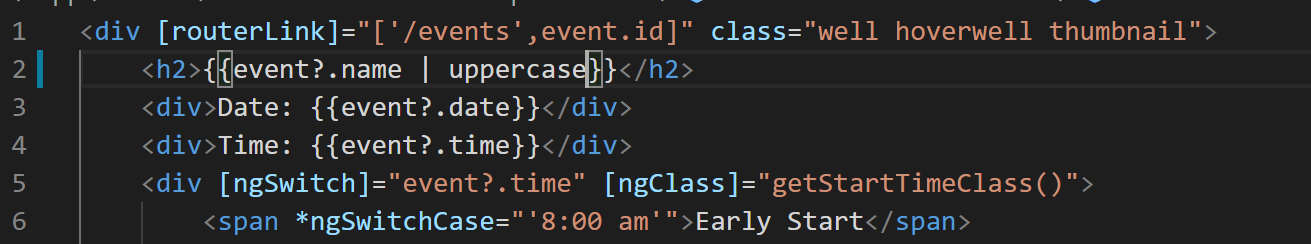
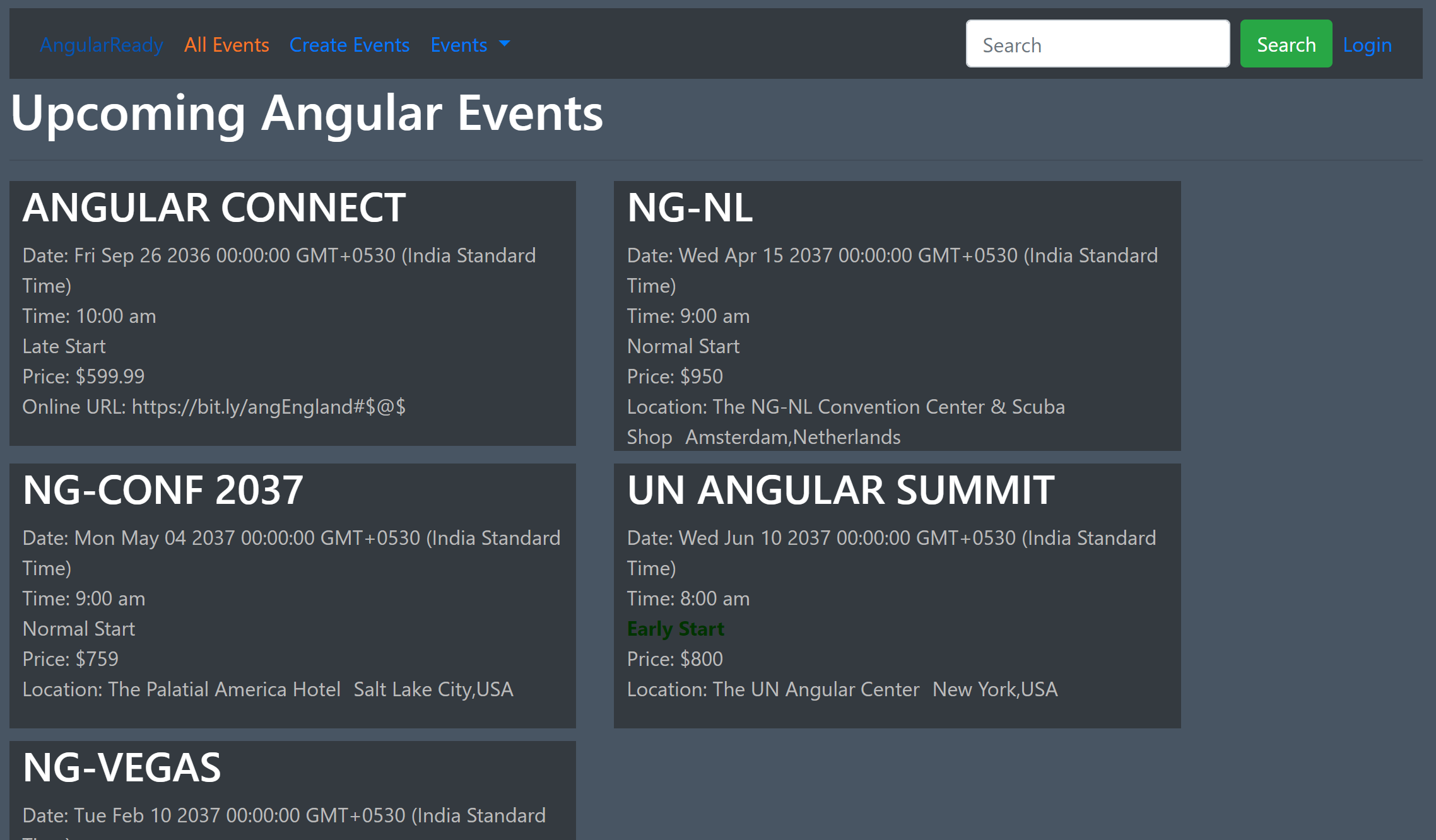
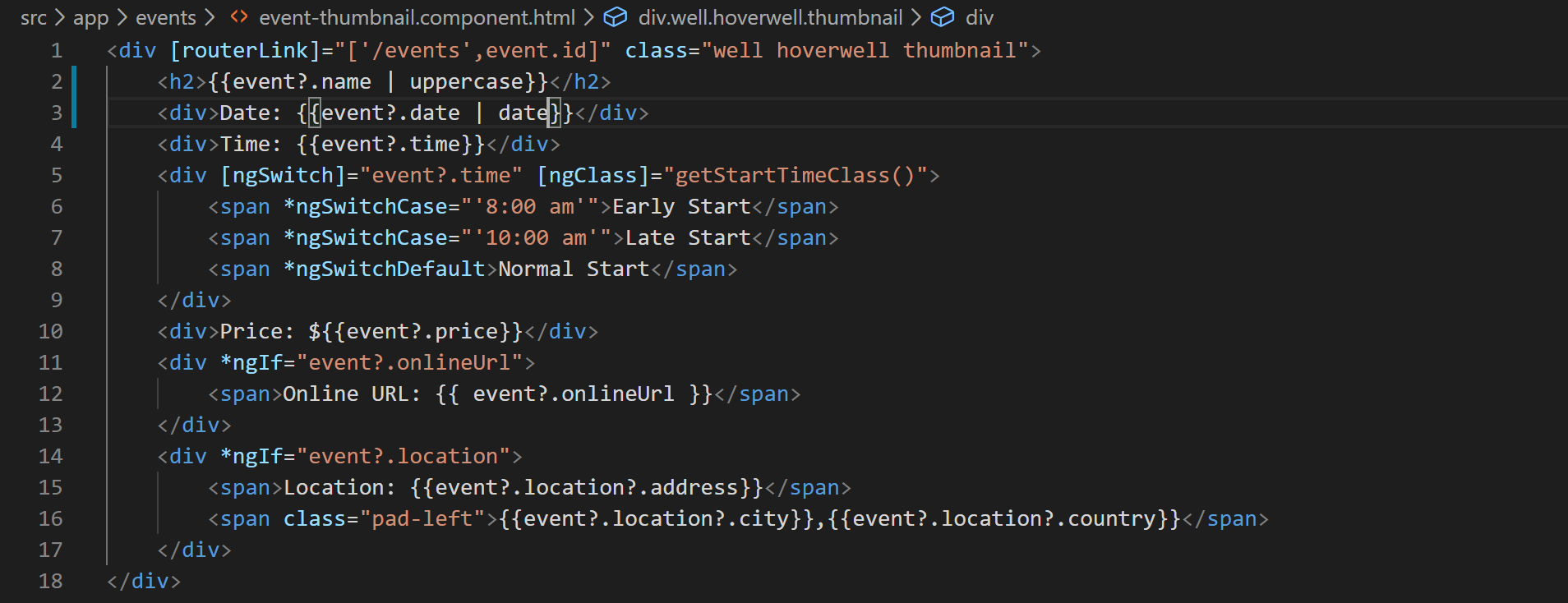
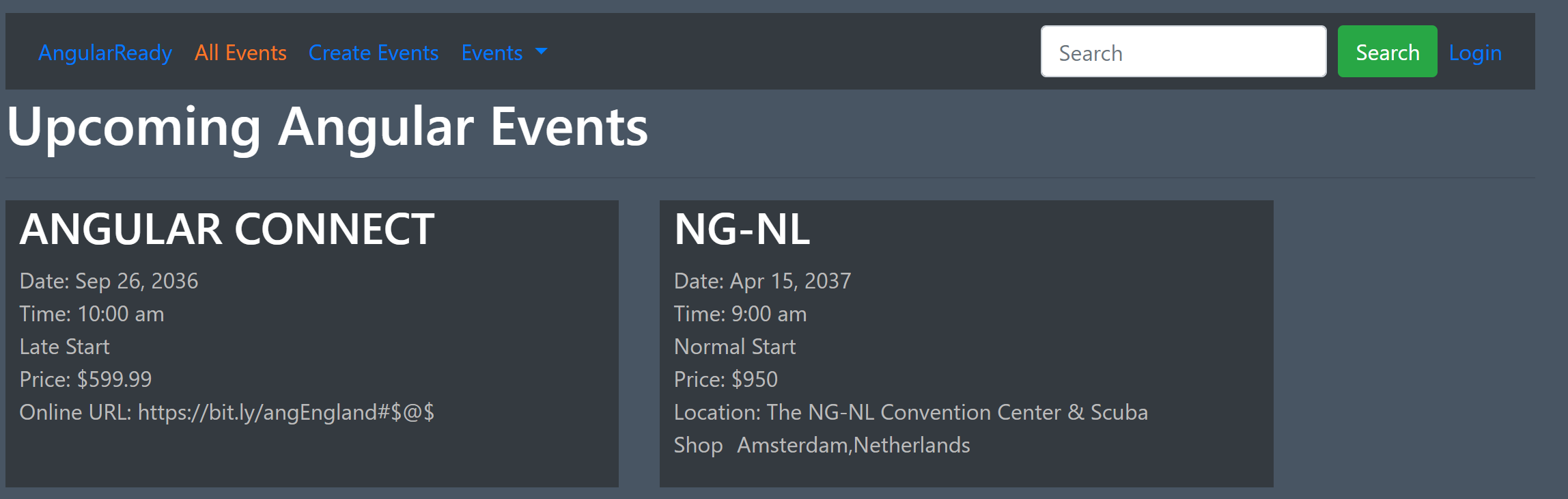
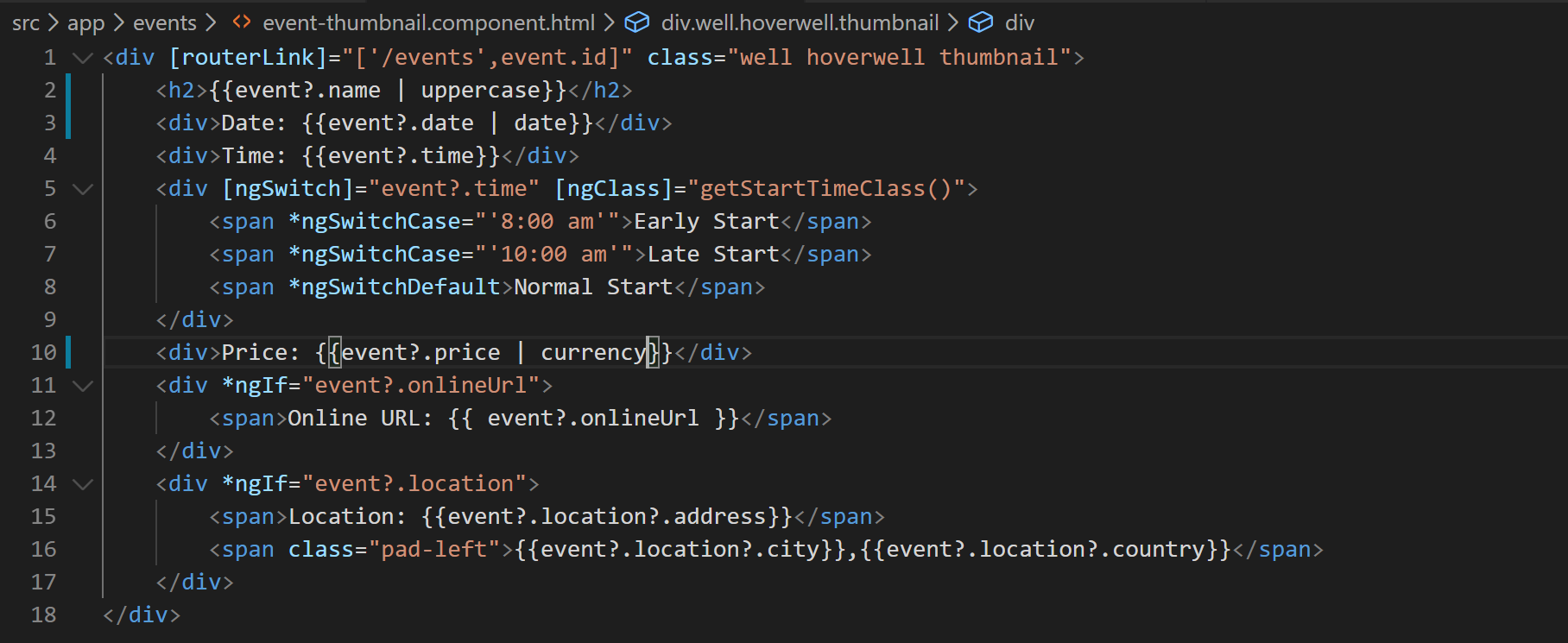
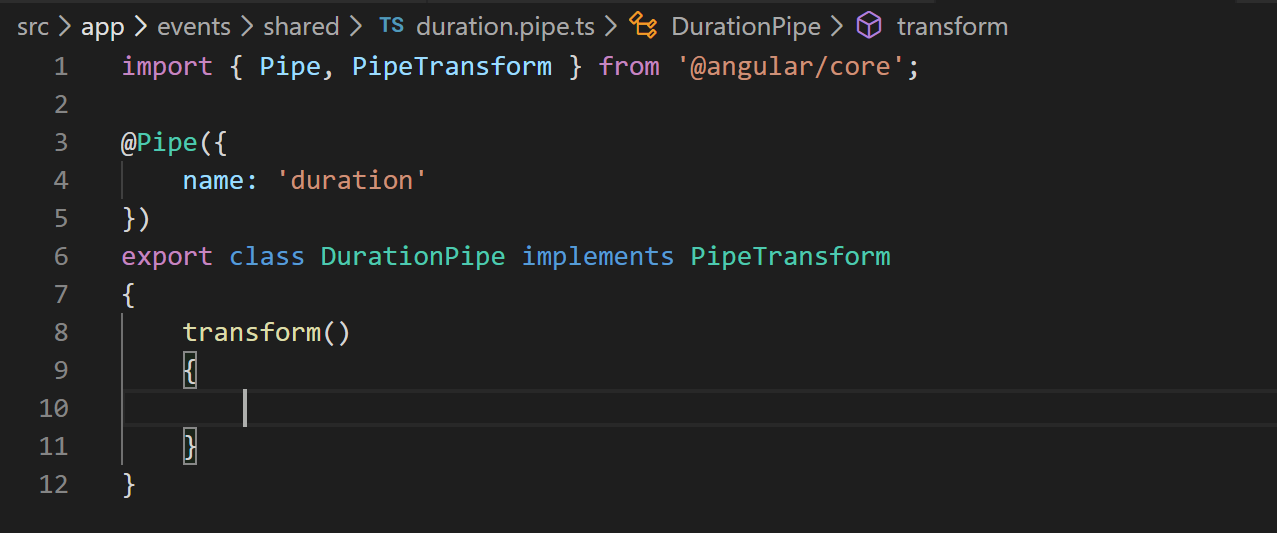
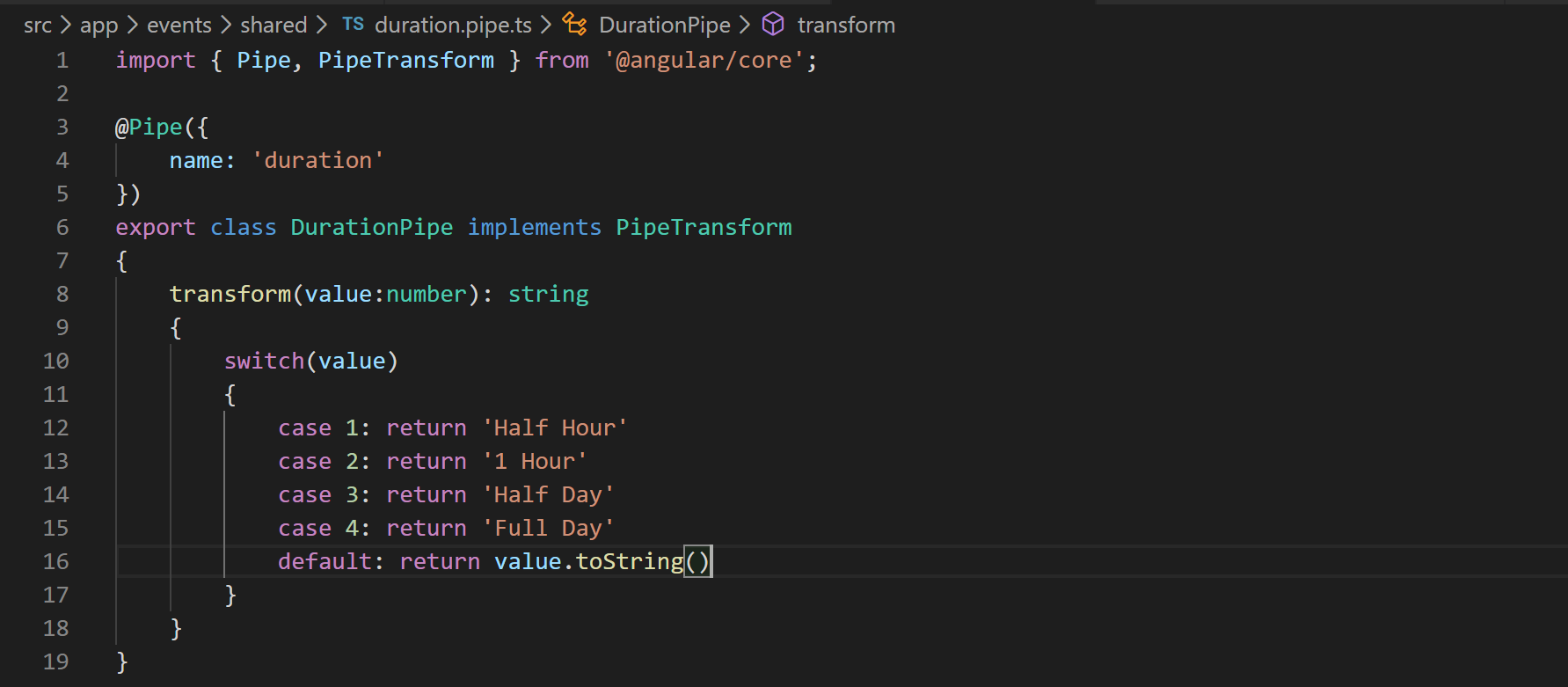
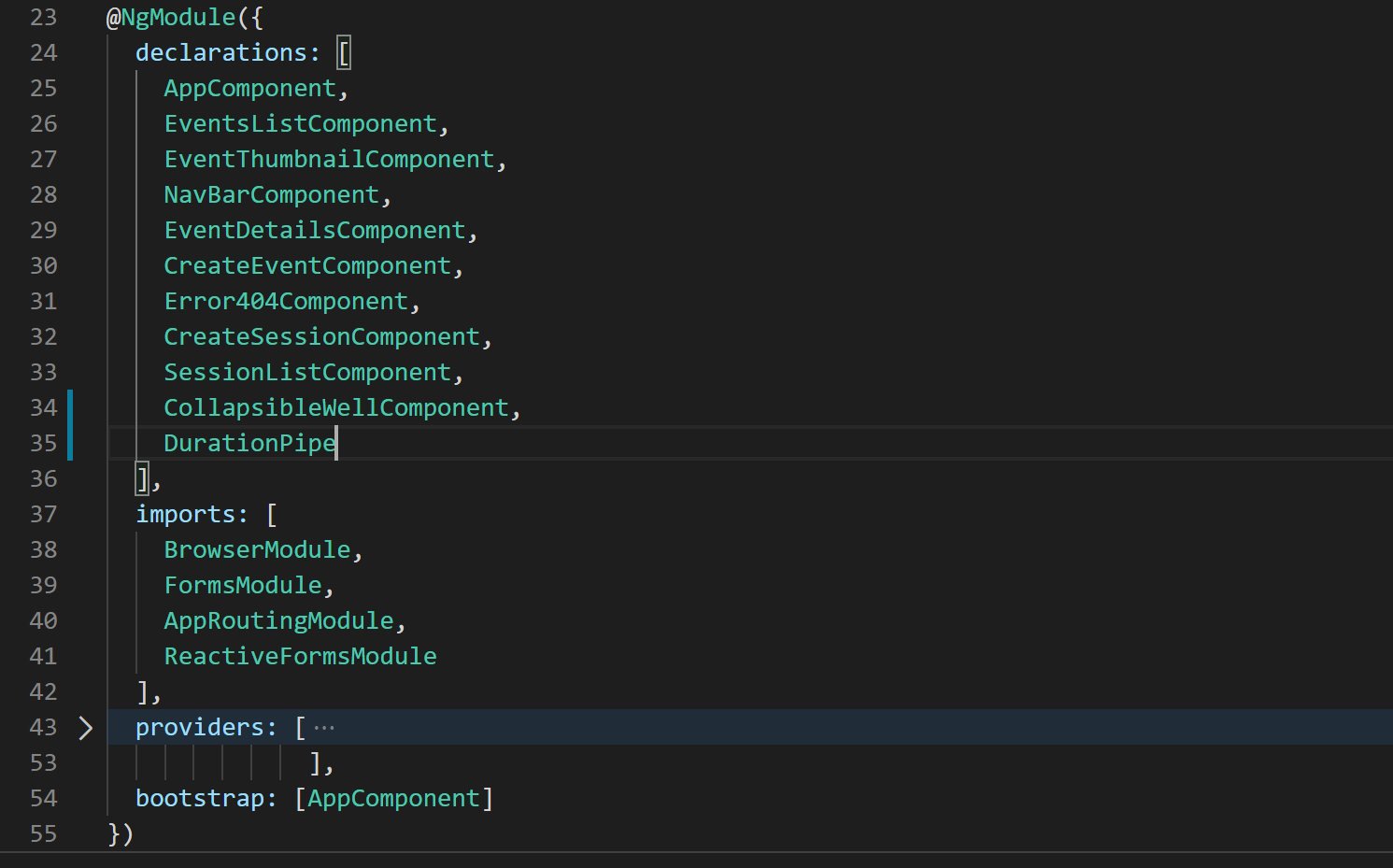
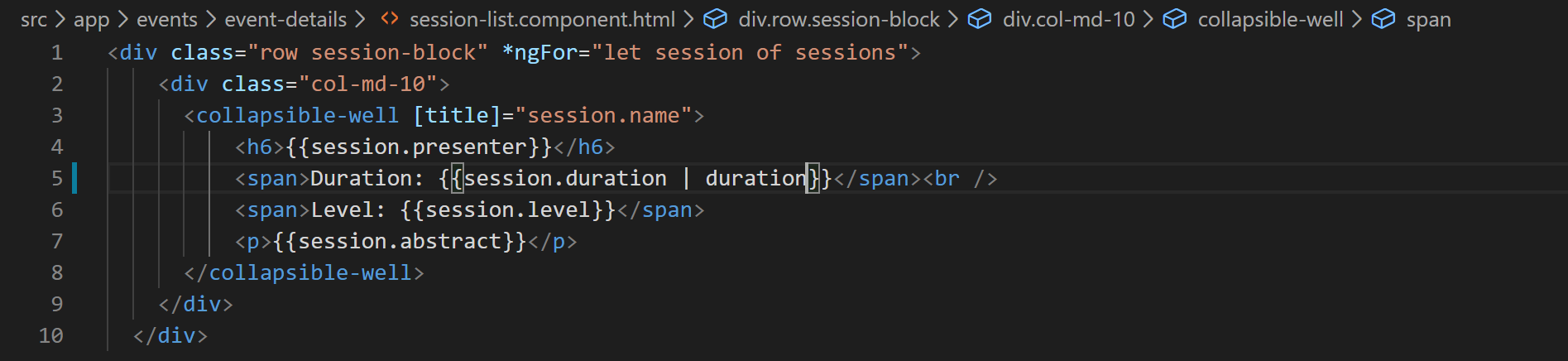
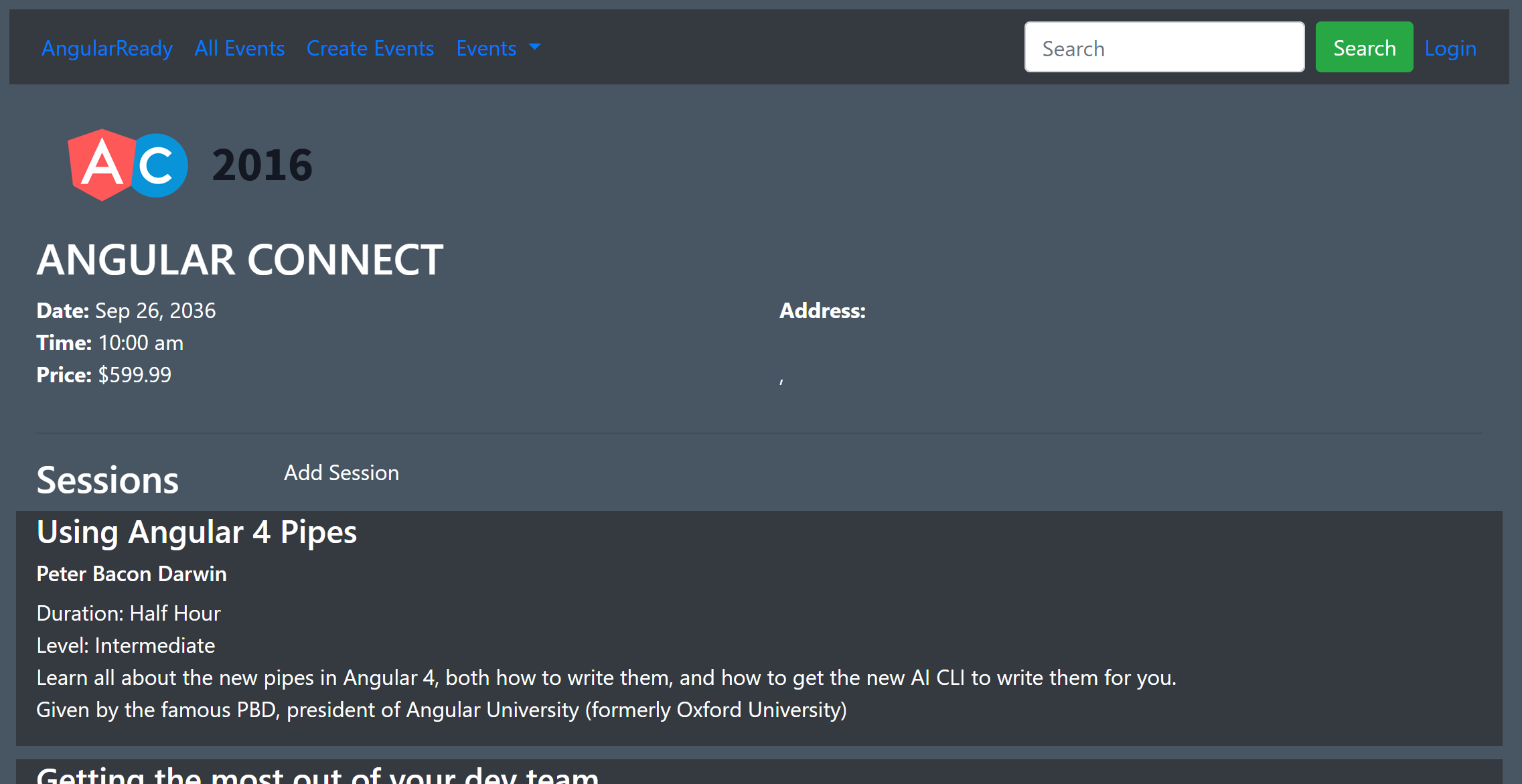
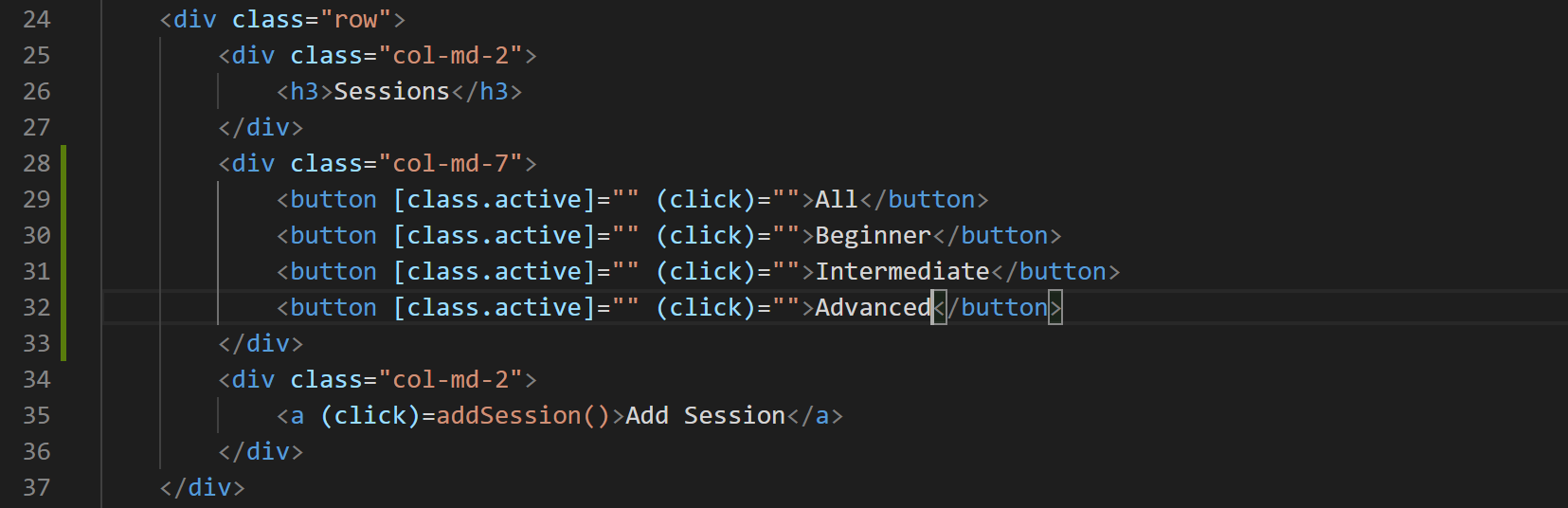
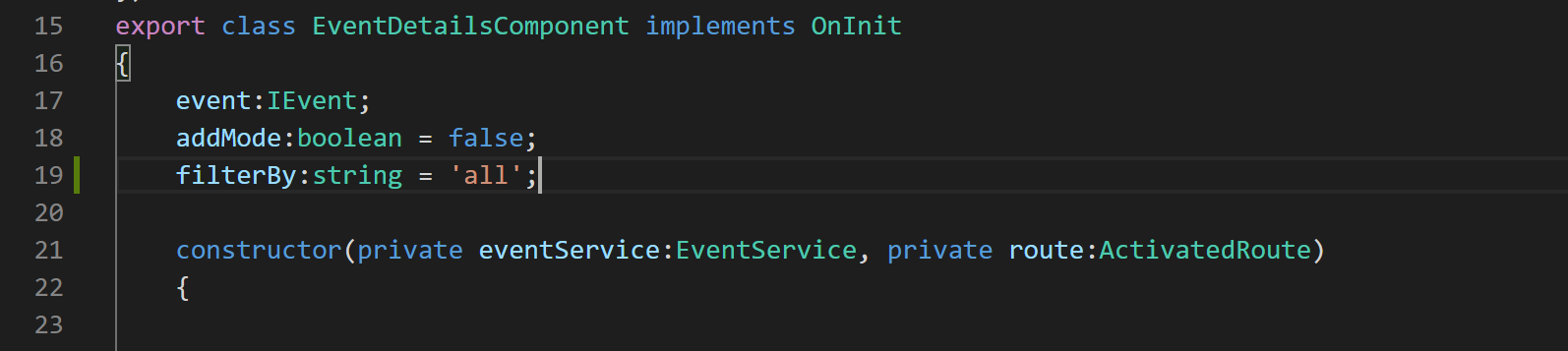
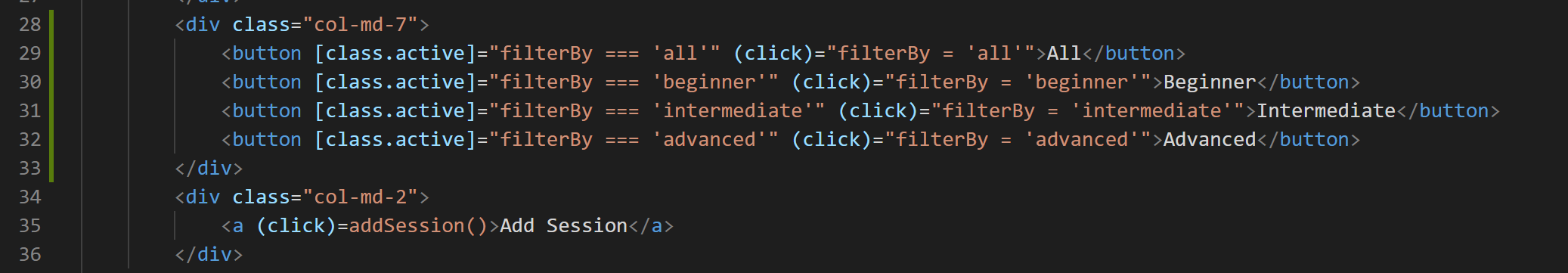
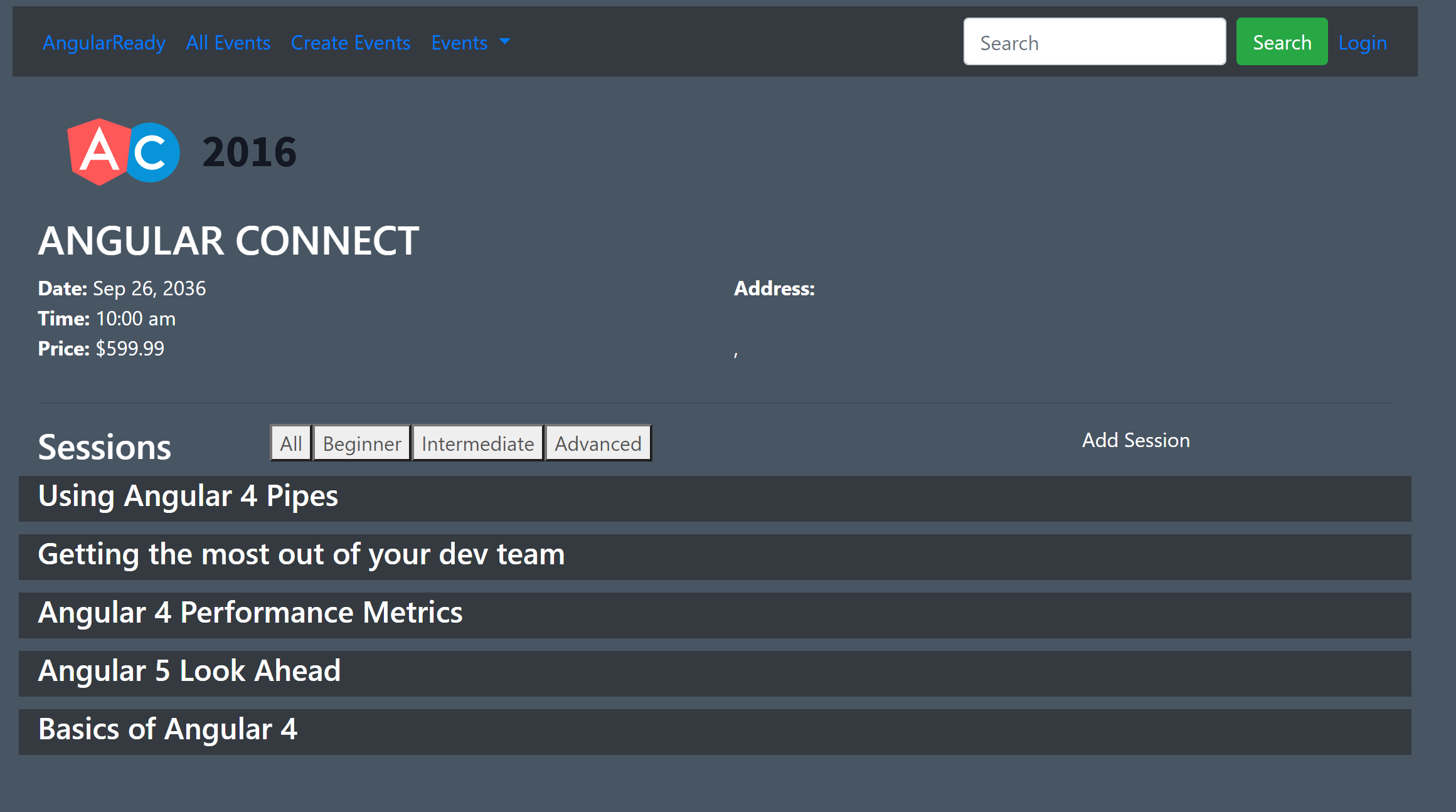
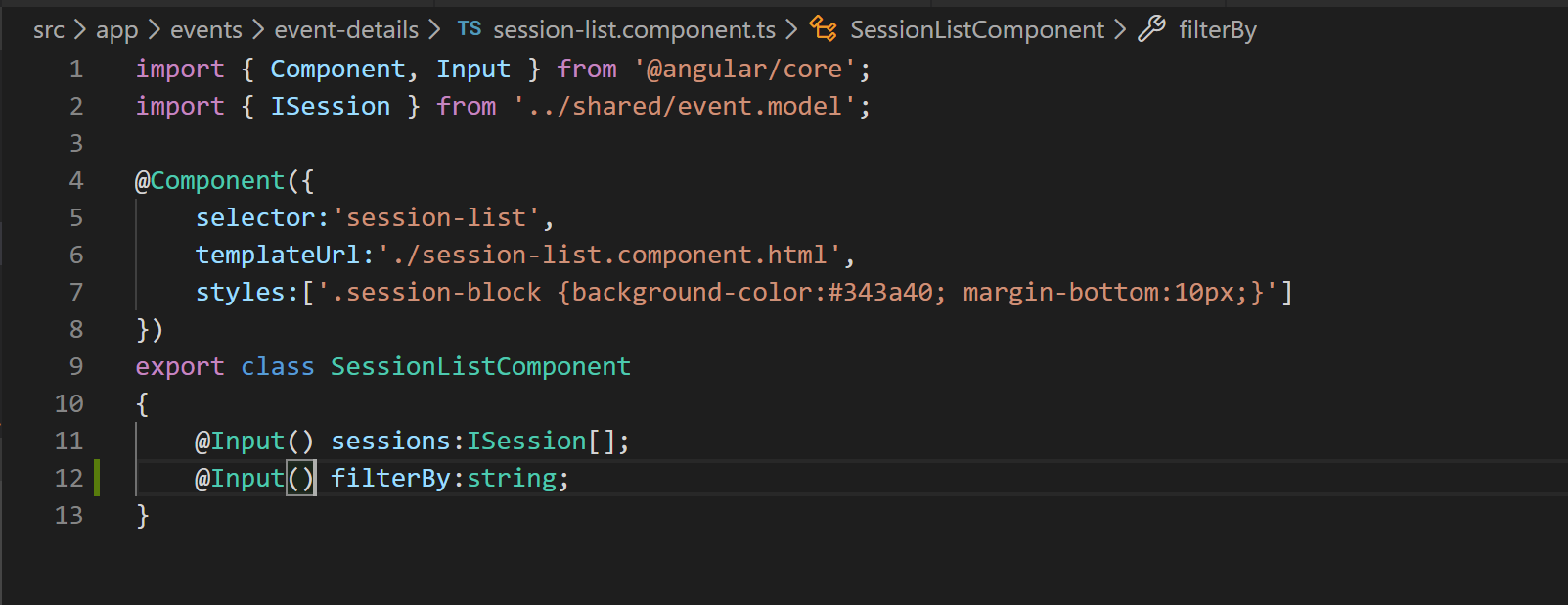
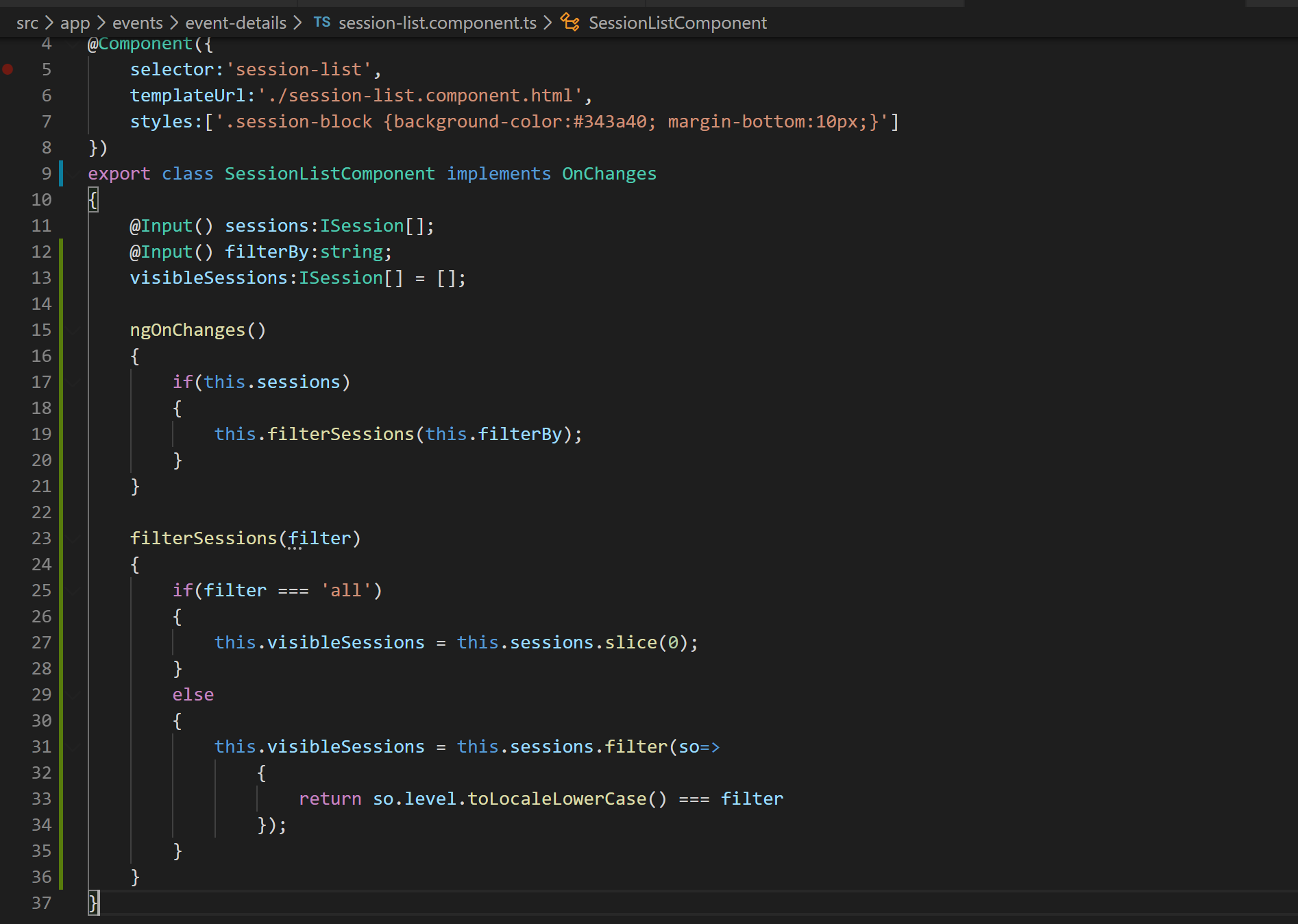
1. In Angular 2+ pipes are used to format the data. So lets start using OOB pipes to format some of the data we are rendering.
2. We will start with formatting event titles in upper case on “event-thumbnail” component.
3. Add the uppercase pipe to event name binding like below:
4. 
5. If you go to all events page now you will notice event names in capital now.
6. 
7. Lets do the same thing in event-details page like below:
8. Now lets add date pipe to date field binding like below:
9. 
10. Now dates start to look better like below:
11. 
12. Now lets use the currency pipe in places where we have hardcoded $ sign like below:
13. 
14. Now let’s fix the display of event duration using a custom pipe. As you know duration field has four values like below:
    1. Half Hour
    2. 1 Hour
    3. Half Day
    4. Full Day
15. We want to display these values instead of just the number. So lets create a custom pipe to do that as there is no OOB pipe that will do it.
16. So right click “shared” folder and create a file called “duration.pipe.ts”.
17. Lets create the basic shell like below:
18. 
19. Now lets implement it like below:
20. 
21. Now we need to add the pipe to declarations array in app module. So lets do that like below:
22. 
23. Now lets put the pipe to use in session-list component like below:
24. 
25. Now the duration will appear in its string form like below:
26. 
27. Lets now implement custom filters to sort our sessions based on session level which have three values
    1. Beginner
    2. Intermediate
    3. Advanced
28. Angular recommends to implement sorting and filtering logic in your application code so that it can be customized in case your application requirements change. This approach better than to have a generic OOB logic which would be expected to work in all custom application specific scenarios.
29. So lets start modifying the “event-details.component.html” and add the filter buttons like below:
30. 
31. We will fix the class binding and click binding later.
32. Now lets add a property to “event-details.component.ts” file that will specify what type of filter to apply on:
33. 
34. Now that we have filterBy property defined we can fix the class binding on filter buttons like below:
35. 
36. Now lets set the filterBy property to corresponding filter value on click of corresponding button like below:
37. 
38. If you go take a look at the application now the buttons are rendered like below:
39. 
40. Now we need to pass this filterBy property to session list component so that it can be used in ngOnChanges method to filter the sessions like below:
41. 
42. Now lets create the property on session-list component.
43. 
44. We are not ready to use it in ngOnChanges method like below:
45. 
46. As you can see we are storing filtered sessions in a new property called “visibleSessions” so lets bind this property to our template like below:
47. 
48. Now if you go to your application and try to filter sessions you will be able to.