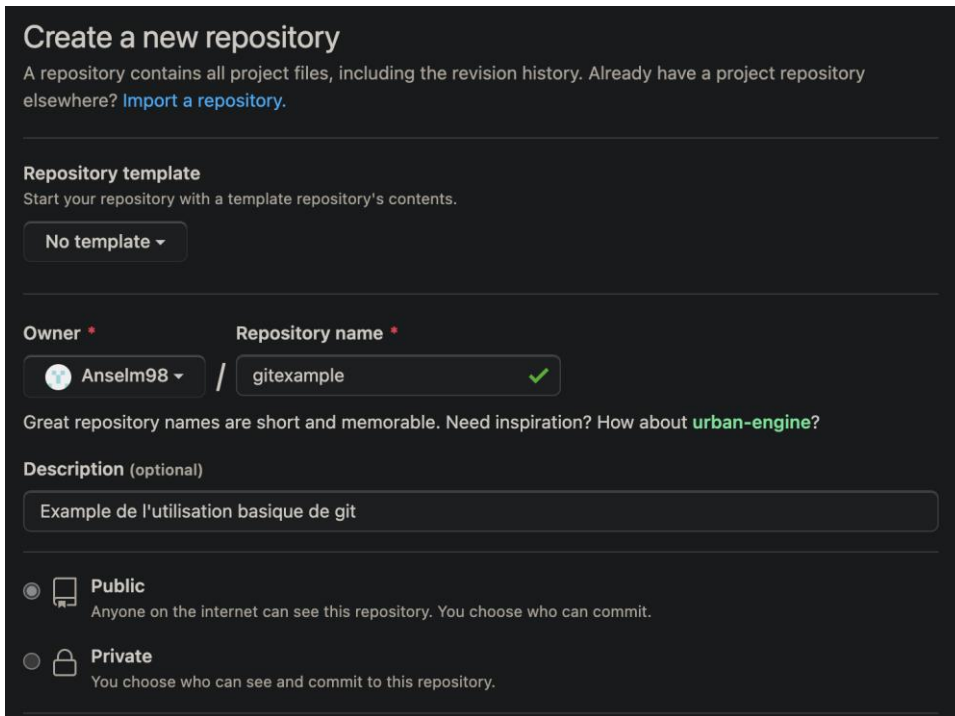


## Git

Lors de mon stage, j'ai eu l'occasion de travailler en utilisant git. Un outil indispensable dans le monde informatique d'aujourd'hui. Nous allons voir comment créer un dépôt git sur notre machine locale et comment l'exporter sur GitHub.

Nous allons tout d'abord créer le dépôt distant sur GitHub.



The screenshot shows the GitHub interface for creating a new repository. The title is 'Create a new repository'. Below it, a subtitle says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. There is a section for 'Repository template' with a dropdown menu set to 'No template'. Below that, the 'Owner' is 'Anselm98' and the 'Repository name' is 'gitexample', which is marked with a green checkmark. A hint suggests repository names should be short and memorable, with an example 'urban-engine'. There is an optional 'Description' field containing 'Exemple de l'utilisation basique de git'. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option description is 'Anyone on the internet can see this repository. You choose who can commit.' The 'Private' option description is 'You choose who can see and commit to this repository.'

Nous pouvons maintenant créer un dossier sur notre machine locale et l'initialiser. Git va créer les fichiers nécessaires à l'utilisation de git. Nous pouvons également configurer un .gitignore et y indiquer les fichiers que nous souhaiterions exclure de git.

```
→ Documents mkdir gitexample && cd gitexample
→ gitexample |
```

Pour configurer un dossier utilisez la commande "git init".

```
→ gitexample git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/cmaroy/Documents/gitexample/.git/
→ gitexample git:(master) |
```

Nous allons maintenant créer un fichier que nous allons commit avec git puis pousser sur notre dépôt distant

```
→ gitexample git:(master) touch fichierexample.txt
→ gitexample git:(master) x |
```

Tout d'abord nous allons devoir stage nos changements. Le stage est la première étape avant le commit. Il s'agit d'une étape où l'utilisateur choisit quels fichiers seront utilisés pour le commit. Nous allons sélectionner tout le dossier avec la commande **git add** . Puis nous allons vérifier quels fichiers sont prêts à être commit avec la commande **git status**

```
→ gitexample git:(master) x git add .
→ gitexample git:(master) x git status
On branch master

No commits yet

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

→ gitexample git:(master) x |
```

Nous pouvons maintenant lancer notre commit suivi d'un message avec **git commit** .

```
→ gitexample git:(master) ✕ git commit -m "mon premier commit"
[master (root-commit) 87a981e] mon premier commit
Committer: Clément MAROY <cmaroy@>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
```

```
git config --global --edit
```

After doing this, you may fix the identity used for this commit with:

```
git commit --amend --reset-author
```

```
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 fichierexemple.txt
→ gitexample git:(master) |
```

Nous pouvons voir l'historique des commit avec **git log**.

```
commit 87a981e4b695e60738d5128a85bd55a78f9e066c (HEAD -> master)
Author: Clément MAROY <cmaroy@>
Date: Tue Jun 21 10:03:09 2022 +0200

    mon premier commit
(END)
```

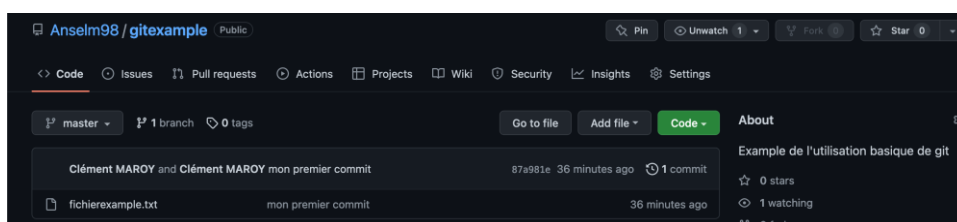
Nous allons maintenant ajouter le dépôt distant.

```
→ gitexample git:(master) git remote add origin git@github.com:Anselm98/gitexample.git
```

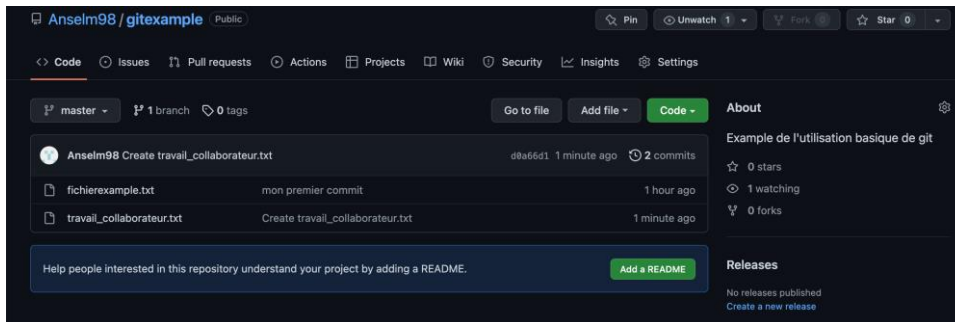
Nous pouvons maintenant pousser notre dépôt local sur github.

```
→ gitexample git:(master) git push --set-upstream origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 246 bytes | 123.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:Anselm98/gitexample.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
→ gitexample git:(master) |
```

Notre fichier est bien arrivé sur le dépôt github.



Nous allons simuler le travail d'un collaborateur en ajoutant un fichier depuis github puis nous allons mettre à jour notre dépôt local avec **git pull**.



```
→ gitexample git:(master) git pull
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
Unpacking objects: 100% (3/3), 705 bytes | 70.00 KiB/s, done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
From github.com:Anselm98/gitexample
   87a981e..d0a66d1  master    -> origin/master
Updating 87a981e..d0a66d1
Fast-forward
 travail_collaborateur.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 travail_collaborateur.txt
→ gitexample git:(master) |
```

Le travail de notre collaborateur a bien ajouté à notre dépôt.

## Git Branching

Nous allons créer une branche appelée "bugFix", modifier le fichier de notre collaborateur et faire une pull request.

Tout d'abord nous créons la branche puis nous nous plaçons dessus.

```
→ gitexample git:(master) git branch "bugFix"
→ gitexample git:(master) git checkout bugFix
Switched to branch 'bugFix'
→ gitexample git:(bugFix) |
```

Nous modifions le travail de notre collaborateur puis nous poussons sur le dépôt github.

```

→ gitexample git:(bugFix) git push --set-upstream origin bugFix
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 325 bytes | 325.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'bugFix' on GitHub by visiting:
remote:   https://github.com/Anselm98/gitexample/pull/new/bugFix
remote:
To github.com:Anselm98/gitexample.git

→ gitexample git:(bugFix) vi travail_collaborateur.txt
→ gitexample git:(bugFix) x git add .
→ gitexample git:(bugFix) x git commit -m "Correction de la syntaxe de la phrase"
[bugFix a66327f] Correction de la syntaxe de la phrase
Committer: Clément MAROY <cmaroy@MBP-ADM-02.admantic.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

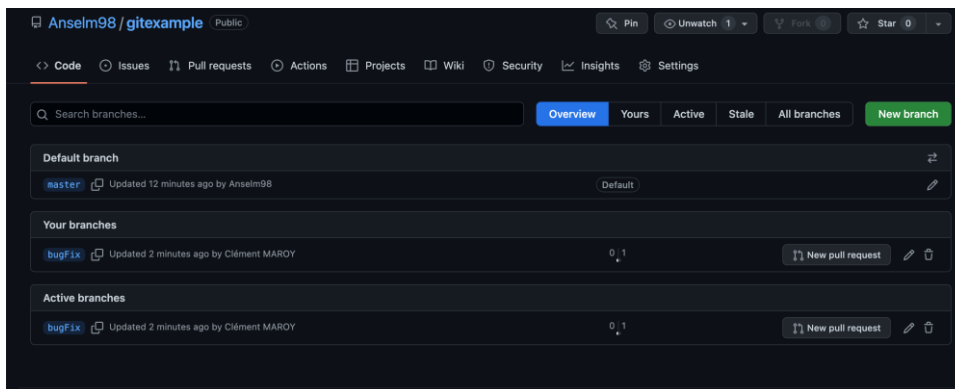
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

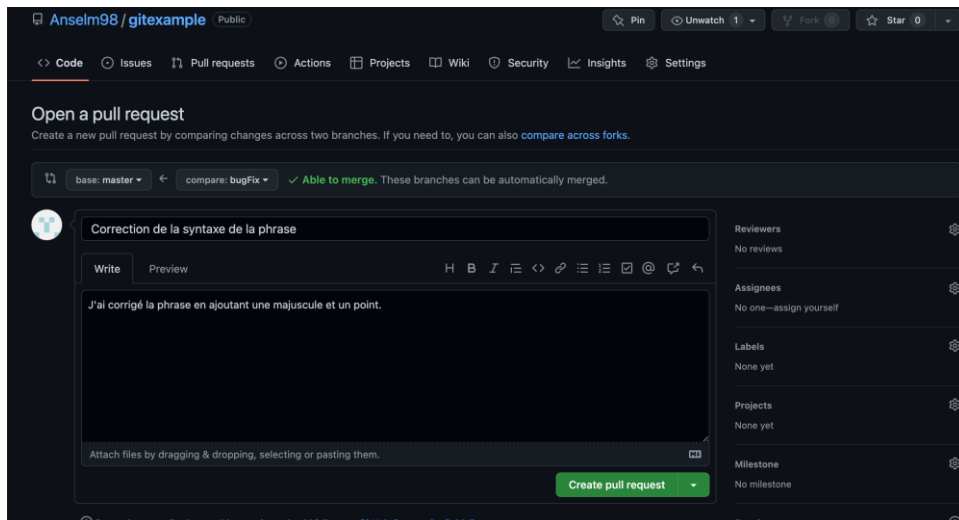
1 file changed, 2 insertions(+), 1 deletion(-)

```

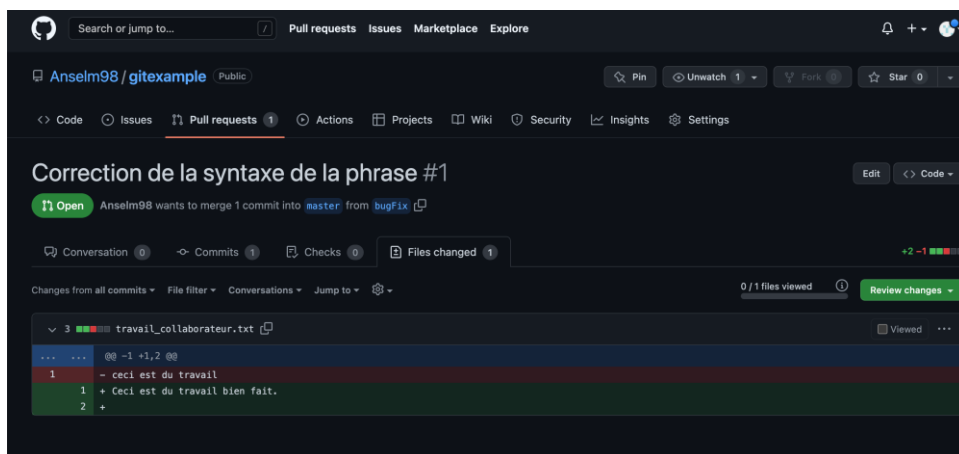
La branche a bien été poussée sur notre dépôt distant.



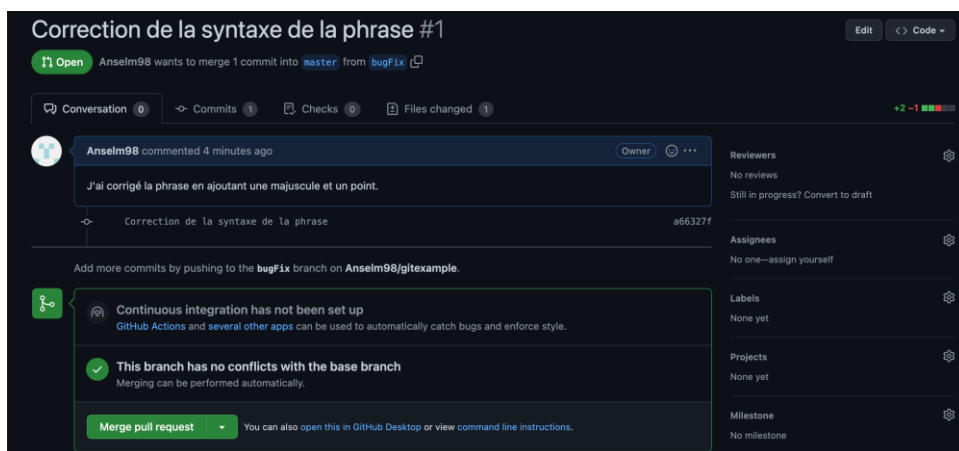
Nous pouvons maintenant faire une pull request afin d'intégrer bugFix à master.



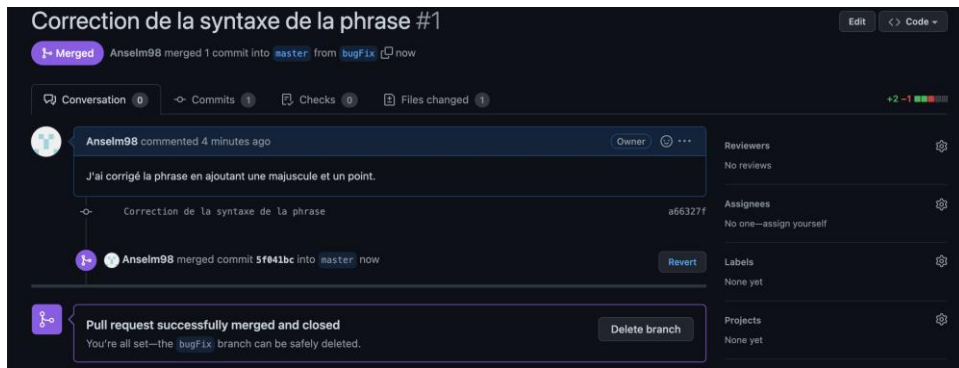
Github nous montre très facilement les changements qui ont été effectués dans le fichier. Nous pouvons accepter la pull request et intégrer la branche à master. Si le changement ne nous convient pas ou nécessite un changement, nous pouvons le proposer directement depuis l'interface de la pull request.



Le changement nous convient, nous allons accepter la pull request en cliquant sur “merge pull request”



La branche a été merged. Nous pouvons maintenant supprimer la branche bugFix.



Notre pouvons voir en vérifiant dans la branche master que le fichier a bien été modifiée.

