23-5-2022

Lint Report

ACME TOOLKITS

Grupo E2.04

https://github.com/RafaJF/Acme-Toolkits

CIEZAR LANZA, EDUARDO | educielan@alum.us.es
JIMÉNEZ FERNÁNDEZ, RAFAEL ÁNGEL | rafjimfer@alum.us.es
RIVERO GALLARDO, JESÚS ANTONIO | jesrivgal@alum.us.es
SALAZAR CABALLERO, ALBERTO | albsalcab@alum.us.es
TORO VALLE, DANIEL | dantorval@alum.us.es
VILLAZÁN TORRES, FRANCISCO | fraviltor@alum.us.es

Contenido

Resumen ejecutivo

En este documento se detallará los datos proporcionados por la herramienta SonarLint respecto a la calidad y análisis del código de nuestro proyecto, sin tener cuenta herramientas de terceros ni datos del framework proporcionado.

Respecto a la **entrega D3**, al ejecutar esta herramienta nos hemos encontrado con varias advertencias del código relacionadas con los formularios de las vistas Patron-dashboard y Administrator-dashboard. Las descripciones de los avisos indicaban todas lo mismo: “Añadir una descripción a la tabla”.

El otro tipo de advertencia que nos ha salido ha sido en la clase AdmnistratorDashboardService.java en la que se nos pide que cambiemos un Map a un EnumMap.

Para la **entrega D4**, la ejecución de la herramienta SonarLint nos ha mostrado exactamente 4 malos olores relacionadas con el código del proyecto. La primera de ellas ha sido la comentada en el anterior entregable de AdministratorDashboardService.java, y las otras relacionadas con un método de la clase InventorToolkitUpdateService y con los nombres de otras dos clases de los paquetes src/test/java acme.testing.authenticated.inventor (SignUpBecomeInventor.java) y acme.testing.authenticated.patron (SignUpBecomePatron.java).

Posteriormente, todos los malos olores relacionados con la **entrega D3** están corregidos, excepto el que muestra AdministratorDashboardService.java. En esta **entrega D4**, se han solucionado todos los avisos que mostraba la herramienta exceptuando el mal olor anterior. Esto es debido a que implicaba un cambio de código en la clase que suponía alterar varias clases del proyecto y supondría un gran coste en cuanto esfuerzo y tiempo que dedicar. Puesto que no supone un gran problema, se ha decidido dejar este mal olor en el proyecto.

Historial de versiones

Versión	Fecha	Descripción del cambio
V1.0	25/04/2022	Se han añadido todos los apartados correspondientes al documento y todos los malos olores relacionados con le entrega D3.
V2.0	23/05/2022	Se han añadido los malos olores detectados en la entrega D4. Todos los nuevos cambios empiezan en el apartado 3.3.

1. Introducción

En este documento se ilustrarán las advertencias que nos ha proporcionado la herramienta SonarLint, tanto los errores mostrados como las soluciones proporcionadas que arreglan los avisos.

En la finalización de esta cuarta entrega, tras ejecutar la herramienta nos ha mostrado 4 malos olores relacionados con la calidad del código.

A continuación, se describirá y añadirán imágenes de los malos olores proporcionados y de las soluciones adoptadas junto con la imagen del resultado final.

2. Malos olores

2.1. Mal olor 1 (D3)

Este mal olor nos ha salido en varias tablas de los formularios de Patron-Dashboard y Administrator-dashboard.

form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.
form.jsp		Add a description to this table.

El problema es que nos pedía añadir una descripción a las clases de las tablas.

```
18 }
19 </style>
20 <acme:form readonly="true">
21
22             <h3>
23                 <acme:message code="administrator.administrator-dashboard.form.label.total-number-of-components"/>
24             </h3>
25             <table>
26                 <tr>
27                 <th id="totalNumberOfComponentsId">
28                     <acme:print value="${totalNumberOfComponents}"/>
29                 </th>
30                 </tr>
31             </table>
32             <h3>
33                 <acme:message code="administrator.administrator-dashboard.form.label.total-number-of-tools"/>
34             </h3>
35             <table>
36
37                 <tr>
```

Como solución hemos añadido la etiqueta <caption></caption>

```
<h3>
    <acme:message
        code="administrator.administrator-dashboard.form.label.total-num
</h3>
<table class="table table-sm">
<caption></caption>
<div class="column">
    <jstl:forEach items="${ totalNumberOfPatronagesByStatus.keySet() }"
        var="key">
        <tr>
            <jstl:set value="${ totalNumberOfPatronagesByStatus.get(key)
                var="amount" />
```

2.2. Mal olor 2 (D3)

Este es el otro mal olor que nos ha salido al final del proyecto

Resource	Date	Description
AdministratorDashboardShowService.java		Convert this Map to an EnumMap.

El error nos pedía que en la entrada de la línea 52 cambiáramos el Map que habíamos declarado en un tipo EnumMap.

```
51
52        final Map<Status, Integer> totalNumberOfPatronagesByStatus = new HashMap<Status, Integer>();
```

Estuvimos mirando si no suponía mucho cambio de código respecto a otras clases, sin embargo como estábamos cerca del día de la entrega, decidimos arreglar este mal olor para la siguiente entrega.

3.3 Mal olor 3 y 4 (D4)

En primer lugar, tras ejecutar SonarLint, nos ha mostrado los siguientes malos olores, centrándonos en este apartado en los dos últimos puesto que informaban de advertencias similares, pero en clases distintas.

4 items

Resource	Date	Description
AdministratorDashboardShowService.java		Convert this Map to an EnumMap.
InventorToolkitUpdateService.java		Merge this if statement with the enclosing one. [+1 location]
SingUpAndBecomeInventor.java		Rename class "SingUpAndBecomeInventor" to match the regular expression: '^((Test|IT)[a-zA-Z0-9_]+|[A-Z][a-zA-Z0-9_]*(Test|Tests|TestCase|IT|ITCase))$'
SingUpAndBecomePatron.java		Rename class "SingUpAndBecomePatron" to match the regular expression: '^((Test|IT)[a-zA-Z0-9_]+|[A-Z][a-zA-Z0-9_]*(Test|Tests|TestCase|IT|ITCase))$'

Los errores relacionados con las clases *SignUpAndBecomeInventor.java* y *SignUpAndbecomePatron.java* indicaban que los nombres de dichas clases no cumplían con los patrones de las expresiones regulares propuestas en los paquetes de *acme.testing.authenticated.inventor* y *acme.testing.authenticated.patron*.

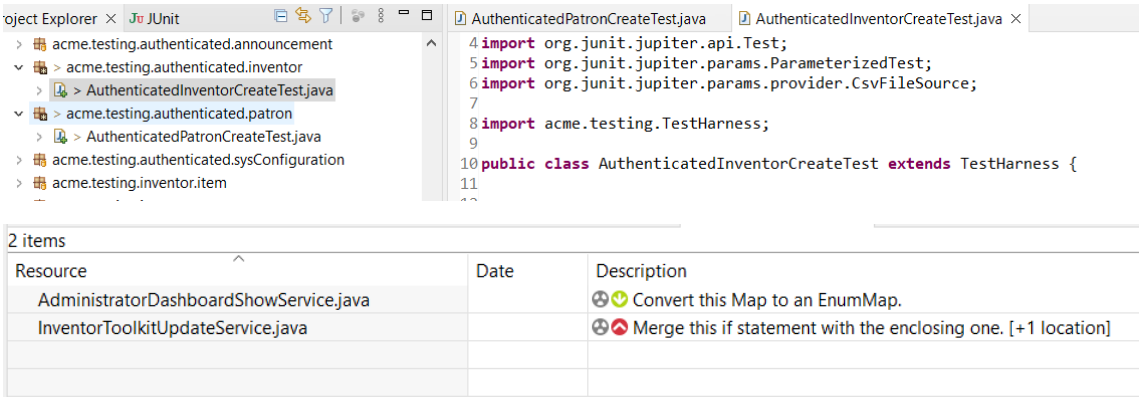
SingUpAndBecomePatron.java

```java
package acme.testing.authenticated.patron;

import org.junit.jupiter.api.Order;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;

import acme.testing.TestHarness;

public class SingUpAndBecomePatron extends TestHarness {


    @ParameterizedTest
    @CsvFileSource(resources = "/authenticated/patron/becomePatron.csv", encoding = "utf-8", numLinesToSkip = 1)
    @Order(10)
    public void positivesSingUpAndBecomePatron(final String company, final String statement,final String info) {

        super.signIn("inventor1", "inventor1");
```

SingUpAndBecomeInventor.java

```java
package acme.testing.authenticated.inventor;

import org.junit.jupiter.api.Order;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;

import acme.testing.TestHarness;

public class SingUpAndBecomeInventor extends TestHarness {


    @ParameterizedTest
    @CsvFileSource(resources = "/authenticated/inventor/becomeInventor.csv", encoding = "utf-8", numLinesToSkip = 1)
    @Order(10)
    public void positivesSingUpAndBecomeInventor(final String company, final String statement,final String info) {
```

Puesto que la herramienta mostraba el error se ha llevado a cabo renombrar las clases siguiendo el patrón propuesto, quedando finalmente de la siguiente manera e indicando la consola que ya no aparecen los malos olores



3.4 Mal olor 5 (D4)

El siguiente mal olor que comentar será el de la clase *InventorToolkitUpdateService.java* el cual indica que unificara las dos sentencias *if* que aparecían seguidas.

```
if(!errors.hasErrors("url")) {
    if(!entity.getUrl().isEmpty()) {
    boolean isUrl;
    isUrl = (entity.getUrl().startsWith("http") || entity.getUrl().startsWith("www")) && entity.getUrl().contains(".");
    errors.state(request, isUrl, "url", "inventor.toolkit.form.error.url");
}
}
```

Para solucionar este caso, se ha incluido la segunda sentencia del *if* junto con la primera añadiendo un operador *&&* para que cumpliera ambas restricciones.

```
if(!errors.hasErrors("url") && (!entity.getUrl().isEmpty())) {
    boolean isUrl;
    isUrl = (entity.getUrl().startsWith("http") || entity.getUrl().startsWith("www")) && entity.getUrl().contains(".");
    errors.state(request, isUrl, "url", "inventor.toolkit.form.error.url");
}
```

3.5 Mal olor 6 (D4)

Finalmente, la consola quedaría de la siguiente manera indicando el último mal olor por resolver.

Resource	Date	Description
AdministratorDashboardShowService.java		Convert this Map to an EnumMap.

Este aviso nos salió en la anterior entrega el cual nos pedía que en la entrada de la línea 52 cambiáramos el *Map* que habíamos declarado en un tipo *EnumMap*.

```
51
52        final Map<Status, Integer> totalNumberOfPatronagesByStatus = new HashMap<Status, Integer>();
```

Como se ha comentado al principio, este mal olor nos suponía cambiar gran parte del código en algunas clases relacionadas del método. Puesto que no supone un problema grave, se ha decidido por mayoría dejar este mal olor en el proyecto.

3. Conclusiones

Como conclusión respecto a este documento la herramienta que se proporciona como SonarLint nos resulta una herramienta muy eficaz y útil puesto que nos muestra diversos casos en los que nuestra calidad del código puede mejorar.

Tendremos en cuenta la frecuencia de ejecución de dicha herramienta durante el desarrollo del proyecto para futuras entregas para poder llegar al final del desarrollo con menos malos olores posibles.

4. Bibliografía

Diapositivas de la lección: *L03 – Displaying Data – S04 – Functional testing (laboratory)*