A code review checklist (see the code being reviewed overleaf)

1. Have the requirements been met? ✅
2. Is the code formatted correctly? ✅
3. Are all imports used? ✅
4. Is unnecessary white space removed✅
5. Do we follow the single responsibility principle?✅
6. Are different errors handled correctly?❌

   There's no visible error handling, especially in the logout method and in the router navigation.
7. Are errors and warnings logged?❌

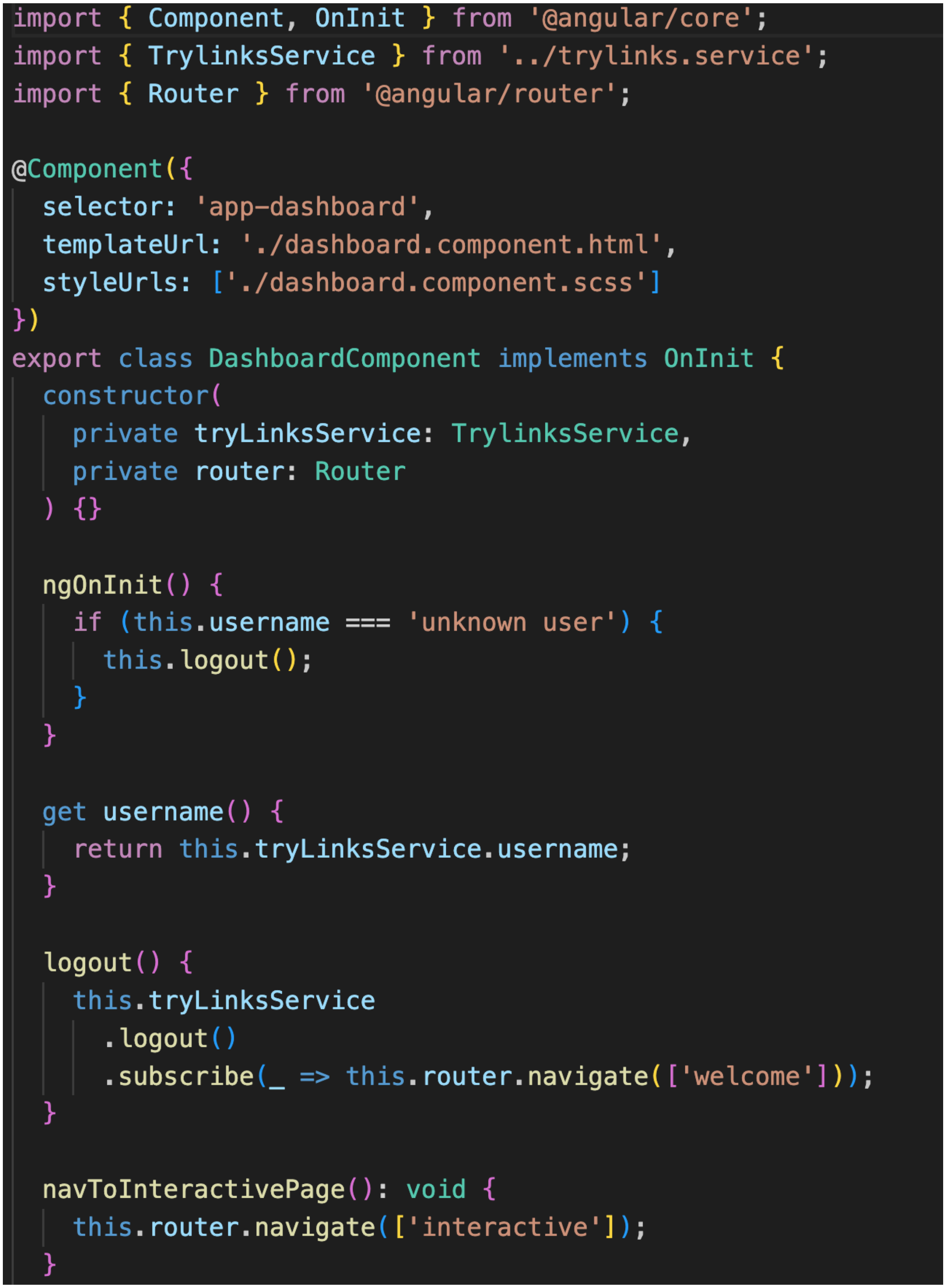
   Due to 5, errors are also not logged.
8. Magic values avoided?✅
9. Minimal nesting used?✅
10. Is the code easy to read?✅
11. Is the code not repeated (DRY Principle)?✅
12. Is the code method/class not too long?✅
13. Is the code performance acceptable?✅
14. Is it secure/free from risk?❌

    This needs to be investigated further.
15. Are separations of concern followed?✅
16. Relevant Parameters are configurable?
17. Have edge cases been tested?✅
18. Are invalid inputs validated? N/A
19. Are inputs sanitized? N/A
20. Is there sufficient documentation?❌

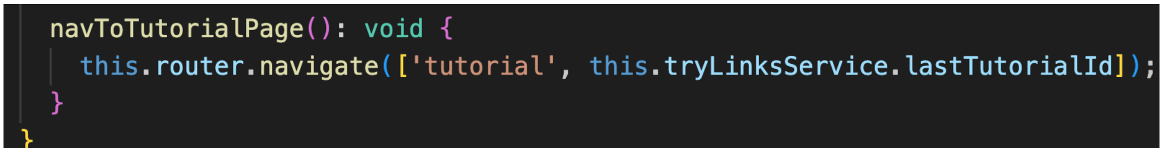
    There is no documentation.
21. Can the lines fit on a standard laptop screen (14 inches) or desktop screen (22-24 inches)?✅
22. Did the dev break the code into easy-to-understand chunks?✅
23. Is the code easy to test and debug?❌

    No, as there is no logging.

A code review checklist (see the code being reviewed overleaf)



```
import { Component, OnInit } from '@angular/core';
import { TrylinksService } from '../trylinks.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-dashboard',
  templateUrl: './dashboard.component.html',
  styleUrls: ['./dashboard.component.scss']
})
export class DashboardComponent implements OnInit {
  constructor(
    private tryLinksService: TrylinksService,
    private router: Router
  ) {}

  ngOnInit() {
    if (this.username === 'unknown user') {
      this.logout();
    }
  }

  get username() {
    return this.tryLinksService.username;
  }

  logout() {
    this.tryLinksService
      .logout()
      .subscribe(_ => this.router.navigate(['welcome']));
  }

  navToInteractivePage(): void {
    this.router.navigate(['interactive']);
  }
```



```
  navToTutorialPage(): void {
    this.router.navigate(['tutorial', this.tryLinksService.lastTutorialId]);
  }
}
```