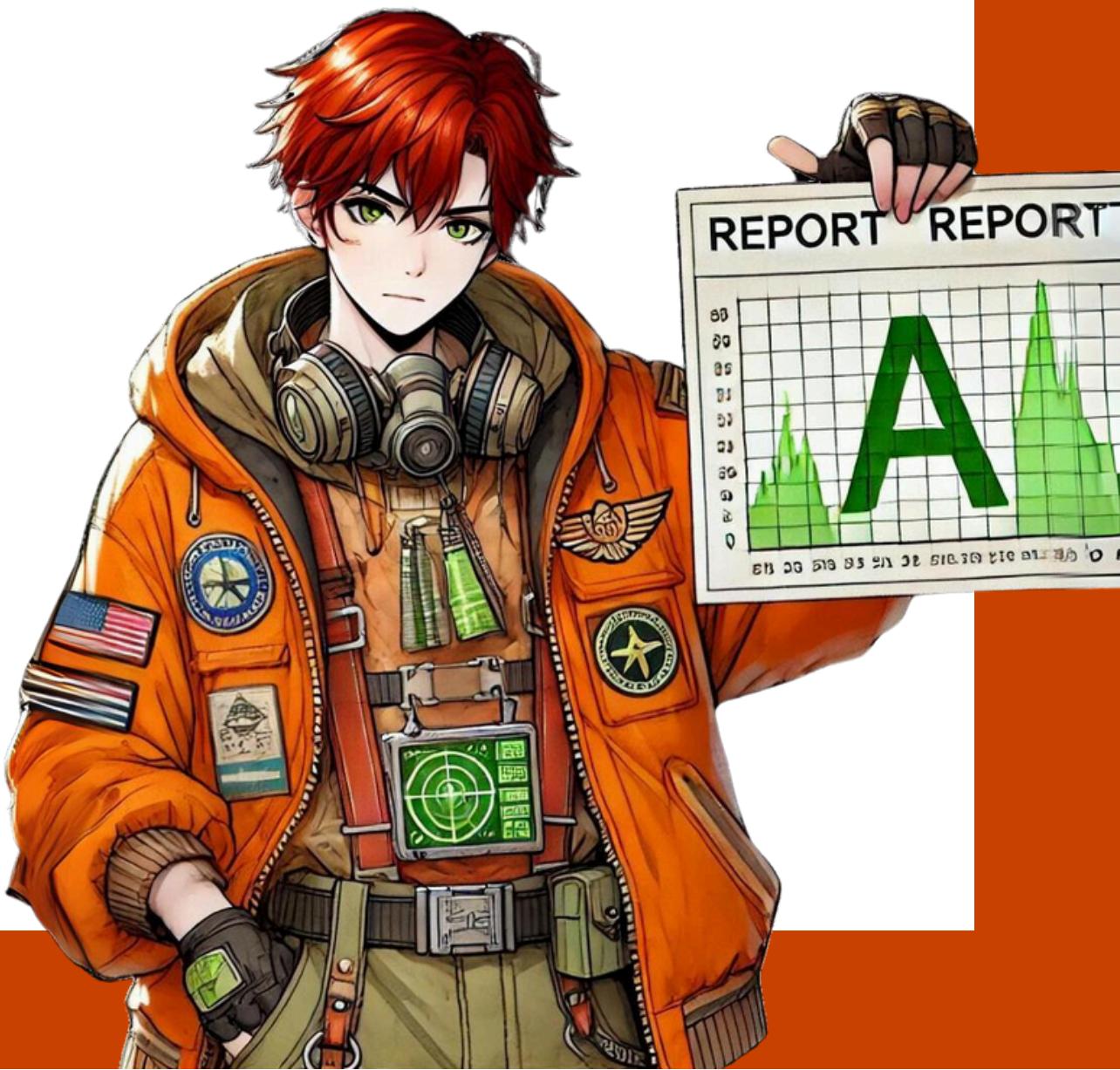


Antony Kervazo-Canut

Sonar for Teenagers

**AMÉLIOREZ VOS PROJETS EN
CONTINUE**



SOMMAIRE

Introduction	3
SonarCloud	4
Token	5
Première analyse	6
Résultat	7
Quality Profile	9
Quality Gate	10
Analyse d'une Pull Request	11
Décorer une Pull Request	12
Sonar Lint	14

Introduction



Sonar est une plateforme de qualité de code qui aide les développeurs à écrire du code propre et sûr. Il propose deux principales offres :

- **SonarQube** : Une solution auto-hébergée pour l'analyse de la qualité du code, idéale pour les entreprises souhaitant une solution sur site.
- **SonarCloud** : Une solution cloud pour l'analyse de la qualité du code, pratique pour les équipes qui préfèrent une solution entièrement gérée sans se soucier de l'infrastructure.

Différences entre SonarQube et SonarCloud

- **Déploiement** : SonarQube nécessite une installation locale, tandis que SonarCloud est une solution SaaS (Software as a Service).
- **Maintenance** : SonarQube demande des efforts de maintenance pour les mises à jour et la gestion des serveurs. SonarCloud est entièrement géré par Sonar.
- **Coût** : SonarQube peut être plus coûteux en termes d'infrastructure et de maintenance. SonarCloud propose des abonnements basés sur l'utilisation.
- **Accessibilité** : SonarCloud est accessible depuis n'importe où, tant qu'il y a une connexion Internet. SonarQube nécessite un accès au réseau où il est installé.

SonarCloud



SonarCloud est une solution SaaS (Software as a Service) pour l'analyse de la qualité du code. Elle offre une plateforme entièrement gérée, éliminant le besoin de maintenance de l'infrastructure. SonarCloud est particulièrement adapté pour les équipes de développement qui recherchent une solution rapide à mettre en place et facile à utiliser.

- **Facilité d'utilisation :** SonarCloud ne nécessite aucune installation locale. Tout ce dont vous avez besoin est un navigateur web pour accéder à la plateforme et commencer à analyser votre code.
- **Maintenance réduite :** Étant une solution SaaS, toutes les mises à jour et la maintenance de l'infrastructure sont gérées par Sonar. Cela permet aux équipes de se concentrer sur le développement plutôt que sur la gestion des serveurs.
- **Coût réduit :** SonarCloud propose des plans d'abonnement basés sur l'utilisation, ce qui peut être plus économique pour les petites équipes ou les projets en démarrage. Les coûts d'infrastructure et de maintenance sont inclus dans l'abonnement.

A screenshot of the SonarCloud web interface. At the top, there are navigation tabs: 'My Projects' (selected), 'My Issues', and 'Explore'. Below the tabs is a search bar and a user icon with a notification count of 1. On the left, there's a sidebar with 'Filters' and sections for 'Quality Gate' (Passed: 0, Failed: 1) and 'Reliability' (A: 0, B: 0, C: 1, D: 0, E: 0). The main content area shows a project named 'Avocachet / Stamping' with a status of 'NEW PRIVATE'. It includes a progress bar for the last analysis on 08/07/2024, showing 4.5k Lines of Code. Below this, there are metrics: 1 Bug, 1 Vulnerability, 20.0% Hotspots Reviewed, 197 Code Smells, 0.0% Coverage, and 1.0% Duplications. At the bottom of the main area, it says '1 of 1 shown'.

Token



La première chose à faire est de se créer un compte sur SonarCloud. Sachez par ailleurs que les projets open source peuvent analyser leurs projets gratuitement. Sinon, il faudra sortir la carte bleue.

Rendez-vous sur cette URL : <https://sonarcloud.io/account/security>

Dessus, vous pourrez générer un token utilise dans vos prochaines utilisations de SonarCloud.

Security

If you want to enforce security by not providing credentials of a real SonarCloud user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name Generate Token

✓ New token "Docker_Token" has been created. Make sure you copy it now, you won't be able to see it again!

██████████ 🔗

Dans les prochaines pages, je vais vous présenter une façon d'utilisation de sonar assez peu documenté mais beaucoup plus pratique : l'utilisation d'un Dockerfile et donc de la CLI.

Pour mes exemples, j'utiliserai un projet .Net.

Première analyse



Pour analyser un projet avec SonarCloud en utilisant la ligne de commande (CLI), nous utiliserons les commandes `begin` et `end` de Sonar Scanner. Ces commandes permettent de définir le début et la fin d'une analyse Sonar, respectivement.

- `begin` : Initialise l'analyse, configure les paramètres du projet et prépare l'environnement pour l'analyse du code.
- `end` : Finalise l'analyse, collecte les résultats et les envoie à SonarCloud. Peut échouer de force en fonction du résultat.

```
# Initialiser l'analyse SonarCloud
dotnet sonarscanner begin \
/k:<project_key> \
/o:<organization_key> \
/d:sonar.host.url="https://sonarcloud.io" \
/d:sonar.login=<your_token> \
/d:sonar.branch.name=<branch_name>

# Compilez le projet avec la configuration Release
dotnet build -c Release

# Lancez les tests en collectant la couverture de code
dotnet test --collect:"XPlat Code Coverage"

# Utilisez la commande suivante pour terminer l'analyse et
# envoyer les résultats à SonarCloud
dotnet sonarscanner end /d:sonar.login=<your_token>"
```

Résultat



Après avoir exécuté une analyse de votre projet avec SonarCloud, vous pouvez consulter les résultats sur le tableau de bord de SonarCloud. Ces résultats vous fournissent des informations détaillées sur la qualité du code et la sécurité.

- **Bugs** : Ce sont des erreurs dans le code qui peuvent causer des comportements inattendus ou des plantages. Ils sont identifiés par SonarCloud en utilisant des règles spécifiques de détection de problèmes.
- **Vulnerabilities** : Ce sont des failles de sécurité dans le code qui pourraient être exploitées par des attaquants. SonarCloud les détecte en se basant sur des règles de sécurité et des patterns de vulnérabilité connus.
- **Hotspots Reviewed** : Ce sont des zones du code qui nécessitent une révision manuelle pour évaluer leur sécurité. Ce ne sont pas nécessairement des vulnérabilités, mais des parties du code qui pourraient potentiellement poser des problèmes de sécurité.
- **Code Smells** : Ce sont des indicateurs de problèmes de maintenabilité. Ils ne provoquent pas directement des erreurs, mais rendent le code plus difficile à comprendre, à modifier ou à étendre.
- **Coverage** : La couverture de code mesure le pourcentage de code qui est testé par les tests automatisés. Une bonne couverture de code est essentielle pour s'assurer que le code fonctionne comme prévu et pour réduire les risques de bugs non détectés.
- **Duplications** : Les duplications de code indiquent combien de code est dupliqué dans le projet. Le code dupliqué peut être difficile à maintenir et à comprendre, et il est généralement préférable de le refactorer pour éliminer les duplications.

Résultat



Avocachet > Stamping > master ✓

The last analysis has warnings. See details

Summary Issues Security Hotspots Measures Code Activity

Main Branch Summary

4.5k Lines of Code ?

Take the Tour



Quality Gate ?

Passed

New Code

Overall Code

Security

7 Open issues

A

Reliability

27 Open issues

Maintainability

163 Open issues

A

Accepted Issues

0

Coverage

0.0%

Duplications

1.0%

Security Hotspots

7

No conditions set
on 1.2k Lines to coverNo conditions set
on 7.5k Lines

Filters

Clear All Filters

Clean Code Attribute

Consistency	0
Intentionality	0
Adaptability	0
Responsibility	5

Software Quality

Security	5
Reliability	5
Maintainability	0

Add to selection + click

Severity	?	(1)	x
High	5		
Medium	0		
Low	0		

 Bulk Change Select issues ▾ Navigate to issue ↺ ↻ 5 issues 25min effort

Controllers/AccountController.cs

 Responsibility

ModelState.IsValid should be checked in controller actions.

asp.net +

 Open ▾ Not assigned ▾ Reliability ⚡ Maintainability ⚡ Security ⚡ Code Smell ⚡ Major
5min effort • 16 days ago

 Responsibility

ModelState.IsValid should be checked in controller actions.

asp.net +

 Open ▾ Not assigned ▾ Reliability ⚡ Maintainability ⚡ Security ⚡ Code Smell ⚡ Major
5min effort • 16 days ago

Controllers/GoogleController.cs

 Responsibility

ModelState.IsValid should be checked in controller actions.

asp.net +

 Open ▾ Not assigned ▾ Reliability ⚡ Maintainability ⚡ Security ⚡ Code Smell ⚡ Major
5min effort • 16 days ago

Quality Profile



Le Quality Profile dans Sonar est un ensemble de règles et de configurations qui définissent comment le code sera analysé et évalué. Chaque langage de programmation peut avoir son propre Quality Profile. Utiliser un Quality Profile adapté est crucial pour obtenir des analyses pertinentes et pour assurer la qualité du code.

Pourquoi utiliser un Quality Profile ?

- **Standardisation** : Assure que tous les projets d'une organisation suivent les mêmes normes de qualité.
- **Personnalisation** : Permet d'adapter les règles de qualité en fonction des besoins spécifiques du projet ou de l'équipe.
- **Amélioration continue** : Facilite l'ajustement des règles pour améliorer continuellement la qualité du code.

Par défaut, Sonar possède quelques règles pré-écrite dans des profiles "Sonar Way". Ces profiles peuvent être copiés, exportés, modifiés. Le but étant d'avoir un quality profile qui correspond à l'équipe afin d'éviter que Sonar ne remonte des erreurs qui n'ont pas lieu d'être.

Intentionality	All branches in a conditional structure should not have exactly the same implementation	ABAP	Reliability OK	Bug	Deactivate
Intentionality	An internal table should be sorted before duplicates are deleted	ABAP	Reliability OK	Bug	Deactivate
Responsibility	Authorization checks should not rely on hardcoded user properties	ABAP	Security CRITICAL	Vulnerability	Deactivate

Exemple de quelques règles C# provenant d'un profil "Sonar Way".

Quality Gate



La Quality Gate est un ensemble de conditions que votre projet doit respecter pour être considéré comme acceptable en termes de qualité de code. Il s'agit d'un mécanisme crucial pour garantir que le code atteint un certain niveau de qualité avant d'être intégré à la base de code principale ou déployé en production.

Il est important de définir sa QualityGate en fonction de vos projets. Il est peut être déroutant ou mal avisé de mettre une Quality Gate trop forte et punitive sur un projet trop legacy.

Sonar way DEFAULT BUILT-IN

[Copy](#)

The only quality gate you need to practice [Clean as You Code](#)

Conditions

Your new code will be clean if: ?

No new bugs are introduced	Reliability rating is A
No new vulnerabilities are introduced	Security rating is A
New code has limited technical debt	Maintainability rating is A
All new security hotspots are reviewed	
New code is sufficiently covered by test	Coverage is greater than or equal to 80.0% ?
New code has limited duplication	Duplicated Lines (%) is less than or equal to 3.0% ?

These conditions apply to the new code of all branches and to pull requests.

Projects ?

Every project not specifically associated to a Quality Gate will be associated to this one by default.

Exemple de la Quality Gate par défaut de SonarCloud.

Analyse d'une Pull Request



L'analyse des Pull Requests (PR) avec SonarCloud permet de vérifier la qualité du code avant de fusionner les modifications dans la branche principale. Cela aide à détecter les bugs, les vulnérabilités et les problèmes de qualité à un stade précoce, garantissant ainsi que seuls les changements de haute qualité sont intégrés.

```
dotnet sonarscanner begin \
/k:"$SONAR_PROJECT_KEY" \
/o:"$SONAR_ORGANIZATION_KEY" \
/d:sonar.host.url="$SONAR_HOST_URL" \
/d:sonar.token="$SONAR_TOKEN" \
/d:sonar.pullrequest.key="$SONAR_PR_KEY" \
/d:sonar.pullrequest.branch="$PR_BRANCH" \
/d:sonar.pullrequest.base="$TARGET_BRANCH" \
/d:sonar.qualitygate.wait=true
```

PR Summary

42 New Lines · [feature/sonarcloud_fix](#) → [master](#)

Quality Gate [?](#) **Passed** Last analysis 22 hours ago · [410fe48a](#)

New Issues 0 No conditions set	Accepted Issues 0 Valid issues that were not fixed	
Coverage A few extra steps are needed for SonarCloud to analyze your code coverage. Set up coverage analysis	Duplications 0.0% Required ≤ 3.0% on 42 New Lines 1.0% Estimated after merge	Security Hotspots 0 No conditions set

Décorer une Pull Request



La décoration des Pull Requests ajoute des commentaires et des informations sur la qualité du code directement dans la PR. Cela fournit aux développeurs un feedback immédiat et contextuel sur les modifications proposées. La condition "qualitygate.wait" permettra à votre pipeline d'échouer en cas de non-respect de la quality gate.

```
dotnet sonarscanner begin \
  /k:"$SONAR_PROJECT_KEY" \
  /o:"$SONAR_ORGANIZATION_KEY" \
  /d:sonar.host.url="$SONAR_HOST_URL" \
  /d:sonar.token="$SONAR_TOKEN" \
  /d:sonar.pullrequest.key="$SONAR_PR_KEY" \
  /d:sonar.pullrequest.branch="$PR_BRANCH" \
  /d:sonar.pullrequest.base="$TARGET_BRANCH"
  /d:sonar.qualitygate.wait=true
/d:sonar.pullrequest.provider=AzureDevOps \
/d:sonar.pullrequest.vsts.instanceUrl=<azure_devops_url> \
/d:sonar.pullrequest.vsts.project=<azure_devops_project> \
/d:sonar.pullrequest.vsts.repository=
<azure_devops_repository>
```

Décorer une Pull Request



Il faudra également demander unitairement à chaque projet dans Sonar de faire des commentaires sur les Pull Requests en allant dans les paramètres du projet.

General Settings

Edit project settings.

Find in settings

Analysis Scope	General
External Analyzers	Select the provider to be used.
General	Provider Azure DevOps Services Key: sonar.pullrequest.provider Default: <no value> (SonarCloud's default)
Integration	Integration with Azure DevOps Services To activate pull request decoration on Azure DevOps Services, specify a personal access token.
JaCoCo	
Languages	
Pull Requests	Personal access token Token of the user that will be used to decorate the pull requests. Need authorized scope: 'Code (read and write)'. Key: sonar.pullrequest.vsts.token.secured Default: <password> (SonarCloud's default)
Repository binding	
SCM	

Integration with GitHub
You may need to specify additional parameters to enable pull request analysis. See the documentation for more information.

Enable summary comment
Enable the summary comment in the Conversation tab when decorating Pull Requests
Key: sonar.pullrequest.github.summary_comment
Default: true (SonarCloud's default)

AccountController.cs

View original diff 8m ago

```

285 313     private async Task<IActionResult> Login(string password, Client client, Func<Task<IActionResult>> action)
286 314 +     private async Task<IActionResult> Login(string password, Client client)
287 315     {
288 316         try
289 317         {
290 318             if (BCrypt.Net.BCrypt.Verify(password, client.HashedPassword))
291 319             {

```

Antony Kervazo-Canut 8m ago

Stamping

Cannot convert null literal to non-nullable reference type. (external_>roslyn:CS8625 ⓘ)
See it in SonarCloud ⓘ

Write a reply...

Sonar Lint



SonarLint est un outil qui aide les développeurs à détecter et corriger les problèmes de qualité de code directement dans leur environnement de développement (IDE). Il complète SonarQube et SonarCloud en offrant un retour immédiat sur les erreurs de codage, les vulnérabilités et les mauvaises pratiques, avant même que le code ne soit commité.

Pourquoi utiliser SonarLint ?

- **Feedback immédiat** : Détecte les problèmes de qualité de code au moment de l'écriture.
- **Intégration transparente** : Fonctionne avec les IDE les plus populaires comme IntelliJ IDEA, Eclipse, Visual Studio, et Visual Studio Code.
- **Complémentarité** : Complète les analyses plus approfondies de SonarQube et SonarCloud.

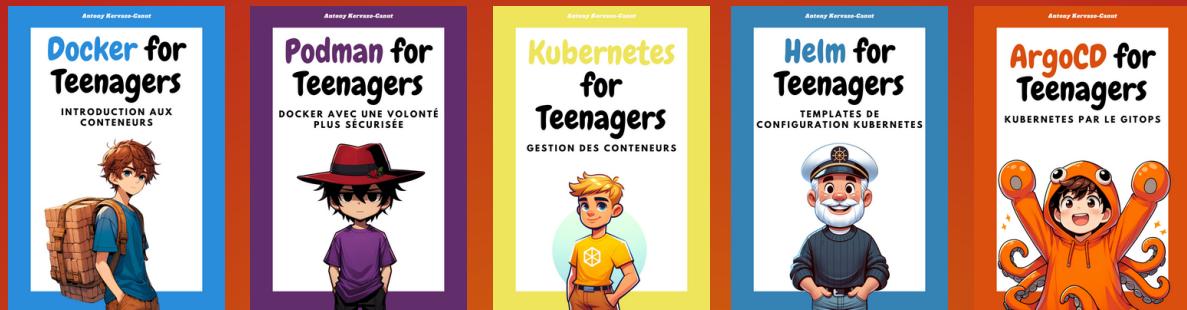
Enfin l'outil n'entraîne pas de surcoût supplémentaire et s'implémente très facilement sur vos projets.

The screenshot shows the SonarLint interface with the following details:

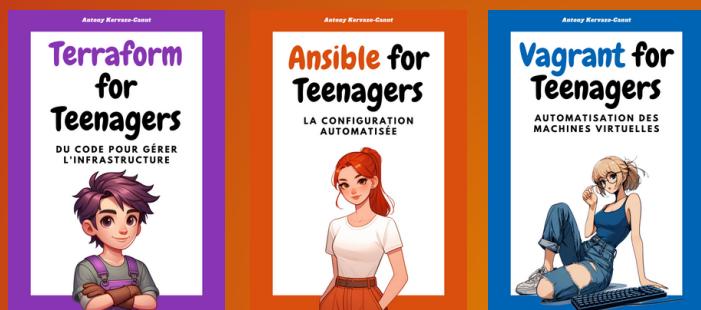
- Menu Bar:** SonarLint, Current File, Report, Security Hotspots, Taint Vulnerabilities, Log.
- Issue List:**
 - Found 2 issues in 1 file
 - c# GoogleController.cs (2 issues)
 - (14, 13) Specify the RouteAttribute when an HttpMethodAttribute or RouteAttribute is specified
 - (63, 37) ModelState.IsValid should be checked in controller actions. 7 months ago
- Icons:** A vertical column of icons including a magnifying glass, a checkmark, a star, and a close button.
- Bottom Status:** Automatic analysis is enabled, What's in this view, and a help icon.

DevOps Collection

Orchestration et Gestion de Conteneurs



Infrastructure as Code



Sécurité & Gestion des secrets



Forges, Packaging & Cloud



↓ FOLLOW ME ↓



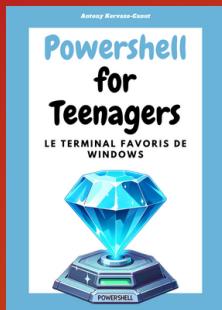
[ANTONYCANUT](#)



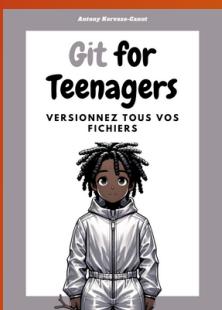
[ANTONY KERVAZO-CANUT](#)

Development Collection

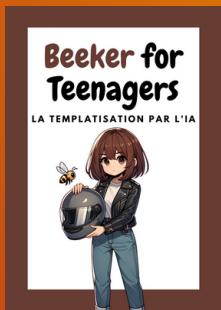
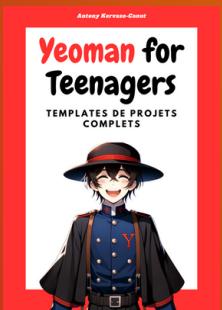
Scripting



Tools



Scaffolding



↓ FOLLOW ME ↓



[ANTONYCANUT](#)



[ANTONY KERVAZO-CANUT](#)