GIT en GitHub – startup

Installeer Git: https://git-scm.com/downloads

Video installatie Git:

https://www.youtube.com/watch?v=J_Clau1bYco

GitHub account setup

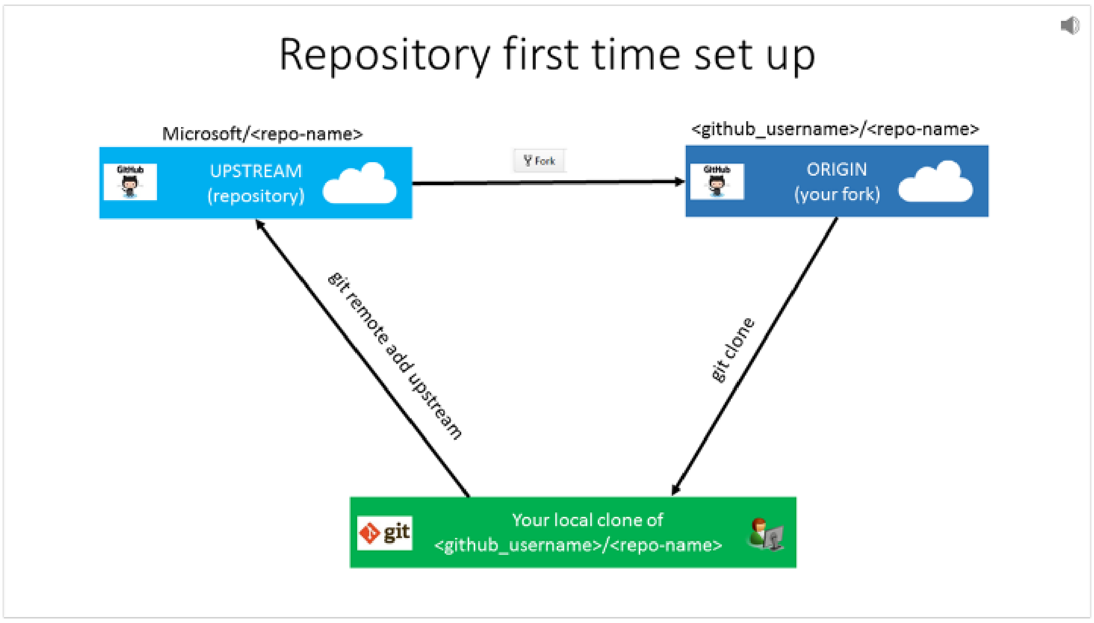
https://docs.microsoft.com/en-us/contribute/get-started-setup-github

1. Create a GitHub account and set up your profile (www.github.com)

2. Install Git (https://git-scm.com/) en Git Bash (https://www.atlassian.com/git/tutorials/git-bash) en eventueel Git for windows (https://gitforwindows.org/)

3. Setup local Git Repository

   https://docs.microsoft.com/en-us/contribute/get-started-setup-local


Repository first time set up
Microsoft/<repo-name>
UPSTREAM (repository)
Fork
<github_username>/<repo-name>
ORIGIN (your fork)
git remote add upstream
git clone
git
Your local clone of <github_username>/<repo-name>

Setup Git en GitHub

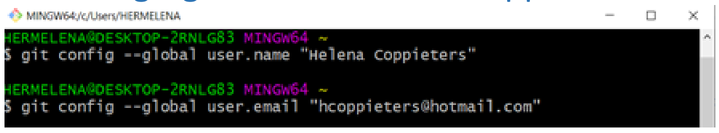
Download and install the latest version of GitHub Desktop(https://desktop.github.com/). This will automatically install Git *and* keep it up-to-date for you.
https://help.github.com/articles/set-up-git/

Git commands using git bash

1. Open git bash
2. Set global username and email

   Git config --global user.name "Helena Coppieters"

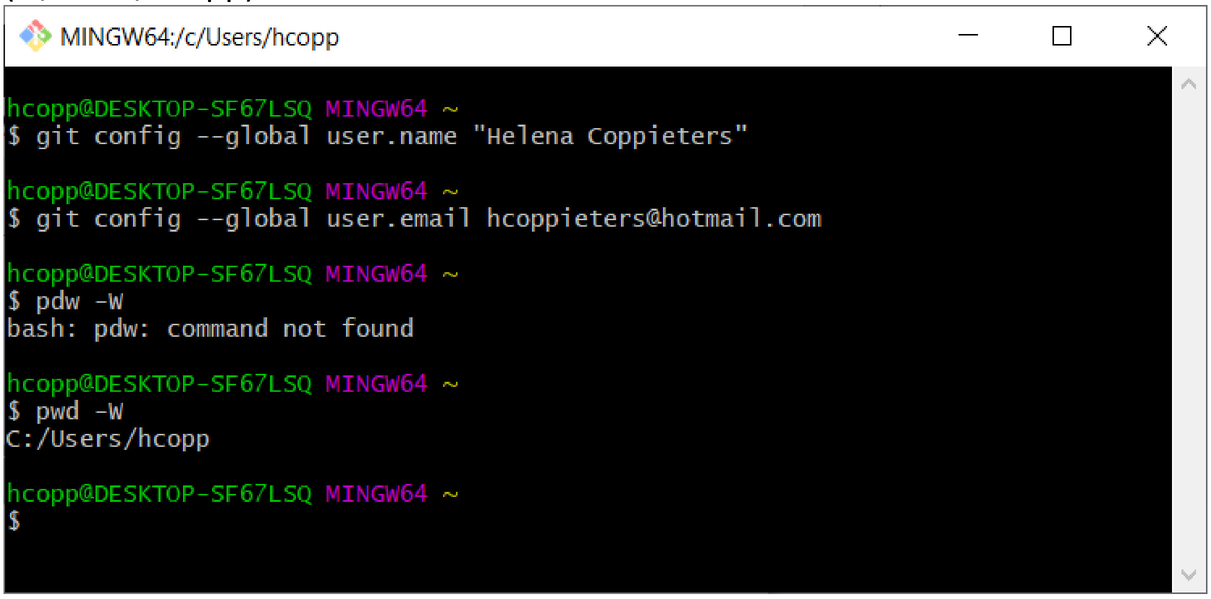
   Git config --global user.email hcoppieters@hotmail.com

   ```
   HERMELENA@DESKTOP-2RNLG83 MINGW64 ~
   $ git config --global user.name "Helena Coppieters"

   HERMELENA@DESKTOP-2RNLG83 MINGW64 ~
   $ git config --global user.email "hcoppieters@hotmail.com"
   ```

3. Get current git directory in git bash (hoofdletter gevoelig!)

   $ pwd -W
   C:/Program Files/Git
   (c:/users/hcopp)

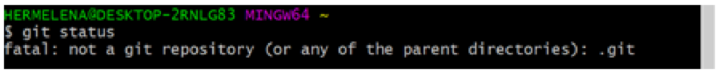
   ```
   hcopp@DESKTOP-SF67LSQ MINGW64 ~
   $ git config --global user.name "Helena Coppieters"

   hcopp@DESKTOP-SF67LSQ MINGW64 ~
   $ git config --global user.email hcoppieters@hotmail.com

   hcopp@DESKTOP-SF67LSQ MINGW64 ~
   $ pdw -W
   bash: pdw: command not found

   hcopp@DESKTOP-SF67LSQ MINGW64 ~
   $ pwd -W
   C:/Users/hcopp

   hcopp@DESKTOP-SF67LSQ MINGW64 ~
   $
   ```

4. Ga naar de juist directory waar je je code bestanden staan die je in source control wil zetten :

   $ cd source/repos
   (het commando cd zal naar een subdirectory gaan)
   $ pwd -W
   (c:/users/hcopp/source/repos)

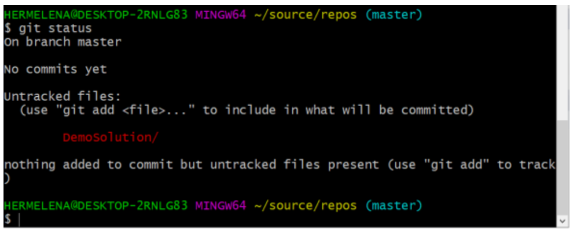
5. gitGit status:

   indien niet geïnitaliseerd geeft deze een melding:



```
HERMELENA@DESKTOP-2RNLG83 MINGW64 ~
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

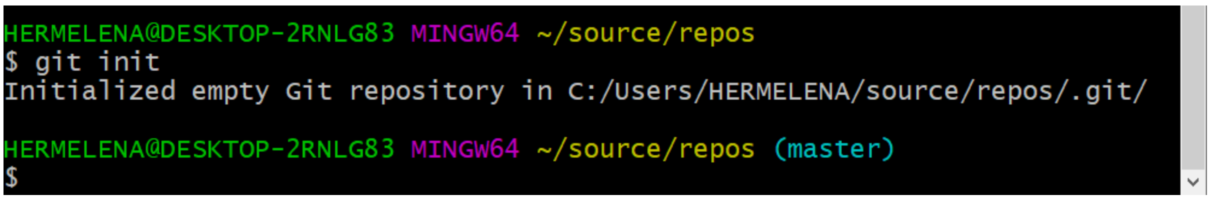
   indien wel geïnitialiseerd geeft deze een lijst van bestanden
   in de working directory die wel en niet aan de source control zijn toegevoegd:



```
HERMELENA@DESKTOP-2RNLG83 MINGW64 ~/source/repos (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        DemoSolution/

nothing added to commit but untracked files present (use "git add" to track
)

HERMELENA@DESKTOP-2RNLG83 MINGW64 ~/source/repos (master)
$ |
```

6. Initialize Git

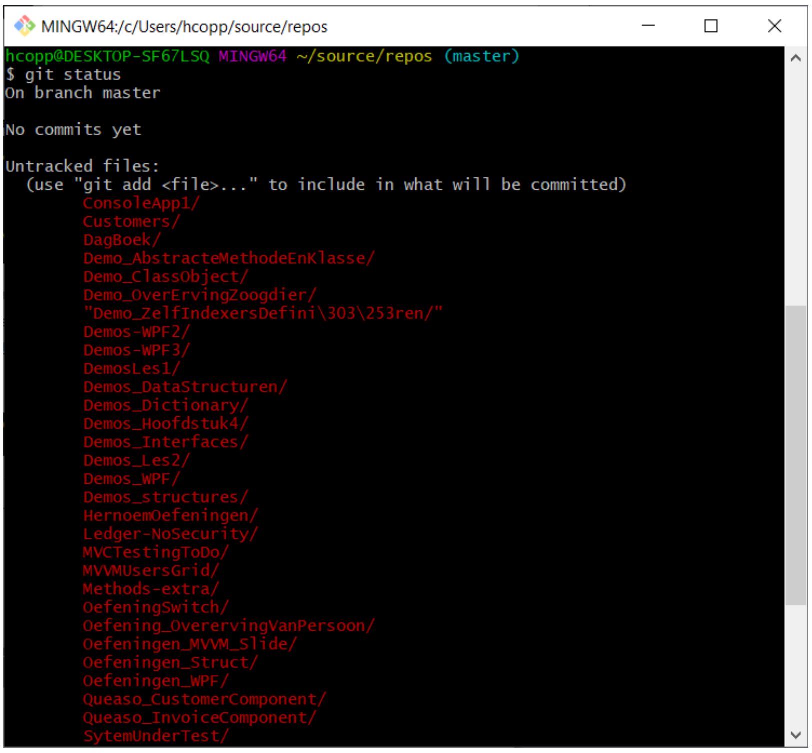
   Deze zal een .git repository aanmaken in de current git working directory



```
HERMELENA@DESKTOP-2RNLG83 MINGW64 ~/source/repos
$ git init
Initialized empty Git repository in C:/Users/HERMELENA/source/repos/.git/

HERMELENA@DESKTOP-2RNLG83 MINGW64 ~/source/repos (master)
$
```

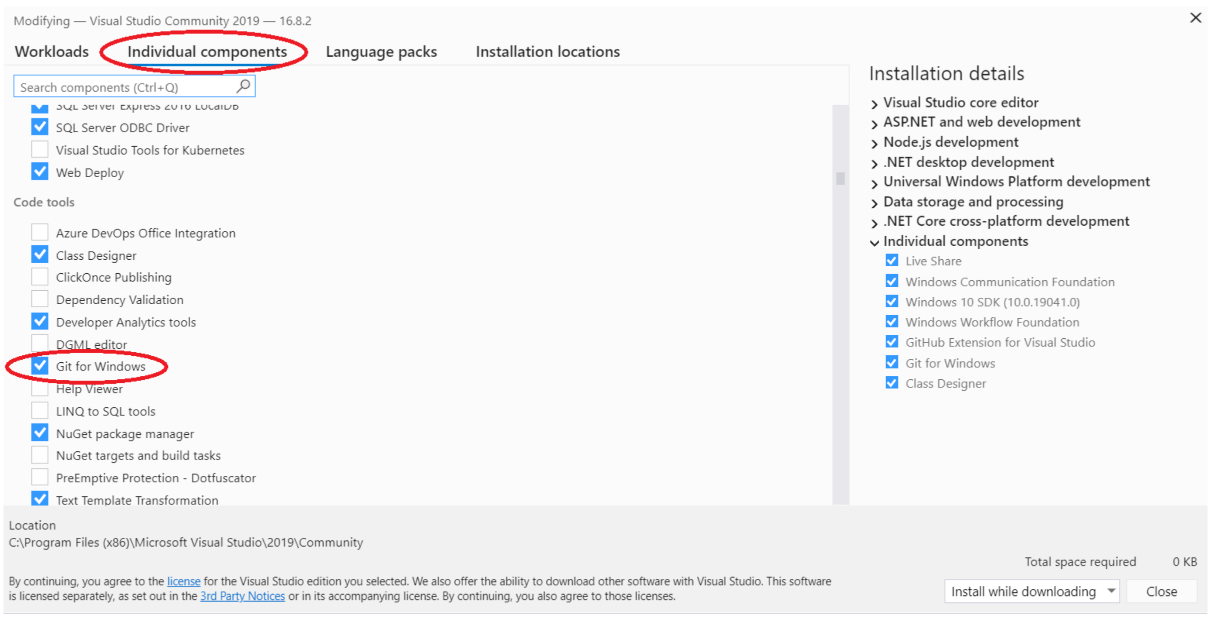
git init initializes a brand new Git repository and begins tracking an existing directory. It adds a hidden subfolder within the existing directory that houses the internal data structure required for version control.(https://guides.github.com/introduction/git-handbook/)

Vraag hierna de opnieuw de git status op :



```
MINGW64:/c/Users/hcopp/source/repos
hcopp@DESKTOP-SF67LSQ MINGW64 ~/source/repos (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ConsoleApp1/
        Customers/
        DagBoek/
        Demo_AbstracteMethodeEnKlasse/
        Demo_ClassObject/
        Demo_OverErvingZoogdier/
        "Demo_ZelfIndexersDefini\303\253ren/"
        Demos-WPF2/
        Demos-WPF3/
        DemosLes1/
        Demos_DataStructuren/
        Demos_Dictionary/
        Demos_Hoofdstuk4/
        Demos_Interfaces/
        Demos_Les2/
        Demos_WPF/
        Demos_structures/
        HernoemOefeningen/
        Ledger-NoSecurity/
        MVCTestingToDo/
        MVVMUsersGrid/
        Methods-extra/
        OefeningSwitch/
        Oefening_OverervingVanPersoon/
        Oefeningen_MVVM_Slide/
        Oefeningen_Struct/
        Oefeningen_WPF/
        Queaso_CustomerComponent/
        Queaso_InvoiceComponent/
        SytemUnderTest/
```

Nu kan je je Solution openen in Visual Studio 2019. Indien je de Git en Github extensies reeds hebt geïnstalleer kan je het volgende doen (indien niet, installeer eerst de Git en github extensies via VS Installer):


Modifying — Visual Studio Community 2019 — 16.8.2

Workloads | Individual components | Language packs | Installation locations

Search components (Ctrl+Q)

- ✓ SQL Server Express 2016 LocalDB
- ✓ SQL Server ODBC Driver
- ☐ Visual Studio Tools for Kubernetes
- ✓ Web Deploy

Code tools

- ☐ Azure DevOps Office Integration
- ✓ Class Designer
- ☐ ClickOnce Publishing
- ☐ Dependency Validation
- ✓ Developer Analytics tools
- ☐ DGML editor
- ✓ Git for Windows
- ☐ Help Viewer
- ☐ LINQ to SQL tools
- ✓ NuGet package manager
- ☐ NuGet targets and build tasks
- ☐ PreEmptive Protection - Dotfuscator
- ✓ Text Template Transformation

Installation details

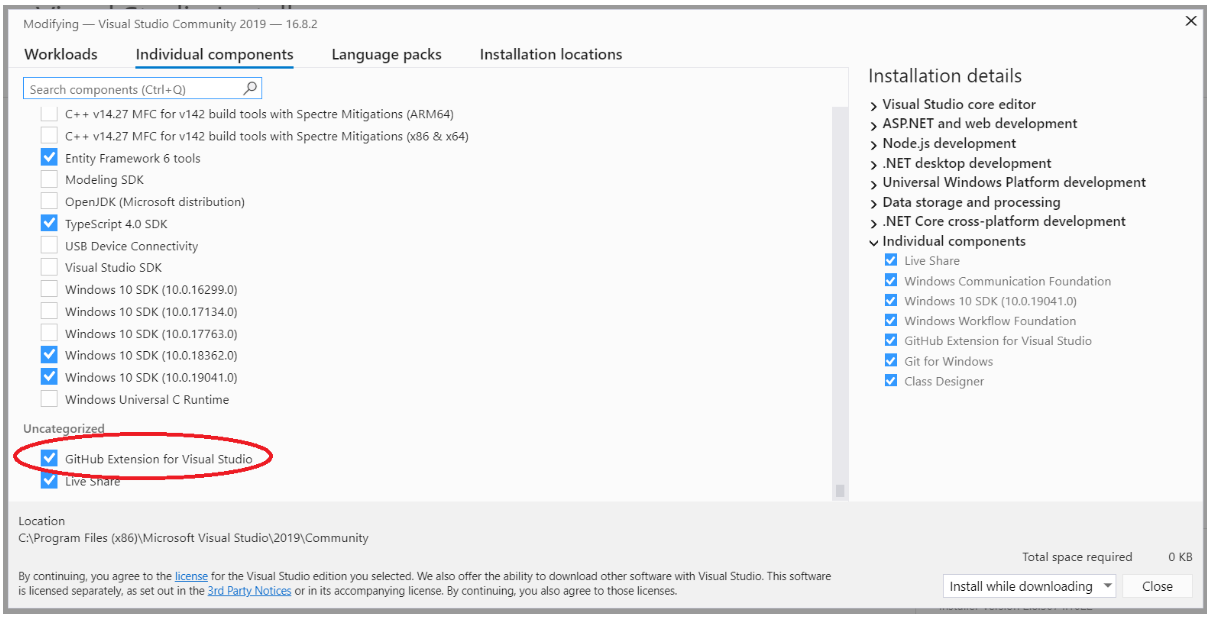
- › Visual Studio core editor
- › ASP.NET and web development
- › Node.js development
- › .NET desktop development
- › Universal Windows Platform development
- › Data storage and processing
- › .NET Core cross-platform development
- ⌄ Individual components
  - ✓ Live Share
  - ✓ Windows Communication Foundation
  - ✓ Windows 10 SDK (10.0.19041.0)
  - ✓ Windows Workflow Foundation
  - ✓ GitHub Extension for Visual Studio
  - ✓ Git for Windows
  - ✓ Class Designer

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

Total space required 0 KB

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Install while downloading | Close


Modifying — Visual Studio Community 2019 — 16.8.2

Workloads | Individual components | Language packs | Installation locations

Search components (Ctrl+Q)

- ☐ C++ v14.27 MFC for v142 build tools with Spectre Mitigations (ARM64)
- ☐ C++ v14.27 MFC for v142 build tools with Spectre Mitigations (x86 & x64)
- ✓ Entity Framework 6 tools
- ☐ Modeling SDK
- ☐ OpenJDK (Microsoft distribution)
- ✓ TypeScript 4.0 SDK
- ☐ USB Device Connectivity
- ☐ Visual Studio SDK
- ☐ Windows 10 SDK (10.0.16299.0)
- ☐ Windows 10 SDK (10.0.17134.0)
- ☐ Windows 10 SDK (10.0.17763.0)
- ✓ Windows 10 SDK (10.0.18362.0)
- ✓ Windows 10 SDK (10.0.19041.0)
- ☐ Windows Universal C Runtime

Uncategorized

- ✓ GitHub Extension for Visual Studio
- ✓ Live Share

Installation details

- › Visual Studio core editor
- › ASP.NET and web development
- › Node.js development
- › .NET desktop development
- › Universal Windows Platform development
- › Data storage and processing
- › .NET Core cross-platform development
- ⌄ Individual components
  - ✓ Live Share
  - ✓ Windows Communication Foundation
  - ✓ Windows 10 SDK (10.0.19041.0)
  - ✓ Windows Workflow Foundation
  - ✓ GitHub Extension for Visual Studio
  - ✓ Git for Windows
  - ✓ Class Designer

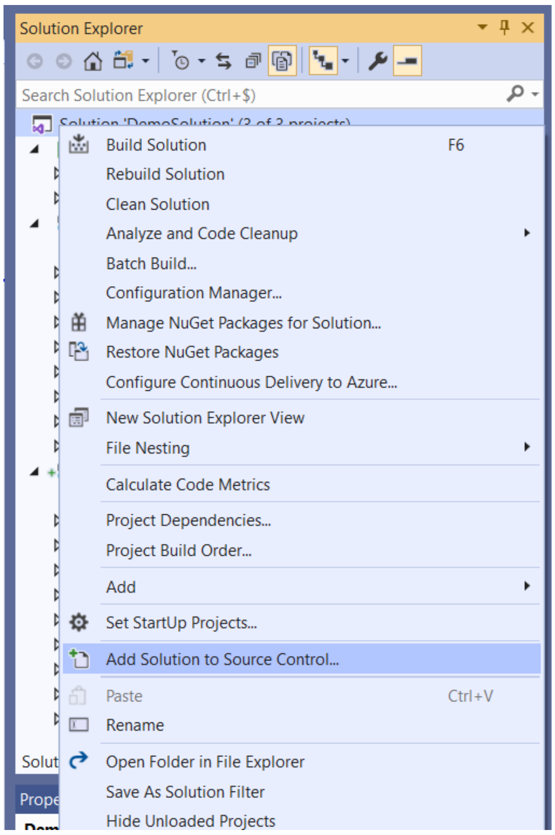
Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

Total space required 0 KB

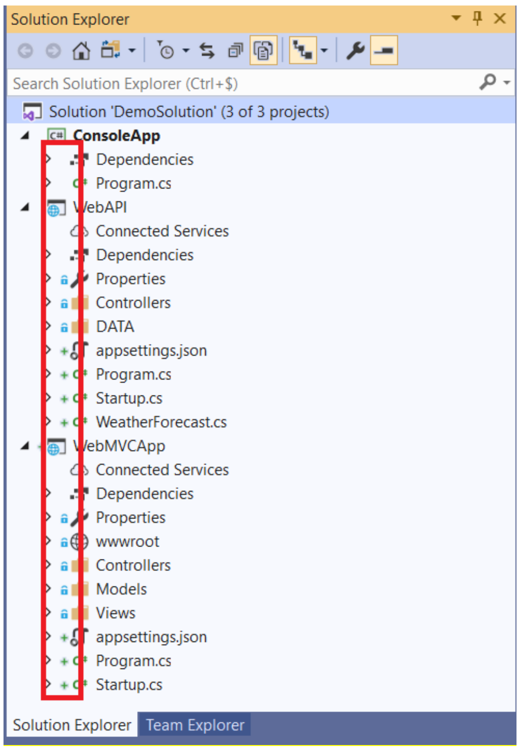
By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Install while downloading | Close

Open je project in VS 2019. In de Solution Explorer, right-click met de muis op de Solution en kies in het context popup-menu «Add solution to source control»



Indien het toevoegen aan de Git source control is gelukt, zal je naast de meeste items in de solution explorer een sleutel of + icoontje zien staan :



Git add stages a change. Git tracks changes to a developer’s codebase, but it’s necessary to stage and take a snapshot of the changes to include them in the project’s history. This command performs staging, the first part of that two-step process. Any changes that are staged will become a part of the next snapshot and a part of the project’s history. Staging and committing separately gives developers complete control over the history of their project without changing how they code and work.

Git commit saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. Anything that’s been staged with `git add` will become a part of the snapshot with `git commit`.

- **git branch** shows the branches being worked on locally.
- **git merge** merges lines of development together. This command is typically used to combine changes made on two distinct branches. For example, a developer would merge when they want to combine changes from a feature branch into the master branch for deployment.
- **git pull** updates the local line of development with updates from its remote counterpart. Developers use this command if a teammate has made commits to a branch on a remote, and they would like to reflect those changes in their local environment.
- **git push** updates the remote repository with any commits made locally to a branch.

Learn more from a full reference guide to Git commands.

Explore more Git commands

For a detailed look at Git practices, the videos below show how to get the most out of some Git commands.

- Working locally
- **git status**
- Two-step commits
- **git pull** and **git push**

How GitHub fits in

GitHub is a Git hosting repository that provides developers with tools to ship better code through command line features, issues (threaded discussions), pull requests, code review, or the use of a collection of free and for-purchase apps in the GitHub Marketplace. With collaboration layers like the GitHub flow, a community of 15 million developers, and an ecosystem with hundreds of integrations, GitHub changes the way software is built.

How GitHub works

GitHub builds collaboration directly into the development process. Work is organized into repositories, where developers can outline requirements or direction and set expectations for team members. Then, using the GitHub flow, developers simply create a branch to work on updates, commit changes to save them, open a pull request to propose and discuss changes, and merge pull requests once everyone is on the same page.

The GitHub flow

The GitHub flow is a lightweight, branch-based workflow built around core Git commands used by teams around the globe—including ours.

The GitHub flow has six steps, each with distinct benefits when implemented:

1. **Create a branch:** Topic branches created from the canonical deployment branch (usually `master`) allow teams to contribute to many parallel efforts. Short-lived topic branches, in particular, keep teams focused and results in quick ships.
2. **Add commits:** Snapshots of development efforts within a branch create safe, revertible points in the project's history.
3. **Open a pull request:** Pull requests publicize a project's ongoing efforts and set the tone for a transparent development process.
4. **Discuss and review code:** Teams participate in code reviews by commenting, testing, and reviewing open pull requests. Code review is at the core of an open and participatory culture.
5. **Merge:** Upon clicking merge, GitHub automatically performs the equivalent of a local 'git merge' operation. GitHub also keeps the entire branch development history on the merged pull request.
6. **Deploy:** Teams can choose the best release cycles or incorporate continuous integration tools and operate with the assurance that code on the deployment branch has gone through a robust workflow.

Learn more about the GitHub flow

Developers can find more information about the GitHub flow in the resources provided below.

- Interactive guide
- GitHub Flow video

Overzicht git commands

https://git-scm.com/docs

Git video's

How to Install and Configure Git and GitHub on Windows

https://www.youtube.com/watch?v=J_Clau1bYco

Team Services Git Tutorial

https://channel9.msdn.com/series/Team-Services-Git-Tutorial

Collaborate using GitHub and Visual Studio 2019 - Pull Requests and Issues

https://www.youtube.com/watch?v=qlxnwgLdc5I

How to work in a team: version control and git

https://blog.hipolabs.com/how-to-work-in-a-team-version-control-and-git-923dfec2ac3b