

## Laboratoire 3 – Le dépôt distant (GitHub)

Le but de ce laboratoire est de vous rendre confortable avec les commandes de connexion sur le dépôt distant.

Commandes :	Utilité :
:	
git remote add :	
git remote -v :	
git show :	
git checkout :	
.gitignore	
git reflog	
git clone :	
git push :	
git pull :	
git remote show	

Revenez compléter ce tableau à la fin de l'exercice.

Prérequis, vous devez avoir installé git sur votre ordinateur et posséder un compte GitHub.

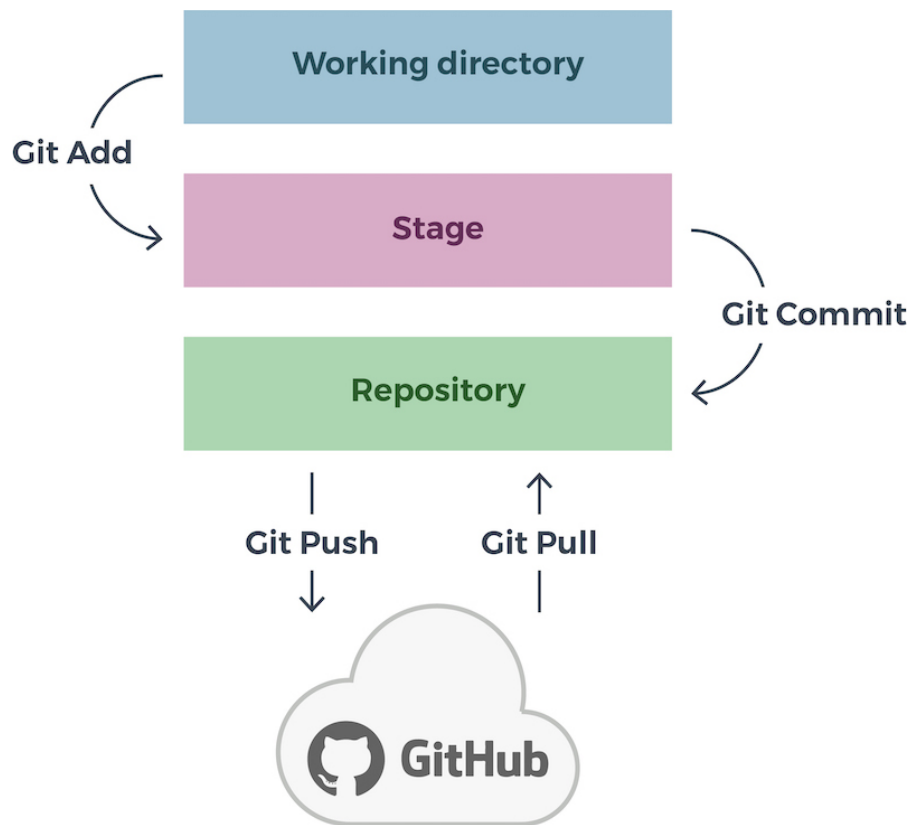
Ce laboratoire va s'effectuer en mode console et sur l'environnement web de GitHub pour démarrer la console aller sur la barre de recherche de démarrage et saisissez **git bash**, ensuite sélectionner ouvrir.

## Travailler avec des dépôts distants

Pour pouvoir collaborer sur un projet Git, il est nécessaire de savoir comment gérer les dépôts distants. Les dépôts distants sont des versions de votre projet qui sont hébergées sur Internet ou le réseau d'entreprise. Vous pouvez en avoir plusieurs, pour lesquels vous pouvez avoir des droits soit en lecture seule, soit en lecture/écriture.

Collaborer avec d'autres personnes consiste à gérer ces dépôts distants, en poussant ou tirant des données depuis et vers ces dépôts quand vous souhaitez partager votre travail. Gérer des dépôts distants inclut savoir comment ajouter des dépôts distants, effacer des dépôts distants qui ne sont plus valides, gérer des branches distantes et les définir comme suivies ou non, et plus encore. Dans cette section, nous traiterons des commandes de gestion distante.<sup>1</sup>

Pour ajouter un nouveau dépôt distant Git comme nom court auquel il est facile de faire référence, lancez : ***git remote add [nomcourt] [url]*** :

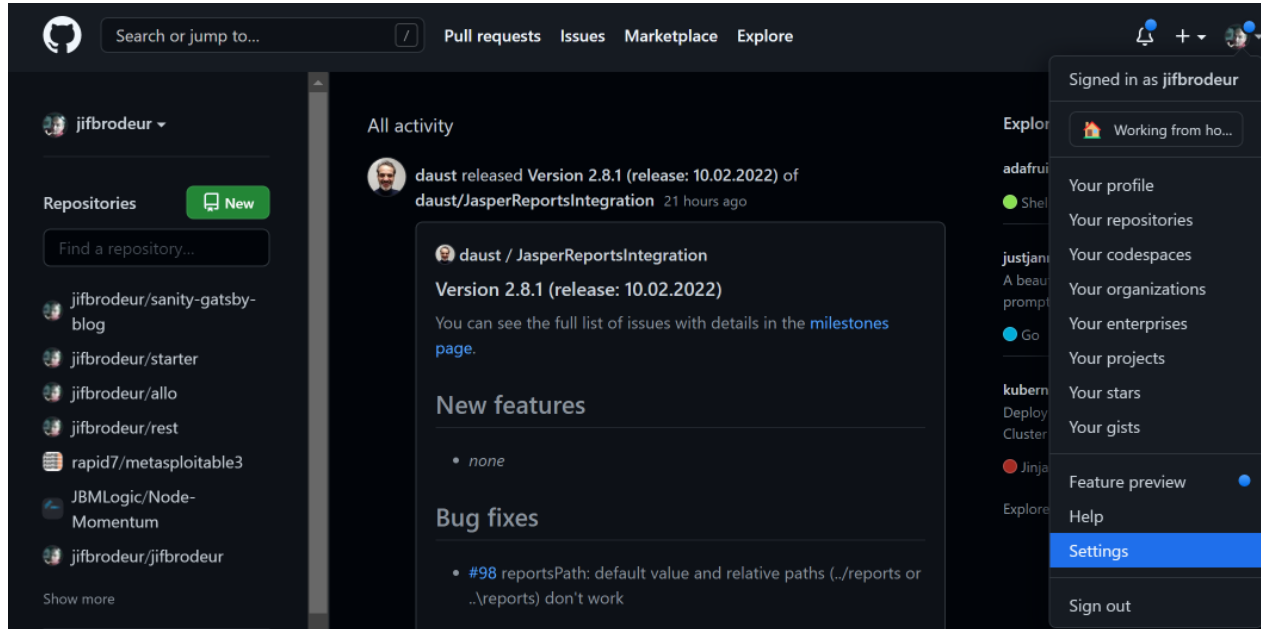


Suite à l'ajout d'un dépôt distant, il vous sera possible par la suite de synchroniser ce dépôt distant avec votre dépôt local à l'aide des commandes git push et git pull tel qu'illustré sur le diagramme ici-haut.

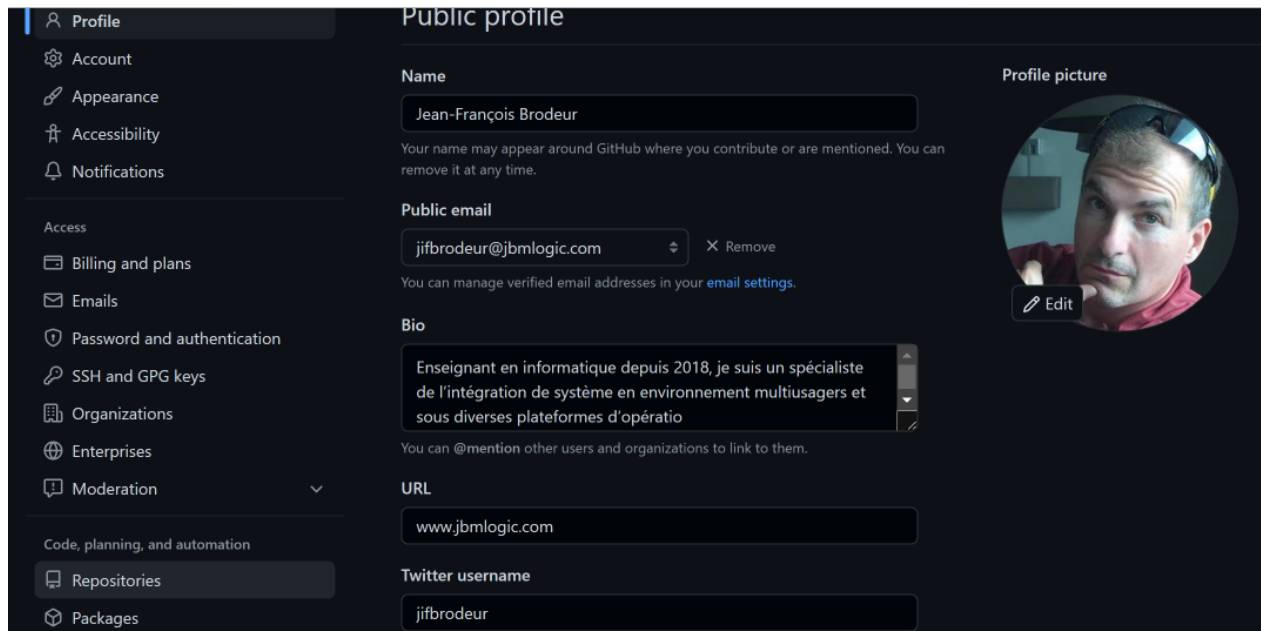
<sup>1</sup> <https://git-scm.com/book/fr/v2/Les-bases-de-Git-Travailler-avec-des-d%C3%A9p%C3%B4ts-distants>

## Création d'un nouveau dépôt distant

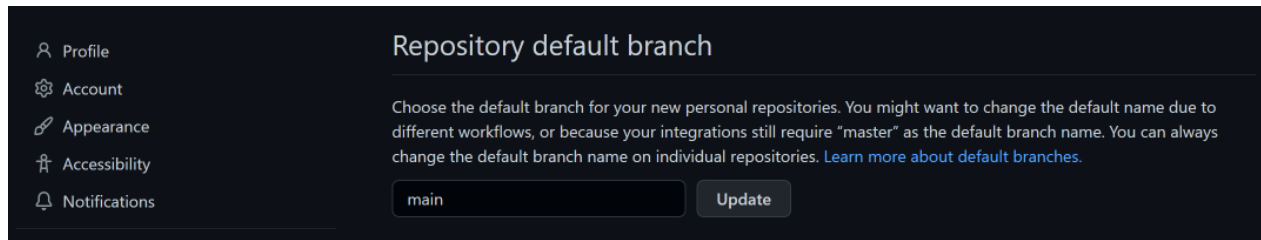
Dans un premier temps, connectez-vous à votre compte GitHub en allant à l'adresse <https://github.com>.



Avant de créer votre dépôt distant, vous devez d'abord vous assurer que le nom de votre branche initiale sur votre git est la même que celui de GitHub. Car git se synchronise par branche et non par dépôt complet. Cliquez sur votre image de profil en haut à droite et sélectionnez le menu Settings.



Ensuite, sélectionnez Repositories en bas dans le menu de gauche pour la configuration du dépôt.



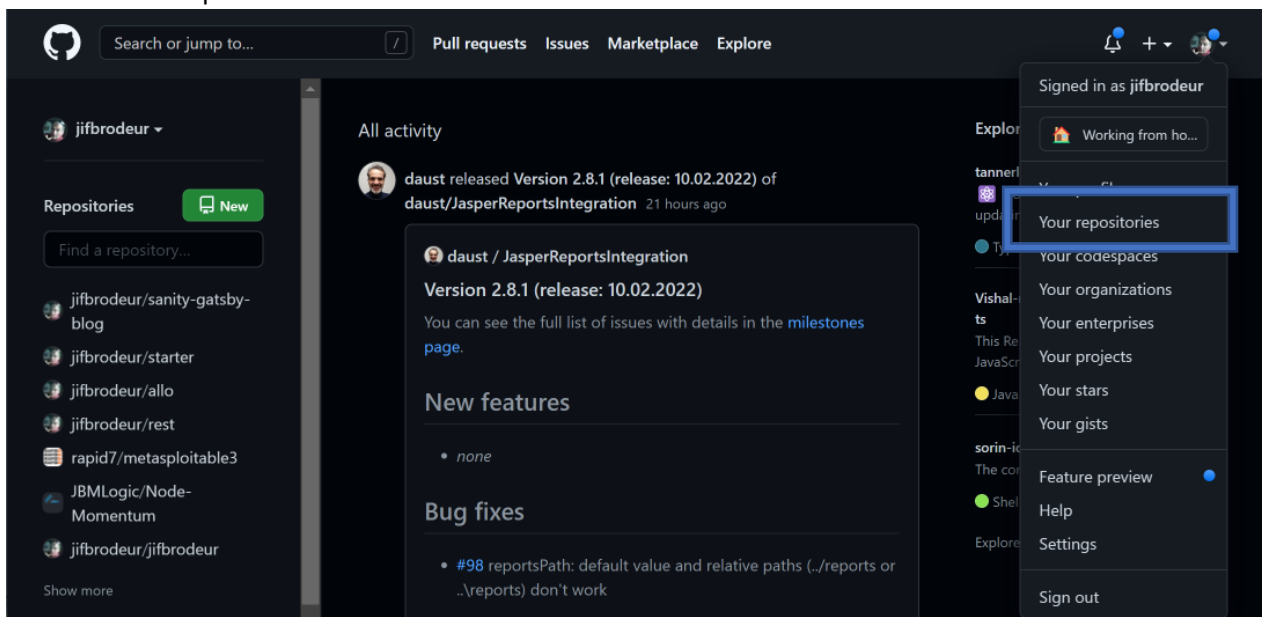
La première option est le nom de votre branche initiale par défaut, ici-haut c'est main indiquez ici le nom de la vôtre : adam20301.

Revenez à l'invite de commande git, puis tapez la commande suivante afin de vérifier le paramètre du nom de la branche initiale de git : **git config --get init.defaultBranch** si la valeur indiquée n'est pas la même que celle de la branche initiale par défaut de GitHub veuillez la modifier avec la commande suivante : **git config --global init.defaultBranch main** (main étant le nom de la branche).

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~  
$ git config --get init.defaultBranch  
master  
  
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~  
$ git config --global init.defaultBranch main  
  
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~  
$ git config --get init.defaultBranch  
main
```

Dans ce laboratoire, vous allez créer un dépôt distant sur votre compte GitHub et le rattacher à un dépôt local afin de le synchroniser et vous familiariser avec les commandes pour travailler avec des dépôts distants. Connectez-vous à votre compte GitHub en allant à l'adresse <https://github.com>.

1. Sélectionnez « Your repositories » (Vos dépôts) dans le menu de votre avatar en haut à droite. D'abord sélectionner l'onglet « Repositories » et « 2 » Cliquer sur le bouton « New » pour la création d'un dépôt.



## 420-2CW-BB Applications Web

1. Choisir un nom disponible : dans la mesure du possible, toujours utiliser des noms significatifs
2. Décider le niveau d'accès, s'assurer que vos enseignants aient accès
3. Ne pas choisir les fichiers à l'initialisation, tentez de découvrir leur signification, essayer de comprendre à quoi servent la licence et le fichier .gitignore?

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)


---

**Repository template**  
Start your repository with a template repository's contents.

No template ▾

---


**Owner \***      **Repository name \***


 jifbrodeur ▾ /

Great repository names are short and memorable. Need inspiration? How about **verbose-disco**?

**Description (optional)**

---

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---

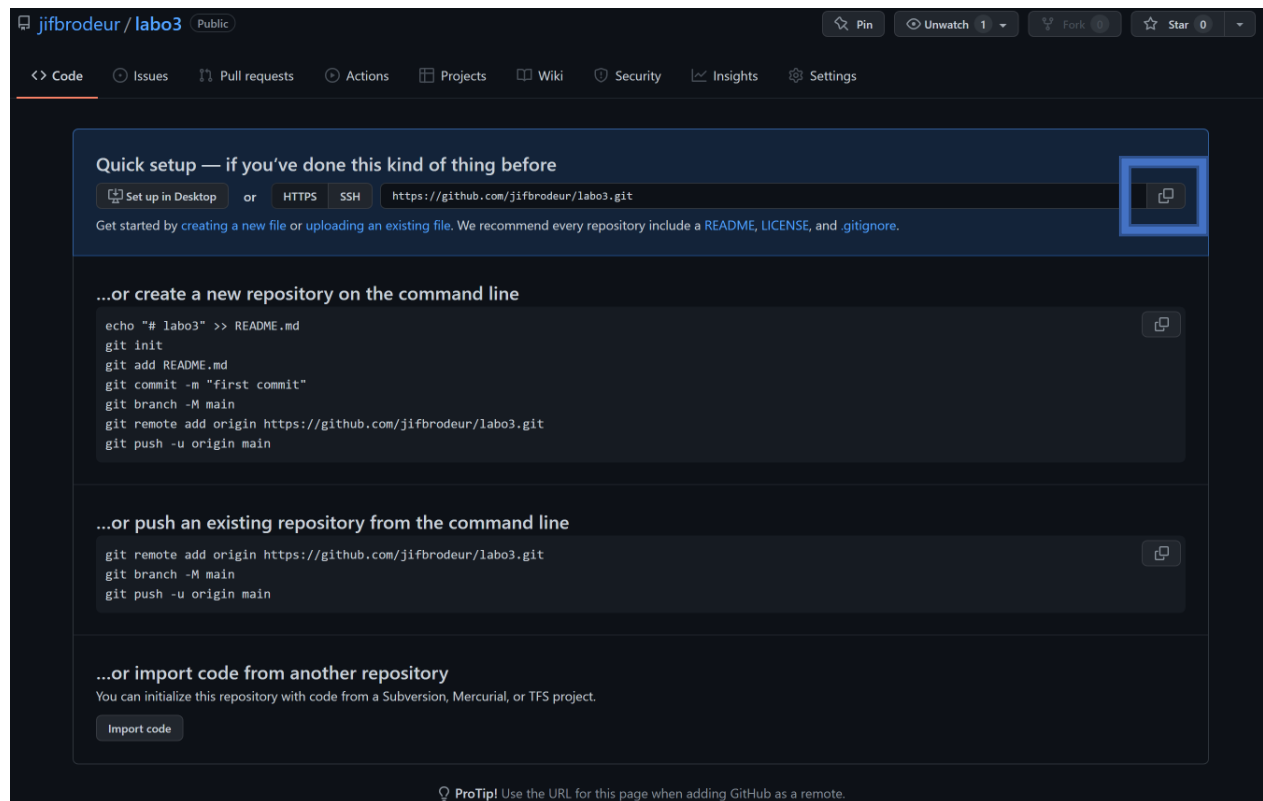
**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

4. Formaliser le tout en créant le dépôt distant



Ici-haut se trouve le résultat attendu, vous avez créé votre premier dépôt (repository). Dans les opérations suivantes vous apprendrez à synchroniser votre dépôt local avec celui-ci.

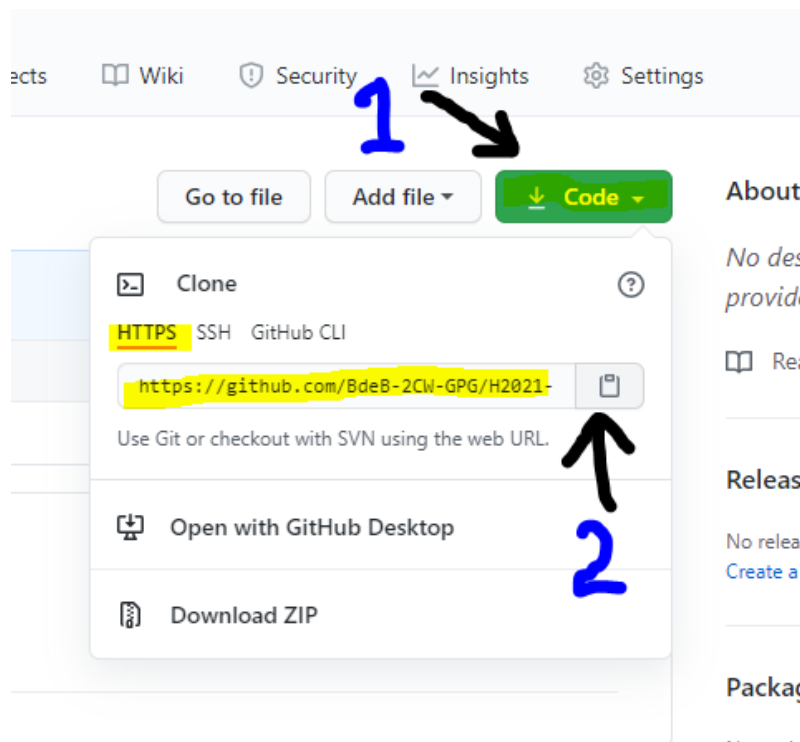
Prenez note que le carré « Quick Setup » affiche le URL exposé de votre dépôt sur GitHub, à droite se trouve un bouton pour copier ce URL dans votre presse-papier et vous pourrez ensuite le coller dans vos lignes de commande pour relier votre dépôt local. En dessous se trouvent aussi les commandes afin de relier ce dépôt distant à votre dépôt local.

## Se connecter au dépôt distant

Vous devez aller chercher les informations pour vous connecter (à partir de votre environnement local) au dépôt distant. Positionnez sur votre dépôt distant dans l'interface de GitHub, cliquez sur l'onglet Code et le bouton « Code » pour avoir accès à l'information de cloning. Si votre dépôt est vide et que vous venez de le créer vous aurez plutôt l'information directement dans l'onglet Code.

Sélectionnez la méthode « HTTPS » qui est plus simple d'utilisation mais requiert l'authentification à chaque session de travail.

Cliquez sur l'icône (2), cela va faire une copie de l'adresse en jaune juste à côté, vous pouvez vérifier en collant l'adresse, dans mon cas : <https://github.com/BdeB-2CW-GPG/H2021-2CW.git>



**420-2CW-BB Applications Web**

Positionnez-vous dans un répertoire de travail qui contient quelques fichiers ou créez un nouveau répertoire projet, labo3 et créez-y deux fichiers soit index.html et index.css faites par la suite **votre git init, git add et git commit -m – "message"**:

**Note : il n'est pas possible de synchroniser un dépôt local qui n'a pas de commit.**

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/labo3 (main)
$ code .

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/labo3 (main)
$ ls
index.css  index.html

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/labo3 (main)
$ git status
On branch main
nothing to commit, working tree clean

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/labo3 (main)
$ git log
commit 55763dbd1b8d8d0472fbc5adb5f7c6b835b2a7d0 (HEAD -> main)
Author: Jean-Francois Brodeur <jifbrodeur@jbmlogic.com>
Date:   Fri Feb 11 10:14:13 2022 -0500

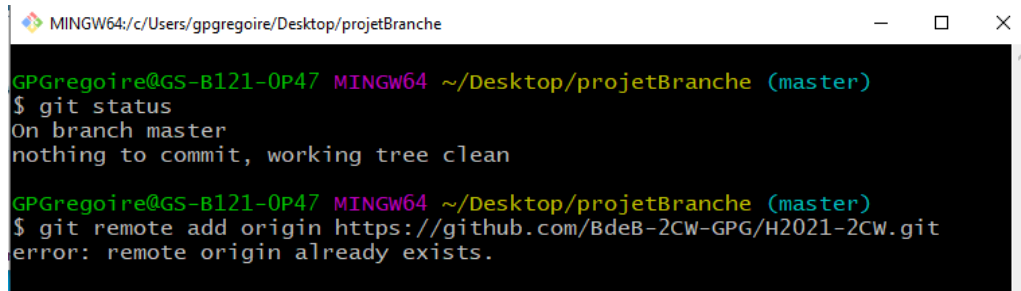
    création de l'index
```

Il faut ensuite ajouter le répertoire distant à la configuration de votre dépôt local en effectuant la commande suivante : ***git remote add origin url\_de\_votre\_dépot***

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/labo3 (main)
$ git remote add origin https://github.com/jifbrodeur/labo3.git
```



À faire seulement une fois, sinon vous obtiendrez le message d'erreur suivant :

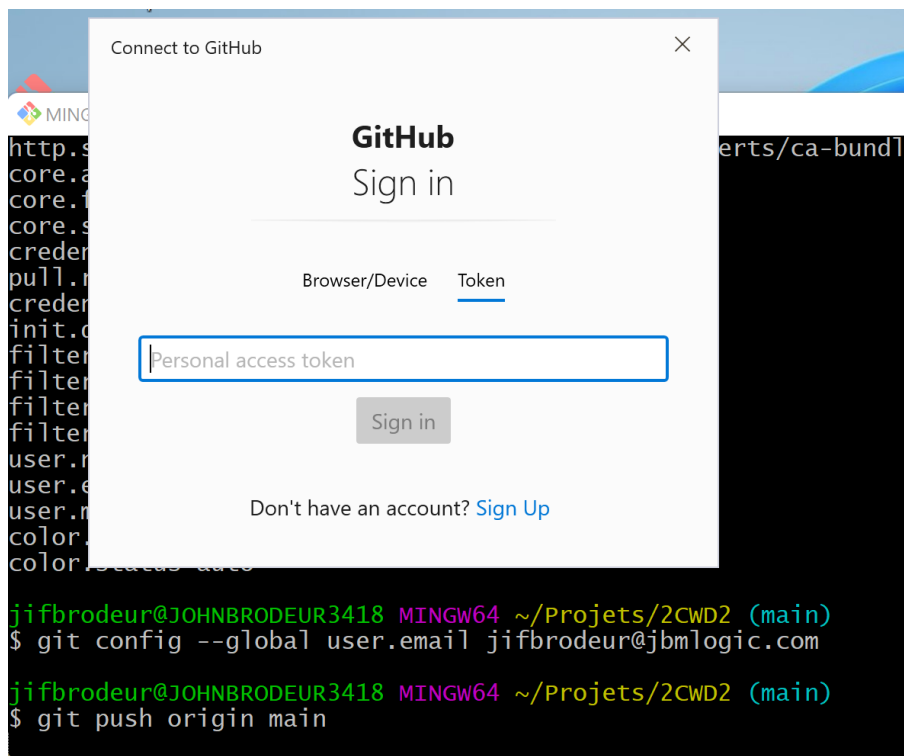


```
MINGW64:/c/Users/gpgregoire/Desktop/projetBranche
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ git status
On branch master
nothing to commit, working tree clean



GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ git remote add origin https://github.com/BdeB-2CW-GPG/H2021-2CW.git
error: remote origin already exists.
```

La connexion requiert une permission. Faites attention, il se peut que cette fenêtre ne soit pas immédiatement visible, il se peut qu'elle apparaisse derrière une autre fenêtre.

Seulement pour un ordinateur macOS, Sélectionnez "Token", pour Windows sélectionnez Browser/Device :



Ensuite saisissez votre nom d'utilisateur/mot de passe du site GitHub :




Sign in to **GitHub**  
to continue to **Git Credential Manager**

---


Username or email address


Password [Forgot password?](#)


Finalement autorisez l'application :




Authorize Git Credential Manager

 **Git Credential Manager by GitCredentialManager**  
wants to access your jifbrodeur account

 **Gists**  
Read and write access

 **Repositories**  
Public and private

 **Workflow**  
Update GitHub Action Workflow files.

Authorizing will redirect to  
<http://localhost:49402>

```
MINGW64/c/Users/gpgregoire/Desktop/projetBranche
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ git push -u origin master
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 8 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (26/26), 2.58 KiB | 1.29 MiB/s, done.
Total 26 (delta 7), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (7/7), done.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/BdeB-2CW-GPG/H2021-2CW/pull/new/master
remote:
To https://github.com/BdeB-2CW-GPG/H2021-2CW.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

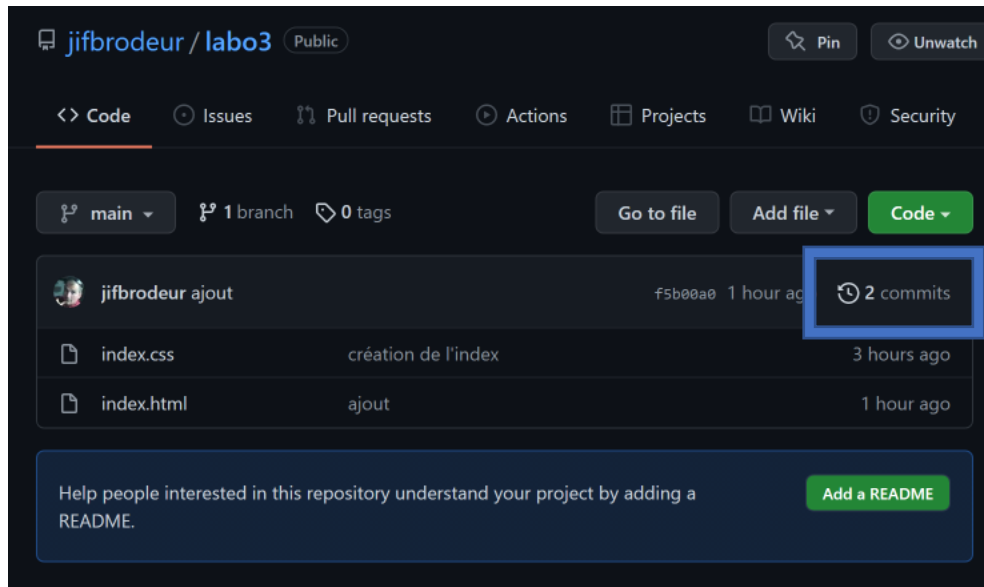
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ |
```

```
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ git reflog
198374d (HEAD -> master, origin/master, branch2) HEAD@{0}: merge branch2:
Fast-forward
3b55c3a HEAD@{1}: checkout: moving from branch2 to master
198374d (HEAD -> master, origin/master, branch2) HEAD@{2}: commit: ajout d
u fichier test2.txt
70e3e9d HEAD@{3}: checkout: moving from master to branch2
3b55c3a HEAD@{4}: checkout: moving from branch2 to master
70e3e9d HEAD@{5}: commit: Modification de test.txt dans la branche branch2
3b55c3a HEAD@{6}: checkout: moving from master to branch2
3b55c3a HEAD@{7}: commit: ajout du fichier test.txt
03288db HEAD@{8}: commit: modification de index.css
0ef5dcf HEAD@{9}: commit (amend): ajout du fichier allo.js pour afficher H
ello
391cd0a HEAD@{10}: commit: ajout du fichier allo.js pour afficher Hello
11942fc HEAD@{11}: reset: moving to HEAD^
4af8831 HEAD@{12}: commit: retirer la couleur de fond de index.css
11942fc HEAD@{13}: reset: moving to HEAD
11942fc HEAD@{14}: commit (amend): Ajout du fichier CSS - index.css
4e03a99 HEAD@{15}: commit: Ajout du fichier CSS
21dc32f HEAD@{16}: commit: Ajout de ma couleur preferee dans la balise p
8c129bc HEAD@{17}: commit (initial): Premiere sauvegarde du projet

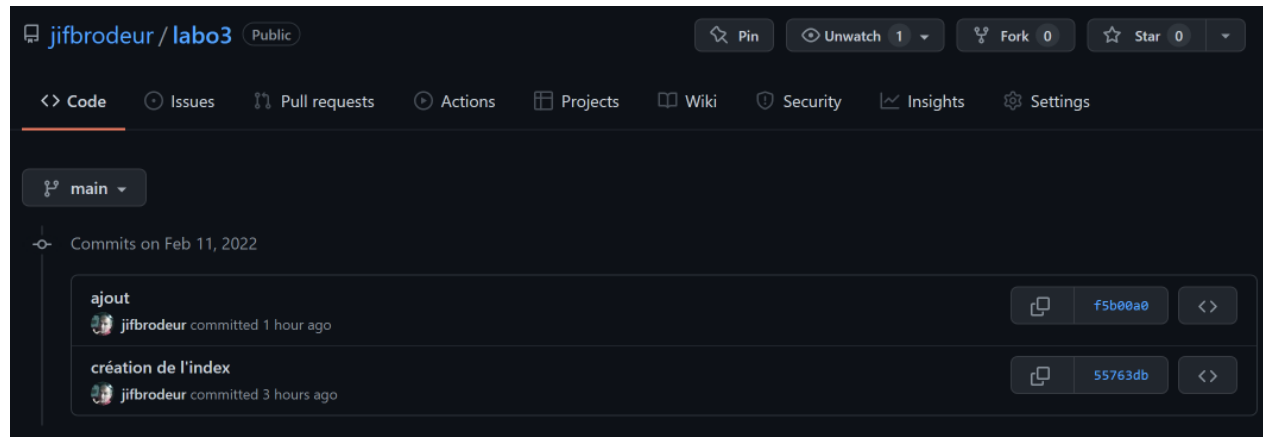
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ |
```

## 420-2CW-BB Applications Web

Aller voir le résultat sur votre compte GitHub et appuyer sur le nombre de commits afin de voir chacun des commits qui ont été télé versés dans votre dépôt distant.



Naviguer dans l'environnement GitHub pour découvrir les différents COMMITS et autres découvertes.



## 420-2CW-BB Applications Web

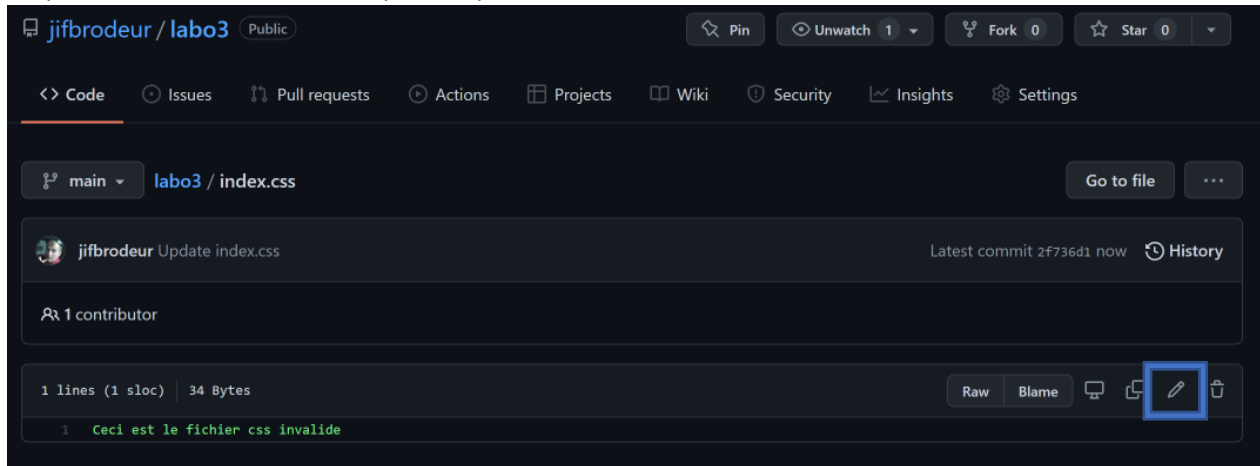
The screenshot shows the GitHub interface for a commit titled 'ajout' in the 'jifbrodeur / labo3' repository. The commit is on the 'main' branch and was made 1 hour ago. It shows a diff for 'index.html' with 1 addition and 1 deletion. The diff highlights the addition of a title tag: `<title>Document labo3 ajout</title>`. The commit hash is f5b00a09ecb0bc977af9f6afe020e336998e1870. The interface includes navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A 'Browse files' button is visible in the top right of the commit view.

On va modifier un fichier dans GitHub pour ensuite PULLer la modification dans votre environnement local. Pour cela, revenez à l'onglet code de votre dépôt GitHub.

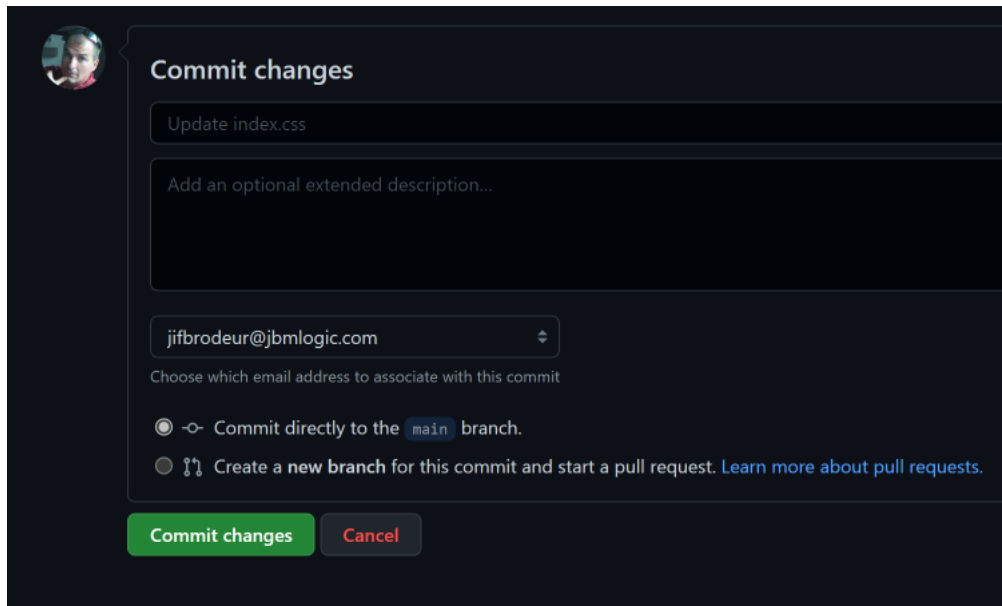
The screenshot shows the GitHub repository page for 'jifbrodeur / labo3'. The 'Code' tab is selected. The repository has 1 branch (main) and 0 tags. The file list shows 'index.css' (création de l'index, 3 hours ago) and 'index.html' (ajout, 1 hour ago). The commit history shows a commit titled 'ajout' by jifbrodeur, 1 hour ago, with 2 commits. The interface includes navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, and Security. Buttons for 'Go to file', 'Add file', and 'Code' are visible.

## 420-2CW-BB Applications Web

Cliquez sur le fichier à modifier puis Cliquer sur « edit this file »




Faites des modifications, comme l'ajout de lignes et appuyez au bas sur COMMIT.








Votre fichier a été modifié récemment.

This branch is 9 commits ahead, 1 commit behind main. [Pull request](#) [Compare](#)

 BdeB-2CW-GPG Update test.txt

b90ddd0 in 26 seconds 9 commits

 allo.js	ajout du fichier allo.js pour afficher Hello	4 days ago
 index.css	modification de index.css	4 days ago
 index.html	ajout du fichier allo.js pour afficher Hello	4 days ago
 test.txt	Update test.txt	now
 test2.txt	ajout du fichier test2.txt	4 days ago

Help people interested in this repository understand your project by adding a README. [Add a README](#)

État du fichier dans votre répertoire local avant le PULL

```
MINGW64/c/Users/gpgregoire/Desktop/projetBranche
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ cat test.txt
fichier à supprimer
Modification de test.txt dans la branche branch2
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
```

Faire le PULL qui téléchargera votre dépôt distant et mettra à jour votre dépôt local.

À l'aide de la commande « cat » du fichier que vous avez modifié, valider que le contenu est modifié.

```
GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ git pull origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 736 bytes | 73.00 KiB/s, done.
From https://github.com/BdeB-2CW-GPG/H2021-2CW
* branch            master       -> FETCH_HEAD
   198374d..b90ddd0  master       -> origin/master
Updating 198374d..b90ddd0
Fast-forward
 test.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

GPGregoire@GS-B121-0P47 MINGW64 ~/Desktop/projetBranche (master)
$ cat test.txt
fichier à supprimer
Modification de test.txt dans la branche branch2
Je modifie le texte dans GitHub
```

La commande **git remote** contient plusieurs options pour afficher l'état du dépôt distant

Saisissez la commande **git remote** pour afficher le dépôt distant courant, car oui vous pouvez en avoir plusieurs :

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (main)
$ git remote
origin
```

L'option -v de la commande (verbose) affiche l'état des push/pull du dépôt distant.

Saisissez la commande **git remote -v** (vous n'aurez pas testing mais plutôt main)

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (testing)
$ git remote -v
origin https://github.com/jifbrodeur/2CWD2.git (fetch)
origin https://github.com/jifbrodeur/2CWD2.git (push)
```

Créez une nouvelle branche dans votre dépôt local intitulé verification, **git branch verification** et positionnez-vous sur cette branch **git switch verification** ajouter un nouveau fichier test.txt et aller poussez cette branche sur le dépôt distant avec la commande **git push origin verification** et enfin visualiser l'état du dépôt distant avec **git remote show origin**.

```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (testing)
$ git branch verification

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (testing)
$ git switch verification
Switched to branch 'verification'

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (verification)
$ touch fichier.txt

jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (verification)
$ git push origin verification
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'verification' on GitHub by visiting:
remote: https://github.com/jifbrodeur/2CWD2/pull/new/verification
remote:
To https://github.com/jifbrodeur/2CWD2.git
 * [new branch] verification -> verification
```



```
jifbrodeur@JOHNBRODEUR3418 MINGW64 ~/Projets/2CWD2 (verification)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/jifbrodeur/2CWD2.git
Push URL: https://github.com/jifbrodeur/2CWD2.git
HEAD branch: main
Remote branches:
  main          tracked
  testing       tracked
  verification  tracked
Local branch configured for 'git pull':
  main merges with remote main
Local refs configured for 'git push':
  main          pushes to main          (up to date)
  testing       pushes to testing       (up to date)
  verification  pushes to verification (up to date)
```

Aller explorer dans GitHub et voyez-vous les branches additionnelles que vous avez créées?

---

---

---

### Problèmes de connexion

Si vous avez des problème de connexion entre git et GitHub il faut investiguer le paramètre `credential.helper = manager-core`