

Fiche de Procédure Git

1. Qu'est-ce que Git ?

Git est un système de gestion de version décentralisé qui permet de suivre les modifications des fichiers d'un projet au fil du temps. Il facilite le travail collaboratif en permettant à plusieurs développeurs de travailler ensemble sans écraser les modifications des autres.

2. Configuration de Git

Avant de commencer à utiliser Git, configurez votre nom et votre adresse e-mail afin d'identifier vos commits :

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

Pour vérifier la configuration :

```
git config --list
```

Fiche de Procédure Git

3. Commandes de base

Initialiser un dépôt :

```
git init
```

Ajouter des fichiers au suivi :

```
git add <fichier> ou git add .
```

Enregistrer les modifications (commit) :

```
git commit -m "Message du commit"
```

Vérifier l'état du dépôt :

```
git status
```

Voir l'historique des commits :

```
git log
```

Pousser les modifications vers un dépôt distant :

```
git push origin <branche>
```

Récupérer les modifications depuis un dépôt distant :

```
git pull origin <branche>
```

Fiche de Procédure Git

4. Gestion des branches

Créer une nouvelle branche :

```
git branch <nom_de_branche>
```

Basculer vers une branche :

```
git checkout <nom_de_branche>
```

Créer et basculer vers une nouvelle branche :

```
git checkout -b <nom_de_branche>
```

Fusionner une branche :

Pour fusionner la branche A avec la Branche B, basculer sur la branche A et faire cette commande

```
git merge <nom_de_branche> (ou B pour l'exemple)
```

5. Utilisation d'un dépôt distant (Gitea)

Cloner un dépôt Gitea :

```
git clone <URL_du_dépôt>
```

Lier un dépôt local à un dépôt Gitea :

```
git remote add origin <URL_du_dépôt>
```

Fiche de Procédure Git

Pousser les modifications vers Gitea :

```
git push -u origin <branche>
```

La première fois, utilise **-u** pour lier la branche locale à celle de Gitea. Par la suite, tu pourras utiliser `git push`.

Récupérer les modifications depuis Gitea :

```
git pull origin <branche>
```

6. Le fichier .gitignore

Le fichier `.gitignore` permet d'exclure certains fichiers ou dossiers du suivi Git.

Exemple de contenu pour `.gitignore` :

```
# Ignorer les fichiers temporaires
```

```
*.log
```

```
*.tmp
```

```
# Ignorer les dossiers de dépendances
```

```
node_modules/
```

```
vendor/
```

7. Annuler un commit ou des modifications

- Annuler un commit tout en conservant les modifications :

```
git reset --soft HEAD~1
```

- Annuler un commit et retirer les fichiers du suivi :

```
git reset --mixed HEAD~1
```

- Annuler un commit et les modifications dans les fichiers :

```
git reset --hard HEAD~1
```