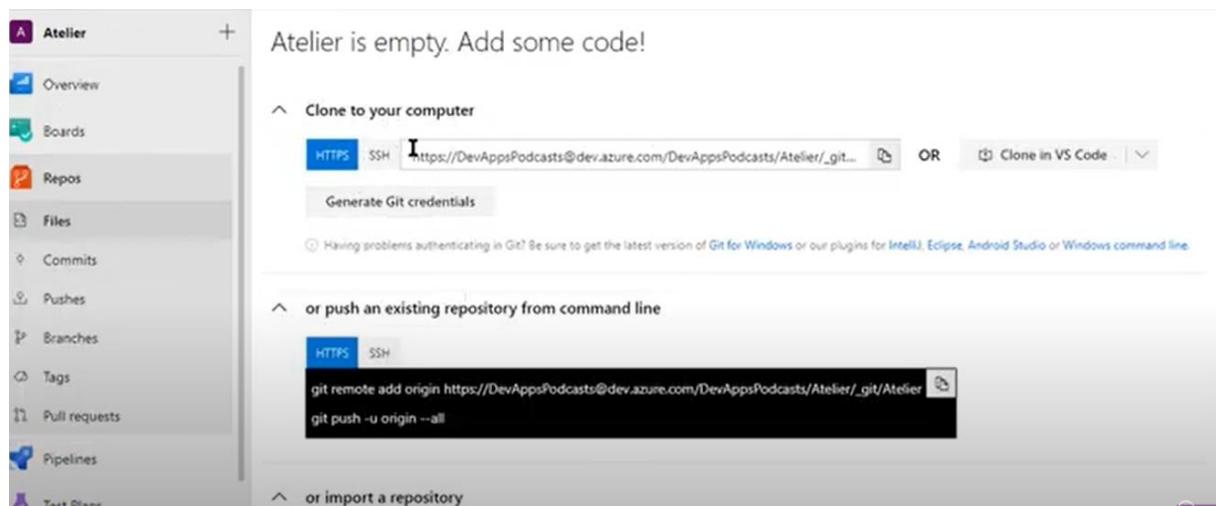


## TP2 : Azure Repos (GIT)

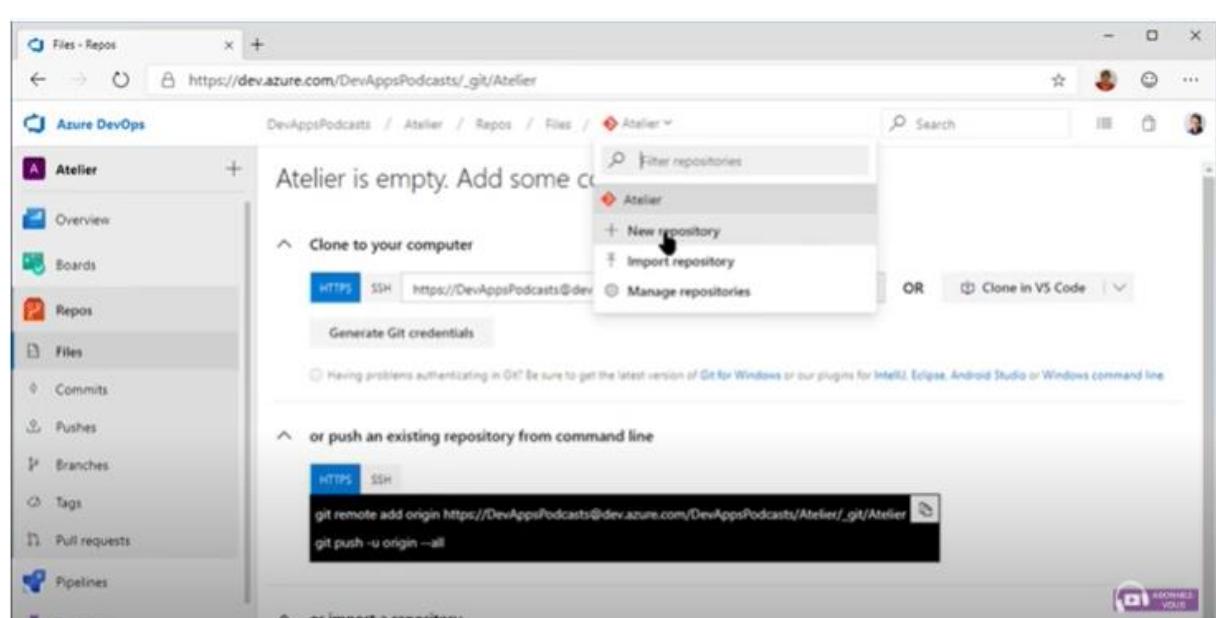
Dans ce TP on va pourvoir utiliser GIT avec la plateforme Azure.

En premier lieu il faut se connecter (utiliser vos comptes microsoft) sur le site web <https://dev.azure.com>.

Dans l'onglet à gauche, choisissez Repos afin de créer votre repository Git. Par défaut, il y a un répertoire qui est déjà créé. Mais vous pouvez toujours créer un nouveau dépendamment de votre besoin.



The screenshot shows the 'Atelier' repository page in Azure DevOps. The left sidebar is a navigation menu with items like Overview, Boards, Repos (which is selected), Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, and Test Plans. The main content area has a heading 'Atelier is empty. Add some code!' and three sections: 'Clone to your computer' (HTTPS and SSH links, 'Generate Git credentials' button, note about authentication), 'or push an existing repository from command line' (code snippets for git remote add origin and git push -u origin --all), and 'or import a repository'. A context menu is open over the 'Atelier' repository name, showing options: 'New repository' (selected), 'Import repository', and 'Manage repositories'.



The second screenshot is identical to the first, except it shows a context menu open over the 'Atelier' repository name. The 'New repository' option is highlighted with a black arrow. The rest of the interface is the same, showing the repository details and cloning/importing options.

Définissez le nom du répertoire, et cochez la case « add a README » ce qui fait partie des bonnes manières.



The screenshot shows the Azure DevOps interface for a project named "test project". On the left, there's a sidebar with options like Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, and Artifacts. The "Files" tab is selected. In the main area, there's a file named "MI README.md". A modal dialog titled "Create a repository" is open on the right. It has a "Repository type" dropdown set to "Git", a "Repository name" input field containing "test", and a checked checkbox for "Add a README". Below these, there are sections for "Introduction", "Getting Started", "Build and Test", and "Contribute". At the bottom right of the modal are "Cancel" and "Create" buttons.

Vous pouvez également choisir la liste des fichiers à ignorer.

Une fois votre répertoire créé, il faut le cloner sur votre machine.

The screenshot shows a Windows desktop environment. The taskbar at the bottom includes icons for File Explorer, Edge browser, Mail, Task View, and others. The system tray shows the date as "07/03/2021" and the time as "19:08". The main window is the Azure DevOps interface for the "test" repository. It displays the same structure as the previous screenshot, with the "Files" tab selected and the "MI README.md" file visible. The "Create a repository" dialog from the previous screenshot is still open in the background.

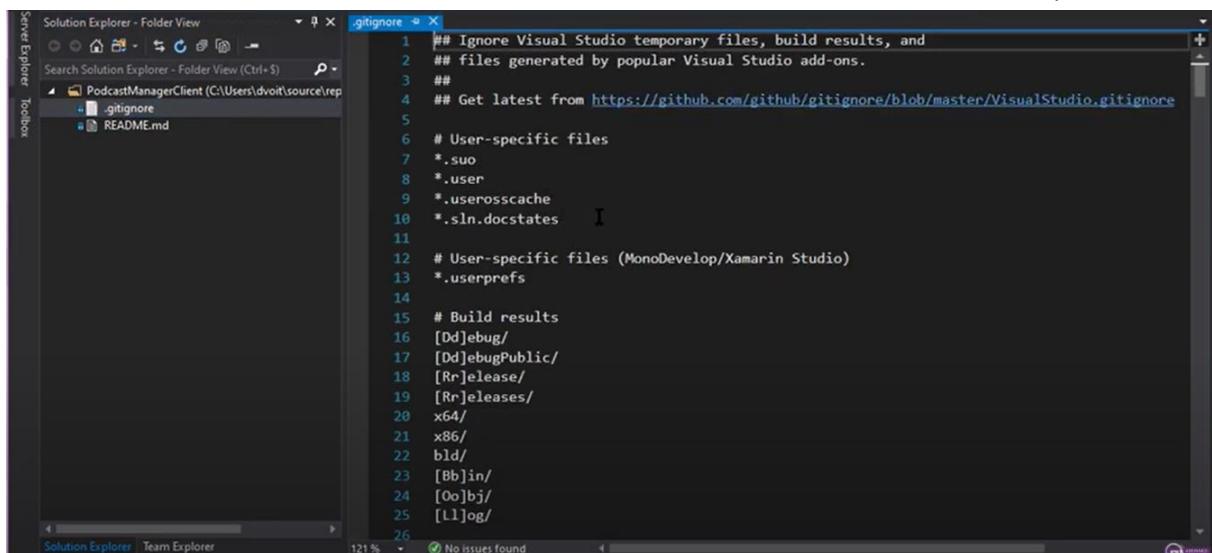


The screenshot shows a Microsoft Edge browser window displaying the Azure DevOps interface for a project named "gestion de client". The left sidebar shows navigation options like Overview, Boards, Repos, Files, Pipelines, Test Plans, and Artifacts. The main area shows a file named "MI README.md". A modal dialog box titled "Clone Repository" is open, showing cloning options via "Command line" (HTTPS selected) or "SSH", and an "IDE" section with "Clone in VS Code". The status bar at the bottom indicates the date as 07/03/2021 and the time as 19:09.

This screenshot is identical to the one above, showing the same Azure DevOps interface and the "Clone Repository" dialog box. The URL in the address bar is dev.azure.com/PCHLIOUIIMANE/gestion%20de%20client/\_git/test. The status bar at the bottom indicates the date as 07/03/2021 and the time as 19:09.

Vous aurez deux choix, soit le cloner dans une IDE choisie soit copier le lien et aller sur le git bash et exécuter la commande **Git clone « lien copié »**

Au cas où vous choisissez de le cloner directement dans un IDE (par exemple : Visual studio), il vous affichera directement votre répertoire sur visual studio. Par la suite vous pouvez travailler directement sur visual studio.

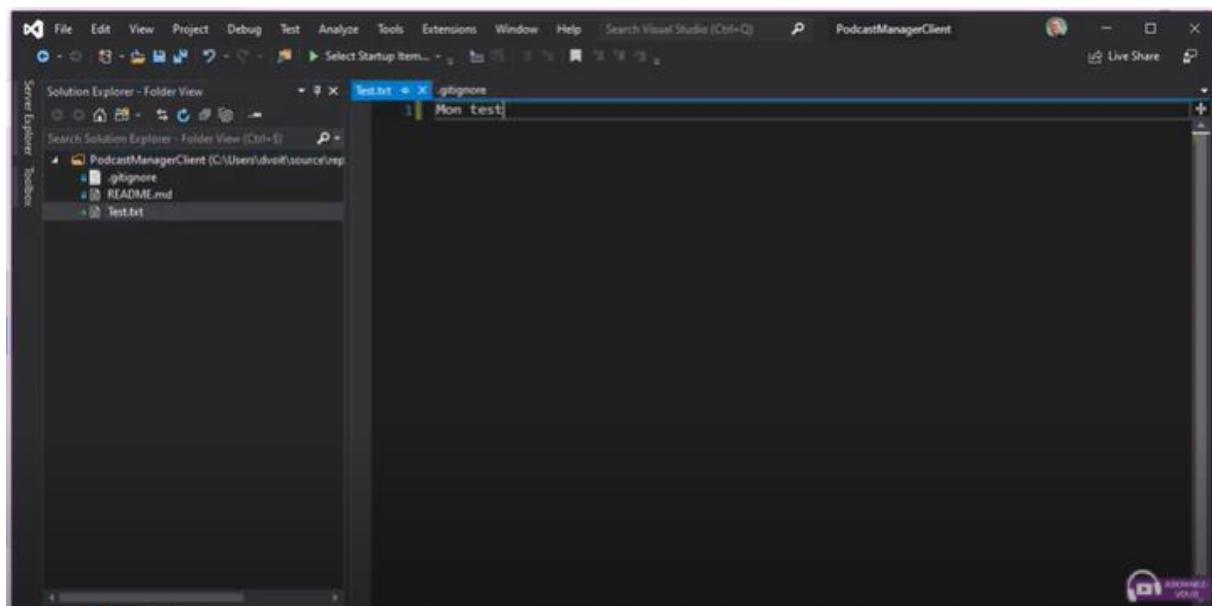


```
## Ignore Visual Studio temporary files, build results, and
## files generated by popular Visual Studio add-ons.
##
## Get latest from https://github.com/github/gitignore/blob/master/VisualStudio.gitignore
#
# User-specific files
*.suo
*.user
*.userosscache
*.sln.docstates

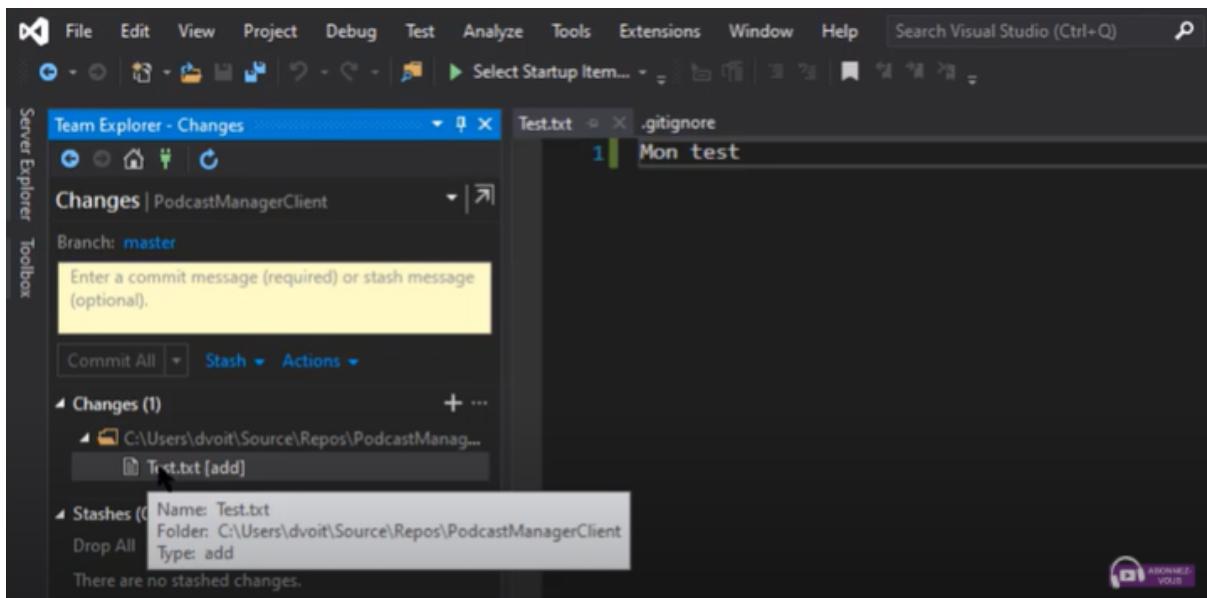
#
# User-specific files (MonoDevelop/Xamarin Studio)
*.userprefs

#
# Build results
[Dd]ebug/
[Dd]ebugPublic/
[Rr]elease/
[Rr]eleases/
x64/
x86/
bld/
[Bb]in/
[Oo]bj/
[Ll]og/
```

Si vous ajoutez un nouveau fichier (test), il sera automatiquement placé dans le paquet du changement.

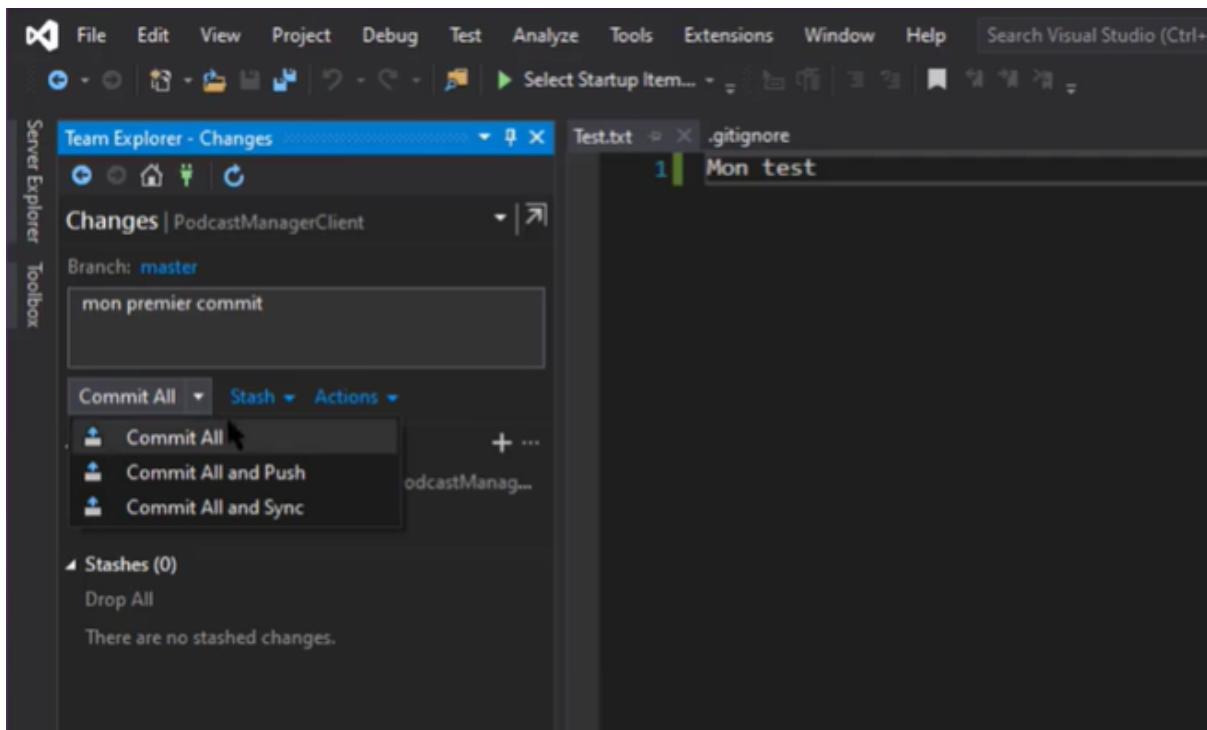


Dans la team explorer vous pouvez visualiser tous les fichiers ajoutés prêts à être commités.

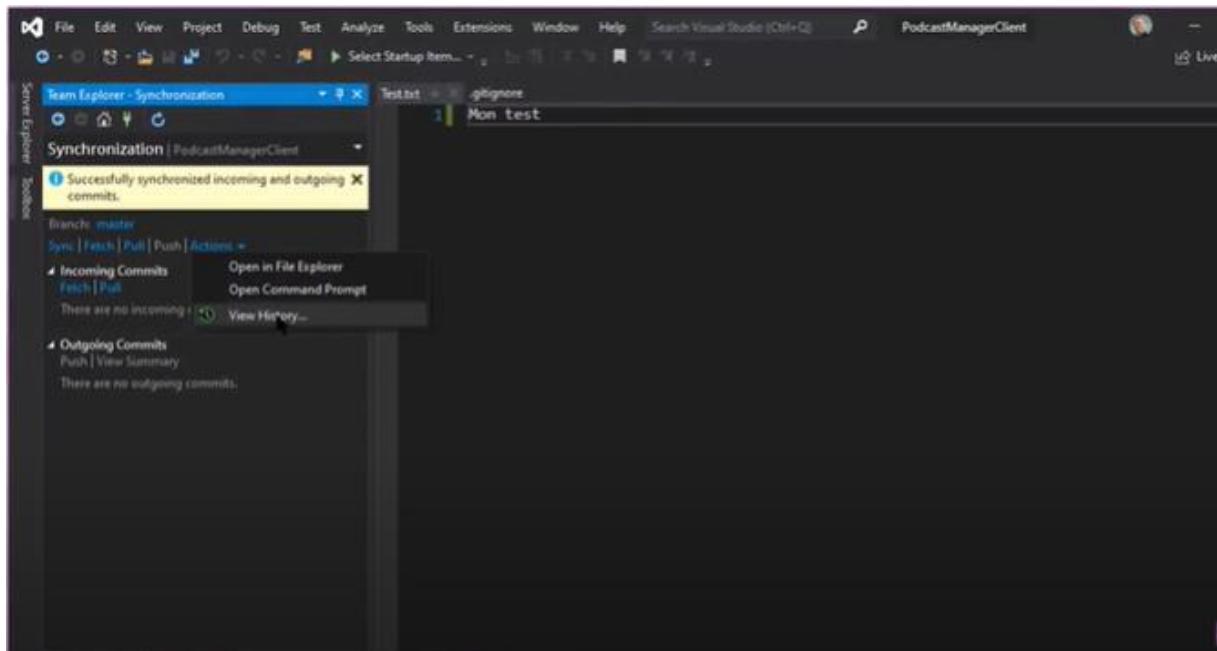


Par la suite, il vous suffira d'ajouter un commentaire et ensuite choisir quel type d'action à faire :

- Commit all : pour commiter localement
- Commit all and push : pour commiter localement et en ligne
- Commit all and Sync : pour synchroniser et commiter (Pull, commit local et ensuite le push)



Vous trouverez toutes les actions possibles à faire que ça soit un push, pull, fetch .... Et bien sur vous avez accès toujours à l'historique

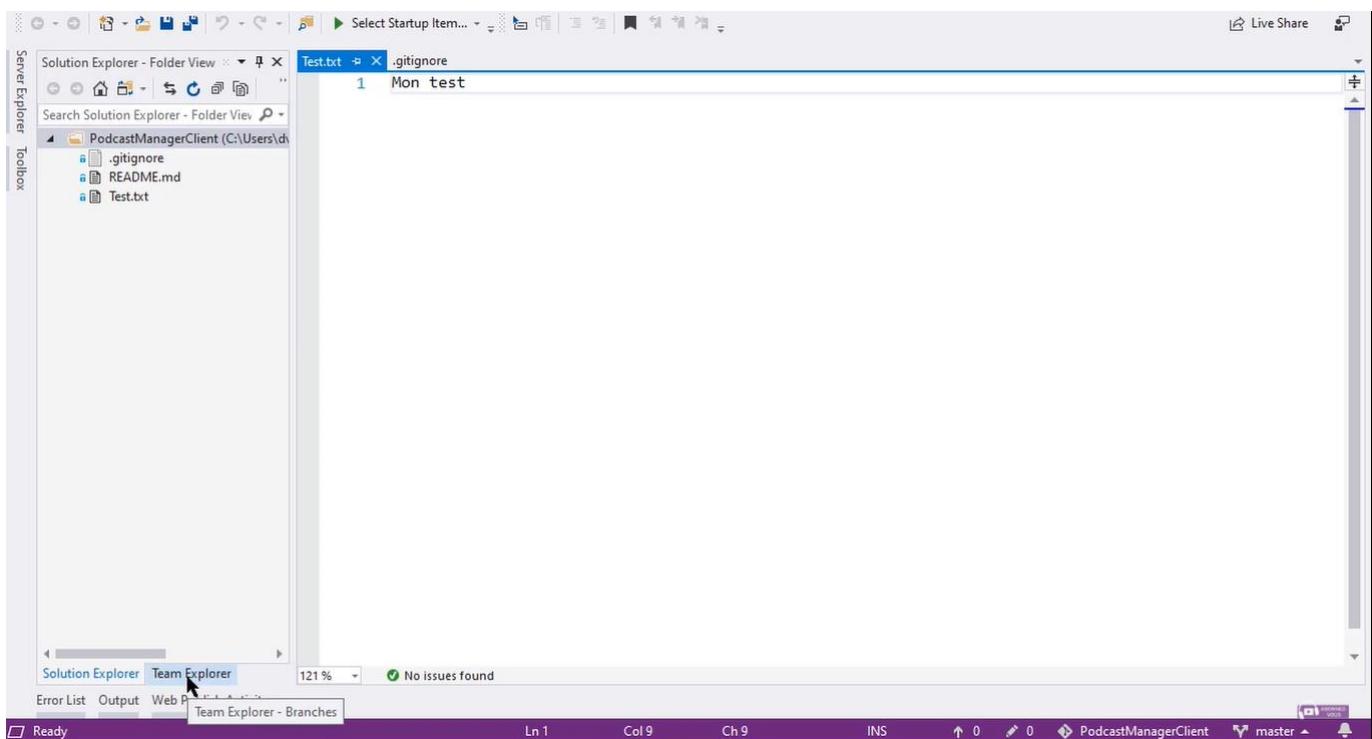


## Branches avec Git

L'objectif de ce chapitre c'est de travailler efficacement avec les branches de GIT.

Etape 1 :

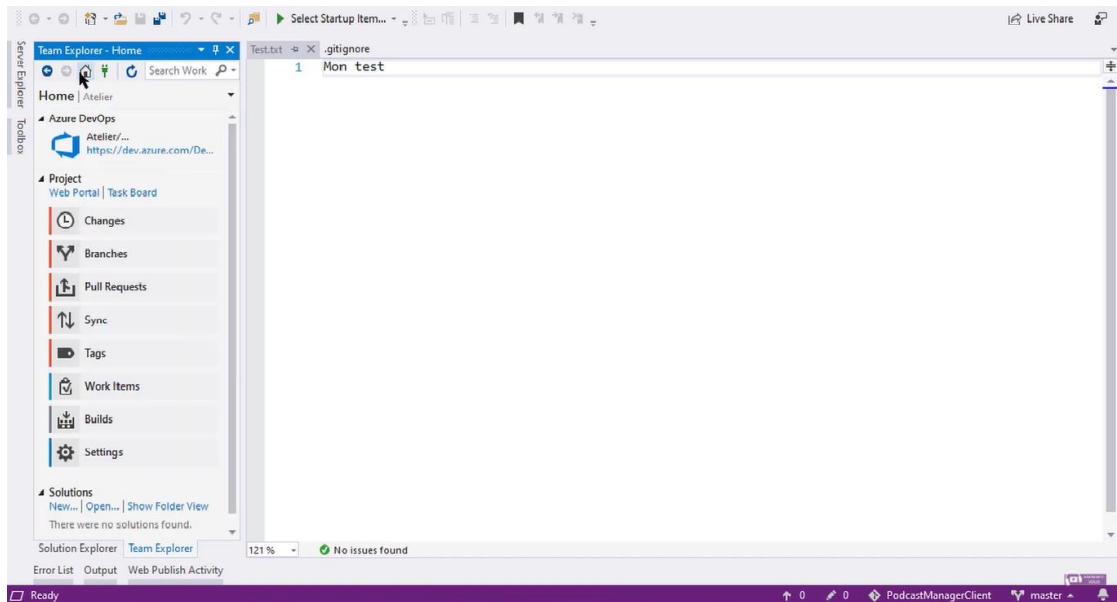
Pour créer une branche dans Visual Studio, positionnez-vous au niveau du code il suffit d'aller dans la zone team explorer





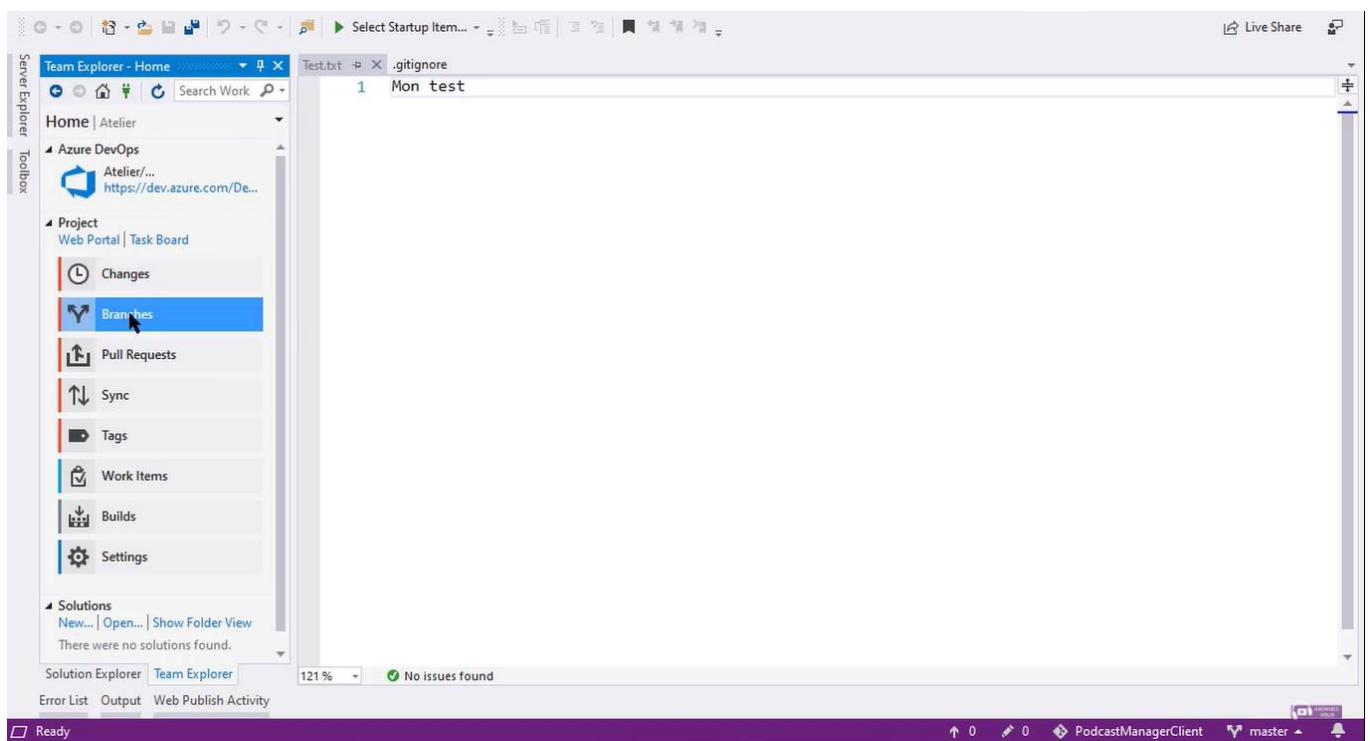
## Etape 2 :

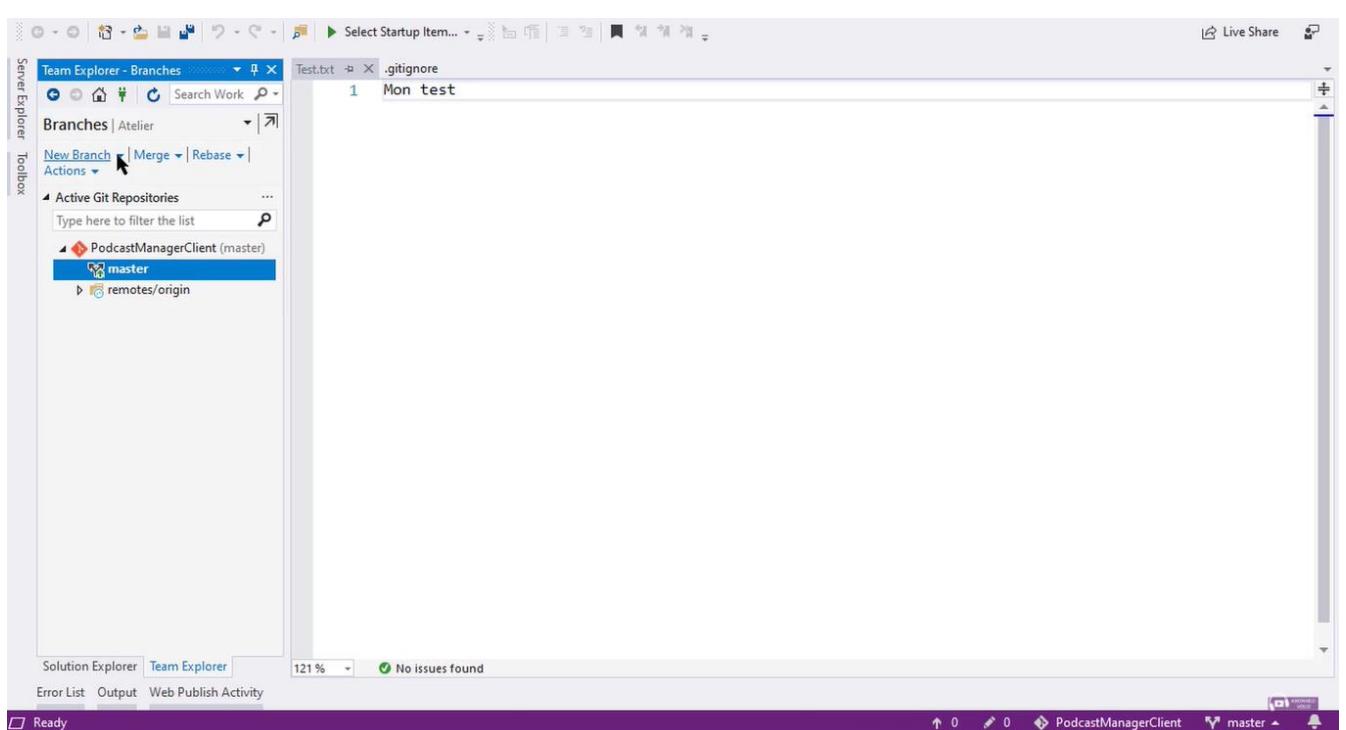
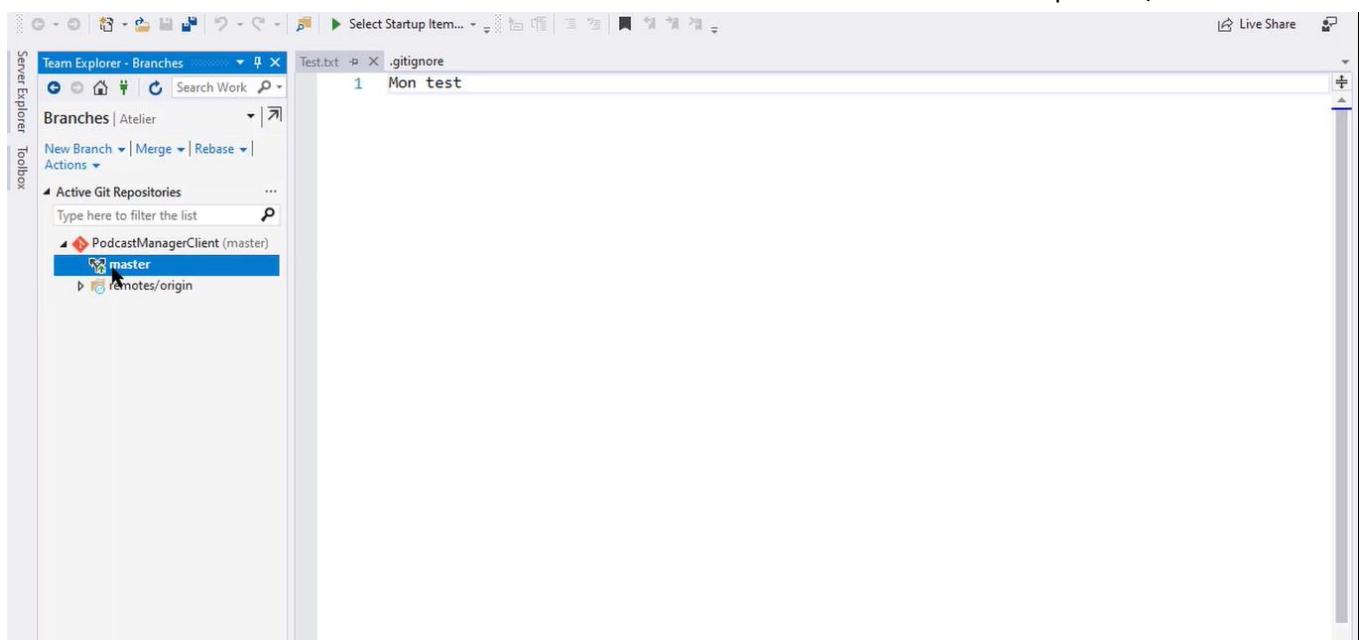
Appuyer sur cette petite maison, on vient à la racine de toutes les fonctionnalités



## Etape 3 :

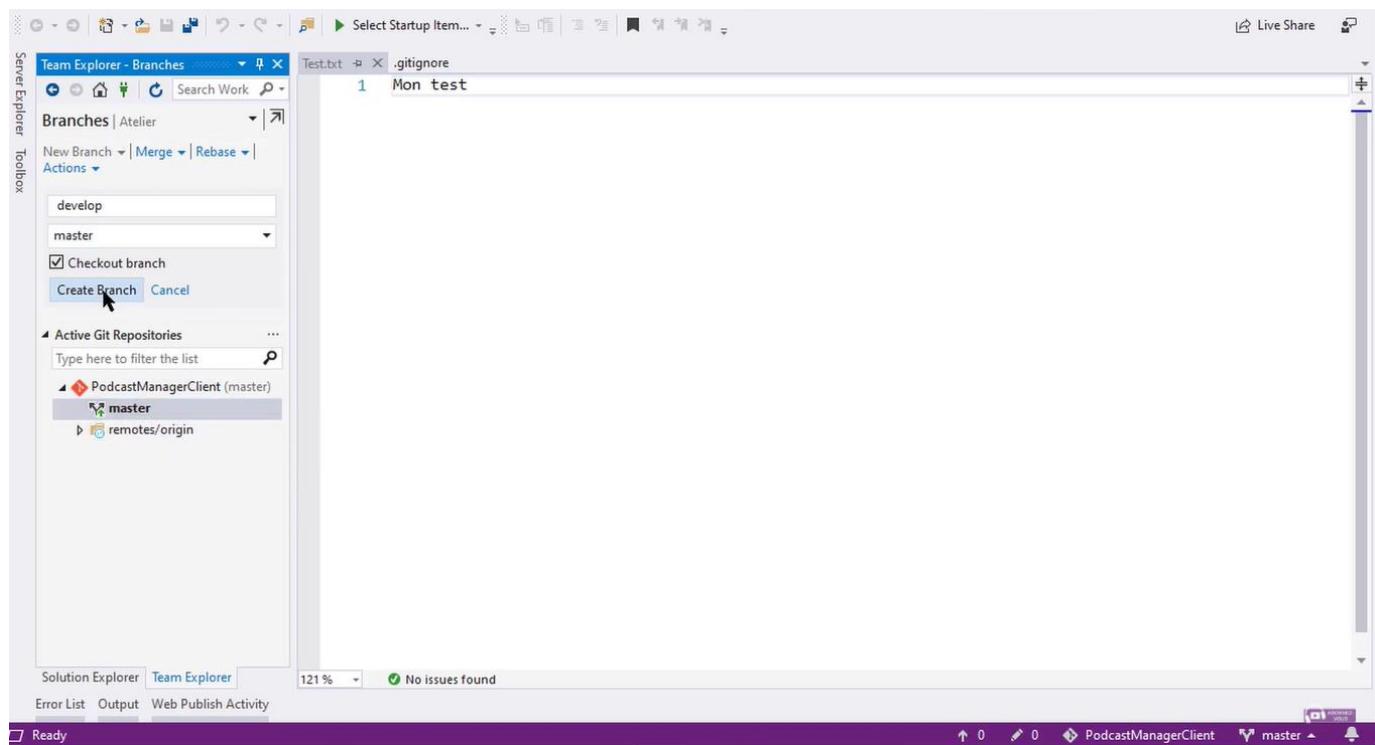
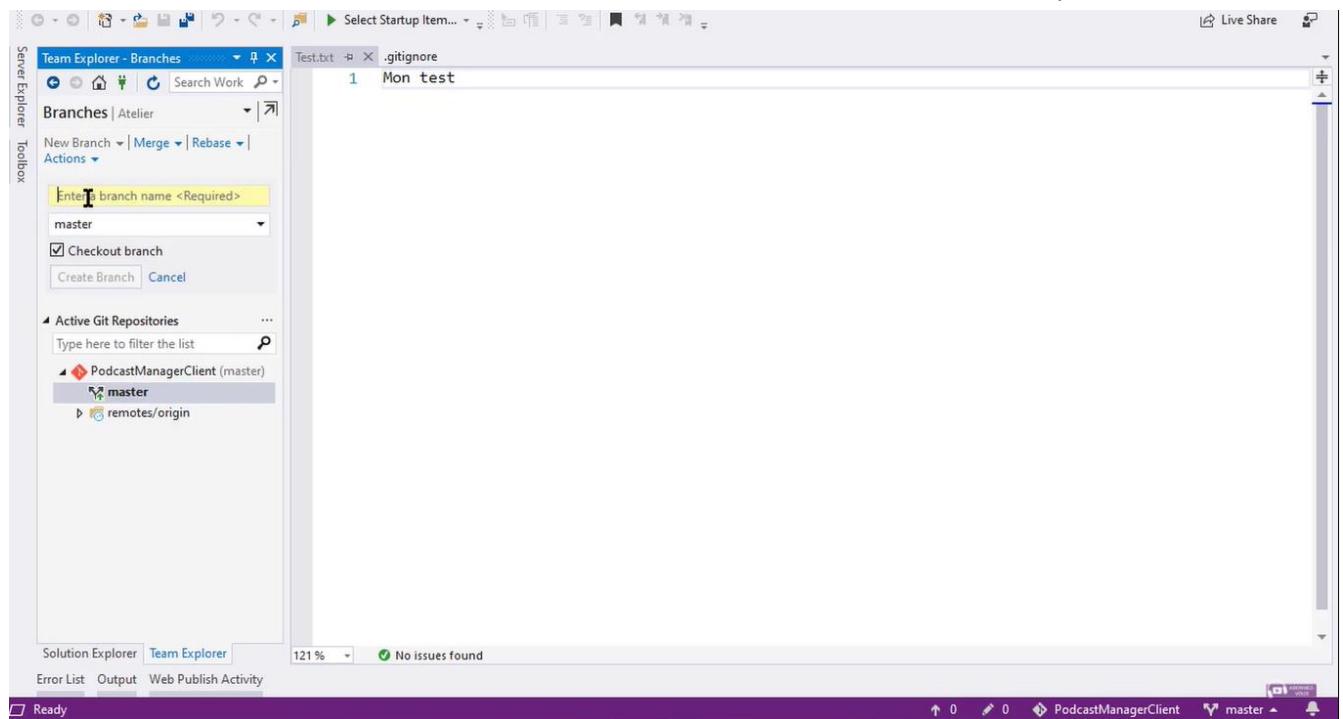
Vous avez une rubrique branche au niveau de la branche active qui est en gras, vous pouvez désormais créer une nouvelle branche.



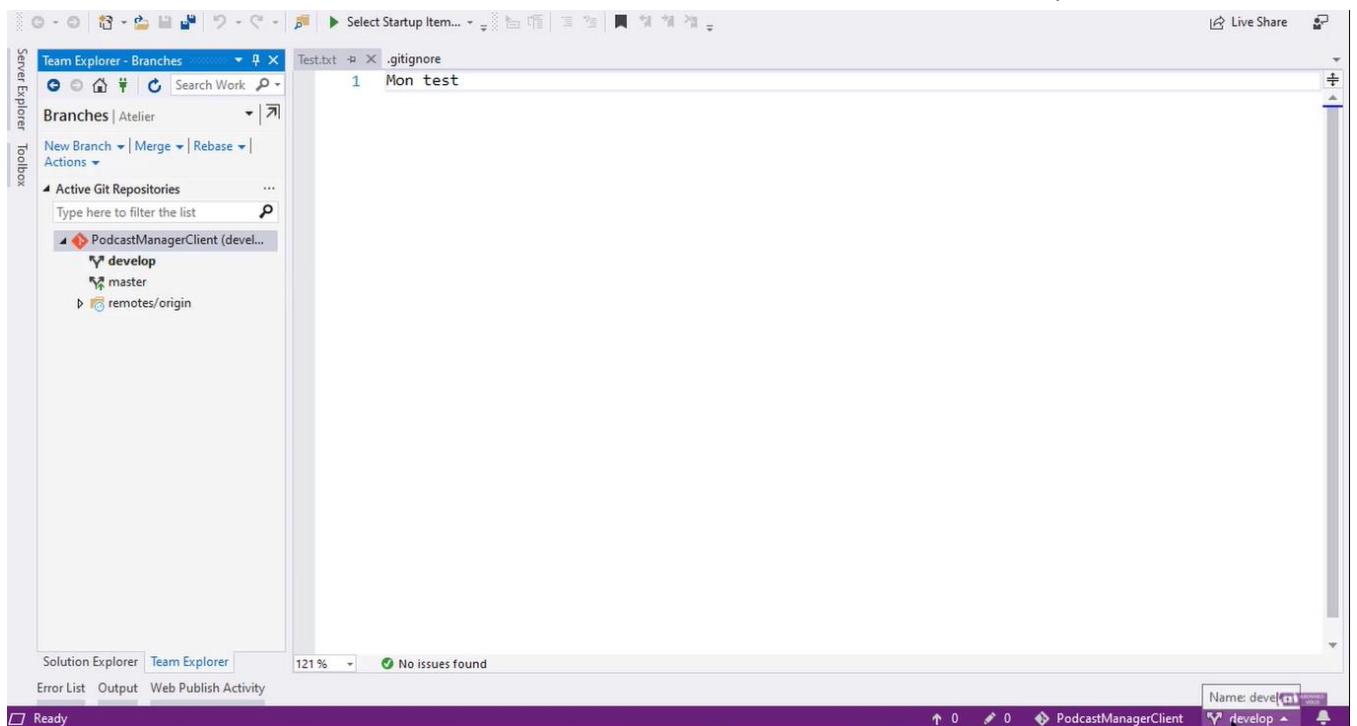


## Etape 6 :

Nommez votre branche « développe » par exemple, et appuyez sur le bouton de création de branche la case check out était cochée ce qui signifie qu’automatiquement il vient se mettre sur cette branche développe.

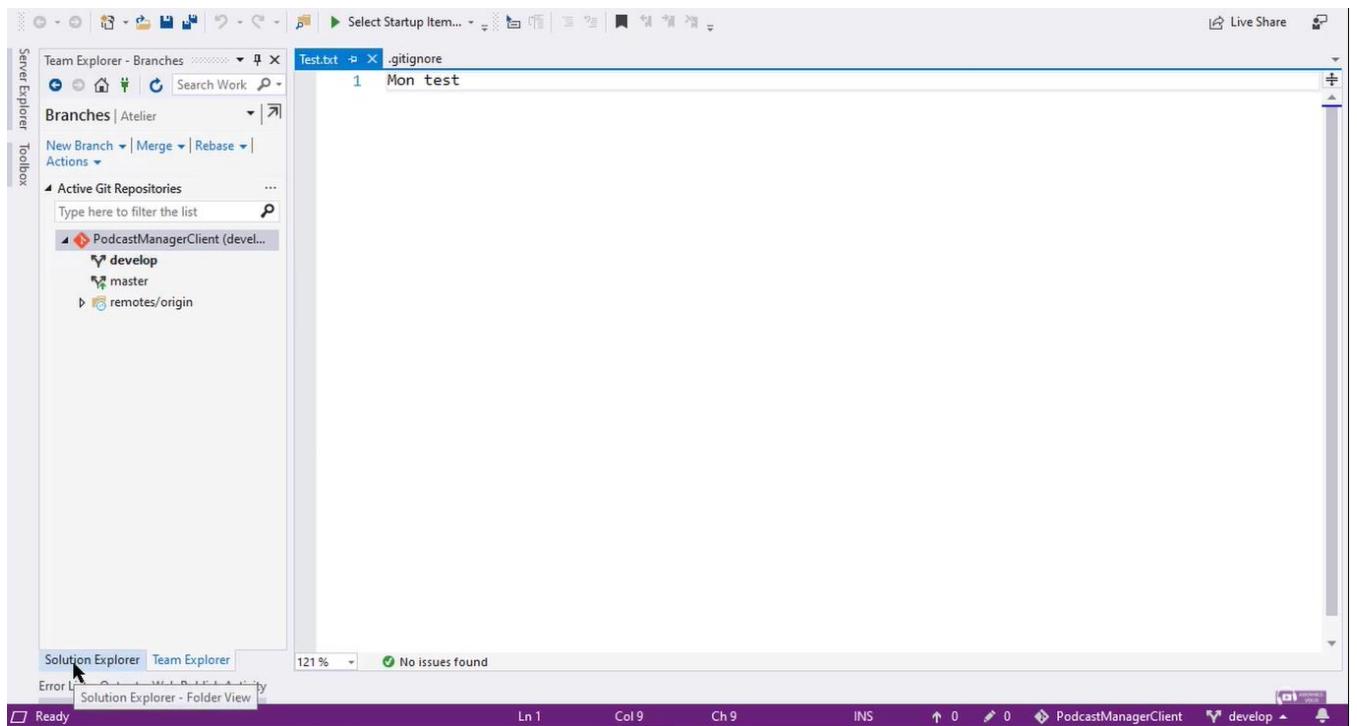


Vous pouvez le voir en permanence en bas à droite de Visual Studio.



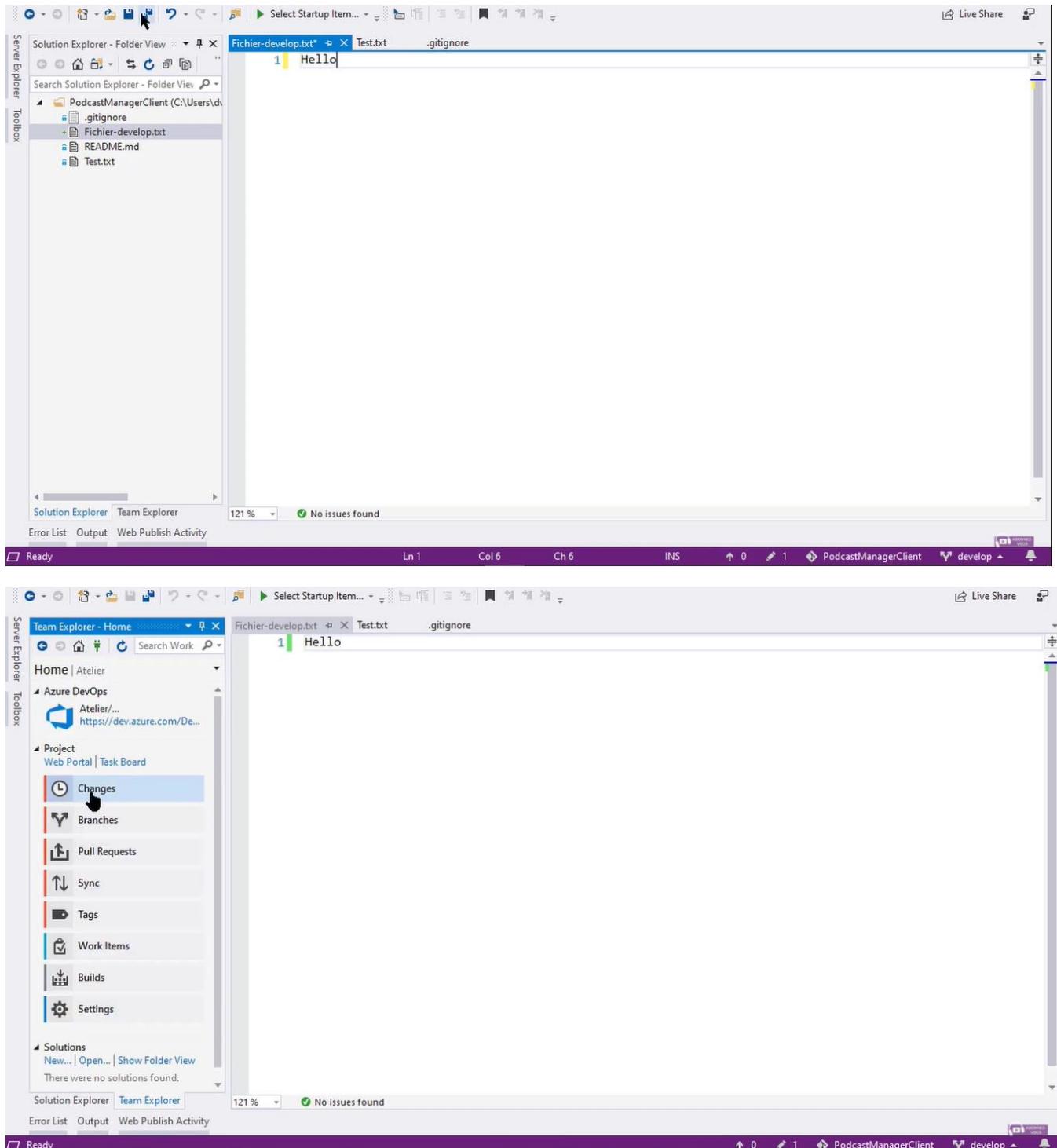
### Etape 9 :

Vous vous retrouvez avec cette zone développer donc vous pouvez faire toutes les modifications de code que vous voulez à ce niveau-ci, soit dans des fichiers existants soit rajouter de nouveaux fichiers.



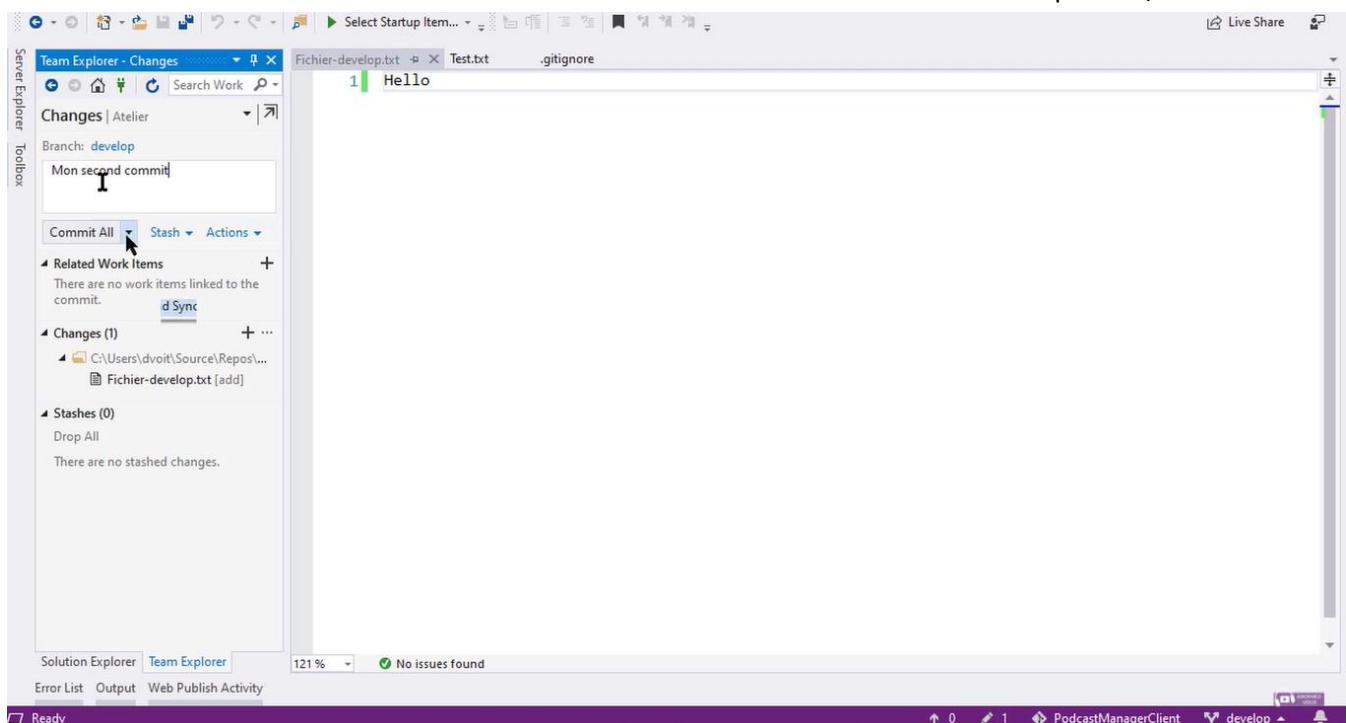
## Etape 10 :

Rajoutez un nouveau fichier develop.txt dans votre projet, vous pouvez revenir dans team explorer pour se rendre compte qu'il a bien détecté ce nouveau fichier qui doit être ajouté à votre projet.



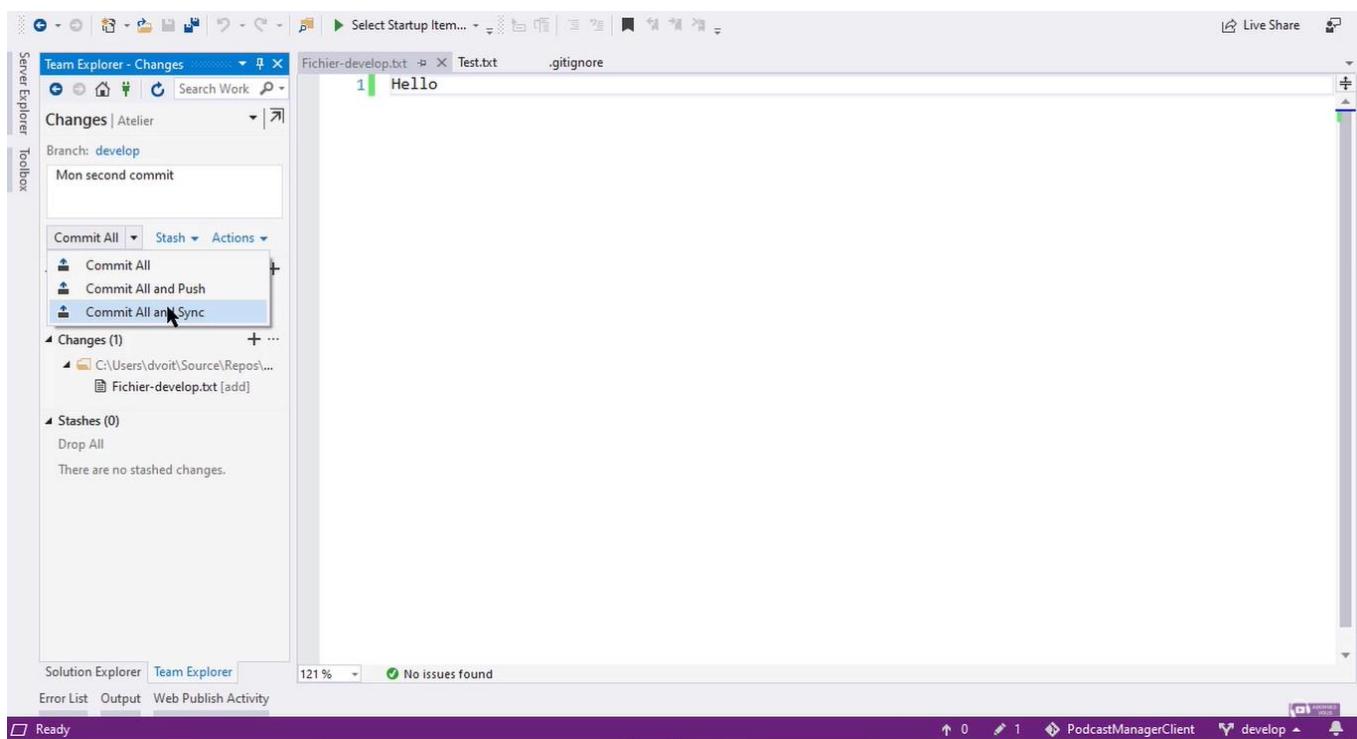
## Etape 12 :

Vous avez simplement besoin de spécifier le commentaire comme vous le fait habituellement.



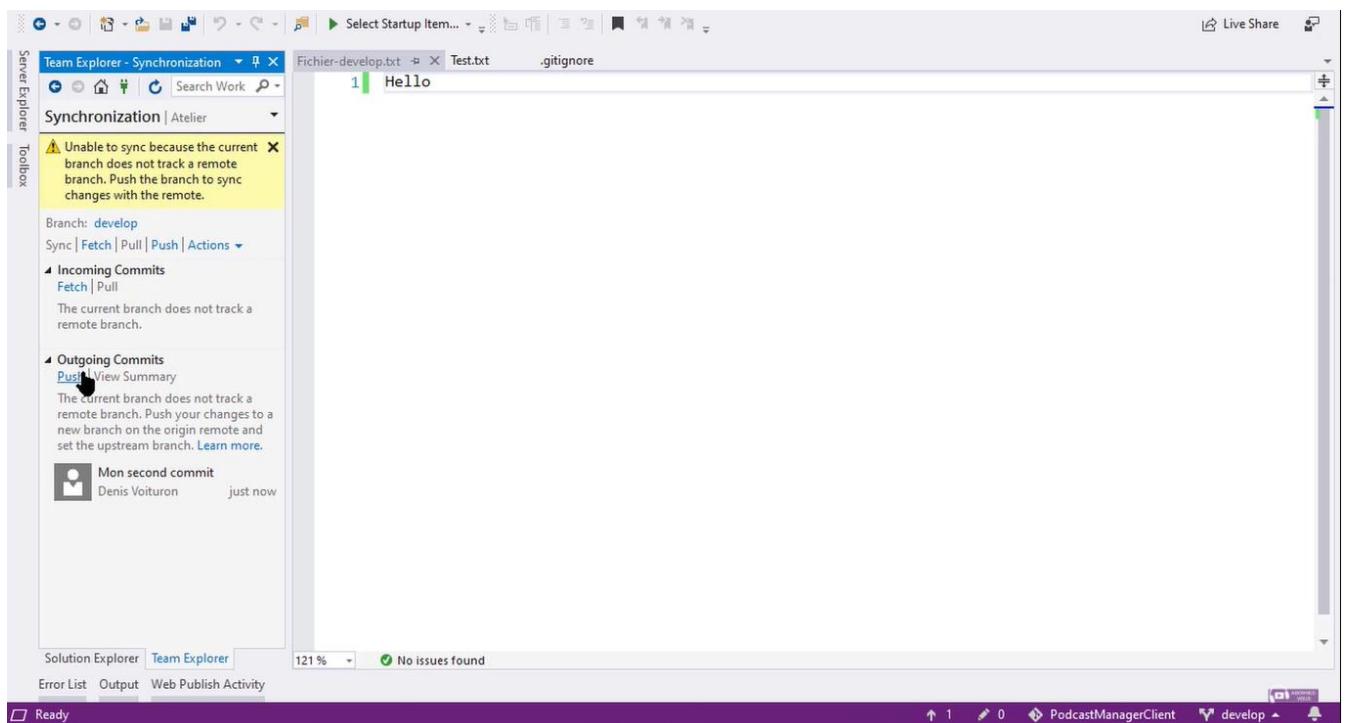
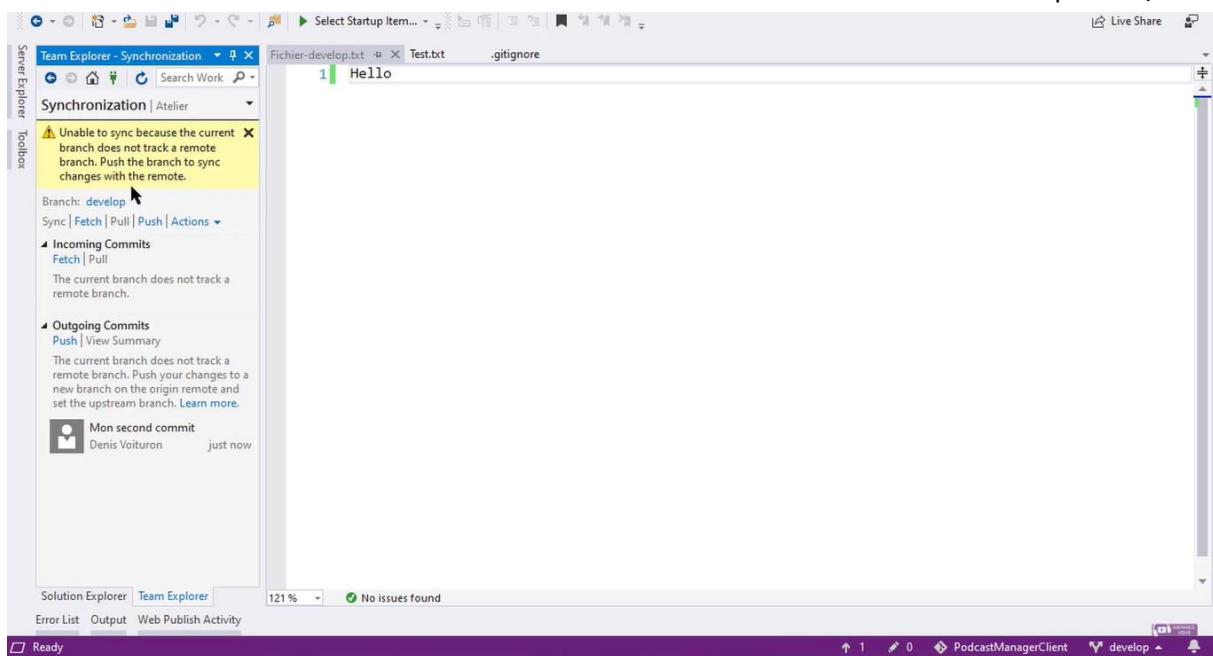
### Etape 13 :

Remonter le code en local et le synchroniser avec le serveur.



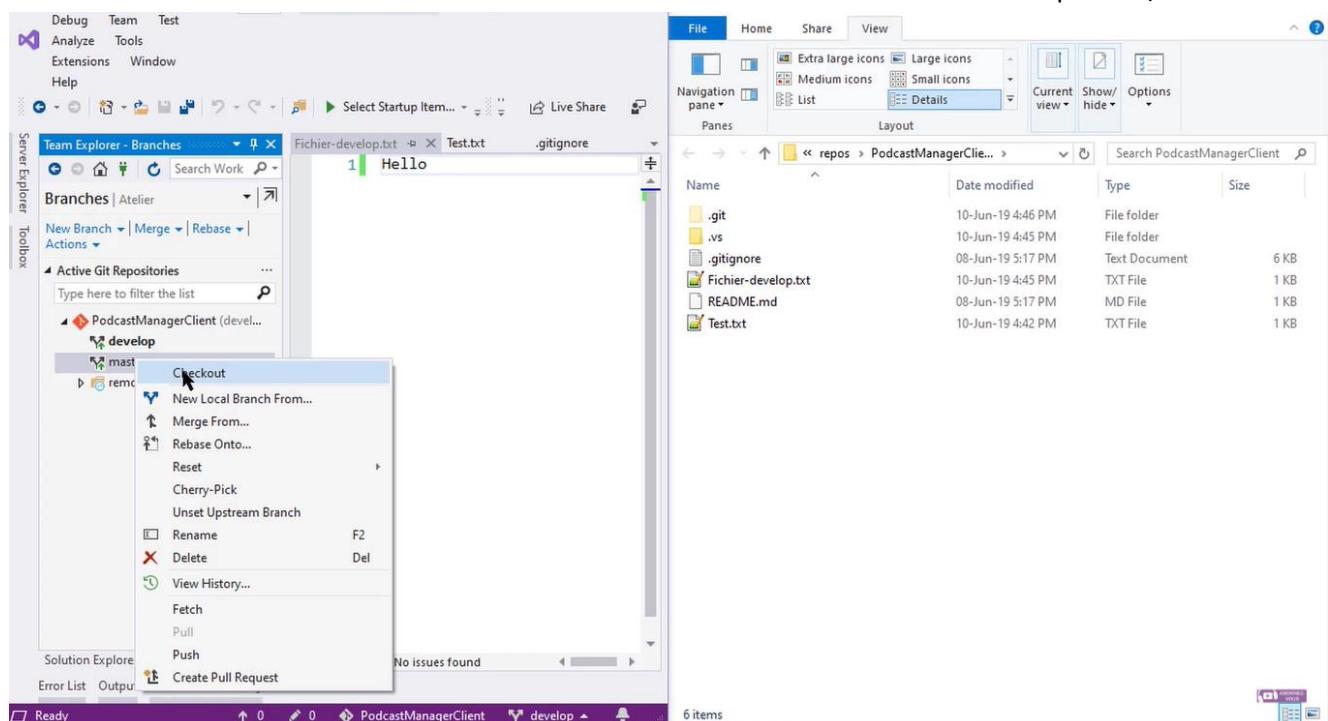
### Etape 14 :

Vous voyez qu'il m'a dit qu'il y a une erreur tout simplement parce que le serveur ne connaît pas encore cette branche et il me propose de venir faire d'abord un push de branche vous pouvez le faire au préalable pour éviter d'avoir ce message d'erreur



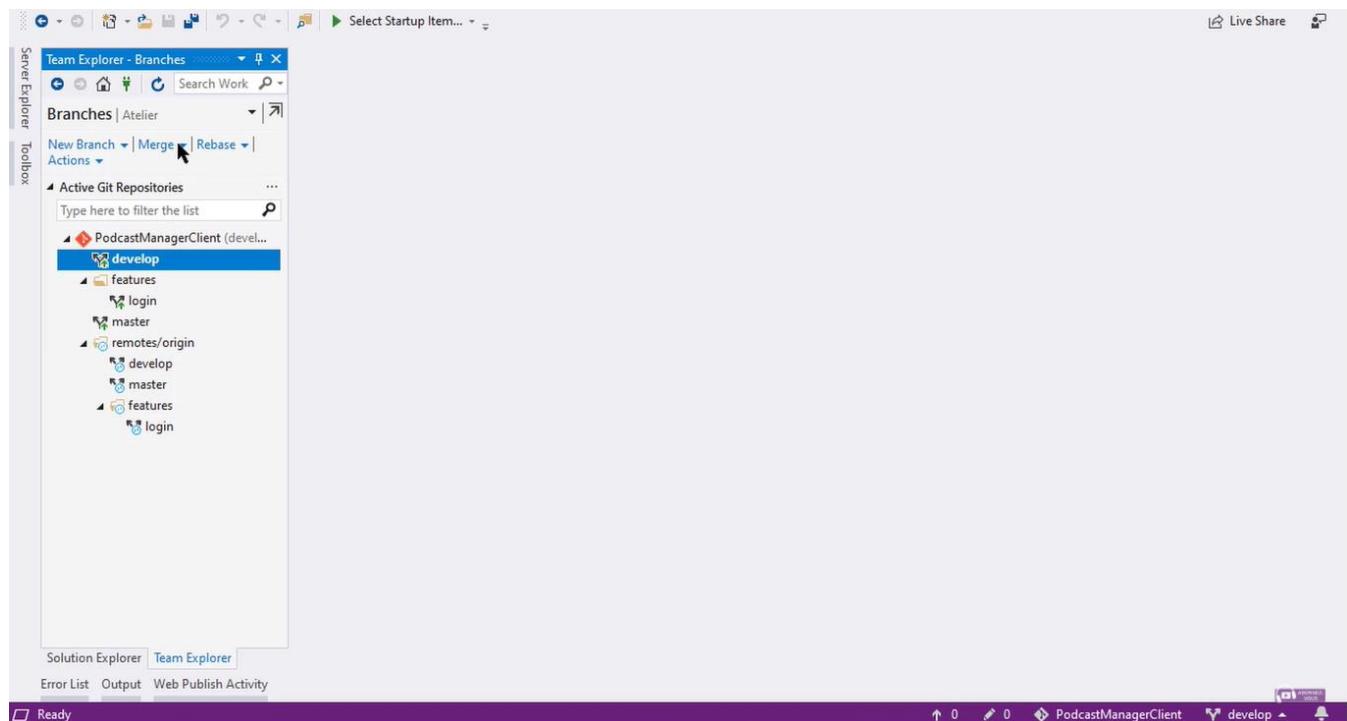
## Etape 16 :

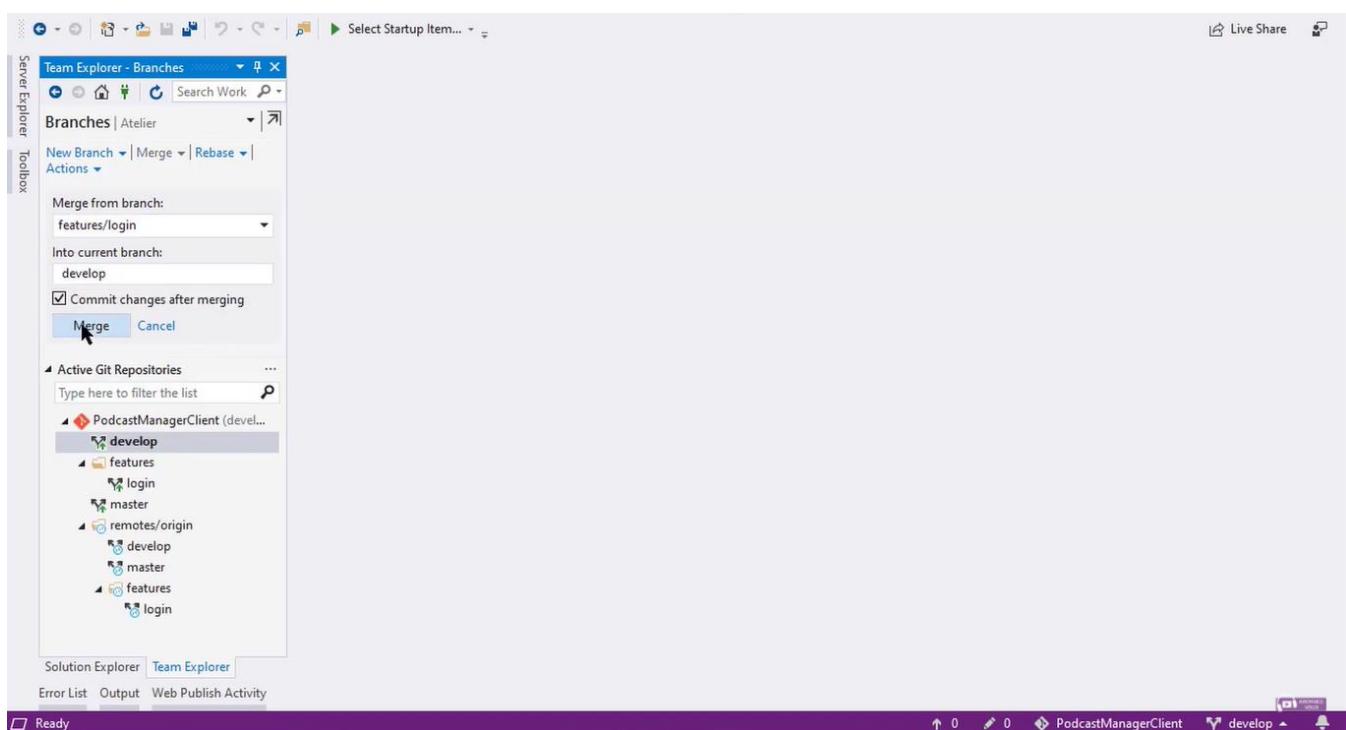
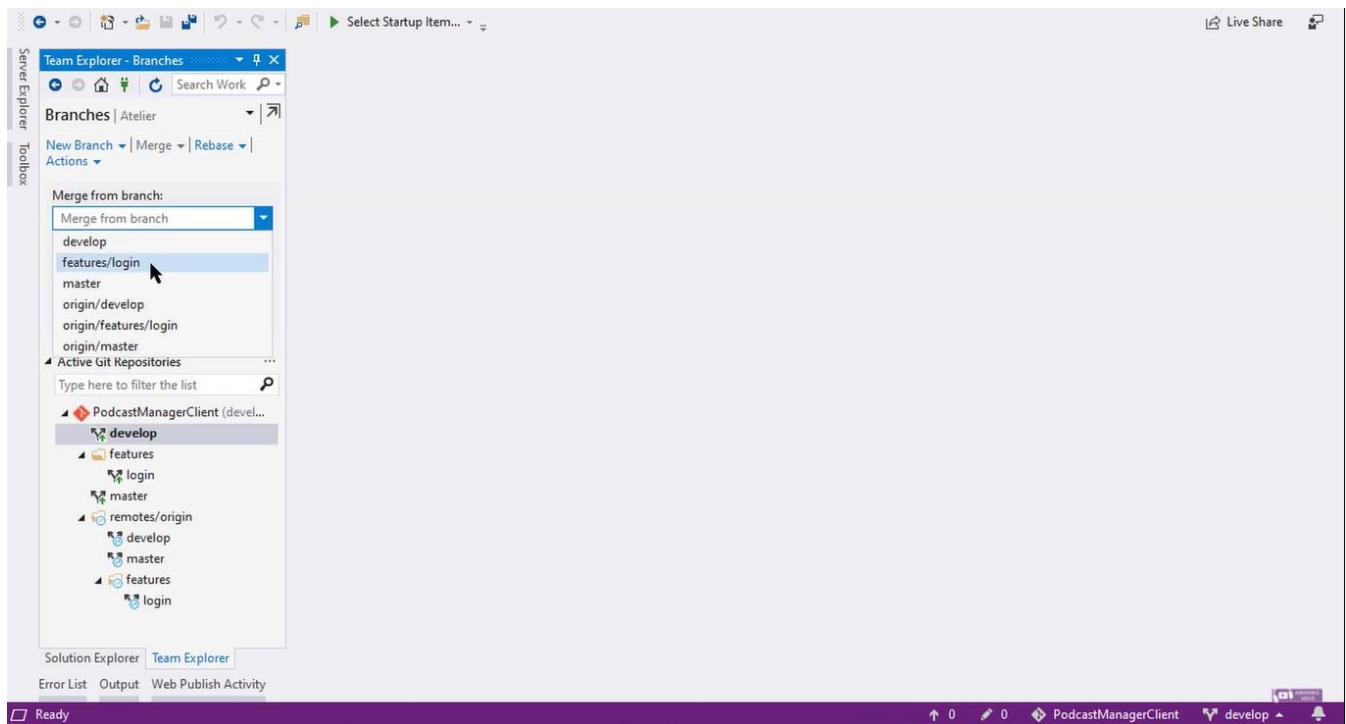
Une fois le code remonté vous pouvez pouvoir changer à souhait de branche.



### Etape 17 :

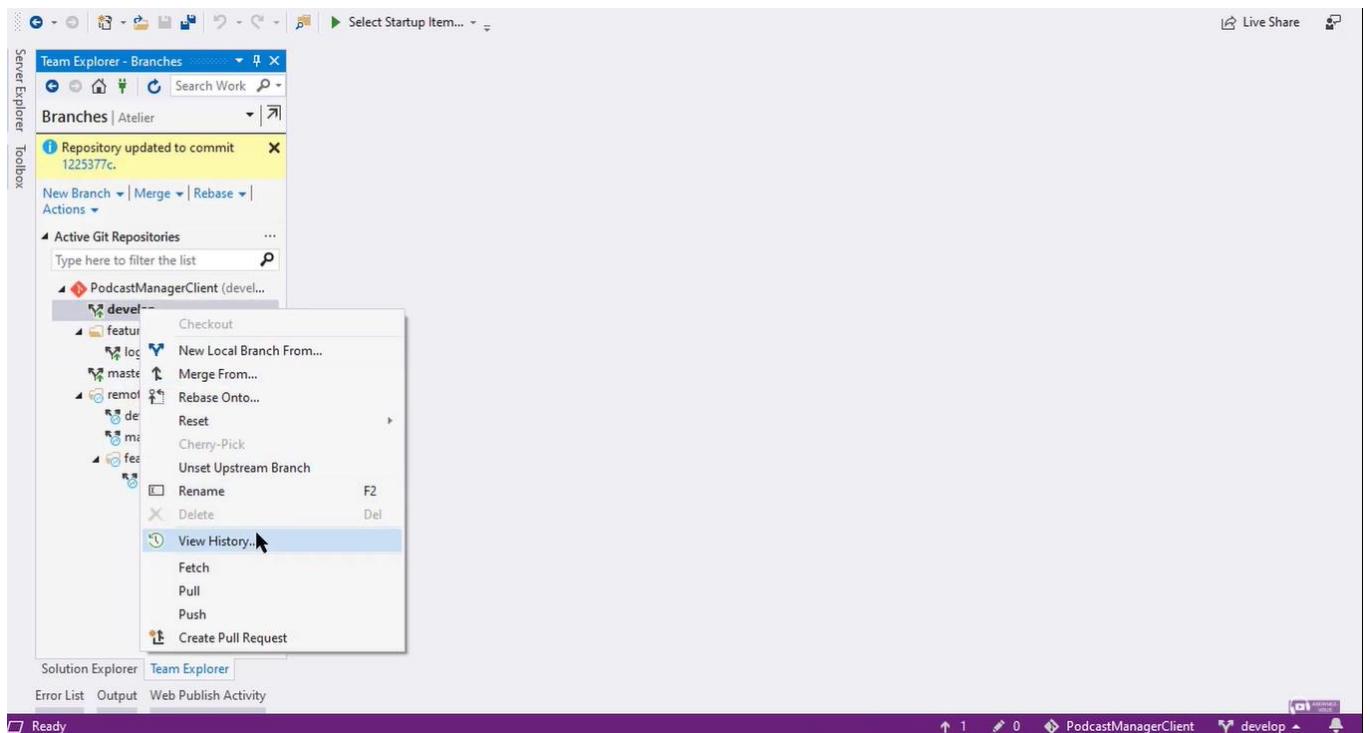
Pour Fusionner le contenu de vos développements (de login par exemple) sur la branche de développement. Il suffit donc de sélectionner cette branche et de venir demander de la fusionner dans la branche qui vous intéresse. On voit ici qu'on peut choisir la branche de départ et la branche de destination. La branche develop est ainsi mise à jour.





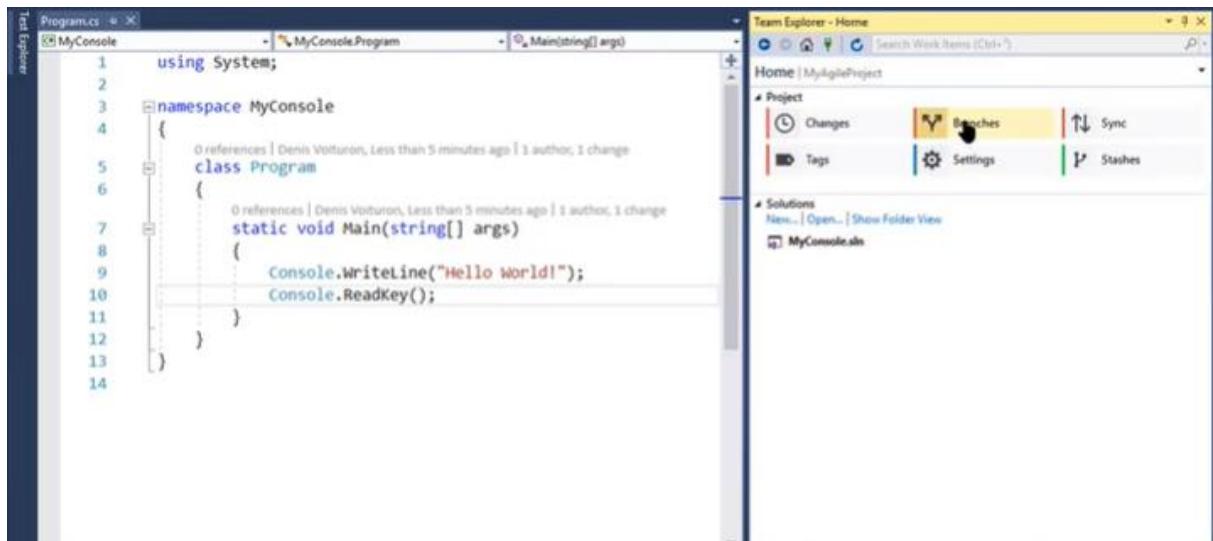
Etape 20 :

Vous pouvez donc constater en faisant un clic droit « View history » l'ensemble des différents comits.



## Commandes Git

D'abord vérifier que votre branche est à jour.



Etape 2 :

Créez une nouvelle branche avec une nouvelle fonctionnalité (par exemple fonctionnalité ajout d'une documentation dans le logiciel).



The screenshot shows the Visual Studio interface with the 'Team Explorer - Branches' window open. In the 'New Branch' input field, the text 'feature/addhelp' is typed. The 'Checkout branch' checkbox is checked, and the 'Create Branch' button is highlighted with a cursor. The 'Active Git Repositories' section shows 'MyAgileProject (develop)' with branches 'develop', 'master', and 'remotes/origin'.

This screenshot is identical to the one above, showing the 'Team Explorer - Branches' window with the 'New Branch' field containing 'feature/addhelp', the 'Checkout branch' checkbox checked, and the 'Create Branch' button highlighted.

#### Etape 4 :

Cette branche est ancrée localement on va pouvoir la publier sur le guide central.



The screenshot shows the Visual Studio IDE interface. On the left, the code editor displays a C# file named Program.cs with the following code:

```
1  using System;
2
3  namespace MyConsole
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10             Console.ReadKey();
11         }
12     }
13 }
14
```

On the right, the Team Explorer - Branches window is open. A context menu is displayed over a branch named "add". The "Push Branch" option is highlighted with a yellow background.

## Etape 5 :

Rajouter du code. Par exemple une explication de votre projet.

The screenshot shows the Visual Studio IDE interface after the push operation. The code editor now contains additional explanatory text at the bottom of the Main method:

```
1  using System;
2
3  namespace MyConsole
4  {
5      class Program
6      {
7          static void Main(string[] args)
8          {
9              Console.WriteLine("Hello World!");
10             Console.WriteLine("usage: app_name [options] required argument");
11             Console.WriteLine("  options: ");
12             Console.WriteLine("    -a, --argument    Does something");
13             Console.WriteLine("    -b required      Does something");
14             Console.WriteLine("    -c, --command required");
15             Console.WriteLine("    -d [optlistitem1 optlistitem2]");
16             Console.ReadKey();
17         }
18     }
19 }
20
```

The Team Explorer - Branches window shows a success message: "Successfully pushed branch feature/addhelp to origin." The "addhelp" branch is now listed under the "origin" remote.

## Etape 6 :

Comitez les changements d'abord localement et puis ensuite les publier.



The screenshot shows the Visual Studio IDE with the 'MyConsole' project open. In the 'Team Explorer - Changes' window, a commit message 'Add new Help documentation' is visible. A context menu is open over the code in 'Program.cs', with the 'Commit All' option highlighted.

The screenshot shows the Visual Studio IDE with the 'MyConsole' project open. In the 'Team Explorer - Home' window, the 'Changes' tab is selected. A portion of the code in 'Program.cs' is highlighted in blue, indicating it has been modified.

Par exemple on supprime quelques parties pour recomiter et le synchroniser avec le serveur.



The screenshot shows the Visual Studio IDE with the 'Program.cs' file open in the code editor. The 'Team Explorer - Changes' window is active, displaying a single change from 'Denis Voituron' on the 'feature/addhelp' branch. The commit message is 'Fix: Correction of Help documentation'. A context menu is open over this commit, with the option 'Commit All and Sync' highlighted.

### Etape 9 :

Fusionnez le code dans la branche develop commune à tous les développeurs

The screenshot shows the Visual Studio IDE with the 'Program.cs' file open in the code editor. The 'Team Explorer - Branches' window is active, showing the local branches 'develop', 'feature/addhelp', 'master', and 'origin/develop'. The 'origin/develop' branch is currently selected. The 'Merge from branch' dropdown also lists 'origin/develop'.

### Etape 10 :

Il faut d'abord vérifier si il y a des conflits lors de cette fusion comme par exemple si d'autres développeurs sont venus modifier les mêmes morceaux de code que le nôtre vous pouvez voir qu'il y a un conflit détecté (testez avec vos collègues).



The screenshot shows a Visual Studio interface with the 'Program.cs' file open. The code editor displays a merge conflict between the local 'HEAD' branch and the remote 'origin/develop' branch. The 'Team Explorer - Branches' window indicates a 'Merge In Progress' operation with one conflict. The 'Active Git Repositories' section shows the 'MyAgileProject' repository with branches 'develop', 'feature', 'addhelp', 'master', and 'remotes/origin'.

The screenshot shows the 'Resolve Conflicts' dialog for the 'Program.cs' file. The 'Merge' button is highlighted, and the message 'Conflicting content changes have been made. Compare Files' is displayed. Options for 'Edited on Source' and 'Edited on Target' are shown with 'Diff' and 'Keep' buttons.

## Etape 12 :

Vous pouvez voir sur la partie droite votre partie de code et sur la partie gauche la partie de code développée ou modifiée par un autre développeur, et il suffit de choisir voir éventuellement corriger le code à fusionner et ensuite de l'accepter pour pouvoir régénérer et recréer un nouveau commit qui va être remonté sur le serveur.

Merge - MyConsole...et (both modified)\* ➔ Program.cs

1 Conflicts (1 Remaining)

Source: MyConsole/Program.cs;Source Target: MyConsole/Program.cs;Target

```

7 (string[] args)
8
9  eLine("Hello World!");
10 eLine("usage: app_name [options]
11  eLine("  options: ");
12 eLine("    -a, --argument
13 eLine("    -b required
14 key();
15
16 }
17 }
18 }
```

Result: MyConsole/Program.cs

```

7 static void Main(string[] args)
8 {
9     Console.WriteLine("Hello World!");
10    Console.ReadKey();
11
12 }
13
14 }
```

Team Explorer - Resolve Conflicts

Resolve Conflicts | MyAgileProject

Commit Merge Abort

Conflicts (1)

Name Path

Program.cs [both mo... MyConsole

Merge

Conflicting content changes have been made. Compare Files

Edited on Source Diff Take Source

Edited on Target Diff Keep Target

Merge - MyConsole...et (both modified)\* ➔ Program.cs

1 Conflicts (0 Remaining)

Source: MyConsole/Program.cs;Source Target: MyConsole/Program.cs;Target

```

10 Console.WriteLine("Hello World!");
11 Console.WriteLine("usage: app_name [options] required_
12  options: ");
13 Console.WriteLine("    -a, --argument      Does something
14 Console.WriteLine("    -b required       Does something
15
16 }
17 }
18 }
```

Result: MyConsole/Program.cs

```

10 static void Main(string[] args)
11 {
12     Console.WriteLine("Hello World!");
13     Console.ReadKey();
14
15 }
16
17 }
```

Team Explorer - Resolve Conflicts

Resolve Conflicts | MyAgileProject

Commit Merge Abort

Conflicts (0)

Name Path

Program.cs [both mo... MyConsole

Merge

Conflicting content changes have been made. Compare Files

Edited on Source Diff Take Source

Edited on Target Diff Keep Target

Etape 14 :



The screenshot shows the Visual Studio interface. In the center is the code editor with a C# file named Program.cs. The code contains a Main() method that prints "Hello World!" and usage instructions for command-line arguments. To the right of the code editor is the 'Team Explorer - Changes' panel. It shows a commit history for a branch named 'feature/addhelp'. The most recent commit is highlighted with a yellow background. A context menu is open over this commit, with the option 'Commit All and Sync' highlighted in yellow. Below the commit history, there is a message stating 'There are no unstaged changes in the working directory.'

Recommencez une autre fois

Etape 15 :

Sur votre interface web Azure, cliquez sur Pull Requests.

The screenshot shows the Azure DevOps interface with the 'Pull Requests' tab selected. On the left, a sidebar shows the repository structure for 'MyAgileProject' with files like MyConsole, .gitignore, MyConsole.sln, and README.md. The main area displays a pull request titled 'You updated feature/addhelp 3 minutes ago — Create a pull request'. It lists four commits from the 'feature/addhelp' branch:

Name	Last change	Commits	Message
MyConsole	6 minutes ago	dd9efc3d	Sample of Help documentation Denis Voituron
.gitignore	03/07/2018	e0b89828	Added README.md, .gitignore (VisualStudio) files Denis Vo...
MyConsole.sln	22 minutes ago	17747a15	First commit Denis Voituron
README.md	03/07/2018	e0b89828	Added README.md, .gitignore (VisualStudio) files Denis Vo...

Etape 16 :

Choisissez votre branche de fonctionnalités.



The screenshot shows the GitHub interface for a repository named 'MyAgileProject'. The 'Pull Requests' tab is selected. A modal window titled 'New Pull Request' is open, showing a pull request from the 'feature/addhelp' branch into the 'master' branch. The 'develop' branch is highlighted in the dropdown menu. The interface includes standard GitHub UI elements like file browser, commit history, and a 'Markdown supported' note.

### Etape 17 :

Fusionnez avec la branche de développement commune à tous les développeurs

The screenshot shows the GitHub interface for a repository named 'MyAgileProject'. The 'Pull Requests' tab is selected. A modal window titled 'New Pull Request' is open, showing a pull request from the 'feature/addhelp' branch into the 'master' branch. The 'Title' field is empty and highlighted with a red border. The 'master' branch is highlighted in the dropdown menu. The interface includes standard GitHub UI elements like file browser, commit history, and a 'Markdown supported' note.

### Etape 18 :



Il suffit de d'écrire un petit titre explicatif de cette fonctionnalité éventuellement une description complète.

The screenshot shows a pull request creation interface. At the top, there's a navigation bar with tabs: Files, Commits, Pushes, Branches, Tags, and Pull Requests (which is currently selected). Below the navigation bar, there's a section for creating a new pull request. It includes fields for the source branch ('feature/addhelp') and target branch ('develop'). A title field contains 'Add new Help documentation'. There's also a 'Description' field with placeholder text 'Describe the code that is being reviewed' and a note 'Markdown supported.' Below the description field are rich text editing tools (bold, italic, etc.) and a button to 'Add commit messages'. The 'Reviewers' section shows '[MyAgileProject]\\MyAgileProject Team' selected. At the bottom, there's a 'Create' button.

### Etape 19 :

Vous choisissez après les reviewers ou l'ensemble de l'équipe. Les reviewer sélectionnés vont ainsi recevoir un e-mail leur permettant d'accéder à cette interface graphique ils vont pouvoir comparer le code. Ils peuvent également écrire des commentaires à destination du développeur lui poser des questions le développeur pourra corriger le code voir également résoudre les commentaires qui lui auront été ainsi soumis ce qui permettra aux reviewer d'approuver finalement le code proposé par le développeur.

The screenshot shows a work item creation interface. At the top, there's a navigation bar with tabs: Files, Commits, Pushes, Branches, Tags, and Pull Requests (which is currently selected). Below the navigation bar, there's a 'Markdown supported.' note and rich text editing tools. The 'Reviewers' section shows '[MyAgileProject]\\MyAgileProject Team' selected. The 'Work Items' section has a search bar 'Search work items by ID or title' and a 'Create' button. At the bottom, there are links for 'Files (1)' and 'Commits (3)'. The interface is mostly blank below these sections.



DevOps 2020/2021

MyAgileProject Files Commits Pushes Branches Tags Pull Requests Create

Files (1) Commits (3)

Showing 1 file change: 1 edit

Side-by-side diff

C# Program.cs +4 -1 /MyConsole/Program.cs

```
... 7     static void Main(string[] args)
8     {
9         Console.WriteLine("Hello World!");
10 -        Console.WriteLine("usage: app_name [options] ");
11
12     }
13
14
15
16 }
```

... 7 static void Main(string[] args)
8 {
9 Console.WriteLine("Hello World!");
10 + Console.WriteLine("usage: app\_name [options] required\_inp");
11 + Console.WriteLine(" options: ");
12 + Console.WriteLine(" -a, --argument Does something");
13 + Console.WriteLine(" -b required Does something wit");
14
15 }
16 }

MyAgileProject Files Commits Pushes Branches Tags Pull Requests Fork Clone

11 ACTIVE Add new Help documentation

Denis Voituron feature/addhelp into develop

Approve Complete ...

Overview Files Updates Commits

All updates Active (1)

Find a file or folder

/MyConsole

c# Program.cs \* May be add

static void Main(string[] args)
{
 Console.WriteLine("Hello World!");
 Console.WriteLine("usage: app\_name [options] required\_inp");
}

May be add m

Markdown supported. Drag & drop, paste, or select

Aa B I ↗ ⓘ ⌂ ⌃ ⌄ ⌅ ⌆

May be add

11 + static void Main(string[] args)
12 + {
13 + Console.WriteLine(" options: ");
14 + Console.WriteLine(" -a, --argument Does something");
15 + Console.WriteLine(" -b required Does something wit");

MyAgileProject Files Commits Pushes Branches Tags Pull Requests Fork Clone

11 ACTIVE Add new Help documentation

Denis Voituron feature/addhelp into develop

Approve Complete ...

Overview Files Updates Commits

Labels

Add label

C# Program.cs /MyConsole/Program.cs

```
... 8
9
10 -    Console.WriteLine("Hello World!");
11
12
13
```

Denis Voituron just now May be add more details?

Write a reply...

Created by Denis Voituron

Approve Approve with suggestions Wait for author Reject Reset feedback

just now

1 Resolved

Reactivate

Etape 23 :



DevOps 2020/2021

Le développeur pourra ainsi finalement venir compléter c'est à dire fusionner physiquement sa branche de travail

MyAgileProject Pull Requests

11 ACTIVE Add new Help documentation

Denis Voituron feature/addhelp into develop

Overview Files Updates Commits

Denis Voituron Approved

Labels Add label

Approved by Denis Voituron

Denis Voituron joined as a reviewer

C# Program.cs /MyConsole/Program.cs

```
8 {
9     Console.WriteLine("Hello World!");
10    Console.WriteLine("usage: app_name")
11 +
12 +
13 + }
```

Denis Voituron just now May be add more details? Resolved

Approve Complete Abandon

just now just now just now

Fork Clone

MyAgileProject Pull Requests

11 ACTIVE Add new Help do

Denis Voituron feature/addhelp into develop

Overview Files Updates Commits

Denis Voituron Approved

Labels Add label

Complete pull request

Add new Help documentation

Enter more details about the pull request

Complete linked work items after merging ⓘ

Delete feature/addhelp after merging

Squash changes when merging [Learn more](#)

Complete merge Cancel

Denis Voituron just now May be add more details? Resolved

Approve Complete

just now just now just now

{

```
Console.WriteLine("Hello World!");
Console.WriteLine("usage: app_name")
Console.WriteLine(" options:");
Console.WriteLine(" -a, --argum
Console.WriteLine(" -b required")
```

Fork Clone