

Utilisation de GitHub en équipe

E.ROMETTE & A.FALIGOT-GIRARDELLI

04/05/20

Contents

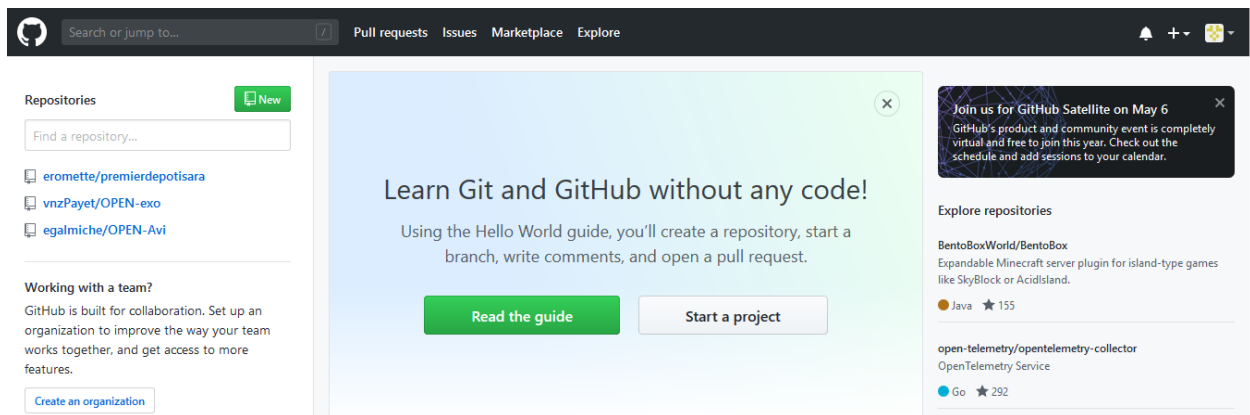
1	Initialisation du repository d'équipe	1
2	Ajouter les membres de l'équipe	4
3	Gestion de fichiers par la console	4
4	Gestion de fichiers par GitHub Desktop	5
5	Références	5



Résumé : Dans cette synthèse nous allons étudier la gestion de GitHub pour un travail d'équipe.

1 Initialisation du repository d'équipe

Une fois votre compte GutHub ouvert, allez au Menu principal du site où vous pouvez avoir une vision sur tous les Répertoires que vous avez créé ainsi que ceux que vous avez cloné.



Pour créer un nouveau Répertoire, cliquez sur le bouton “New” en vert. Cela vous amènera sur la page suivante :


Create a new repository

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

[Import a repository.](#)

Owner


Repository name *

 **eromette** /

Great repository names are short and memorable. Need inspiration? How about **redesigned-goggles?**

Description (optional)

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

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

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▼

Add a license: **None** ▼



Create repository

“**Owner**” indique le propriétaire du répertoire, le compte qui a tous les droits : l’administrateur.

“**Repository name**” correspond à l’emplacement où vous pouvez choisir le nom du répertoire. Il est conseillé de l’avoir en 1 seul mot (peut être composé de plusieurs mots mais attachés avec des caractères spéciaux comme & - _).

“**Description**” correspond à la description qui sera faite en haut de votre répertoire.

“**Public/Private**” permet de choisir si le répertoire est visible de tout ceux qui visitent le profil du créateur ou non. En cas de “Private”, il faut une autorisation d’accès.

“**README.Md**” correspond à un fichier en Markdown donnant les informations principales du répertoires. Il est souvent utilisé pour donner les consignes, des indications ou les notes de mises à jour.

“**Add .gitignore**” permet de créer un fichier `.gitignore` qui peut être édité textuellement. Ainsi, si le dossier contient :

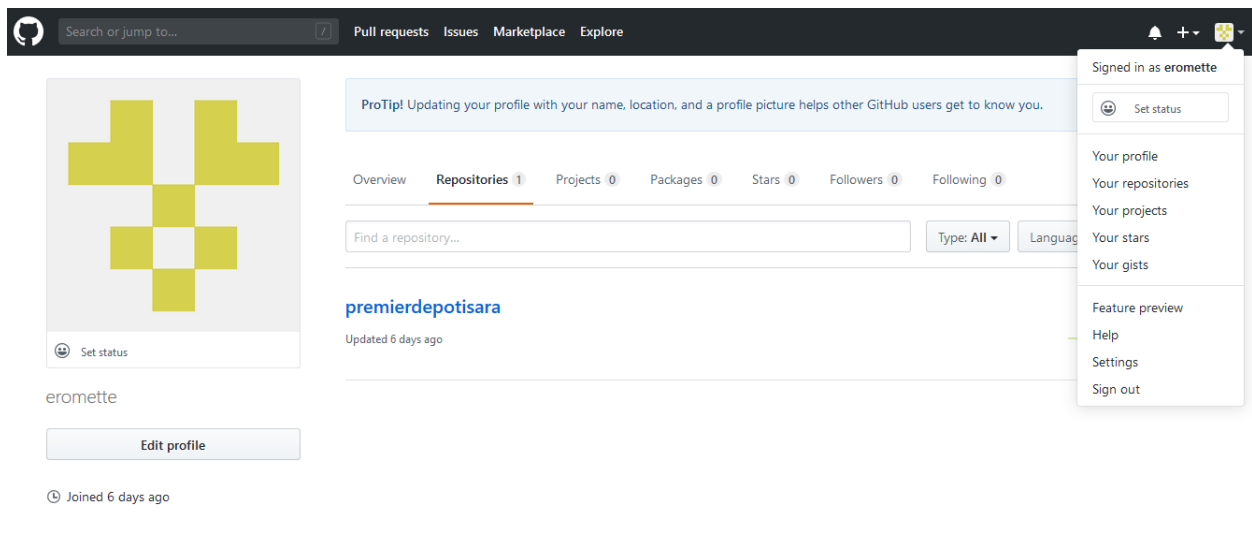
'*.txt' -> le fait de mettre une astérisque permet d’ignorer lors du push tous les fichiers se terminant par l’extension `.txt`.

'Projet1.docx' -> cette ligne permettra de ne pas synchroniser avec le répertoire en ligne le fichier appelé `Projet1.docx`.

Par contre, `*.docx` fera que lors du push, tous les fichiers `.docx` seront ignorés.

Cela est pratique lors de la création en commun de dossiers contenant des fichiers R car celui-ci crée automatiquement des fichiers `.Rhistory` et ainsi, écrire dans le `.gitignore` `.Rhistory` ignorera automatiquement tous les fichiers `.Rhistory`.

“**Add a license**” permet d’ajouter une license au répertoire, c’est à dire indiquer aux personnes, qui téléchargent/utilisent le répertoire et ce qu’il contient, qu’ils doivent respecter les conditions de la License (CC BY, ...).



Une fois votre répertoire créé (image ci-dessus), celui-ci est disponible dans votre onglet “Repositories” et vous pouvez le gérer à votre guise.

2 Ajouter les membres de l'équipe

Pour ajouter des collaborateurs, il faut aller dans notre dépôt/repository. Une fois dedans :

1. Aller dans *Settings*
2. Puis dans *Manage Access*
3. Cliquer sur *Invite a collaborator*
4. Entrer le nom de notre collaborateur
5. Lui envoyer une invitation

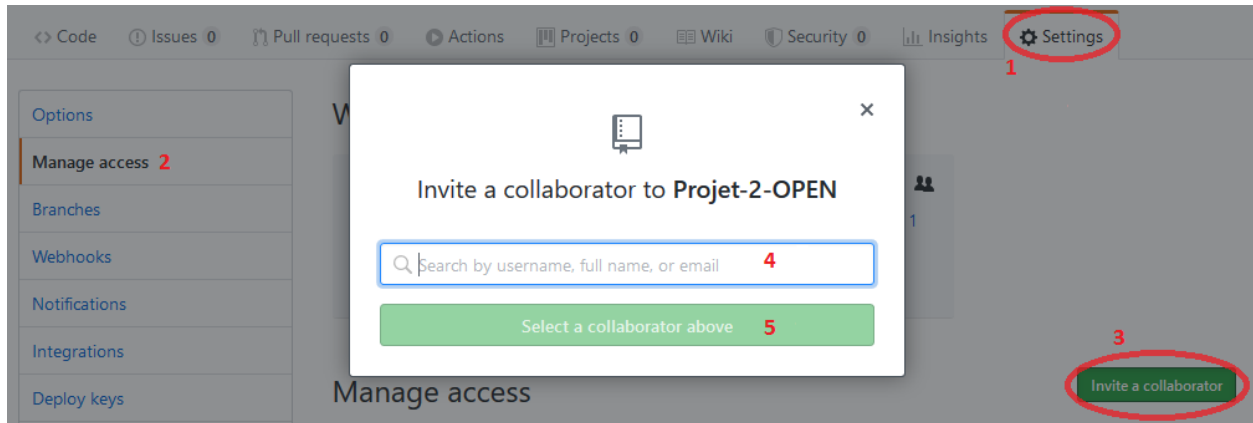


Figure 1: Ajout d'un collaborateur

Une fois que la personne aura acceptée l'invitation, elle pourra utiliser le dépôt librement.

3 Gestion de fichiers par la console

Pour gérer les fichiers présents dans son dépôt, on crée une copie locale de celui-ci. Pour cela on ouvre l'invite de commandes de notre ordinateur et on accède au lieu où l'on veut stocker notre dépôt par la commande `cd` :

Par exemple : pour accéder au fichier Groupe2 dans ses documents on ouvre son invite de commande et on tape `cd`

- `git clone 'adresse du dépôt'`

Une fois le dépôt cloné, on peut modifier les fichiers directement sur notre ordinateur.

Pour mettre à jour notre travail sur notre dépôt en ligne on commence par mettre à jour notre fichier local par un pull :

- `git pull`

Puis on fait un add, un commit et un push, pour le renvoyer sur git en ligne :

- `git add 'nom du projet ou du fichier'`
- `git commit -m "Commentaire"`

- `git push`

→ Pensez à faire entrer après chaque ligne de commande !

Normalement, si l'on vérifie en ligne on peut voir le fichier mis à jour.

4 Gestion de fichiers par GitHub Desktop

5 Références

<https://ndpsoftware.com/git-cheatsheet.html#loc=index;>

<https://education.github.com/git-cheat-sheet-education.pdf>

<https://desktop.github.com/>

<https://github.com/>