# GIT

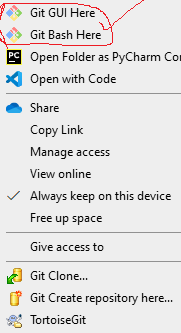
1. [Prerequisite](#_Prerequisite:)
   1. [Git](#_Git)
   2. [TortoiseGit](#_TortoiseGit)
   3. [IDE](#_IDE)
2. [Git Repository](#_Git_Repository)
3. [Git Branch](#_Branch)
   1. [See Current Branch – Preferably use your Git Bash](#_See_Current_Branch)
   2. [Synchronize – Online Branch to Local](#_Synchronize_-_Online)
   3. [Change Branch – Preferably use your IDE](#_Change_Branch_–)
      1. [Eclipse](#_Eclipse)
      2. [IntellIJ](#_IntellIJ)
      3. [Visual Studio](#_Visual_Studio)
      4. [VS Code](#_VS_Code)
4. [Git Clone – Preferably use TortoiseGit](#_Git_Clone_–)
5. [Git Pull – Preferably use TortoiseGit](#_Git_Pull_–)
6. [Git Commit – Preferably use TortoiseGit](#_Git_Commit_–)
7. [Git Push – Preferably use TortoiseGit](#_Git_Push_–)
8. [Git Merge – Preferably use Git Bash](#_Git_Merge_–)
   1. [Conflict Resolve – Preferably use TortoiseGit](#_Conflict_Resolve_–)
9. [Daily](#_Daily)

## Prerequisite

### Git

git bash - <https://git-scm.com/downloads>

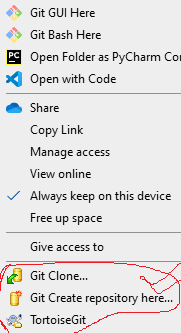
On installing git bash these two options will appear in right click menu. Right click has to be in blank space of the folder. Not on any file or folder.



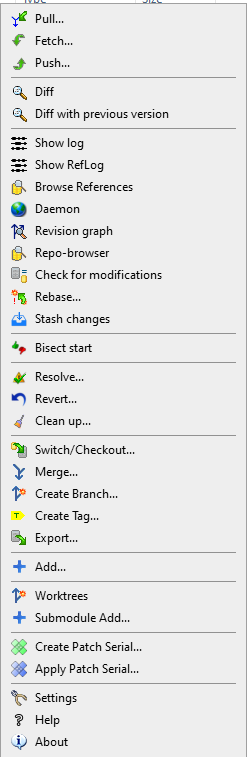
### TortoiseGit

Tortoise git - <https://tortoisegit.org/download/>

On installing tortoise git these options will appear in right click menu.



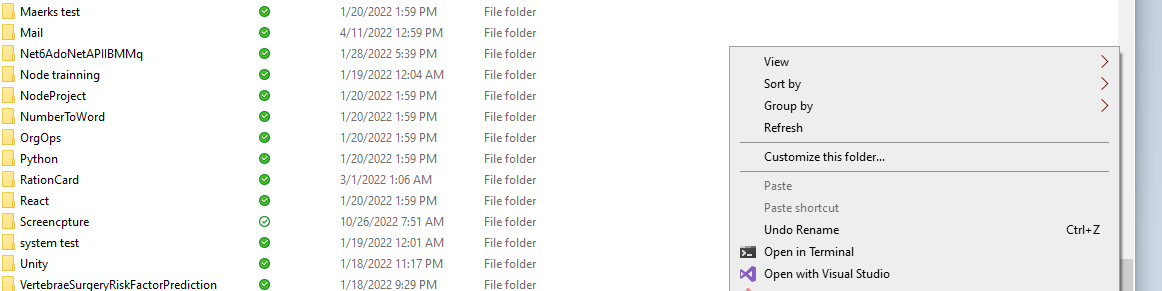
Under TortoiseGit menu options



### IDE

IDE -> Visual Studio, VS Code, Eclipse, and IntellIj etc.

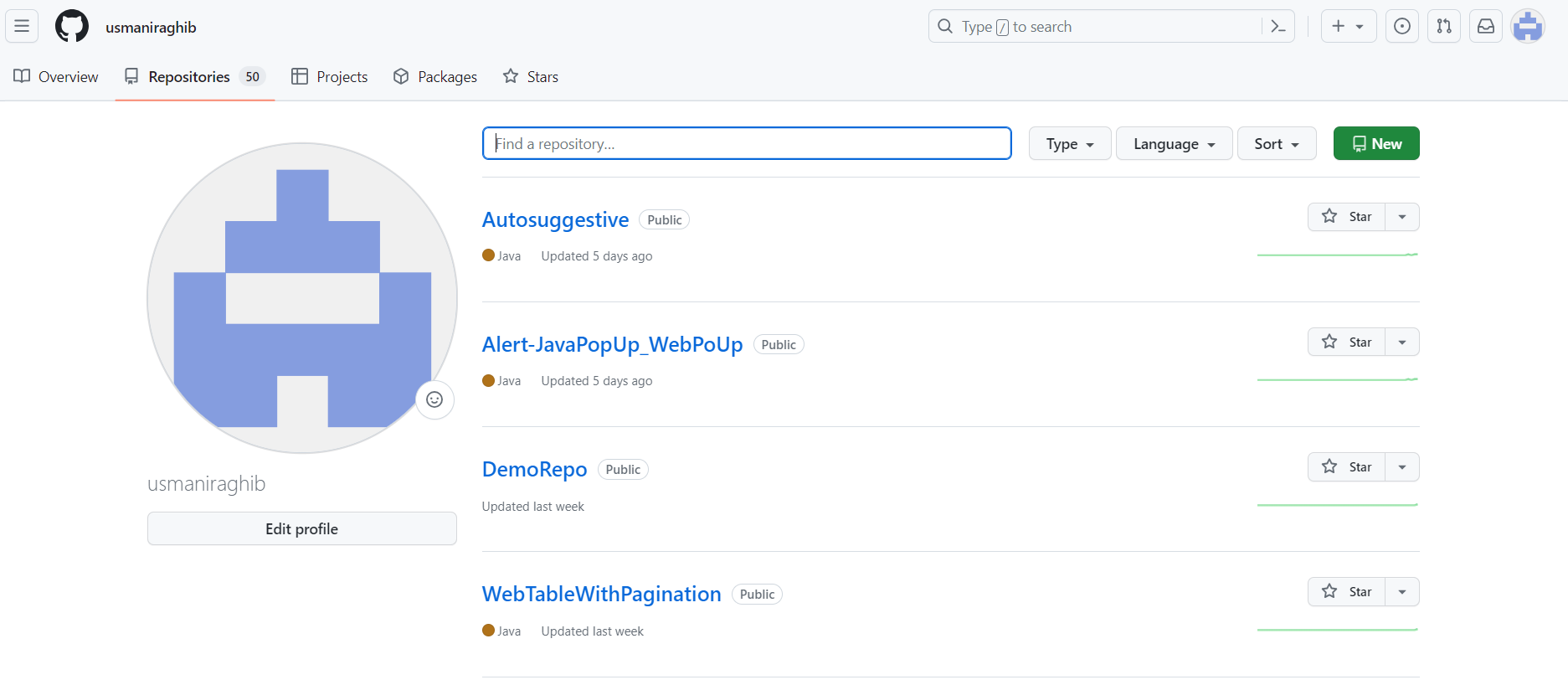
Now in this document we will use mix of all three ways [**Gitbash, TortoiseGit and IDE**] so that we run minimul command with options. Git bash is ultimate but has many options to pass on and itself is a big subject. So please be strict with the process for ease of development.

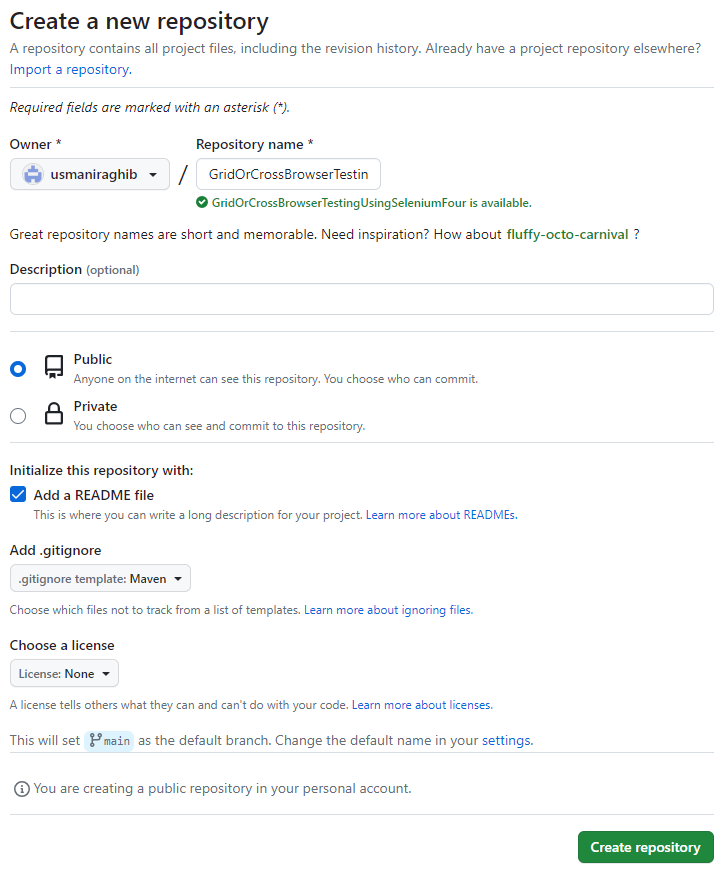


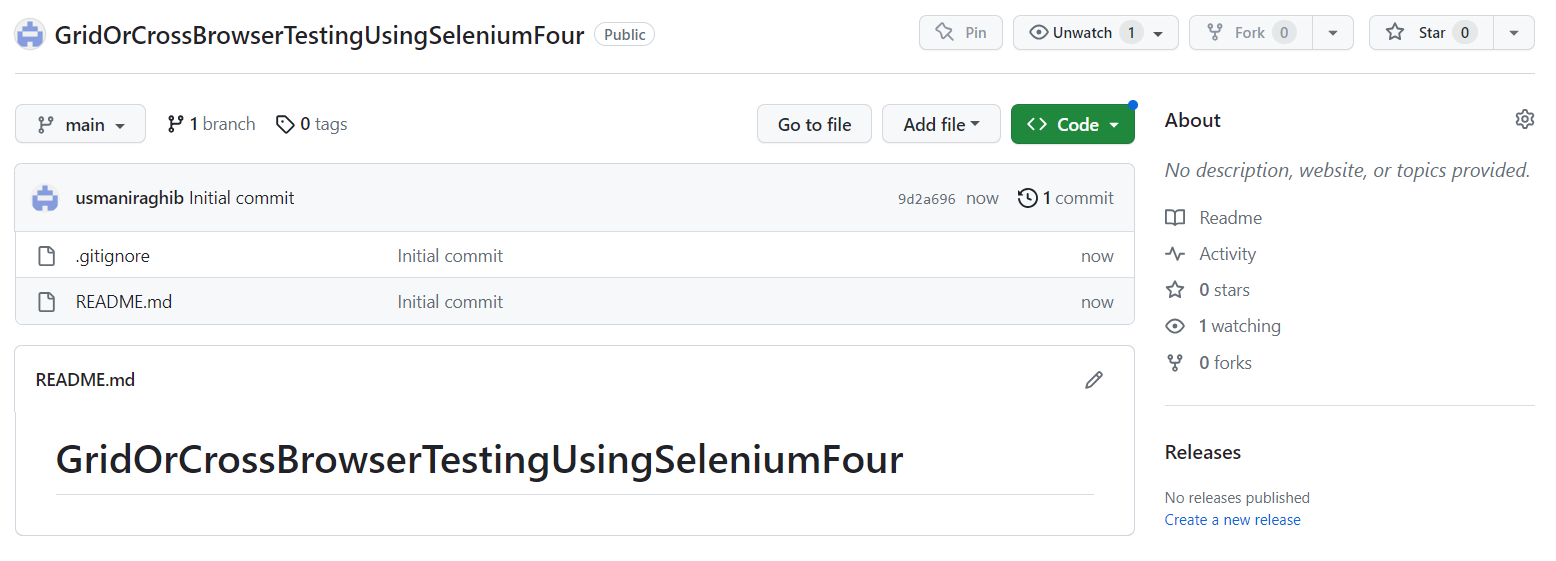
Right click has to be in blank space of the folder. Not on any file or folder.

## Git Repository

Create repository into GitHub.

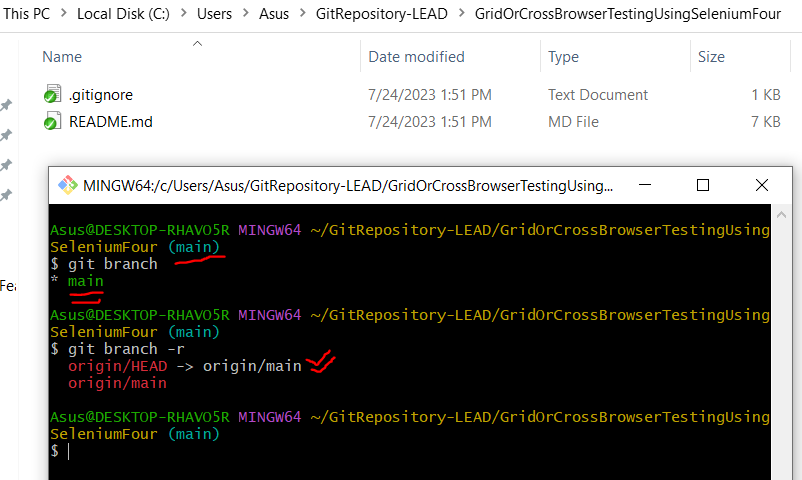
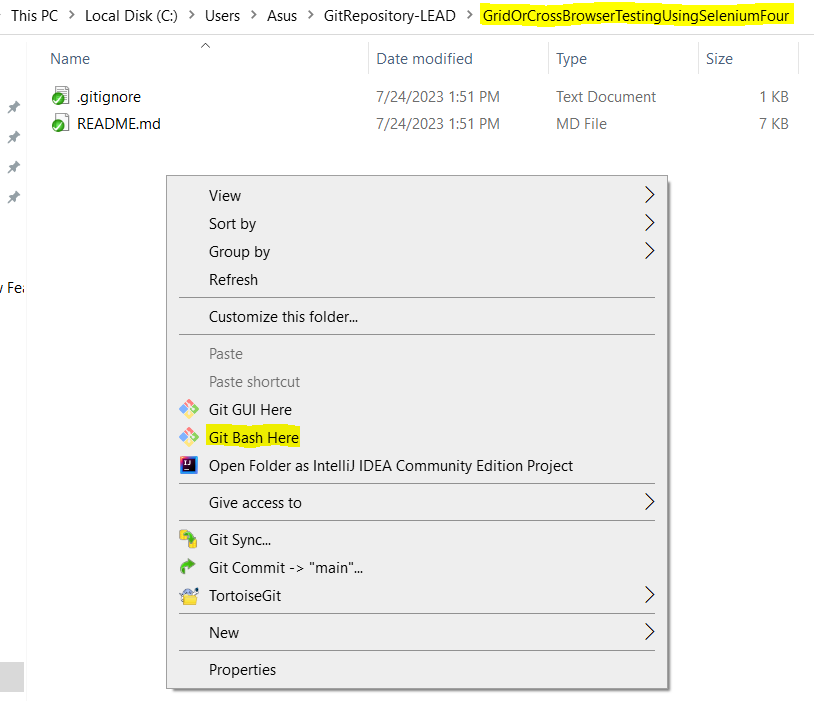






## Git Branch

### See Current Branch - Preferably use your Git Bash



git branch

git branch -r

### Synchronize - Online Branch to Local

Create a branch on web

A screenshot of a computer

Description automatically generated A screenshot of a chat

Description automatically generated A screenshot of a computer

Description automatically generated

Now in your local repo folder “Git bash here”

Magically you don’t see that newly created branch as remote branch also.

Now run command “**git pull --prune**”

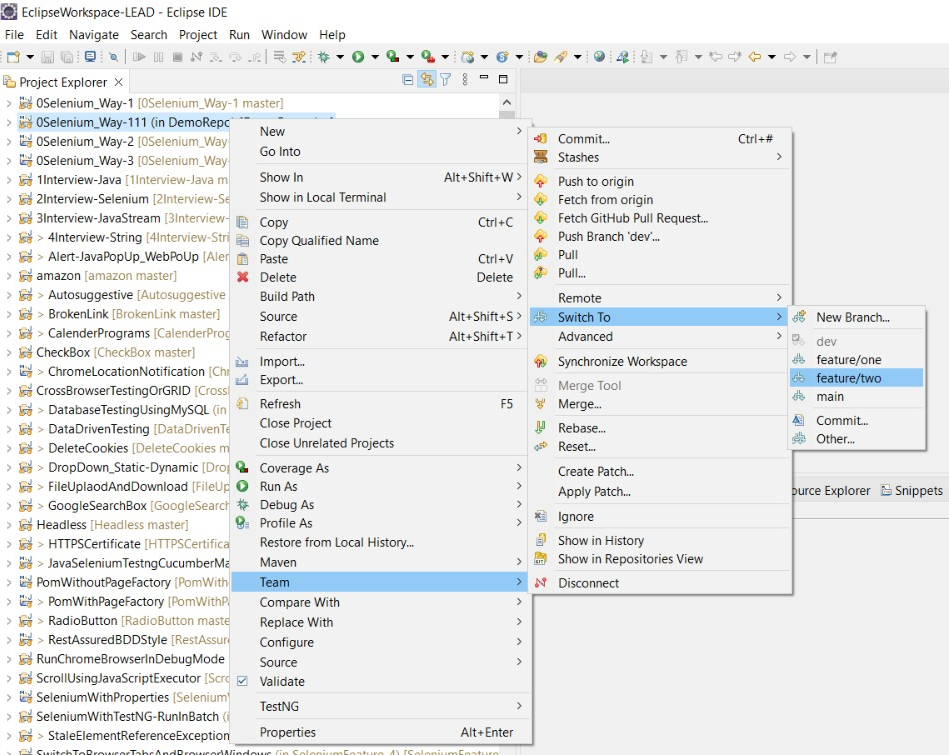
Check remote branch now, it’s synched

A computer screen shot of a program

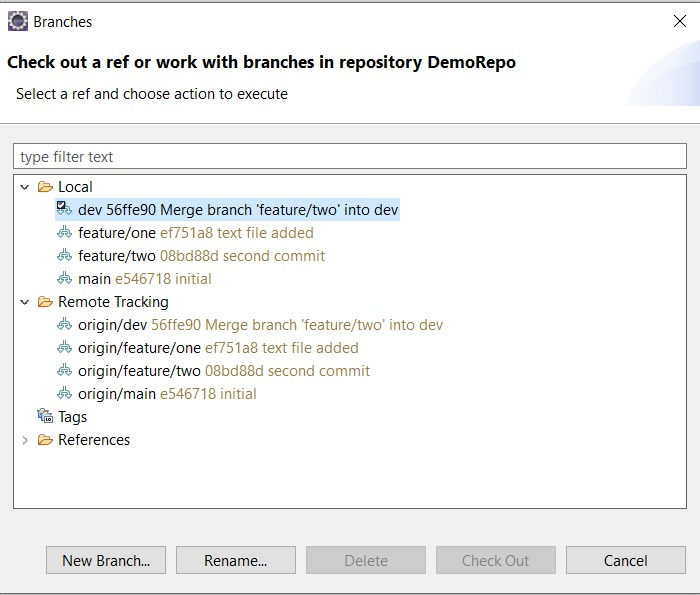
Description automatically generated

### Change Branch – Preferably use your IDE

#### Eclipse

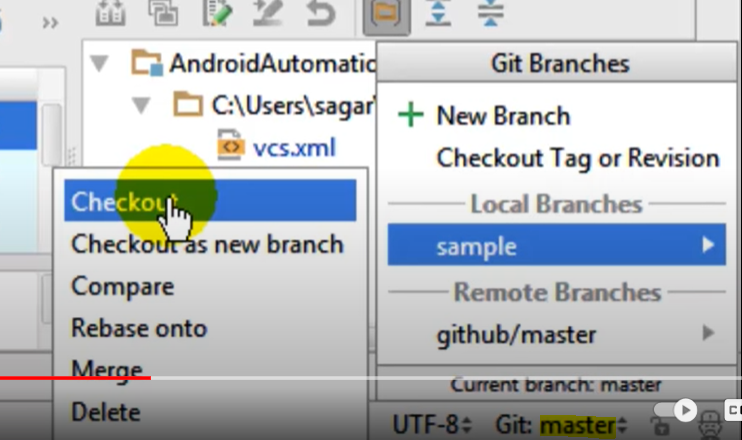


If branch not coming then go to **Team à Switch To à Others** and double click on branch to map it local



#### IntellIJ

Switch from “**master**” branch to “**sample**” branch. Click on Checkout option.



#### Visual Studio

Simply double click on branch name under local

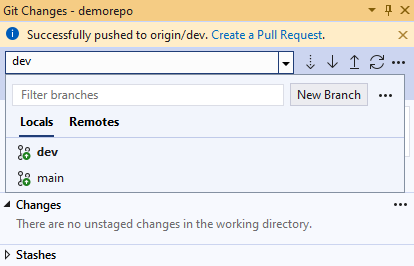
If not under local go to remote and double click on it

Then come to local and double click to checkout or change to that branch

A screenshot of a computer

Description automatically generatedA screenshot of a computer

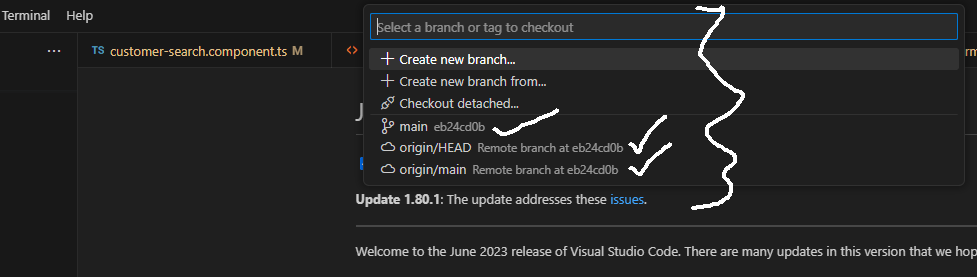
Description automatically generated



#### VS Code

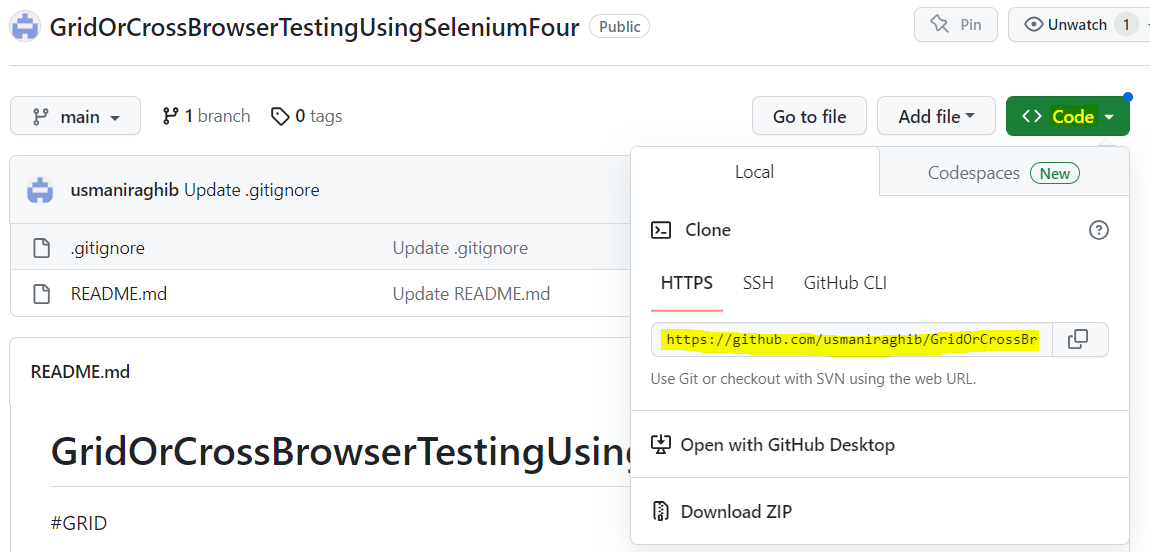
Click on branch below and select the branch shown to change to that particular branch.

A screenshot of a computer

Description automatically generated

## Git Clone – Preferably use TortoiseGit

Copy repository URL from online repository e.g., bitbucket, github



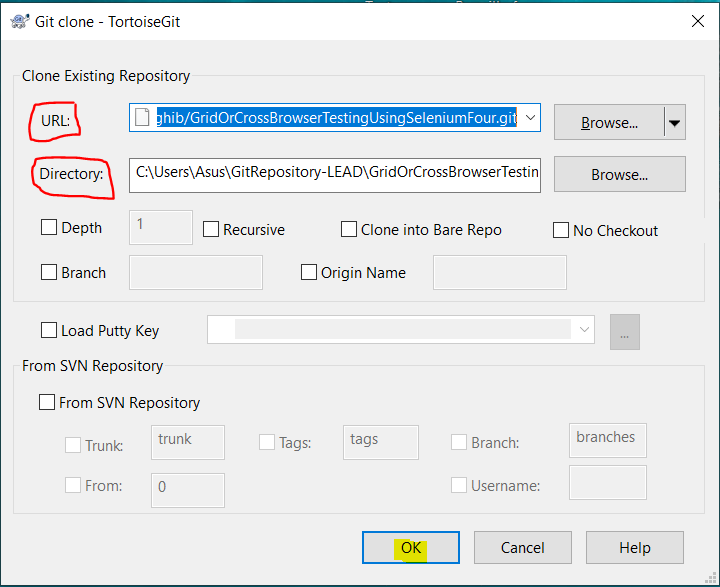
Right click on blank space inside the folder path where you want to clone the repository or download the code.

Click on “Git Clone”

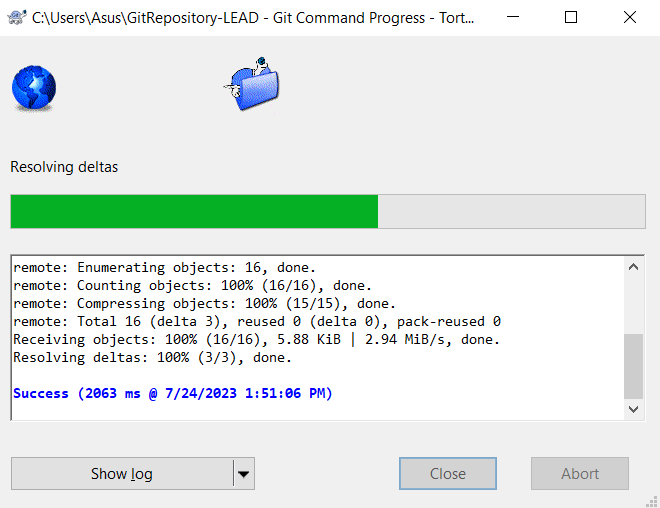
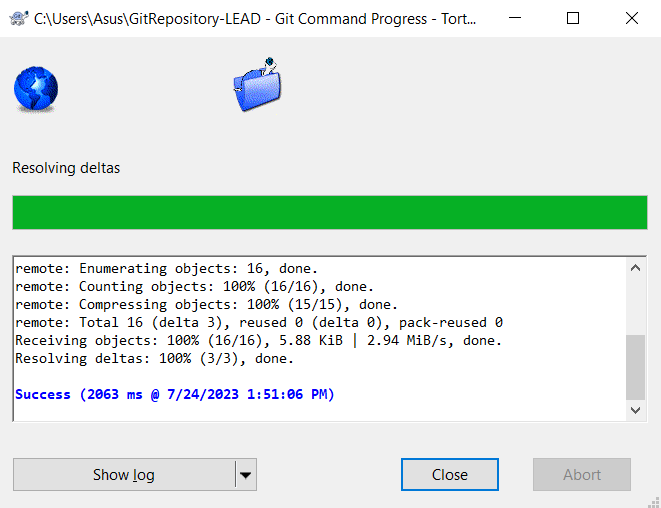
A screenshot of a computer

Description automatically generated

Paste the repository URL and mention the path. Click OK.



It downloads the code.

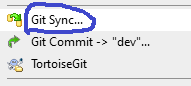
 



## Git Pull – Preferably use TortoiseGit

Go to “**Git Sync…**” and “**Pull**”

**Pull = Fetch + Merge**



A screenshot of a computer

Description automatically generated

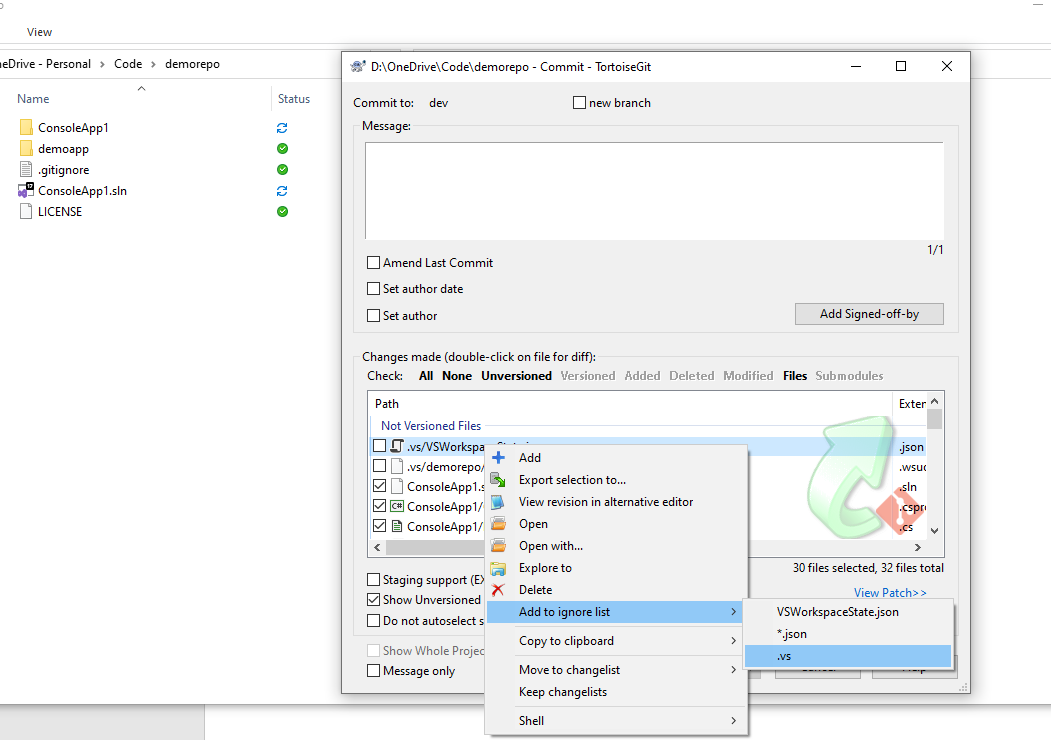
## Git Commit – Preferably use TortoiseGit

A screenshot of a computer

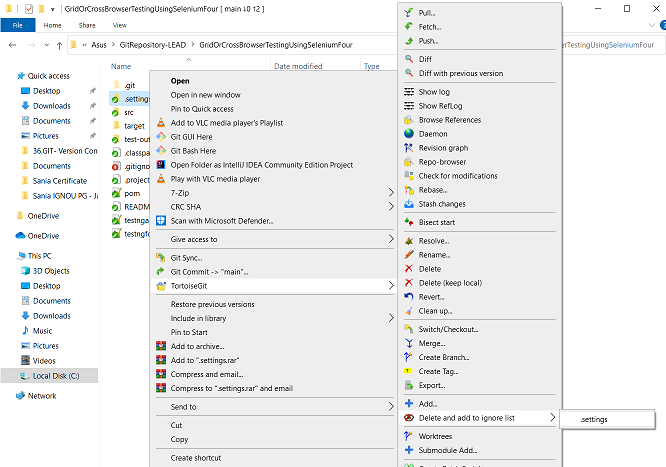
Description automatically generated

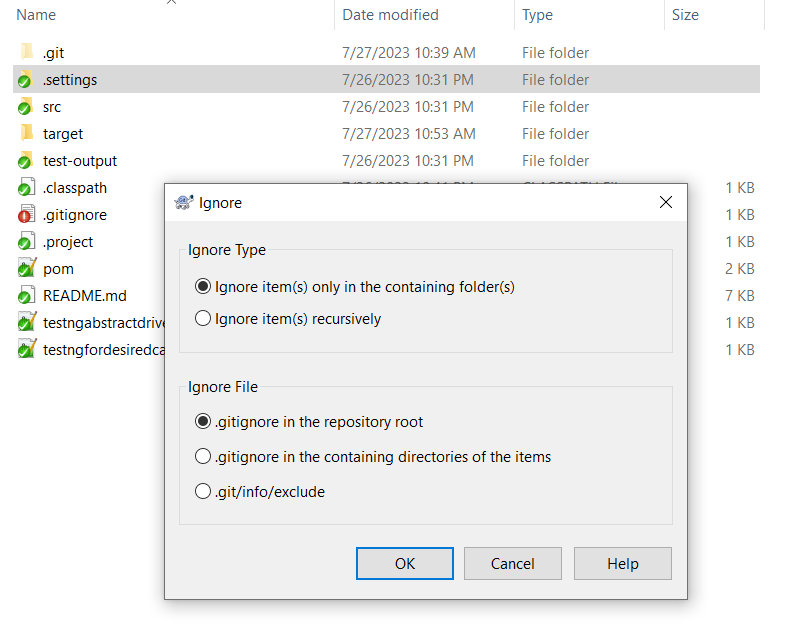
Add unnecessary files to ignore list

**WAY-1**



**WAY-2**





Then select the files and add a message

A screenshot of a computer

Description automatically generated

Click on commit

Don’t select any other type of commit A screenshot of a computer

Description automatically generated

## Git Push – Preferably use TortoiseGit

Go to “**Git Sync…**” and “**Pull**” and then “**Push**”

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Git Merge – Preferably use Git Bash

Update source and destination branch at your local

Change your repo to destination branch

Open “Git Bash Here”

Run command “git merge <<source branch>>”

### Conflict Resolve – Preferably use TortoiseGit

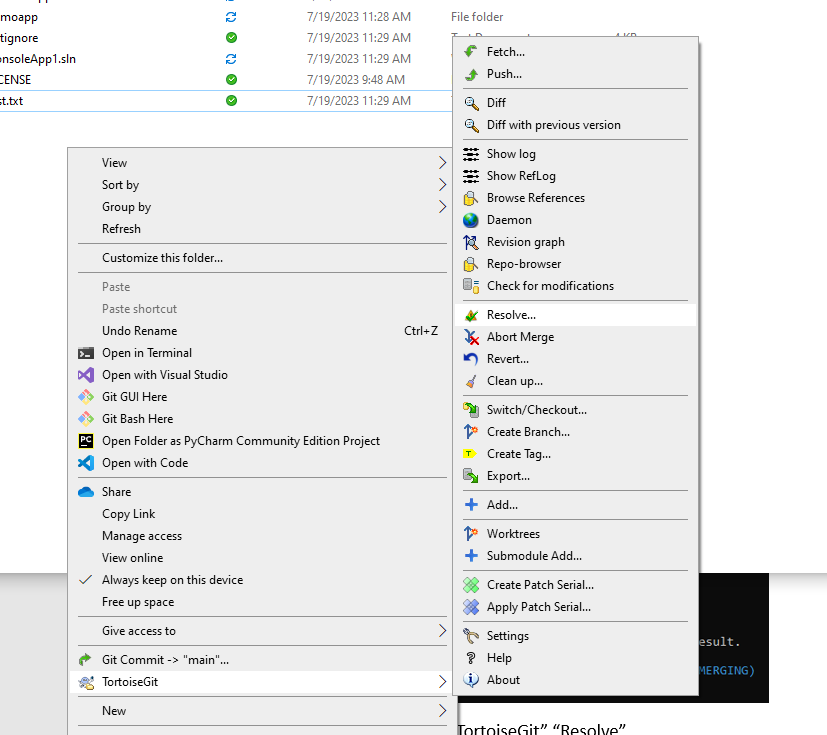
A blue rectangle with white text

Description automatically generated

A screen shot of a computer code

Description automatically generated

Open right click menu “TortoiseGit” “Resolve”

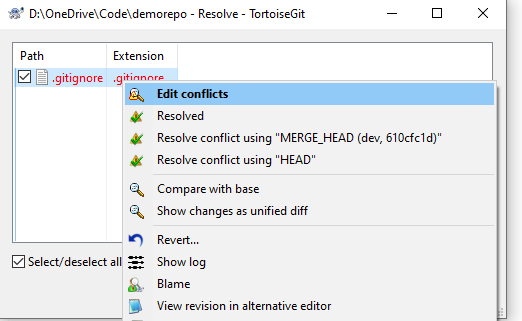


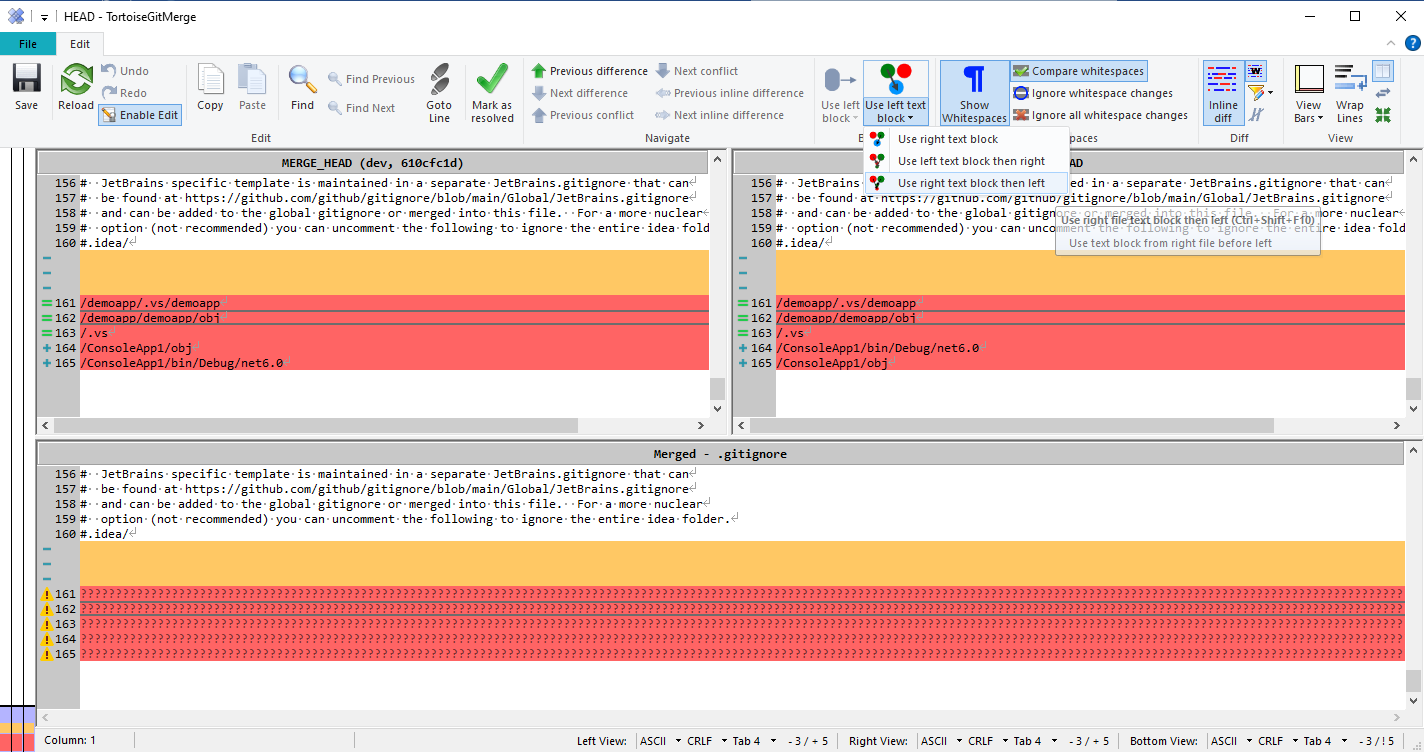
List of files appear with conflict

A screenshot of a computer error

Description automatically generated

Edit Conflict





Click on **Save** and **Mark as resolved** at the end

A screenshot of a computer

Description automatically generated

## Daily

* Commit all your logical changes in local working feature branch. Don’t push
* Change branch to dev or main or sprint branch where all developers are pushing code
* Pull dev or main or sprint branch
* Change branch to local working feature branch
* Merge dev or main or sprint branch to local working feature branch