the last lecture, we saw how to **retrieve route parameters** and it was working fine.
2. But there are places where this break. We will see a typical example.
3. In the list lecture, we learnt how to retrieve route parameters and this was working fine. But we can break this working example.
4. Let’s do that.
5. 
6. 
7.   
   In the template, the ID must be 10 and name 🡺 10 after clicking on 🡺 **Load Anna (10)**
8. **Why is it so**?
9. Actually, it’s not a bug. It’s **default behavior**.
10. What is happening here?
11. Let’s try to understand the procedure when a new route is loaded.
    1. **Step 1**: Angular looks for the route in the app.module.ts  
         
       If found, then proceed further to load the component.
    2. **Step 2**: Component’s ngOnInit() method calls where we’re accessing the route parameters using **ActivatedRoute.snapshot.params**[‘placeHolderName’];  
       (This is initialization phase where we set the component’s state)
    3. **NOTE: These two steps happen only if we’re not already on the fitting route (The current route making the request is not the same as that route for which the request is being made). **
12. If same route is loaded, Angular doesn’t re-instantiate this component as it costs us performance. Why would we re-render that we’re already on it. We can say because the data changed but Angular doesn’t know. And it’s good that it doesn’t recreate the loaded component and destroy the old one if we’re already on that component.
13. But still, you may want to access the **updated data** and you can.
14. **Solution** 🡺**this.ActivatedRoute.params**🡺 **Observables are features** added by 3rd party package, not by Angular but heavily used by Angular which allow you to easily work with **asynchronous tasks** and this is asynchronous task because the parameters of your currently loaded route might change at some point in the future if the user clicks this link **Load Anna (10)**. But, you don’t know **when, you don’t know “if”, & you don’t know how long it will take.**   
    So you can’t block your code and wait for this to happen because this might never happen.  
    **Observable: It is an easy way to subscribe to some event which might happen in the future, to then execute some code when it happens without having to   
    wait for it now.  
    subscribe() can take 3 functions as arguments.  
    1st:** This first function is executed when new data is sent through this observable (Whenever the parameters change).  
    (params: Params)🡺{}🡸 Params would contain the properties you defined in the route as properties.
15. **The following arrow function would not be run when ngOnInit() is run but subscription would be set up.**
16. 
17. 
18. **NOTE**: If you’re sure that the component would not be reloaded from within that component as we’re doing then you might not need this addition. You might use simply snapshot property because you know this component would be 100% recreated whenever we reach this component as there is no way to reach this component while being inside this component.