GitHub et Visual Studio

Table des matières

[1 Gestion de versions 2](#_Toc477078530)

[1.1 Les besoins 2](#_Toc477078531)

[1.2 Les 2 architectures possibles 2](#_Toc477078532)

[1.2.1 Gestion centralisée 2](#_Toc477078533)

[1.2.2 Gestion décentralisée 2](#_Toc477078534)

[2 Git et GitHub 2](#_Toc477078535)

[2.1 Présentation 2](#_Toc477078536)

[2.2 Création d’un compte GitHub et d’un référentiel 3](#_Toc477078537)

[3 Intégration dans Visual Studio 4](#_Toc477078538)

[3.1 Installation de l’extension GitHub 4](#_Toc477078539)

[3.2 Connexion au compte GitHub 6](#_Toc477078540)

[3.3 Clonage ou création d’un référentiel 6](#_Toc477078541)

# Gestion de versions

## Les besoins

La réalisation d’un projet informatique introduit les besoins suivants :

* Pouvoir travailler à plusieurs sur le même code source
* Pouvoir travailler de façon décentralisée
* Pouvoir suivre les modifications (qu’est-ce qui a été modifié depuis telle date, par qui…) et revenir en arrière si certaines posent problème
* Pouvoir gérer successivement ou en parallèle plusieurs versions du projet

Ces besoins sont couverts par les logiciels de « gestion de versions ».

NB/ On utilise aussi l’expression « logiciel de contrôle de code source ». En anglais, on emploi l’expression « Version Control »

Un logiciel de contrôle de version permet entre autres de :

* Conserver différentes versions d’un même code source
* Comparer 2 versions entre elles, afin de savoir quelles modifications ont été apportées dans la version la plus récente
* Identifier l’auteur d’une modification
* Fusionner les modifications de deux développeurs travaillant sur les mêmes sources

Les logiciels les plus connus : Git, Team Fundation System (TFS), CVS, SVN, Mercurial, Bazaar…

## Les 2 architectures possibles

Les logiciels de gestion de versions peuvent être classés selon leur type d’architecture pour la gestion des sources :

### Gestion centralisée

Dans cette architecture, l’ensemble des sources et versions est centralisé sur un seul serveur qui fait référence. On désigne d’ailleurs cet ensemble par le mot « Référentiel ». Chaque développeur se connecte au serveur pour récupérer les sources en local, puis les renvoyer après modification.

TFS, CVS et SVN sont dans ce type d’architecture

### Gestion décentralisée

Dans cette architecture, chaque développeur possède la totalité des sources, et peut transmettre ses modifications à d’autres développeurs. Il n’y a en théorie pas besoin de serveur central.

Git et Mercurial sont dans ce type d’architecture

Dans la pratique, même pour une gestion décentralisée, un serveur est tout de même utilisé pour les échanges, et il existe la notion de branche principale (master) qui est la référence de l’état actuel du projet.

# Git et GitHub

## Présentation

Git est un système de gestion de versions créé par Linus Torvalds en 2005, et qui s'est énormément développé depuis. C'est actuellement le plus populaire pour les logiciels libres.

GitHub est un service en ligne d’hébergement de sources, utilisant Git pour la gestion des versions. Il propose des comptes gratuits pour les projets de logiciels libres, c’est-à-dire dont les sources sont accessibles par tous.

Il propose également un service payant pour la gestion de référentiels privés.

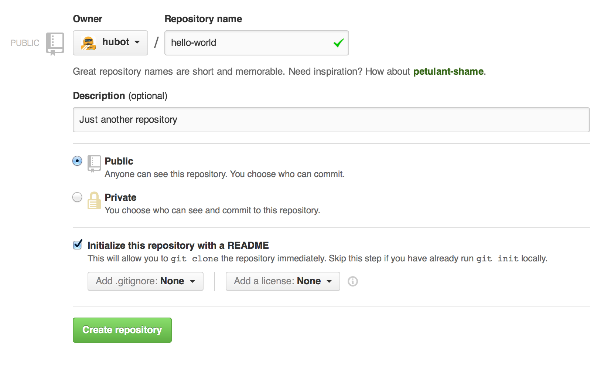
Le site fournit également des fonctionnalités destinées à la collaboration, comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

GitHub a été lancé le 10/04/2008. En avril 2016, il a annoncé avoir dépassé les 14 millions d'utilisateurs et plus de 35 millions de dépôts de projets, le plaçant comme le plus grand hébergeur de code source au monde.

## Création d’un compte GitHub et d’un référentiel

Créer un comte perso sur <https://github.com/>

Interface de création d’un référentiel (repository) :



Chaque référentiel a une URL du type [https://github.com/<login>/<referentiel>.git](https://github.com/%3clogin%3e/%3creferentiel%3e.git)

Exemple : https://github.com/adupont/exercices.git

Pour la formation, créer 2 référentiels :

* Exercices
* Projets

Le référentiel Projet devra avoir la structure de répertoire suivante :

* Console
* SGBD
* Winforms
* WPF

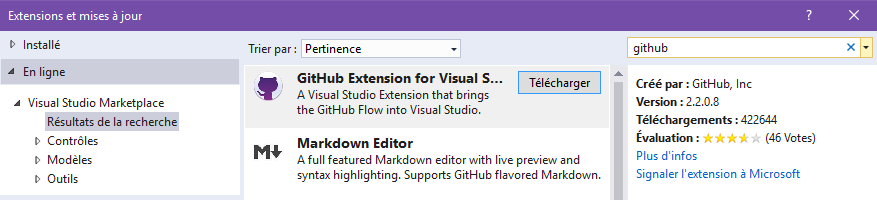
# Intégration dans Visual Studio

## Installation de l’extension GitHub

L’extension « GitHub Extension for Visual Studio » permet d’utiliser facilement un référentiel GitHub dans Visual Studio. Elle est compatible avec VS 2015 et versions supérieures.

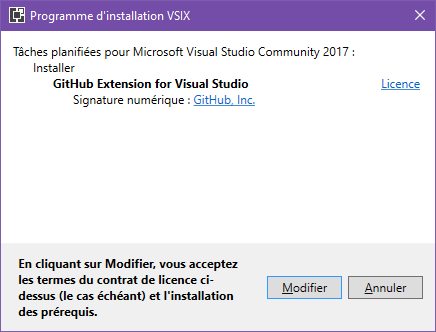
Pour l’installer :

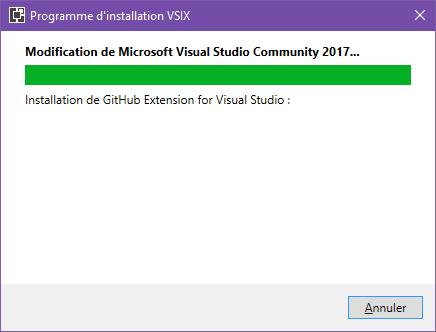
* Ouvrir la fenêtre de gestion des extensions via le menu « Outils \ Extension et mises à jour »
* Sélectionner la rubrique « En ligne » dans la partie gauche
* Saisir Github dans la zone de recherche

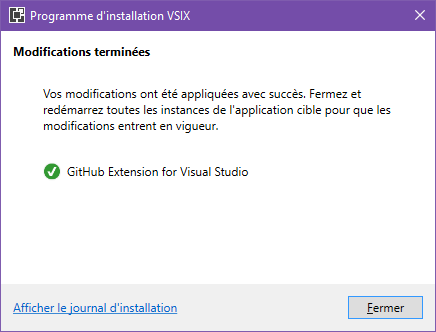


* Puis cliquer sur Télécharger

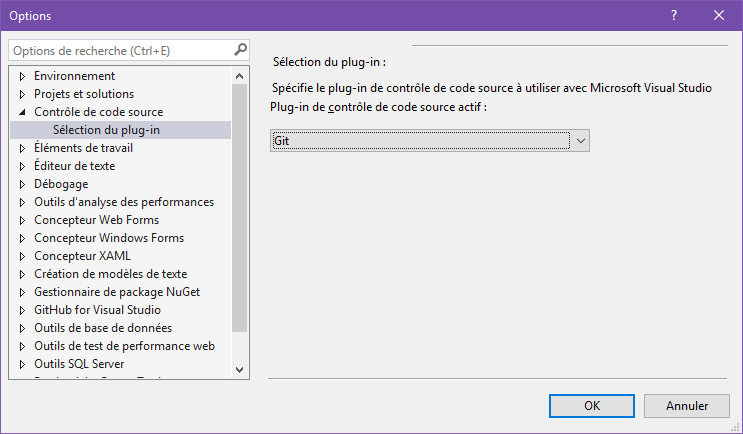
L’installation ne démarre qu’à la fermeture de Visual Studio. Les fenêtres suivantes s’affichent alors :





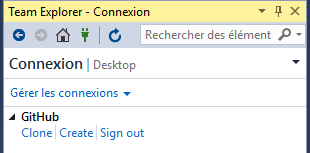


Dans les paramètres de Visual Studio, définir Git comme plug-in de contrôle de code source :



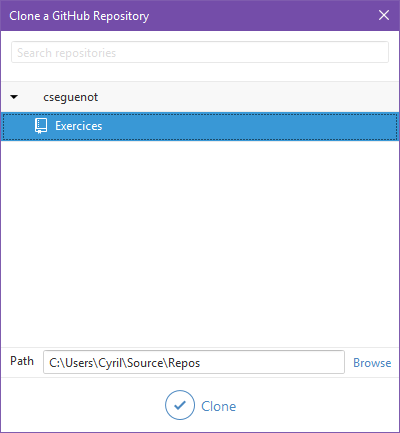
## Connexion au compte GitHub

Cliquer sur le menu « Equipe \ Gérer les connexions ». Ceci ouvre la rubrique Connexions de Team Explorer, dans laquelle on peut se connecter au compte GitHub :



## Clonage ou création d’un référentiel

Dans la rubrique Connexions de Team Explorer, un clic sur le lien « Clone » ouvre la fenêtre suivante, qui permet de sélectionner le référentiel GitHub à récupérer en local :



Il faut définir le chemin du répertoire local dans lequel seront récupérés les fichiers, puis cliquer sur Clone

Le lien « Create » permet également de créer un nouveau référentiel directement depuis Visual Studio