

Rockpedia : Bien collaborer

Swagger

Swagger est la documentation de l'application. Elle est accessible en local à l'adresse <http://localhost:8080/swagger-ui.html> et en ligne : <https://groupe1.m2gi.win/swagger-ui.html>

Jenkins

Pour éviter les phrases du type "...mais ça fonctionne sur ma machine", on fait de l'intégration continue : On fait compiler le code sur une machine autre que les postes depuis le dépôt Git.

Cela permet à tous les collaborateurs de voir que les tests ne passent plus à partir de ce commit, et donc de corriger plus vite les bugs.

Lancer un build Jenkins

- Rendez vous à l'adresse <https://jenkins.m2gi.win/>
- Connectez vous. Si tout se passe bien, vous devriez arriver sur un écran comme celui-ci :

The screenshot shows the Jenkins dashboard. On the left is a sidebar with navigation links: Nouveau Item, Utilisateurs, Historique des constructions, Relations entre les builds, Vérifier les empreintes numériques, and Mes vues. The main area displays a table of jobs. The job 'g1-rockpedia-new' is highlighted, showing a success status with a sun icon. Below the table, there are sections for 'File d'attente des constructions' (empty) and 'État du lanceur de compilations' (showing 'Au repos' for two items). At the bottom, there are links for 'Légende', 'Atom feed pour tout', 'Atom feed de tous les échecs', and 'Atom feed juste pour les dernières compilations'.

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée
		g1-rockpedia-new	5 h 0 mn - #34	s. o.	1 mn 10 s

- Cliquez sur le nom du projet
- Cliquez sur lancer un build
- Vous devriez voir que quelque chose a démarré. Cliquez sur le #nombre à côté du point clignotant.
- Cliquez sur Console Output pour voir le déroulement du build

Si un build ne passe pas, il faut vérifier en local que le build s'effectue correctement : `mvn install`

SonarQube

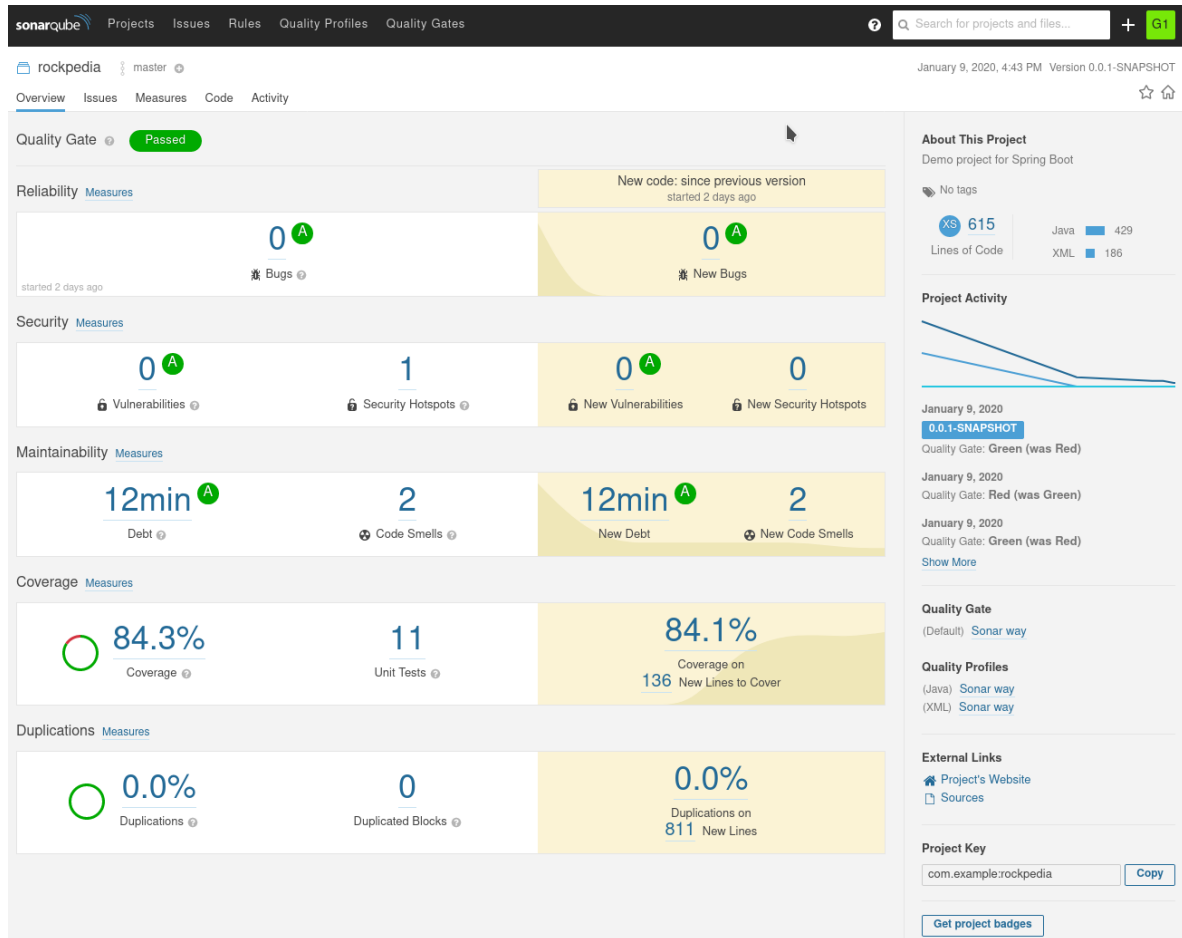
Une application bien conçue ne doit pas comporter de bugs, être bien testée (bonne couverture de code), avoir peu de redondance de code, et respecter les standards le plus possible.

SonarQube permet de surveiller tout cela, et en particulier deux choses :

- la couverture de code : c'est le pourcentage de code qui est testé
- la dette technique : c'est le temps nécessaire pour corriger les bugs, failles, redondances...

Accéder à SonarQube

- Rendez vous à l'adresse <https://sonarqube.m2gi.win/dashboard?id=com.example%3Arockpedia>
- Vous arrivez sur le tableau de bord de l'application



- On peut distinguer les bugs, les failles de sécurité, la dette technique, la couverture de code et afficher le détail de chacun.

SonarQube en local, dans l'IDE

Il existe un plugin pour Eclipse, IntelliJ, VS Code pour surveiller la qualité du code en local : SonarLint