HW1: Mid-term assignment report

Gonçalo Correia Passos [88864], v2021-05-12

1 Introduction

1.1 Overview of the work

AirQuality is a website where where users can check the current quality of the air in all cities/regions in the world. It was developed using spring boot application. It provides parameters related to gases information like CO, O3, SO2, NO2, matter concentrations like PM10, PM25 and pollen levels in the air, like tree, grass weed and mold.

Current limitations

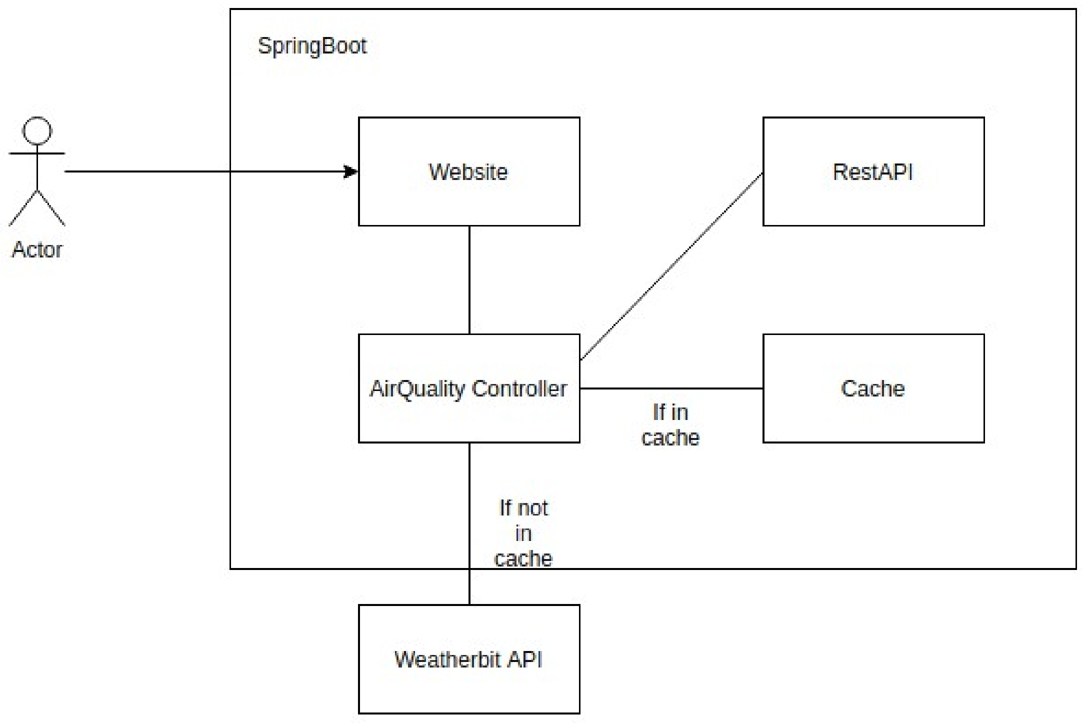
It is not possible to search this information based on coordinates, only by the cities' name. Beyond that, it depends of WeatherBit external API. If for some reason the API stops working, the AirQuality web page cannot be used.

2 Product specification

2.1 Functional scope and supported interactions

When the user has access to the website, all he has to do is search for the wanted city and the information will be displayed on the same page. The web page is simple and very minimalist, which makes it very intuitive and perfect for someone that wants to know the current air conditions in their city.

2.2 System architecture



2.3 API for developers

http://localhost:8080/ - sends client to the main page.
http://localhost:8080/AirQuality – sends client to the main page.
http://localhost:8080/AirQuality/{city} – sends client to the city's webpage and shows the corresponding air quality information.

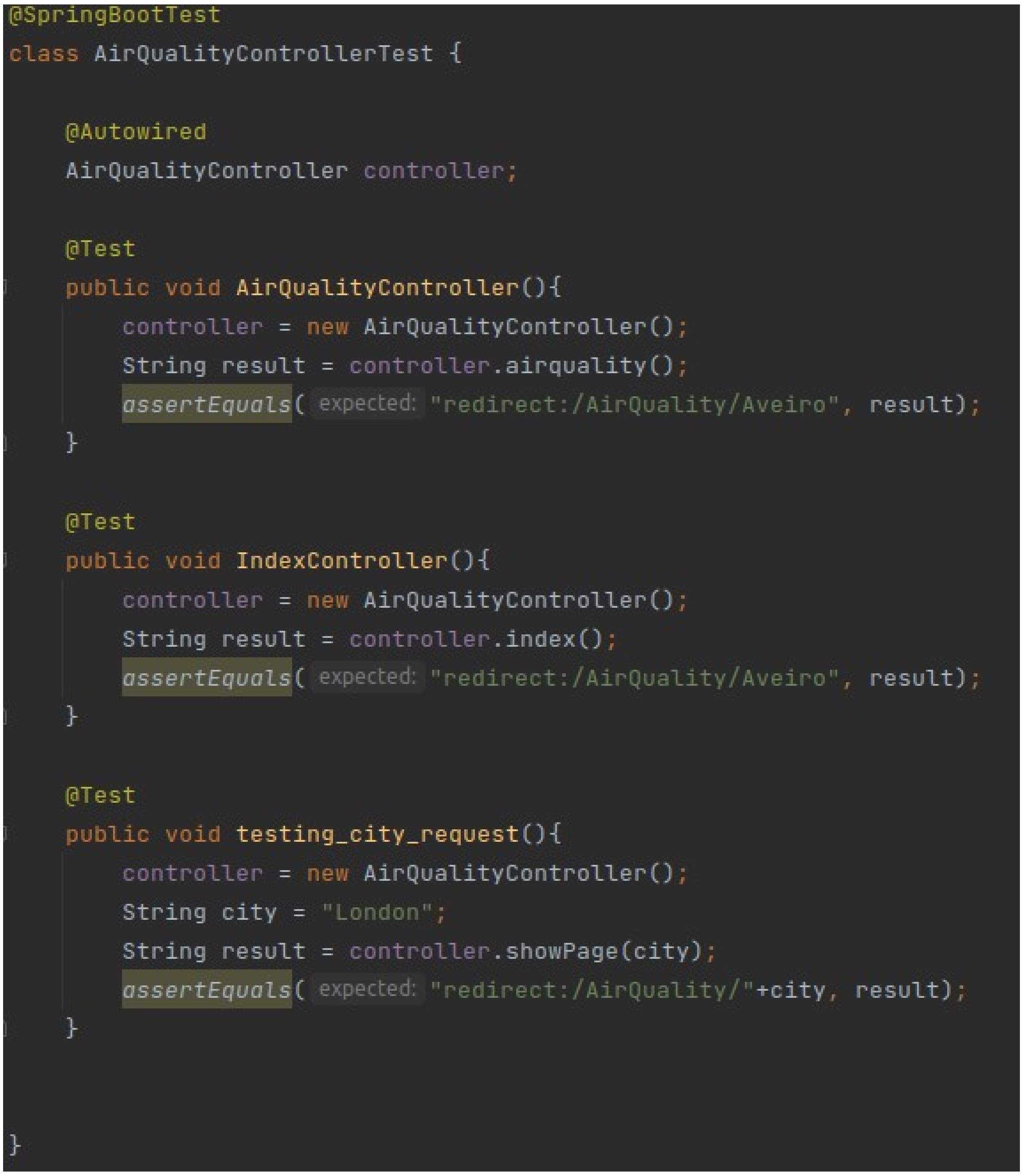
3 Quality assurance

3.1 Overall strategy for testing

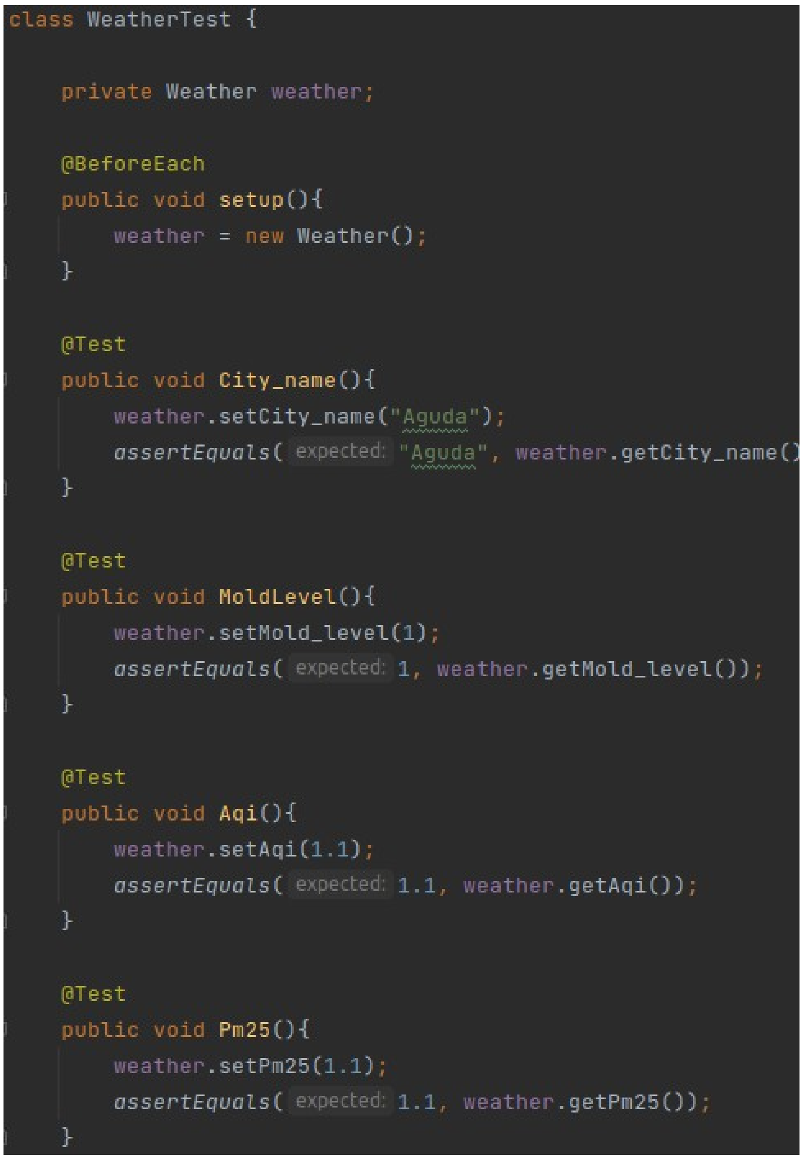
Overall, the tests were all developed using jUnit and the Selenium IDE. Didn't use Cucumber or Hamcrest because they are not very useful on simple projects like this, since they aim to simplify testing. Sonarqube was used in the end to identify some code smells.

3.2 Unit and integration testing

JUnit was used to test the AirQualityController class, to see if it was returning the correct web page and on the Weather class to see if it was returning the correct air quality data.



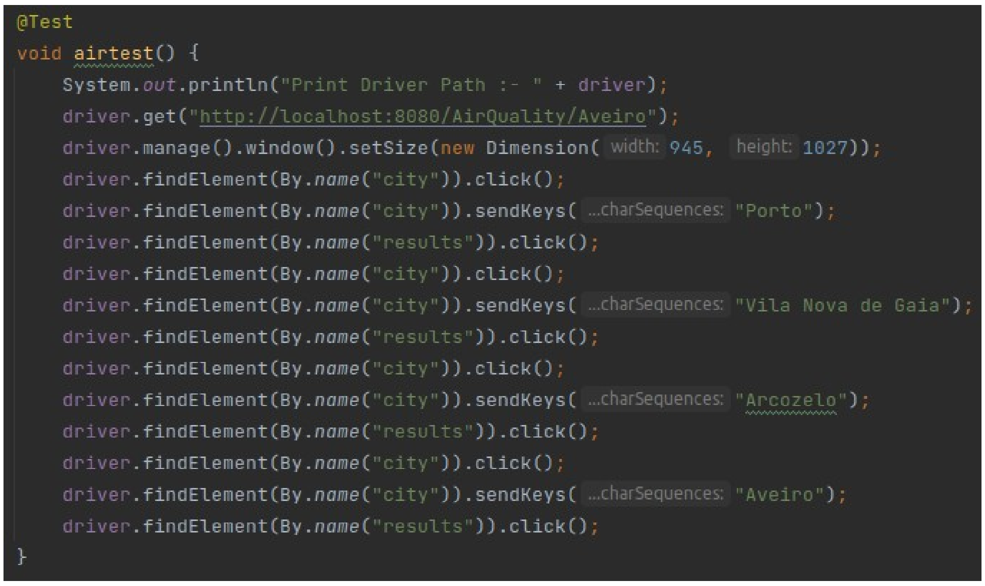
```java
@SpringBootTest
class AirQualityControllerTest {

    @Autowired
    AirQualityController controller;

    @Test
    public void AirQualityController(){
        controller = new AirQualityController();
        String result = controller.airquality();
        assertEquals( expected: "redirect:/AirQuality/Aveiro", result);
    }

    @Test
    public void IndexController(){
        controller = new AirQualityController();
        String result = controller.index();
        assertEquals( expected: "redirect:/AirQuality/Aveiro", result);
    }

    @Test
    public void testing_city_request(){
        controller = new AirQualityController();
        String city = "London";
        String result = controller.showPage(city);
        assertEquals( expected: "redirect:/AirQuality/"+city, result);
    }


}
```



```java
class WeatherTest {

    private Weather weather;

    @BeforeEach
    public void setup(){
        weather = new Weather();
    }

    @Test
    public void City_name(){
        weather.setCity_name("Aguda");
        assertEquals( expected: "Aguda", weather.getCity_name()
    }

    @Test
    public void MoldLevel(){
        weather.setMold_level(1);
        assertEquals( expected: 1, weather.getMold_level());
    }

    @Test
    public void Aqi(){
        weather.setAqi(1.1);
        assertEquals( expected: 1.1, weather.getAqi());
    }

    @Test
    public void Pm25(){
        weather.setPm25(1.1);
        assertEquals( expected: 1.1, weather.getPm25());
    }
```

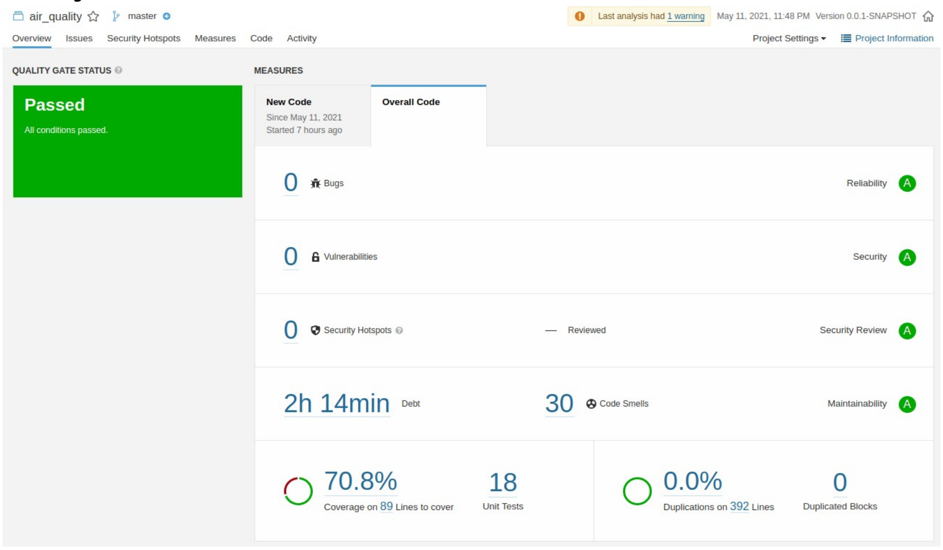
3.3 System Testing

I used the Selenium IDE for system testing. I decided to use the Firefox Geckodriver since Chromedriver stopped working on my PC due to Google Chrome updated version. The test opens Firefox and searches for four cities. It's not very useful since it's a very simple project.



```
@Test
void airtest() {
    System.out.println("Print Driver Path :- " + driver);
    driver.get("http://localhost:8080/AirQuality/Aveiro");
    driver.manage().window().setSize(new Dimension( width: 945,  height: 1027));
    driver.findElement(By.name("city")).click();
    driver.findElement(By.name("city")).sendKeys( ...charSequences: "Porto");
    driver.findElement(By.name("results")).click();
    driver.findElement(By.name("city")).click();
    driver.findElement(By.name("city")).sendKeys( ...charSequences: "Vila Nova de Gaia");
    driver.findElement(By.name("results")).click();
    driver.findElement(By.name("city")).click();
    driver.findElement(By.name("city")).sendKeys( ...charSequences: "Arcozelo");
    driver.findElement(By.name("results")).click();
    driver.findElement(By.name("city")).click();
    driver.findElement(By.name("city")).sendKeys( ...charSequences: "Aveiro");
    driver.findElement(By.name("results")).click();
}
```

3.4 Static code analysis



Last analysis had 1 warning   May 11, 2021, 11:48 PM   Version 0.0.1-SNAPSHOT

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code
Since May 11, 2021
Started 7 hours ago

Overall Code

0 Bugs — Reliability A

0 Vulnerabilities — Security A

0 Security Hotspots — Reviewed — Security Review A

2h 14min Debt — 30 Code Smells — Maintainability A

70.8% Coverage on 89 Lines to cover — 18 Unit Tests

0.0% Duplications on 392 Lines — 0 Duplicated Blocks

When I used Sonarqube for the first time, I detected 70 code smells and tests were only covering 60% of the code. Fixing some of these code smells (unused imports, the use of logger, etc), increased the coverage to 70%. The remaining code smells refer to the name of some variables, like "city_name" and I don't think that's a problem. There were no bugs, vulnerabilities or security hotspots.

4 References & resources

Project resources

- https://github.com/gpassos99/AirQuality

Reference materials

https://start.spring.io
https://www.weatherbit.io
https://www.baeldung.com/spring-cache-tutorial
https://www.youtube.com/watch?v=WPKv8NA-ZhE
https://stackoverflow.com/questions/53191468/no-creators-like-default-construct-exist-cannot-deserialize-from-object-valu
https://stackoverflow.com/questions/52713028/fail-to-parse-entry-in-bootstrap-index-maven-sonar-analysis-with-gitlab
https://www.bootstrapdash.com/use-bootstrap-with-html/
https://www.baeldung.com/thymeleaf-in-spring-mvc
https://www.free-css.com/free-css-templates/page264/host-cloud