



Koda : Forge pour les laboratoires

**Présentation de Koda, la
forge pour les laboratoires**

JoSy, Septembre 2021

Sommaire de présentation

- 1. Forge Logicielle ?**
- 2. Fonctionnalités / outils prévue pour la Forge labos**
- 3. Utilisateurs identifiés**
- 4. Autres fonctionnalités**
- 5. Organisation**
- 6. Planning**
- 7. Mot de la fin**

Forge Logicielle ?

Objectifs d'une forge logicielle

- **Gestion du code source**
 - Historisation
 - Travail collaboratif
- **Construction du projet (= « build », « intégration continue »)**
 - Génération automatique des binaires
 - Lancement des tests unitaires
 - Reproductibilité de la construction de l'application
 - "L'intégration continue est le principe de faire d'un processus d'intégration logicielle un «non-événement»." (Martin Fowler)
- **Stockage des artefacts générés**
 - Stockage du résultats du build (binaires exécutables, PDF, zip, ...)
 - Gestion des dépendances
- **Analyse de la qualité du code**
 - Suivi de l'évolution de la qualité du code dans le temps

Historique

- **Utilisation d'une forge par la DSI depuis plus de 10 ans**
 - CVS
 - Subversion (2012), Git (2013) avec annuaire LDAP dédié
 - Puis Hudson (→ Jenkins) pour les builds
 - Et Nexus pour le stockage des artéfacts (et proxy Maven)
- **2021 → Mise à niveau de la forge DSI**
 - Gitlab
 - Code source
 - Build
 - Artéfacts
 - Sonar
 - Pour le personnel de la DSI et des SSI des délégations régionales

1

Exemples

- Code source : Gitlab

The screenshot shows the GitLab interface for a project named 'Démono Josy'. The breadcrumb trail is 'DSI > Tests > demo-josy'. The project name is 'Démono Josy' with a lock icon and 'Project ID: 32'. It has 0 stars and 0 forks. Statistics include 445 commits, 2 branches, 22 tags, 538.8 MB files, and 1 GB storage. The current branch is 'master' in the 'demo-josy' directory. A recent merge commit is shown: 'Merge branch 'develop' into 'master'' by DERACO Stéphane, 2 weeks ago, with commit hash 0667beb7. Below the merge are buttons for 'README', 'CI/CD configuration', and 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', and 'Add Kubernetes cluster'. A table lists the project's directory structure and commit history.

Name	Last commit	Last update
config	update checkstyle rules	4 months ago
doc	Livraison sprint 25	2 months ago
gradle/wrapper	test fix publish	1 year ago

The screenshot shows the Koda Synchro IDE interface. The top-left pane displays the project structure with files like .mvn, src, .gitignore, .gitlab-ci.yml, ci_settings.xml (selected), mvnw, mvnw.cmd, and pom.xml. The main editor pane shows the content of ci_settings.xml, which is an XML configuration for Maven settings. The XML content is as follows:



```
1 <settings xmlns="http://maven.apache.org/SETTINGS/1.1.0" xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.1.0 http://maven.apache.org/SETTINGS/1.1.0.xsd">
2
3   <servers>
4     <server>
5       <id>gitlab-maven</id>
6       <configuration>
7         <httpHeaders>
8           <property>
9             <name>Job-Token</name>
10            <value>${env.CI_JOB_TOKEN}</value>
11          </property>
12        </httpHeaders>
13      </configuration>
14    </server>
15
16  </servers>
17 </settings>
18
```



DSI > Tests > demo-josy > Merge requests > !2



[Open](#) Created 2 weeks ago by [DERACO Stéphane](#) Owner [Edit](#) [Mark as draft](#) ▼


Exemple de Merge request


[Overview](#) 0 [Commits](#) 1 [Pipelines](#) 2 [Changes](#) 1

 **Request to merge** `template`  **into** `master` [Open in Web IDE](#) [Check out branch](#) [Download](#) ▼


 **Detached merge request pipeline #397** passed for `ca836305` 2 weeks ago ✓ ✓ ✓ [Download](#) ▼
Test coverage 32.00% from 1 job 









 [Approve](#) Approval is optional 

 Test summary contained no changed test results out of 12 total tests [View full report](#) [Expand](#)

 [Merge](#) Delete source branch

> **1 commit** and **1 merge commit** will be added to master. [Modify merge commit](#)

0 0  [Oldest first](#) ▼ [Show all activity](#) ▼

[Write](#) [Preview](#) **B** *I* ” </>        

Write a comment or drag your files here...

• Intégration continue : Gitlab CI

DSI > ... > Forge > Koda Synchro > Pipelines > #448

passed Pipeline #448 triggered 2 weeks ago by DERACO Stéphane Delete

Update pom.xml, ci_settings.xml, .gitlab-ci.yml files

🕒 6 jobs for `main` in 3 minutes and 16 seconds (queued for 16 seconds)

🚩 latest

🔗 75847f81

🔍 No related merge requests found.

Pipeline Needs Jobs 6 Tests 21

Build	Test	Quality	Publish	Tower
✓ build:wrapper	✓ test:wrapper	✓ display test c...	✓ publish:wrap...	✓ tower
		✓ sonarqube:w...		

 passed	#395 latest		 master → 0667beb7 Merge branch 'develop' into...		 00:09:55  2 weeks ago
 passed	#394 latest detached		 1 → 4ecbdcdbf Déduplication		 00:08:03  2 weeks ago
 passed	#393 detached		 1 → e26eae7f Ajout env		 00:07:22  2 weeks ago

• Stockage d'artéfacts : Gitlab

Package Registry

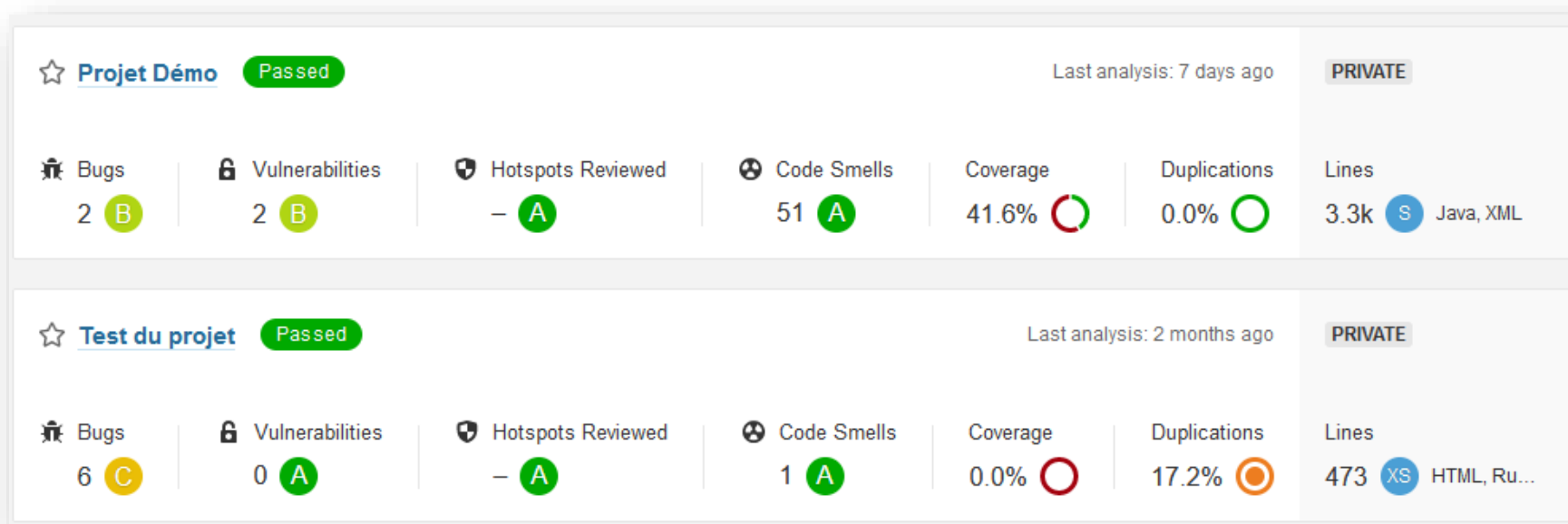
📦 2 Packages

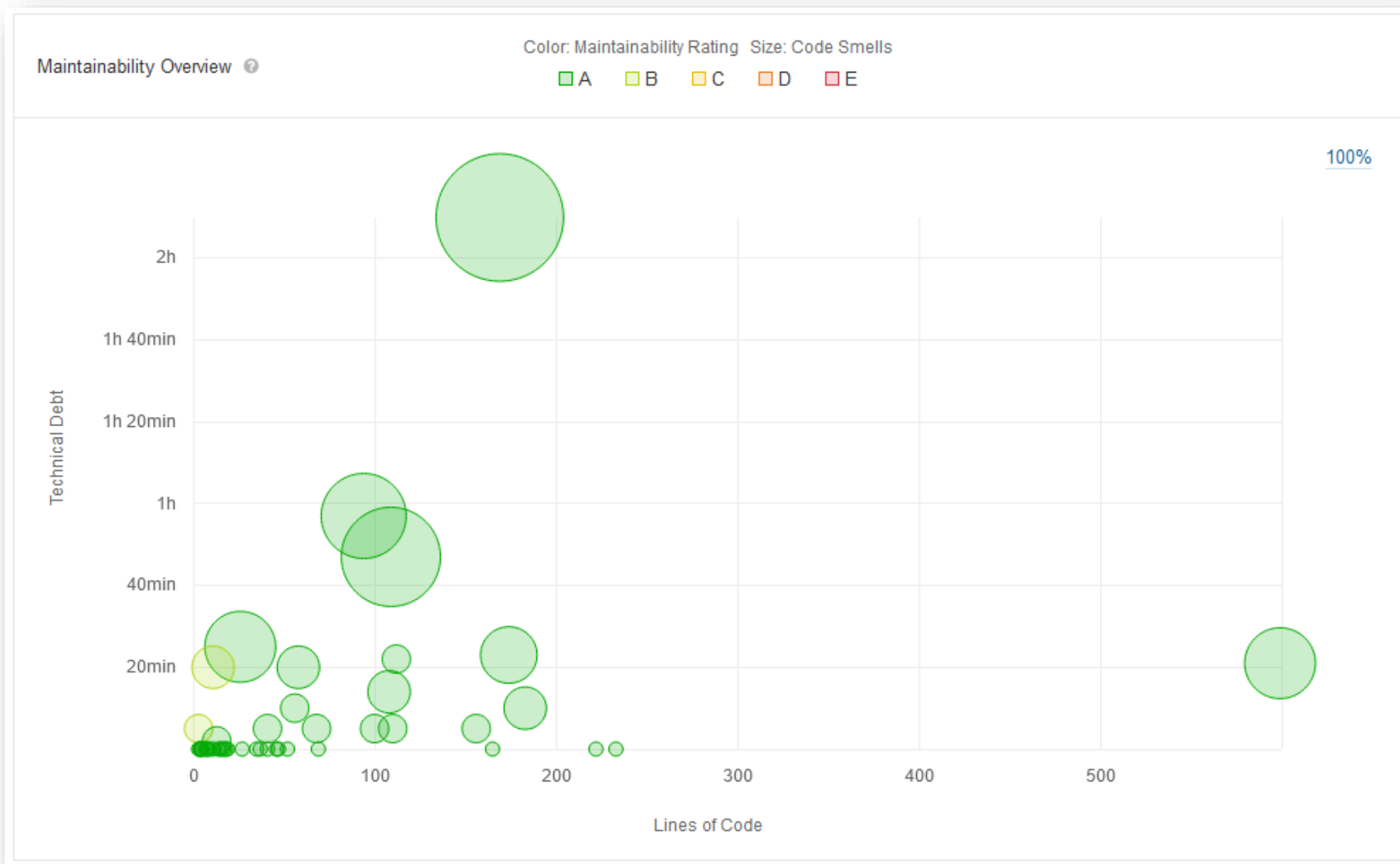
Publish and share packages for a variety of common package managers. [More information](#)

Published

fr/cnrs/dsi/koda/koda-synchro 1.0.1 published by DERACO Stéphane Maven	main 9a5894d9 Created 4 months ago
fr/cnrs/dsi/koda/koda-synchro 0.0.2 published by DERACO Stéphane Maven	develop be575ccf Created 8 months ago

• Analyse statique de code : Sonarqube





🛡️ 19 Security Hotspots to review

Review priority: **HIGH**

Authentication 4 ▼

SQL Injection 8 ▲

Make sure using a dynamically formatted SQL query is safe here.
TO REVIEW

Make sure using a dynamically formatted SQL query is safe here.
TO REVIEW

Make sure using a dynamically formatted SQL query is safe here.
TO REVIEW

Make sure using a dynamically formatted SQL query is safe here.
TO REVIEW

Make sure using a dynamically formatted SQL query is safe here. Add Comment Open in

Category: SQL Injection

Review priority: **HIGH**

Assignee: Not assigned

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk. ▼

src/.../java/org/cysecurity/cspf/jm/controller/EmailCheck.java

```
43         JSONObject json=new JSONObject();
44         if(con!=null && !con.isClosed())
45         {
46             ResultSet rs=null;
47             Statement stmt = con.createStatement();
48             rs=stmt.executeQuery("select * from users where email='"+email+"'");
49             if (rs.next())
50             {
51                 json.put("available", "1");
52             }
53             else
```

Démarche DSI

- **Beaucoup de demandes d'utilisation de la forge DSI par des personnels en laboratoire**
 - Les aspects techniques et organisationnels ne permettaient pas d'y répondre favorablement jusqu'à présent
 - Mais le besoin n'a pas été oublié
- **Entretiens avec des labos entre fin 2019 et mi 2021 pour identifier au mieux les besoins**
 - Confirmation du besoin pour une forge « institutionnelle » CNRS

Fonctionnalités / outils prévues pour la Forge labos

2

Gestion du code source

- **Gitlab**

- Version Community Edition
- Logiciel connu et reconnu
- Beaucoup de fonctionnalités
- Git over HTTPS ✓
- Git over SSH : à l'étude ?
- Wiki ✓
- Issues ✓

Intégration continue

• Gitlab CI

- Jenkins était une piste, mais gestion des droits plus compliquée
- Des runners à disposition (RHEL + docker)
 - Toujours possible de monter son propre runner
 - Restrictions sur les flux de sortie (en réflexion) :
 - Via proxy squid avec allow-list
 - Via un gestionnaire d'artéfacts dédiés = Artifactory
 - Déjà ouvert pour les outils les plus communs : Maven Central, Pypi, Packagist, Npm, Github, Gitlab, Docker Hub, ... (plus d'autres à définir comme CPAN, CRAN, ...)
 - → Ticket e-dem à faire si nouveau besoin (validation par équipe sécurité)
- Images Docker : allow-list également (éviter des images potentiellement malicieuses)

Stockage des artéfacts

- **A priori : Artifactory**

- On peut stocker des artéfacts dans Gitlab

- + : gestion des droits liée à celle du projet Gitlab
 - - : dans Gitlab CE, on n'a pas toutes les fonctionnalités

- Licence Artifactory Pro (1 serveur)

- Artifactory OSS gère uniquement Maven, Gradle et Ivy
 - Artifactory Pro gère en plus Bower, Chef, Cocoa, Conan, Debian, Docker, Git LFS, NPM, Nuget, PHP Composer, Puppet, Pypi, RPM, Rubygems, Vagrant, ...

- **Problématique sur le volume**

Analyse de la qualité du code

- **Sonarqube**

- Version Community Edition
- Suivi d'une branche du projet (recommandé)
- Analyse statique du code pour différents langages
 - Bonnes pratiques
 - « code smell » (variables non utilisées, condition du if toujours true, ...)
 - Points bloquants/critiques
 - Problèmes de sécurité (SQL Injection, Command Injection, mots de passe en clair, ...)
 - Suivi de l'évolution de ces critères
- Possibilité de mettre en place une « Quality Gateway » pour s'assurer que le nouveau code commité respecte les critères de qualité (et ne dégrade pas la qualité du projet)

- **Depuis Gitlab CI**

- Possibilité d'appeler Sonar et d'attendre la Quality Gateway pour continuer le build ou non

Utilisateurs identifiés

Qui aura accès

- **Entretiens avec des labos**

- Des besoins différents et parfois antagonistes
 - Forge la plus ouverte possible
 - vs projets sensibles et confidentiels

- **Décision de s'orienter vers 2 forges**

	Forge « ouverte »	Forge « sécurisée »
Population	Janus (+ UHPI et ext)	Janus
Authentification	Janus	MFA (clé Yubikey)
Visibilité des projets	Interne par défaut	Privé par défaut
Visibilité max. possible	Publique	Interne
Runners	Dédiés	Dédiés (distincts des autres)
Mises à jour (OS, composants)	Standard	ASAP

3

MFA = Multi-Factor Authentication

- **Objectif : renforcer l'authentification**
- **Clé FIDO, type Yubikey**



⚠ Modèles de clés autorisés : à l'étude

- Login = « ce que je suis »
- Mot de passe = « ce que je sais »
- **Clé = « ce que je possède »**

Autres fonctionnalités

Fonctionnalités supplémentaires

- **Gitlab**

- Wiki
- Issues / milestones
- Étude de faisabilité
 - mattermost intégré à Gitlab
 - Gitlab Pages

- **Sécurité**

- A l'étude !
- OWASP Dependency-Track (<https://dependencytrack.org/>)
 - Analyse des dépendances obsolètes / vulnérables
 - Proposition de mise à jour dans une merge-request

Organisation

Technique

- **Serveurs hébergés à l'IN2P3**
- **Gitlab**
 - 2 instances de Gitlab-Rails
 - base PG dédiée
 - stockage git dédié (Gitaly)
 - serveur Redis dédié

5

Humaine

- **Opéré par la DSI**
- **Taux de Disponibilité, support**
 - Mode Best Effort
- **Demandes d'accès, création de comptes, ...**
 - Automatique à l'authentification
- **Communauté**
 - Liste Sympa : koda@services.cnrs.fr

Planning

Planning prévisionnel

- **(non contractuel ;)**
- **Forge « ouverte »**
 - Fin 2021 : Accès Gitlab de base
 - T1 2022 : Ajout Sonar
 - T1 2022 : Ajout Artifactory
 - T1/T2 2022 : Si possible : mattermost
 - T2 2022 : Si possible Gitlab Pages
- **Forge « sécurisée »**
 - Nécessite de l'organisation sur l'enrôlement des clés FIDO
 - T1 2022 : Accès Gitlab de base
 - T2 2022 : Accès Sonar + Artifactory

Mot de la fin

- **Ces services seront gratuits**
- **La DSI CNRS fait son maximum pour répondre à ces besoins**
 - Mais pas d'équipe **dédiée** « Forge »
- **Si des labos souhaitent être pilotes :**
 - dsi.forge2@cnrs.fr

Merci !

Questions ?