
TP INITIATION A GIT

AYEDJO Kokou Wisdom, DOMLAN Karl Jason
Akoété

September 10, 2025

Contents

Juste après l'installation	2
Créons un dépôt git en local!	2
Envoyons tout sur un dépôt distant sur github	4
Un petit dernier, pour tout intégrer	4

Juste après l'installation

- Dans vos commit, git ajoutera des informations sur qui vous êtes, c'est à dire votre nom et votre email. Il faut donc les configurer:

```
1 $ git config --global user.name "nom et prenom"
2 $ git config --global user.email email@exemple.com
```

vous pouvez les checker avec

```
1 $ git config user.name
2 $ git config user.email
```

- Ensuite vous pouvez choisir l'éditeur de texte par défaut que git lancera quand ça sera nécessaire. Sous windows on choisira notepad (bloc note) avec la commande:

```
1 $ git config --global core.editor notepad
```

- Renommer la branche par défaut `master` en `main`:

```
1 $ git config --global init.defaultBranch main
```

Cela évitera des conflits de noms avec le remote car github et gitlab nomment la branche principale `master` et non `main`.

- configurer git pour faire un rebase lorsqu'on fait un pull avec:

```
1 $ git config --global pull.rebase "true"
```

La notion de `rebase` est un peu compliquée à expliquer pour l'instant. Mais comprenez juste qu'elle permet d'avoir un historique plus "propre", même lorsque le dépôt distant a été modifié entre temps.

Voilà nous sommes maintenant prêts à commencer!

Créons un dépôt git en local!

1. Créez un dossier nommé `projet` dans votre explorateur de fichiers et copiez son chemin d'accès. Ouvrez un terminal PowerShell et faites:

```
1 $ cd chemin/du/dossier/projet
```

2. initialisez le dépôt:

```
1 $ git init
```

3. Ajouter un fichier `essai.txt` puis ouvrez le notepad (bloc note)

```
1 $ notepad essai.txt
```

Ajoutez-y du texte et enregistrez-le.

4. faire le commit initial e

```
1 $ git add essai.txt
2 $ git commit -m "commit initial"
```

5. Vérifier le statut des fichiers du dépôt et son historique. Interpréter les résultats affichés

```
1 $ git status
2 $ git logs
```

5. Modifier le fichier `essai.txt`:

```
1 $ notepad essai.txt
```

6. Vérifier et interpréter le statut:

```
1 $ git status
```

7. L'ajouter au stage puis vérifier de nouveau le statut:

```
1 $ git add .
2 $ git status
```

8. Ajouter un nouveau fichier `histoire.txt`, puis consulter à nouveau le statut du dépôt et interpréter la sortie:

```
1 $ notepad histoire.txt
2 $ git status
```

9. faire le commit final

```
1 $ git commit -m "essai.txt modifié et ajout d'une belle histoire"
```

Envoyons tout sur un dépôt distant sur github

1. Créer un repository (dépôt) sur github (sans README)
2. copier son URL https
3. venir dans le dossier de notre projet et y lier le dépôt distant en faisant:

```
1 $ git remote add origin https://github.com/compte/projet
```

4. Envoyer notre historique en prenant bien soin de faire un pull pour fusionner l'historique local et le distant:

```
1 $ git pull origin main
2 $ git push -u origin main
```

Un petit dernier, pour tout intégrer

Maintenant que nous avons les bases du versionnage avec git, vous allez faire un petit travail:

- créez un fork du repository suivant: <https://github.com/AyedjoKokouWisdom/TP.git>. Vous y trouverez un hello world basique en java
- clonez le sur votre machine en local
- modifiez le texte `Hello World` en `Hello World je suis nom prenom`
- faites un commit puis envoyez-le sur votre dépôt distant.