

# Atelier Git Client (Git Command Line)

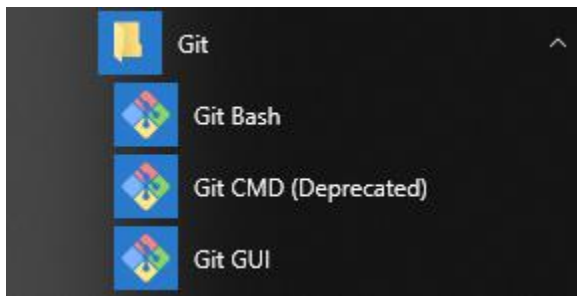
- I. Ce guide vous montre comment partager votre code dans un référentiel Git dans GitHub à l'aide de la ligne de commande (Git Client).

## Prérequis :

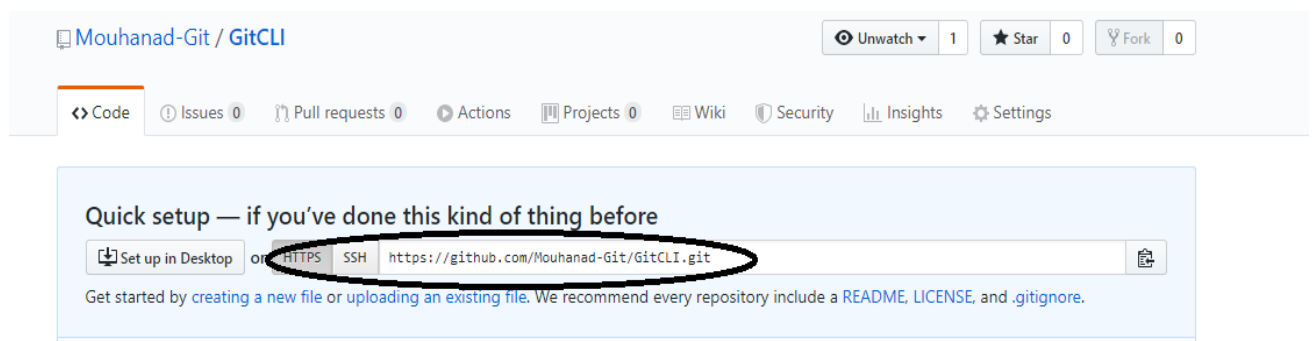
Pour cet atelier, vous devez télécharger Git Client, ci-dessous un lien vers un téléchargement Git :

<https://git-scm.com/download/win>

Une fois installé, vous pouvez lancer l'invite de commande Git (Git CMD) :



1. La première étape consiste à créer un répertoire Git local sur votre machine (par la suite, on va procéder à la connexion de ce répertoire avec une repository sur GitHub).
2. Lancez Visual Studio, créer un projet WebForm dans votre répertoire Git local.  
NB : pas besoin de cocher l'option : « Créer un Dépôt Git »
3. Par la suite, connectez-vous à votre compte GitHub et créer une nouvelle Repository Public.



4. Ouvrez la fenêtre de commande Git. Accédez au dossier où vous souhaitez que le code du référentiel sera stocké sur votre ordinateur et exécutez les commandes suivantes :

- git Init

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

- git add .

Cette commande ajoute les fichiers créer dans votre Git local.

- git commit -m "First commit"

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

- git remote add origin  
[https://ELFILALIMOUHANADProfC@dev.azure.com/ELFILALIMOUHANADProfC/DevOps/\\_git/DevOps](https://ELFILALIMOUHANADProfC@dev.azure.com/ELFILALIMOUHANADProfC/DevOps/_git/DevOps)

Cette commande pour connecter votre Git local avec la repository sur le serveur GitHub.

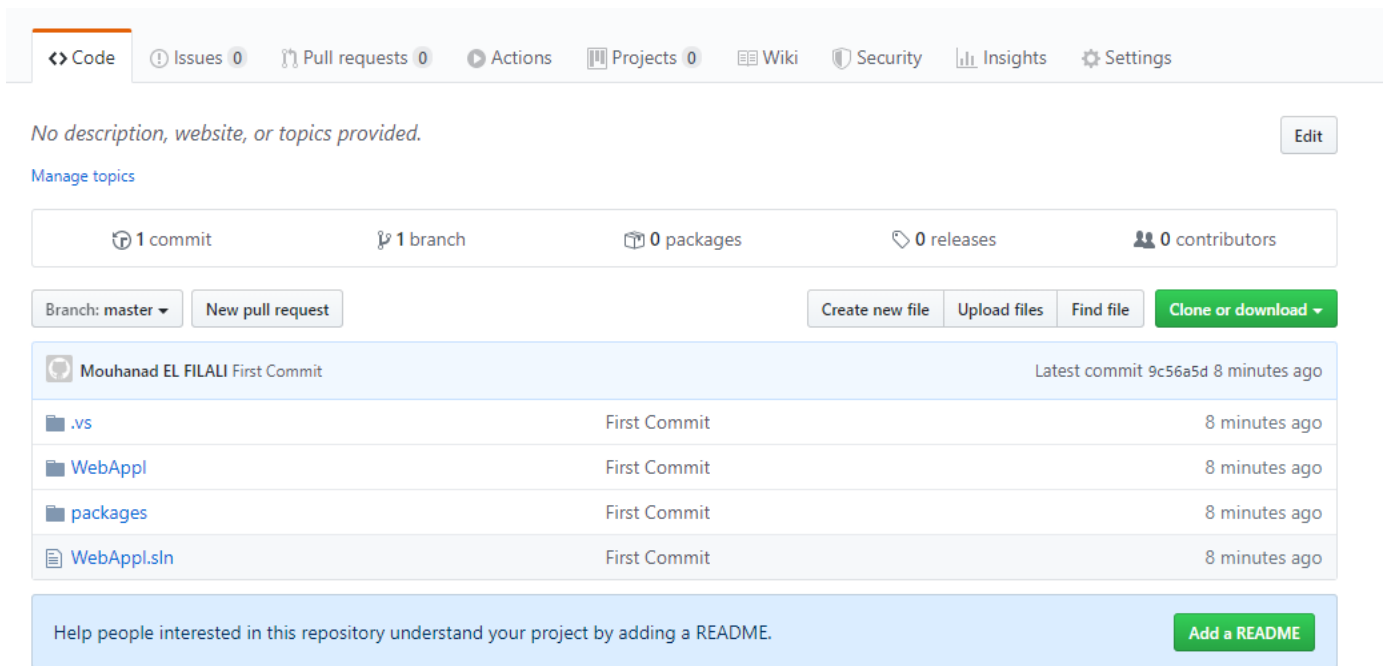
- git push origin master

Cette commande pour envoyer les fichiers du Git Local vers le serveur sur la branche master.

Ci-dessous à quoi ressemble l'exécution des commandes, maintenant, vous avez réussi à envoyer votre code sur le serveur GitHub.

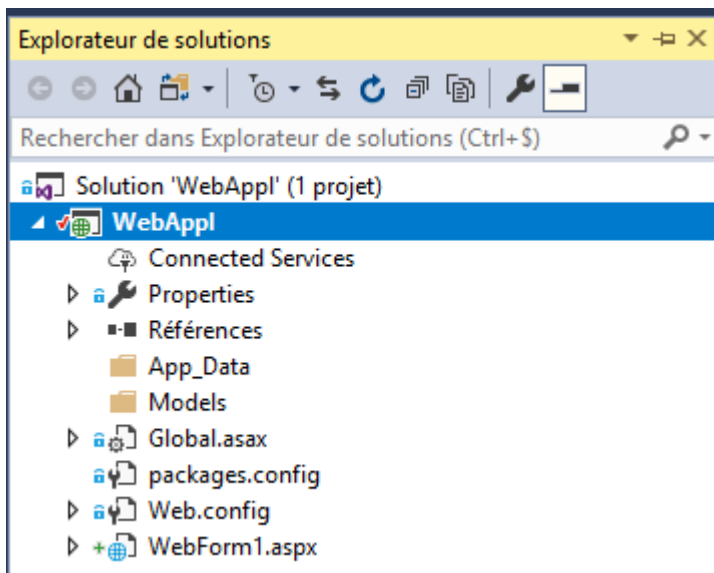
```
C:\Cours-2019\02 - DevOps\01 - Cours DevOps\02 - Demo\Seance 1 - Demo\GitLocal>git remote add origin https://github.com/Mouhanad-Git/GitLocal.git
C:\Cours-2019\02 - DevOps\01 - Cours DevOps\02 - Demo\Seance 1 - Demo\GitLocal>git push origin master
Enumerating objects: 102, done.
Counting objects: 100% (102/102), done.
Delta compression using up to 4 threads
Compressing objects: 100% (93/93), done.
Writing objects: 100% (102/102), 12.94 MiB | 6.70 MiB/s, done.
Total 102 (delta 29), reused 0 (delta 0)
remote: Resolving deltas: 100% (29/29), done.
To https://github.com/Mouhanad-Git/GitLocal.git
 * [new branch]      master -> master
C:\Cours-2019\02 - DevOps\01 - Cours DevOps\02 - Demo\Seance 1 - Demo\GitLocal>
```

Vous pouvez par la suite vérifier votre code sur la repository créée dans l'étape 3.



Nous allons procéder à des modifications au niveau de notre projet (Solution Visual Studio) et faire le merge avec le contrôleur de code source via l'invite de commande Git.

5. Ajoutez une nouvelle page Web Form à votre solution (WebForm1).



6. Puis, exécutez les 3 commandes : Git Add, Git Commit, et Git Push à partir de l'invite de commande Git.

7. Allez au niveau du serveur GitHub et vérifiez si votre code a été reporté sur la version serveur.

Mouhanad EL FILALI Ajout Page Client		Latest commit 3f7bd5d 13 minutes ago
..		
Properties	First Commit	4 hours ago
bin	Ajout Page Client	13 minutes ago
obj/Debug	Ajout Page Client	13 minutes ago
Global.asax	First Commit	4 hours ago
Global.asax.cs	First Commit	4 hours ago
Web.Debug.config	First Commit	4 hours ago
Web.Release.config	First Commit	4 hours ago
Web.config	First Commit	4 hours ago
WebAppl.csproj	First Commit	4 hours ago
WebAppl.csproj.user	First Commit	4 hours ago
WebForm1.aspx	Ajout Page Client	13 minutes ago
WebForm1.aspx.cs	Ajout Page Client	13 minutes ago
WebForm1.aspx.designer.cs	Ajout Page Client	13 minutes ago
packages.config	First Commit	4 hours ago

## II. Gestion des branches avec Git Command Line

L'objectif maintenant est de créer une branche à partir de l'invite de commande, faire des push sur cette nouvelle branche, puis effectuer un merge sur la branche master.

Une fois le merge effectué sur la branche master, nous allons effectuer un clonage (récupération) de la version du code source vers un nouveau répertoire Git local.

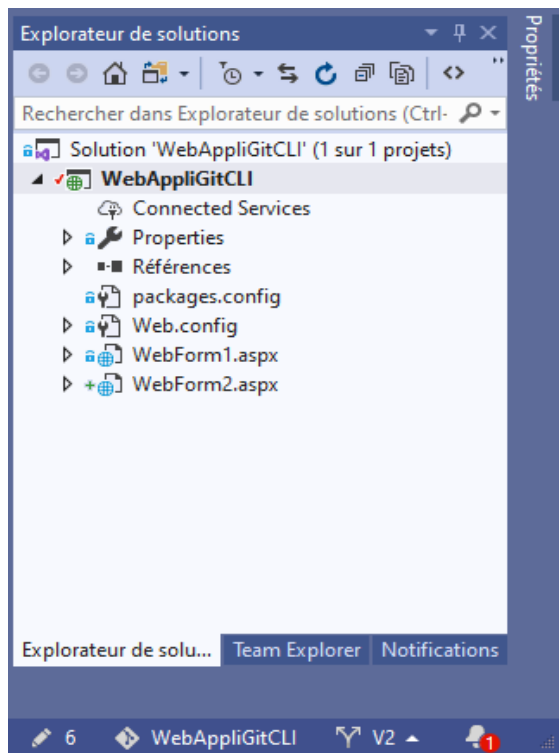
1. La commande git checkout peut être utilisée pour créer des branches ou pour basculer entre elles :

- **git checkout -b <nom-branche>**

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>git checkout -b V2  
Switched to a new branch 'V2'  
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>
```

2. Une fois notre nouvelle branche est créée, nous allons procéder à une MAJ dans notre projet (ajout d'une nouvelle page) et faire un commit sur cette branche.

Sur la figure ci-dessous, nous remarquons l'ajout d'une nouvelle page (WebForm2) et que la version est basculée sur la branche V2 au lieu de la branche master.





















3. Nous devons procéder à un commit des modifications effectuées dans la version actuelle sur la branche V2, pour le faire, il faut réexécuter les commandes Git Add et Git Commit.

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git add .  
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>Git commit -m "MAJ sur la branche V2"  
[V2 8b91358] MAJ sur la branche V2  
2 files changed, 8 insertions(+)  
rewrite .vs/WebAppliGitCLI/v16/.suo (65%)  
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```

- Une fois le commit est effectué, nous pouvons faire un push vers le serveur Git Hub.

```
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git push origin V2
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 4 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (16/16), 9.52 KiB | 1.59 MiB/s, done.
Total 16 (delta 6), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (6/6), completed with 5 local objects.
remote:
remote: Create a pull request for 'V2' on GitHub by visiting:
remote:   https://github.com/Mouhanad-Git/DemoGitCLI/pull/new/V2
remote:
To https://github.com/Mouhanad-Git/DemoGitCLI.git
 * [new branch]      V2 -> V2
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```

- Vous pouvez visualiser vos modifications sur la branche V2 sur GitHub.

 Mouhanad-User MAJ sur la branche V2		8b91358 7 minutes ago	 4 commits
	.vs/WebAppliGitCLI	MAJ sur la branche V2	7 minutes ago
	Properties	First Commit	28 minutes ago
	bin	First Commit	28 minutes ago
	obj/Debug	MAJ sur la branche V2	8 minutes ago
	packages/Microsoft.CodeDom.Provid...	First Commit	28 minutes ago
	Web.Debug.config	First Commit	28 minutes ago
	Web.Release.config	First Commit	28 minutes ago
	Web.config	First Commit	28 minutes ago
	WebAppliGitCLI.csproj	MAJ sur la branche V2	7 minutes ago
	WebAppliGitCLI.csproj.user	First Commit	28 minutes ago
	WebAppliGitCLI.sln	First Commit	28 minutes ago
	WebForm1.aspx	First Commit	28 minutes ago
	WebForm1.aspx.cs	First Commit	28 minutes ago
	WebForm1.aspx.designer.cs	First Commit	28 minutes ago
	WebForm2.aspx	MAJ sur la branche V2	8 minutes ago
	WebForm2.aspx.cs	MAJ sur la branche V2	8 minutes ago

6. Pour lister les branches présentes dans le dépôt Git, utilisez la commande git branch :

- **git branch**

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git branch
* V2
  master

C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```

7. L'objectif maintenant est de faire un merge entre de la branche V2 vers la branche master à partir de l'invite de commande Git, pour le faire, vous devez vous positionner sur la branche master, puis effectuer le merge et faire par la suite un push vers le serveur github.

- **git checkout master**

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git checkout master
Switched to branch 'master'

C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```

8. L'exécution de la commande git log génère le log d'une branche

- **git log**

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>Git log
commit c59a27000aca433da8c709b8d2c1dd61655185ad (HEAD -> master, origin/master)
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 09:41:44 2021 +0100

    MAJ projet

commit 1c840ae05caa9def171f1445156009483ec6cd92
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 09:40:19 2021 +0100

    First Commit

C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```



9. La **commande git merge** est utilisée pour fusionner une branche dans la branche active, dans notre cas, ça sera à partir de la branche V2 vers la branche master, par la suite, vous devez effectuer un push vers le serveur github.

- **git merge V2**

```
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>git merge V2
Updating c59a270..8b91358
Fast-forward
 .vs/WebAppGitCLI/v16/.suo          | Bin 32768 -> 50688 bytes
 WebAppGitCLI.csproj                | 8 ++++++
 WebForm2.aspx                     | 16 ++++++++
 WebForm2.aspx.cs                   | 17 ++++++++
 WebForm2.aspx.designer.cs          | 26 ++++++++
 .../DesignTimeResolveAssemblyReferencesInput.cache | Bin 8378 -> 8457 bytes
 .../WebAppGitCLI.csprojAssemblyReference.cache      | Bin 424 -> 67363 bytes
 7 files changed, 67 insertions(+)
 create mode 100644 WebForm2.aspx
 create mode 100644 WebForm2.aspx.cs
 create mode 100644 WebForm2.aspx.designer.cs
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>
```

10. Pour vérifier l'exécution du merge, vous pouvez lancer la commande **git log** sur la branche master, ci-dessous le résultat après le merge.

```
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>git log
commit 8b91358d6affc92a41d6693b93880d19067f55f3 (HEAD -> master, origin/V2, V2)
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 10:01:00 2021 +0100

    MAJ sur la branche V2

commit a367e63d5a9234268d91c906a1fc61b3fe7f7b58
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 10:00:06 2021 +0100

    MAJ sur la branche V2

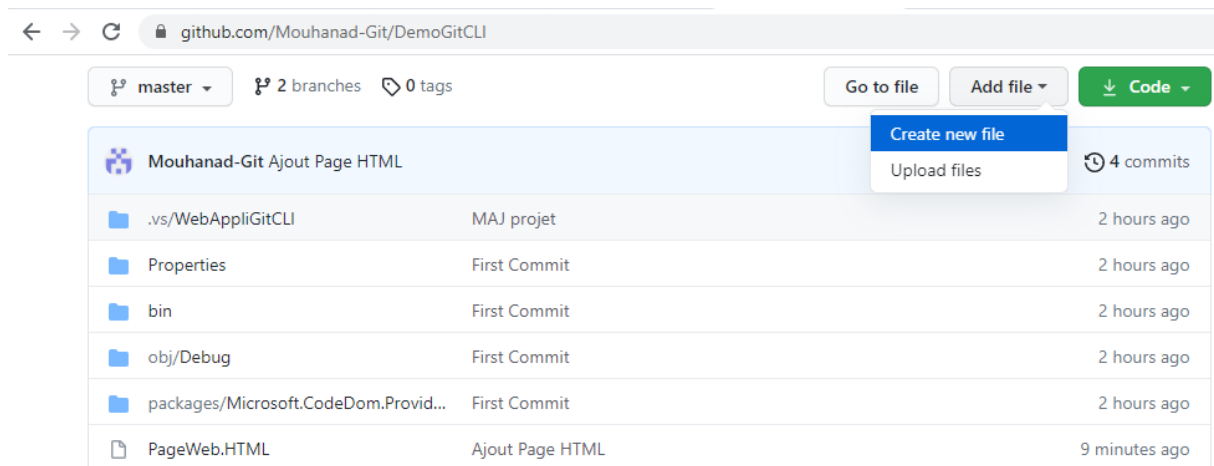
commit c59a27000aca433da8c709b8d2c1dd61655185ad (origin/master)
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 09:41:44 2021 +0100

    MAJ projet

commit 1c840ae05caa9def171f1445156009483ec6cd92
Author: Mouhanad EL FILALI <mouhanadelfilali@hotmail.com>
Date: Fri Apr 2 09:40:19 2021 +0100

    First Commit
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppGitCLI>
```

11. La commande **git pull** permet à un utilisateur d'extraire tous les fichiers du dépôt distant qui ne sont pas actuellement dans le répertoire de travail local, pour faire un exemple, ajouter un fichier HTML directement à partir du serveur Github, dans mon exemple j'ai rajouté la page « PageWeb.HTML ».



12. Puis exécuter la commande git pull

- **git pull origin master**

```
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git pull origin master
From https://github.com/Mouhanad-Git/DemoGitCLI
* branch      master      -> FETCH_HEAD
Merge made by the 'recursive' strategy.
 PageWeb.HTML | 1 +
 README.md    | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 PageWeb.HTML
 create mode 100644 README.md
C:\Users\dell\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```

13. Vous pouvez vérifier par la suite la présence de la pageweb.HTML dans votre répertoire local et votre Solution Visual studio.

14. Pour créer une copie de dépôt de travail local à partir d'un autre git local, il faut créer un nouveau répertoire de clonage, puis lancer la commande `git init` sur ce nouveau répertoire, se positionner dans ce dernier et exécuter la commande `git clone` à partir d'un git local existant.

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitClineCLI>git clone C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI
Cloning into 'WebAppliGitCLI'...
done.
C:\Users\de11\Desktop\Demo-DevOps\DemoGitClineCLI>
```

15. Pour supprimer une branche :

- **`git branch -d <nom-branche>`**

```
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>git branch -d V2
Deleted branch V2 (was 8b91358).
C:\Users\de11\Desktop\Demo-DevOps\DemoGitCLI\WebAppliGitCLI>
```