Fiche de cours : GIT (outil de versionning)

Permet de faciliter le travail collaboratif entre plusieurs développeur sur un même projet. L'intérêt est d'avoir un historique des modifications (appelée version) que l'on peut rétablir à tout moment.

Git permet de suivre les évolutions d'un projet en utilisant le principe de "branche".

Chaque branche correspond à une version de votre projet qui évolue de manière INDEPENDANTE ! Cela permet de pouvoir expérimenter des solutions sans impacter le code existant.

Pour commencer, nous travaillerons avec une seule branche afin de mieux comprendre ce que ce principe implique.

Good to know : La branche "principale" d'un projet et souvent appellée "master" ou "main". Cette branche doit être la plus saine possible !

En pratique

Création d'un dépot git

```
cd project
```

```
git init
```

Ajout des fichiers

2 solutions

Ajouter les fichiers individuellement :

```
git add nomfichier
```

Ajouter tous les fichiers du projet

```
git add .
```

Faire un commit

Un commit est comme une sorte de point d'ancrage qui permet au développeur de sauvegarder une version du code à un instant T avec une description de son travail. Un commit est identifé de manière UNIQUE via un hash qui lui est attribué.

```
git commit -a -m "Message du commit"
```

- -a pour add (on ajoute au suivi les nouveaux fichiers créés s'ils y en a)
- -m pour message (on décrit les modifications apportées au code)

Un commit doit toujours être le plus EXPLICITE possible tout en étant CONCIT dans sa description

ATTENTION : vous devez au préalable configurer votre git local afin qu'il puisse vous reconnaître avant de pouvoir réaliser un commit. Pour cela, tapez les commandes suivantes (une seule fois) :

```
git config --global user.name "username"
```

```
git config --global user.email "votreemail@email.fr"
```

Ajouter un dépôt distant (remote)

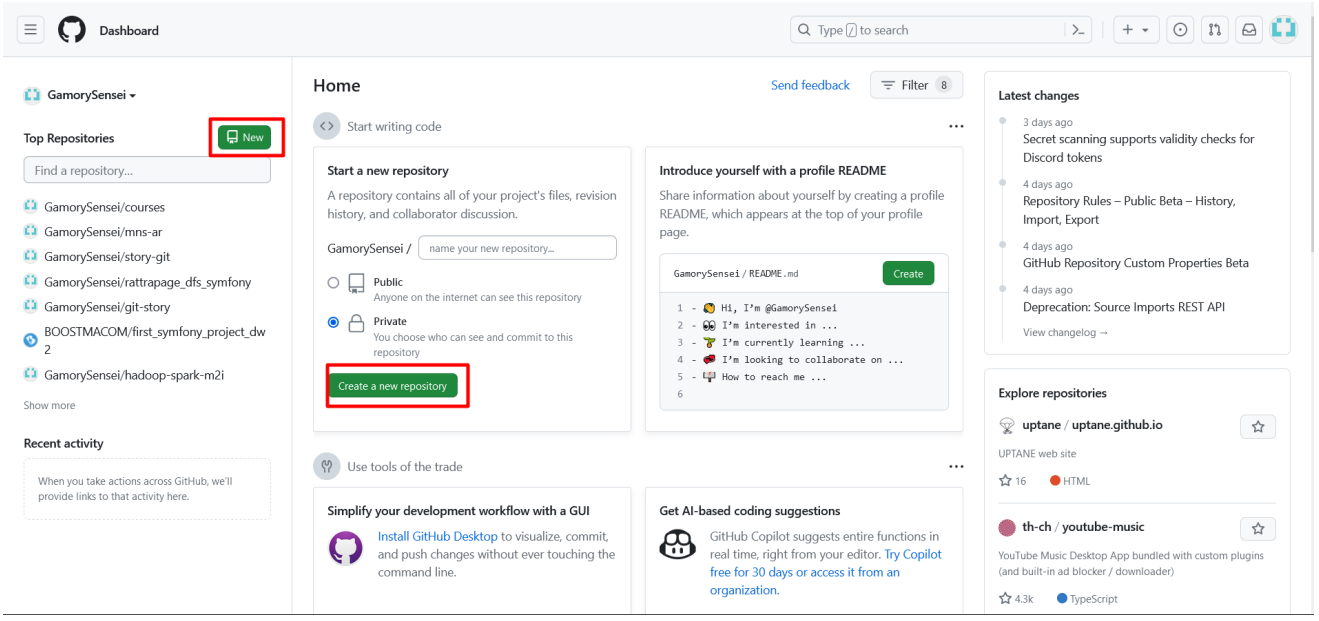
Lorsque l'on crée un dépôt LOCAL, on a souvent besoin de le synchroniser avec un dépôt DISTANT dans le but de pouvoir partager ensuite notre code avec d'autres dévs. Pour cela nous avons besoin d'avoir un compte chez un hébérgeur de dépôt et 3 choix s'offrent à nous :

1. GitHub
2. GitLab
3. BitBucket

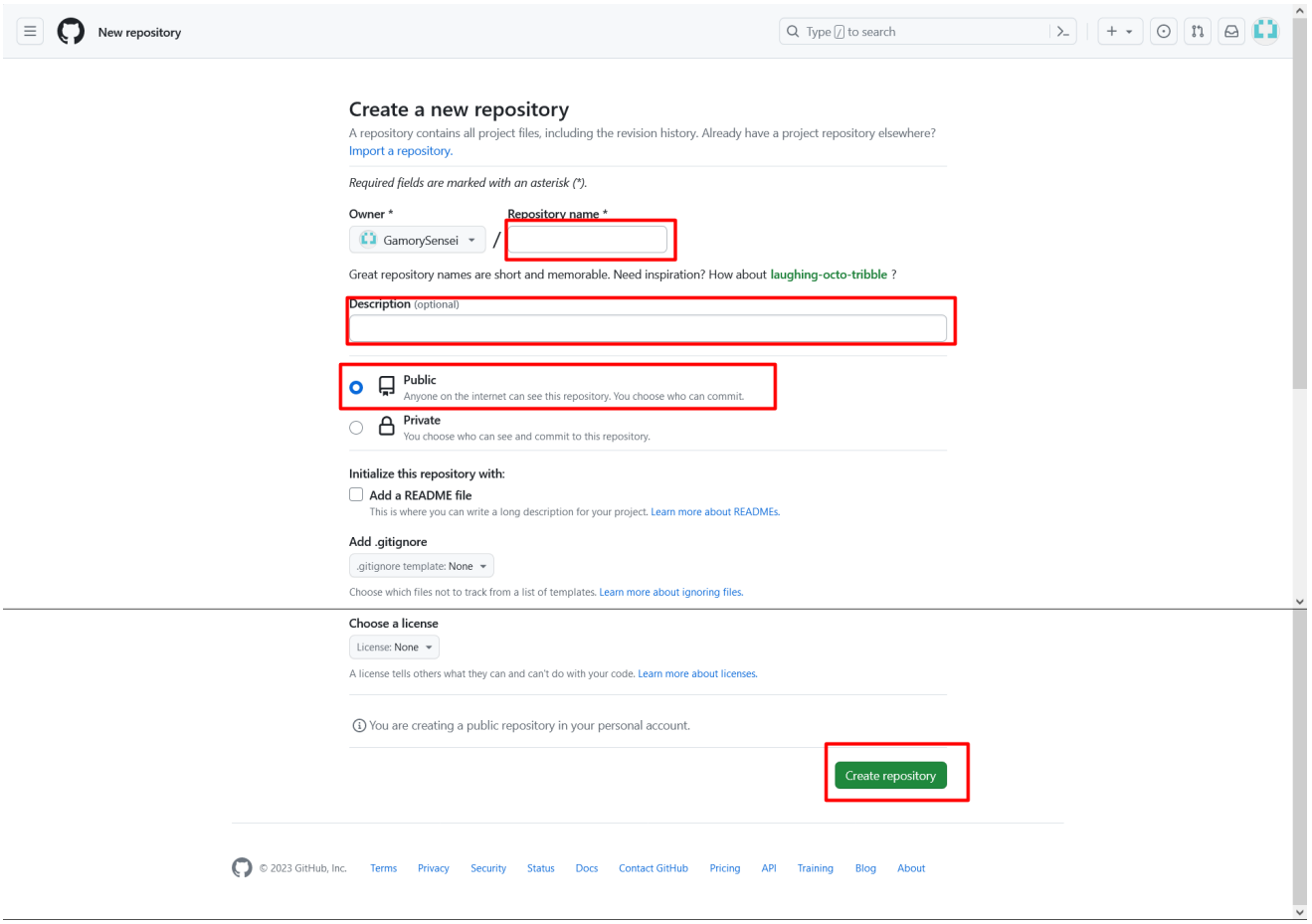
Nous allons partir sur GitHub car c'est le plus utilisé.

1. Créer un compte sur GitHub : https://github.com/signup

2. Créer un nouveau dépôt :

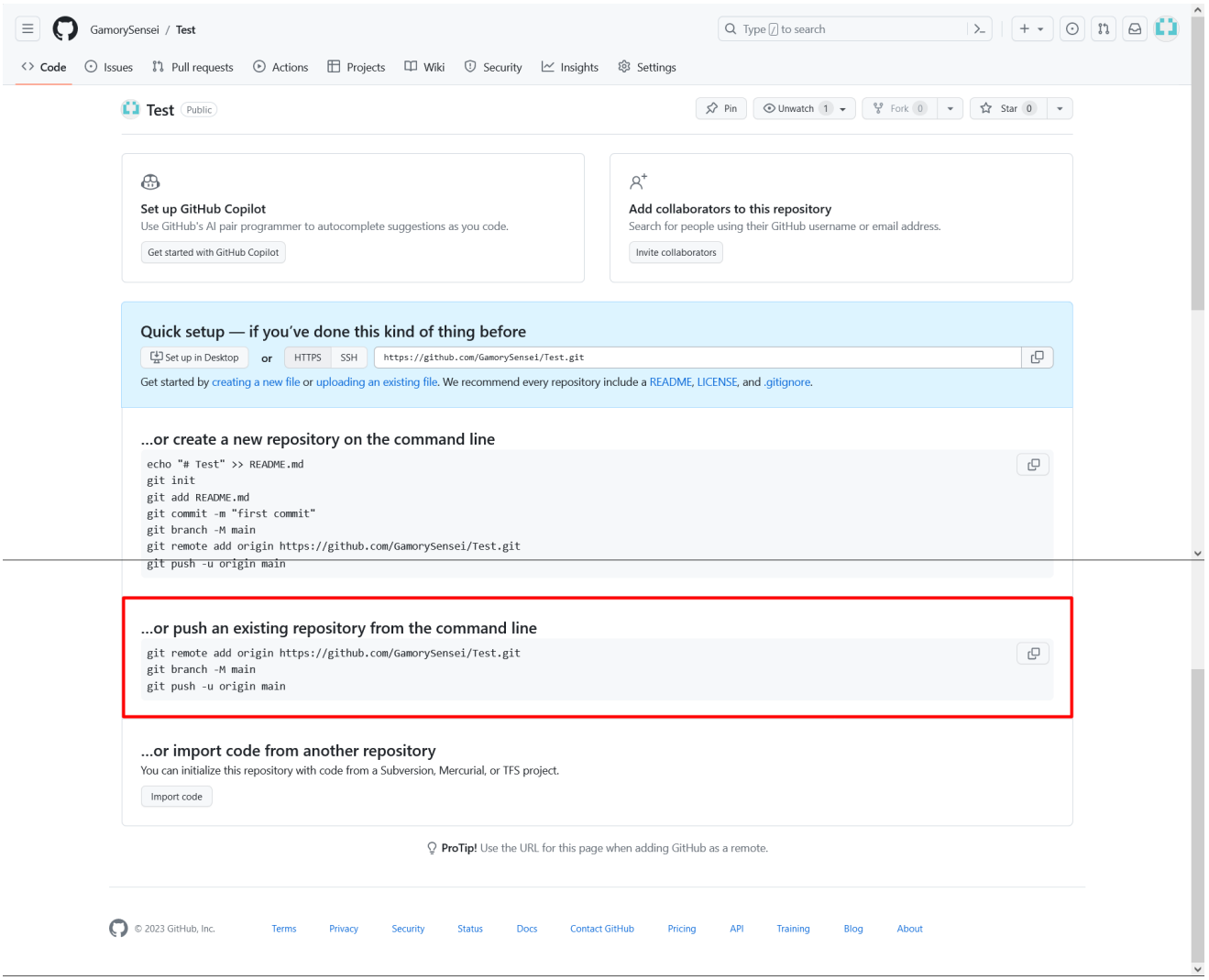


3. Saisir le nom du dépot (nom du projet en général) et la description



Laissez le en PUBLIC pour le moment puis "Create repository"

4. Suivez les commande en renommant la branche "main" en "master" si vous avez commencé par créer votre dépôt local :



Synchronisation des dépôt

Récupération du code du dépôt distant :

```
git pull
```

Envoi du code local vers le déppôt distant :

```
git push
```

ATTENTION :

Vous devez toujours effectuer un commit avant de pouvoir "push" ! Il est également recommandé de toujours "pull" avant de "push" afin d'éviter de "casser" le dépôt distant si vous êtes plusieurs à travailler sur un même branche.