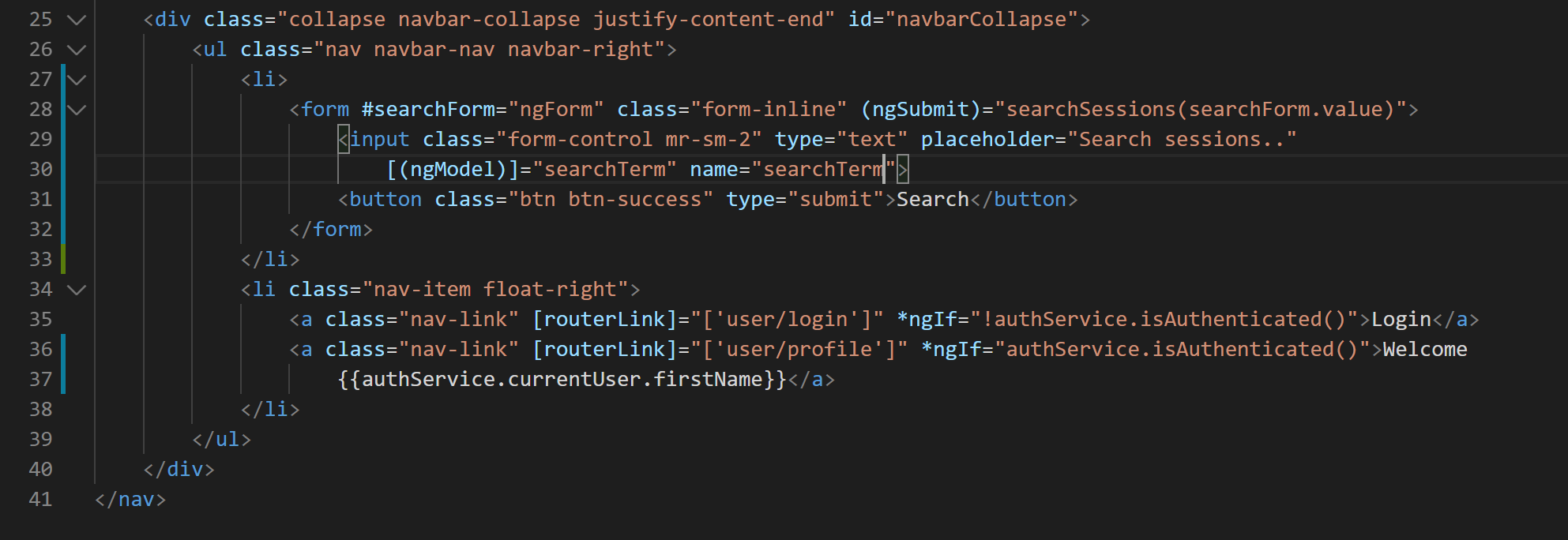
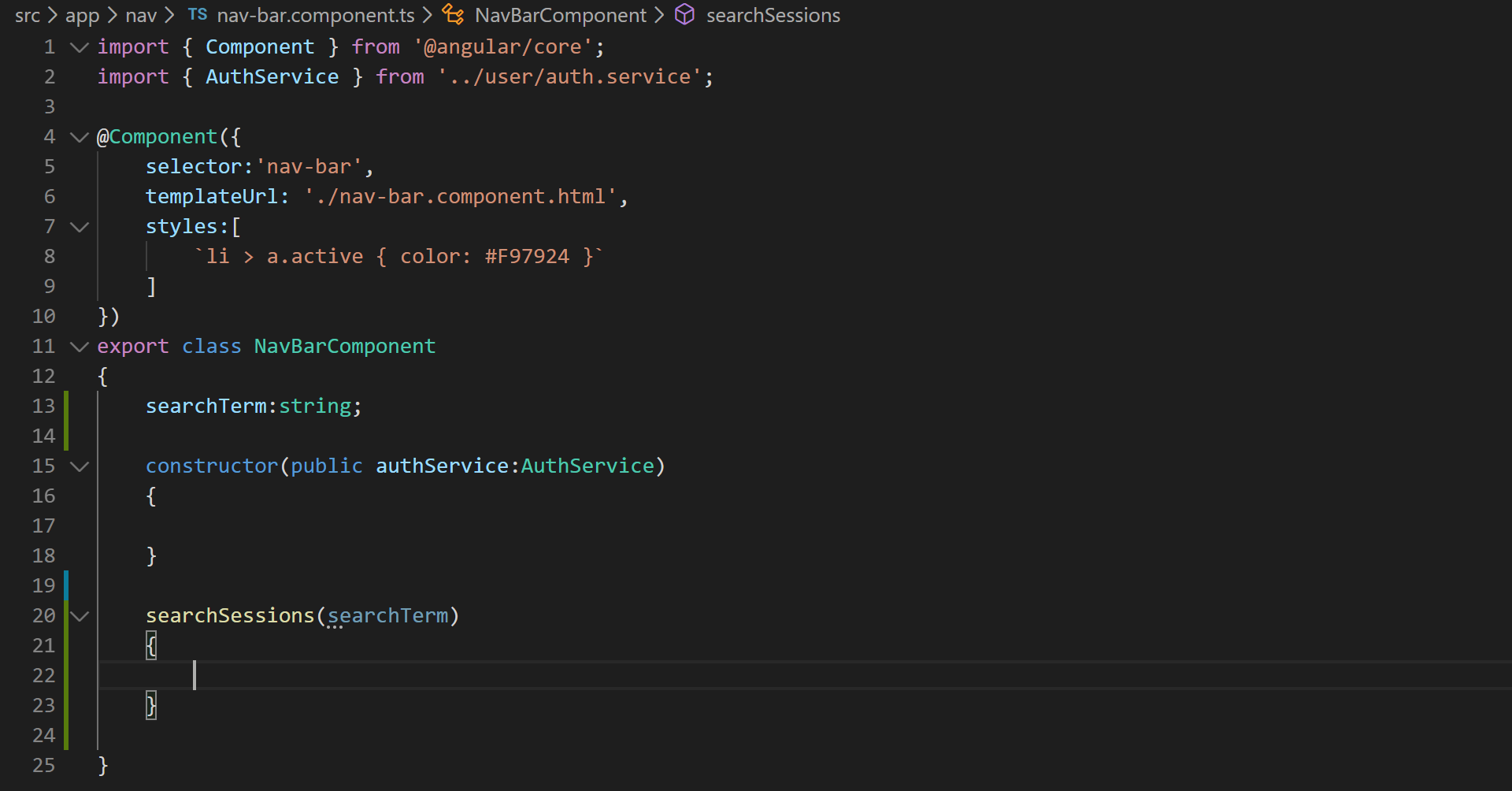
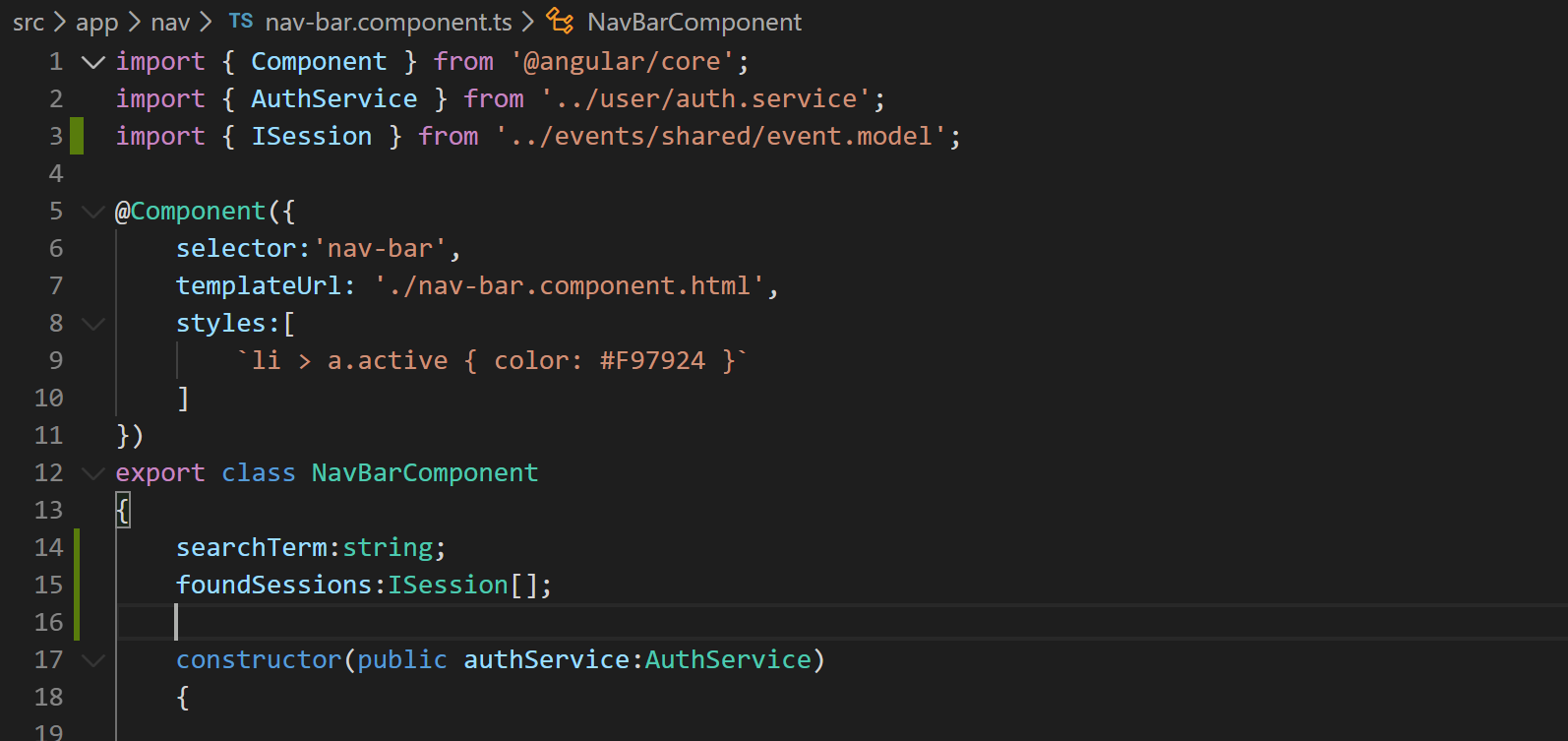
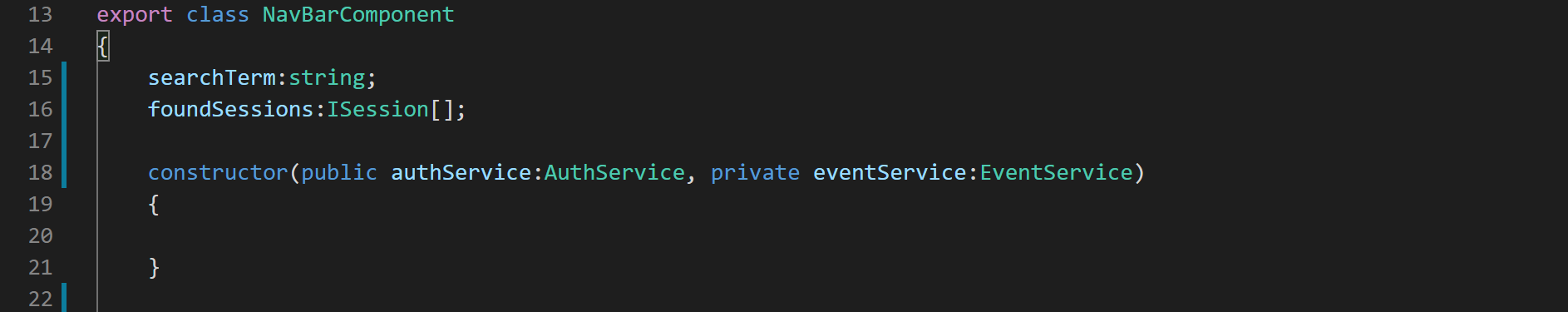
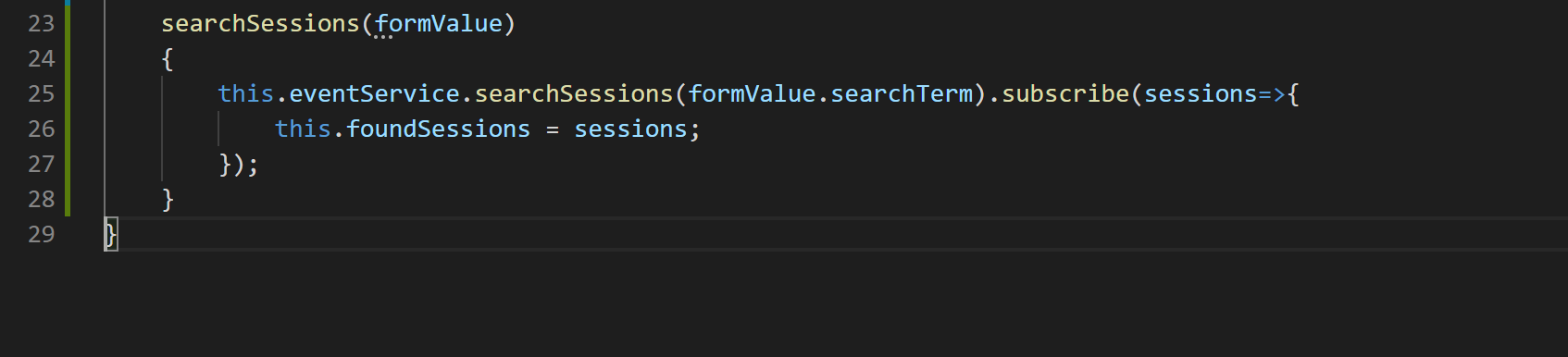
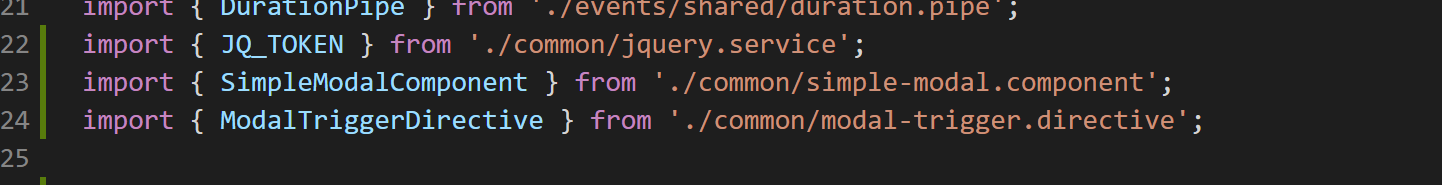
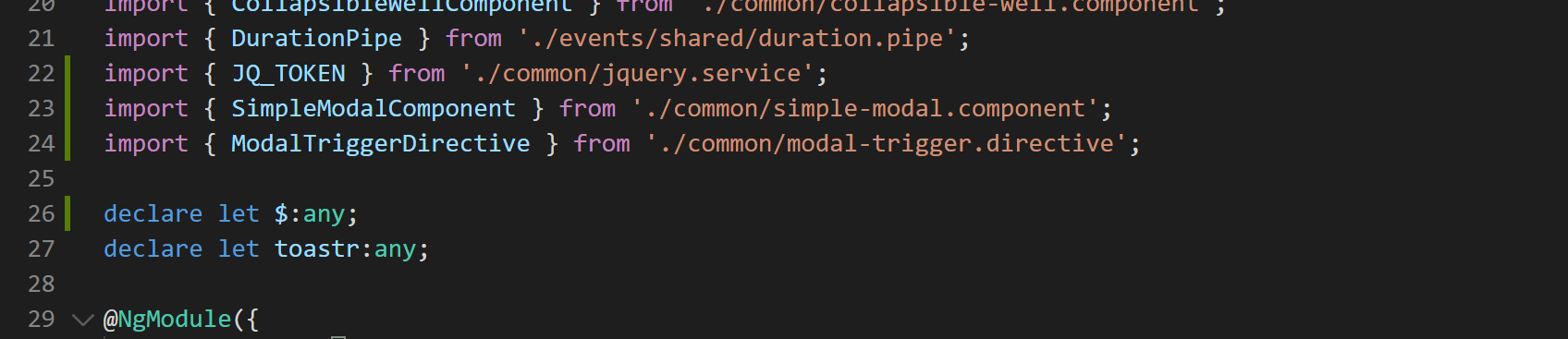
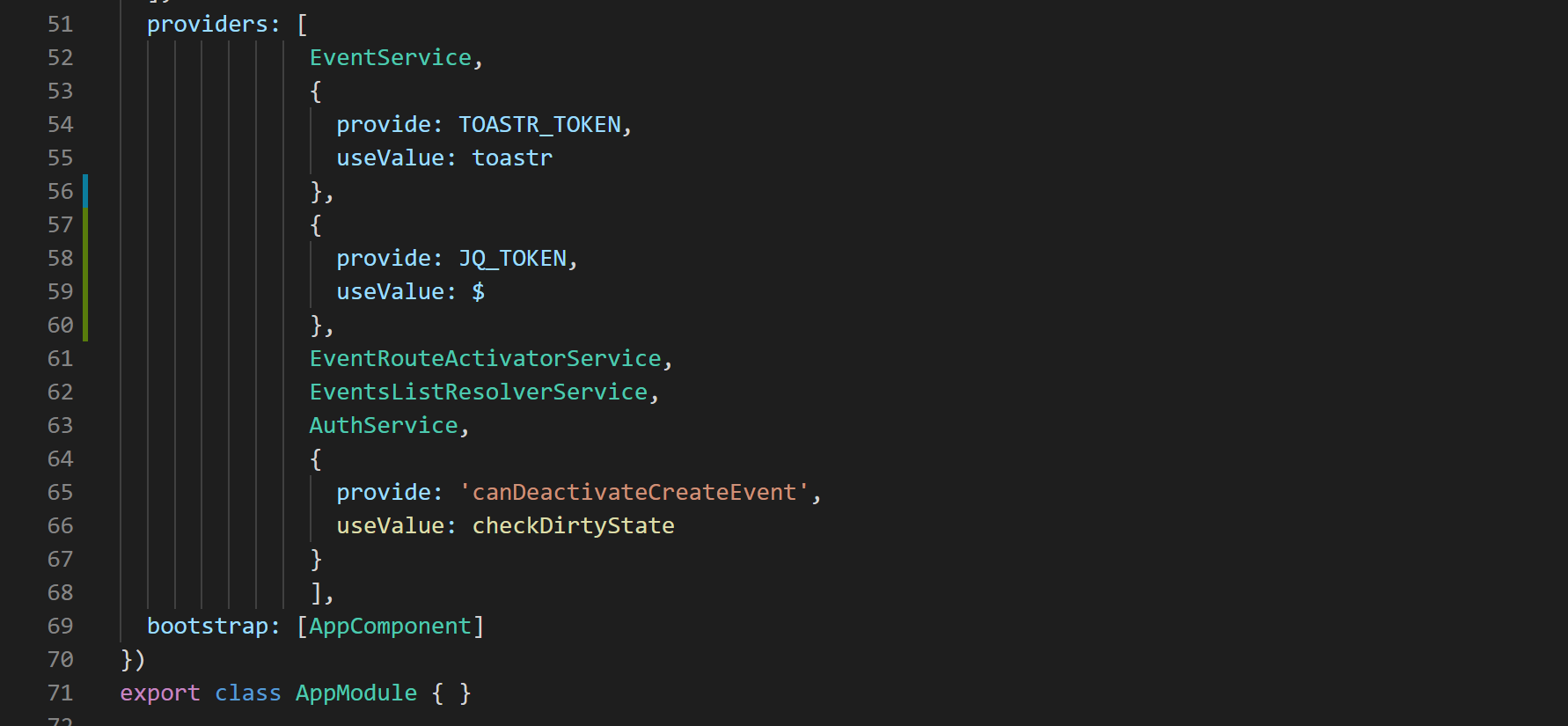
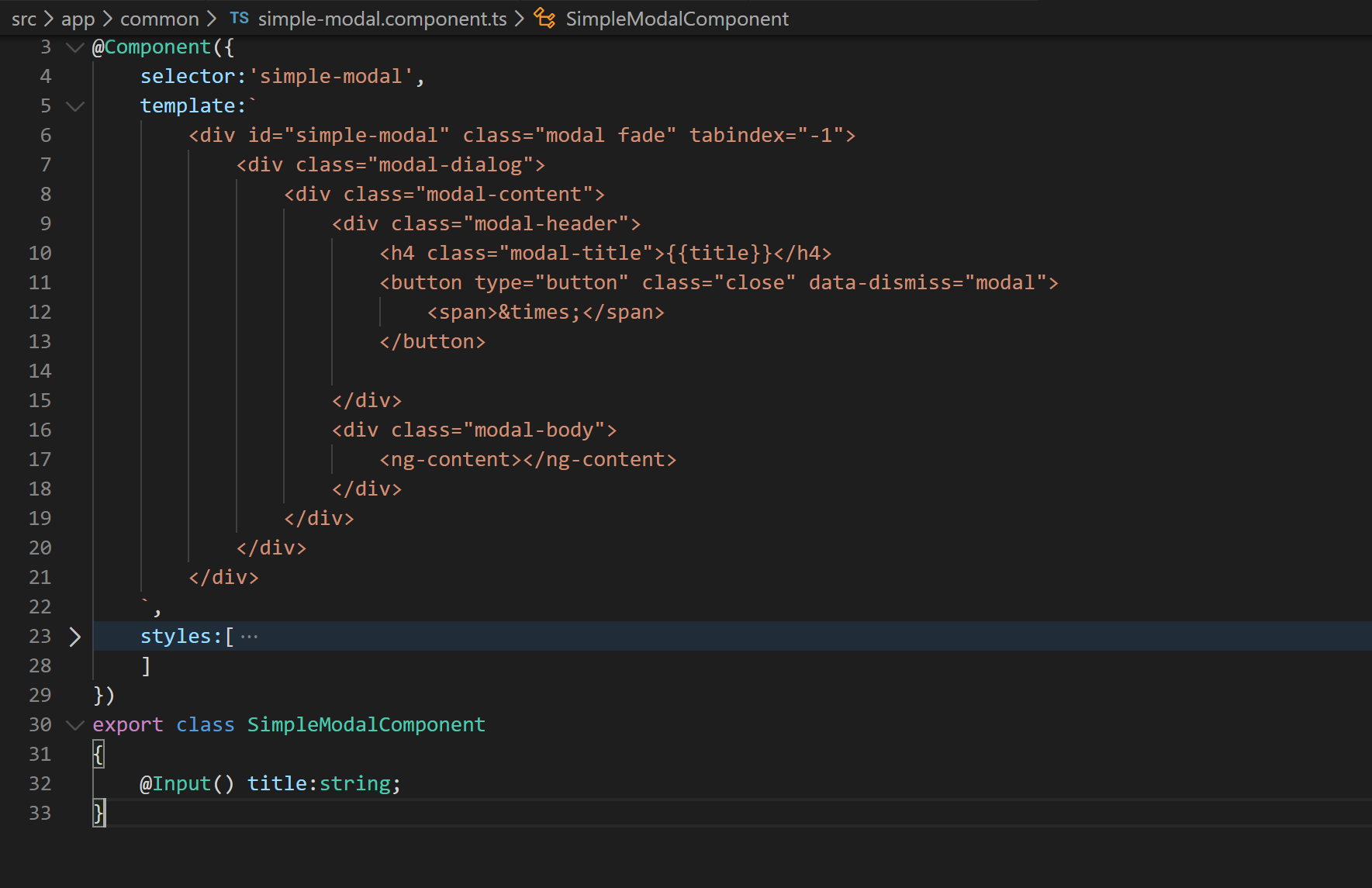
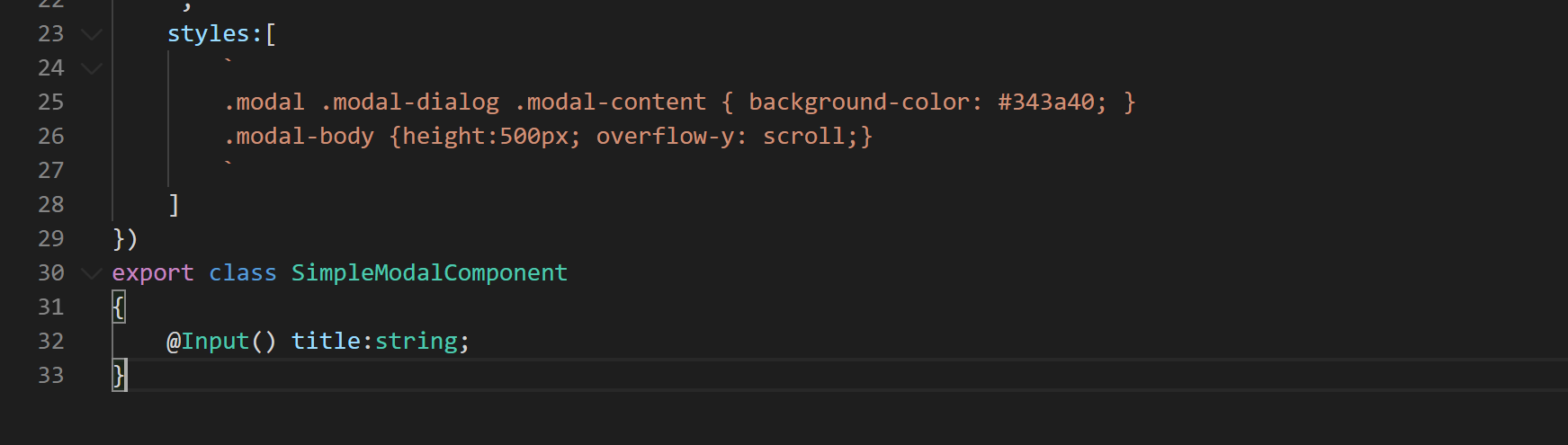
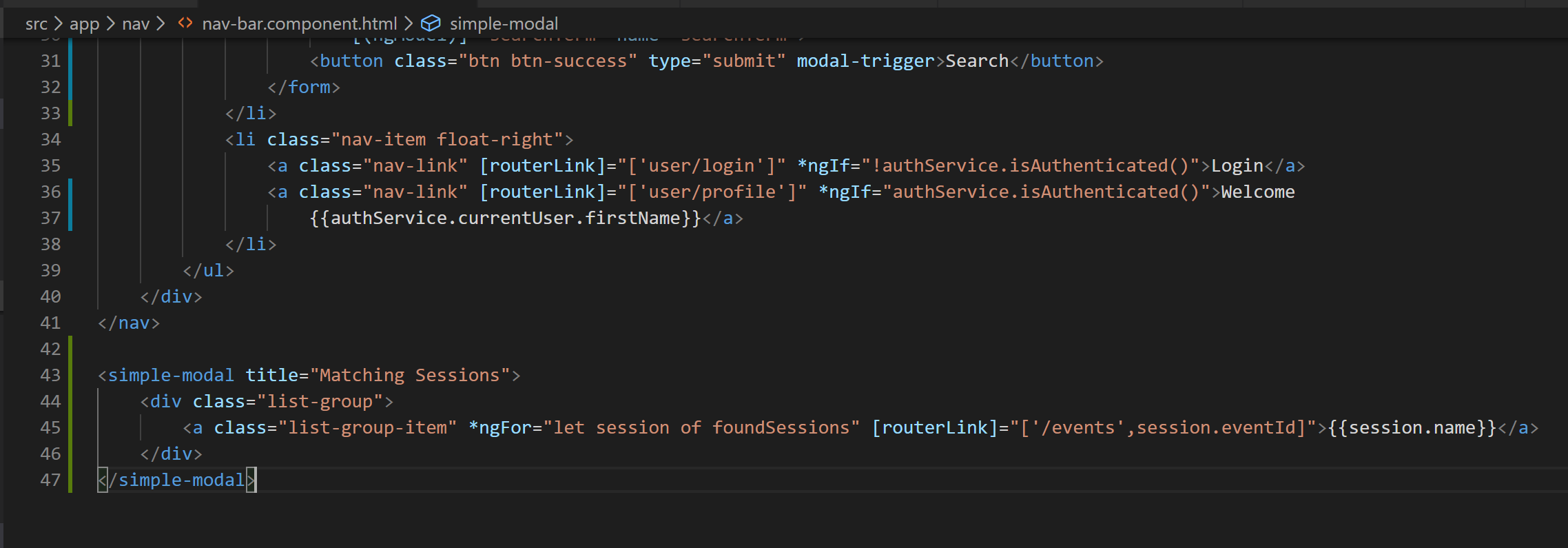
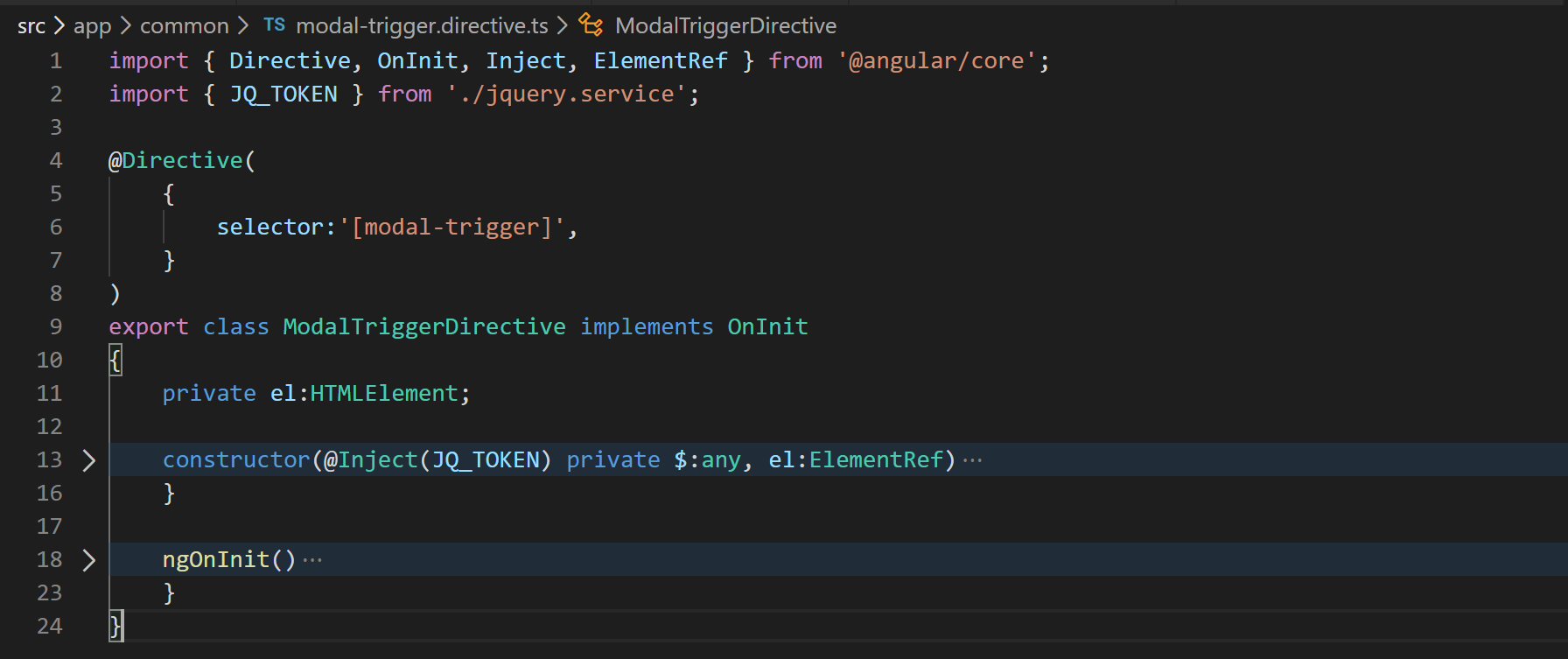
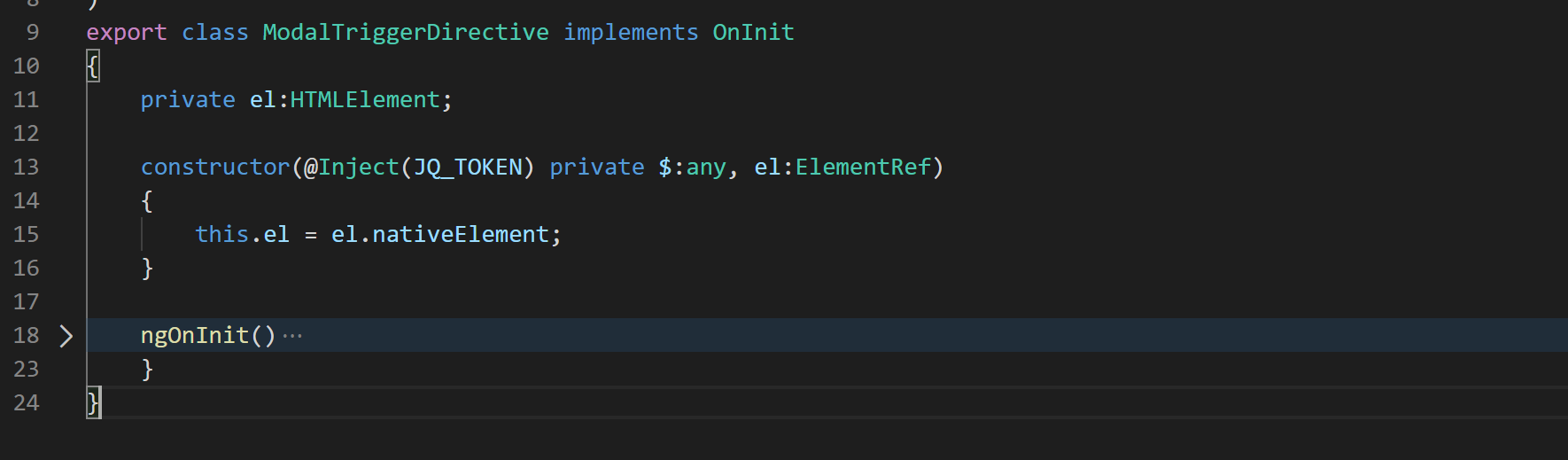
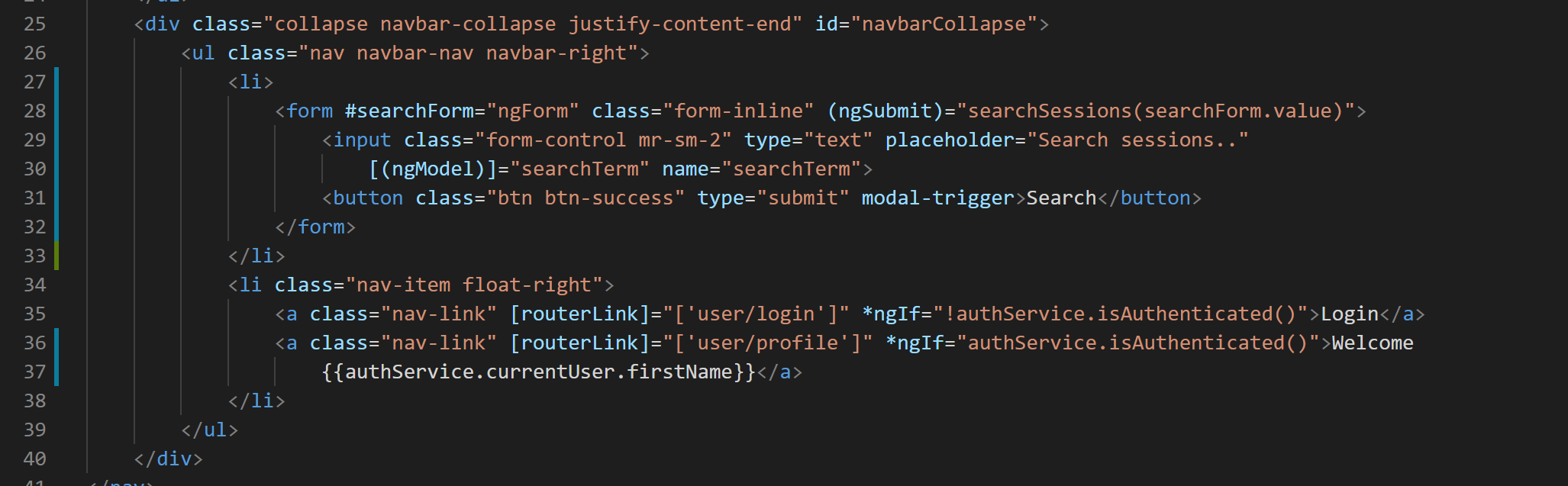
1. Now we will implement the search functionality available on the navigation bar. And while we implement that we will learn a few concepts like bootstrap modal dialog boxes etc. We want users to be able to search sessions not events. When they search for a session a list of search results will be popped in a modal dialog and on clicking a session result user will be redirected to event details page.
2. So lets start by hooking into the event in navigation bar HTML template like below:
3. 
4. Lets create the property and the method on the nav-bar.component.ts like below:
5. 
6. Now lets add a property of type ISession which will hold the resultant sessions like below:
7. 
8. We will implement the search logic in event.service.ts so lets inject that in our constructor like below:
9. 
10. Now lets go to “event.service.ts” and implement the searchSessions method like below:
11. 
12. Now lets call this method in our “nav-bar.component.ts” class in a method called “searchSessions” like below:
13. 
14. We are now ready to render the search results in a modal dialog:
15. The modal window will be shown using Bootstrap but that is implemented using jQuery so we need to bring jQuery in our application now. So we will have to create an injection token and inject jQuery in our application so lets create a file called “jquery.service.ts” in “common” folder.
16. Now lets create the injection token object like below:
17. 
18. Lets now import this in our app module like below:
19. 
20. Let us now declare the global jquery object like below:
21. 
22. We are now ready to create the mapping in providers array in app module like below:
23. 
24. We are now ready to create our modal component. So lets create a file in “common” folder called “simple-modal.component.ts” and define the basic component shell like below:
25. 
26. Lets grab the HTML for this component from “misc” folder of cloned or downloaded repository and paste it like below:
27. 
28. Ok now lets define some styles like below:
29. 
30. We are now ready to add this component in declarations array of app module like below:
31. 
32. Now that the component is in declarations lets try to use it in our nav-bar component HTML template like below:
33. 
34. Ok we want to initiate this modal when user clicks the search button. There are two ways we can do it:
    1. Inside the searchSessions method in nav-bar component TS file inside observable subscription after assigning the result to foundSessions property.
    2. Create a directive and in that directive bind the “click” event of the “Search” button to a handler method and that handler method will initiate the modal.
35. The second approach is preferable because it keeps the nav-bar component reusable enough. If you use first approach then cannot reuse the nav-bar component elsewhere as it will always try to initiate a modal on “Search” button click. So lets use second approach and create the directive.
36. Right click the “common” folder and create a file called “modal-trigger.directive.ts”.
37. Define the basic directive shell like below:
38. 
39. Now we need to use jQuery to fire the modal so lets inject jQuery like below:
40. 
41. As you have seen already we also need reference to an Angular class called ElementRef. This ElementRef class gives us access to the HTML element on which we will apply this directive. So inject ElementRef as well.
42. Now create a property on the directive in which we will store the reference to the HTML element. This property as you can see above is of type HTMLElement. So lets define that.
43. Now assign the “nativeElement” property of ElementRef class to our “el” property. This “nativeElement” property gives access to the HTML element.
44. Now in the ngOnInit method we will bind the click handler to fire the modal like below:
45. 
46. Now our directive is complete. Please add it to declarations array in app module.
47. We can use this directive now on the “Search” button in the nav-bar HTML. So open file “nav-bar.component.html” and on the “Search” add the directive selector as an attribute like below:
48. 
49. Alright now if you go to application and type ‘angular’ in search box and click “Seach” button it should render the modal with results like below:
50. 