

GITHUB

F. Lemainguo
Novembre 2016

Introduction

Après GIT, examinons GITHUB. Lorsque vous travaillez sur un projet sur votre machine, il est important d'avoir une sauvegarde de votre code sur une autre machine, au cas où la vôtre tombe en panne par exemple. Pour ce faire, vous devez envoyer vos *commits* sur une machine distante, qui peut être :

- Interne à votre réseau local, si vous possédez plusieurs ordinateurs ;
- Externe, à l'aide de services comme GitHub ou BitBucket.

Nous avons vu lors de l'examen de Git que faire d'un ordinateur un serveur Git pour pouvoir s'y connecter à distance n'a rien de simple. L'alternative GitHub est largement préférable...

Première chose à bien comprendre : GitHub en lui-même n'a rien d'un outil de développement, ce n'est qu'un réseau social comme Facebook. Vous construisez un profil, vous y déposez des projets à partager et vous vous connectez (éventuellement) avec d'autres utilisateurs en suivant leurs comptes. Même si la plupart des utilisateurs y déposent des projets de programmes ou de code, rien ne vous empêche d'y placer des textes ou tout type de fichier à présenter dans vos répertoires de projets. Vous pouvez parcourir les projets d'autres utilisateurs de Github, et même y télécharger des copies pour vous-même afin de les modifier, apprendre ou les enrichir. D'autres utilisateurs peuvent faire la même chose avec vos projets, s'ils sont publics.

Et GitHub présente un intérêt certain : contrairement à d'autres réseaux sociaux, vous conservez la propriété complète de tous les projets que vous y déposez :

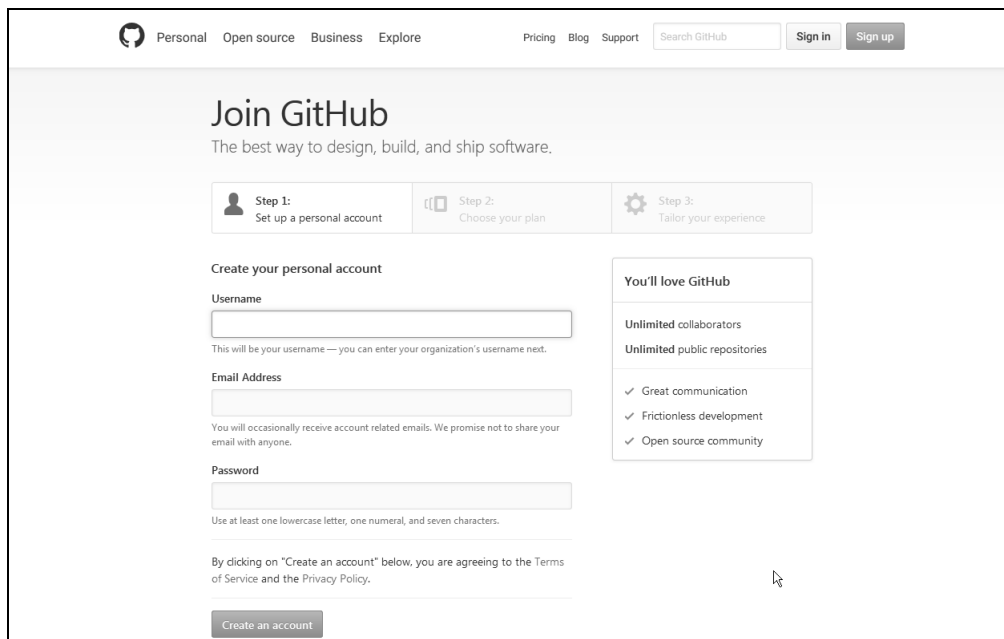
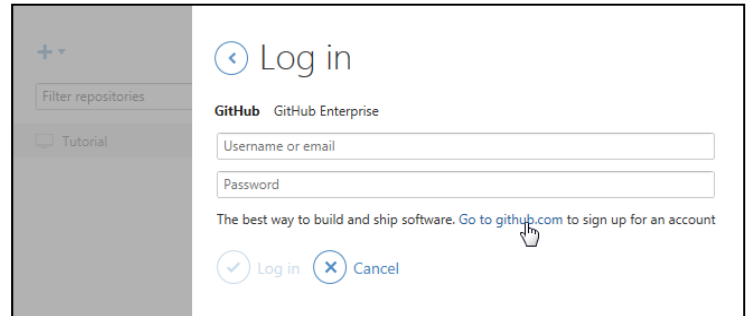
<p>“ <i>We claim no intellectual property rights over the material you provide to the Service. Your profile and materials uploaded remain yours. ”</i></p> <p>(Extrait des conditions générales GitHub)</p>

GitHub est fondé sur Git, ce qui signifie que pour l'employer au mieux vous devez déjà maîtriser les concepts, le vocabulaire et les commandes de Git. Il en facilite toutefois l'utilisation sur deux points : il procure *via* le logiciel GitHub Desktop (mais vous pourriez tout faire en ligne de commande avec Git, si vous y tenez vraiment, ou même à l'aide d'autres interfaces) une interface graphique (analogue sans être identique à celle désormais proposée pour Git) et apporte ainsi les contrôles de versions à vos projets web en leur conférant des fonctionnalités de réseaux sociaux. Son grand intérêt reste toutefois indéniablement la simplicité de configuration et d'emploi par rapport à un serveur Git autonome.

1. GitHub : le site

1.1 - Création d'un compte

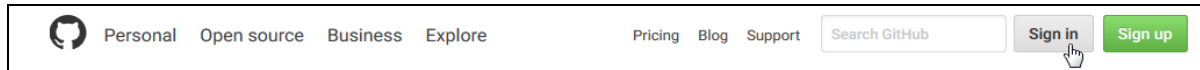
- Ouvrez votre explorateur sur l'adresse **github.com**.
- Choisissez **Join GitHub for Free** (ou une autre option si vous le souhaitez). Dans la page qui s'ouvre, choisissez un identifiant, saisissez votre (ou une, mais vous devrez pouvoir y récupérer un message) adresse électronique et définissez un mot de passe. Celui-ci doit comporter 7 caractères dont au moins une lettre minuscule et un chiffre. Cliquez ensuite sur **Create an account**.



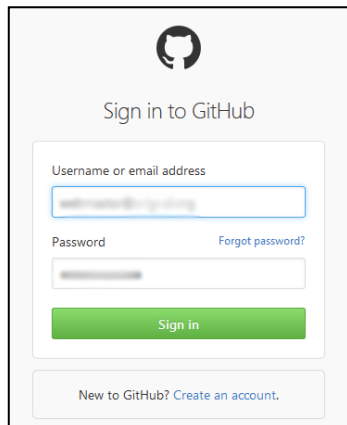
Remplissez ensuite les champs nécessaires dans les pages qui suivent, et cliquez sur **Next** ou **Submit** selon les cas. Il est ensuite procédé à une vérification de l'adresse électronique avant d'ouvrir votre compte GitHub. GitHub vous enverra un mail et vous devrez juste cliquer sur le lien contenu à l'intérieur pour activer votre compte.

1.2 - Création d'un dépôt distant

Il faut maintenant créer un dépôt pour votre projet. Git et Github appellent cela un *repository*, un répertoire numérique ou un espace de stockage où vous pouvez accéder à votre projet avec tous vos fichiers et toutes leurs versions.

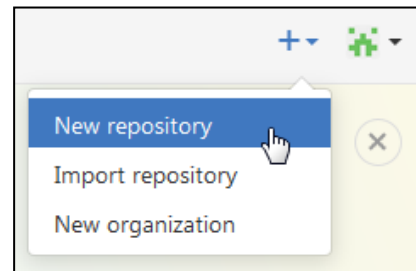


1. Retournez sur GitHub.com. Si vous n'aviez pas mémorisé votre identifiant, cliquez sur **Sign in**. Sinon, si vous accédez directement à votre compte, passez en 3.




2. Dans la nouvelle page, renseignez votre adresse électronique (celle employée pour ouvrir le compte) et votre mot de passe, puis cliquez sur à nouveau sur **Sign in**.

Cliquez sur la flèche située à côté du +, à droite, près de votre profil et choisissez **New repository**.



Create a new repository
A repository contains all the files for your project, including the revision history.

Owner:  / Repository name:

Great repository names are short and memorable. Need inspiration? How about [probable-disco](#).

Description (optional)

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

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

☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** | Add a license: **None** ⓘ

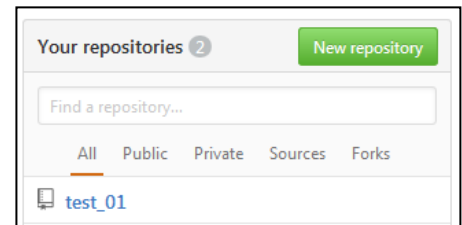
Create repository

3. Donnez à votre dépôt un nom court et mémorable. Saisissez si vous le désirez une courte description. Laissez cochée la case **Public** : il ne peut être rendu privé que si vous avez pris un compte payant. Vous avez la possibilité ici de transformer votre compte gratuit en compte payant. Cliquez enfin sur **Create repository**.

C'est fait : vous disposez d'un espace de stockage en ligne. La

fenêtre qui s'affiche donne quelques indications sur ce qui peut être fait de ce nouveau dépôt : créer un dépôt vierge ou ajouter (*push*) un dépôt déjà créé localement. Puisque nous partons en principe de rien, contentons-nous de créer un dépôt vierge. Le plus important est de mémoriser l'adresse Web d'accès à votre dépôt. C'est l'adresse de cette dernière page, soit `https://github.com/identifiantGitHub/nomDepot`, où *identifiantGitHub* est votre identifiant et *nomDepot* le nom donné à votre dépôt.

Par la suite, pour créer un nouveau dépôt vous cliquerez sur le bouton vert **New repository** du cadre **Your repositories**.

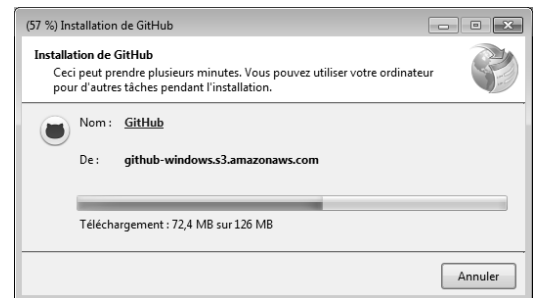


2. GitHub Desktop

Vous pourriez parfaitement travailler avec GitHub à l'aide de la ligne de commande de Git, ou d'une de ses GUI, mais nous allons ici employer le produit dédié, GitHub Desktop.

2.1 - Installation

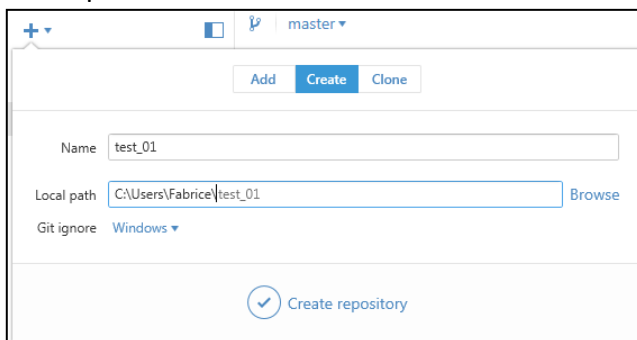
- Téléchargez GitHub pour Windows à l'adresse <http://github-windows.s3.amazonaws.com/GitHub-bSetup.exe>
- Lancez le fichier téléchargé, cliquez sur **Accepter**
- Cliquez sur **Installer**. GitHub va télécharger les données dont il a besoin, puis installer le logiciel.



2.2 - Création d'un dépôt local

Comme vous travaillerez le plus souvent sur votre ordinateur, vous devez créer un équivalent local du dépôt distant précédemment créé.

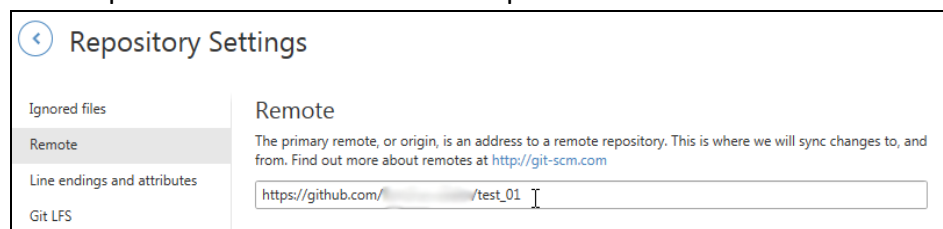
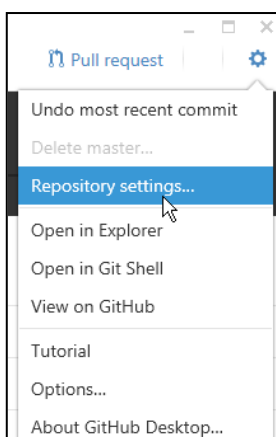
1. Commencez par créer depuis l'Explorateur Windows un répertoire nommé *nomDepot*, où *nomDepot* est le nom du dépôt distant que vous avez créé à l'étape précédente; Mieux vaut toujours garder une cohérence maximale, vous pourriez avoir à créer d'autres dépôts locaux.



2. Dans GitHub, cliquez sur la flèche du signe + en haut à gauche. Sélectionnez **Create** si nécessaire.
3. Saisissez le nom du dépôt (toujours *nomDepot* dans notre cas). Ajustez si besoin est le chemin d'accès vers le répertoire créé à l'étape 1, puis cliquez sur **Create repository**.

Parfait ! Le dépôt local est créé. Il faut maintenant le lier à notre dépôt distant.

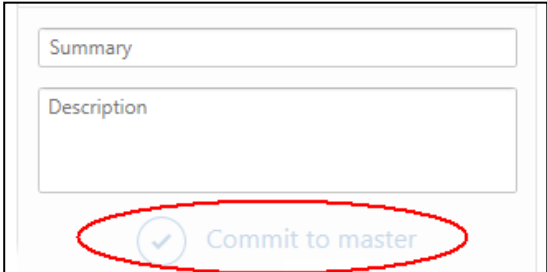
4. Cliquez à droite sur la roue dentée et choisissez **Repository settings**. Dans la fenêtre qui s'ouvre, choisissez **Remote** puis saisissez l'URL de votre dépôt sur GitHub.com.



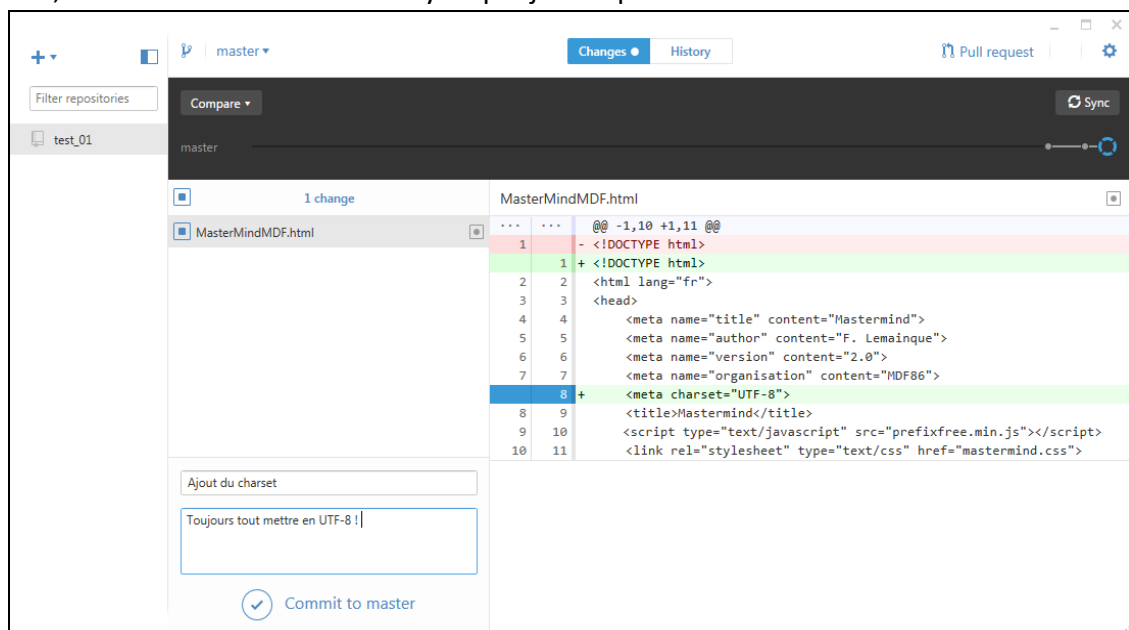
Vous venez d'établir la connexion entre vos deux dépôts : vous disposez désormais à la fois d'un dépôt local et d'un dépôt distant pour votre projet.

3. Travail en local et à distance

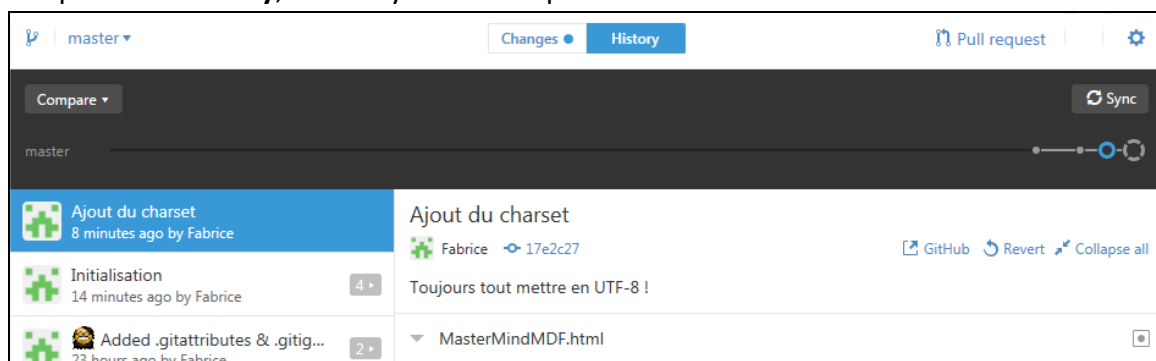
A partir de ce moment, les choses sont d'une simplicité extrême : vous travaillez localement comme à l'habitude, en enregistrant vos fichiers dans le répertoire local créé. Cela fait, vous devez initialiser le répertoire distant : créez un premier commit en remplissant dans GitHub Desktop les champs Summary et Description, puis cliquez sur le lien **Commit to master**. Il est encadré en rouge ci-contre car le lien apparaît en grisé tant que vous n'avez pas renseigné les champs.



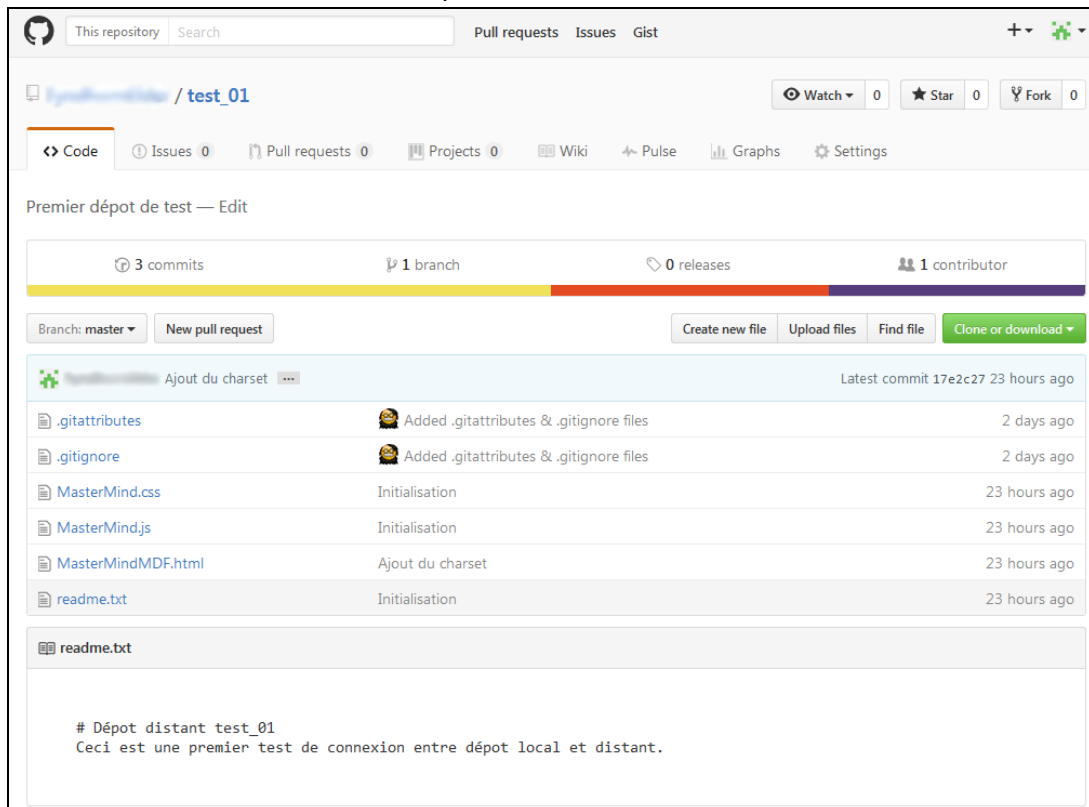
Par la suite, en cliquant sur **Change** vous verrez les modifications apportées à un fichier en particulier, comme ci-dessous. Vous voyez que je vais procéder à un *commit* sur le Master.



En cliquant sur **History**, vous voyez l'historique des modifications.



Vous pouvez à tout moment cliquer sur **Sync** pour synchroniser votre Master local et le Master distant. Vous verrez alors le dépôt distant refléter vos modifications.



Il n'est montré ici que le travail sur un seul poste : si vous travaillez à plusieurs, vous travaillerez sur des branches du Master (reportez-vous au cours sur Git). Ce sont ces branches qui seront actualisées et le créateur du dépôt sur le site gitHub devra autoriser la fusion d'une branche avec le master, ce qui revient à accepter les modifications. Mais, comme avec Git, vous gardez un historique complet des modifications et pouvez revenir en arrière sur n'importe quel commit.

Ce type d'outil peut paraître complexe initialement, mais s'il est déjà fortement utile pour un individu il devient indispensable lors d'un travail en groupe.