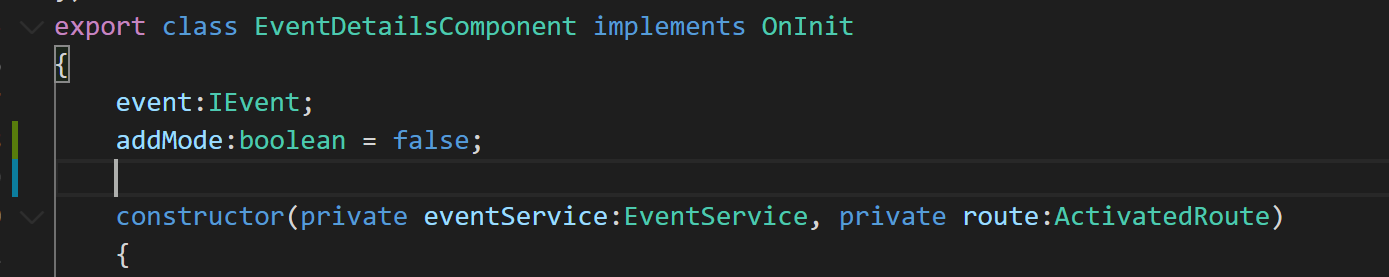
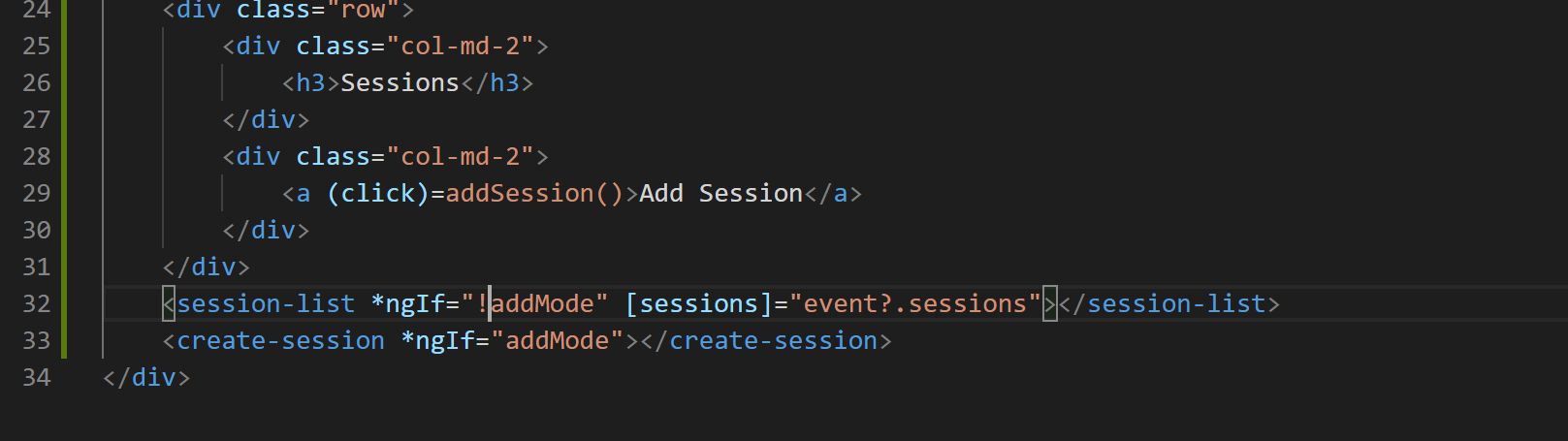
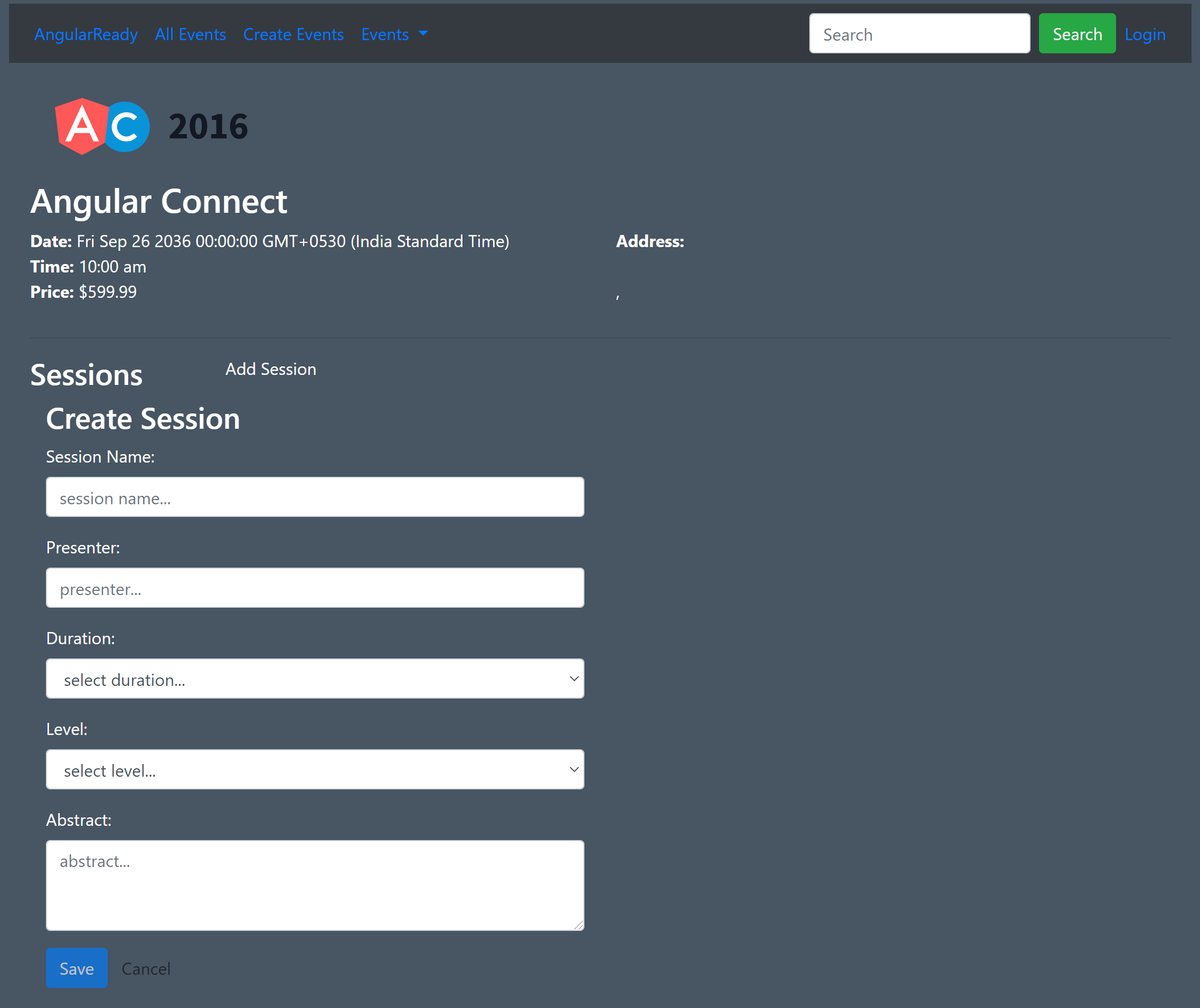
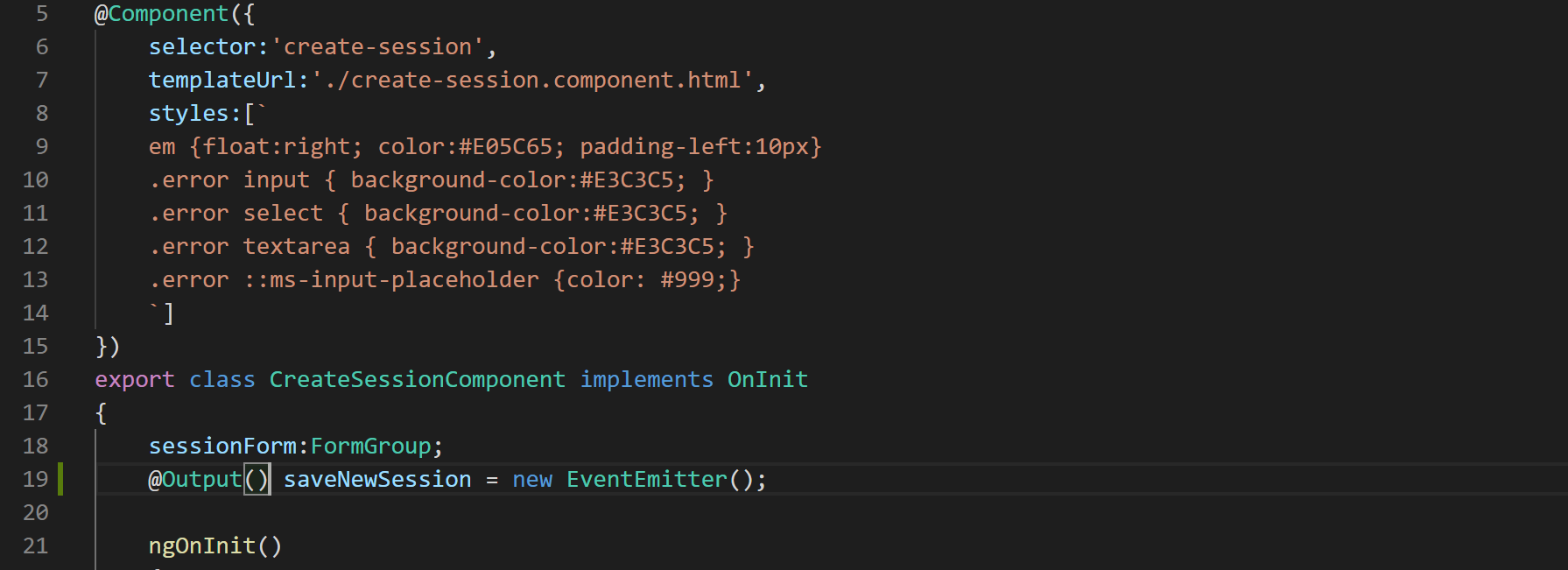
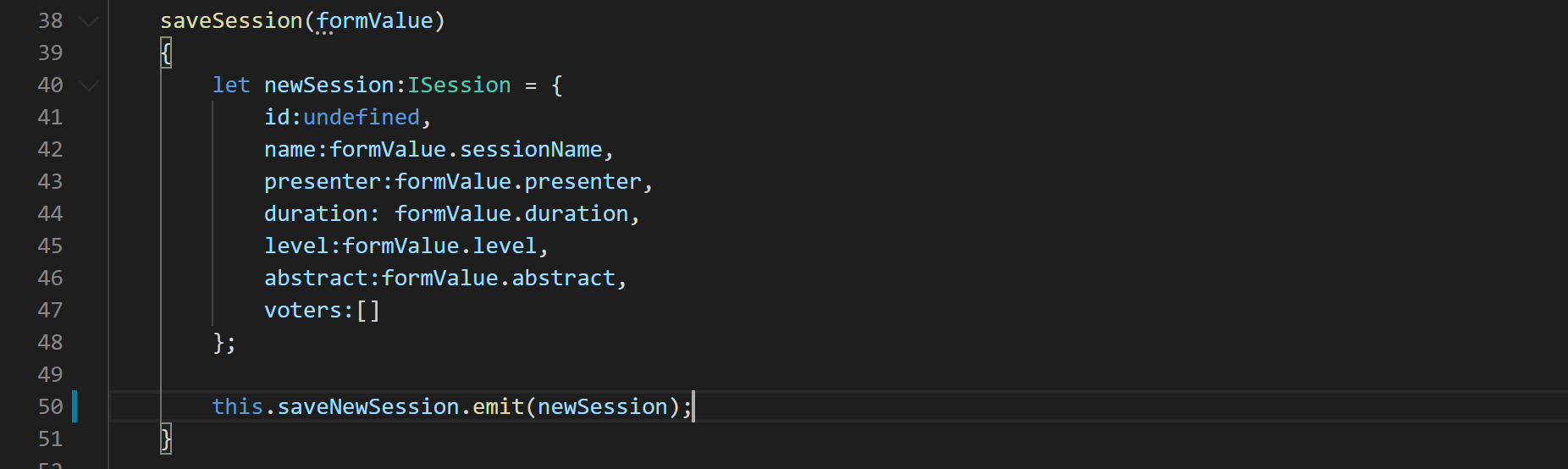
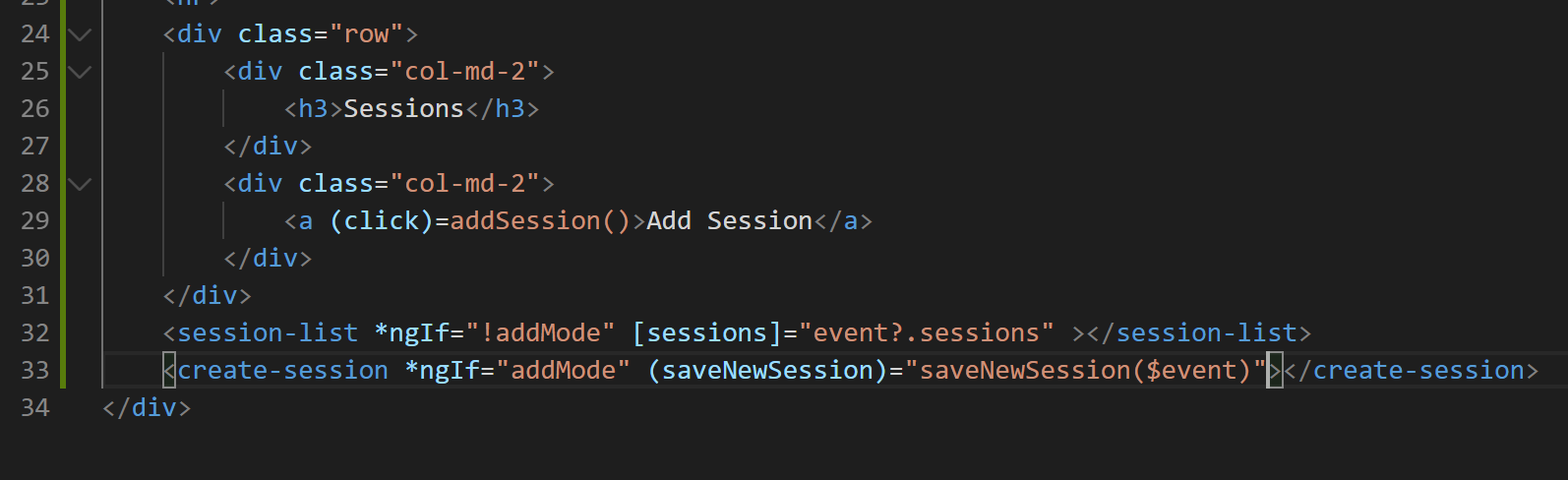
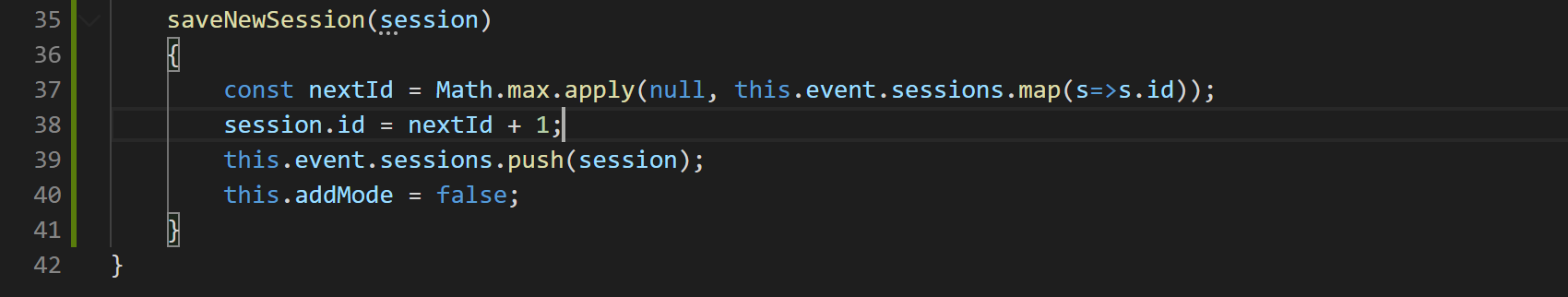
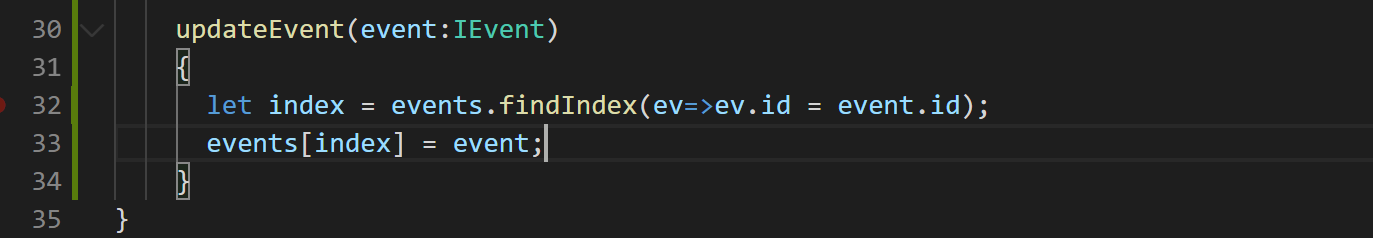
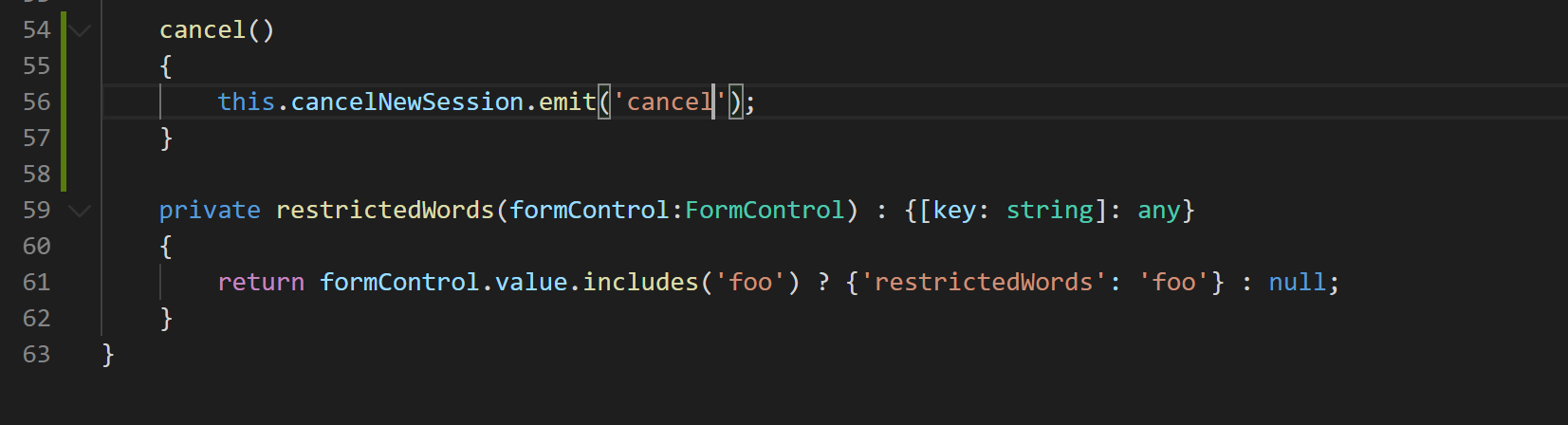
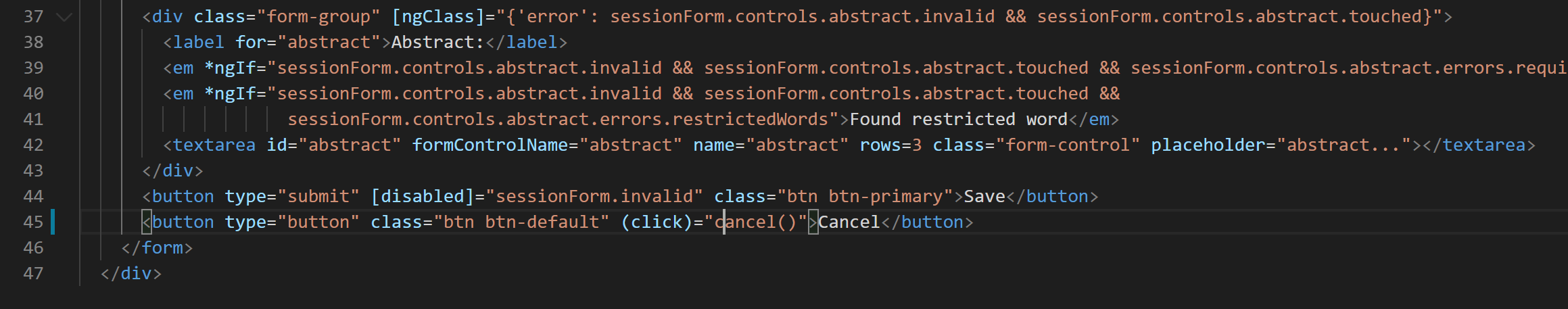
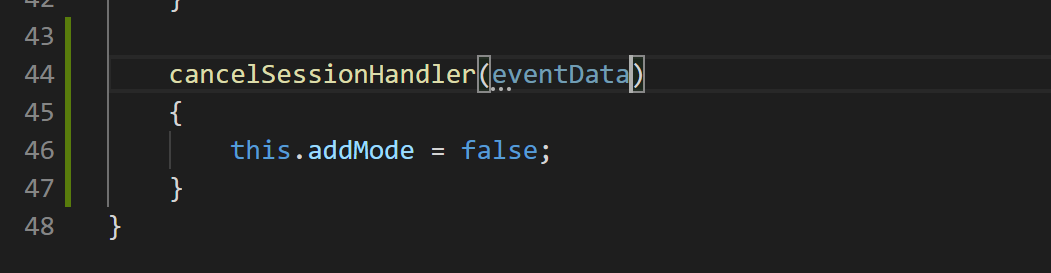
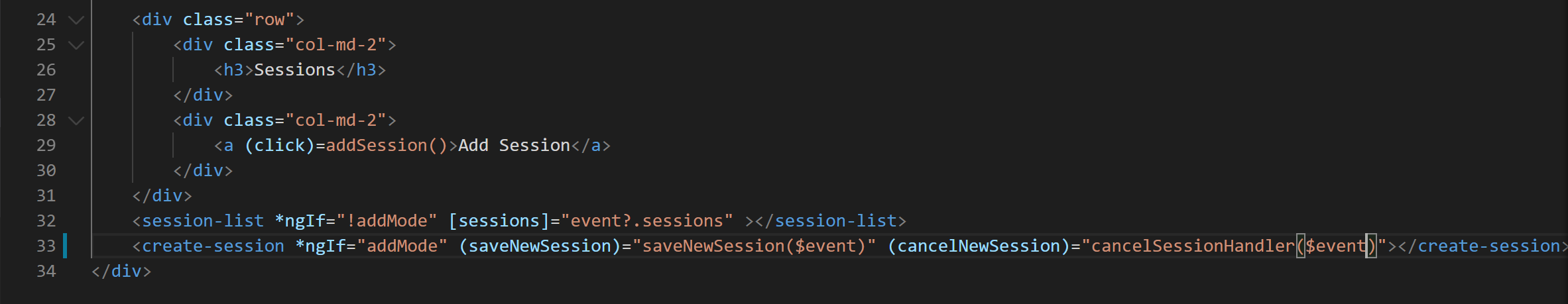
1. Sometimes we have a need to pass data from child component to the parent component. that is generally done using the output property. So let's try that.
2. You would remember that we created a component called create session that component is only accessible through a URL as of now. Now let's try and link that component to event details page so that we can create the sessions right out of the event details page.
3. Let's start with adding a sessions header and a link to create session page on the events details page like below:
4. 
5. Now lets add the method “addSession()” to our “event-details.component.ts” file like below:
6. 
7. Lets add a style to styles array in “event-details.component.ts” file so that the anchor tag has hand cursor even without href specified like below:
8. 
9. Now the “addSession” method is supposed to put us in add mode. So detect that lets add a property called “addMode” to our EventDetailsComponent like below:
10. 
11. Lets set it to true in in addSession conmethod like below:
12. With the help of this property what we want to do is we want to hide the session details component and render add session component like below. So lets add a ngIf so <session-list> element like below:
13. 
14. We also need to show create-session component when we are in add mode as coded in image above.
15. Now if you go to the application and on event details page and click create session link you will notice that session list component gets hidden and create session component is shown like below:
16. 
17. Now we are ready to emit the session object from create-session component back to the parent component so lets do that using Angular’s Output object and EventEmitter.
18. First lets define the output property of EventEmitter type in create-session component like below:
19. 
20. Now lets remove the console.log statement and emit the session object like below:
21. 
22. Now that we are emitting the event along with session data, lets bind to this event in our parent component and pass the event data to a handler method that will ultimately save the data to the session object like below:
23. 
24. Lets define the “saveNewSession” method in our “event-details.component.ts” file.
25. 
26. Now lets define the “updateEvent” method on our “event.service.ts” file like below:
27. 
28. Now if you go to the application in the browser change and click on add session. after filling all the session details how do you get when you click on save button the new session will be added to the current event object and will be displayed at the very bottom.
29. Call the updateEvent service method in “saveNewSession” method.
30. Finally let's bind the cancel button on the create session form using an output property and event emitter like below:
31. Define the output property on “create-session.component.ts” like below:
32. 
33. Now lets define the cancel method like below:
34. 
35. And bind it to the create session HTML template like below:
36. 
37. Finally lets handle the event in event details component like below:
38. First define the method cancelSessionHandler like below:
39. 
40. Now call this method on our <create-session> element in “event-details.component.html” like below:
41. 
42. Now if you click cancel button on the create session component you will go back to even details page.