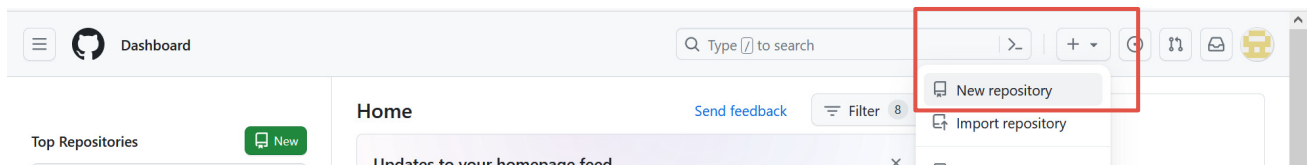


Tutorial 1 : Git et GitHub

NB : ce tutoriel est inspiré du site <https://openclassrooms.com/fr/courses/7162856-gerez-du-code-avec-git-et-github>

Activité 1 : configurer Git et GitHub

1. Accéder à <https://github.com/> et créer un nouveau compte (veuillez-vous rappeler du username et de l'email utilisés)
2. Créer un nouveau repository (ou dépôt), c'est un dossier distant dans lequel toutes vos modifications seront enregistrés. Nommé ce repository « tpgit », le choisir comme public.



3. Télécharger et installer Git sur votre machine (aucune configuration spécifique n'est nécessaire)
4. Configurer Git sur votre Pc avec votre identité (cette étape est réalisée une fois seulement), vous devez vous rappeler du nom d'utilisateur et du mot de passe. Lancer Git bash ou même une Cmd et lancer les commandes suivantes (**remplacer xx par son username et yy par votre email**).

```
git config --global user.name "xx"  
git config --global user.email "yy"
```

Activité 2 : Premiers pas avec git

Pour travailler sur Git, vous devez créer un dépôt local, c'est-à-dire un dossier dans votre machine dans lequel toutes vos modifications seront enregistrées. C'est ce qu'on a

appelle **initialiser un dépôt Git**.

1. Créer un dossier **tpgit** sur votre machine. Placez vous dans ce fichier à l'aide de la cmd ou gitbash (utiliser la commande **cd**).
2. Initialiser ce dossier comme dépôt git local avec la commande : **git init**

3. Remarquer la création d'un dossier caché (.git) qui contient tous les éléments non visibles de Git (la configuration, les "logs", les branches...)
4. Maintenant, nous allons ajouter des fichiers : créer sous le dossier tpgit deux fichiers : index.html et style.css

index.html	style.css
<pre><!DOCTYPE html> <html> <head> <link rel="stylesheet" type="text/css" href="styles.css"> </head> <body> <h1>Git et Github</h1> </body> </html></pre>	<pre>h1 { color: red; }</pre>

5. Vous avez créé 2 fichiers, index.html et styles.css. Ces fichiers sont situés dans le dépôt Git. Pour créer une nouvelle version de votre projet, vous allez maintenant indexer les fichiers avec la commande : `git add index.html style.ss`
6. Maintenant que vos fichiers modifiés sont indexés, vous pouvez créer une version, **c'est-à-dire archiver le projet en l'état**. Pour ce faire, utilisez la commande "git commit" en indiquant un message relatif aux changements effectués : `git commit -m "ajout fichiers index et style"`. Vous avez maintenant créé une version de votre projet.
7. Exécuter la commande `gitk` pour visualiser les différentes versions de votre projet



Activité 3 : Envoyer votre commit sur le dépôt distant avec le protocole SSH

Après avoir créé une version locale de votre projet, vous devez désormais passer votre commit de à votre dépôt distant. Pour se faire, vous devez vous connecter au dépôt distant :

- **En utilisant le protocole Https** : On vous demandera une confirmation de votre identifiant et mot de passe pour chaque action sensible.
- **En utilisant le protocole SSH (recommandé)** : Le protocole SSH permet de vous authentifier sur GitHub sans confirmer votre mot de passe ou code de confirmation pour chaque action. Il permet aussi de vous identifier lorsque vous faites un commit.

1. Créer votre Clé SSH (veuillez suivre les étapes décrites dans cette adresse : <https://www.atlassian.com/fr/git/tutorials/git-ssh>).

a. Sous git bash, exécuter la commande (en remplaçant par votre email) :

```
ssh-keygen -t rsa -b 4096 -C "badismouna@yahoo.fr"
```

b. Suivre le reste des étapes en cliquant sur la touche Entrer. Ainsi, la clé semble à ceci :

```
$ ssh-keygen -t rsa -b 4096 -C "badismouna@yahoo.fr"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Hassene/.ssh/id_rsa):
Created directory '/c/Users/Hassene/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Hassene/.ssh/id_rsa
Your public key has been saved in /c/Users/Hassene/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:bm8z2gdpFcSHceWL5uP8j5NLHQLkfjMM//AycXVFNQ badismouna@yahoo.fr
The key's randomart image is:
+---[RSA 4096]---+
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
|                 |
+---+-----+

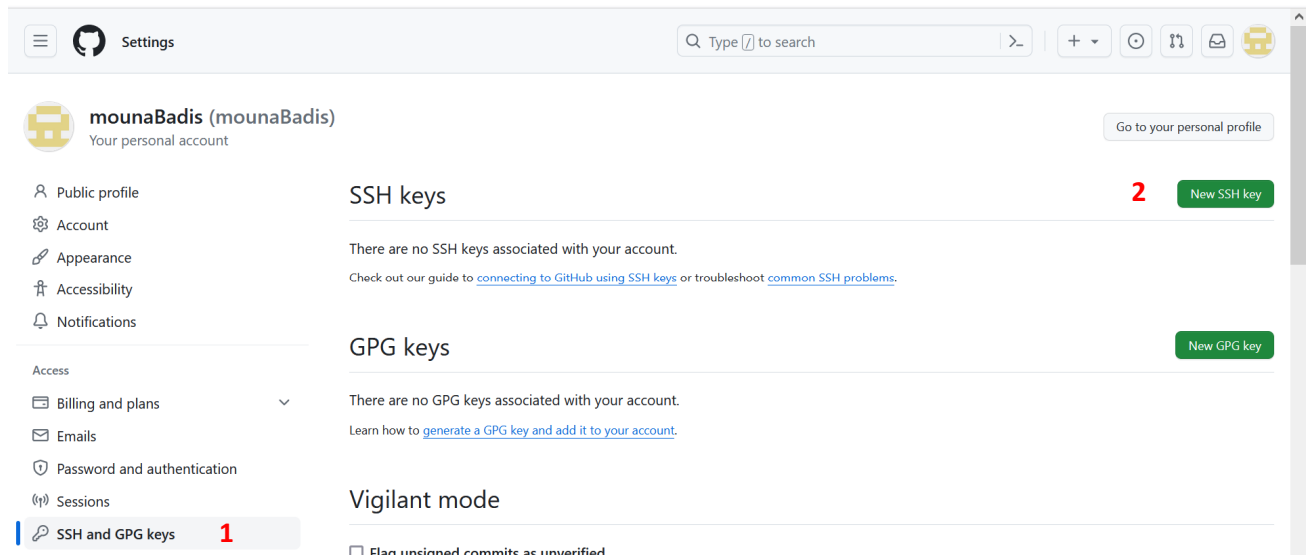
```

chemin où
sauvegardée la
clé publique

c. Afficher la clé publique avec la commande **cat** suivi par le chemin de sauvegarde de la clé publique. Copier le résultat.

2. Rendez-vous sur votre compte GitHub, clic sur l'image de votre profil, puis settings.

3. Dans vos paramètres, sélectionnez SSH and GPG Keys



4. Donner le titre tpgit à la clé et coller la clé public déjà copier dans sa zone.

Add new SSH Key

Title

tpgit

Key type

Authentication Key ↕

Key

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Copier la clé publique
générée


Add SSH key

- Voilà, maintenant vous pouvez faire le commit de votre dépôt local vers le dépôt distant. Pour relier votre projet local au dépôt github, exécuter les commandes suivantes (**en modifiant selon les données de votre projet**):
`git remote add origin git@github.com:mounaBadis/tpgit.git`
- Créer une branche principale main (copie principale du code source du projet) : `git branch -M main`
- Chaque fois que vous voulez faire le commit du dépôt local au dépôt distant, utiliser la commande push comme suit : `git push -u origin main`
- Vérifier que les nouveaux fichiers apparaissent sous GitHub
- Revenir à votre machine, modifier les fichiers index.html et style.css. Faire le nécessaire pour sauvegarder les nouvelles versions sous GitHub. Essayer de visualiser l'historique des modifications effectués.


Activité 4 : Apprendre à utiliser les branches Git




Une branche est une copie du projet main principal. Elle sera utilisée pour y ajouter des modifications dans le code.

- Lister toutes branches (les copies) de votre repository local : `git branch`
- Créer maintenant une branche nommée backend : `git branch backend`
- Basculer vers cette branche : `git checkout backend`
- Ajouter maintenant à votre dossier de travail un fichier `add.php`
- Valider le fichier ajouté : `git add add.php` et `git commit -m "realisation de add"`
- Faire le commit de backend vers le dépôt distant : `git push -u origin backend`
- Vérifier que cette branche a été créée sous GitHub

← → ↻  <https://github.com/mounaBadis/tpgit/tree/backend> 5

This branch is 1 commit ahead of [main](#) . [Contribute](#) ▾

 **mounaBadis** realisation de add c00db48 · 2 minutes ago 🕒 4 Commits

 add.php	realisation de add	2 minutes ago
 index.html	modification des fichiers	1 hour ago
 style.css	modification des fichiers	1 hour ago

8. Fusionner le projet principal avec le projet backend
 - a. se positionner sous la branche main : **git checkout main**
 - b. fusionner main avec backend : **git merge backend**
 - c. faire le commit distant : **git push -u origin main**