

# Guide d'utilisation de Git et GitHub

Ammar Djebabla

October 16, 2024

## 1 Introduction à Git et GitHub

Git est un système de contrôle de version distribué utilisé pour suivre les modifications apportées à des fichiers. Il permet à plusieurs personnes de collaborer simultanément sur un projet. **GitHub** est une plateforme qui permet d'héberger des dépôts Git en ligne, facilitant ainsi le partage et la collaboration.

Dans ce guide, nous allons voir comment cloner un dépôt, ajouter et valider des modifications, puis pousser ces modifications sur GitHub.

## 2 Étapes pour pusher votre code sur GitHub

### 2.1 1. Cloner un dépôt avec `git clone`

La commande `git clone` permet de créer une copie complète d'un dépôt Git hébergé à distance sur votre machine locale. Cette copie contient l'historique des modifications, ce qui vous permet de travailler hors ligne.

#### 2.1.1 Commande:

```
git clone https://github.com/nom-utilisateur/nom-depot.git
```

#### 2.1.2 Explication :

- `https://github.com/nom-utilisateur/nom-depot.git` est l'URL du dépôt distant.
- Une fois cloné, vous pouvez naviguer dans le répertoire du projet avec `cd nom-depot`.

### 2.2 2. Ajouter des modifications avec `git add`

Après avoir effectué des modifications sur les fichiers du projet, vous devez les ajouter à l'index (ou zone de staging) avant de pouvoir les valider.

### 2.2.1 Commande:

```
git add .
```

### 2.2.2 Explication :

- `git add .` ajoute tous les fichiers modifiés ou créés dans l'index. Vous pouvez aussi ajouter des fichiers spécifiques en remplaçant `.` par le nom du fichier.

## 2.3 3. Valider les modifications avec `git commit`

Une fois les fichiers ajoutés à l'index, vous devez valider (commit) les modifications pour créer un snapshot de l'état actuel du projet.

### 2.3.1 Commande:

```
git commit -m "Message de validation"
```

### 2.3.2 Explication :

- `-m "Message de validation"` permet d'ajouter un message expliquant les modifications apportées, pour garder une trace claire de l'historique des changements.

## 2.4 4. Pousser les modifications avec `git push`

Après avoir validé les modifications localement, vous devez les pousser sur le dépôt distant (GitHub) pour partager vos changements avec d'autres collaborateurs.

### 2.4.1 Commande:

```
git push origin main
```

### 2.4.2 Explication :

- `origin` est le nom par défaut du dépôt distant.
- `main` est la branche principale du projet. Selon la configuration, il est possible que vous travailliez sur une branche appelée `master`.

### 3 Branche main et gestion des branches

La branche `main` (ou `master`) est la branche principale dans un dépôt Git. Elle contient le code prêt à être utilisé ou déployé. Voici quelques commandes utiles pour travailler avec des branches :

- Créer une nouvelle branche :  
`git branch nom-de-branche`
- Passer à une autre branche :  
`git checkout nom-de-branche`
- Fusionner une branche dans `main` :  
`git merge nom-de-branche`

### 4 Conclusion

En suivant ces étapes, vous pouvez cloner un dépôt, ajouter et valider des modifications, puis les pousser sur GitHub. Git et GitHub offrent des fonctionnalités puissantes pour collaborer efficacement sur des projets de développement. N'oubliez pas de toujours fournir des messages de validation clairs pour garder un historique compréhensible.