



# Plugin ecoCode

## Installation

## Test du plugin

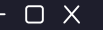


*Juillet 2023*





# <Sommaire />



<Présentation />

<Intégration du plugin />

<Tests du plugin />

<Annexes />

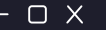
L'objectif de ce document est d'expliquer comment installer et configurer le plugin ecocode dans un environnement professionnel, sur exemple du Crédit Agricole.

Cette première version sera enrichie au fur et à mesure de vos retours.





# <Présentation />



ecoCode est un projet collectif visant à réduire l'empreinte environnementale des logiciels au niveau du code. L'objectif du projet est de fournir une liste d'analyseurs de code statiques permettant de mettre en évidence les structures de code pouvant avoir un impact écologique négatif : surconsommation d'énergie et de ressources, "fatware", raccourcissement de la durée de vie des terminaux, etc.

**Link Github :** [ecoCode/src at main · cnumr/ecoCode · GitHub](https://github.com/ecoCode/src)



ecoCode s'appuie sur des catalogues évolutifs de bonnes pratiques, pour différentes technologies. Un plugin SonarQube implémente ensuite ces catalogues comme règles pour scanner les projets.

5 technologies sont actuellement supportées par le plugin :

- ✓ Java
- ✓ PHP
- ✓ Python
- ✓ Android
- ✓ iOS
- ✓ Javascript

**CHALLENGE**  
< /> ecoCode



Un hackathon sur l'écoconception a été organisé par le Crédit Agricole et a pour ambition de rassembler toute personne souhaitant contribuer à la progression de l'outil ecoCode. 2 jours étaient dédiés à la spécification et au développement de nouvelles règles pour enrichir le plugin SonarQube d'au moins 2 nouvelles règles qui permettraient d'intégrer la dimension d'écoconception à la construction des codes sources.





# < Présentation Sonarqube />



Développé par [SonarSource](#), SonarQube est un logiciel de gestion de la qualité du code. Il est utilisé pour inspecter le code source des logiciels et applications en développement. Il permet de mesurer la qualité de plusieurs façon : respect des règles de code, documentation du code, analyse des tests unitaires mis en place, détecter des bugs, vulnérabilités de sécurités, instances de code dupliqué et autres anomalies pouvant nuire à la qualité du code source. SonarQube se base sur l'analyse statique du code, ce qui lui permet d'étudier le code source avant même l'exécution de l'application.

Lors de l'analyse d'un projet sous Sonarqube, un profil de qualité est appliqué en fonction du langage détecté. Ce profil contient l'ensemble des règles qui vont servir à identifier les points à corriger. Vous pouvez modifier le profil afin de répondre aux besoins spécifiques d'un projet. Il existe des profils pour Java, C, C++, C#, PHP, Objective C et bien d'autres.

C'est là qu'on va pouvoir configurer les différents profils de qualité par langage. On peut activer/désactiver les règles et modifier leurs importances dans chaque profil





## Installation du plugin ecoCode

- Sur la market place de SonarQube ou depuis le github « green code initiative », télécharger les plugins ecoCode compatibles avec votre version de SonarQube
- Pour chaque plugin, copier le fichier JAR du plugin dans le répertoire des extensions de SonarQube
- Activer le plugin : Dans la section "Administration", rechercher le menu "Extensions" ou "Plugins".

Une fois le plugin ecoCode installé et configuré, vous pouvez effectuer une vérification pour vous assurer que tout fonctionne correctement. Accédez à votre projet SonarQube et vérifiez si les fonctionnalités et les règles du plugin sont correctement appliquées et visibles dans les résultats d'analyse.





# < Integration du plugin />



## Disponibilité du Plugin

Après l'intégration du plugin, des règles taguées « ecoCode » apparaissent dans la liste des règles SonarQube

The screenshot shows the SonarQube interface with the 'Rules' tab selected. A search filter for 'ecocode' is applied, showing 34 results. The list of rules includes:

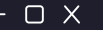
Rule Name	Language	Category	Tags
Avoid creating getter and setter methods in classes	Python	Code Smell	eco-design, eco...
Avoid getting the size of the collection in the loop	Java	Code Smell	eco-design, eco...
Avoid multiple if-else statement	Java	Code Smell	eco-design, eco...
Avoid Spring repository call in loop	Java	Code Smell	eco-design, eco...
Avoid SQL request in loop	Java	Code Smell	eco-design, eco...
Avoid SQL request in loop	Python	Code Smell	eco-design, eco...
Avoid SQL request in loop	PHP	Code Smell	eco-design, eco...

Interface sonarqube, contexte Crédit Agricole





# < Règles Java />

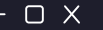


Règles	Description
<b>Avoid Spring repository call in loop</b>	L'utilisation d'une interface Spring Repository en boucle induit des calculs inutiles par le CPU
<b>Avoid SQL request in loop</b>	L'exécution de requêtes SQL en boucle induit des calculs inutiles par le processeur, l'utilisation de la RAM et le transfert réseau.
<b>Avoid the use of Foreach with Arrays</b>	L'utilisation de List au lieu de Arrays avec Foreach économise les calculs de cycles CPU et la consommation de RAM
<b>Do not call a function when declaring a for-type loop</b>	Eviter d'appeler des fonctions lors de la déclaration d'une boucle de type for afin d'éviter les appels de fonction à chaque itération. ça économise des cycles CPU
<b>Do not unnecessarily assign values to variables</b>	Eviter d'attribuer inutilement des valeurs aux variables. Ça augmente l'utilisation de la mémoire RAM
<b>Don't use the query SELECT * FROM</b>	Les serveurs de bases de données doivent résoudre des champs concernant le schéma. Connaître et utiliser le schéma directement permet d'économiser les cycles CPU et le transfert réseau.
<b>Use ++i instead of i++</b>	La forme i++ crée une variable temporaire alors que ++i ne le fait pas. Cela permet d'économiser des cycles CPU.
<b>Avoid getting the size of the collection in the loop</b>	Lorsqu'on effectue une itération sur une collection, récupérer la taille de la collection à l'avance pour éviter de la récupérer sur chaque itération, cela permet d'économiser les cycles CPU, et donc de consommer moins d'énergie.
<b>Avoid multiple if-else statement</b>	Lorsqu'on utilise trop de déclarations conditionnelles if-else, cela a un impact sur les performances puisque JVM devra comparer les conditions. On peut envisager d'utiliser une instruction switch au lieu d'une instruction multiple si possible. La instruction switch a un avantage de performance sur if – else.
<b>Free resources</b>	try-with-resources Statement doit être implémenté pour tout objet qui implémente l'interface AutoCloseable, il économise des ressources informatiques.





# < Règles Java />



Règles	Description
<b>Avoid using global variables</b>	Lors de l'appel d'une variable globale, le moteur d'interprétation doit vérifier qu'elle existe dans tous les périmètres, qu'elle a une valeur, etc. Passer des variables globales en arguments leur donne le statut des variables locales à l'intérieur de la fonction, ce qui permet d'économiser du temps de calcul (cycles CPU)
<b>Avoid usage of static collections</b>	Si vous souhaitez utiliser des collections statiques les rendre définitives et créer par exemple une singleton si nécessaire contenant les collections. Les champs statiques sont plus compliqués à gérer pour le Garbage Collector et peuvent conduire à des fuites de mémoire.
<b>Avoid using Pattern.compile() in a non-static context</b>	Cette opération nécessite une quantité non négligeable de puissance de calcul, En utilisant un seul match économise les cycles CPU et la consommation de RAM.
<b>Don't concatenate Strings in loop, use StringBuilder instead.</b>	Les chaînes sont immuables donc chaque fois que vous concatenez une chaîne, une nouvelle chaîne est créée. C'est un gaspillage de mémoire et de CPU.
<b>Don't set const parameter in batch update =&gt; Put its in query</b>	La création de ce paramètre et sa destruction consomme inutilement les cycles CPU et la RAM.
<b>Initialize builder/buffer with the appropriate size</b>	Si vous savez à l'avance combien de caractères seront ajoutés, initialisez le builder/buffer avec la taille appropriée. Ils n'auront donc jamais besoin d'être redimensionnés. Cela économise les cycles CPU et consomme donc moins d'énergie.
<b>Optimize read file exceptions</b>	L'optimisation des exceptions de lecture de fichiers consiste à gérer efficacement les erreurs potentielles lors de la lecture de fichiers, afin d'améliorer les performances et de garantir une exécution fluide du programme.
<b>Use PreparedStatement instead of Statement</b>	Utilisez PreparedStatement au lieu de Statement, parce que SQL n'exécutera la requête qu'une seule fois, alors que si vous n'utilisiez qu'une seule instruction, elle exécutera la requête à chaque fois et induirait ainsi des calculs inutiles par le CPU et donc une consommation d'énergie superflue.
<b>Use System.arraycopy to copy arrays</b>	Les programmes passent la plupart du temps dans des boucles. Celles-ci peuvent être gourmandes en ressources, surtout lorsqu'elles intègrent un traitement lourd (accès IO). De plus, la taille des données et le traitement à l'intérieur des boucles ne permettront pas une utilisation complète des mécanismes matériels tels que le cache ou les mécanismes d'optimisation du compilateur.







# < Règles Python />

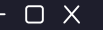


Règles	Description
<b>Avoid using try-catch-finally statement</b>	A l'intérieur des parties de code complexes (par exemple plusieurs boucles, constructions de données complexes...), éviter d'utiliser try...catch...finally. Lorsqu'une exception est levée, une variable (l'exception elle-même) est créée dans un bloc catch et sa destruction consomme des cycles CPU et de la RAM inutiles. Préférez utiliser des tests logiques dans ce cas.
<b>Do not call a function when declaring a for-type loop</b>	Eviter d'appeler des fonctions lors de la déclaration d'une boucle de type for afin d'éviter les appels de fonction à chaque itération. ça économise des cycles CPU
<b>Do not call global variables directly inside functions</b>	Lorsque une fonction appelle des variables globales, beaucoup de cycles CPU sont consommés
<b>Don't use the query SELECT * FROM</b>	Les serveurs de bases de données doivent résoudre des champs concernant le schéma. Connaître et utiliser le schéma directement permet d'économiser les cycles CPU et le transfert réseau.
<b>Avoid SQL request in loop</b>	L'exécution de requêtes SQL en boucle induit des calculs inutiles par le processeur, l'utilisation de la RAM et le transfert réseau.
<b>Avoid creating getter and setter methods in classes</b>	Eviter de créer des méthodes getter et setter dans les classes est recommandé pour plusieurs raisons. Cela permet de préserver le principe d'encapsulation en cachant les détails internes de la classe, de maintenir la flexibilité en évitant les dépendances excessives et les mises à jour fastidieuses.





# < Règles PHP />



Règle	Description
<b>Avoid SQL request in loop</b>	L'exécution de requêtes SQL en boucle induit des calculs inutiles par le processeur, l'utilisation de la RAM et le transfert réseau.
<b>Avoid using double quote ("), prefer using simple quote (')</b>	PHP permet de déclarer une chaîne de caractères avec des guillemets simples ou doubles. L'utilisation de guillemets doubles permet aux développeurs d'insérer des variables qui seront substituées lors de l'exécution. Lorsque la chaîne n'a pas de variables, l'utilisation de guillemets simples évite à PHP de rechercher des variables inexistantes. Cela permettra d'économiser la consommation de cycles CPU.
<b>Avoid using try-catch-finally statement</b>	A l'intérieur des parties de code complexes (par exemple plusieurs boucles, constructions de données complexes...), éviter d'utiliser try...catch...finally. Lorsqu'une exception est levée, une variable (l'exception elle-même) est créée dans un bloc catch et sa destruction consomme des cycles CPU et de la RAM inutiles. Préférez utiliser des tests logiques dans ce cas.
<b>Do not call a function when declaring a for-type loop</b>	Eviter d'appeler des fonctions lors de la déclaration d'une boucle de type for afin d'éviter les appels de fonction à chaque itération. ça économise des cycles CPU
<b>Don't use the query SELECT * FROM</b>	Les serveurs de bases de données doivent résoudre des champs concernant le schéma. Connaître et utiliser le schéma directement permet d'économiser les cycles CPU et le transfert réseau.
<b>Use ++i instead of i++</b>	La forme i++ crée une variable temporaire alors que ++i ne le fait pas. Cela permet d'économiser des cycles CPU.
<b>Prefer local variables to globals</b>	Il est préférable d'utiliser des variables locales plutôt que des variables globales, car cela permet d'améliorer l'encapsulation, la cohésion et la modularité du code, tout en réduisant les effets de bord et les collisions de noms de variables. L'utilisation de variables locales favorise un code plus clair, plus facile à maintenir et à réutiliser.
<b>Use of methods for basic operations</b>	L'utilisation de méthodes pour les opérations de base présente plusieurs avantages : réutilisation du code, lisibilité améliorée, facilité de maintenance et abstraction modulaire. En encapsulant les opérations de base dans des méthodes, on évite la duplication de code.






## Création d'un quality profile ecoCode pour intégrer les nouvelles règles

Pour réaliser un test avec le plugin, il est nécessaire de créer un quality profile

### Exemple Crédit Agricole

Afin de ne pas créer d'effet de bord sur des projets Sonar qui utilisent les profiles existants, un nouveau profil de qualité « ecocode » sur le langage Java est créé. Ce profil surcharge le profil par défaut Java des projets CACIB « CACIB Security » et inclut les nouvelles règles Java du plugin ecoCode

**Etape 1 :** Création du profil « ecocode2 » surchargeant le profile existant « CACIB Security: en cliquant sur le bouton **Copy** dans le menu  , à droite du profile « CACIB Security » et en renseignant le nom du nouveau profile « ecoCode2 »

**Etape 2 :** Sauvegarde du fichier de config XML du profile pour mise à jour en cliquant sur le bouton **Back up** dans le menu 



The screenshot shows the SonarQube interface for Quality Profiles. The table lists several profiles, and a context menu is open over the 'CACIB security' profile, showing 'Back up' and 'Compare' options. The right sidebar shows a list of rules.

Profile Name	Issues	Quality Gates	Created	Updated	Rules
CACIB quality	0	466	last year	Never	
CACIB security	0	466	last year	Never	
Java-Quality (SECAPI ROOT, 02/19/2019, DO NOT MODIFY)	0	468	last year		
CAGIP quality	0	466	last year	Never	

Context menu for 'CACIB security':

- Back up
- Compare

Rules list (partial):

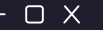
- Don't use the query SELECT \* FROM PHP, not yet activated
- Do not call a function when declaring a for-type loop Python, not yet activated
- Avoid using try-catch-finally statement Python, not yet activated
- Don't use the query SELECT \* FROM Python, not yet activated
- Do not call global variables directly inside functions

Exemple de quality profile, contexte Crédit Agricole





# < Tests du plugin />



## Création d'un quality profile ecoCode pour intégrer les nouvelles règles

Etape 3 : Récupérer les Ids des nouvelles règles dans SonarQube.

Pour le langage Java et dans la version installée dans le plugin, on a 19 nouvelles règles

Rule Name	Language	Type	Tag
Avoid getting the size of the collection in the loop	Java	Code Smell	eco-design, eco...
Avoid multiple if-else statement	Java	Code Smell	eco-design, eco...
Avoid Spring repository call in loop	Java	Code Smell	eco-design, eco...
Avoid SQL request in loop	Java	Code Smell	eco-design, eco...
Avoid the use of Foreach with Arrays	Java	Code Smell	eco-design, eco...
Avoid usage of static collections.	Java	Code Smell	eco-design, eco...
Avoid using global variables	Java	Code Smell	eco-design, eco...
Avoid using Pattern.compile() in a non-static context.	Java	Code Smell	eco-design, eco...
Do not call a function when declaring a for-type loop	Java	Code Smell	eco-design, eco...
Do not unnecessarily assign values to variables	Java	Code Smell	eco-design, eco...

Pour chaque règle on récupère l'ID : par exemple pour la règle «Avoid getting the size of the collection in the loop »

**Avoid getting the size of the collection in the loop** (ecocode-java:EC3)

Code Smell, Minor, eco-design, ecoCode, performance

Available Since May 23, 2023 | ecoCode (Java) | Constant/Issue: 5min

When iterating over any collection, fetch the size of the collection in advance to avoid fetching it on each iteration, this saves CPU cycles, and therefore consumes less power. The example provided below illustrates what should be avoided.

Noncompliant Code Example






## Création d'un quality profile ecoCode pour intégrer les nouvelles règles

**Etape 4** : Ajouter les nouvelles règles dans le fichier de config XML

Pour le langage Java et dans la version installée dans le plugin, on a 7 nouvelles règles :

Dans le fichier, on indique l'identifiant de la règle obligatoire et non modifiable en vert et comment on les caractérise en bleu

**Etape 5** : intégrer le nouveau fichier de config XML en cliquant sur le bouton « **Restore** » dans le menu 

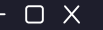
à droite du profile « eCocode2 »

Attention : Le bouton « Restore » est désactivé pour les utilisateurs non admin de Sonar





# < Tests du plugin />



## Création d'un quality profile ecoCode pour intégrer les nouvelles règles

### Etape 6: les nouvelles règles apparaissent dans le nouveau profil

Search for rules...

Language

Type CODE SMELL Clear

- Bug 0
- Vulnerability 0
- Code Smell 7
- Security Hotspot 0

Tag ECO-CONCEPTION Clear

eco 7

eco-conception 7

1 of 1 shown

Ctrl + click to add to selection

- Repository
- Default Severity
- Status
- Security Category
- Available Since
- Template
- Quality Profile CACIB E... Clear

Rule	Language	Category	Tag
Avoid Spring repository call in loop	Java	Code Smell	eco-conception
Avoid SQL request in loop	Java	Code Smell	eco-conception
Avoid the use of Foreach with Arrays	Java	Code Smell	eco-conception
Do not call a function when declaring a for-type loop	Java	Code Smell	eco-conception
Do not unnecessarily assign values to variables	Java	Code Smell	eco-conception
Don't use the query SELECT * FROM	Java	Code Smell	eco-conception
Use ++ instead of i++	Java	Code Smell	eco-conception

7 of 7 shown

### Avoid Spring repository call in loop

Code Smell Minor eco-conception Available Since Jul 12, 2022 Collectif Conception Numérique Responsable (Java) Constantissue: 50min

Using Spring repository in loop induced unnecessary calculation by the cpu so unless energy consumption

Noncompliant Code Example

```
private final List<Integer> ids = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

List<Employee> employees = new ArrayList<>();

for (Integer id: ids) {
    Optional<Employee> employee = employeeRepository.findById(id); // Noncompliant
    if (employee.isPresent()) {
        employees.add(employee.get());
    }
}
```

Compliant Solution

```
private final List<Integer> ids = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
List<Employee> employees = employeeRepository.findAllById(ids);
```

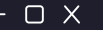
Quality Profiles

CACIB ecocode 2 Minor





# < Tests du plugin />



## Configuration du quality profile ecoCode pour scanner les projets

### Association du Quality profile à un projet

The screenshot displays the SonarQube interface for a project named 'kube-demo'. The top navigation bar includes 'Overview', 'Issues', 'Security Hotspots', 'Security Reports', 'Measures', 'Code', 'Activity', and 'More'. The main content area is divided into two sections: 'QUALITY GATE STATUS' and 'MEASURES'. The 'QUALITY GATE STATUS' section shows a green box with the text 'Passed' and 'All conditions passed.'. The 'MEASURES' section is divided into two columns: 'New Code' and 'Overall Code'. The 'New Code' column shows 'Since 2' and 'Started 20 days ago'. The 'Overall Code' column shows 'Reviewed'. The 'MEASURES' section also includes a table of metrics:

Metric	Value	Grade
New Bugs	0	A
New Vulnerabilities	0	A
New Security Hotspots	0	A
Added Debt	0	A
New Code Smells	0	A

The right sidebar shows 'Project Information' with the following details:

- Description: PRIVATE
- No tags
- Lines of Code: 0
- Quality Gate used: (Default) Sonar way
- Quality Profiles used: (Java) CACIB ecoCode 2
- Project Key: com.cacib.unc.kube-demo (Copy)
- Set notifications: >





# < Tests du plugin />



## Scan du projet

Les nouvelles règles d'ecoconception apparaissent dans le rapport d'analyse du projet

The screenshot displays the SonarQube interface for a project named 'kube-demo'. The main area shows a list of issues related to 'eco-conception' (eco-design) rules. The issues are categorized as 'Code Smell' and are marked as 'Minor' severity. The issues include:

- Remove this useless assignment to local variable "result". Why is this an issue? (15min effort)
- Remove this unused "result" local variable. Why is this an issue? (5min effort)
- The variable result is not assigned. Why is this an issue? (5min effort)
- The variable k is not assigned. Why is this an issue? (5min effort)
- Remove this useless assignment to local variable "result". Why is this an issue? (15min effort)
- Remove this unused "result" local variable. Why is this an issue? (5min effort)
- The variable result is not assigned. Why is this an issue? (5min effort)
- The variable k is not assigned. Why is this an issue? (5min effort)

The interface also shows a sidebar with filters and a top navigation bar with tabs for Overview, Issues, Security Hotspots, Security Reports, Measures, Code, Activity, and More. The top right corner indicates the date and time of the scan: September 30, 2022, 9:30 AM, Version 3.







# < Tests du plugin />



## Scan du projet

Les résultats sont simples à interpréter par un développeur : possibilité de filtrer sur le tag « ecoCode » pour afficher la liste des issues liées à l'ecoConception

The screenshot shows a web interface for a code quality tool. The top navigation bar includes 'Overview', 'Issues', 'Security Hotspots', 'Security Reports', 'Measures', 'Code', 'Activity', and 'More'. The current view is 'Issues' for the project 'kube-demo' and feature 'feature/eco-code'. The interface displays a list of issues filtered by the tag 'ECO-CONCEPTION'. The left sidebar shows a filter menu with 'Tag' set to 'ECO-CONCEPTION'. The main area shows four issues, all of which are 'Code Smell' with a severity of 'Minor' and a resolution of '5min effort'. The issues are: 'The variable result is not assigned', 'The variable k is not assigned', 'The variable result is not assigned', and 'The variable k is not assigned'. Each issue has a 'Comment' link and a 'Why is this an issue?' link. The interface also shows a 'Bulk Change' button, a 'Project Information' button, and a summary of '1 / 4 issues' and '20min effort'.

Severity	Count
Blocker	0
Critical	0
Major	0
Minor	4
Info	0

Tag	Count
eco-conception	4
unused	4
cert	2
cwe	2

Issue Title	Severity	Resolution	Effort
The variable result is not assigned	Minor	Open	5min effort
The variable k is not assigned	Minor	Open	5min effort
The variable result is not assigned	Minor	Open	5min effort
The variable k is not assigned	Minor	Open	5min effort





Challenge ecoCode: [Challenge ecoCode](#)

ecoCode : <https://www.ecocode.io/>

Plugin ecoCode Github : [ecoCode/src at main · cnumr/ecoCode · GitHub](#)

SonarQube : <https://docs.sonarqube.org/latest/>

SonarQube – Installation plugin : <https://docs.sonarqube.org/latest/setup/install-plugin/>

SonarQube – Quality Profiles : <https://docs.sonarqube.org/latest/instance-administration/quality-profiles/>

