**Git - Azure Repos – VS 2019**

1. Dans cet atelier, nous allons apprendre la création d’un projet avec Visual Studio et le partager via l’invite de commande Git client vers Azure DevOps.

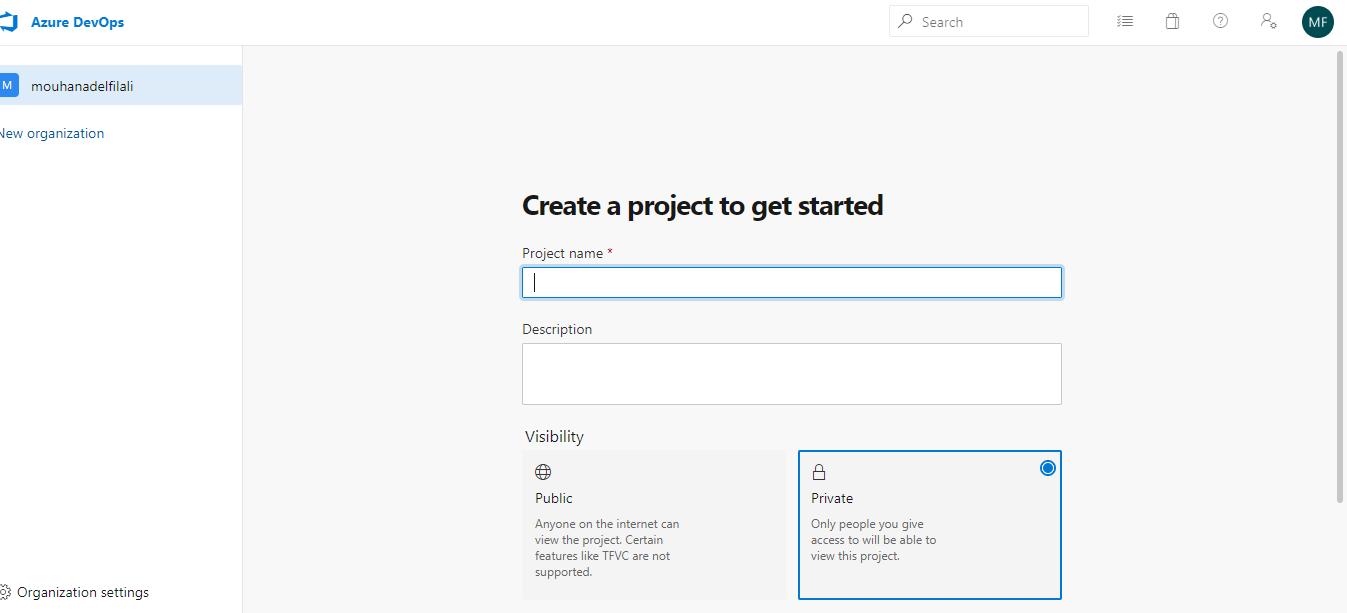
L’objectif est de créer un projet avec Visual Studio 2017/2019, créer un Git local (dépôt local) puis publier le Git local sur un serveur de contrôle de code source (qui sera Azure Repos).

Nous allons apprendre aussi dans cet atelier la gestion des branches (Ajout, Suppression et le merge entre branche).

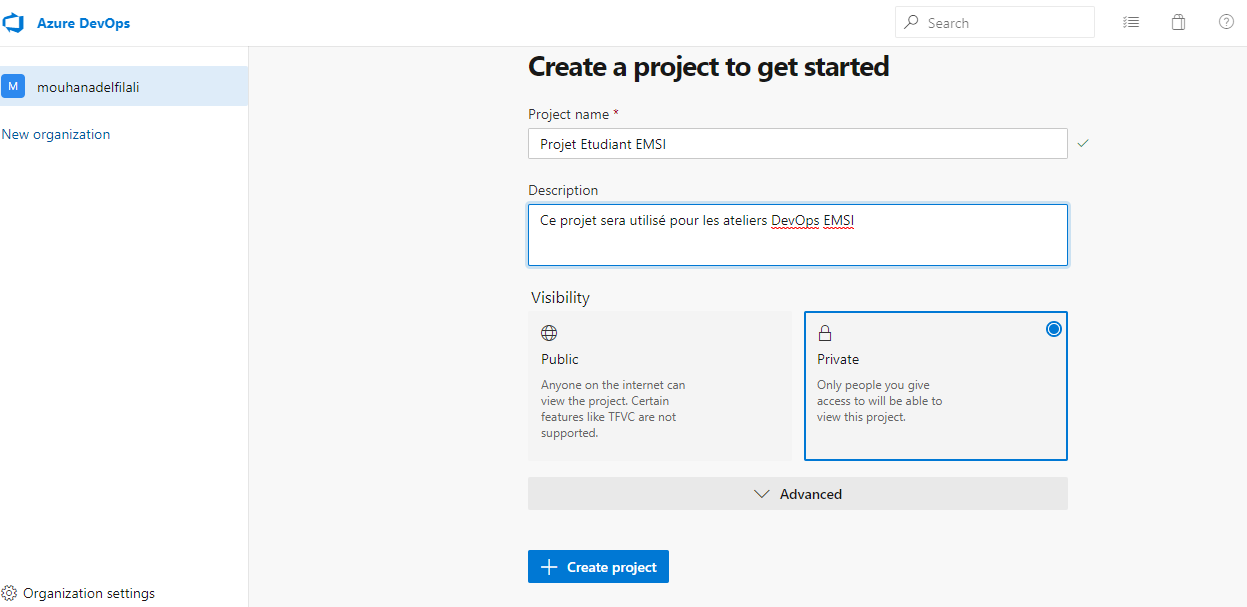
1. La première étape consiste à créer un compte sur le portail Azure DevOps : <https://dev.azure.com/>

NB : de préférence avoir une adresse email Microsoft pour pouvoir par la suite faire le lien entre VS et Azure DevOps.

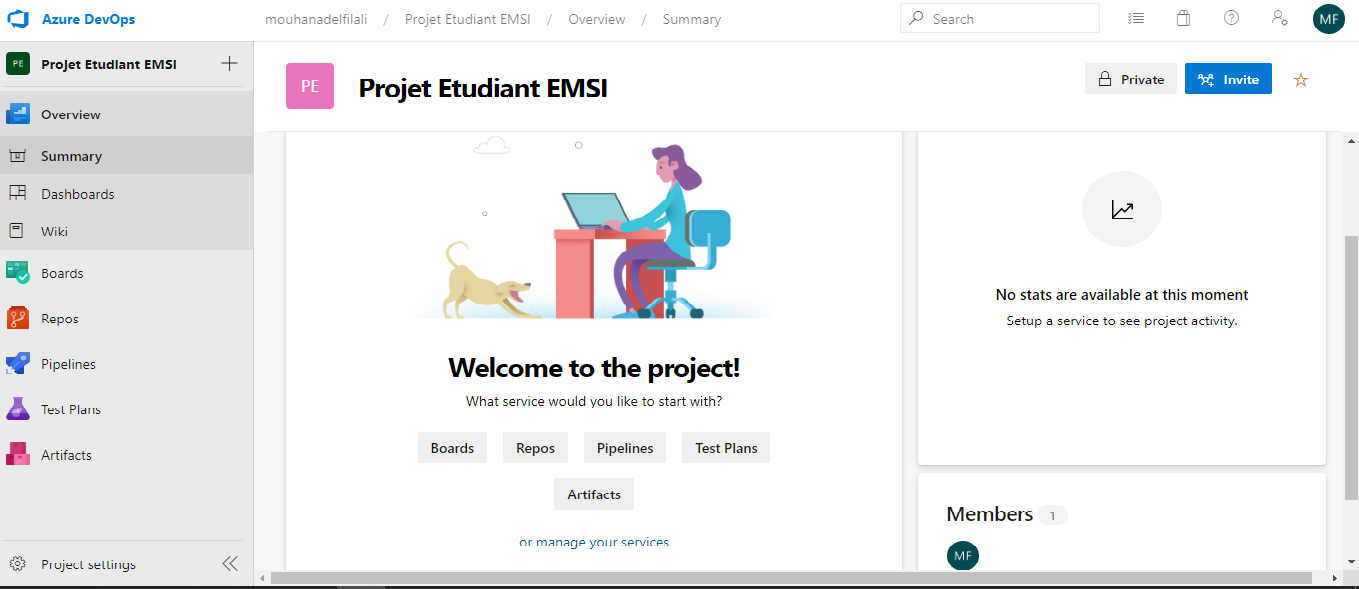
1. Connectez-vous par la suite avec votre compte et mot de passe, le premier écran doit être similaire à celui en bas, avant de créer un répertoire pour le partage du code source, il faut passer par la création d’une organisation (facultative), puis projet (Obligatoire) puis vous aurez a possibilité de créer votre Repos Devops (l’endroit où vous allez partager votre code source).



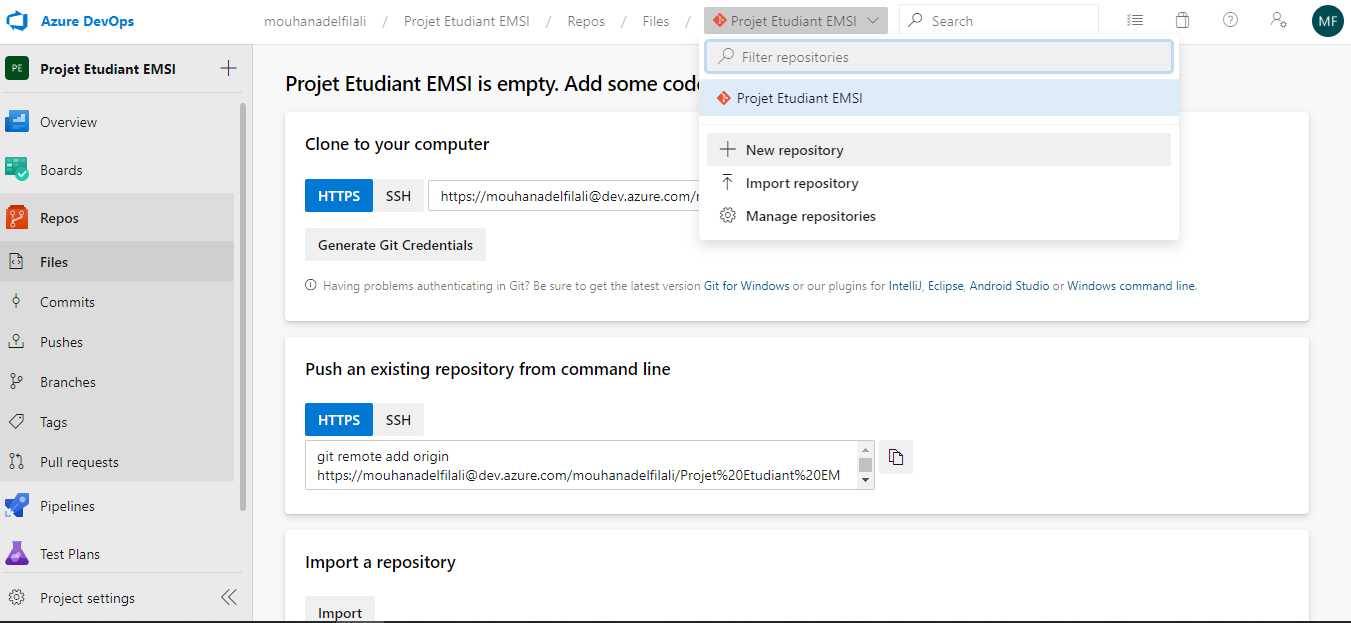
1. Commençons-nous par la création d’un projet :



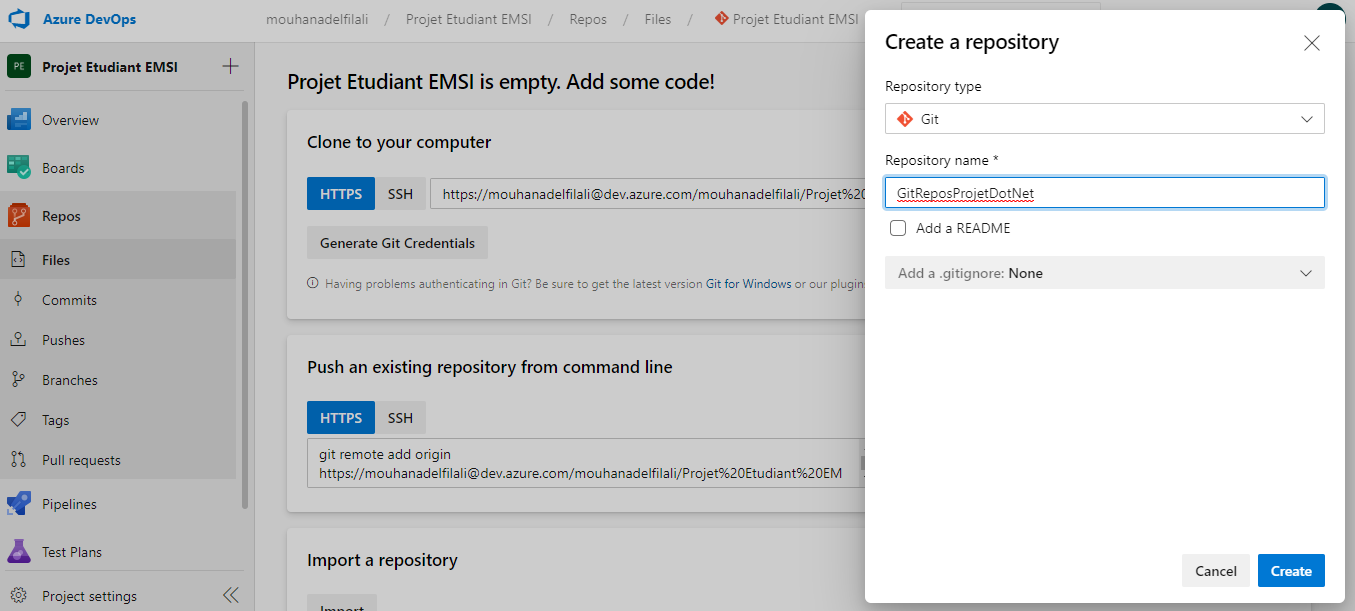
Voilà à quoi ça ressemble la création du projet :



1. Une fois le projet est créé, la deuxième étape consiste à créer un Repos DevOps, Cliquez sur le bouton Repos en bas ou sur l’onglet Repos à gauche de l’écran, puis choisissez l’option New Repository au niveau du nom de votre projet (en haut de la page Repos).



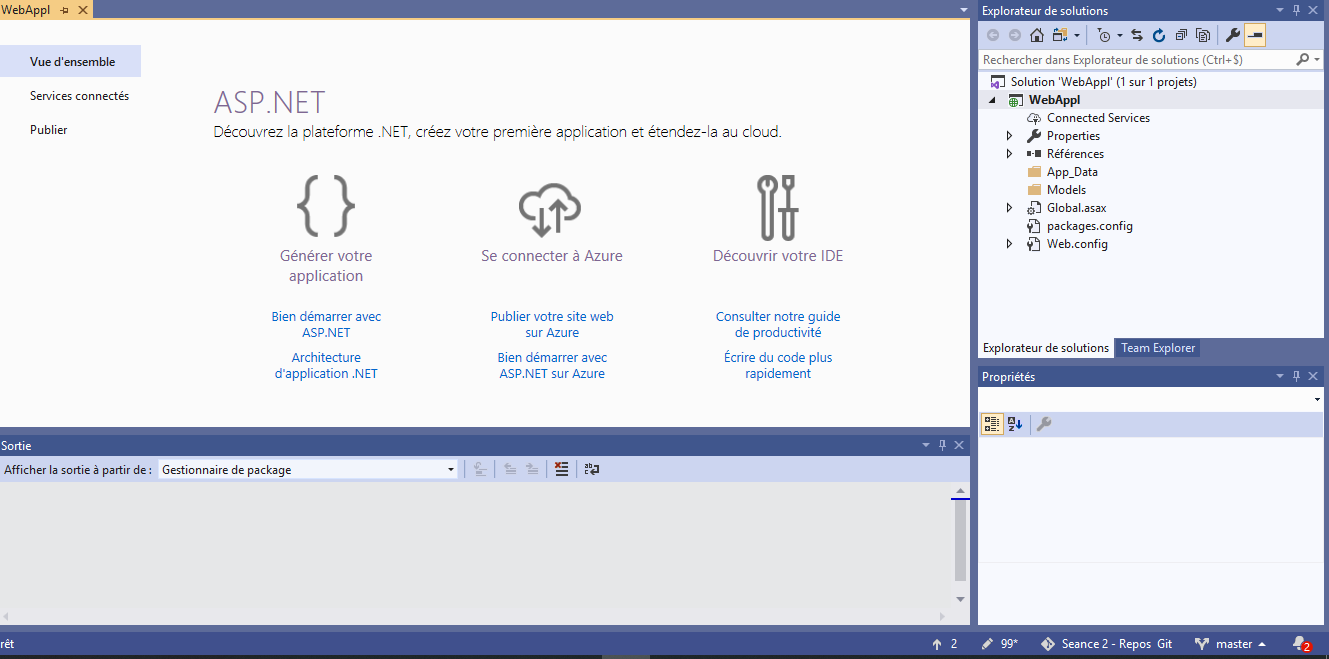
1. Donnez un nom à votre Repos puis cliquez sur le bouton Create.



Etape suivante consiste à créer un répértoire local (Git local) qui va contenir notre projet (Solution Visual studio) qui qu’on va par la suite publier sur Azure DevOps.

Nous devons installer sur la machine local Git Command Line (<https://git-scm.com/download/win>).

1. Créez un répertoire local dans lequel la solution sera déposée.
2. Une fois c’est fait, lancez Visual Studio pour la création d’un nouveau projet Web Form qui doit avoir comme localisation votre dossier local crée en étape 6 de cet atelier.



1. Ouvrez la fenêtre de commande Git.

****

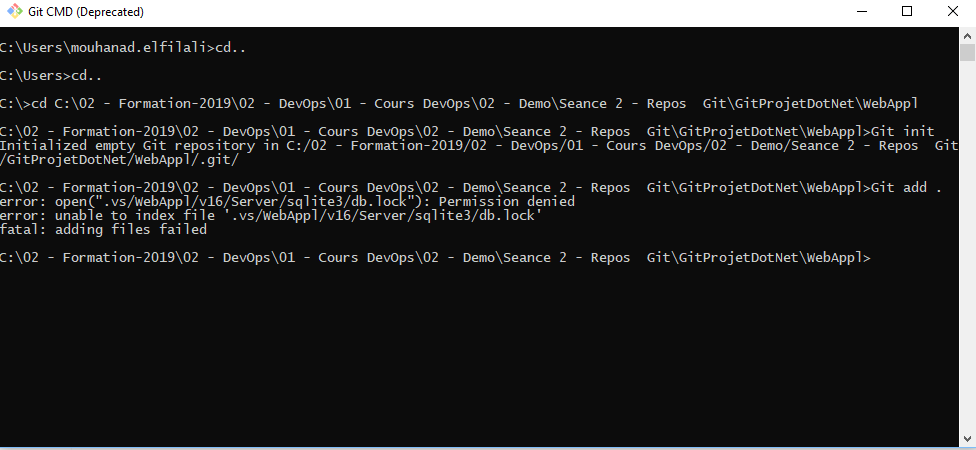
1. Accédez au dossier où vous souhaitez que le code sera stocké sur votre ordinateur
2. Exécutez les commandes suivantes :

* git Init

Cette commande créer un Git local (c’est le répertoire que vous avez créé sur votre machine lcoale).

* git add .

Cette commande ajoute les fichiers créer dans votre Git local.



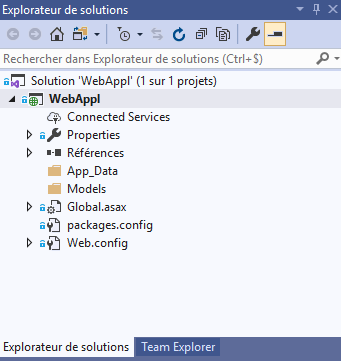
NB : Si vous avez une erreur d’exécution de la commande Git Add, fermez VS.

* git commit -m "First commit"

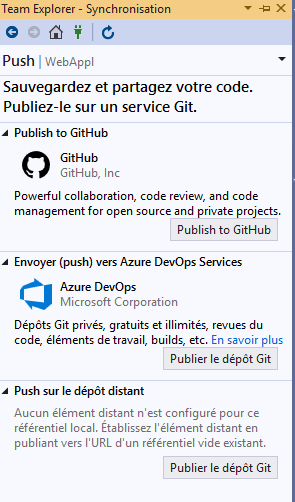
Validez les fichiers que vous avez transférés dans votre référentiel local.

1. Ouvrez votre projet à partir de VS.

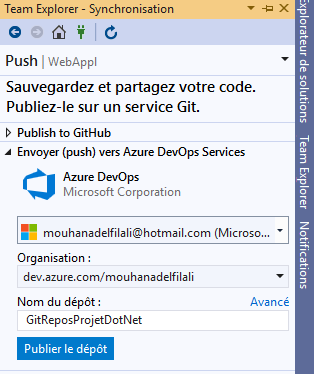
NB : notez bien l’ajout des cadenas devant chaque fichier.



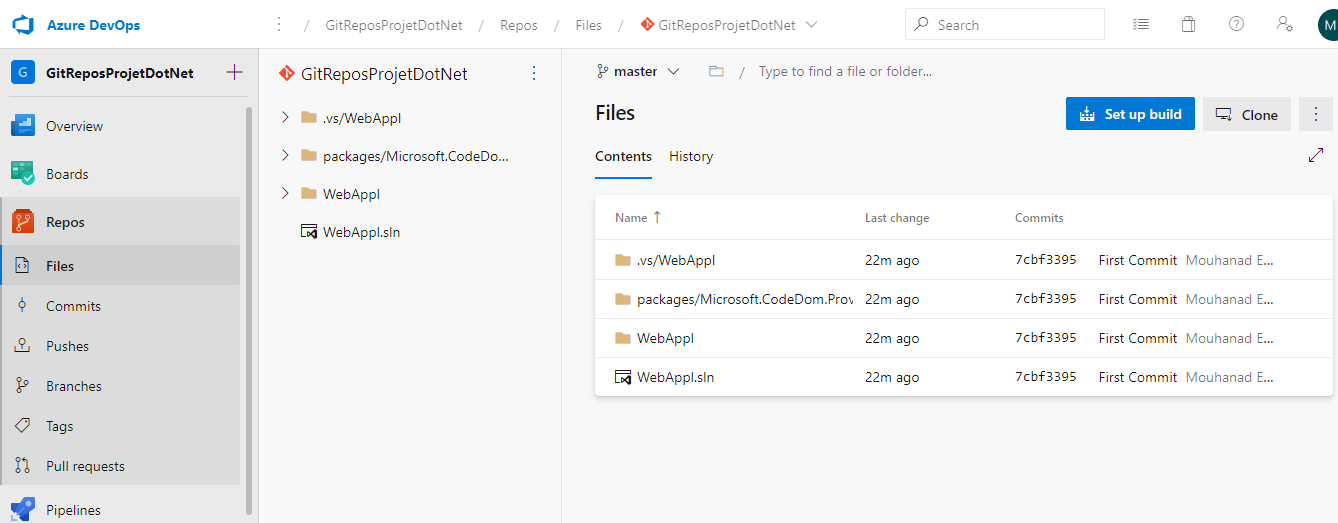
1. Allez à la fenêtre Team Explorer, puis choisissez Synchroniser.
2. L’objectif est de synchroniser sur Azure DevOps, choisissez l’option « Publier le Dépôt Git » de Azure Devops Services .



1. Renseignez l’ensemble des informations qui concernent votre compte Azure, Votre Organisation/Projet et le nom de votre Repos Azure Devops que nous avons créé en étape 4 de cet atelier.

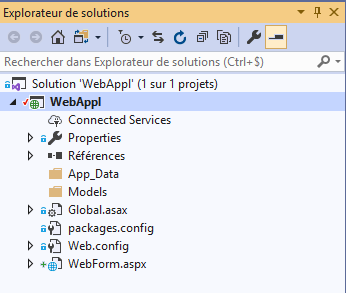


1. Puis cliquez sur Publier le dépôt.
2. Une fois la publication s’est bien passé, vous devez avoir un message : « La branche master a fait l'objet d'un Push réussi dans origin. Créez une demande de tirage (Pull request) pour passer en revue vos changements. ».
3. Vous pouvez revenir au portail Azure, vous allez trouver que votre code source a été bien partagé sur le Repos Azure Devops.

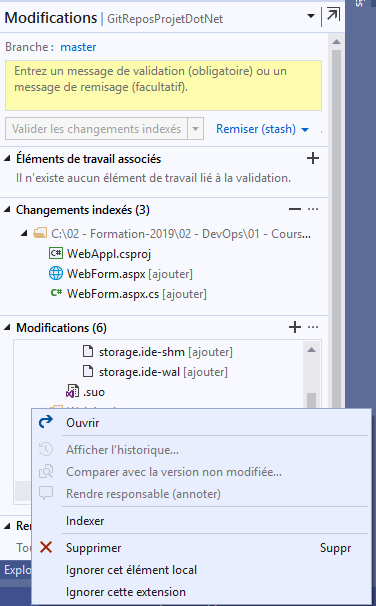


Après que vous avez réussi à mettre votre projet sur Azure Devops, nous allons procéder à une modification au niveau de notre projet, l’ajout d’une page web form.

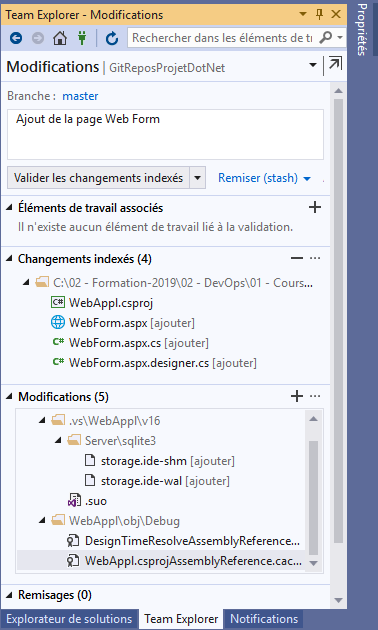
1. Cliquez sur bouton droit au niveau de votre solution, Ajoutez une nouvelle page, la page ajoutée apparait avec un « + ».



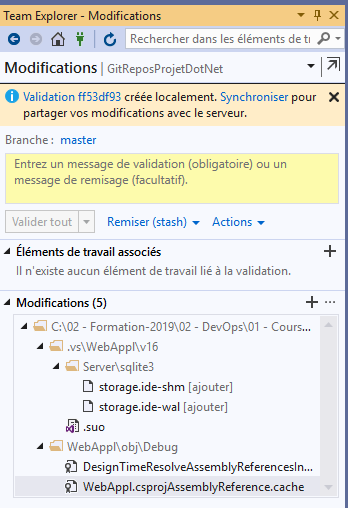
1. Vous pouvez revenir à l’invite de commande Git, et répéter les commandes Git Add, puis Commit, sinon vous pouvez passer par la fenêtre Team Explorer, vu que nous avons déjà étudié le push avec les commandes Git, nous allons procéder à le faire maintenant à partir de la fenêtre Team Explorer.
2. Allez à la fenêtre Team Explorer, puis choisissez l’option Modification.



1. Cliquez sur les éléments que vous souhaitez envoyer sur le serveur (bouton droit), puis choisissez l’option Indexer, c’est l’équivalent de la commande Git Add . , les éléments indexés seront sélectionnés dans la zone « Changements indexés (3) ».
2. Rajoutez un commentaire (c’est obligatoire), puis cliquez sur « Valider les changement indexés », c’est l’équivalent à la commande « Commit ».



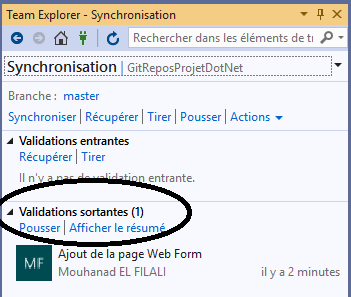
1. Vous allez recevoir le message suivant :



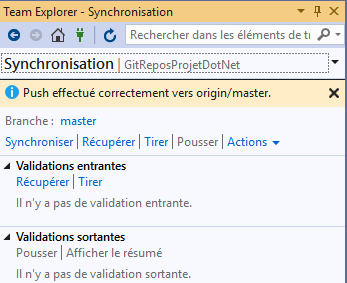
1. Maintenant, nous allons envoyer les modifications vers le serveur Azure, allez à la fenêtre « Synchroniser » au niveau du Team Explorer.
2. Vous allez remarquer que vous avez des modifications sortantes (Validations Sortantes), cliquez sur le bouton « Pousser ».

NB : vous pouvez effectuer aussi cette manipulation par l’invite de commande Git, la commande Git Push va faire un Pull des modification vers la branche master du serveur Azure Devops :

* git push origin master



1. Un message de confirmation s’affiche (Push effectué correctement …..).

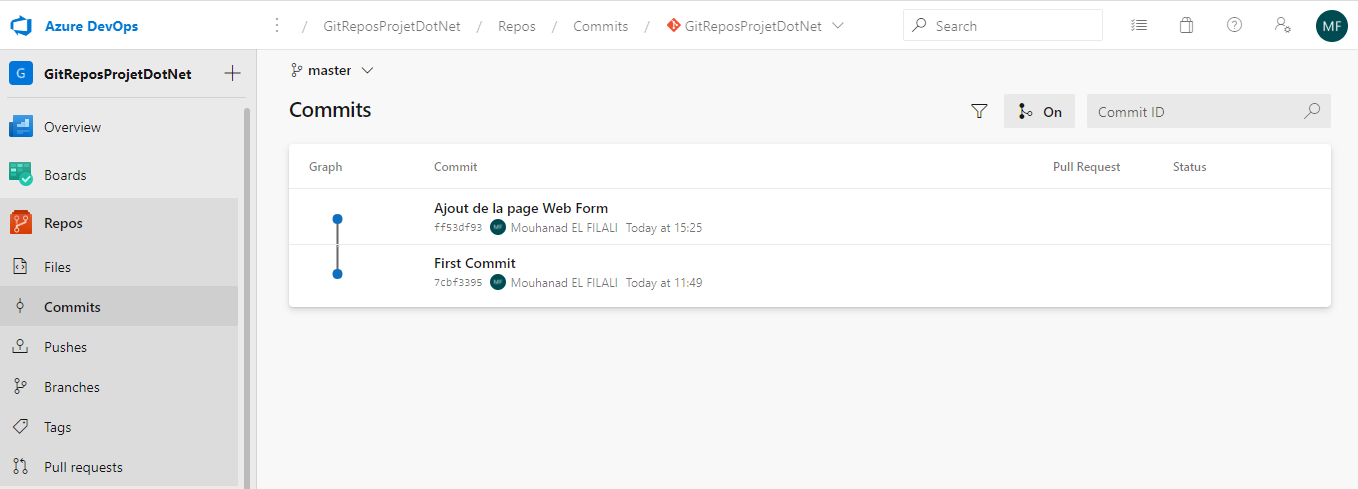


1. Vous pouvez revenir à votre Repos DevOps pour vérifier l’ajout de la nouvelle page dans le code source partagé.

Si vous cliquez sur l’onglet « Commits » vous allez trouver les commits effectués sur votre projet avec les commentaires :

Le premier commentaire concerne la commande Git commit –m ….. que nous avons effectué en étape 10 de cet atelier.

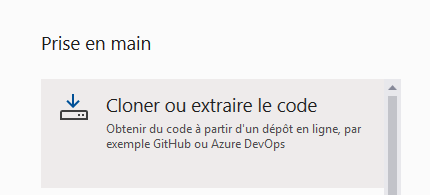
Le deuxième commentaire concerne l’ajout de la page effectué par la fenêtre team explorer, l’étape 22 de cet atelier.



1. La deuxième étape de cet atelier va servir à faire un clonage d’un projet à partir d’Azure Devops vers un répertoire local de votre machine.

L’opération de clonage consiste récupérer un code d’un contrôleur de code source vers un Git local.

1. Sur l’écran de lancement de visual studio 2019, vous allez trouver une option « Cloner ou extraire le code ».

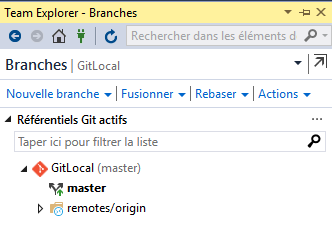


1. Mettez l’emplacement de votre Repos Azure DevOps puis le chemin local de votre Git local.
2. Puis cliquez sur le bouton « Cloner ».

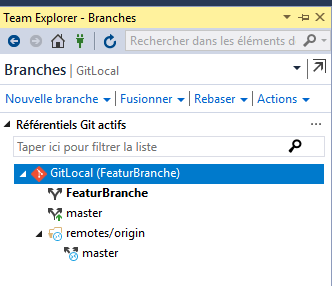


La solution est récupérée sur votre machine localement à partir du Repos DevOps.

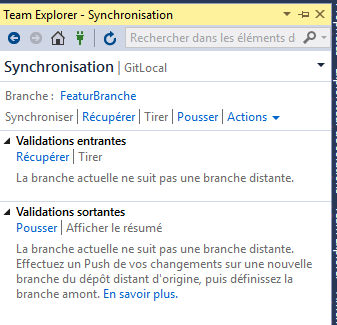
1. La troisième étape consiste à créer une nouvelle branche pour suivre les modifications sur cette dernière et faire un merge sur la branche master.
2. A partir de Visual studio, ouvrez la fenêtre Team Explorer.
3. Allez à « Branches ».



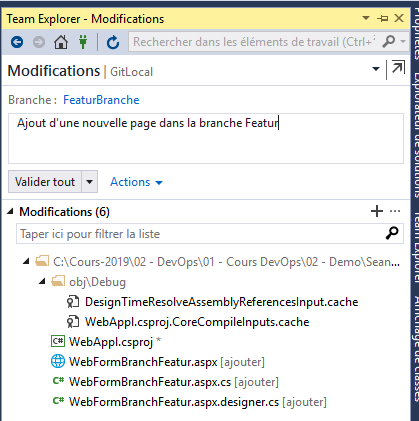
1. Puis, cliquez sur Nouvelle branche à partir de la branche master.
2. Donnez un nom à votre branche, dans mon exemple, c’est « FeatureBranch ».



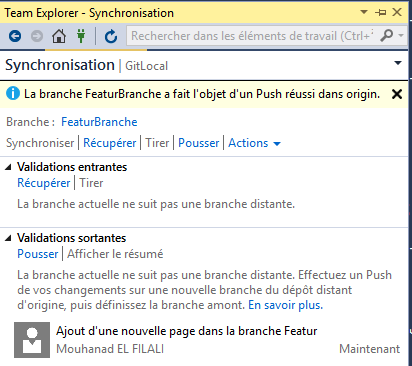
1. Vous pouvez vérifier maintenant que votre projet a basculé de la branche « master » vers la nouvelle branche qu’on vient de créer.



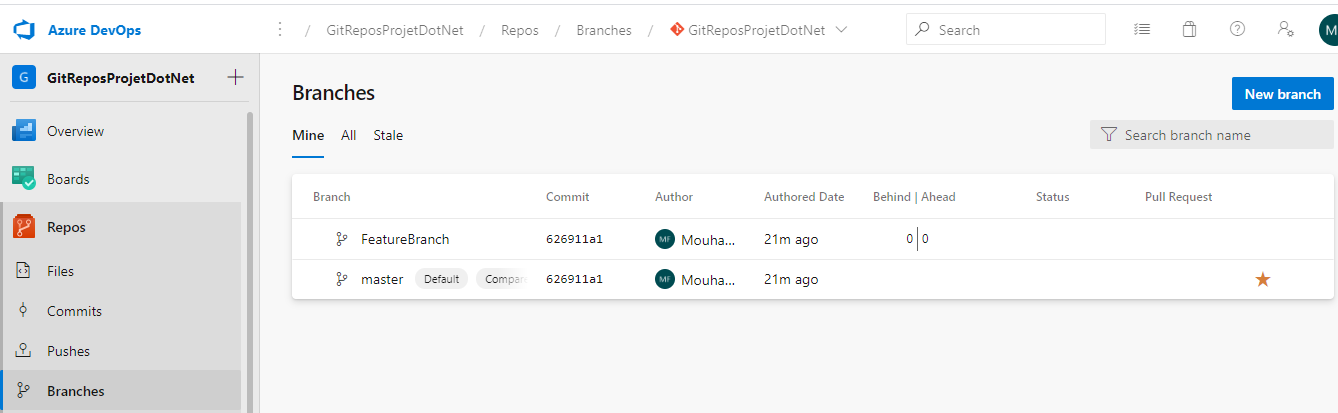
1. Ajoutez une nouvelle page dans votre projet Visual Studio, et faire un push vers le serveur Azure DevOps.



1. Ajoutez une nouvelle page dans votre projet Visual Studio, et faire un push vers le serveur Azure DevOps.



1. Si vous allez maintenant sur le portail Azure DevOps, au niveau de votre Repos, vous allez remarquer l’ajout de la nouvelle branche créée à partir de Visual Studio.



1. Basculez à la nouvelle Branche créée, vous allez trouver la page rajoutée dans votre solution VS.

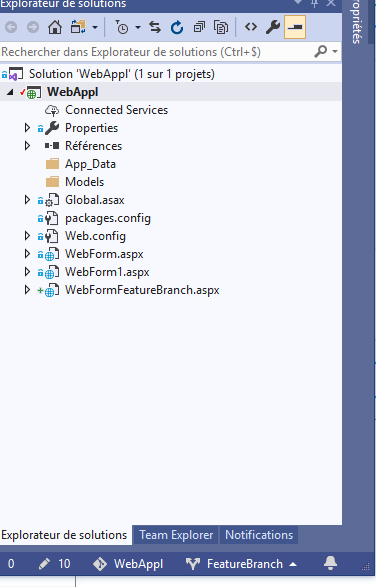
Cela dit, que vous avez créé une nouvelle branche à partir de la branche master et vous continuez le développement sur cette nouvelle branche, à n’importe quel moment, vous pouvez faire un merge vers la branche master pour reporter les développements que vous avez fait sur la nouvelle branche « FeatureBranch ».

1. Merge de branche

Vous pouvez poursuivre vos modifications dans la nouvelle branche créée dans la 3eme partie de cet atelier……..

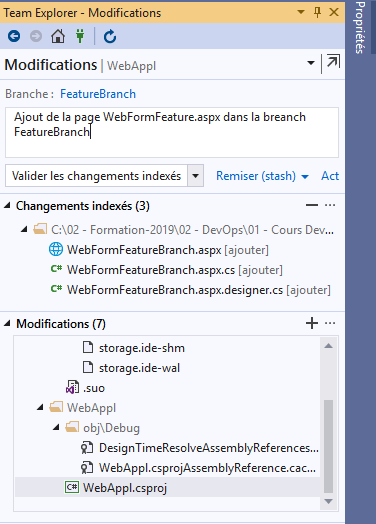
Dans l’exemple ci-dessous, nous avons rajouté une nouvelle page « WebFormFeatureBranch ».

NB : Vous pouvez remarquer en bas de la capture d’écran, le détail des modifications effectuées, la branche utilisée et le chemin vers le dépôt Git local.

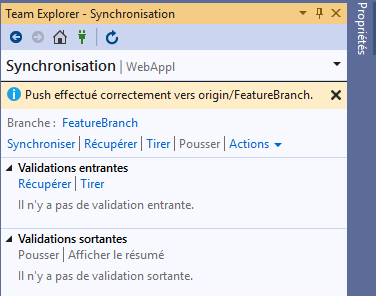


1. Vous pouvez envoyer à tout moment vos modifications vers la branche FeatureBranch sur Azure DepOps, soit par Git command line, soit à partir de la fenêtre Team Explorer.

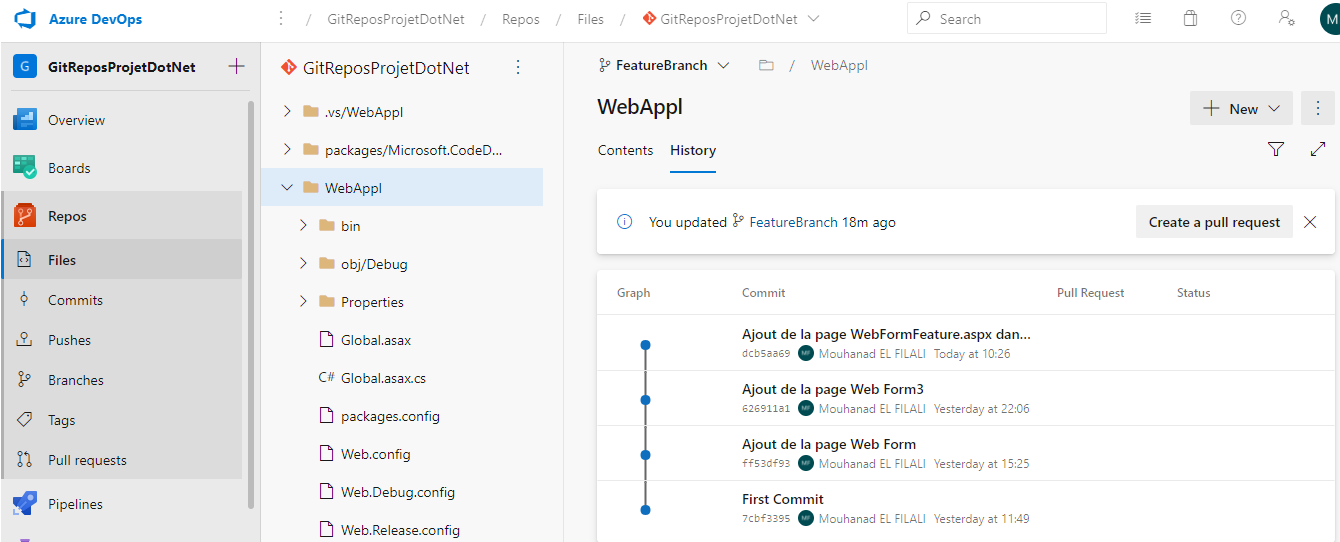
NB : n’oubliez pas de rajouter un commentaire à chaque modification.



1. Puis, Push vers le serveur Azure DevOps.

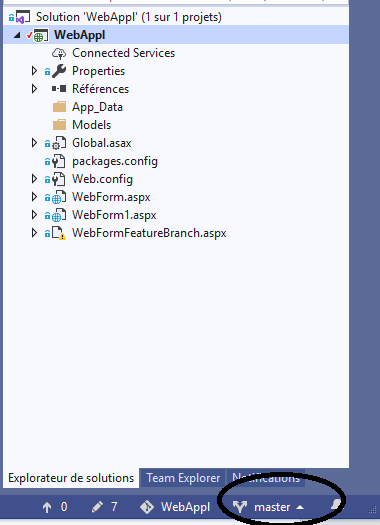


1. Vous pouvez à tout moment vérifier les modifications apportées sur votre nouvelle branche « FeatureBranch » sur le Repos Azure DevOps.



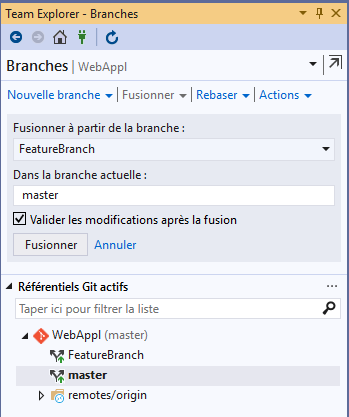
Une fois vos modifications sur la branche Feature sont arrêtés, et vous avez décidé de faire un merge sur la branche master, vous pouvez le faire soit par Git command line, soit à partir de la fenêtre Team Explorer.

1. Basculez maintenant votre solution VS sur la branche master, vous pouvez le faire, en changeant la branche de FeatureBranch à master.

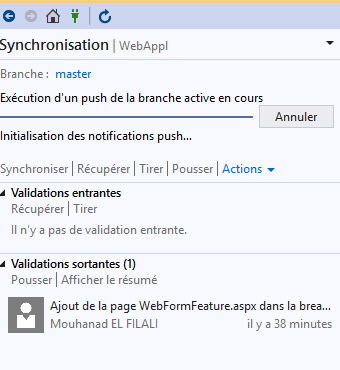


Vous allez remarquer un warning devant la page que nous avons créé dans la branche FeatureBranch, chose normale, cas cette page n’existe pas dans la branche master.

1. Passez à la fenêtre Team Explorer, Branches, puis sélectionner la branche master, bouton droit et choisissez « Fusionner à partir de ».
2. Cliquez sur le bouton « Fusionner ».

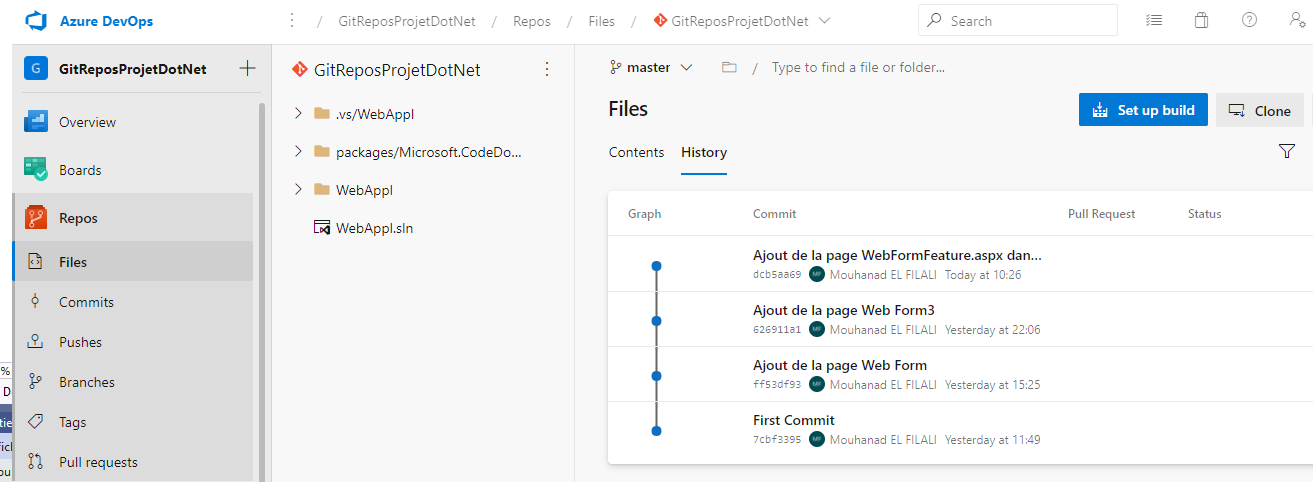


1. Puis, faites un Push vers la branche master, un message de confirmation s’affiche après l’opération de push « Push effectué correctement vers origin/master. »

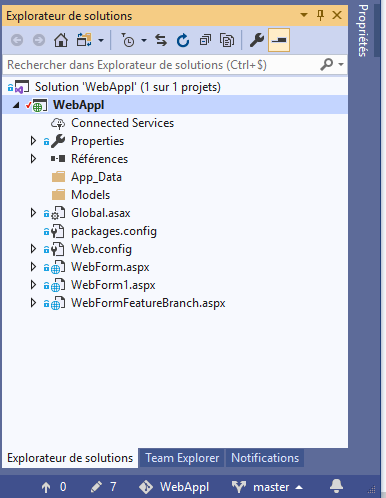


1. Allez à votre Repos Azure DevOps, sélectionnez la branche master et regardez l’historique, vous allez remarquer l’ajout de la page WebFormFeatureBranch.

Vous venez de faire un merge entre les branches featureBranch et master.



1. Au niveau de votre solution VS de la branche master, vous allez remarquer que le warning affiché devant la page WebFormFeatureBranch a disparu.



1. Suppression d’une branche
2. Pour supprimer une branche, allez à la fenêtre Branche du Team Explorer, et cherchez la branche en question.
3. Sélectionnez la branche que vous voulez supprimer, puis cliquez sur le bouton droit, puis choisissez « Supprimer ».

NB : Vous ne pouvez pas supprimer une branche sur laquelle vous êtes connecté, ci-dessous l’exemple.

