

2. Créer un dépôt git

Il y a deux manières de créer un *repository* git. Soit initialiser un dépôt dans un répertoire existant, soit cloner un dépôt Git déjà existant.

Commençons par créer un dossier `git-test` avec la commande `mkdir git-test`.

Déplaçons notre terminal dans le dossier `git` nouvellement créé avec la commande `cd git-test`.

Enfin, créons un fichier `test.txt` en utilisant la commande `touch test.txt`.

Initialiser un dépôt dans un répertoire existant

```
| git init
```

Pour initialiser un dépôt git de cette manière, nous devons utiliser la commande `git init`.

```
git init
Initialized empty Git repository in
/home/gabagool/Projects/git/.git/
```

`git status`

Si on crée un fichier `test.txt` dans notre dépôt nouvellement créé, nous pouvons recevoir un rapport de l'état de nos fichiers par la commande `git status`.

```
On branch master
```

```
No commits yet
```

```
Untracked files:
```

```
  (use "git add <file>..." to include in what will be  
  committed)
```

```
    test.txt
```

```
nothing added to commit but untracked files present  
(use "git add" to track)
```

Cette commande nous donne 4 informations importantes :

- Notre fichier `test.txt` est *untracked*, c'est à dire qu'il a le statut non-suivi.
- Aucun fichier n'a été validé ou `commit` et envoyé dans le *repository*
- Aucun fichier n'a été indexé ou *stages* et envoyé dans la *staging area*
- Nous sommes actuellement dans la branche `master`

| `git add`

Pour envoyer notre fichier `test.txt` dans la *staging area*, nous devons utiliser la commande `git add` avec un argument. Cela peut être le nom fichier, le nom du répertoire pour indexer tous les fichiers contenu dedans, ou un *fileglob* pour sélectionner des fichier ayant un format donné.

```
git add *.txt
```

On envoi vers la *staging area* tous les fichiers ayant l'extension `.txt`.

```
git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   test.txt
```

Nous avons bien ajouté notre fichier `test.txt` dans la *staging area*, il a maintenant le statut suivi.

| `git commit`

Pour valider notre fichier, et l'ajouter en base, on va maintenant utiliser la commande `git commit`.

```
git commit
GNU nano 6.2
/home/gabagool/Projects/git/.git/COMMIT_EDITMSG
# Please enter the commit message for your changes.
Lines starting
# with '#' will be ignored, and an empty message
aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   test.txt
#
Message de commit
Première version du projet.
```

Mettre un message est obligatoire pour effectuer le `commit`. Cela permet de documenter notre travail auprès des autres développeurs qui travaillent sur le même *repository*. Une fois le message rédigé, on peut l'enregistrer et sortir de l'interface si vous êtes sur vim avec `ESC` puis `:wq` et `ENTREE`, ou avec `nano` en tapant `Ctrl+X`, puis `Y` et `ENTREE`.

 **Message inline**

On peut écrire le message du `commit` en même temps que la commande elle-même sans passer par un écran intermédiaire avec la commande `git commit -m [message]`

Git nous donne un rapport de notre commit.

```
[master (root-commit) 7c17df9] Message de commit  
Première version du projet.  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 test.txt
```

Si nous refaisons un `git status` à ce stade, git nous donne le message suivant :

```
git status  
On branch master  
nothing to commit, working tree clean
```

Git nous explique ici que tous les fichiers du répertoire sont suivis, et qu'aucune modification n'a été soulevé qui n'est pas déjà enregistré en base.

Attention

Le `commit` se basera sur l'état du fichier au moment du `git add`. Si il y a eu des changements entre le `git add` et le `git commit`, ils ne seront pas enregistrés.

Cloner un dépôt git

On peut cloner un dépôt git existant avec la commande `git clone`.

```
cd ..  
git clone git-test git-test-2  
Cloning into 'git-test-2' ...  
done.
```

Ici, nous avons cloner notre *repository* `git-test` dans un dossier `git-test-2`.

Exercice

Dans un dossier créer un dossier "git-1"

Initialiser un dépôt git dans le dossier "git-1" nouvellement créé

Y créer un fichier texte "fichier.txt"

Effectuer un `git status` pour constater l'état de suivi de votre fichier

Ajouter ce fichier dans la *staging area*

Refaire un `git status` et constater l'évolution du suivi du fichier

Committer ce fichier avec le message suivant : "commit de mon fichier"

Effectuer un dernier `git status`