DP2 2021-2022

LintReport

Acme Toolkits

Repositorio:

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

Miembros:

- Caro Albarrán, Francisco Andrés (fracaralb@alum.us.es)
- Gallego Huerta, Alberto(albgalhue@alum.us.es)
- Martín Luque, José Manuel (josmarluq@alum.us.es)
- Reyes Madrid, Francisco (frareymad@alum.us.es)
- Sillero Manchón, Jorge (jorsilman@alum.us.es)

GRUPO E3.06

23 de mayo de 2022

Índice

Resumen ejecutivo

Este informe contendrá los posibles malos olores que reporte Lint sobre nuestro proyecto, así como las correcciones de estos.

Tabla de revisiones

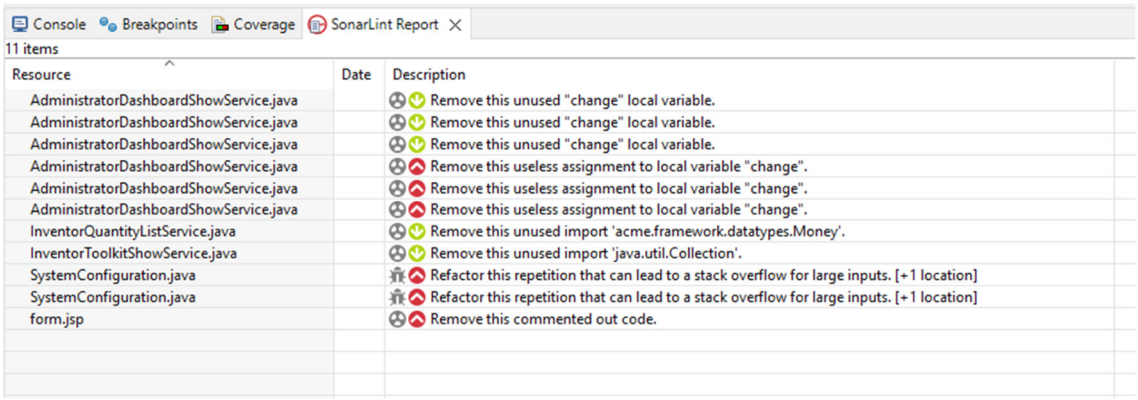
FECHA	VERSIÓN	DESCRIPCIÓN
2022-05-23	1	Versión inicial

Introducción

En este documento se va a detallar la información sobre malos olores de código que reporte Lint. Se justificarán los distintos tipos de malos olores si es necesario.

Contenido

El resultado de ejecutar SonarLint es el siguiente:



Los seis primeros olores son los mismos, pero en diferentes métodos de la clase AdministratorDashboardShowService.

```
111         for(Object[] o: this.repository.getAverageBudgetOfPatronagesByStatus()) {
112             Money m = new Money();
113             m.setAmount((Double) o[1]);
114             m.setCurrency(o[2].toString());
115             AuthenticatedMoneyExchangePerformService moneyExchange = new AuthenticatedMoneyExchangePerformService();
116             MoneyExchange change = moneyExchange.computeMoneyExchange(m, systemCurrency);
117 
118             avgBudgetOfPatronagesByStatus.put(o[0].toString(), (Double) o[1]);
119         }
```

Para solucionarlo, lo que hemos hecho ha sido eliminar la variable, ya que no se usaba.

Los olores números siete y ocho se tratan de un 'import' que no se usa. La solución es muy simple, los hemos eliminado, ya que no se usan.

```
11 import acme.framework.controllers.Request;
12 import acme.framework.datatypes.Money;
13 import acme.framework.services.AbstractListService;
```

```
11 import acme.framework.controllers.Request;
12 import acme.framework.datatypes.Money;
13 import acme.framework.services.AbstractListService;
```

Los dos siguientes son debido a que a la hora de hacer la tarea de spam intentamos cambiar el patrón de las palabras para que nos resultase más fácil hacer la tarea.

```
    @NotBlank
    Refactor this repetition
    @Pattern(regexp = "^[\\p{L}]+([ ']{1}[\\p{L}]+)*(, [\\p{L}]+([ ']{1}[\\p{L}]+)*)*$")
    //@Pattern(regexp = "^[\\p{Ll}]+([ ']{1}[\\p{Ll}]+)*(, [\\p{Ll}]+([ ']{1}[\\p{Ll}]+)*)*$") // lowercase only
    protected String            strongSpamWords;

    @Digits(integer = 2, fraction = 2)
    @Range(min=1, max=100)
    protected Double            strongSpamThreshold;

    @NotBlank
    @Pattern(regexp = "^[\\p{L}]+([ ']{1}[\\p{L}]+)*(, [\\p{L}]+([ ']{1}[\\p{L}]+)*)*$")
    //@Pattern(regexp = "^[\\p{Ll}]+([ ']{1}[\\p{Ll}]+)*(, [\\p{Ll}]+([ ']{1}[\\p{Ll}]+)*)*$") // lowercase only
    protected String            weakSpamWords;

    @Digits(integer = 2, fraction = 2)
    @Range(min=1, max=100)
    protected Double            weakSpamThreshold;
```

Esto lo hemos solucionado dejando el patrón como lo teníamos inicialmente, puesto que la tarea de spam no la hemos hecho de momento. Si hacemos la tarea de spam para el siguiente entregable cambiaremos la forma de implementarla, para que no salga ese olor.

Para finalizar, el último olor es debido a un comentario en un 'form.jsp'.

```
<acme:form>
    <acme:input-textbox code="administrator.announcement.form.label.title" path="title" placeholder="administrator.announcement.form.label.title"/>
    <acme:input-textbox code="administrator.announcement.form.label.body" path="body" placeholder="administrator.announcement.form.label.body"/>
    <!--<acme:input-moment code="administrator.announcement.form.label.creationMoment" path="creationMoment"/> -->
    <acme:input-checkbox code="administrator.announcement.form.label.critical" path="criticalFlag"/>
    <acme:input-textbox code="administrator.announcement.form.label.link" path="link" placeholder="administrator.announcement.form.label.link"/>
    <acme:input-checkbox code="administrator.announcement.form.label.confirm" path="confirm"/>

    <acme:submit code="administrator.announcement.form.button.create" action="/administrator/announcement/create"/>

</acme:form>
```

Lo hemos eliminado para que se solucione y no de problemas en un futuro.

Conclusiones

SonarLint nos ha reportado 11 errores de código, la mayoría de estos se solucionaban eliminado la línea que daba error. Gracias a esta herramienta podemos seguir las buenas prácticas recomendadas y nos evitamos tener problemas en un futuro.

Al ejecutar por segunda vez el SonarLint, ya no aparecen olores de código.

Bibliografía

Intencionalmente en blanco.