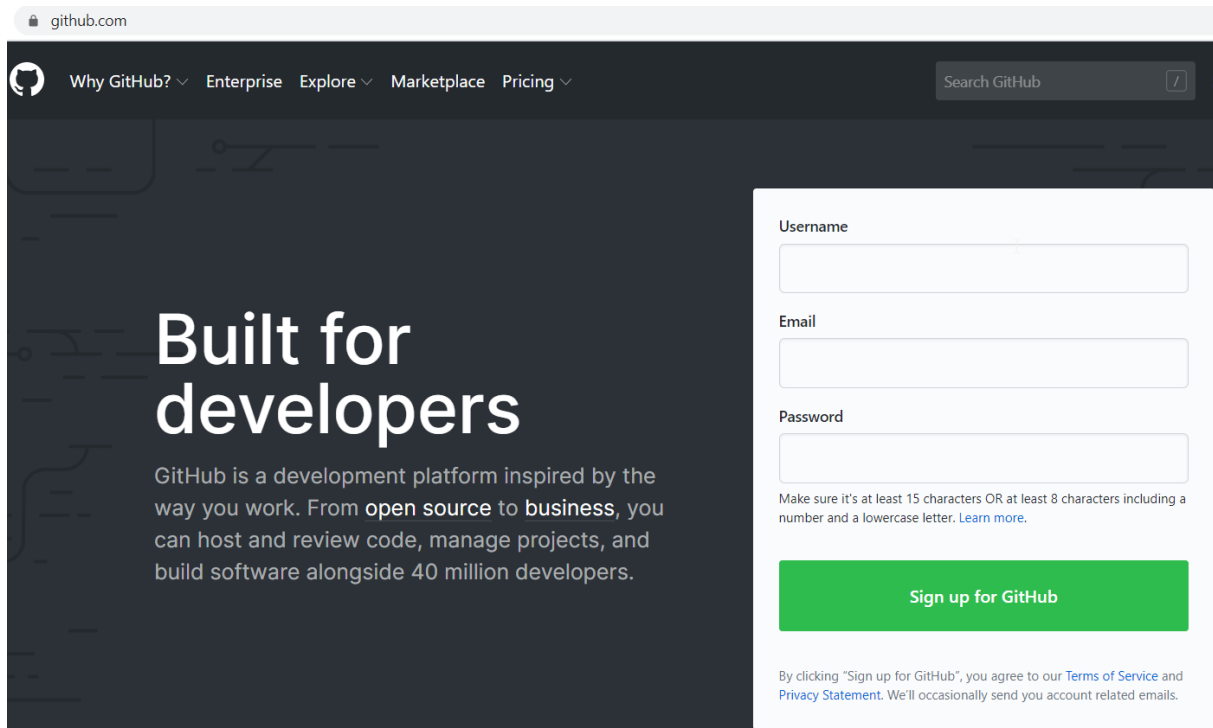
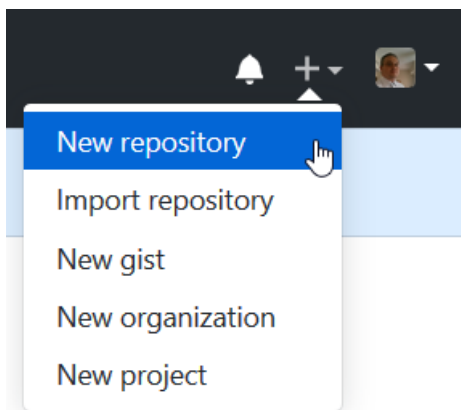


Travailler avec des dépôts distants

1- Créez un compte sur GitHub et un nouveau projet (repository) sur github.com :



The screenshot shows the GitHub homepage with a dark background. On the left, the text "Built for developers" is prominently displayed, followed by a description of GitHub as a development platform. On the right, there is a white sign-up form with fields for Username, Email, and Password. Below the Password field, there is a note about password requirements and a link to "Learn more". A green "Sign up for GitHub" button is at the bottom of the form. At the very bottom of the form, there is a small disclaimer about agreeing to the Terms of Service and Privacy Statement.



Tout d'abord, si ce n'est pas déjà fait, connectez-vous à votre compte GitHub. Cliquez sur le bouton "New" symbolisé par un signe "+" en haut à droite de votre écran, puis "New repository".

GitHub vous demandera alors de préciser quelques détails sur votre repository:

- son nom
- sa description
- son statut public ou privé.

Vous avez également une option "Initialize this repository with a README", qui vous permet de cloner votre repository sur votre machine. Cette option est à cocher uniquement dans le cas

où vous n'avez pas encore créé le repository en question sur votre machine (ce qui est bien notre cas ici, cochez donc !)

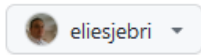
Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *



Repository name *

/ test-demo

✓ test-demo is available.

Great repository names are short and memorable. Need inspiration? How about **glowing-fiesta** ?

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:

☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

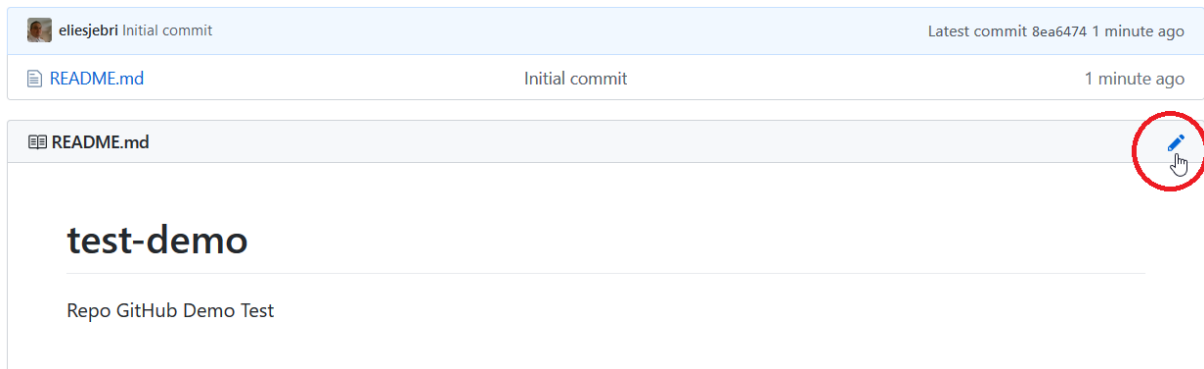
A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set main as the default branch. Change the default name in your [settings](#).

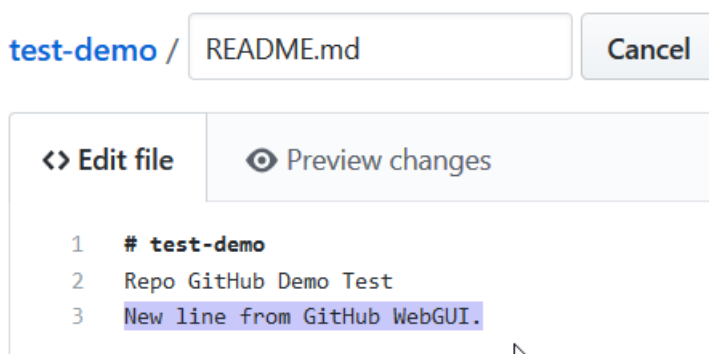
You are creating a public repository in your personal account.

Create repository

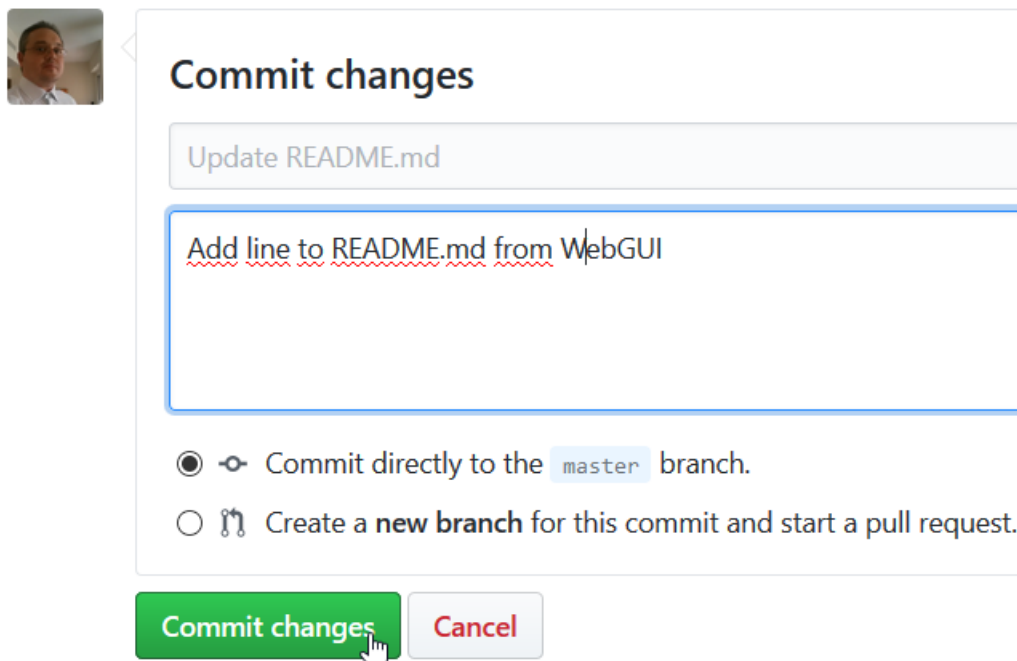
Editez le fichier « README.md »



Et ajoutez le contenu suivant :

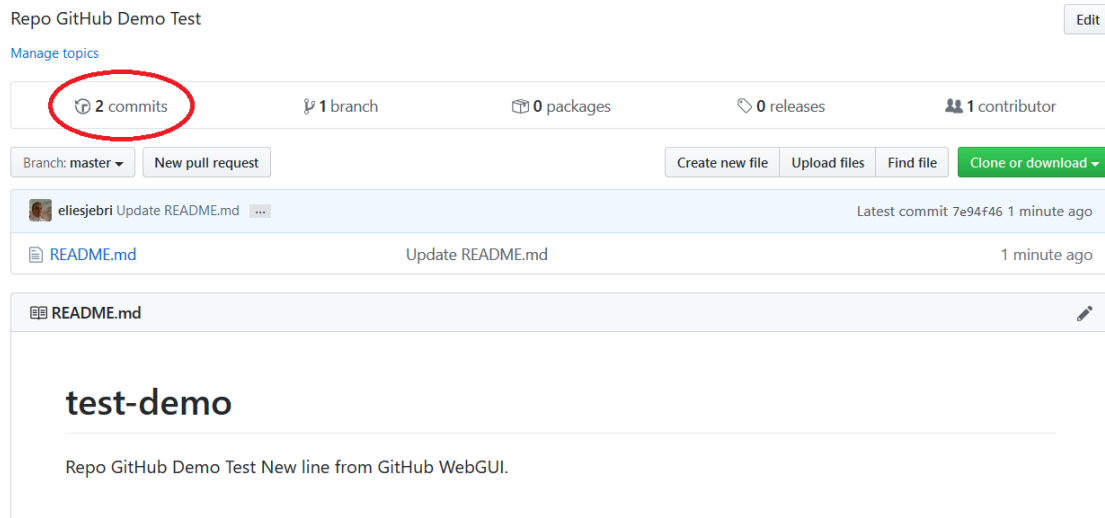


Effectuez ensuite un commit de votre changement :

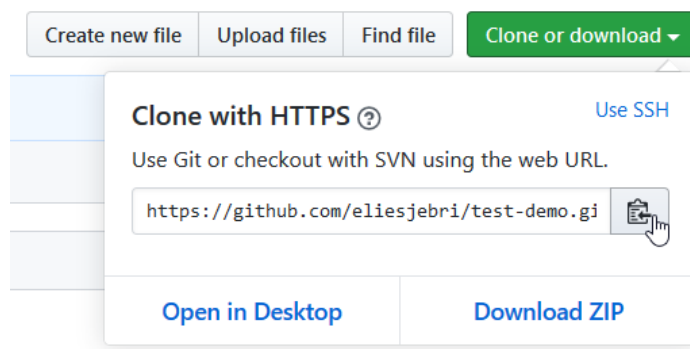


Vous pouvez maintenant :

- Voir les commits effectués sur votre repository (par vous et vos éventuels contributeurs)



- Cloner votre projet et le ramener sur votre machine avec la commande git clone. Copiez l'URL du dépôt pour ensuite le cloner localement.



```
[user@localhost ~]$ cd ; cd LabsGit/
```

```
[user@localhost ~/LabsGit]$ git clone https://github.com/eliesjebri/test-demo.git
```

2- Afficher les dépôts distants

Pour visualiser les serveurs distants que vous avez enregistrés, vous pouvez lancer la commande git remote depuis le répertoire de ce dépôt. Elle liste les noms des différentes références distantes que vous avez spécifiées.

```
[user@localhost ~/LabsGit]$ cd test-demo/
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git remote origin
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git remote -v
origin https://github.com/eliesjebri/test-demo.git (fetch)
origin https://github.com/eliesjebri/test-demo.git (push)
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git remote show origin
* remote origin
Fetch URL: https://github.com/eliesjebri/test-demo.git
Push URL: https://github.com/eliesjebri/test-demo.git
HEAD branch: main
Remote branch:
  main tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (up to date)
```

3- Récupérer et tirer depuis des dépôts distants

Essayons d'une autre façon (add-fetch-pull) de créer un clone local du dépôt distant :

```
[user@localhost ~/LabsGit/fetch-test]$ cd ; cd LabsGit/
```

```
[user@localhost ~/LabsGit]$ mkdir fetch-test
```

```
[user@localhost ~/LabsGit]$ cd fetch-test/
```

```
[user@localhost ~/LabsGit/fetch-test]$ git init
Initialized empty Git repository in /home/user/LabsGit/fetch-test/.git/
```

```
[user@localhost ~/LabsGit/fetch-test]$ git remote add origin
https://github.com/eliesjebri/test-demo.git
```

```
[user@localhost ~/LabsGit/fetch-test]$ git remote
origin
```

```
[user@localhost ~/LabsGit/fetch-test]$ git remote -v
origin https://github.com/eliesjebri/test-demo.git (fetch)
origin https://github.com/eliesjebri/test-demo.git (push)
```

```
[user@localhost ~/LabsGit/fetch-test]$ git fetch origin
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
```

```
Unpacking objects: 100% (6/6), done.  
From https://github.com/eliesjebri/test-demo  
* [new branch]      main      -> origin/main
```

```
[user@localhost ~/LabsGit/fetch-test]$ git remote show origin  
* remote origin  
  Fetch URL: https://github.com/eliesjebri/test-demo.git  
  Push URL: https://github.com/eliesjebri/test-demo.git  
  HEAD branch: main  
  Remote branch:  
    main tracked
```

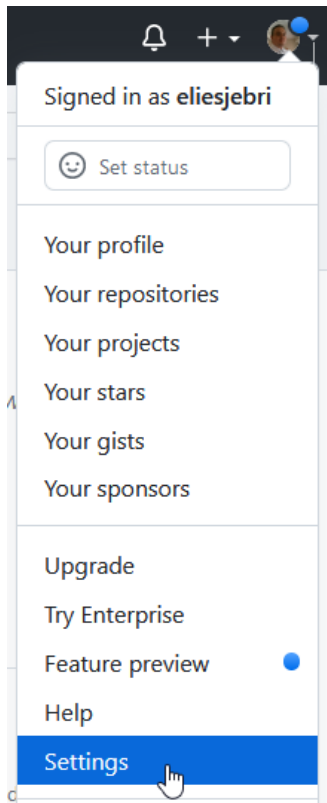
```
[user@localhost ~/LabsGit/fetch-test]$ git pull origin main  
From https://github.com/eliesjebri/test-demo  
* branch            main      -> FETCH_HEAD
```

```
[user@localhost ~/LabsGit/fetch-test] (main)$ ls  
README.md
```

```
[user@localhost ~/LabsGit/fetch-test] (main)$ git log --oneline --decorate  
7e94f46 (HEAD, origin/main, main) Update README.md  
8ea6474 Initial commit
```

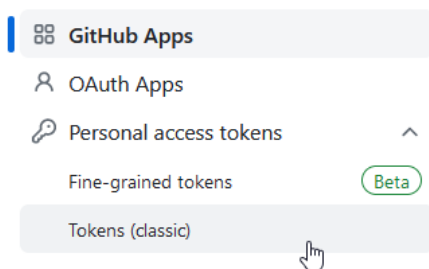
4- Mettre en place l'authentification par token

- a. Connectez-vous à GitHub et accédez à la Settingspage comme indiqué ci-dessous :

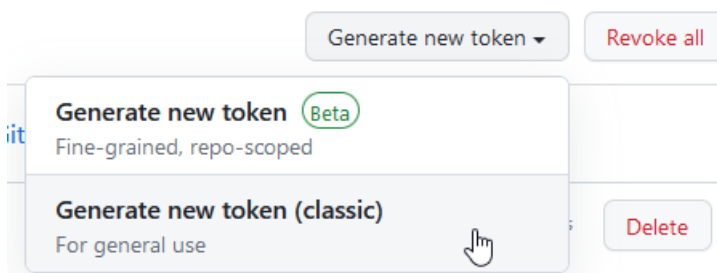


- b. Cliquez sur Developer Settings
c. Cliquez sur Jetons d'accès personnels

[Settings](#) / Developer settings



- d. Cliquez sur Generate new token



- e. Saisissez maintenant le nom du jeton et sélectionnez les étendues ou les autorisations que vous souhaitez accorder à ce jeton. Assurez-vous de tout sélectionner. Cliquez sur Generate token.
- f. Copiez le Token (Attention cette action n'est disponible qu'une seule fois)

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_T2cphSB65AK5EarGK e0PUWDZ  Delete

Une fois que vous avez un jeton, vous pouvez l'utiliser à la place de votre mot de passe lorsque vous effectuez des opérations git push sur HTTPS comme ci-dessous :

```
git push
https://<GITHUB_ACCESS_TOKEN>@github.com/<GITHUB_USERNAME>/<REPOSITORY_NAME>.git
```

Remarque : n'oubliez pas de remplacer <GITHUB_ACCESS_TOKEN>, <GITHUB_USERNAME>, <REPOSITORY_NAME> par votre jeton, votre nom d'utilisateur et votre référentiel.

OU

```
git remote set-url origin https://eliesjebri:<TOKEN>@github.com/eliesjebri/test-demo.git
pour ajouter de façon permanente ce token à votre fichier de configuration .git/config
```

5- Pousser son travail sur un dépôt distant

Lorsque votre dépôt vous semble prêt à être partagé, il faut le pousser en amont. La commande pour le faire est simple : git push [nom-distant] [nom-de-branche].

```
[user@localhost ~/LabsGit/fetch-test] (main)$ cd ../test-demo/
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git remote -v
origin https://github.com/eliesjebri/test-demo.git (fetch)
origin https://github.com/eliesjebri/test-demo.git (push)
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ ls
README.md
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ vim README.md
# test-demo
Repo GitHub Demo Test
New line from GitHub WebGUI.
New line from locat git repo.
```



```
[user@localhost ~/LabsGit/test-demo] (main)$ git status
# On branch main
# 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:   README.md
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git add -A
[user@localhost ~/LabsGit/test-demo] (main)$ git commit -m "Add new line from
local repo to README.md"
[main 5bb78a9] Add new line from local repo to README.md
1 file changed, 1 insertion(+)
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git log --oneline --decorate --
graph
* 5bb78a9 (HEAD, main) Add new line from local repo to README.md
* 7e94f46 (origin/main, origin/HEAD) Update README.md
* 8ea6474 Initial commit
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git push
warning: push.default is unset; its implicit value is changing in
Git 2.0 from 'matching' to 'simple'. To squelch this message
and maintain the current behavior after the default changes, use:

    git config --global push.default matching

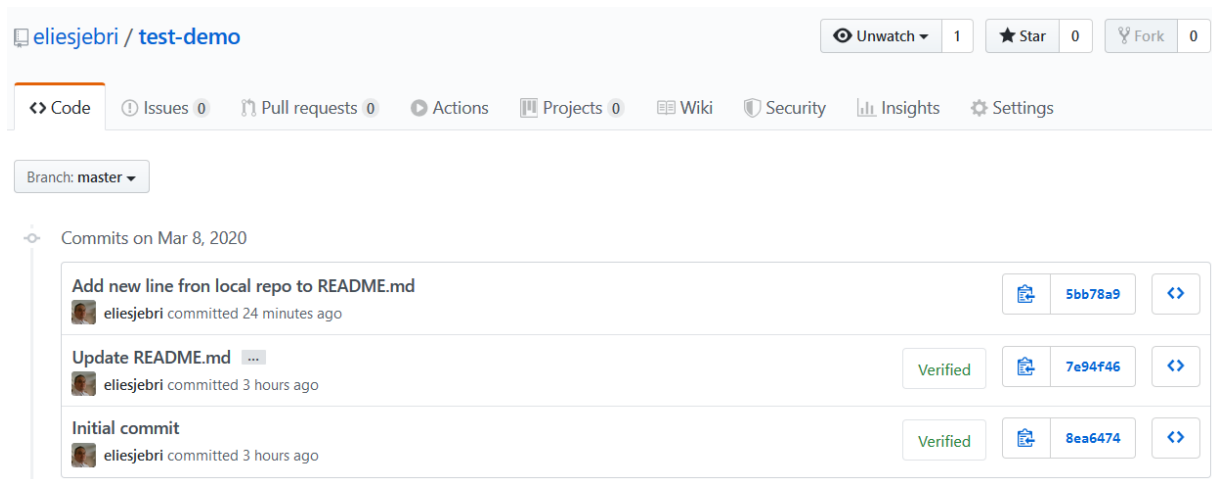
To squelch this message and adopt the new behavior now, use:

    git config --global push.default simple

See 'git help config' and search for 'push.default' for further information.
(the 'simple' mode was introduced in Git 1.7.11. Use the similar mode
'current' instead of 'simple' if you sometimes use older versions of Git)

Username for 'https://github.com': elies.jebri@gmail.com
Password for 'https://elies.jebri@gmail.com@github.com':
Counting objects: 5, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 340 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/eliesjebri/test-demo.git
7e94f46..5bb78a9  main -> main
```

Vérifiez sur GitHub que le dernier push y est bien



6- Travailler avec les branches à distance

Vous allez créer une branche en local puis faire un push sur GitHub

```
[user@localhost ~/LabsGit/test-demo] (main)$ git branch feature-1
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git checkout feature-1
Switched to branch 'feature-1'
```

```
[user@localhost ~/LabsGit/test-demo] (feature-1)$ echo "Feature-1 content" >
feature1.txt
```

```
[user@localhost ~/LabsGit/test-demo] (feature-1)$ git status
# On branch feature-1
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       feature1.txt
nothing added to commit but untracked files present (use "git add" to track)
```

```
[user@localhost ~/LabsGit/test-demo] (feature-1)$ git add -A
```

```
[user@localhost ~/LabsGit/test-demo] (feature-1)$ git commit -m "Add Branch
Feature-1 and file feature1.txt"
[feature-1 da188b7] Add Branch Feature-1 and file feature1.txt
1 file changed, 1 insertion(+)
create mode 100644 feature1.txt
```

```
[user@localhost ~/LabsGit/test-demo] (feature-1)$ git log --oneline --
decorate --graph
da188b7 (HEAD, feature-1) Add Branch Feature-1 and file feature1.txt
991804b (origin/main, origin/HEAD, main) Add file1.txt locally
```

```
5bb78a9 Add new line from local repo to README.md
7e94f46 Update README.md
8ea6474 Initial commit
```

```
[user@localhost ~/LabsGit/test-demo](feature-1)$ git checkout main
```

```
[user@localhost ~/LabsGit/test-demo](main)$ git config credential.helper
store
```

```
[user@localhost ~/LabsGit/test-demo](main)$ git config --global
credential.helper 'cache --timeout 7200'
```

```
[user@localhost ~/LabsGit/test-demo](main)$ git push origin feature-1
Username for 'https://github.com': elies.jebri@gmail.com
Password for 'https://elies.jebri@gmail.com@github.com':
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 348 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature-1' on GitHub by visiting:
remote:   https://github.com/eliesjebri/test-demo/pull/new/feature-1
remote:
To https://github.com/eliesjebri/test-demo.git
 * [new branch]      feature-1 -> feature-1
```

7- Créer une demande d'extraction sur GitHub PULL REQUEST

Les projets open source hébergés dans des référentiels publics bénéficient des contributions apportées par la communauté des développeurs au sens large par le biais de demandes d'extraction, qui demandent qu'un projet accepte les modifications que vous avez apportées à son référentiel de code.

Le répo en question sera test-demo, vous devrez soit créer un autre compte GitHub soit travailler en binome.

Introduisez tout d'abord une erreur qui fera l'objet d'un fix puis d'un pull request.

```
[user@localhost ~/LabsGit/test-demo](main)$ vim README.md
# test-demo
Repo GitHub Demo Test
New line from GitHub WebGUI.
New line from local git repo.
ERROR
```

```
[user@localhost ~/LabsGit/test-demo](main)$ git branch
feature-1
* main
```

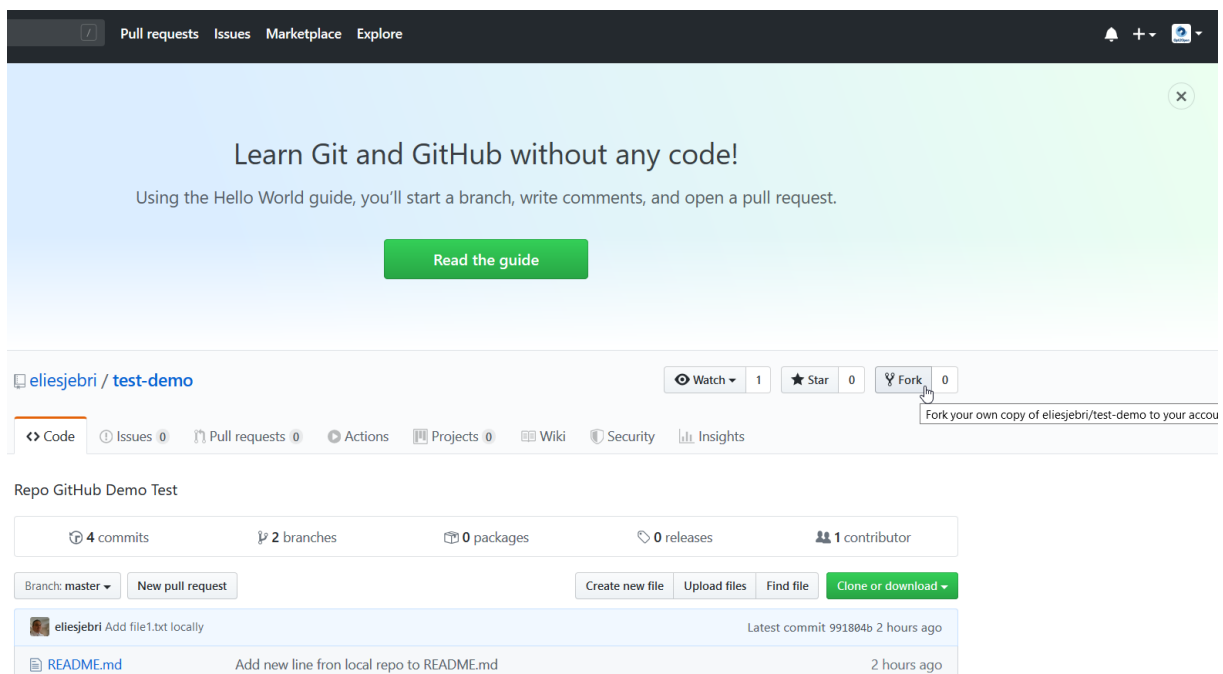
```
[user@localhost ~/LabsGit/test-demo] (main)$ git add -A
```

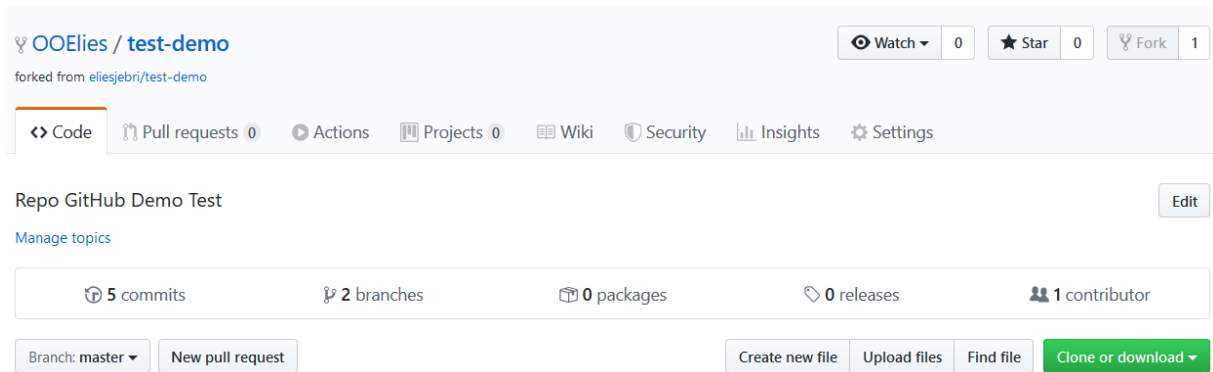
```
[user@localhost ~/LabsGit/test-demo] (main)$ git commit -m "Add Error"
[main 812f85c] Add Error
1 file changed, 1 insertion(+)
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ git push origin main
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 307 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/eliesjebri/test-demo.git
991804b..812f85c  main -> main
```

A. Fork du référentiel

A l'aide d'un autre compte (votre voisin) allez sur la page principale du référentiel test-demo, vous verrez un bouton "Fork" dans le coin supérieur droit de la page :





B. Clonez le référentiel

Pour créer votre propre copie locale du référentiel auquel vous souhaitez contribuer, utilisez `git clone`.

Mais tout d'abord créez un autre compte utilisateur chez qui vous clonerez le référentiel et qui se chargera de créer le correctif (fix).

```
[user@localhost ~]$ sudo useradd elies
[user@localhost ~]$ sudo passwd elies
[user@localhost ~]$ su - elies
```

C. Créer une nouvelle branche

Chaque fois que vous travaillez sur un projet collaboratif, vous et les autres programmeurs contribuant au référentiel aurez des idées différentes pour de nouvelles fonctionnalités ou correctifs à la fois. Pour cette raison, il est important de créer une branche du référentiel afin de pouvoir gérer le flux de travail, isoler votre code et contrôler les fonctionnalités qui le ramènent à la branche principale du référentiel de projet.

```
[elies@localhost ~]$ git clone https://github.com/OOElies/test-demo.git
test-demo-fix
```

```
[elies@localhost ~/test-demo-fix] (main)$ git log
commit 812f85c7d581dbbdd287e95dd8b8cb0d5416cd51
Author: elies_jebri <elies.jebri@gmail.com>
Date: Sun Mar 8 17:08:11 2020 +0100
```

Add Error

...

```
[elies@localhost ~/test-demo-fix] (main)$ git config --global user.email
elies.jebri@opt2open.com
[elies@localhost ~/test-demo-fix] (main)$ git config --global user.name
"OOElies"
```

```
[elies@localhost ~/test-demo-fix] (main)$ git branch
* main
```

```
[elies@localhost ~/test-demo-fix] (main)$ cat README.md
# test-demo
Repo GitHub Demo Test
New line from GitHub WebGUI.
New line from locat git repo.
ERROR
```

```
[elies@localhost ~/test-demo-fix] (main)$ git branch fix-typo
```

```
[elies@localhost ~/test-demo-fix] (main)$ git checkout fix-typo
Switched to branch 'fix-typo'
```

D. Apportez des modifications localement

Une fois que vous avez modifié des fichiers existants ou ajouté de nouveaux fichiers au projet, vous pouvez les ajouter à votre référentiel local, ce que nous pouvons faire avec la commande `git add` puis `git commit`.

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ vim README.md
# test-demo
Repo GitHub Demo Test
New line from GitHub WebGUI.
New line from locat git repo.
Fixed ERROR
```

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git add -A
[elies@localhost ~/test-demo-fix] (fix-typo)$ git commit -m "Fix error in README.md"
```

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git log
commit ale1392cb4832020e9234549be674f72ead57358
Author: OOElies <elies.jebri@opt2open.com>
Date: Sun Mar 8 17:18:56 2020 +0100

    Fix error in README.md
...
```

À ce stade, vous pouvez utiliser la commande `git push` pour envoyer les modifications à la branche actuelle de votre référentiel forké:

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git push origin fix-typo
Username for 'https://github.com': elies.jebri@opt2open.com
Password for 'https://elies.jebri@opt2open.com@github.com':
Counting objects: 5, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 328 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
```

```
remote: Create a pull request for 'fix-typo' on GitHub by visiting:
remote:      https://github.com/OOElies/test-demo/pull/new/fix-typo
remote:
To https://github.com/OOElies/test-demo.git
* [new branch]      fix-typo -> fix-typo
```

À ce stade, il est possible de faire une demande d'extraction vers le référentiel d'origine, mais si vous ne l'avez pas déjà fait, vous voudrez vous assurer que votre référentiel local est à jour avec le référentiel en amont.

E. Mettre à jour le référentiel local

Pendant que vous travaillez sur un projet aux côtés d'autres contributeurs, il est important que vous gardiez votre référentiel local à jour avec le projet car vous ne voulez pas faire de demande de pull pour du code qui causera des conflits. Pour garder à jour votre copie locale de la base de code, vous devrez synchroniser les modifications.

Afin de pouvoir synchroniser les modifications que vous effectuez dans un fork avec le référentiel d'origine avec lequel vous travaillez, vous devez configurer un remote qui pointe le référentiel en amont. Vous ne devez configurer le remote sur le référentiel en amont qu'une seule fois.

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git remote -v
origin https://github.com/OOElies/test-demo.git (fetch)
origin https://github.com/OOElies/test-demo.git (push)
```

Ensuite, nous allons spécifier un nouveau référentiel distant en amont (upstream) pour synchroniser avec le fork. Ce sera le référentiel d'origine à partir duquel nous nous sommes bifurqués. Nous allons le faire avec la commande git remote add.

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git remote add upstream
https://github.com/eliesjebri/test-demo.git
[elies@localhost ~/test-demo-fix] (fix-typo)$ git remote -v
origin https://github.com/OOElies/test-demo.git (fetch)
origin https://github.com/OOElies/test-demo.git (push)
upstream https://github.com/eliesjebri/test-demo.git (fetch)
upstream https://github.com/eliesjebri/test-demo.git (push)
```

F. Synchronisez le fork

Une fois que nous avons configuré un remote vers le référentiel upstream sur GitHub, nous sommes prêts à synchroniser notre fork pour le maintenir à jour.

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git fetch upstream
From https://github.com/eliesjebri/test-demo
* [new branch]      feature-1 -> upstream/feature-1
* [new branch]      main -> upstream/main
```

Passons à la branche maître locale de notre référentiel:

```
[elies@localhost ~/test-demo-fix] (fix-typo)$ git checkout main
Switched to branch 'main'
```

Nous allons maintenant fusionner toutes les modifications apportées à la branche principale du référentiel d'origine, auxquelles nous accéderons via notre branche locale upstream/main, avec notre branche principale locale:

```
[elies@localhost ~/test-demo-fix] (main)$ git merge upstream/main
Already up-to-date.
```

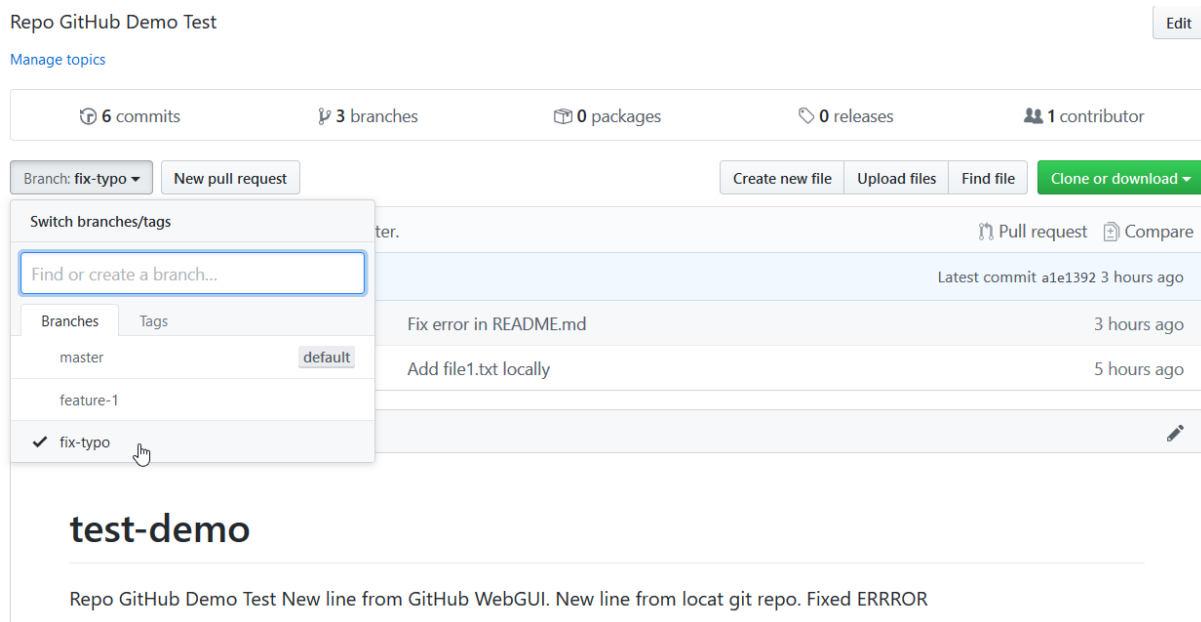
La sortie ici sera Already up-to-date si aucune modification n'a été apportée depuis que vous avez forké le référentiel.

En fonction de votre propre flux de travail et du temps que vous passez à apporter des modifications, vous pouvez synchroniser votre fork avec le référentiel upstream d'origine autant de fois que cela vous convient. Mais vous devez certainement synchroniser votre fork juste avant de faire une pull request pour vous assurer que vous ne contribuez pas au code en conflit.

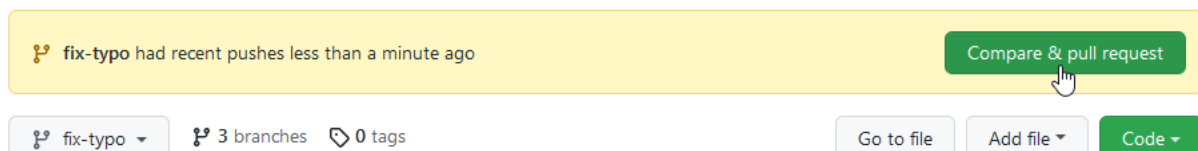
G. Créer un Pull Request

À ce stade, vous êtes prêt à effectuer une demande d'extraction (pull request) vers le référentiel d'origine.

Commencez par sélectionner la branche du fix :





Vous devez accéder à votre référentiel forké et appuyer sur le bouton «New Pull request» sur le côté gauche de la page.






H. Valider un Pull Request



Fix error in README.md #1

 **OOElies** wants to merge 1 commit into `eliesjebri:master` from `OOElies:fix-typo` 


Conversation 0 Commits 1 Checks 0 Files changed 1


 **OOElies** commented now  

Fixed Error in file README.md


  Fix error in README.md a1e1392



Add more commits by pushing to the **fix-typo** branch on `OOElies/test-demo`.

 **This branch has no conflicts with the base branch**
Only those with [write access](#) to this repository can merge pull requests.


 **Write** **Preview** AA B i “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

 **Close pull request**  **Comment**


Sur le repo d'origine, test-demo vous allez trouver une notification de « Pull Request »



 `eliesjebri / test-demo` 👁 Unwatch 1 ★ Star 0 🍴 Fork 1


[Code](#) [Issues 0](#) [Pull requests 1](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Label issues and pull requests for new contributors [Dismiss](#)

Now, GitHub will help potential first-time contributors [discover issues](#) labeled with `good first issue`

Filters is:  **Fix error in README.md #1**
Fixed Error in file README.md



 1 Open Labels 9 Milestones 0  **New pull request**

 **Fix error in README.md**
#1 opened 4 minutes ago by `OOElies`


Projects Milestones Reviews Assignee Sort

Avant de valider cette requête vous pouvez vérifier les fichiers qui vont changer, cliquez sur « Files Changed » :



Fix error in README.md #1

 **Open** OOElies wants to merge 1 commit into `eliesjebri:master` from `OOElies:fix-typo` 



Conversation 0 Commits 1 Checks 0 Files changed 1


 OOElies commented 7 minutes ago First-time contributor + 😊 ...

Fixed Error in file README.md

  Fix error in README.md a1e1392

Add more commits by pushing to the **fix-typo** branch on OOElies/test-demo.



  **Continuous integration has not been set up**
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).


Fix error in README.md #1

[Edit](#)

 **Open** OOElies wants to merge 1 commit into `eliesjebri:master` from `OOElies:fix-typo` 

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -1






Changes from all commits File filter... Jump to... 0 / 1 files viewed [Review changes](#)

File	Changes
README.md	 -2,4 +2,4
2	2 Repo GitHub Demo Test
3	3 New line from GitHub WebGUI.
4	4 New line from locat git repo.
5	- ERRORR
5	+ Fixed ERRORR

Si vous approuvez ou rejeter les changements alors cliquez sur « Review changes »

Fix error in README.md #1



Edit

 OOElies wants to merge 1 commit into `eliesjebri:master` from `OOElies:fix-typo`  Conversation 0 Commits 1 Checks 0 Files changed 1+1 -1 

Changes from all commits ▾ File filter... ▾ Jump to... ▾ ⚙ ▾

0 / 1 files viewed ⓘ

Review changes ▾


2  README.md 

2	2	Repo GitHub Demo Test
3	3	New line from GitHub WebGU
4	4	New line from locat git re
5	-	ERROR
5	+	Fixed ERROR

Write Preview

AA B i “ < > ↺ ⋮ ⋮ ⋮ @ 📌 ↶

Approved For Merge

Attach files by dragging & dropping, selecting or pasting them. 

☐ Comment

Submit general feedback without explicit approval.

☒ Approve


Submit feedback and approve merging these changes.


☐ Request changes

Submit feedback that must be addressed before merging.


Submit review


Add more commits by pushing to the **fix-typo** branch on OOElies/test-demo.




 **Changes approved** [Show all reviewers](#)

1 approving review [Learn more](#).

 **1 approval** ▾


 **Continuous integration has not been set up**
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**
Merging can be performed automatically.

Merge pull request ▾

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add more commits by pushing to the **fix-typo** branch on OOElies/test-demo.



Merge pull request #1 from OOElies/fix-typo

Fix error in README.md

Confirm merge

Cancel

Vérifiez que les branches « fix-typo » et « main » ont fusionnées et que la modification est appliquée.

Repo GitHub Demo Test

[Manage topics](#)

7 commits

2 branches

0 packages

Branch: master ▼

New pull request



eliesjebri Merge pull request #1 from OOElies/fix-typo ...



README.md

Fix error in README.md



file1.txt

Add file1.txt locally

I. Récupérez les modifications de GitHub sur votre ordinateur

En ce moment, le dépôt sur GitHub est un peu différent de ce que vous avez sur votre machine locale. Par exemple, la validation que vous avez effectuée dans votre branche et fusionnée dans la branche main n'existe pas dans la branche main sur votre machine locale.

Afin d'obtenir les modifications les plus récentes que vous ou d'autres avez fusionnées sur GitHub, utilisez la commande `git pull origin main` (lorsque vous travaillez sur la branche principale).

```
[user@localhost ~/LabsGit/test-demo] (main)$ git pull origin main
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/eliesjebri/test-demo
* branch          main      -> FETCH_HEAD
Updating 812f85c..7bd3240
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
[3]+  Fini                  gitk    (wd : ~/website)
(maintenant, wd : ~/LabsGit/test-demo)
```

```
[user@localhost ~/LabsGit/test-demo] (main)$ cat README.md
# test-demo
Repo GitHub Demo Test
New line from GitHub WebGUI.
New line from locat git repo.
Fixed ERROR
```