

**Université Cadi Ayyad**  
**Faculté des sciences et techniques**

**Manuel d'utilisation  
de Git & Github**

**V1.0**

**PROJET:**

**Plateforme d'identification, valorisation et formation de l'artisanat  
marocain**

Module: Conduite de projet

**ENCADRÉ PAR : M.RAKRAK Said**



**MEMBRES DE L'ENTREPRISE SECURE STEP:**

- KHOUDRAJI OUIAM
- EL ANSARI AMINA
- EL MOUMNAOUI KAOUTAR
- ELORF LAHCEN

## PROJET:

### Plateforme d'identification, valorisation et formation de l'artisanat marocain

VERSION	DATE	MODIFICATIONS
1.0	le 12/03/2023	

## SOMMAIRE:

1	INTRODUCTION	p.3
2	INSTALLATION :	p.4
3	CONFIGURATION :	p.5
4	LES COMMANDES DE BASE DE GIT :	p.6
5	TRAVAILLER AVEC GITHUB	p.7
6	LA GESTION DES PULL REQUESTS	p.8
7	GITHUB ACTIONS	p.9
8	CONCLUSION	p.10

## 1-INTRODUCTION :

Git est un système de gestion de version distribué, qui permet de suivre les modifications apportées à un ensemble de fichiers au fil du temps. Github, quant à lui, est une plateforme web qui utilise Git pour fournir des fonctionnalités supplémentaires de collaboration de projet. En utilisant Git et Github, vous pouvez facilement travailler en collaboration avec d'autres développeurs sur le même code, suivre l'historique complet des modifications apportées à votre code et restaurer des versions précédentes de votre code en cas de problème. Ce manuel vous guidera à travers les étapes de l'utilisation de Git et Github pour la gestion de version et la collaboration de projet.

## 2-INSTALLATION :

### Installation de Git sur votre machine locale

1. Allez sur le site web officiel de Git (<https://git-scm.com/downloads>) pour télécharger la version appropriée de Git pour votre système d'exploitation.
2. Suivez les instructions d'installation à l'écran pour installer Git sur votre machine.

### Cloner un repository existant depuis Github

1. Ouvrez le repository Github que vous souhaitez cloner dans votre navigateur.
2. Cliquez sur le bouton "Code" vert en haut à droite de la page.
3. Copiez l'URL du repository.
4. Ouvrez un terminal sur votre machine locale.
5. Naviguez jusqu'au répertoire où vous souhaitez cloner le repository en utilisant la commande "cd".
6. Utilisez la commande "git clone" suivie de l'URL du repository pour cloner le repository sur votre machine locale. Par exemple :  
`git clone https://github.com/nom_utilisateur/nom_repository.git`
7. Une fois que le processus de clonage est terminé, vous pouvez commencer à travailler sur le code du repository sur votre machine locale.

Il est important de noter que pour cloner un repository privé depuis Github, vous devez avoir l'autorisation d'accès au repository de la part du propriétaire ou de l'administrateur du repository.

### 3-CONFIGURATION :

Pour configurer Git avec votre nom d'utilisateur, votre adresse e-mail et vos préférences de ligne de commande, suivez les étapes suivantes :

1-Ouvrez un terminal sur votre machine locale.

2-Entrez la commande suivante pour définir votre nom d'utilisateur :

```
git config --global user.name "Votre nom d'utilisateur"
```

Remplacez "Votre nom d'utilisateur" par votre nom d'utilisateur Github ou le nom d'utilisateur que vous souhaitez utiliser pour Git.

3-Entrez la commande suivante pour définir votre adresse e-mail :

```
git config --global user.email "votreadresse@domaine.com"
```

Remplacez "votreadresse@domaine.com" par votre adresse e-mail Github ou l'adresse e-mail que vous souhaitez utiliser pour Git.

4-Vous pouvez également définir d'autres préférences de ligne de commande pour Git en utilisant la commande suivante :

```
git config --global option valeur
```

Remplacez "option" par l'option que vous souhaitez définir, par exemple "color.ui" pour activer la coloration syntaxique dans la sortie de Git, et "valeur" par la valeur que vous souhaitez définir pour cette option.

Par exemple, pour activer la coloration syntaxique, vous pouvez utiliser la commande suivante :

```
git config --global color.ui true
```

5-Vous pouvez vérifier les paramètres de configuration que vous avez définis en utilisant la commande suivante :

```
git config --list
```

Cela affichera une liste de toutes les configurations définies pour votre installation Git.

## 4-LES COMMANDES DE BASE DE GIT :

### Initialiser un repository Git

1. Ouvrez un terminal sur votre machine locale.
2. Naviguez jusqu'au répertoire où vous souhaitez initialiser le repository en utilisant la commande "cd".
3. Utilisez la commande suivante pour initialiser le repository Git : `git init`

### Ajouter et committer des fichiers à un repository

1. Ajoutez les fichiers que vous souhaitez inclure dans votre commit en utilisant la commande "git add" suivie du nom du fichier. Par exemple : `git add nom_fichier.txt`
2. Utilisez la commande "git commit" suivie d'un message de commit pour committer les fichiers ajoutés : `git commit -m "Message de commit"`

### Créer, fusionner et supprimer des branches

1. Utilisez la commande "git branch" suivie du nom de la nouvelle branche pour créer une nouvelle branche : `git branch nom_branche`
2. Utilisez la commande "git checkout" suivie du nom de la branche pour basculer sur une branche existante : `git checkout nom_branche`
3. Utilisez la commande "git merge" suivie du nom de la branche que vous souhaitez fusionner pour fusionner une branche avec la branche actuelle : `git merge nom_branche`
4. Utilisez la commande "git branch -d" suivie du nom de la branche pour supprimer une branche : `git branch -d nom_branche`

### Annuler les modifications et revenir à un commit précédent

1. Utilisez la commande "git log" pour afficher l'historique des commits et trouver le commit précédent auquel vous souhaitez revenir.
2. Utilisez la commande "git checkout" suivie de l'identifiant du commit pour revenir à un commit précédent : `git checkout identifiant_commit`

### Gérer les conflits de fusion

1. Utilisez la commande "git merge" pour fusionner une branche avec la branche actuelle.
2. Si Git détecte un conflit de fusion, ouvrez le fichier en conflit et résolvez-le en modifiant le code du fichier.
3. Ajoutez les fichiers modifiés en utilisant la commande "git add" suivie du nom du fichier.
4. Utilisez la commande "git commit" pour committer les modifications de fusion avec un message de commit.

## 5-TRAVAILLER AVEC GITHUB :

Voici les étapes pour travailler avec Github :

### 1. Comment créer un repository sur Github :

- Connectez-vous à votre compte Github.
- Cliquez sur le bouton "New" en haut à gauche de la page d'accueil de Github.
- Donnez un nom à votre repository, ajoutez une description et choisissez les options de visibilité et de licence.
- Cliquez sur le bouton "Create repository".

### 2. Comment pousser des modifications locales vers un repository Github :

- Ouvrez le terminal de votre ordinateur et accédez au dossier local contenant le repository que vous voulez mettre à jour.
- Utilisez la commande "git add" pour ajouter les fichiers modifiés au commit.
- Utilisez la commande "git commit" pour créer un commit avec un message de description.
- Utilisez la commande "git push" pour pousser les modifications vers le repository distant sur Github.

### 3. Comment synchroniser votre repository local avec un repository Github :

- Ouvrez le terminal de votre ordinateur et accédez au dossier local contenant le repository que vous voulez synchroniser.
- Utilisez la commande "git pull" pour récupérer les dernières modifications du repository distant sur Github et les fusionner avec votre repository local.

### 4. Comment cloner un repository Github sur votre machine locale :

- Ouvrez le terminal de votre ordinateur et accédez au dossier dans lequel vous souhaitez cloner le repository.
- Utilisez la commande "git clone" suivie de l'URL du repository Github que vous souhaitez cloner.
- Une fois le clonage terminé, vous pouvez commencer à travailler sur le repository local comme vous le feriez pour tout autre dossier.



## 6-LA GESTION DES PULL REQUESTS (PR)

La gestion des Pull Requests (PR) est une fonctionnalité clé de Github qui permet aux membres de l'équipe de collaborer sur le code et de fusionner des modifications dans la branche principale de manière contrôlée. Voici comment gérer les PR sur Github :

### 1. Créer une PR :

- Ouvrez votre repository sur Github
- Cliquez sur le bouton "New pull request" situé en haut de la page
- Sélectionnez la branche de base (généralement la branche principale) et la branche que vous souhaitez fusionner
- Github comparera automatiquement les modifications entre les deux branches
- Ajoutez une description pour expliquer les modifications apportées
- Cliquez sur "Create pull request"

### 2. Examen de la PR :

- Les membres de l'équipe peuvent examiner la PR en apportant des commentaires, en posant des questions et en demandant des modifications si nécessaire
- Vous pouvez également ajouter des labels pour organiser les PR et les classer par type ou par priorité
- Les membres de l'équipe peuvent également vérifier les modifications en exécutant des tests et en examinant les changements de code

### 3. Fusionner une PR :

- Une fois que la PR a été examinée et approuvée, vous pouvez la fusionner en cliquant sur le bouton "Merge pull request"
- Si des conflits de fusion se produisent, vous devrez les résoudre avant de pouvoir fusionner la PR
- Vous pouvez également cocher la case "Delete branch when merged" pour supprimer automatiquement la branche fusionnée

### 4. Fermer une PR :

- Si la PR n'est pas prête à être fusionnée ou si elle est abandonnée, vous pouvez la fermer en cliquant sur le bouton "Close pull request"
- Cela indique clairement aux membres de l'équipe que la PR n'est plus active

### 5. Réviser une PR :

- Si vous souhaitez apporter des modifications à une PR existante, vous pouvez créer une nouvelle branche à partir de la PR, effectuer les modifications nécessaires et créer une nouvelle PR pour cette branche.

## 6-GITHUB ACTIONS

Github Actions est une fonctionnalité de Github qui vous permet d'automatiser des tâches dans votre workflow de développement. Les actions sont des scripts ou des programmes qui s'exécutent en réponse à des événements spécifiques, tels que la création d'une PR ou la mise à jour d'une branche. Voici comment utiliser Github Actions :

### 1. Créer un fichier de workflow :

- Ouvrez votre repository sur Github
- Créez un dossier nommé ".github/workflows"
- Créez un fichier YAML dans ce dossier pour décrire le workflow

### 2. Configurer l'événement déclencheur :

- Dans le fichier YAML, spécifiez l'événement déclencheur qui déclenchera l'exécution de l'action
- Par exemple, vous pouvez configurer l'action pour s'exécuter lors de la création d'une PR ou de la mise à jour d'une branche

### 3. Décrire les étapes de l'action :

- Dans le fichier YAML, décrivez les étapes que l'action doit suivre
- Par exemple, vous pouvez configurer l'action pour exécuter des tests, construire le projet, déployer le code, etc.

### 4. Configurer des paramètres et des secrets :

- Vous pouvez configurer des paramètres qui définissent le comportement de l'action et qui peuvent être utilisés dans les scripts
- Vous pouvez également configurer des secrets pour stocker des informations sensibles, telles que des clés d'API ou des informations d'identification

### 5. Exécuter l'action :

- Lorsque l'événement déclencheur se produit, Github exécute automatiquement l'action
- Vous pouvez surveiller le statut de l'action et consulter les journaux pour vérifier son exécution

Github Actions est très flexible et peut être utilisé pour automatiser presque toutes les tâches de développement, depuis les tests jusqu'au déploiement. Il existe également de nombreuses actions prédéfinies qui peuvent être utilisées pour simplifier la configuration de votre workflow.

## 8-CONCLUSION :

En résumé, ce manuel vous a présenté les fonctionnalités clés de Github et vous a fourni des conseils pratiques pour une utilisation efficace de la plateforme, notamment la gestion de référentiels, les branches de développement, les demandes de fusion et la gestion des problèmes et des projets.