

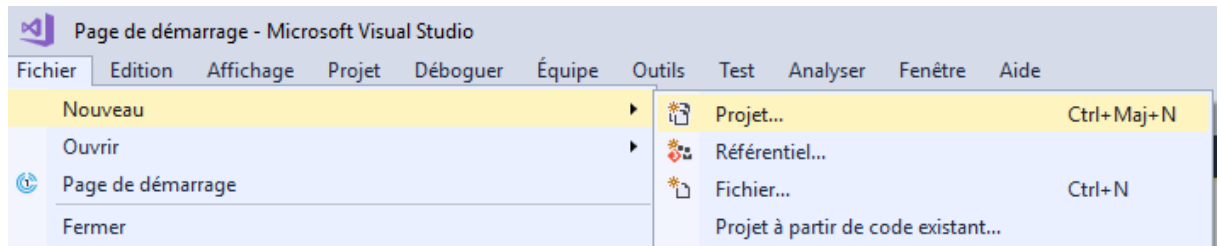
# GitHub et Visual Studio

- I. Dans cet atelier, nous allons apprendre la création d'un projet avec Visual Studio et le partager sur un contrôleur de code source (GitHub) et suivre les versions.

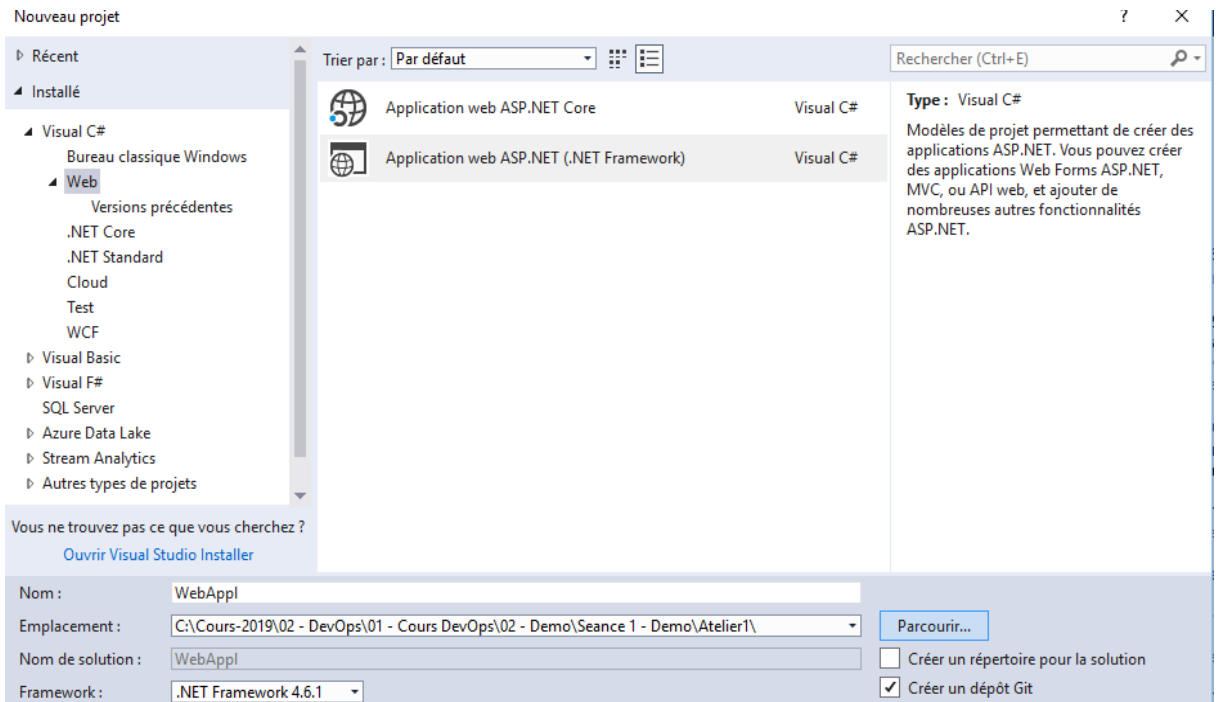
L'objectif est de créer un projet avec Visual Studio 2017, créer un Git local (sur la machine locale) puis publier de Git local sur un serveur de contrôle de code source (GitHub pour cet atelier).

Il existe deux façons pour le faire, soit passer par Visual Studio et le portail GitHub, soit de passer par l'invite de commande CIL. Dans un premier lieu, on va le faire à partir de Visual Studio.

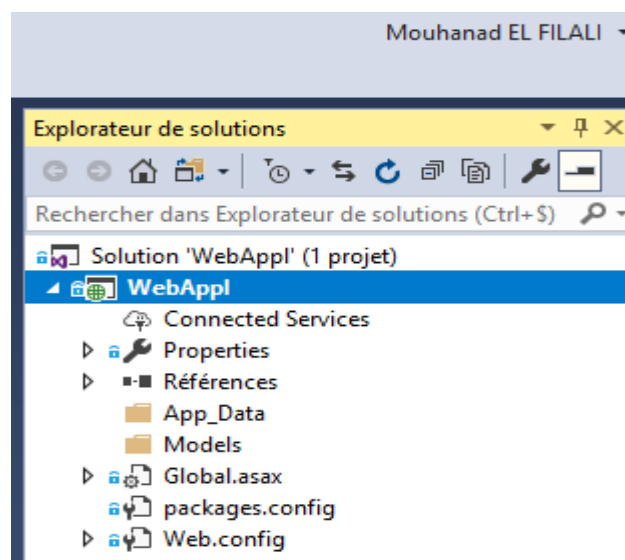
1. La première étape consiste à créer un projet (Web Form) avec un Git local à partir de VS, pour le faire, lancez Visual Studio 2017.



2. Il faut choisir comme type de projet « Application Web ASP.Net(.Net Framework).  
3. Choisissez un répertoire local pour la solution « Emplacement ».  
4. Cochez la case « Créer un dépôt Git ».  
5. Puis « Ok ».



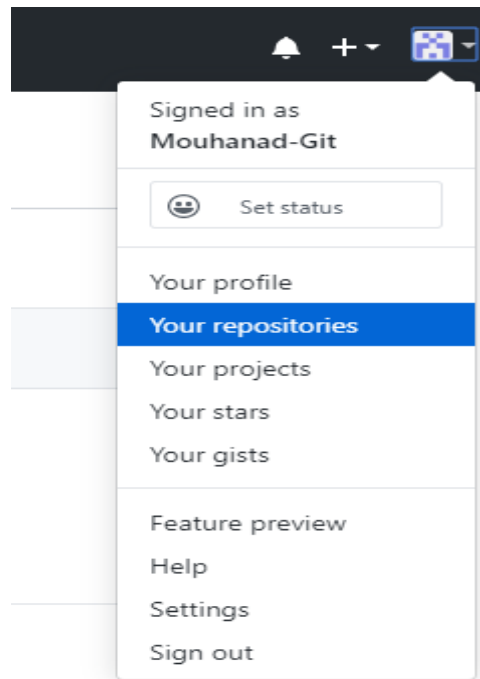
Cela va créer un répertoire local sur votre machine (Git Local), et un projet VS qui ressemble à l'arborescence ci-dessous.



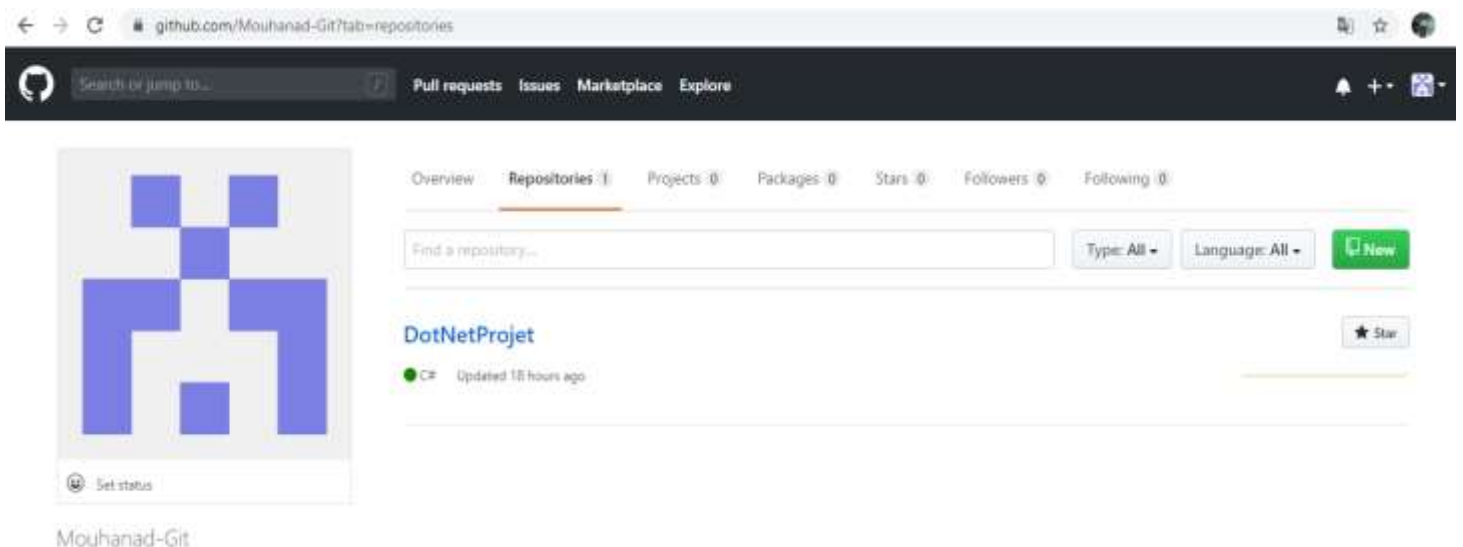
A ce stade avant de connecter le projet VS, le répertoire local avec le contrôleur code source qu'on va utiliser (GitHub), vous devez créer un compte GitHub.

6. Pour le faire, allez sur le site officiel de GitHub <https://github.com/>
7. Créez un compte GitHub (Merci bien de noter le login et le mot de passe, cela va nous aider pour la suite des ateliers DevOps).
8. Une fois le compte GitHub est créé, la suite consiste à créer un répertoire sur GitHub pour la sauvegarde de notre code source.

9. Cliquez sur le bouton en haut à droit, puis choisir « Your Repositories »



Cet écran liste l'ensemble des répertoires créés pour le compte en question, dans mon cas, il existe un seul répertoire intitulé « DotNetProjet ».



L'objectif maintenant est de créer un nouveau répertoire sur le contrôleur de code source pour partager le code source du projet qu'on vient de créer avec Visual Studio.

10. Cliquez sur le bouton « New » pour ajouter un nouveau répertoire, cela affiche l'écran ci-dessous.
11. Donnez un nom à votre répertoire, dans mon exemple, « ProjetAtleierGit ».
12. Vous pouvez garder l'option Public, cela dit, que n'importe qui peut se connecter à votre répertoire, on va explorer par la suite les options qui existent pour la gestion des restrictions.

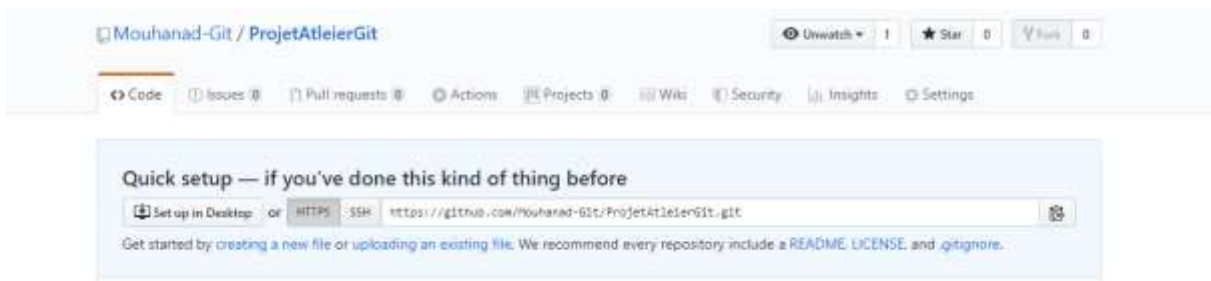
13. Cliquez sur « Create Repository ».

The screenshot shows the GitHub 'Create repository' form. At the top, the 'Owner' is 'Mouhanad-Git' and the 'Repository name' is 'ProjetAtleierGit', which is highlighted with a blue border and a green checkmark. Below this, a message says: 'Great repository names are short and memorable. Need inspiration? How about **stunning-palm-tree**?'. There is a 'Description (optional)' text area. Under 'Visibility', the 'Public' option is selected with a radio button, and the text says 'Anyone can see this repository. You choose who can commit.' The 'Private' option is unselected. A message says 'Skip this step if you're importing an existing repository.' Below that, the 'Initialize this repository with a README' checkbox is unselected, with the text 'This will let you immediately clone the repository to your computer.' At the bottom, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by an information icon. A large green 'Create repository' button is at the bottom.

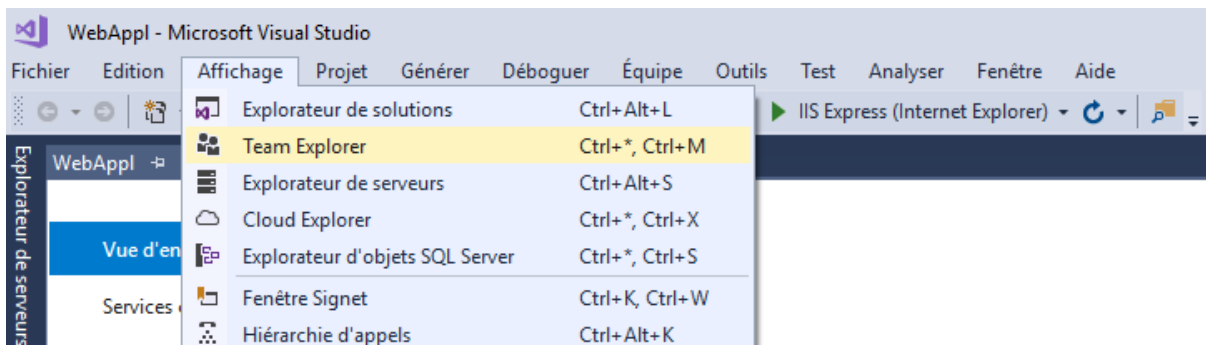
Notre Répository est créé sur le serveur sous l'adresse suivante :

<https://github.com/Mouhanad-Git/ProjetAtleierGit.git>

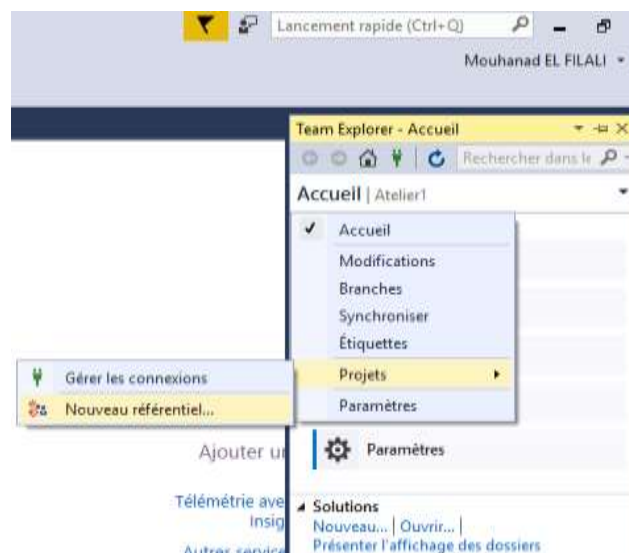
C'est l'URL qu'on va utiliser pour lier notre projet VS avec GitHub.



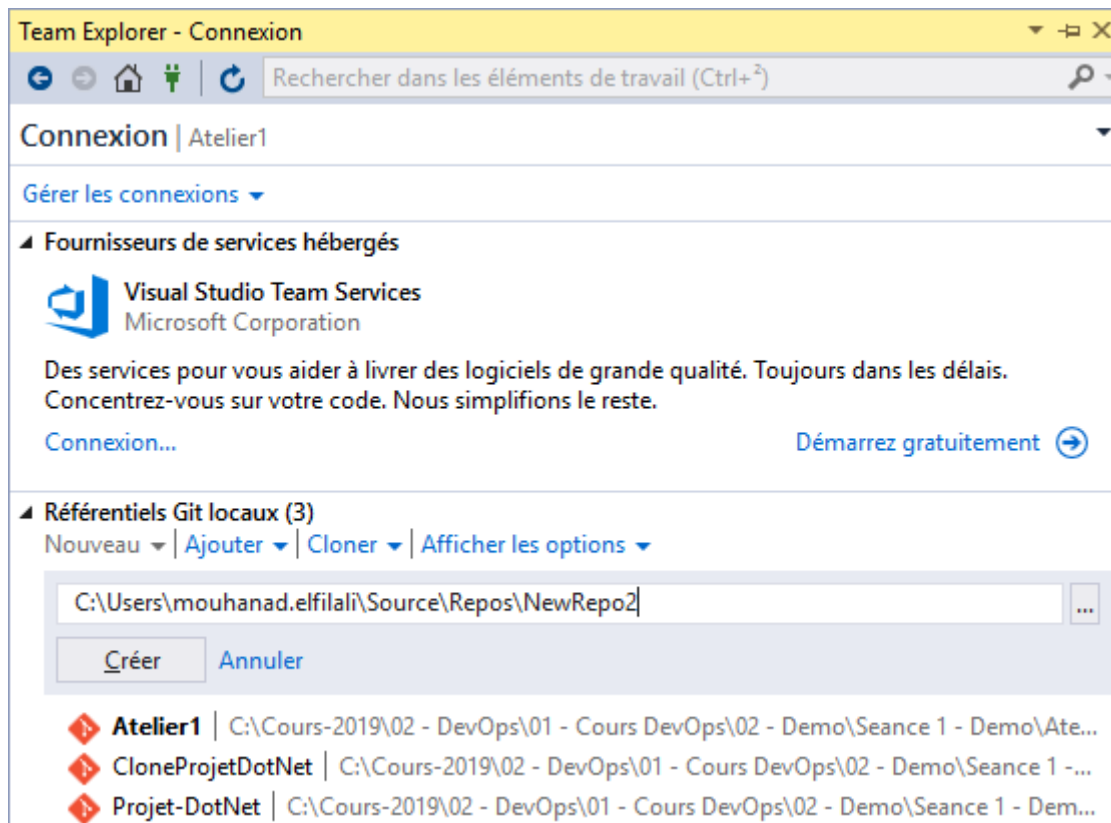
14. Maintenant, on va revenir à Visual studio pour faire la liaison entre le contrôleur de code source et le projet, pour le faire, il faut choisir Team Explorer dans l'onglet Affichage, la fenêtre TE s'affiche.



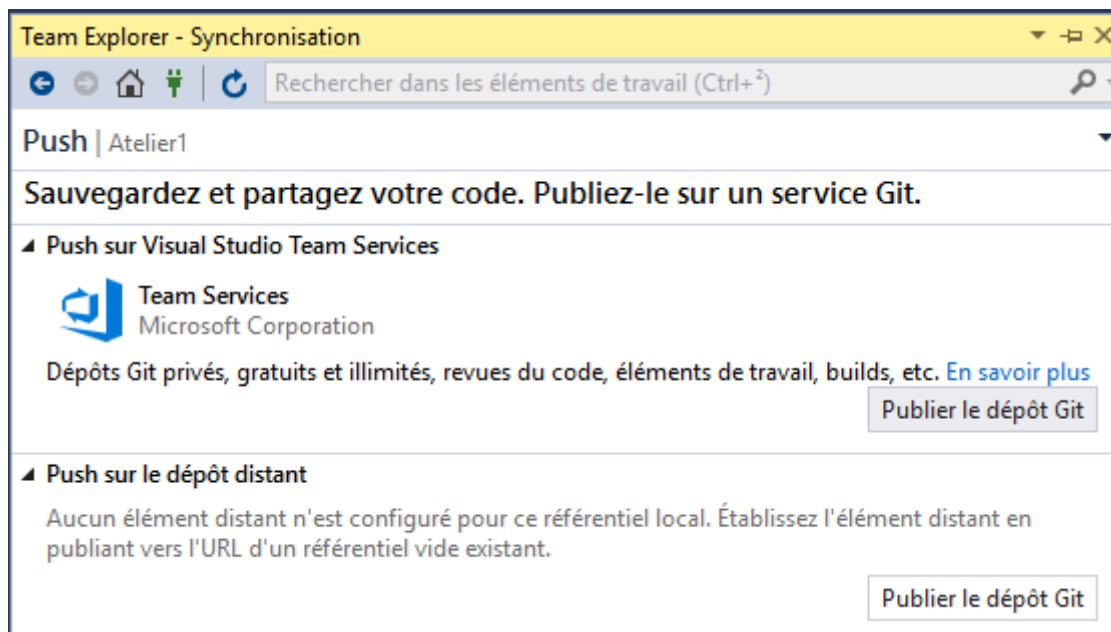
15. Maintenant, on va revenir à Visual studio pour faire la liaison entre le contrôleur de code source et le projet, pour le faire, il faut choisir Team Explorer dans l'onglet Affichage, la fenêtre TE s'affiche.



16. Cliquez sur Accueil, puis Choisir l'option Nouveau référentiel, la fenêtre suivante s'affiche avec la liste des répertoires Git local, dans la liste des répertoires, on trouve le répertoire de notre projet intitulé « Atelier1 », ce répertoire Git local a été créé car nous avons coché la case « Créer un dépôt Git » au moment de la création du projet, sinon, on peut le créer à ce stade.



17. Ce qui reste maintenant, c'est de faire le lien entre le projet et le Repository sur le GitHub, pour le faire, cliquez sur Accueil pour Synchroniser, et choisir l'option « Publier le Dépôt Git »



18. Une fenêtre s'affiche pour renseigner l'URL de dépôt Git sur le serveur.

#### ▲ Push sur le dépôt distant

Aucun élément distant n'est configuré pour ce référentiel local. Établissez l'élément distant en publiant vers l'URL d'un référentiel vide existant.

Publier le dépôt Git

Entrez l'URL d'un dépôt Git vide <Obligatoire>

Publier

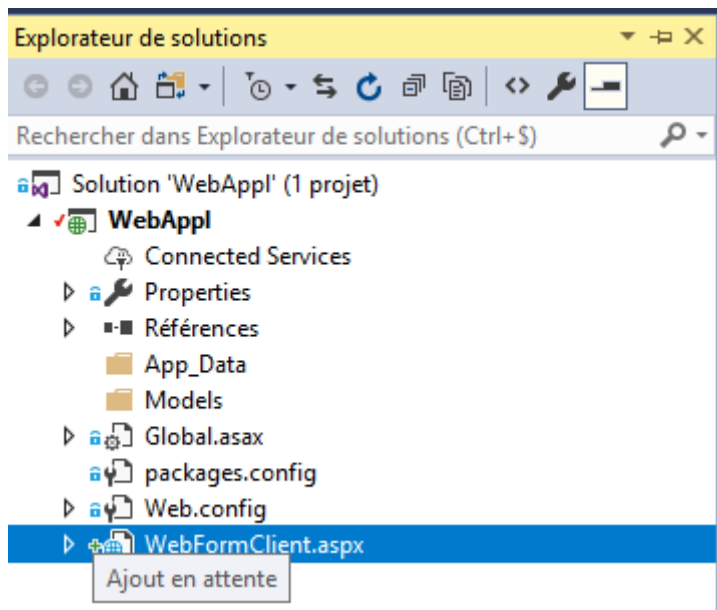
19. Collez l'URL de la repository que nous avons créé sur le serveur  
<https://github.com/Mouhanad-Git/ProjetAtleierGit.git>
20. Cliquez par la suite sur « Publier ».
21. Si la publication a réussie, vous aurez le message « La branche master a fait l'objet d'un Push réussi dans origin »

Vous pouvez revenir maintenant au portail GitHub et faire un Update de notre Repository, vous allez remarquer que le projet est maintenant sur le contrôleur de code source suite à l'opération de publication effectuée à partir de Visual Studio, vous avez réussi votre premier partage de code source sur un contrôleur de code source GitHub.

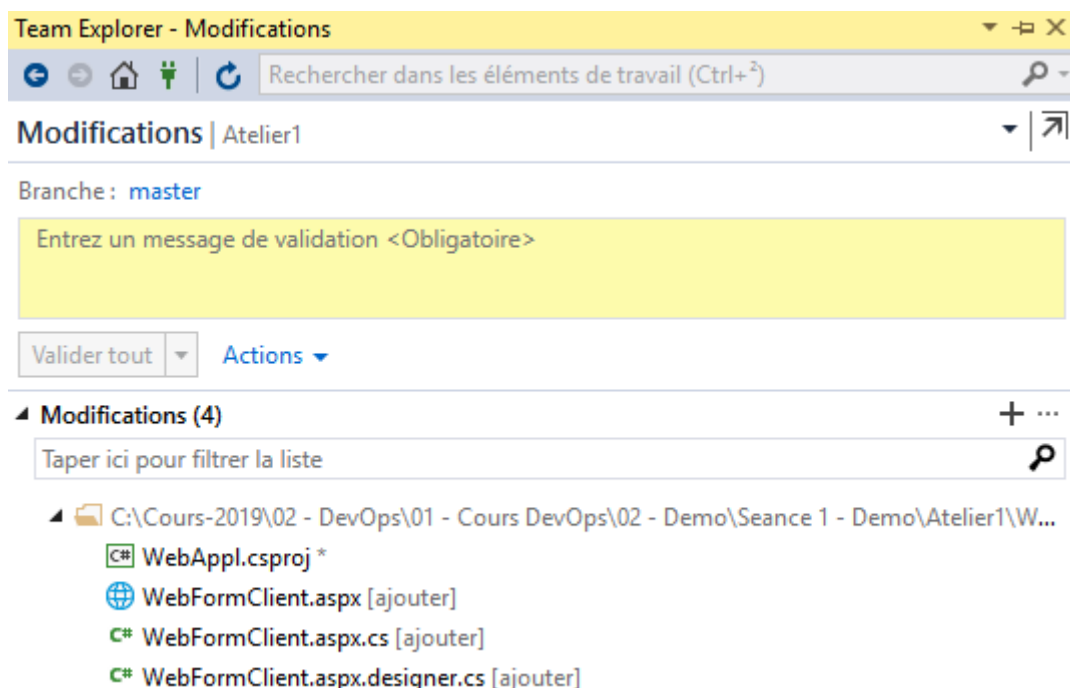
The screenshot shows the GitHub interface for a repository named 'ProjetAtleierGit' by user 'Mouhanad-Git'. At the top, there are buttons for 'Unwatch', 'Star' (1), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues' (0), 'Pull requests' (0), 'Actions', 'Projects' (0), 'Wiki', 'Security', 'Insights', and 'Settings'. A message states 'No description, website, or topics provided.' with an 'Edit' button. Below this, statistics show '2 commits', '1 branch', '0 packages', '0 releases', and '0 contributors'. A bar indicates the 'Branch: master' and provides options for 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table shows the latest commit by 'Mouhanad EL FILALI' with the message 'Ajoutez des fichiers projet.' at 13 hours ago. The table lists files: 'WebAppl', '.gitattributes', '.gitignore', and 'WebAppl.sln', all added in the same commit.

File	Commit Message	Time
WebAppl	Ajoutez des fichiers projet.	13 hours ago
.gitattributes	Ajoutez les fichiers .gitignore et .gitattributes.	13 hours ago
.gitignore	Ajoutez les fichiers .gitignore et .gitattributes.	13 hours ago
WebAppl.sln	Ajoutez des fichiers projet.	13 hours ago

22. Maintenant, on va faire une modification au niveau de notre projet VS et faire un Push vers le contrôleur de code source, cela est très simple, dans VS ajoutez une nouvelle page WebForm à votre solution. On remarque bien que la page qu'on vient d'ajouter porte le signe « + » et un message « Ajout en attente » et que le projet en mode modification (Nom du projet procédé par le signe « Validation »



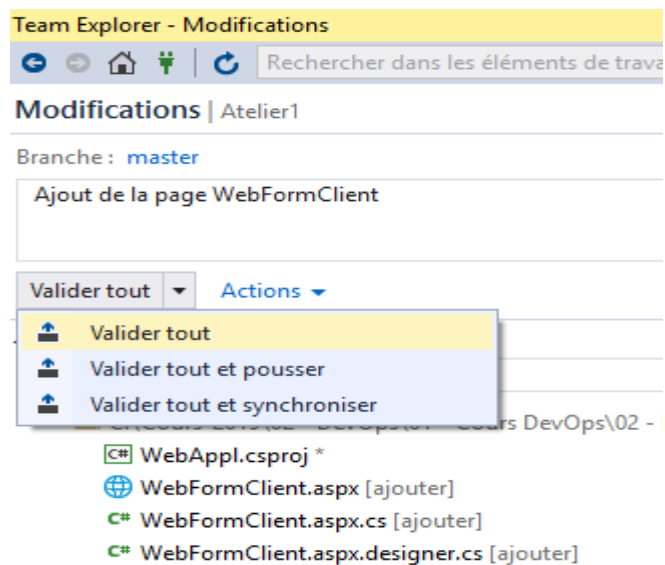
23. Vous pouvez revenir maintenant à la fenêtre Team Explorer, et choisir Modification, l'écran ci-dessous s'affiche avec la liste des modifications effectuées au niveau de votre solution, cela dit, que les modifications sont effectuées sur votre version locale à ce stade, et nous l'avons pas encore publier sur le serveur GitHub.



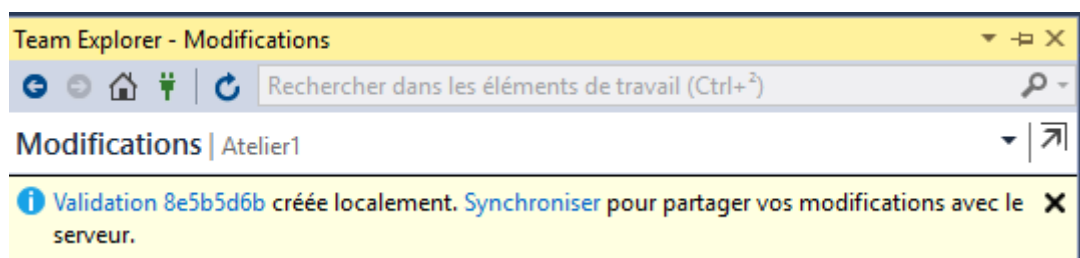
24. La première étape consiste à rajouter un message suite à chaque modification, cela aide sur le contrôle des versions par la suite, du coup, vous pouvez ajouter un message dans la zone « Entrez un message de validation <Obligatoire> », comme : « Ajout de la page WebForm »



25. Puis, cliquez sur « valider tout ».



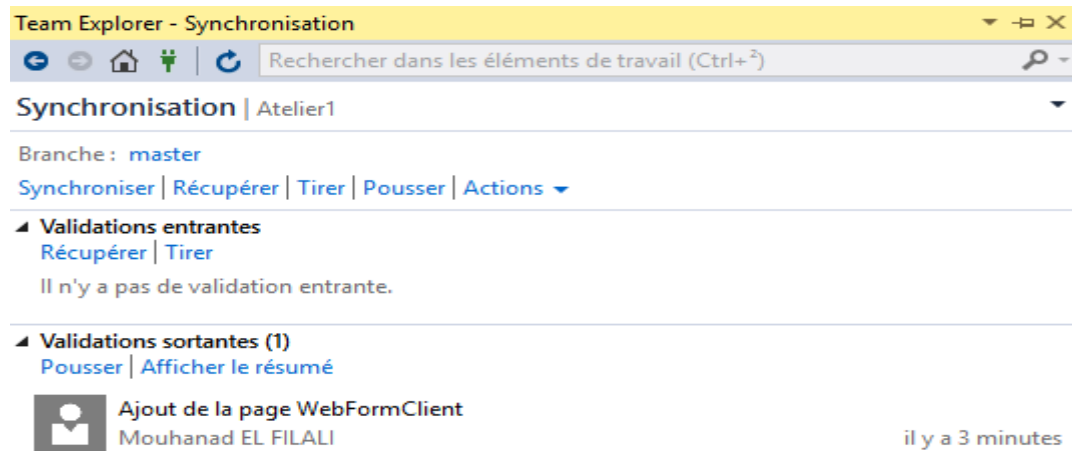
26. Le message suivant s'affiche.



27. nous pouvons maintenant faire une synchronisation de la version locale avec le serveur en cliquant sur « Synchroniser » ou le faire à partir de la fenêtre de Synchronisation.

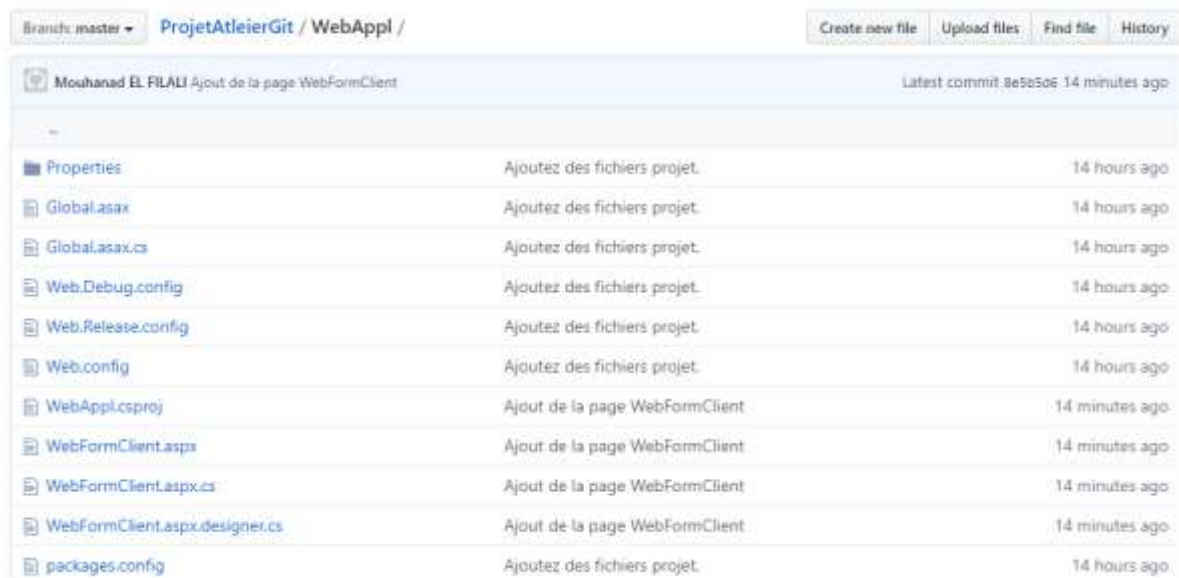
28. Vous allez remarquer que au niveau de la « Validation Sortants » nous avons un élément (1), cela dit, que nous avons un élément modifié sur la version locale, que nous pouvons Pousser sur la version serveur.

29. Cliquez par la suite sur « Pousser ».



30. Un message s'affiche par la suite « Push effectué correctement vers origin/master ».

31. Vous pouvez aller sur le portail GitHub, et faire une vérification pour s'assurer si la page WebFormClient a été bien publiée sur le serveur.



32. Vous pouvez voir l'historique des modifications effectuées sur la version, si vous cliquez sur le bouton « History » en haut à droite.

Mouhanad-Git / **ProjetAtleierGit**

Unwatch

1

Star

0

Fork

0

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

History for **ProjetAtleierGit** / **WebAppl**

Commits on Feb 7, 2020

Ajout de la page WebFormClient

Mouhanad El FICALI committed 24 minutes ago

8e5b5d6

Commits on Feb 6, 2020

Ajoutez des fichiers projet.

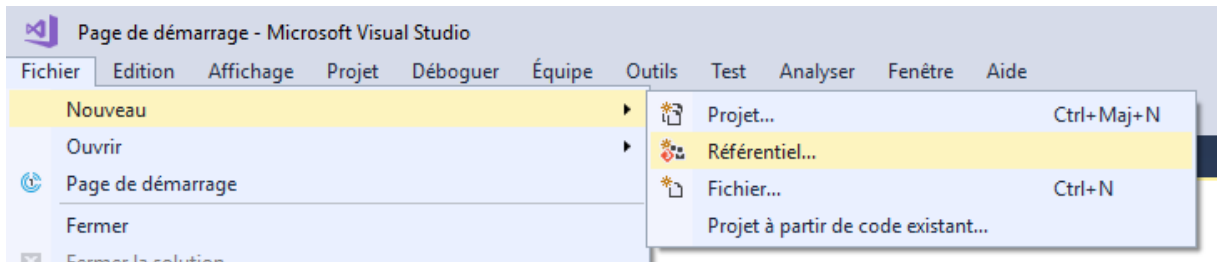
Mouhanad El FICALI committed 14 hours ago

83aa404

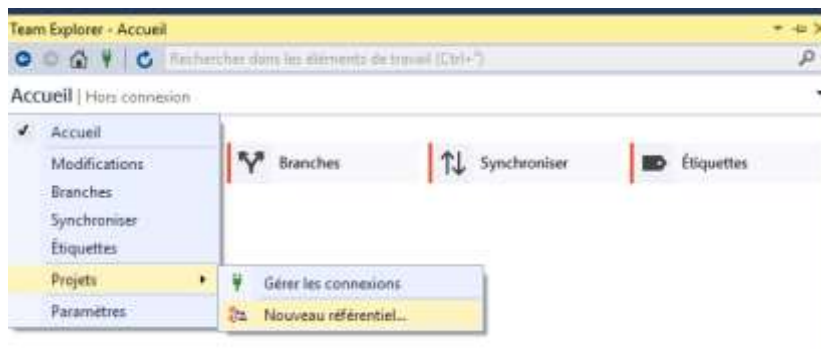
11

II. Maintenant, après que nous avons réussi à partager un nouveau projet .Net sur un contrôleur de code source (GITHUB), nous allons par la suite créer un projet avec VS à partir d'un code déjà publié sur GitHub, l'approche est différente de la première création. Merci de bien suivre les étapes suivantes.

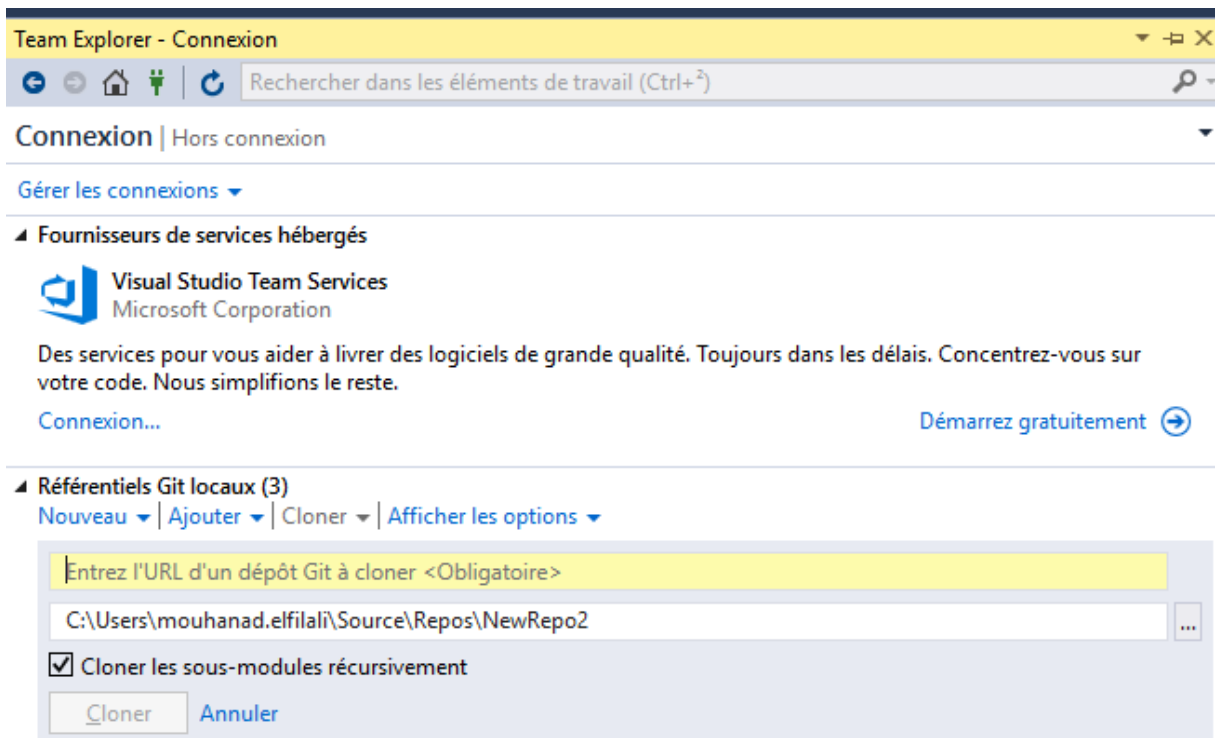
1. Lancez une nouvelle instance de Visual Studio pour la création d'un nouveau projet .Net, puis : Fichier => Nouveau => Référentiel



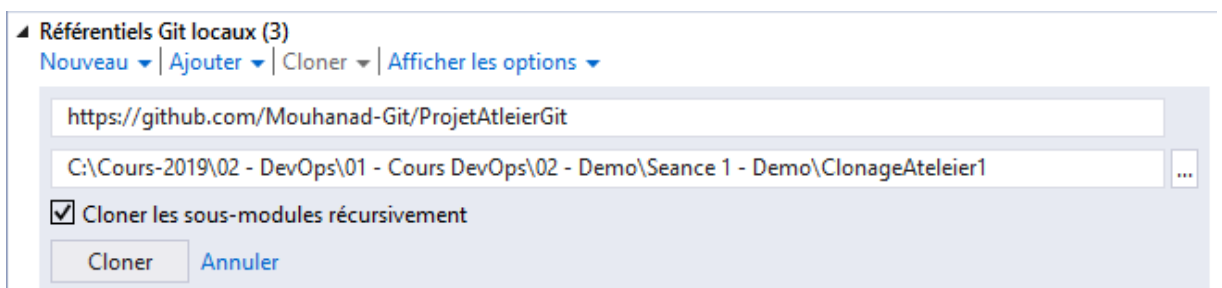
Ou à partir de la fenêtre Team Explorer, choisir Nouveau référentiel.



La fenêtre suivante s'affiche :



2. Cliquez sur le lien « Cloner » pour créer à la fois un répertoire Git Local à partir du projet publier sur le serveur GitHub, c'est-à-dire, on va procéder à un téléchargement de la version à partir du serveur vers le Git local.
3. Dans la zone « Entrez l'URL d'un dépôt Git à cloner », il faut mettre l'URL du repository sur le serveur de code source GitHub, dans mon cas, c'est « <https://github.com/Mouhanad-Git/ProjetAtleierGit> »
4. Donnez par la suite le lien vers un répertoire Git local pour télécharger votre version.
5. Puis cliquer sur le bouton Cloner.



6. Allez vérifier le nouveau dépôt git que vous avez créé sur votre machine, normalement, vous devez trouver une copie du projet récupérer à partir du GitHub sur votre machine locale.
7. Vous pouvez faire les manipulations faite dans la première partie de l'atelier 1.