



Vous utiliserez Git pour gérer les versions de vos fichiers et pour publier votre code sur GitHub.

Git et GitHub sont très sécurisés :

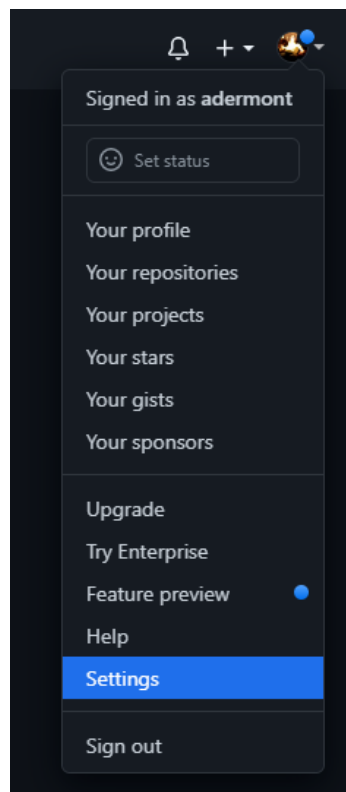
- **SSH** (Secure Shell) est utilisé pour vous authentifier et ouvrir une **connexion sécurisée** à votre compte GitHub.
- **GnuPG** est utilisé pour le **chiffrement des données** publiées sur GitHub.
- **GitHub** propose enfin de **masquer votre adresse email principale** pour qu'elle n'apparaisse pas dans l'historique de vos fichiers.

Vous allez donc devoir réaliser la configuration nécessaire au fonctionnement de SSH et GPG.

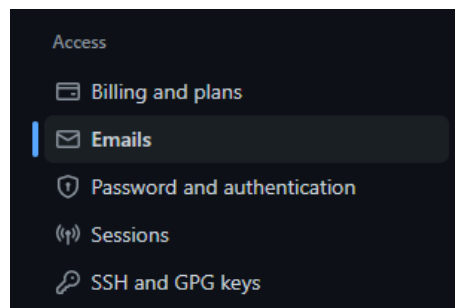
Pré-requis : installer GitHub Desktop

Vérification de l'email associé à votre compte GitHub

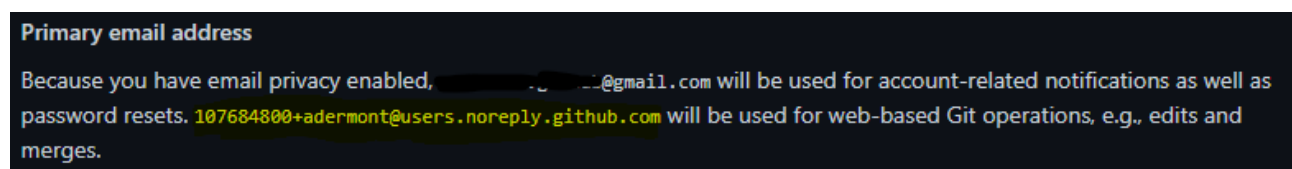
Allez sur la page "Settings" de votre compte GitHub :



Cliquez sur le menu **Emails** dans la barre latérale gauche :



Notez l'adresse email qui a été générée par GitHub pour protéger votre identité :



Vérifiez que vous avez coché les cases suivantes :

☒ Keep my email addresses private

☒ Block command line pushes that exposes my email

☒ **Keep my email addresses private**

We'll remove your public profile email and use 107684800+adermont@users.noreply.github.com when performing web-based Git operations (e.g. edits and merges) and sending email on your behalf. If you want command line Git operations to use your private email you must [set your email in Git](#).

Commits pushed to GitHub using this email will still be associated with your account.

☒ **Block command line pushes that expose my email**

When you push to GitHub, we'll check the most recent commit. If the author email on that commit is a private email on your GitHub account, we will block the push and warn you about exposing your private email.

Génération d'une nouvelle clé SSH

Suivez les étapes suivantes :

1. Génération d'une nouvelle clé SSH :

<https://docs.github.com/fr/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#generating-a-new-ssh-key>

2. Ajout de votre clé SSH à ssh-agent :

<https://docs.github.com/fr/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent#adding-your-ssh-key-to-the-ssh-agent>

3. Testez votre connexion SSH :

<https://docs.github.com/fr/authentication/connecting-to-github-with-ssh/testing-your-ssh-connection>

Génération d'une nouvelle clé GPG

1. Téléchargez et installez GnuPG4Windows : <https://gpg4win.org/download.html>
2. Lancez un **Terminal** (PowerShell ou Invite de commande) et un terminal **GitBash**.
3. Suivez les étapes décrites ci-après **en utilisant l'adresse email notée précédemment** :
<https://docs.github.com/fr/authentication/managing-commit-signature-verification/generating-a-new-gpg-key>
4. Dans votre terminal tapez la commande suivante et notez les valeurs en **bleu** :

```
$ gpg --list-secret-keys --keyid-format=long
gpg: checking the trustdb
gpg: marginals needed: 3 completes needed: 1 trust model: pgp
gpg: depth: 0 valid: 2 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 2u
/c/Users/Votre NOM/.gnupg/pubring.kbx
-----
sec  rsa4096/4F43F1AB4CB5C1D8 2023-01-14 [SC]
      0D14G95A4EC17E0ADBE851584F73F1EE4080C1B3
uid          [ultimate] votrelogin (GitHub account) <votreemail@users.noreply.github.com>
ssb  rsa4096/57FBE6F89C6BC212 2023-01-14 [E]
```

5. Exécutez ces commandes en remplaçant les valeurs en jaune par les vôtres :

```
$ git config --global core.autocrlf true
$ git config --global user.name votrelogin
$ git config --global user.email votreemail@users.noreply.github.com
$ git config --global commit.gpgsign true
$ git config --global user.signingkey=57FBE6F89C6BC212
```