GitHub

Table des matières

GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Le nom GitHub est composé du mot « git » faisant référence à un système de contrôle de version open-source et le mot « hub » faisant référence au réseau social bâti autour du système Git.

1. Introduction

GitHub est un service web d'hébergement et de gestion collaborative de développement de logiciels basé sur le programme Git. De manière résumée, à travers GitHub vous pouvez accéder aux fichiers d'un projet que quelqu'un d'autre a créé ou bien créer votre projet et permettre à d'autres utilisateurs d'y accéder. L'historique de l'évolution des projets dans github est également conservé, ce qui, au besoin, permet de pouvoir revenir à des versions antérieures. À cela s'ajoute bien d'autres fonctionnalités. L'objectif de ce chapitre est de donner des bases pour l'utilisation de GitHub.

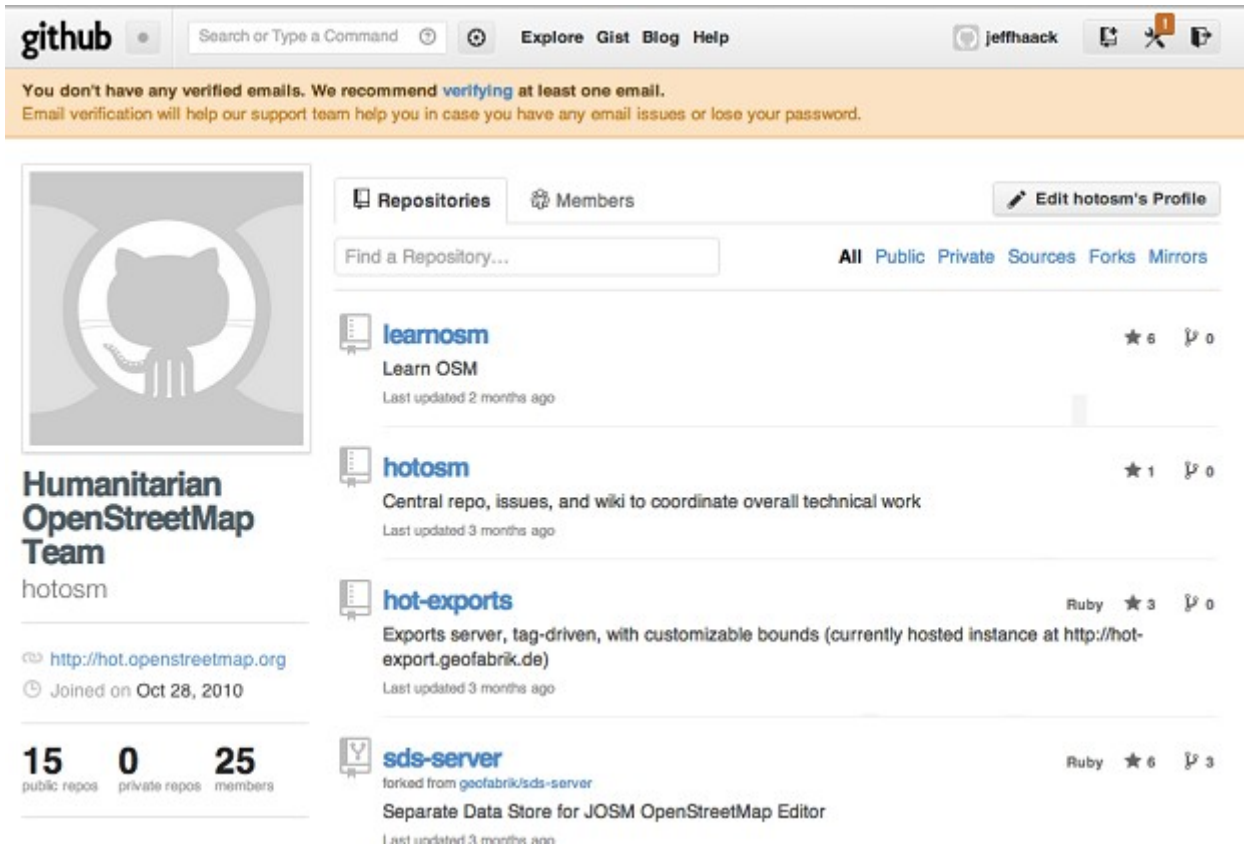
2. Les dépôts

Avec votre navigateur, accédez à la page d'accueil de GitHub. Rappelons qu'il y a une différence entre Github et Git. Git est un programme de suivi de version qui peut transformer en dépôt (repository) tout dossier dans votre machine. Git prend en compte également le suivi de tous les changements qui s'opèrent dans un dépôt donné. GitHub est un service web qui permet aux usagers de partager leurs dépôts Git et ainsi de facilement collaborer sur des projets. Essentiellement GitHub permet à ses utilisateurs de créer des dépôts qui peuvent contenir de multiples fichiers, dossiers et sous-dossiers.

Un dépôt est l'équivalent d'un dossier de projet. Quand on regarde un dépôt sur GitHub, on voit la version la plus récente des fichiers.

Les dépôts peuvent être publics ou privés. Pour rendre un dépôt privé sous GitHub il faut un compte payant. Dans un cadre open-source, nous travaillerons uniquement avec des dépôts publics.

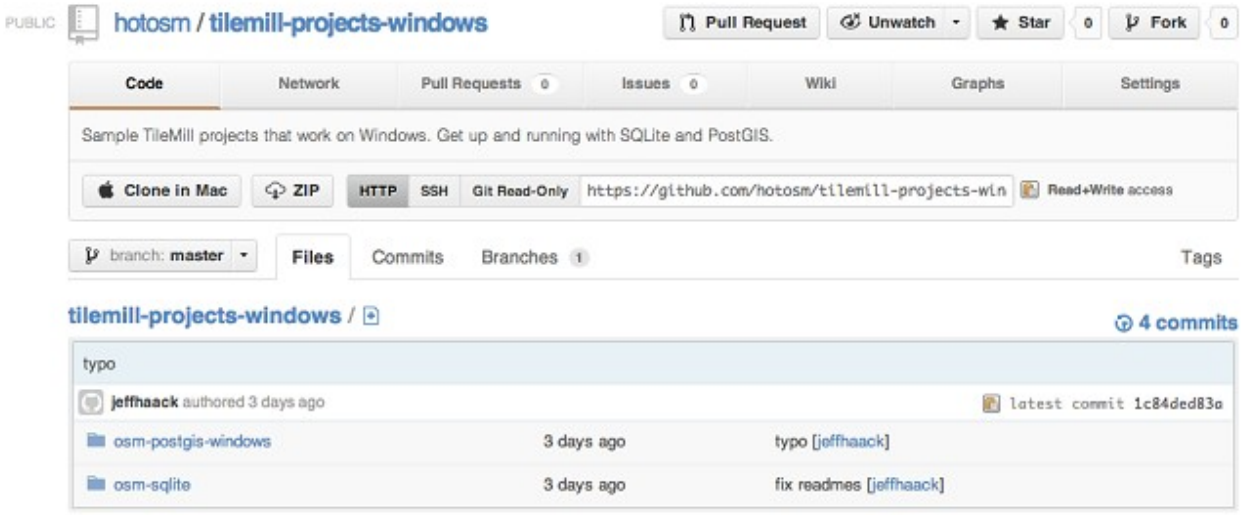
Quand un dépôt est public, on peut facilement télécharger son contenu et l'utiliser. Si vous avez les permissions, vous pouvez également modifier les fichiers et sauvegarder vos changements dans GitHub.



3. Accéder à un dépôt

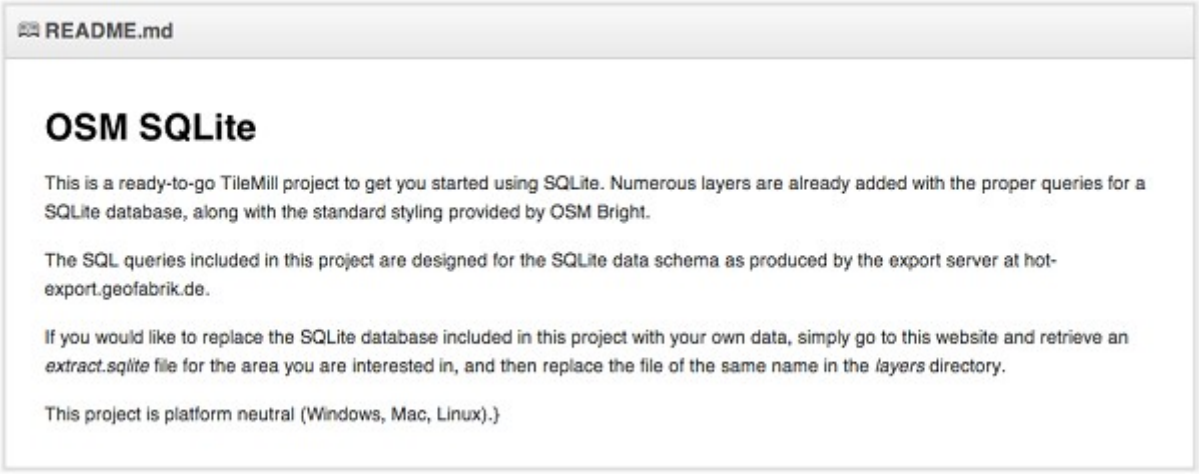
Il est utile de comprendre comment accéder aux fichiers d'un dépôt pour une utilisation personnelle.

Repérez le dépôt nommé tilemill-projects-windows. Ce dépôt contient plusieurs projets TileMill pouvant s'exécuter sous Windows. Un de ces projets utilise une base de données PostGIS comme source de données et le deuxième utilise une base SQLite. Si vous n'avez pas repéré le dépôt, vous y accéderez à cette adresse.



Vous pouvez visualiser le contenu des dossiers et tous les fichiers listés dans ce dépôt. Ceci vous

donnera une idée du contenu du projet. La plupart des projets contiennent un fichier nommé README.md. Le readme est un fichier texte qui contient des informations basiques à propos du dépôt : généralement on y retrouve toutes les instructions pour l'installation du projet. Le contenu du fichier readme sera affiché juste après la liste des fichiers contenus dans le dépôt. Vous pouvez donc directement le lire. Ouvrez le sous-dossier osm-sqlite et allez jusqu'en bas de page pour voir le contenu du readme.


README.md

OSM SQLite

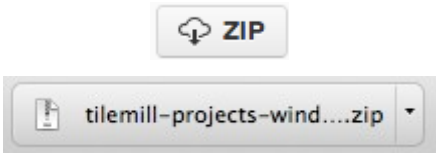
This is a ready-to-go TileMill project to get you started using SQLite. Numerous layers are already added with the proper queries for a SQLite database, along with the standard styling provided by OSM Bright.

The SQL queries included in this project are designed for the SQLite data schema as produced by the export server at hot-export.geofabrik.de.

If you would like to replace the SQLite database included in this project with your own data, simply go to this website and retrieve an *extract.sqlite* file for the area you are interested in, and then replace the file of the same name in the *layers* directory.

This project is platform neutral (Windows, Mac, Linux).)

Pour télécharger l'ensemble des ressources du projet, cliquez simplement sur le bouton "ZIP" et tous les fichiers du projet seront téléchargés sur votre ordinateur dans un dossier zippé.



Si vous voulez uniquement accéder aux fichiers du projet à travers GitHub, ce qui précède est tout ce dont vous avez besoin. Si vous désirez en savoir plus sur Git et GitHub, continuez la lecture de ce chapitre.

4. Présentation de Git

4.1. Les principales étapes

Git et GitHub sont deux applications distinctes. Github est un site web pour héberger des dépôts et qui facilite ainsi la collaboration sur des projets entre plusieurs intervenants. Le site GitHub fonctionne avec Git qui est un programme qui permet à des utilisateurs de sauvegarder différentes versions des fichiers durant le cycle de vie d'un projet.

Regardons de plus près comment le programme Git fonctionne. Git garde l'historique des fichiers d'un dépôt. Plutôt que d'enregistrer un fichier à chaque changement, Git sauvegarde le fichier une fois. Par la suite, à chaque fois que vous enregistrez une nouvelle version, Git sauvegarde les changements qui s'appliquent à votre fichier. Ceci rend la sauvegarde de fichiers plus efficace.

En permettant de travailler en mode déconnecté ou sur des copies de vos fichiers en ligne et sauvegardés dans GitHub, Git réduit le risque de perte et d'endommagement des fichiers. Quand vous le jugez opportun, vous pouvez faire un commit (envoi de vos changements) pour ainsi synchroniser votre travail avec le dépôt GitHub. Quand vous placez un projet dans GitHub,

plusieurs personnes peuvent copier et modifier le même fichier.

Les principales étapes pour travailler avec un dépôt GitHub sont :

1. **Clone** : dupliquer le dépôt de GitHub vers votre machine
2. **Modify** : modifier les fichiers copiés en local
3. **Stage** : déterminer les changements que vous voulez synchroniser
4. **Commit** : envoyer vos modifications
5. **Sync** : synchroniser vos changements avec le dépôt

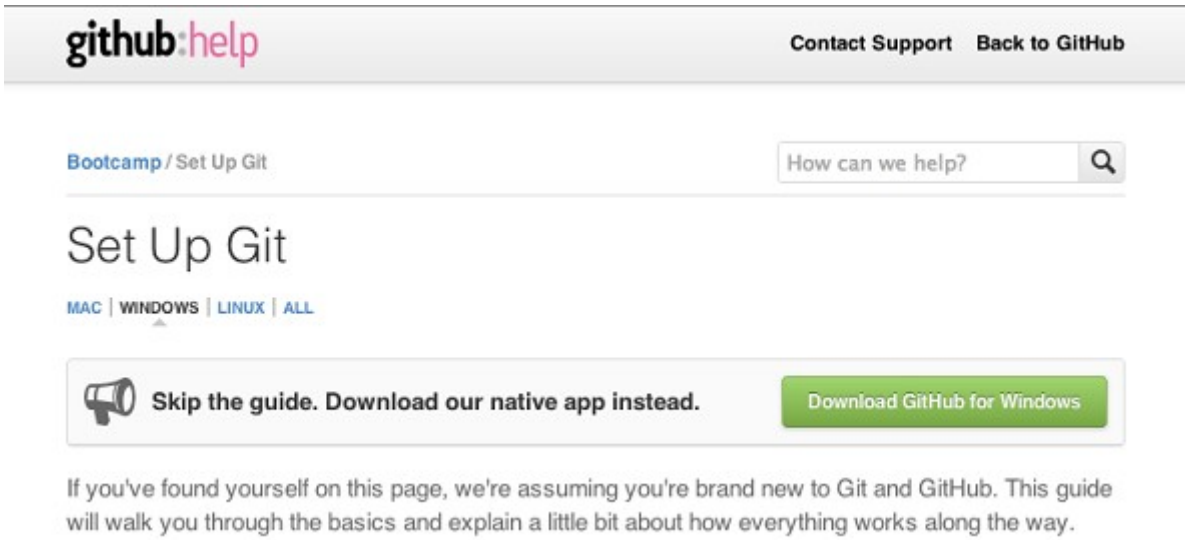
4.2. Le vocabulaire « Git »

Voici les plus connus :

- **Ligne de Commande** : Le programme de l'ordinateur que nous utilisons pour entrer des commandes Git. Cela s'appelle un Terminal : au lieu d'utiliser une souris, vous saisissez des commandes basées sur le texte, connues comme des invites de commande.
- **Dépôt** : Un répertoire ou un espace de stockage où vos projets peuvent vivre. Parfois les utilisateurs GitHub raccourcissent ça en "repo". Il peut être local vers un répertoire sur votre ordinateur, ou ce peut être un espace de stockage sur GitHub ou un autre hébergeur en ligne. Vous pouvez conserver des fichiers de code, des fichiers texte, des images, à l'intérieur d'un dépôt.
- **Contrôle de Version** : Basiquement, l'objectif pour lequel Git a été conçu. Quand vous avez un fichier LibreOffice, vous l'écrasez à chaque fois que vous faites une nouvelle sauvegarde, ou sinon vous sauvegardez plusieurs versions. Avec Git, vous n'êtes plus obligé de faire ça. Il conserve des "instantanés" de chaque point dans l'historique d'un projet, par conséquent vous ne pouvez jamais le perdre ou l'écraser.
- **Commit** : C'est la commande qui donne à Git toute sa puissance. Quand vous "committez", vous prenez un "instantané", une "photo" de votre dépôt à ce stade, vous donnant un checkpoint que vous pouvez ensuite évaluer de nouveau ou restaurer votre projet à tout état précédent.
- **Branche** : Comment plusieurs personnes peuvent travailler sur un projet en même temps. Généralement, elles se "débranchent" du projet principal avec leurs propres versions pleines de modifications qu'elles ont chacune produites de leur côté. Après avoir fait ça, il est temps de "fusionner" cette branche là avec le "master", le répertoire principal du projet.

5. Installation de Git

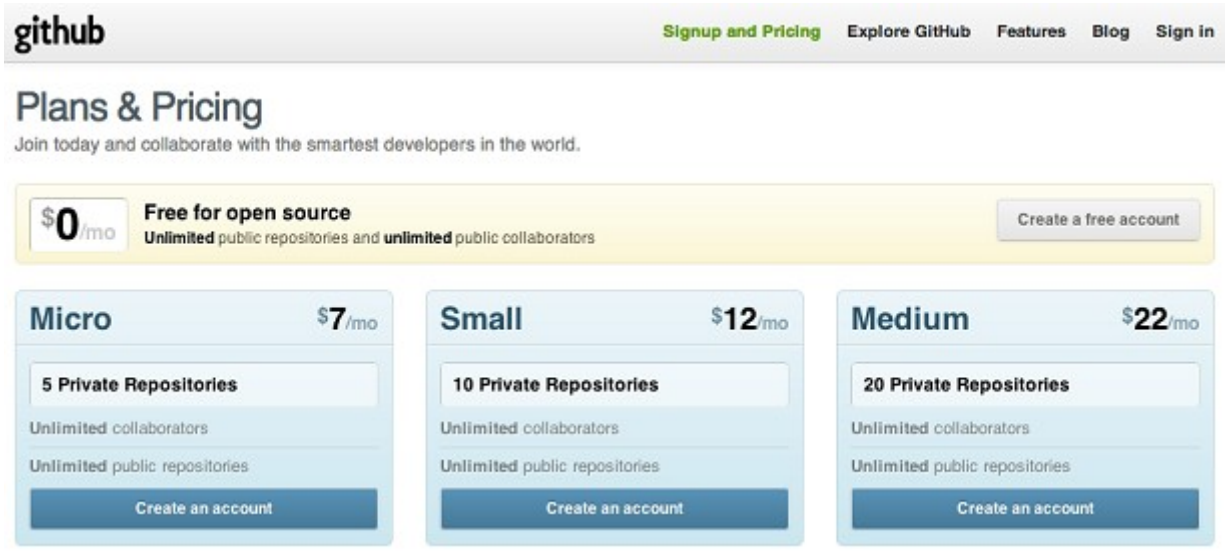
Pour installer une interface pour travailler avec Git, rendez-vous à cette adresse, télécharger puis installer le programme. Il y a bien sûr différentes versions pour différents systèmes d'exploitation.



Cliquez sur “Download GitHub for Windows” pour télécharger l’installateur.

6. Créer un compte GitHub

Pour créer un compte GitHub, rendez-vous à cette adresse. Si vous prévoyez de travailler uniquement avec des dépôts publics, vous pouvez simplement choisir de créer un compte gratuit.

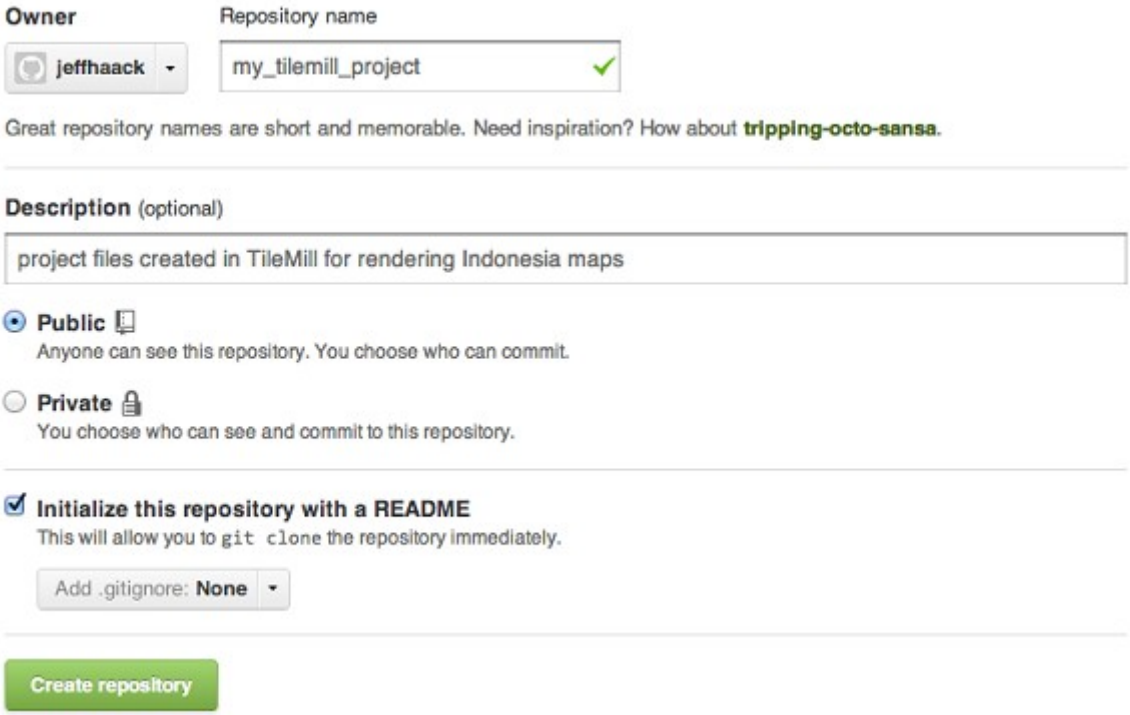


Cliquez sur “Create a free account” et suivez les instructions pour créer votre compte.

7. Créer un nouveau dépôt

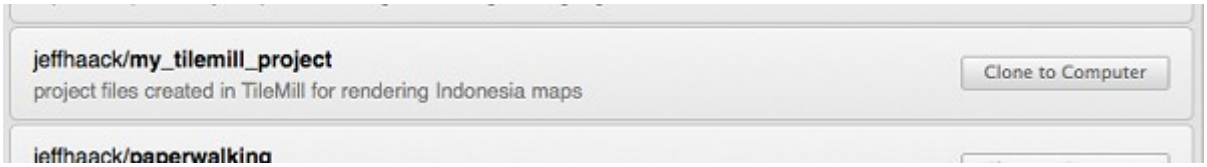
Cliquez sur “Create a New Repo” dans la section à droite en haut du site.

Donnez un nom et une description à votre projet. Par défaut vous aurez un dépôt public et le projet sera vide à sa création. Nous pouvons y placer un fichier readme, en cochant la case “Initialize this project with a README”.



Maintenant que le dépôt est créé, vous pouvez le cloner en local, sur votre ordinateur. Lancez le programme GitHub que vous venez d'installer. Vous allez ajouter les informations GITHUB pour que le programme puisse accéder à votre compte. Vous pourrez par la suite changer ces informations en vous rendant sur la page des préférences.

Retrouvez la liste de vos dépôts GitHub en cliquant sur votre nom d'utilisateur. Quand vous avez repéré le nouveau projet, cliquez sur "Clone to Computer".

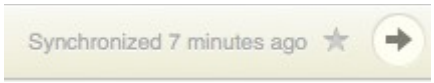


Une boîte de dialogue va s'ouvrir et vous allez parcourir votre disque local pour indiquer dans quel dossier le nouveau projet sera copié.

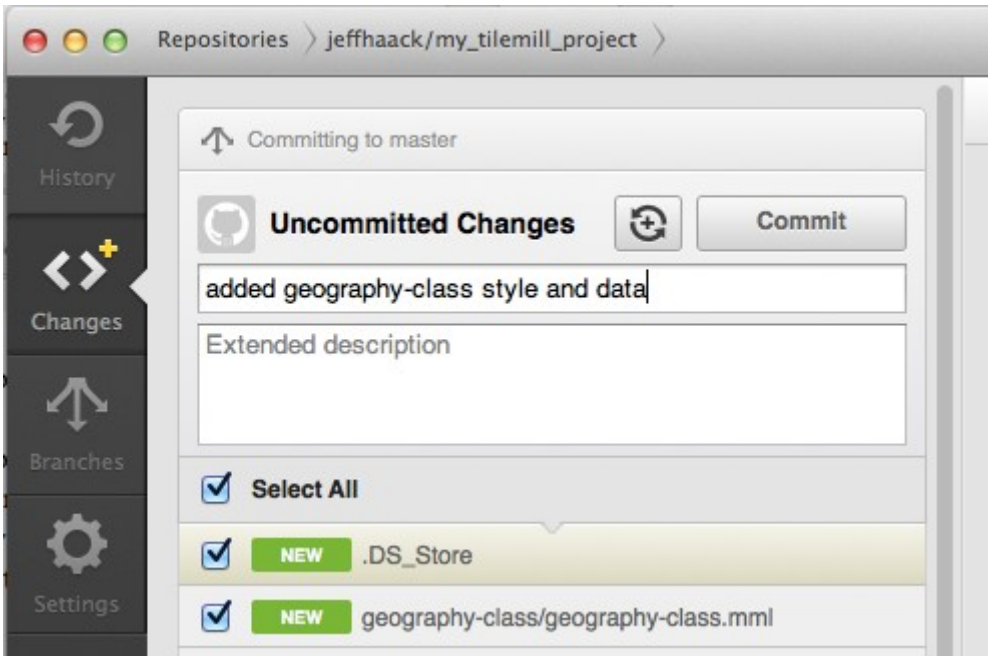
Une fois le dépôt cloné, vous pouvez travailler dans le dossier du projet de votre ordinateur comme sous n'importe quel autre dossier. La seule différence est que le dossier de votre projet contient des fichiers cachés qui retracent les changements que vous avez effectués et qui permettent de synchroniser le dépôt local avec le dépôt en ligne.

Dans notre exemple nous avons copié le sous-dossier geography-class.

Retournez au programme Git et cliquez sur la flèche à droite du nom de votre projet.

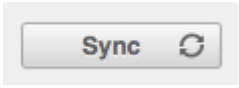


Vous allez voir la liste de tous les fichiers que avez modifiés ou ajoutés. Dans notre exemple, nous avons ajouté tous les nouveaux fichiers dans le sous-dossier geography-class. Pour sauvegarder nos changements, nous allons en saisir un résumé descriptif et faire un commit.

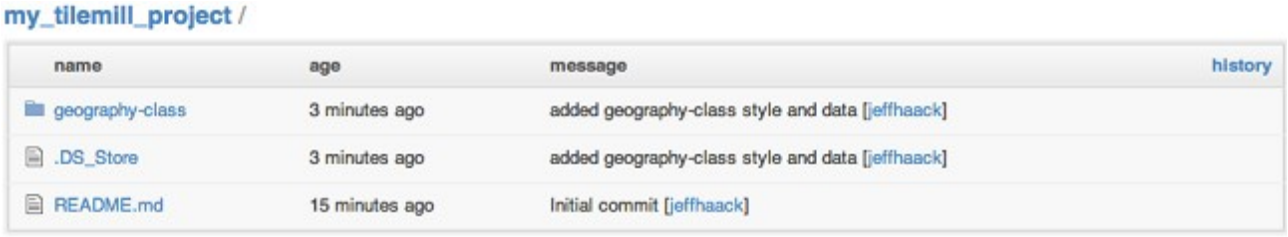


Cliquez sur “Commit” pour faire un commit au niveau local.

Enfin nous avons besoin de synchroniser nos changements avec le dépôt distant dans GitHub. Cliquez sur “Sync” pour lancer la synchronisation.



Retournez à votre navigateur et allez visiter la page de votre dépôt sur GitHub. Vous verrez que les fichiers de notre commit ont été transférés avec succès.



Pour en savoir plus, cliquer ici