



ecoCode

now there's a tool for mobile developers

Olivier Le Goaër



Le Pitch

Tandis que de plus en plus d'acteurs s'intéressent à l'éco-conception des sites Web, peu de travaux se sont intéressés aux apps mobile alors que ces dernières représentent un gisement d'économies d'énergie non-négligeable.

Mais comment former la future génération de développeurs mobile à l'écriture d'un code plus responsable ?

Pour tenter de répondre à cette question, cette présentation introduit un catalogue extensible d'*Energy Code Smells* pour Android ainsi qu'un outil d'analyse automatique pour les faire respecter.

Olivier Le Goaër

Qui suis-je ?

Enseignant-Chercheur en Informatique

➡ Collège STEE (Sciences & Technologies pour l'Energie et l'Environnement)

Recherche en Software Engineering

➡ Comment construire des logiciels + verts ?

Responsable du module Android en Master 2

➡ Cours en ligne (128k vues)





Climat : trajectoire +2°C

Est-ce que l'industrie lourde du numérique mobile va faire sa part ?

Ébriété numérique mobile

Mobile app developers

5,9

millions
Android devs



Available apps

2,56

millions
(Google Play Store)



Annual app downloads

109

milliards
(Google Play Store)



SmartPhone users

3,8

milliards
72% Android
27% iOS
1% autres

Behavior

4,2

heures / jour



2,8

millions
iOS devs



1,85

millions
iOS Apple Store



38

milliards
iOS Apple Store



Décarboner le numérique mobile

Software Editors

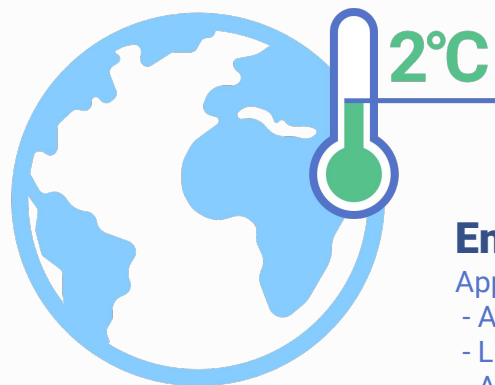
Développer des apps mobiles
moins énergivores

Tech Providers

Indice de réparabilité
Eco-rating
Batteries nouvelle
génération (graphène)
OS optimisés pour l'énergie
...

End-users

Appliquer des "éco-gestes" :
- Acheter en reconditionné
- Limiter son temps d'écran
- Activer le mode économie d'énergie dès que possible
...



Ordres de grandeur

éco-conception MATÉRIELLE

considère le cycle de vie des smartphones/tablettes, de la fabrication à la gestion des déchets et au recyclage



PRODUCTION

90%

de la consommation d'énergie
(et son corollaire en GES)

UTILISATION

10%

de la consommation d'énergie
(et son corollaire en GES)

éco-conception LOGICIELLE

vise essentiellement à réduire la consommation d'énergie des apps mobiles pendant leur phase d'usage



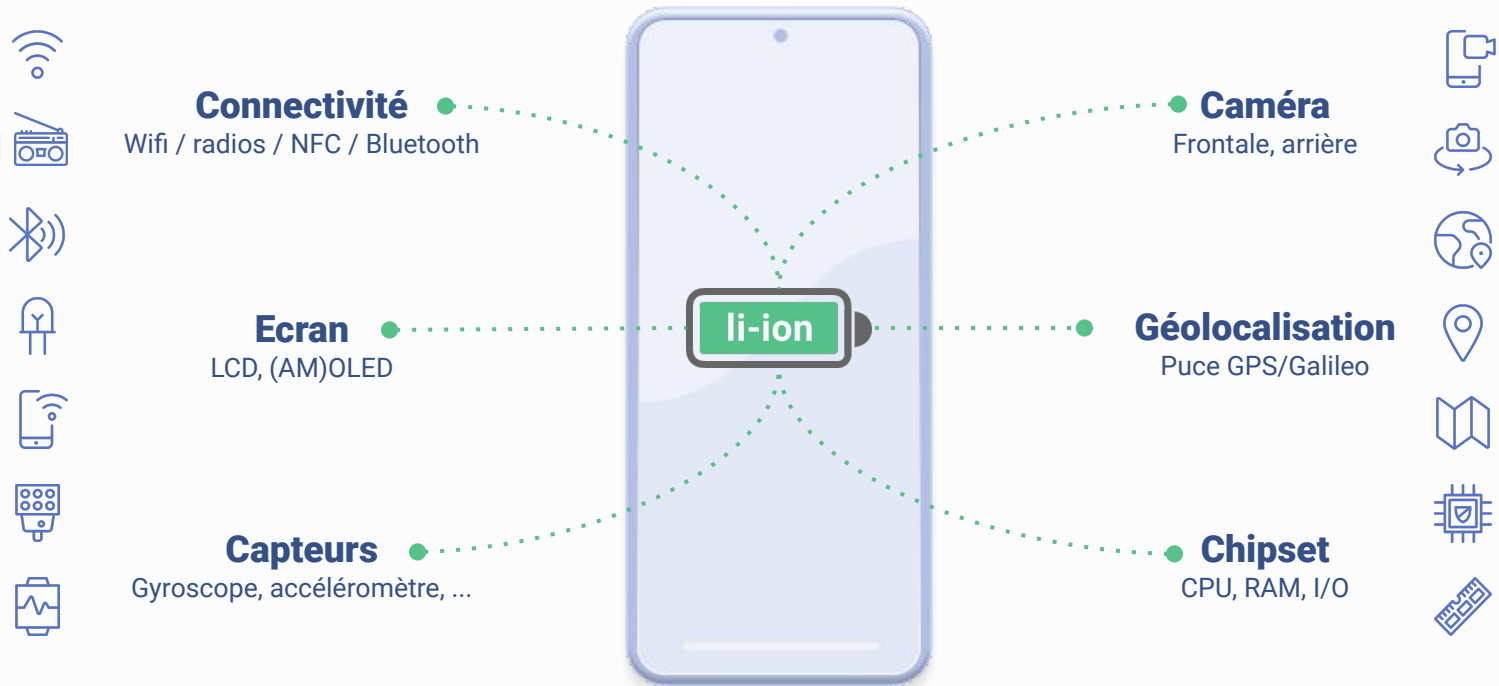
Energie en phase d'usage

Considérons un smartPhone moyenne gamme doté d'une batterie de **3,6V** et d'une capacité de **3000mAh**, qui consomme toute sa batterie chaque jour. Sa consommation d'énergie journalière est donc de $3Ah \times 3,6V = 10Wh$, et au bout d'un an, de **3,65kWh**.

Avec **3,8 milliards** de smartPhones actifs dans le monde, la consommation annuelle atteint **15 Téra watt-heure**.

- ➡ La facture énergétique va fatalement s'accroître dans le futur
- ➡ Agir sur les 10% permet aussi d'agir sur les 90% (*teaser :-)*

Mais qu'est ce qui vide la batterie ?



De l'intérêt des apps "eco-friendly"

Le logiciel gouverne le matériel gourmand en énergie. Une app qui gaspille de précieux micro-watts :

- 1 réduit la **durée de vie** de l'appareil car la batterie a un nombre de cycles de recharges limité
- 2 a un **effet cumulatif** important, à chaque exécution et sur chaque appareil où l'app est installée
- 3 peut créer des utilisateurs **mécontents** qui notent négativement l'app sur les magasins

Eco-conception logicielle

Concevoir des apps en gardant à l'esprit qu'elles ont une empreinte écologique.
Former la nouvelle génération de développeurs à l'éco-conception.

“ Les mesures relatives à l'écoconception des services numériques sont très faibles aujourd'hui. Quand on est formé au développement, on est peu sensibilisé à « l'écologie du code » ; il est nécessaire d'avancer sur ce sujet-là. ”

Cédric O, secrétaire d'État chargé de la transition numérique et des communications électroniques (décembre 2020)

Proposition de loi 2021

Proposition de loi "Réduire l'empreinte environnementale du numérique" 16 décembre 2020

ADOPTÉ

La commission de l'aménagement du territoire et du développement durable a adopté l'amendement présenté par **Guillaume CHEVROLIER** et **Jean-Michel HOULLEGATTE**, rapporteurs, visant à :

créer un référentiel général de l'écoconception fixant des critères de conception durable des sites web que devront respecter les plus gros fournisseurs de contenu.

Commission de l'AMÉNAGEMENT DU TERRITOIRE ET DU DÉVELOPPEMENT DURABLE

www.senat.fr

Ministère de l'Écologie @Ecologie_Gouv

Faisons de l'écologie et du numérique un duo gagnant !

@barbarapompili et @cedric_o présentent une feuille de route pour faire converger le numérique avec l'exigence environnementale. C'est une demande forte de la @Conv_Citoyenne.

+ d'infos : ecologie.gouv.fr/numerique-et-e...

La feuille de route « numérique et environnement », c'est la mobilisation de tous pour faire de l'écologie et du numérique un duo gagnant.

4:10 PM · 23 févr. 2021

57 51 Partager ce Tweet

Quelques critiques cependant

Les mastodontes sont logiquement ciblés en priorité...

- ➡ Par exemple, **2.291 milliards d'utilisateurs utilisent YouTube chaque mois** dans le monde ! Mais en réalité toutes les applications doivent faire leur part.

La loi se focalise sur l'éco-conception des sites web. C'est bien, mais cela occulte une réalité :

- ➡ **Environ 70% des vidéos YouTube sont vues depuis un périphérique mobile**, dont une large part via l'app mobile (disponible sur SmartPhone, Tablette et SmartTV)

L'éco-conception est un tout...

Une app mobile c'est un programme **côté client** uniquement

Cela ne dit rien sur l'efficiencia du programme **côté serveur**, ni de la consommation électrique des **infrastructures réseaux**



Même si l'app YouTube (Android/iOS) est éco-conçue, cela ne signifie pas que le service numérique YouTube l'est.



Comment agir ?

C'est à dire sans nécessairement travailler chez Google ou Apple

Les leviers d'action des développeurs

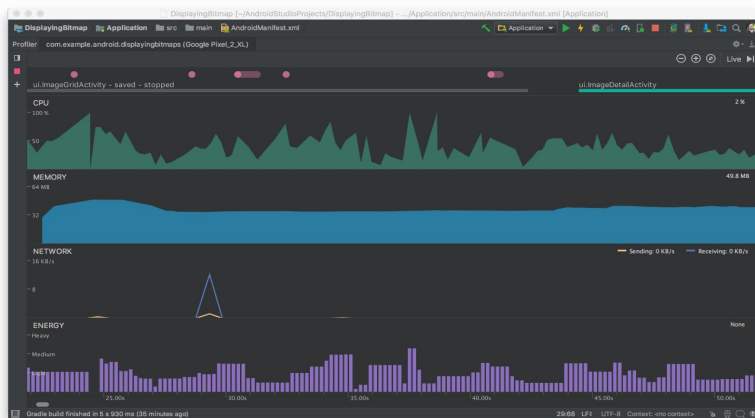
- **Dégraisser les fonctionnalités** des apps (phénomène d'"obésiciel") est sans conteste le gisement d'économie le plus fort, mais exige une étude **ad-hoc**.

➡ *Mesurer, comprendre, arbitrer, agir*

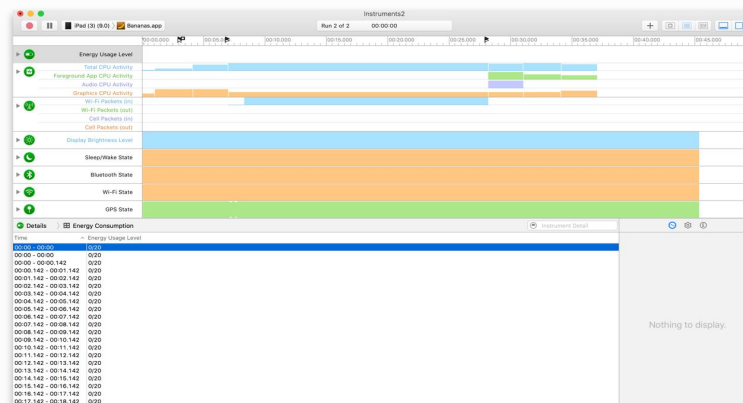
- **Améliorer la qualité du code** des apps a un impact plus modeste, mais est **systématisable** à l'ensemble de la production logicielle mondiale

➡ *Analyse et correction automatique*

Trouver les racines du mal n'est pas simple

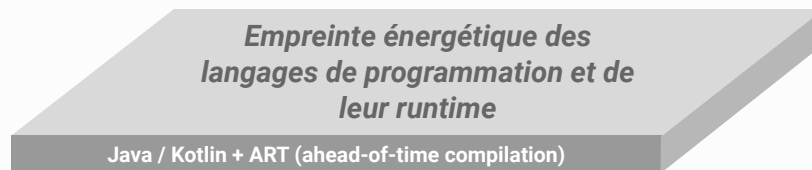


Android Energy Profiler (Android Studio)



iOS Energy Profiler (Xcode)

Pile logicielle Android et énergie



Code smells

“ Code Smells are patterns of code that suggest there might be a problem, that there might be a better way of writing the code or that more design perhaps should go into it. ”

- ➡ Concept popularisé via le **Clean Code**
- ➡ **N'est pas un bug** car n'empêche pas le programme de fonctionner
- ➡ Les bad smells contribuent à la **dette technique**

Energy code smells pour Android

Il semble logique d'appliquer ce concept à l'énergie, *a fortiori* sur des terminaux limités par leur batterie :

- ➡ Détecter les structures de code “pas terribles” pour la batterie
- ➡ Les smells se nichent potentiellement partout dans un projet Android : code source, manifest, gradle, ProGuard, ressources (xml, images, ...)
- ➡ Évaluer le temps nécessaire pour y remédier (rembourser la dette technique)

8 Bonnes pratiques

OPTIMIZED API

Take a look to battery-efficient APIs that have been specifically designed to substitute regular APIs

LEAKAGE

Check that an acquired resource is always released, to avoid unnecessary battery drain

BOTTLENECK

Avoid accumulation of data or operations that will require an energy peak to be processed

SOBRIETY

If acceptable from a user experience viewpoint, choose the less energy-greedy alternative

IDLENESS

When the app enter in a idle state, please adapt the energy consumption accordingly

POWER

Battery-driven operations help prolong the battery life

BATCH

Grouping individually costly operations allows to save energy globally

RELEASE

Favour the compile-time tasks that improve the energy efficiency of the final app

29 Energy Code Smells

OPTIMIZED API (2)

Fused Location, Bluetooth Low-Energy

LEAKAGE (3)

Media Leak, Sensor Leak, Everlasting Service

BOTTLENECK (4)

*Internet In The Loop, Wifi Multicast Lock,
Uncompressed Data Transmission, Uncached Data
Reception*

SOBRIETY (6)

*Dark UI, Day Night Mode, Brightness Override, Thrifty
Geolocation, Thrifty BLE, Thrifty Motion Sensor*

IDLENESS (6)

*Keep Screen On, Keep CPU On, Durable Wake Lock,
Rigid Alarm, Continuous Rendering, Keep Voice Awake*

POWER (4)

*Ignore Battery Optimizations, Companion in
background, Charge Awareness, Save Mode
Awareness*

BATCH (3)

Service@Boot-time, Sensor Coalesce, Job Coalesce

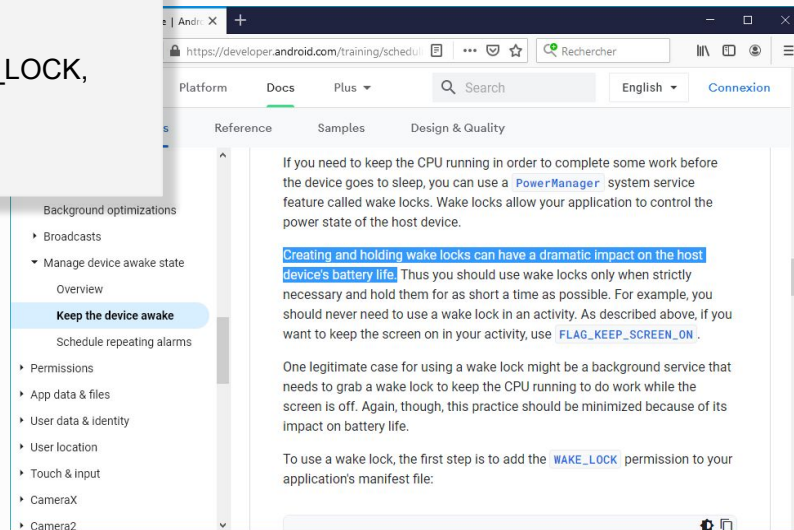
RELEASE (1)

Disable Obfuscation

Idleness

Keep CPU On

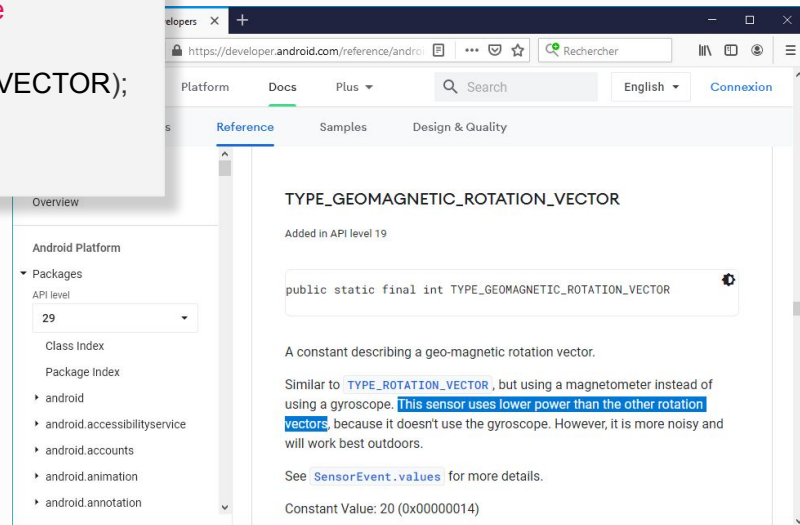
```
PowerManager powerManager = (PowerManager)
getSystemService(POWER_SERVICE);
WakeLock wakeLock =
powerManager.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,
    "MyApp::MyWakelockTag");
wakeLock.acquire();
```



Sobriety

Thrifty Motion Sensor

```
SensorManager mSensorManager = (SensorManager)
getSystemService(SENSOR_SERVICE);
// Accelerometer, Magnetometer, AND (when present) Gyroscope
mRotationSensor =
mSensorManager.getDefaultSensor(Sensor.TYPE_ROTATION_VECTOR);
mSensorManager.registerListener(this, mRotationSensor,
SENSOR_DELAY);
```





Outils pour le développeur

Computer-Aided Software Engineering (CASE)

Scanners de code pour Android

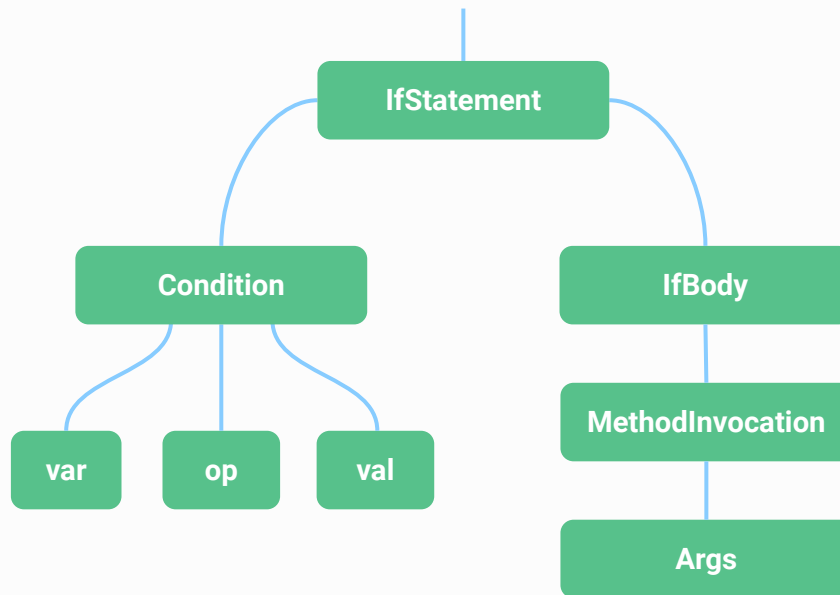
Embold		
Android Lint		
Detekt		
SonarQube		
SonarLint (version IntelliJ)		
Semgrep		
Codacy		
...		

Abstract Syntax Tree (AST)

```
WakeLock wakeLock =  
powerManager.newWakeLock(PowerManager.  
PARTIAL_WAKE_LOCK,  
"MyApp::MyWakelockTag");
```

```
if (wakeLock != null) {  
    wakeLock.acquire();  
}
```

demo.java



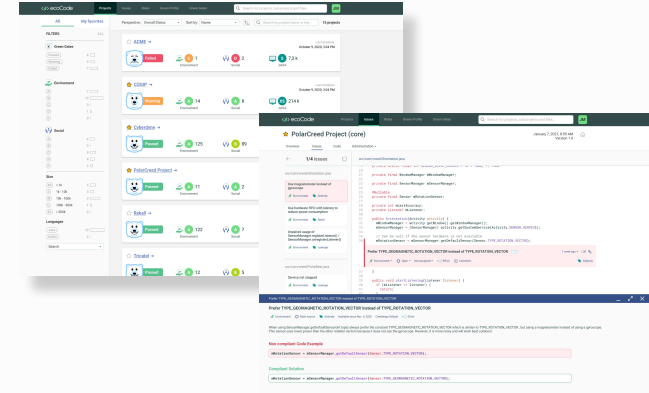
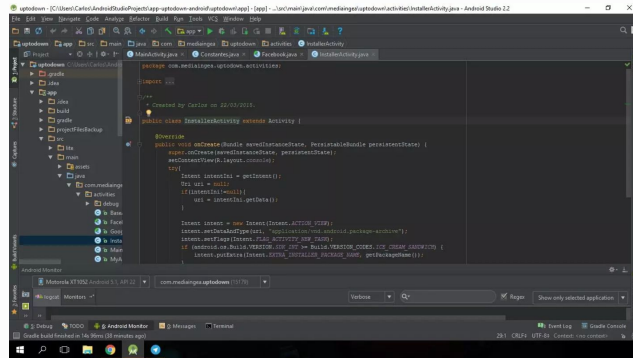
ecoCode

Un plugin SonarQube

- SonarQube est l'outil de qualité de code le plus populaire du marché
- **Analyse statique** de code source : le programme n'est jamais exécuté
- Indépendant de la taille, de la catégorie et des fonctionnalités de l'app
- S'adresse aux chefs de projets et lead devs
- SonarSource (éditeur de sonarQube) s'attend à voir émerger des plugins "Green"

ecoCode

En mode SaaS



Projet local avec Android Studio



<https://sonarqube.ecocode.io>

ecoCode

Base de règles

The screenshot shows the ecoCode web application interface. The top navigation bar includes 'ecoCode', 'Projects', 'Issues', 'Rules' (active), 'Quality Profiles', and 'Quality Gates'. A search bar is on the right. The left sidebar contains filters for 'Type' (CODE SMELL), 'Tag', 'Repository', 'Default Severity', 'Status', 'Security Category', 'Available Since', and 'Template'. The 'Quality Profile' is set to 'ECOCOD...'. The main area displays a list of 19 rules, each with a severity icon, name, language, and tags.

Severity	Rule Name	Language	Tags
🔴	Bottleneck: Internet In The Loop	Java	Code Smell, ecocode, environment, optimized-api
🔴	Idleness: Keep Screen On (addFlags)	Java	Code Smell, ecocode, environment, idleness
🔴	Idleness: Keep Screen On (setFlags)	Java	Code Smell, ecocode, environment, idleness
🔴	Leakage: Camera Leak	Java	Code Smell, ecocode, environment, leakage
🔴	Optimized API: Fused Location	Java	Code Smell, ecocode, environment, optimized-api
🔴	Idleness: Keep Cpu On	Java	Code Smell, ecocode, environment, idleness
🔴	Idleness: Durable Wake Lock	Java	Code Smell, ecocode, environment, idleness
🔵	Optimized API: Bluetooth Low-Energy	Java	Code Smell, ecocode, environment, optimized-api
🟢	Sobriety: Thrifty Motion Sensor	Java	Code Smell, ecocode, environment, sobriety
🔴	Leakage: Location Leak	Java	Code Smell, ecocode, environment, leakage
🟢	Sobriety: Brightness Override	Java	Code Smell, ecocode, environment, optimized-api
🔴	Leakage: SensorManager Leak	Java	Code Smell, ecocode, environment, leakage
🟢	Bottleneck: Wifi Multicast Lock	Java	Code Smell, bottleneck, ecocode, environment
🔴	Leakage: Media Leak (MediaRecorder)	Java	Code Smell, ecocode, environment, leakage

ecoCode

Projets analysés

ecoCode

ProjectsIssuesRulesQuality ProfilesQuality Gates

Search for projects...

SS

My FavoritesAll

Perspective: Overall StatusSort by: NameSearch by project name or key2 projects

Filters

Quality Gate

Reliability (Bugs)

Security (Vulnerabilities)

Maintainability (Code Smells)

Passed0

Failed2

A2

B0

C0

D0

E0

A2

B0

C0

D0

E0

A1

Spotify

Last analysis: yesterday

FAILED

0 Bugs

0 Vulnerabilities

79 Code Smells

0.0% Coverage

0.0% Duplications

9k Java, Xml

QuelPrixImmo

Last analysis: 5 hours ago

FAILED

0 Bugs

0 Vulnerabilities

8 Code Smells

0.0% Coverage

2.7% Duplications

2.2k Java, Xml

2 of 2 shown

This application is based on SonarQube™ but is not an official version provided by SonarSource SA.

Community Edition - Version 8.6.1 (build 3632) - LGPL v3 - Community - Documentation - Plugins - Web API - About

ecoCode

Qualité d'un projet

ecoCode

Projects Issues Rules Quality Profiles Quality Gates

Search for projects...

SS

QuelPrixImmo


master

June 23, 2021, 1:09 PM Version unspecified

Project Information

Overview Issues Security Hotspots Measures Code Activity

QUALITY GATE STATUS



FAILED

On Overall Code

8 Issues is greater than 4

MEASURES

New Code

Since June 23, 2021
Started 6 hours ago

Overall Code

0 Bugs

Reliability A

0 Vulnerabilities

Security A

0 Security Hotspots

Reviewed

Security Review A

2h 40min Debt

8 Code Smells

Maintainability A

ecoCode

Problèmes détectés

The screenshot displays the ecoCode interface for the 'QuelPrixImmo' project. The top navigation bar includes 'ecoCode', 'Projects', 'Issues', 'Rules', 'Quality Profiles', and 'Quality Gates'. A search bar and a 'SS' button are on the right. The project name 'QuelPrixImmo' and branch 'master' are shown, along with the date 'June 23, 2021, 1:09 PM' and 'Version unspecified'. The 'Issues' tab is active, showing a list of detected issues. On the left, a sidebar lists filters for 'Type' (Bug, Vulnerability, Code Smell), 'Severity', 'Scope', 'Resolution', 'Status', 'Security Category', 'Creation Date', 'Language', and 'Rule'. The main area shows a list of issues, each with a checkbox, a description, a 'See Rule' link, a date, a severity level, and a list of tags. The issues are sorted by 'Bulk Change' and show 1/8 issues with a total effort of 2h 40min.

ecoCode Projects Issues Rules Quality Profiles Quality Gates

QuelPrixImmo master June 23, 2021, 1:09 PM Version unspecified

Overview Issues Security Hotspots Measures Code Activity Project Information

Type

- Bug 0
- Vulnerability 0
- Code Smell 8

Severity

Scope

Resolution

Status

Security Category

Creation Date

Language

Rule

Search for rules...

(Java) Sobriety: Thrifty Geolocation 4

(Java) Leakage: Location Leak 2

(XML) Idleness: Keep Screen On 1

(XML) Sobriety: Dark UI (Theme) 1

4 shown

Bulk Change

↑ ↓ to select issues ← → to navigate 1 / 8 issues 2h 40min effort

src/main/AndroidManifest.xml

Using a light theme may have a significant impact on energy consumption on (AM)OLED screens. See Rule 2 days ago L15 ecocode, environment, sobriety

Code Smell Major Open Not assigned 1h effort Comment

src/main/java/fr/univpau/quelpriximmo/ResultListFragment.java

Failing to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L528 ecocode, environment, leakage

Code Smell Major Open Not assigned 20min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

Failing to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L528 ecocode, environment, leakage

Code Smell Major Open Not assigned 20min effort Comment

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528 ecocode, environment, optimized-api

Code Smell Major Open Not assigned 10min effort Comment

ecoCode

Détail d'un problème

The screenshot displays the ecoCode web application interface. At the top, there's a navigation bar with tabs for Projects, Issues, Rules, Quality Profiles, and Quality Gates. A search bar is present on the right. The main content area shows a project named 'QuelPrixImmo' with a 'master' branch. The 'Issues' tab is selected, showing a list of 3 issues. The first issue, 'Falling to call android.location.LocationManager#r...', is highlighted. The right pane shows the source code for 'ResultListFragment.java' with line numbers 523 to 528. The code includes an '@Override' annotation and a 'public void onResume()' method. A red line highlights the 'mLocationManager.requestLocationUpdates' call. Below the code, three issue details are listed, each with a description, a 'See Rule' link, a date (9 days ago), a severity (L528), and tags like 'ecocode, environment, leakage' or 'optimized-api'. The bottom pane shows the 'Sobriety: Thrifty Geolocation' rule details, including its severity (Code Smell), impact (Major), and a description of the rule's purpose: to conserve battery life by ensuring location updates are only sent when necessary.

ecoCode Projects Issues Rules Quality Profiles Quality Gates

Search for projects...

QuelPrixImmo master

June 23, 2021, 1:09 PM Version unspecified

Overview Issues Security Hotspots Measures Code Activity

Project Information

3 / 8 issues

...pau/quelpriximmo/ResultListFragment.java

Falling to call android.location.LocationManager#r... can drain the battery in just a few hours. Code Smell

Location updates should be done with a minimum time and distance interval. Code Smell

Location updates should be done with a minimum time and distance interval.

```
@SuppressWarnings("MissingPermission")
@Override
public void onResume() {
    super.onResume();
    mLocationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, this);
}
```

Falling to call android.location.LocationManager#removeUpdates() can drain the battery in just a few hours. See Rule 9 days ago L528

Code Smell Major Open Not assigned 20min effort Comment ecocode, environment, leakage

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528

Code Smell Major Open Not assigned 10min effort Comment ecocode, environment, optimized-api

Location updates should be done with a minimum time and distance interval. See Rule 9 days ago L528

Code Smell Major Open Not assigned 10min effort Comment ecocode, environment, optimized-api

Sobriety: Thrifty Geolocation

ecoCode-java:ESOB005

Code Smell Major ecocode, environment, optimized-api Available Since Jun 22, 2021 ecoCode (Java) Constant/Issue: 10min

With a call to `LocationManager#requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener)`, the location provider will only send your application an update when the location has changed by at least `minDistance` meters AND at least `minTime` milliseconds have passed. So `minTime` should be the primary tool to conserve battery life, and, to a lesser extent, `minDistance`. These two must imperatively be greater than 0.

ecoCode

Analyse SWOT

FORCES

Une démo fonctionnelle
Une longueur d'avance

FAIBLESSES

Il faut + de règles écologiques,
mais aussi sociales (≈100)

OPPORTUNITÉS

Intérêt publique/politique croissant
La réglementation évolue

MENACES

Les *big players* arrivent
Risque de *greenwashing*

A demain pour la démo :)

