Contents

[Using GitHub 1](#_Toc498584487)

[Team Source Management Tips 1](#_Toc498584488)

[Extra Backup 1](#_Toc498584489)

[About the Visual Studio GitHub Plugin and How to Install It 2](#_Toc498584490)

[Installing the GitHub Extension 2](#_Toc498584491)

[Adding New Solutions to Source Control 2](#_Toc498584492)

[Logging in to GitHub 3](#_Toc498584493)

[Publishing Changes 4](#_Toc498584494)

[Getting Your Repository Address 6](#_Toc498584495)

[Adding Collaborators 7](#_Toc498584496)

[Saving Your Changes to the Repository 7](#_Toc498584497)

[Getting Changes from the Repository 10](#_Toc498584498)

[Opening an Existing Project from GitHub 10](#_Toc498584499)

[Reviewing File History 12](#_Toc498584500)

[Rolling Back Changes 13](#_Toc498584501)

[Allowing Automatic Downloading of NuGet Packages 14](#_Toc498584502)

# Using GitHub

Sign up at. <https://education.github.com/pack> with your **my.bcit.ca** email address.

## Team Source Management Tips

* Before starting a new code session and at the start of each day, retrieve the latest copy of the team repository from GitHub. DO NOT START A CODING SESSION WITH OLD LOCAL CODE.
* Keep a current zipped archive of your working project locally at all times just in case your online repository gets corrupted.
* When ending a coding session commit your changes. Don’t go home without committing your work.

### Extra Backup

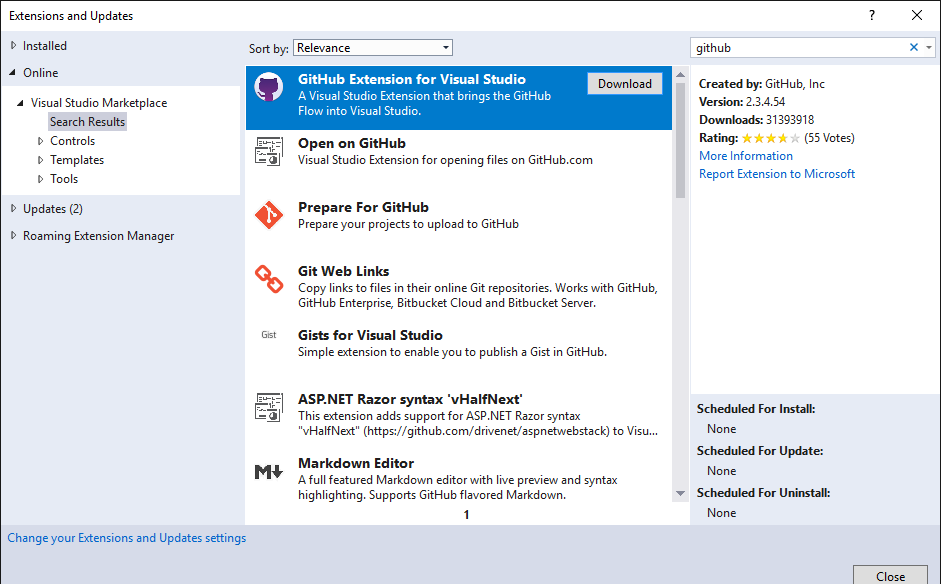
Do not only rely on GitHub. When you have a project that works locally which you like, zip the project and store the zip archive on your machine just in case your project on GitHub gets corrupted.

## About the Visual Studio GitHub Plugin and How to Install It

You will hear some people say you should use the command line to control GitHub. There is definitely merit in using the command line but you are going to want to use the GitHub Visual Studio plug-in too. When we get into advanced .NET projects the VS GitHub plug-in automates management of these files. You will be dealing with tonnes of binary files and it is virtually impossible to manage an advanced.NET application without the plug-in – please use it for your .NET projects.

### Installing the GitHub Extension

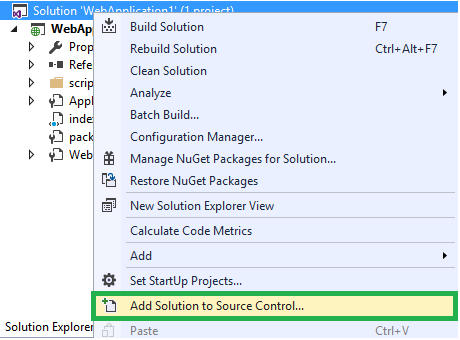
To obtain access to GitHub, search for the GitHub extension from Tools | Extensions and Updates. Then expand the online tab and search for GitHub.



Once selected, you will be prompted to shut Visual Studio down. Then give permission when asked if you want to modify Visual Studio. After Visual Studio will download and install the plug in. This could take a while so start early today and work on your designs while you wait.

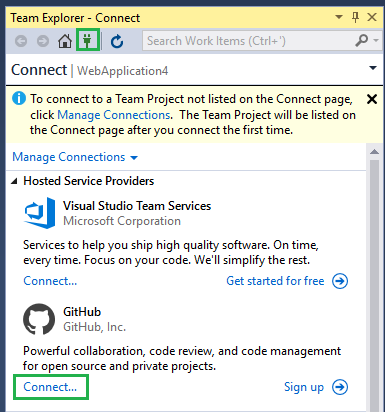
## Adding New Solutions to Source Control

First create your a ASP.NET Core Web Application. Choose the Model View Controller option when prompted. Right click solution and choose 'Add to source control'.



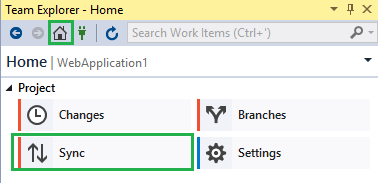
## Logging in to GitHub

To log in to GitHub select the connection icon and choose connect.

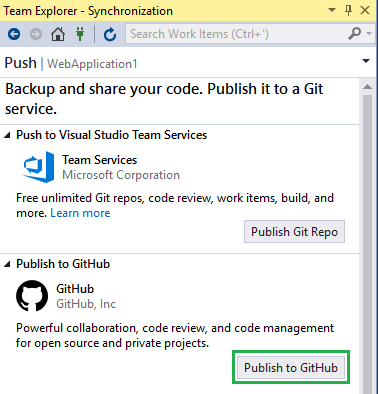


## Publishing Changes

If you have added your project to source control as described above you can publish changes to it. To do this, in the Team Explorer, from the Home tab you can choose Sync.

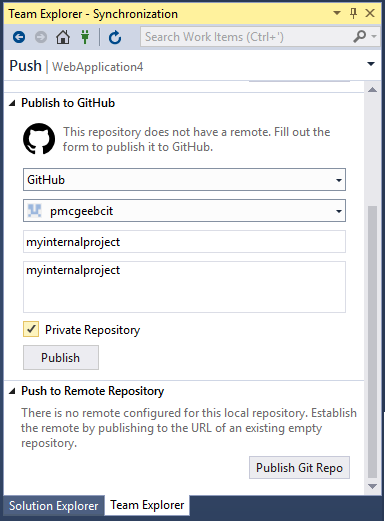


From there you can choose Publish to GitHub:

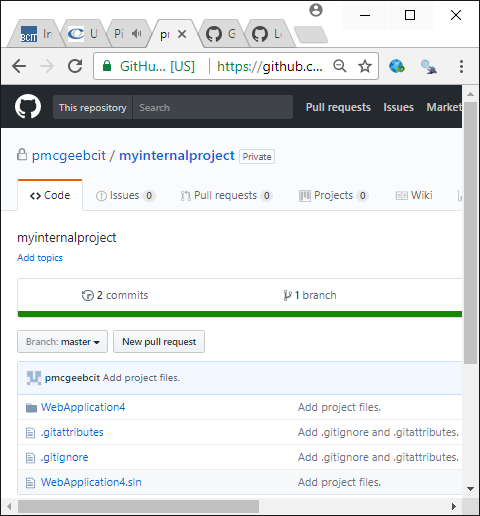


When prompted login to GitHub, give your project repository a name, select private, and choose Publish. If you do not have a private repository you can get one through GitHub education:

<https://education.github.com/>

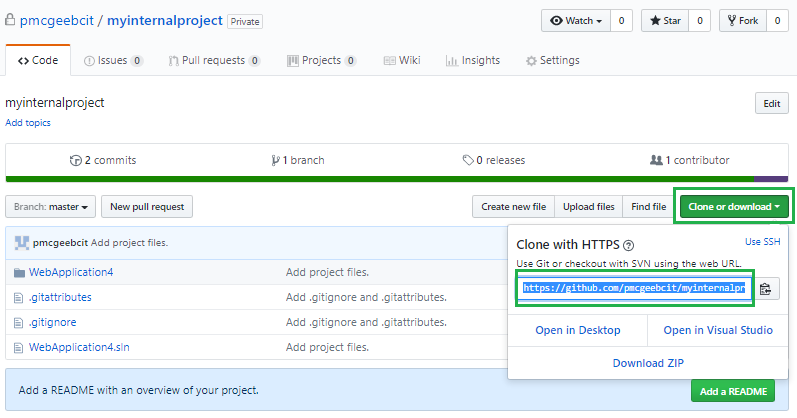


This action will create your repository on GitHub.



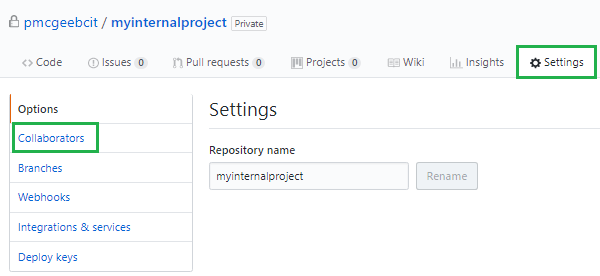
## Getting Your Repository Address

You can find your repository address by clicking Clone or Download.



## Adding Collaborators

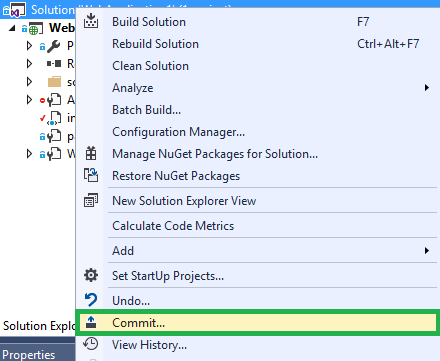
To add people to work on shared code with you, choose **Settings**.



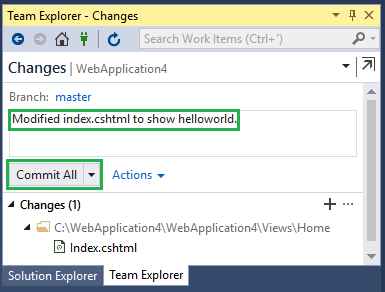
Then choose **Collaborators** on the left. Next, enter the user name of the person you like to give write privileges and click **Add Collaborator**.

## Saving Your Changes to the Repository

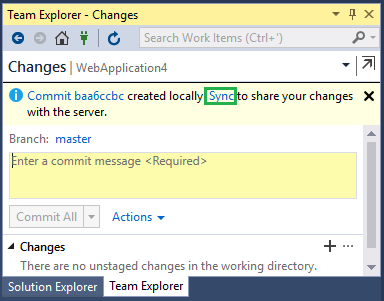
Make some changes to your index.html file. Right click the solution and choose 'Commit'.



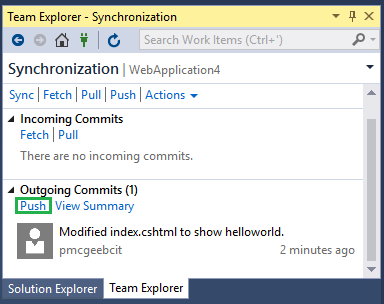
When prompted enter a description of your change and choose Commit All.



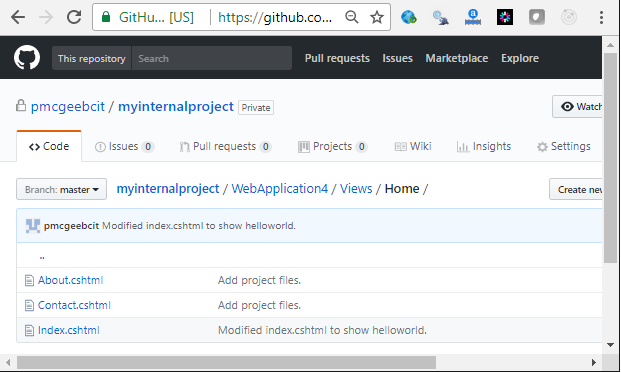
After click Press Synch.



Click Push.

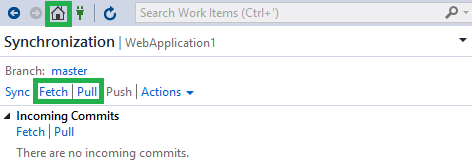


Now when you go to GitHub you will see the changes have been applied in the repository online.



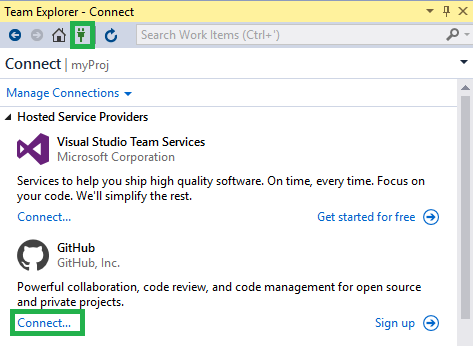
## Getting Changes from the Repository

To get changes from the repository, you can choose **Fetch** or **Pull** from the **Home | Synch** page in **Team Explorer**. Fetch lists the changes without integrating them in your project. Pull integrates the changes into your project.



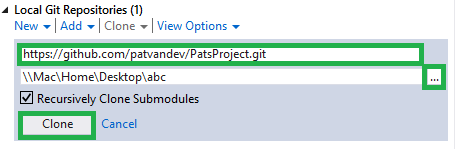
## Opening an Existing Project from GitHub

Click the connections icon. Then click Connect under GitHub and login when prompted.

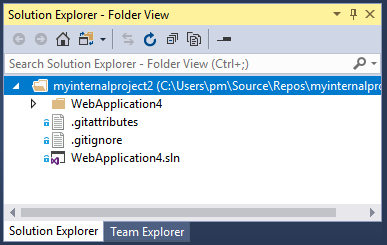


Then, after logging in, from the connections tab click Clone and enter your repository address.

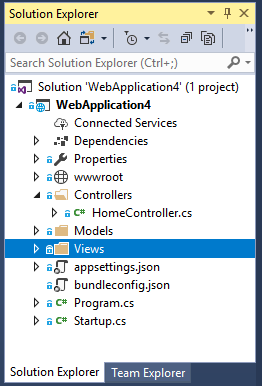
At the start of every coding session especially at the start of the day, to ensure you always get a fresh clean build I strongly recommend that you create a new folder some place where you can find it. Then click Clone. Do not clone to an existing local folder with code in it – otherwise you risk polluting your project with old out-of-date or broken code.



After, navigate to the Solution Explorer and click the solution file to launch it.

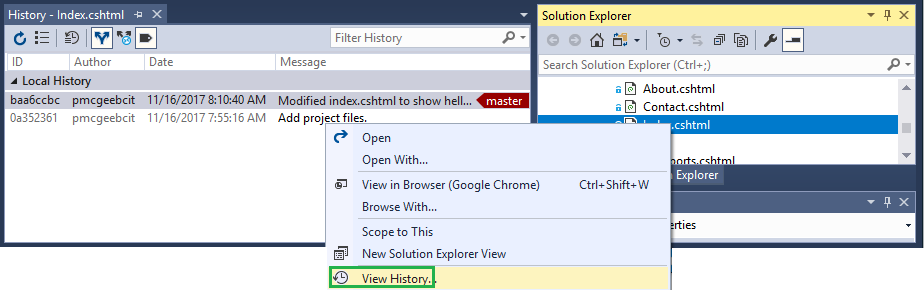


After you will see your project files in the solution explorer. You can now start modifying the files and pushing your changes.

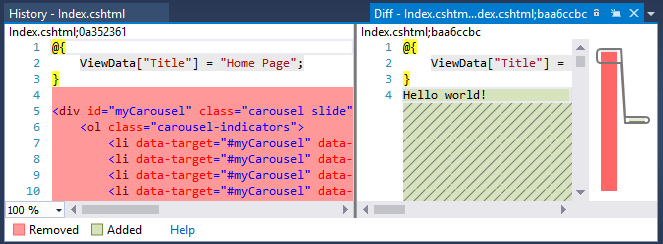


## Reviewing File History

Sometimes you will want to examine the history of a file. To do this, right click your code file and choose View History.

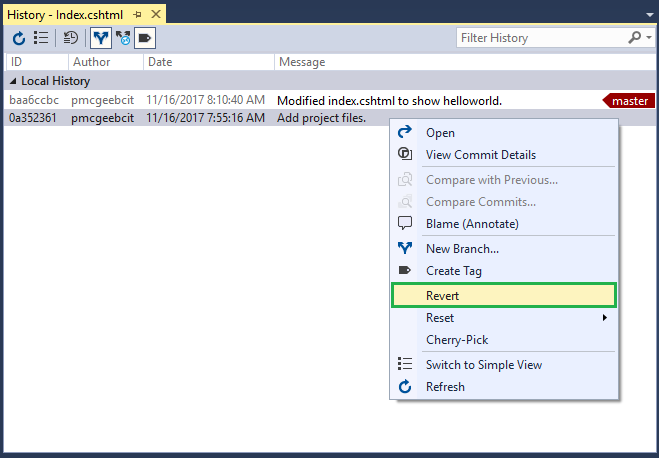


Clicking any commit allows you to compare that version with a previous version.



## Rolling Back Changes

To revert to an older version, right click the code file and choose history and then right click on the entry you wish to restore and choose revert.



## Allowing Automatic Downloading of NuGet Packages

It is really important to ensure you are downloading NuGet packages rather than storing them in source control. To allow Visual Studio to obtain the proper NuGet package go to Tools | Options |NuGet Package Manager | General. Make sure that the following items shown in the screenshot are checked.

