



DP2 – REFACTORING

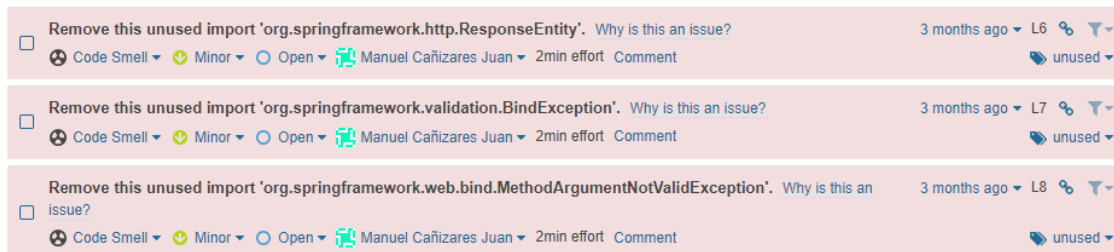
Report

Manuel Cañizares Juan
mancanjua@alum.us.es

The object of this report is to show all the code smells detected by SonarCloud in the project regarding the base project and the US 001, 002, 007, 008 and 0012, explain the reason of why they are present on our project and a way to fix them if necessary.

- Remove unused imports

This bad smell is given when we have imports that are not used in the specified class. Some examples are:

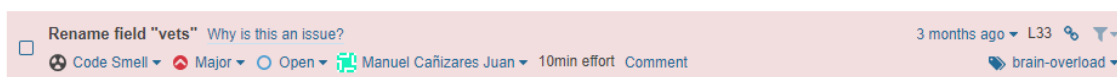


This screenshot shows three SonarCloud issues related to unused imports. Each issue entry includes a checkbox, a title describing the unused import, a 'Why is this an issue?' link, a severity level (L6, L7, L8), a date (3 months ago), a user (Manuel Cañizares Juan), and an effort (2min). The issues are: 'Remove this unused import 'org.springframework.http.ResponseEntity'', 'Remove this unused import 'org.springframework.validation.BindException'', and 'Remove this unused import 'org.springframework.web.bind.MethodArgumentNotValidException''. Each issue has a 'Code Smell' icon, a 'Minor' severity, an 'Open' status, and a 'Comment' link. A 'unused' tag is visible on the right of each issue.

They can be easily fixed removing those imports. In eclipse, an efficient way to do this is using the Ctrl + Shift + O key combination to reorganize them.

- Rename field

This smell is given when a field is called the same as the class. One example is:



This screenshot shows a SonarCloud issue titled 'Rename field "vets"'. It includes a checkbox, a title, a 'Why is this an issue?' link, a severity level (L33), a date (3 months ago), a user (Manuel Cañizares Juan), and an effort (10min). The issue is categorized as a 'Code Smell' with a 'Major' severity and an 'Open' status. A 'brain-overload' tag is visible on the right.

It can be fixed by renaming the field with a more descriptive name. In this case, *vetList* for example.

- Remove the declaration of a thrown exception which is a runtime exception

This smell is given when a method throws an exception that can cause a runtime exception and lead in an unexpected system behavior. Some examples are:



This screenshot shows four SonarCloud issues related to thrown exceptions. Each issue entry includes a checkbox, a title, a 'Why is this an issue?' link, a severity level (L37, L43, L48, L52, L57), a date (3 months ago or last month), a user (Manuel Cañizares Juan), and an effort (5min). The issues are: 'Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.' (L37), 'Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.' (L43), 'Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.' (L48), and 'Remove the declaration of thrown exception 'org.springframework.dao.DataAccessException' which is a runtime exception.' (L52, L57). Each issue has a 'Code Smell' icon, a 'Minor' severity, an 'Open' status, and a 'Comment' link. A tag 'clumsy, error-handling, redundant, unu...' is visible on the right of each issue. The first three issues are associated with the file 'src/.../samples/petclinic/service/AuthoritiesService.java' and the last one with 'src/.../samples/petclinic/service/OwnerService.java'.

It can be fixed by removing the explicit thrown of an exception, like in this method:

```
@Transactional(readOnly = true)
public Owner findOwnerById(final int id) throws DataAccessException {
    return this.ownerRepository.findById(id);
}
```

Which change will result in:

```
@Transactional(readOnly = true)
public Owner findOwnerById(final int id) {
    return this.ownerRepository.findById(id);
}
```

- Define a constant instead of duplicating literals n times

This code smell is given when there are duplicated strings with the same goal distributed all over the class, because if an update is needed, you must be sure to change all the occurrences. Some examples are:

src/.../samples/petclinic/web/MedicineController.java

<input type="checkbox"/> Define a constant instead of duplicating this literal "medicine" 5 times. Why is this an issue? Code Smell Critical Open Manuel Cañizares Juan 12min effort Comment	3 months ago L59
<input type="checkbox"/> Define a constant instead of duplicating this literal "Medicine not found" 4 times. Why is this an issue? Code Smell Critical Open Manuel Cañizares Juan 10min effort Comment	3 months ago L89

src/.../petclinic/web/validators/MedicineValidator.java

<input type="checkbox"/> Define a constant instead of duplicating this literal "Required" 4 times. Why is this an issue? Code Smell Critical Open Manuel Cañizares Juan 10min effort Comment	3 months ago L35
--	------------------

This can be fixed adding some constants to the class and use them where the literals were.

```
private static final String MEDICINE_LITERAL = "medicine";
private static final String MEDICINE_NOT_FOUND_MESSAGE = "Medicine not found";
```

- Remove public modifier

This smell is given when we set the visibility to public in the test methods and classes using Junit5, which is not necessary since this version. Some examples are:

src/.../samples/petclinic/integration/PetFinderAPITest.java	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 26 days ago L18 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 26 days ago L23 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 26 days ago L45 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 26 days ago L63 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
src/.../samples/petclinic/model/AdoptionValidatorTests.java	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> last month L13 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
src/.../samples/petclinic/model/MedicineValidatorTests.java	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 2 months ago L21 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>
src/.../samples/petclinic/model/PrescriptionValidatorTest.java	<div> <input type="checkbox"/> Remove this 'public' modifier. Why is this an issue? </div> <div> 2 months ago L17 Code Smell Info Open Manuel Cañizares Juan 2min effort Comment </div> <div> junit, tests </div>

This can be fixed removing the visibility of all the testing methods and classes and let them to be by default:

```
@Test
void getOrganizationsListTest() {
    given()
        .auth()
        .oauth2(token)
    .when()
        .get("https://api.petfinder.com/v2/organizations")
    .then()
        .statusCode(200)
    .and()
        .assertThat()
            .body("organizations.size()", equalTo(20))
            .body("pagination.count_per_page", equalTo(20))
            .body("pagination.current_page", equalTo(1))
    .and()
        .time(LessThan(20L), TimeUnit.SECONDS);
}
```

By doing all the changes regarding this code smells, the total count of them is reduced from 303 to 263 which entails a reduction of 40 code smells.

