


TP-4- Jules Anduze

Tout d'abord, nous allons commencer par créer un dépôt vide en utilisant la commande `git init sandwich`.

```
MINGW64:/c/Users/JulesANDUZE
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~
$ git add burger.txt
fatal: not a git repository (or any of the parent directories): .git
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~
$ git init sandwich
Initialized empty Git repository in C:/Users/JulesANDUZE/sandwich/.git/
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~
$
```

En parallèle on crée un fichier.txt nommé « burger » avec ligne par ligne un ingrédient.

Ensuite on l'enregistre dans le dossier Sandwich :

Nom	Modifié le	Type	Taille
 Burger	20/02/2024 11:01	Document texte	1 Ko

```

OneDrive - MEWO'/
PhpstormProjects/
Recent@
sandwich/
'Saved Games'/
Searches/
SendTo@
'Sports Interactive'/
Videos/
'VirtualBox VMs'/
'Voisinage d'impression"@
'Voisinage réseau'@

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~
$ cd sandwich/

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ ls
Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$

```

J'ai commencé par la commande `ls` pour vérifier que le répertoire sandwich était bien là, si il est bien dedans je fais la commande `cd sandwich/` pour rentrer dedans, je fais a nouveau `ls` dans le dossier sandwich pour vérifier si burger.txt est bien là.

Ensuite pour ajoute le burger.txt je fais la commande `git add burger.txt`.

```

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master

No commits yet

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

        Burger.txt

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

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status

```

J'ai ensuite fait la commande `git add burger.txt` pour l'ajouter.

Ensuite pour vérifier, je fais la commande `git status`, je me rend compte en lisant l'erreur que le fichier n'est pas commit, je vois alors que j'ai oublier la majuscule a Burger.

Je retape alors la commande avec la bonne syntaxe qui a Ducoup bien fonctionner puisqu'en vert est affiché **new file : Burger.txt**

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

    Burger.txt

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

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master

No commits yet

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

    new file:   Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

```
No commits yet

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

    new file:   Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git diff --cached
diff --git a/Burger.txt b/Burger.txt
new file mode 100644
index 0000000..be4184e
--- /dev/null
+++ b/Burger.txt
@@ -0,0 +1,5 @@
+Steak
+Salade
+Tomate
+Cornichon
+Fromage

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

Je tape ensuite la commande **git diff --cached** qui nous sert a voir si des modifications sur le fichier on étai faites.

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git commit -m "modification de la recette"
[master (root-commit) 37616a4] modification de la recette
1 file changed, 5 insertions(+)
create mode 100644 Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

Je fais la commande `git commit -m « entrez texte »` pour mettre un nom et savoir la modification faites qui nous permet de garder un historique.

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

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

A l'aide de la commande `git status` nous pouvons voir qu'aucun changement n'était fait.

Avec la commande `git log`, nous pouvons voir l'auteur, l'heure et la date. Ainsi que le numéro SHA1 qui pour ma part est : `37616a414a294d90953363d1ddfc6673063d21d9`.

```
MINGW64:/c/Users/JulesANDUZE/sandwich
+Tomate
+Cornichon
+Fromage

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git commit -m "modification de la recette"
[master (root-commit) 37616a4] modification de la recette
1 file changed, 5 insertions(+)
create mode 100644 Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master
nothing to commit, working tree clean

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git log
commit 37616a414a294d90953363d1ddfc6673063d21d9 (HEAD -> master)
Author: Juliooo57 <jules.anduze@gmail.com>
Date: Tue Feb 20 12:09:04 2024 +0100

    modification de la recette

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

J'ai choisi pour ma part de créer de nouveau fichier.txt depuis git directement. Pour ce faire :

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ echo patate > hotdog.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ ls
Burger.txt  hotdog.txt
```

Nous utilisons la commande `echo (entrer_texte) > (le_nom_du_fichier)`

Pour vérifier si notre commande a bien marché on utilise la commande `ls` et on voit bien les deux fichiers

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git log
commit 809a282192aef5f3fe2e91d3b8daab269aabe2fd (HEAD -> master)
Author: Juliooo57 <jules.anduze@gmail.com>
Date: Tue Feb 20 12:35:29 2024 +0100

    une bonne raclette pour même

commit 37616a414a294d90953363d1ddfc6673063d21d9
Author: Juliooo57 <jules.anduze@gmail.com>
Date: Tue Feb 20 12:09:04 2024 +0100

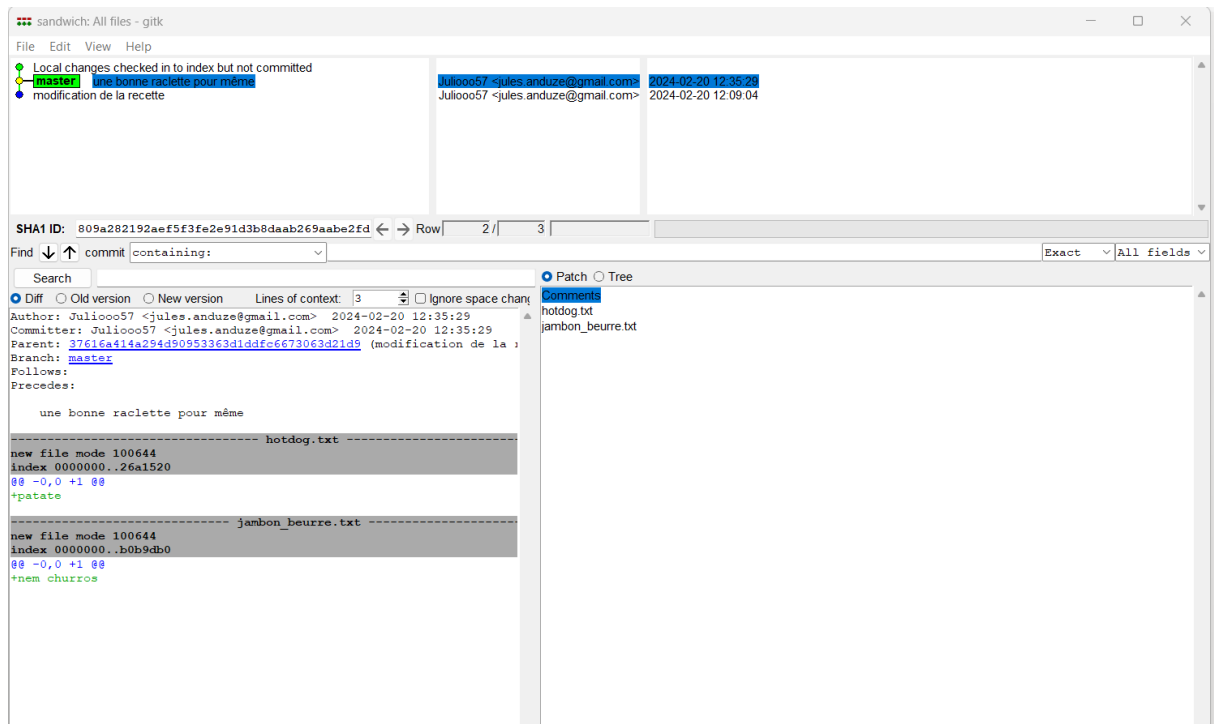
    modification de la recette

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

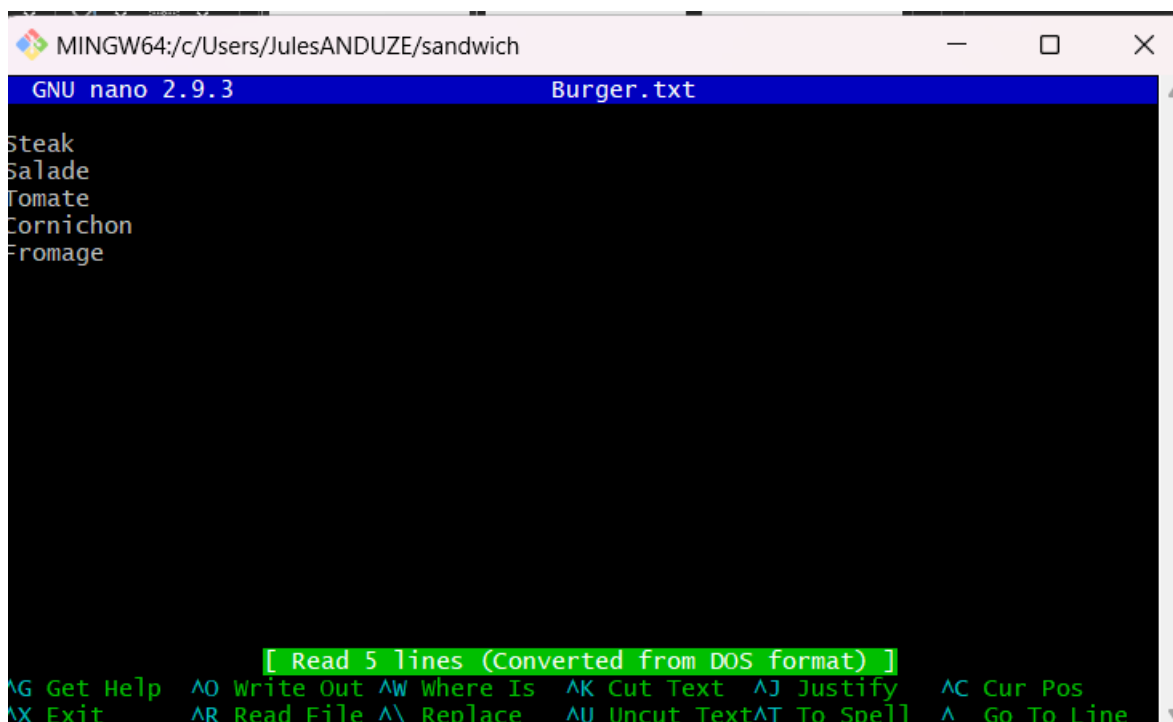
J'ai ensuite créer un nouveau fichier nommé `raclette.txt` que j'ai ensuite `git commit -m (entrez_un_texte)` puis j'ai `git add raclette.txt` j'ai ensuite fais un `git status` pour voir si mon fichier avait bien était ajouté, après la commande `git log` pour voir les modifications, on voit donc tout les commit avec message.

J'ai ensuite changé d'interface graphique avec la commande `gitk` dans cette interface nous pouvons directement voir le contenu du fichier aussi.

Tp4 – Git -Mewo – Jules Anduze



Pour modifier un fichier en ligne de code : utilisez la commande `nano (nom_du_fichier)`



Ce qui nous ramène ici : nous pouvons ensuite modifier et après enregistré la nouvelle modification.

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ nano Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git add Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   Burger.txt
        new file:   raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ .....
```

Après nous faisons un **git add** pour l'ajouter a l'index. Puis un **git status** pour vérifier.

```
        new file:   raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git reset Burger.txt
Unstaged changes after reset:
M       Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   raclette.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
```

Il faut faire la commande : **git reset** Pour revenir en arrière et renvoyez le fichier dans le working copy et annuler le git add. Une fois un **git status** fait on voit donc qu'il est rouge pour montrer qu'il est modifié.

```
MINGW64:/c:/Users/JulesANDUZE/sandwich

    new file:   raclette.txt
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git checkout Burger.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$
```

Avec la commande **git checkout**, on récupère l'ancien fichier en supprimant le nouveau, ensuite un git status est fait pour voir le résultat

Ensuite avec la commande `git checkout « le_chiffre_du_commit »` qui nous permet de revenir en arrière.

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git log
commit 809a282192aef5f3fe2e91d3b8daab269aabe2fd (HEAD -> master)
Author: Juliooo57 <jules.anduze@gmail.com>
Date: Tue Feb 20 12:35:29 2024 +0100

    une bonne raclette pour même

commit 37616a414a294d90953363d1ddfc6673063d21d9
Author: Juliooo57 <jules.anduze@gmail.com>
Date: Tue Feb 20 12:09:04 2024 +0100

    modification de la recette

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git checkout
A      raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git checkout 37616a414a294d90953363d1ddfc6673063d21d9
Note: checking out '37616a414a294d90953363d1ddfc6673063d21d9'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 37616a4 modification de la recette
A      raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((37616a4...))
$ |
```

A la suite de ça et pour vérifier si la commande a bel et bien fonctionné, un nouveau `git status` est nécessaire

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((37616a4...))
$ git status
HEAD detached at 37616a4
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((37616a4...))
$ |
```

Grâce a ça on voit bien que la commande a bien fonctionné.

Important : avec la commande `git log --all` nous pouvons voir les commits postérieurs.

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich (master)
$ git checkout head
Note: checking out 'head'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at 809a282 une bonne raclette pour même
A      raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git status
HEAD detached at head
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   raclette.txt
```

Par curiosité : j'ai voulu essayer la commande `git checkout head` pour voir ce qui se passerai

A la suite de ça j'ai fait un `git status`, ce qui me mets Head detached at head, j'ai donc fait la commande `git reset head raclette.txt`

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git log
commit 809a282192aef5f3fe2e91d3b8daab269aabe2fd (HEAD, master)
Author: Juliooo57 <jules.anduze@gmail.com>
Date:   Tue Feb 20 12:35:29 2024 +0100

    une bonne raclette pour même

commit 37616a414a294d90953363d1ddfc6673063d21d9
Author: Juliooo57 <jules.anduze@gmail.com>
Date:   Tue Feb 20 12:09:04 2024 +0100

    modification de la recette

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git reset head raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git status
HEAD detached at head
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        raclette.txt

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

A la suite de ça j'ai suivi git en faisant un `git add raclette.txt`.

```
AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git add raclette.txt
warning: LF will be replaced by CRLF in raclette.txt.
The file will have its original line endings in your working directory.

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((809a282...))
$ git commit -m "coucou"
[detached HEAD 8f24b28] coucou
1 file changed, 1 insertion(+)
create mode 100644 raclette.txt

AzureAD+JulesANDUZE@SLAM-1675 MINGW64 ~/sandwich ((8f24b28...))
$ git log
commit 8f24b287b9bfbfb616479c6ab0e013fe8fda1608 (HEAD)
Author: Juliooo57 <jules.anduze@gmail.com>
Date:   Wed Feb 21 14:23:51 2024 +0100

    coucou

commit 809a282192aef5f3fe2e91d3b8daab269aabe2fd (master)
Author: Juliooo57 <jules.anduze@gmail.com>
Date:   Tue Feb 20 12:35:29 2024 +0100

    une bonne raclette pour même

commit 37616a414a294d90953363d1ddfc6673063d21d9
Author: Juliooo57 <jules.anduze@gmail.com>
Date:   Tue Feb 20 12:09:04 2024 +0100

    modification de la recette
```

J'ai ensuite logiquement `git commit -m « entrez_un_txt »`, pour me souvenir de ce que j'ai fait.

Grâce a ces commandes j'ai donc remis le fichier raclette.txt dans le head, comme au paravant.