

**MANUEL D'UTILISATION****GIT****GROUPE 6:****DABO ALI****KABLAM MARIE****2024****SUPERVISEUR:**

Dr JOHNSON

## **SOMMAIRE**

1. Introduction
2. Installation de Git
3. Configuration de Git
4. Initialisation d'un Référentiel Git
5. Ajout de Fichiers au Suivi de Version
6. Validation des Changements
7. Lier le Référentiel Local avec GitHub
8. Récupération de l'Historique des Commits
9. Collaboration avec GitHub
10. Conclusion
11. Scénario complet de création d'un référentiel sur Git et GitHub, suivi de la gestion d'un projet Manuel d'Utilisation de Git et GitHub

# Introduction

Git est un système de contrôle de version distribué qui permet de suivre les changements dans les fichiers et de coordonner le travail entre plusieurs personnes. GitHub est une plateforme d'hébergement de code qui facilite la collaboration et la gestion de projets utilisant Git. Ce manuel vous guidera à travers les concepts fondamentaux de Git ainsi que son utilisation conjointe avec GitHub.

## 1. Installation de Git

Pour commencer, vous devez installer Git sur votre système. Vous pouvez le télécharger depuis le site officiel de Git et suivre les instructions d'installation spécifiques à votre système d'exploitation.

## 2. Configuration de Git

Après l'installation, configurez Git avec votre nom d'utilisateur et votre adresse e-mail en utilisant les commandes suivantes :

```
```bash
git config --global user.name "Votre Nom"
git config --global user.email "votre@email.com"
````
```

---

## 3. Initialisation d'un Référentiel Git

Avant de pouvoir utiliser Git dans un projet, vous devez initialiser un référentiel Git. Cela crée un dossier .git qui contiendra toutes les informations de suivi des versions.

```
```bash
cd chemin/vers/votre/projet
git init
````
```

## 4. Ajout de Fichiers au Suivi de Version

Une fois le référentiel initialisé, vous pouvez ajouter des fichiers à suivre en utilisant la commande `git add`.

```
```bash
```

```
git add nom_du_fichier  
````
```

## 5. *Validation des Changements*

Après avoir ajouté des fichiers, vous devez valider les changements en créant un "commit". Cela enregistre une version des fichiers dans l'historique du référentiel.

```
```bash  
git commit -m "Message de validation des changements"  
````
```

## 6. *Lier le Référentiel Local avec GitHub*

Sur GitHub :

- Créez un nouveau référentiel en cliquant sur le bouton "New repository".
- Donnez un nom à votre référentiel, ajoutez une description, et cliquez sur "Create repository".

Sur votre terminal :

```
```bash  
git remote add origin url_du_referentiel_github  
git push -u origin master  
````
```

## 7. *Récupération de l'Historique des Commits*

Vous pouvez afficher l'historique des commits à l'aide de la commande `git log`.

```
```bash  
git log  
````
```

## 8. *Collaboration avec GitHub*

Sur GitHub, vous pouvez :

- Inviter d'autres utilisateurs à collaborer sur votre projet.

- Créer des problèmes pour suivre les tâches et les bugs.
- Créer des branches pour travailler sur des fonctionnalités isolées.
- Fusionner les branches avec la branche principale via des pull requests.

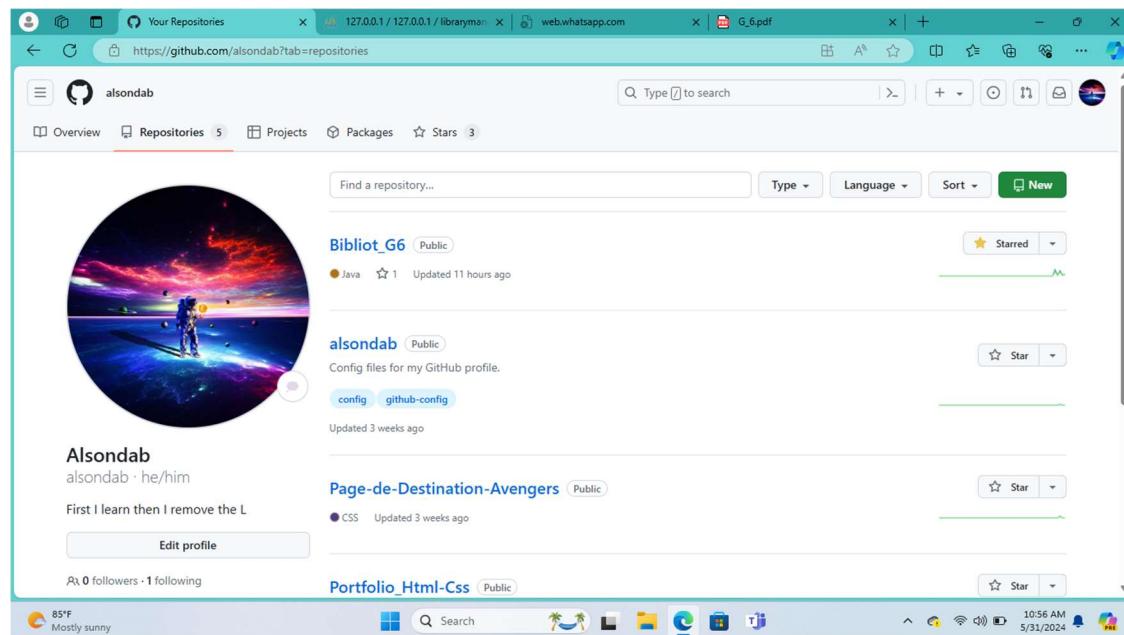
## **scénario complet de création d'un référentiel sur Git et GitHub, suivi de la gestion d'un projet**

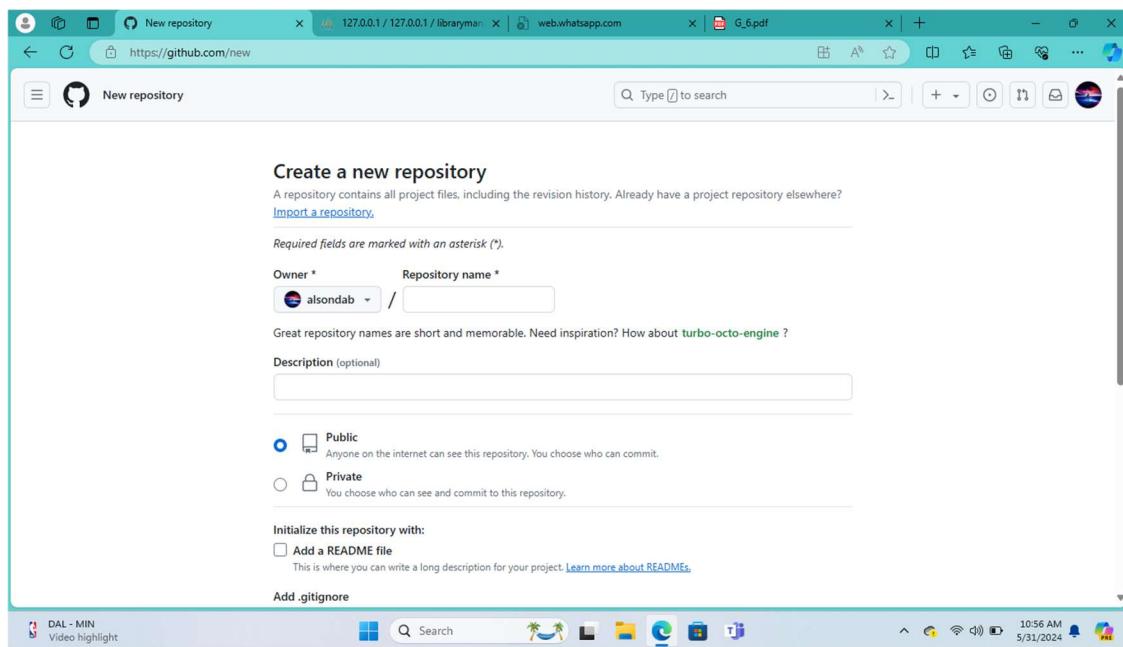
### *1. Crédit d'un Référentiel Local sur Git :*

- Ouvrez votre terminal et accédez au répertoire de votre projet en utilisant la commande `cd chemin/vers/votre/projet` .
- Initialisez un référentiel Git en exécutant la commande `git init` .
- Ajoutez vos fichiers au suivi de version en utilisant la commande `git add .` pour inclure tous les fichiers, ou `git add nom\_du\_fichier` pour des fichiers spécifiques.
- Validez vos changements en créant un commit avec la commande `git commit -m "Message de validation des changements"` .

### *2. Crédit d'un Référentiel sur GitHub :*

- Allez sur le site GitHub et connectez-vous à votre compte.
- Cliquez sur le bouton "New" pour créer un nouveau référentiel.





c. Donnez un nom à votre référentiel, ajoutez une description facultative, choisissez si vous souhaitez le rendre public ou privé, et cliquez sur "Create Repository".

### 3. Lier le Référentiel Local avec GitHub :

- Copiez l'URL du référentiel que vous venez de créer sur GitHub.
- Dans votre terminal par exemple vs Code, utiliser les commandes

```
`git init`  
`git remote add origin`  
`git add .`  
`git commit -m " message"`  
`git push -u origin master`.  
`git push`
```

### 4. Gestion du Projet :

- Travailler sur vos fichiers localement, en ajoutant, modifiant et supprimant des fichiers selon les besoins du projet.
- À chaque fois que vous effectuez des modifications significatives, ajoutez-les au suivi de version avec `git add` et validez-les avec `git commit`.
- Pour mettre à jour votre référentiel distant sur GitHub avec les dernières modifications, utilisez la commande `git push origin master`.

## *5. Collaboration :*

- a. Si vous travaillez en équipe, vos collaborateurs peuvent cloner le référentiel depuis GitHub en utilisant la commande `git clone url\_du\_referentiel`.
- b. Ils peuvent créer leurs propres branches pour travailler sur des fonctionnalités spécifiques en utilisant `git checkout -b nom\_de\_la\_branche`.
- c. Une fois leurs modifications terminées, ils peuvent les fusionner avec la branche principale à l'aide de `git merge` et envoyer ces changements vers le référentiel distant avec `git push`.

## *Conclusion*

Ce manuel couvre les bases de l'utilisation de Git ainsi que son intégration avec GitHub. Pour une compréhension plus approfondie, consultez la documentation officielle de Git et de GitHub, et explorez les nombreuses fonctionnalités avancées qu'ils offrent.

En suivant ces étapes, vous pouvez créer un référentiel Git local, le lier à un référentiel sur GitHub, gérer votre projet, et collaborer efficacement avec d'autres développeurs.