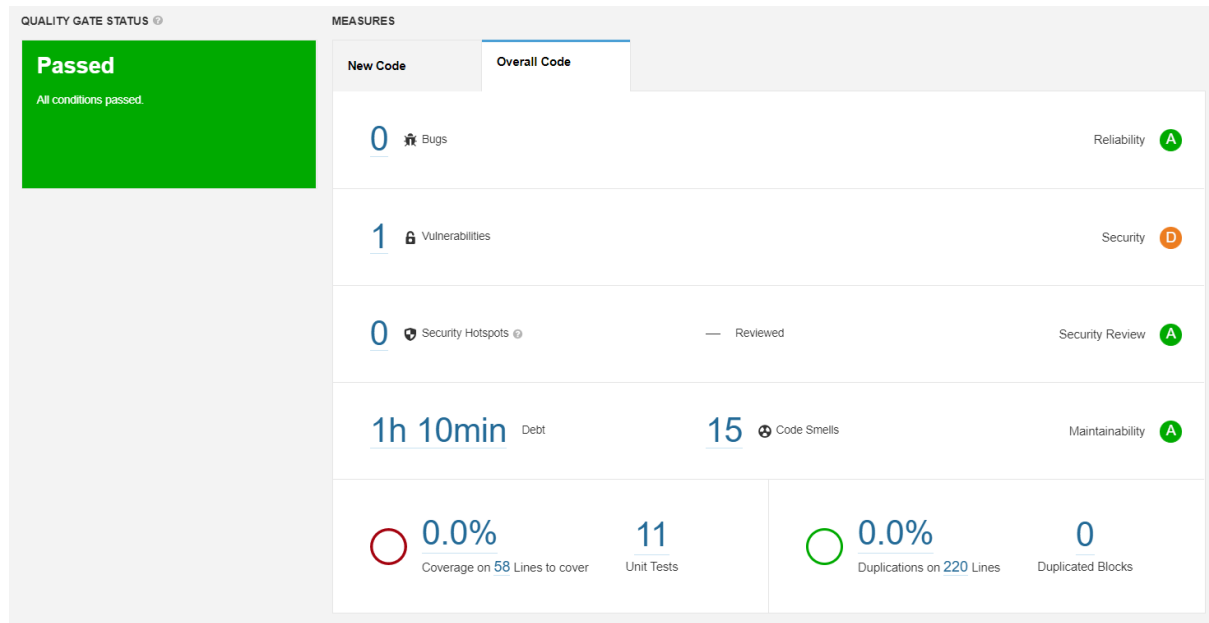### 6.3. Technical debt (Cars)
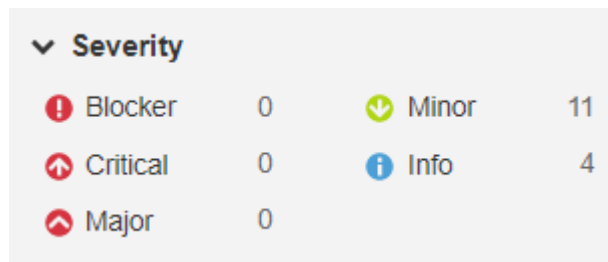
(a)
Technical debt found: 1h10min.
This means that it would take approximately 1h10min to rework the code and fix the identified problems.

The dashboard:

(b)
There were no critical or major code smells that needed to be fixed.

Additionally, I used the Entity data type as parameter in the API methods and, thus, got the following vulnerability:

```
@PostMapping("/cars" )
public ResponseEntity<Car> createCar(@RequestBody Car car) {
```

Replace this persistent entity with a simple POJO or DTO object.   Why is this an issue? 15 hours ago ▾ L25 🔗
🔒 Vulnerability ▾ 🔺 Critical ▾ ○ Open ▾ Not assigned ▾ 10min effort Comment   🏷 cwe, owasp-a5, spring ▾

This I corrected by introducing the CarDTO entity and making the necessary adjustments to the Controller class. Running a new code analysis, the vulnerability is now gone:
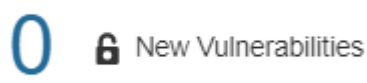
(d)

After configuring the Jacoco plugin and running the static analysis once again, the results for new code are:

71.43%
Coverage on 21 New Lines to cover

27.85%
Duplications on 79 New Lines

And for overall code:

38.6%
Coverage on 82 Lines to cover

11
Unit Tests

12.3%
Duplications on 280 Lines

2
Duplicated Blocks

There were 82 lines not covered and 2 conditions: coverage on new code and duplicated lines (%) on new code, both of which failed.

QUALITY GATE STATUS

Failed

2 conditions failed

On New Code

71.43% Coverage on New Code is less than 80.0%

27.85% Duplicated Lines (%) on New Code is greater than 3.0%