

# Introduction à GitHub



Médoune Siby Georges  
Baldé

Ahmadou Bamba  
Diagne



**Médoune Siby  
Georges Baldé**

Software Engineer at EYONE Medical

Partner Manager Galsen Dev

Consultant IT **& Super Hero**



**Ahmadou Bamba  
Diagne**

Full Stack Developer || Cloud & DevOps

enthusiast || AWS Certified Cloud

**Practitioner**



# Initiation à GitHub



# Introduction

Qui ici a déjà perdu un fichier important en écrasant une version précédente ?

Bonne nouvelle : GitHub est là pour sauver vos fichiers et éviter les galères !



# Introduction

GitHub est une plateforme de gestion de version décentralisée basée sur Git, permettant de collaborer facilement sur des projets de développement logiciel.

Elle héberge des dépôts **Git** et offre diverses fonctionnalités comme la gestion des branches, des pull requests, et des issues pour coordonner les efforts entre développeurs



# Git en 3 points



## 01 Définition

C'est un système de gestion de versions qui permet de suivre l'évolution des fichiers.

## 02 Tracabilité

Il permet de connaître l'intégralité de la chronologie et des modifications (ajout, suppression, modification) qui lui ont été apportées.

## 03 Décentralisation

Il facilite la collaboration, car plusieurs personnes peuvent travailler sur le même projet sans tout casser.





## Concepts clé de Git

- **Dépôt** : Base de données qui stocke toutes les versions et l'historique d'un projet. Il contient plusieurs branches.
- **Historique** : Liste des modifications du code source, enregistrées par Git.
- **Branches** : Espaces de travail isolés pour développer des fonctionnalités ou correctifs sans impacter la branche principale.
- **Collaboration** : Git permet à plusieurs développeurs de travailler simultanément via des branches distinctes.
- **Commit** : Instantané des modifications d'un projet avec un message descriptif.



## Bonnes pratiques de Git

- ❖ **Pour ajouter une fonctionnalité, créez une branche dédiée et développez dessus.**
- ❖ **Rédigez des messages de commit clairs (voir Conventional Commits).**
- ❖ **Ne poussez jamais de code directement sur la branche principale.**
- ❖ **Excluez du dépôt les fichiers inutiles ou sensibles (dépendances, fichiers .env, etc.).**





## Exemple de bon Message de Commit

- feat(auth): ajout du middleware JWT pour l'authentification
- fix(api): correction de l'erreur "utilisateur non trouvé" lors de la connexion
- refactor(bdd): optimisation de la requête de recherche des utilisateurs
- docs(readme): mise à jour des instructions d'installation
- chore(deps): mise à jour de Java vers la version 24



# Les commandes de base de **Git**

git init

git clone

git add

git pull

git commit

git status

git remote

git log

git push

git checkout



## Alors GitHub c'est quoi ?

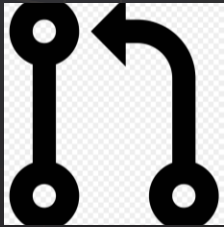
**"GitHub, c'est comme un Google Drive, mais pour les techs !"**

GitHub est une plateforme cloud qui permet de stocker,  
partager et collaborer à plusieurs sur des Projets, en ligne.

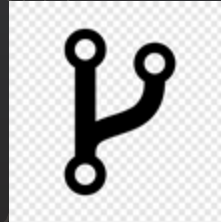




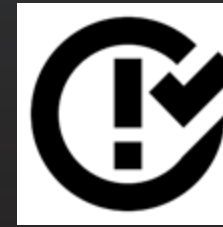
# Fonctionnalités de Github



Pull Request



Fork



Issue



`git init`

Crée un nouveau dépôt Git (un dossier suivi par Git)  
Exemple : Créer un projet "MonSuperProjet" et l'initialiser avec Git.

`git add`

Ajouter et sauvegarder un fichier  
`git add nom_du_fichier` → Ajoute le fichier à la "zone de préparation"

`git commit`

Enregistre une version  
`git commit -m "Ajout de mon premier fichier"`



## git remote & push

Envoyer sur GitHub

git remote add origin URL\_DU\_REPO → Lier GitHub

git push -u origin main → Envoyer son projet

## git checkout

Permet de changer de branche ou de restaurer des fichiers.

git checkout master → pour passer à la branche master.

## git merge

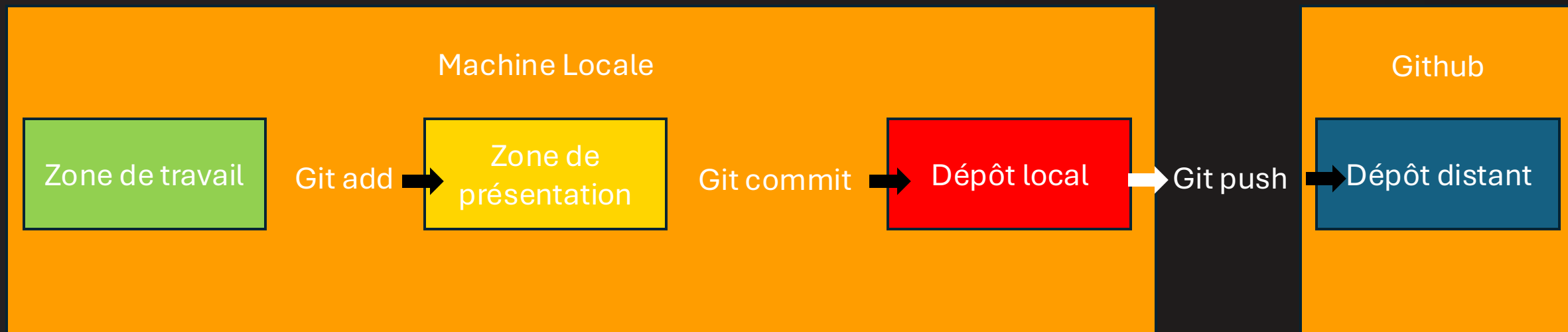
Fusionne la branche spécifiée dans la branche courante.

git merge develop





# Étapes des modifications du dépôt local au dépôt distant





# Live & Questions



Merci de votre attention