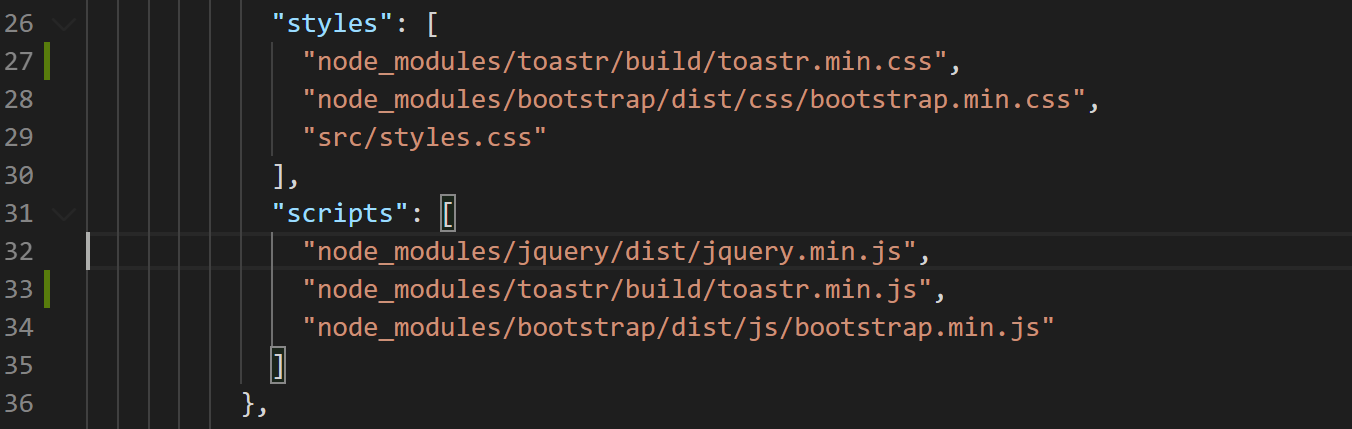
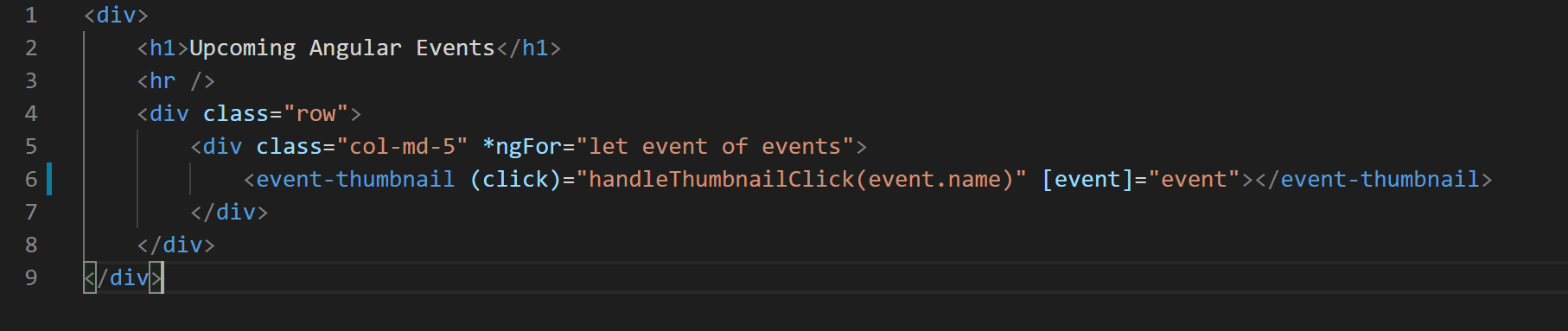
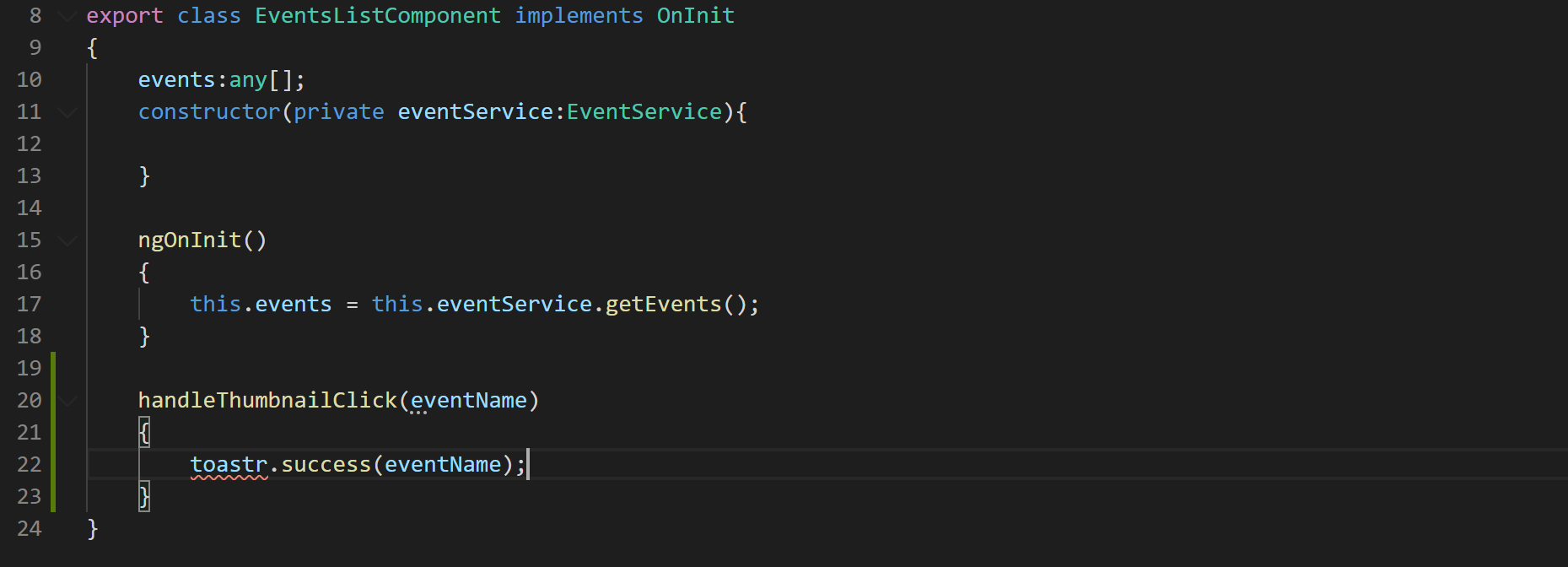
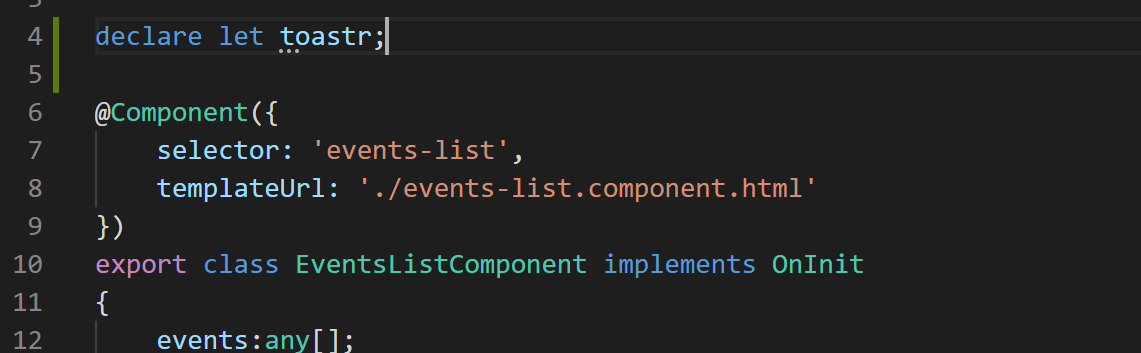
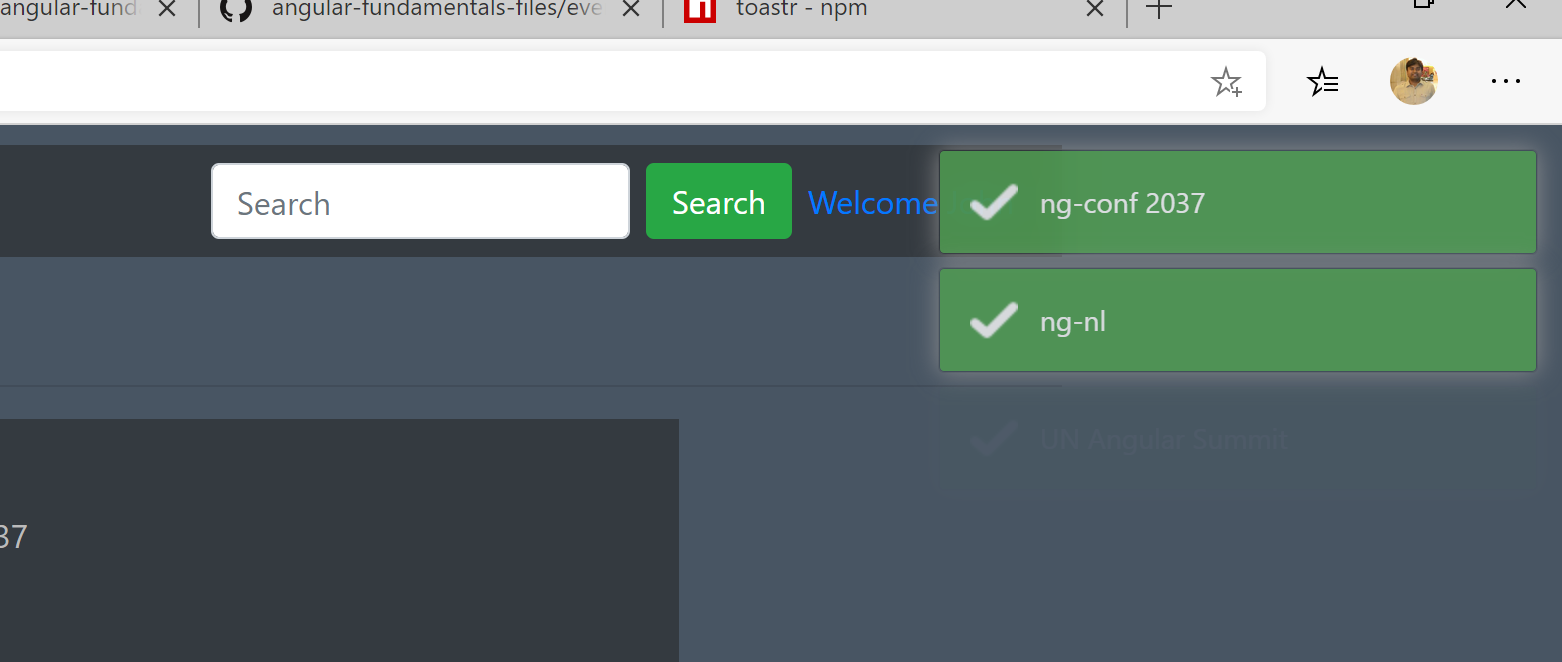
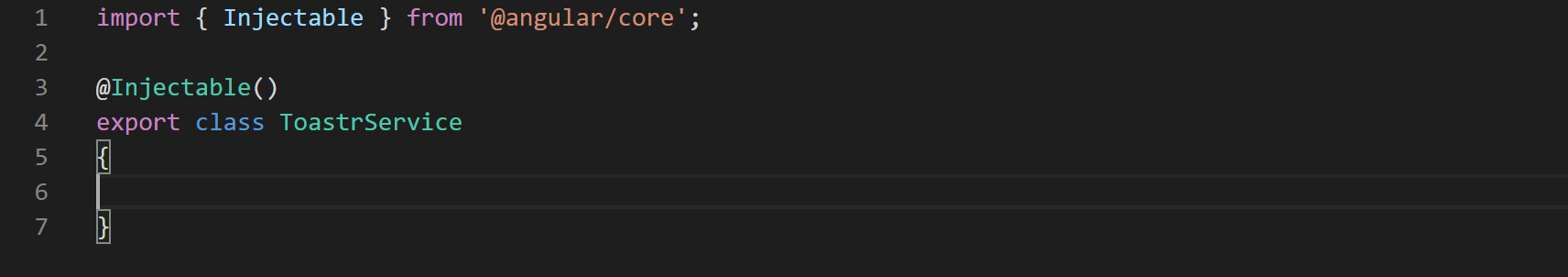
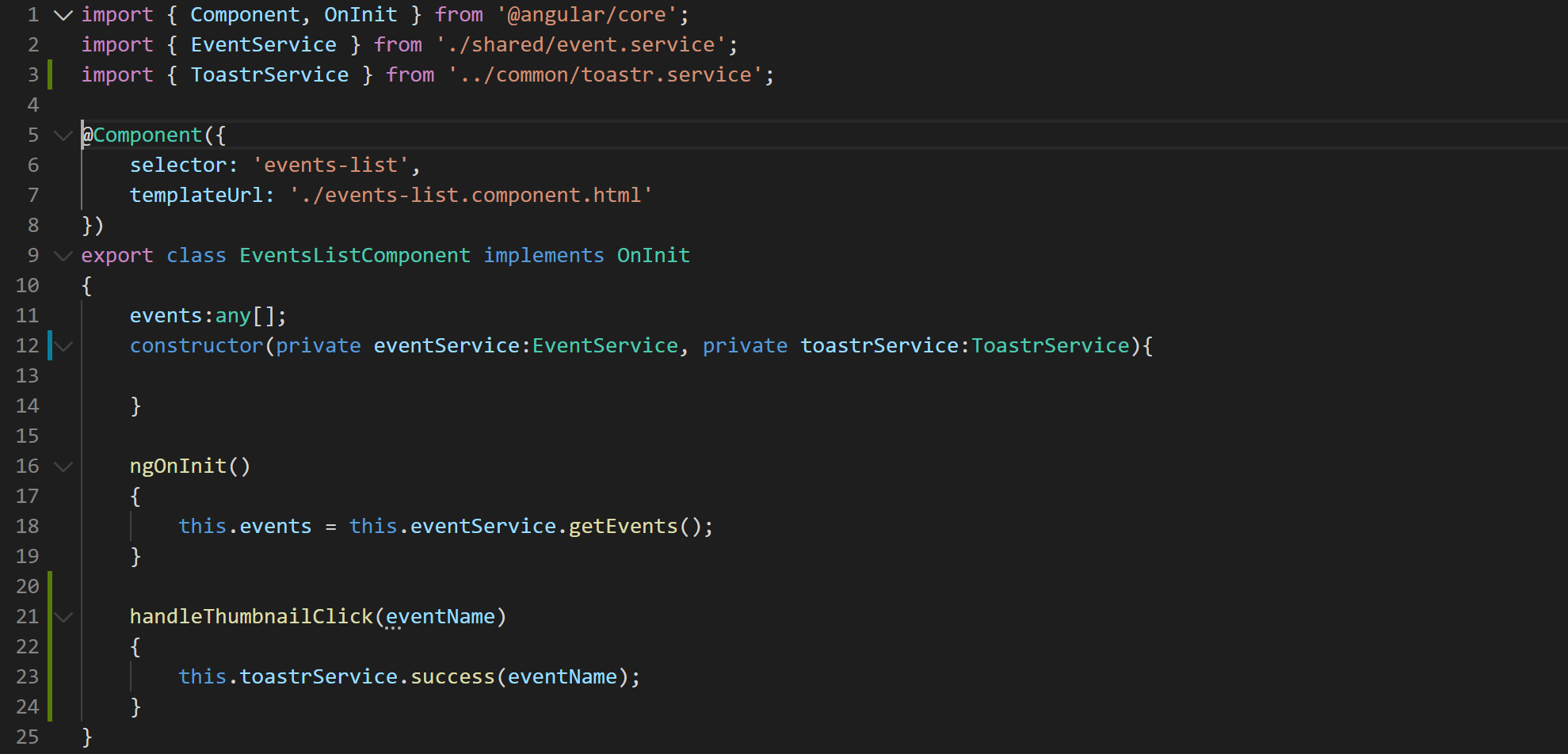
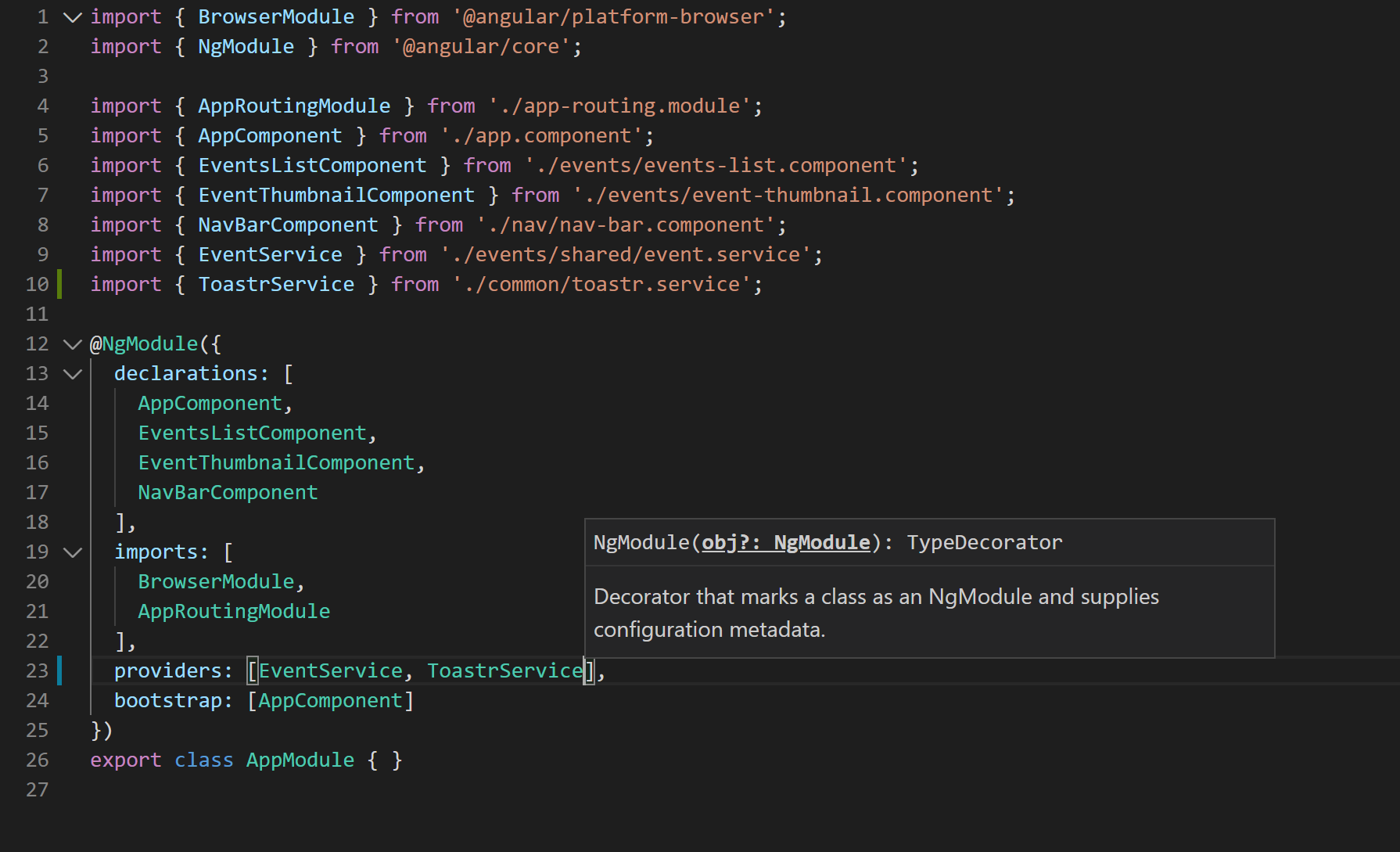
1. Go to Visual Studio code terminal and press control C.
2. Now type following command to install “toastr”: “npm i toastr”.
3. The toaster package will get a JavaScript file and a CSS file to use. So let's import those file in angular.json like below:
4. 
5. We will display a toast when an event thumbnail is clicked so let's go to the “events-list.component.html” and on the “<event-thumbnail>” element bind the click event like below:
6. 
7. Now lets add this “handleThumbnailClick()” method to our “events-list.component.ts” file. So open the file and add the method like below.
8. Since we registered the “toastr” JavaScript library in “angular.json” it is available globally in our application so we can directly call the “toastr” method now. Only problem is that “TypeScript” will not be able to find the type for it. So let's call the method and see what error type script gives:
9. 
10. To remove this type script error we will have to declare the “toastr” variable as a type on top of the class like below:
11. 
12. As soon as you declare the “toastr” variable you will notice that the error is gone.
13. Now go to the application in the browser and click on the event thumbnails you will notice a off green color “toastr” to the top right of the screen like below:
14. 
15. Even though we are able to use the toaster library now there are two problems with it:
    1. We are using a globally available object which is not good
    2. The “toastr” library is not testable meaning we cannot unit test it
16. To overcome the issues described above we will wrap the “toastr” library in an angular service and then inject that service in our component.
17. Since “toastr” is a common library that will be used by many components in the application we will create the wrapper angular service in a “common” folder. So let's create a “common” folder in the “app” folder.
18. In the “common” folder create a file called “toastr.service.ts” and export a class from it called “ToastrService” like below:
19. 
20. Now let's wrap the “toastr” library methods that we're interested in in this service class like below:
21. 
22. Now that our “toastr” service is ready let's inject it in our events list component and use it like below:
23. 
24. Let's not forget to remove the “toastr” type we declared on top of the class.
25. Now that we have created and consumed the “toastr” service let's make it available across the application by registering it in the providers array in app.module.ts like below:
26. 
27. Now go to the browser and our application should still be showing the toasts like previously.