

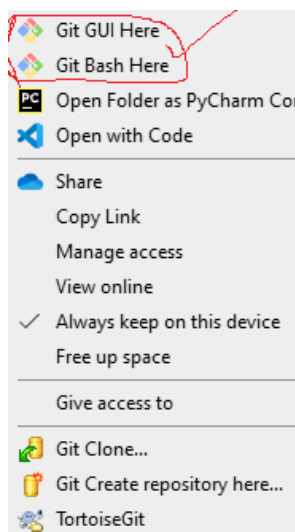
GIT

1. Prerequisite
 - a. Git
 - b. TortoiseGit
 - c. IDE
2. Git Repository
3. Git Branch
 - a. See Current Branch – Preferably use your Git Bash
 - b. Synchronize – Online Branch to Local
 - c. Change Branch – Preferably use your IDE
 - i. Eclipse
 - ii. IntelliJ
 - iii. Visual Studio
 - iv. VS Code
4. Git Clone – Preferably use TortoiseGit
5. Git Pull – Preferably use TortoiseGit
6. Git Commit – Preferably use TortoiseGit
7. Git Push – Preferably use TortoiseGit
8. Git Merge – Preferably use Git Bash
 - a. Conflict Resolve – Preferably use TortoiseGit
9. Daily

Prerequisite

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

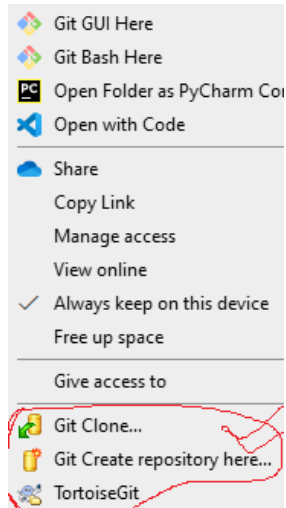
On installing git bash these two options will appear in right click menu. Right click has to be in blank space of the folder. Not on any file or folder.



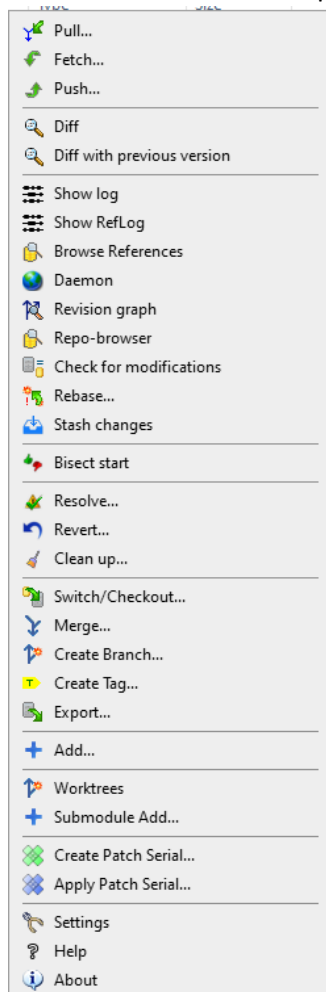
2. TortoiseGit

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

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



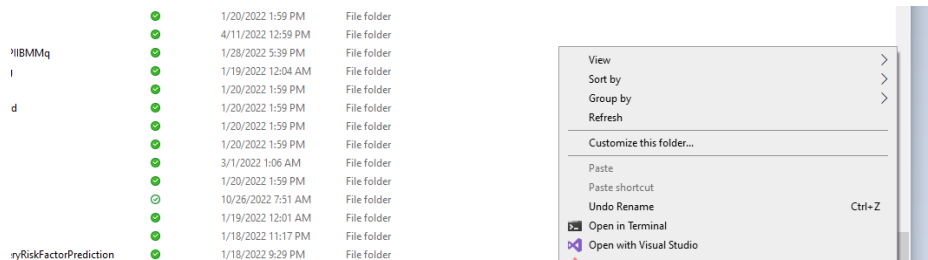
Under TortoiseGit menu options



3. 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.

usmaniraghib

Overview Repositories 50 Projects Packages Stars

Find a repository...

Type Language Sort New

Autosuggestive Public

Java Updated 5 days ago

Star

Alert-JavaPopUp_WebPoUp Public

Java Updated 5 days ago

Star

DemoRepo Public


Updated last week

Star

WebTableWithPagination Public

Java Updated last week

Star



usmaniraghib

Edit profile

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *

usmaniraghib

Repository name *

GridOrCrossBrowserTestin

GridOrCrossBrowserTestingUsingSeleniumFour is available.

Great repository names are short and memorable. Need inspiration? How about fluffy-octo-carnival ?

Description (optional)

- ☒

Public

Anyone on the internet can see this repository. You choose who can commit.

☐

Private

You choose who can see and commit to this repository.
- #### Initialize this repository with:
- ☒ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)
- #### Add .gitignore
- .gitignore template: Maven
- Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)
- #### Choose a license
- License: None
- A license tells others what they can and can't do with your code. [Learn more about licenses.](#)
- This will set `main` as the default branch. Change the default name in your [settings](#).
- i** You are creating a public repository in your personal account.
- Create repository
- GridOrCrossBrowserTestingUsingSeleniumFour Public

Pin Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags

Go to file Add file Code

About

usmaniraghib Initial commit 9d2a696 now 1 commit

.gitignore Initial commit now

README.md Initial commit now

README.md

GridOrCrossBrowserTestingUsingSeleniumFour

No description, website, or topics provided.

Readme

Activity

0 stars

1 watching

0 forks

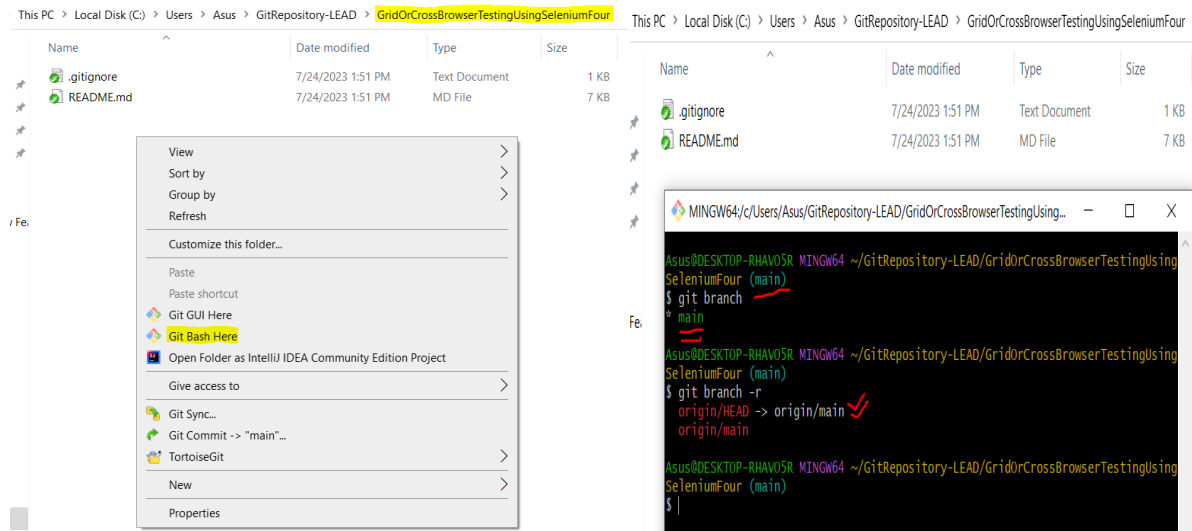
Releases

No releases published

Create a new release

Git Branch

1. See Current Branch- Preferably use your Git Bash

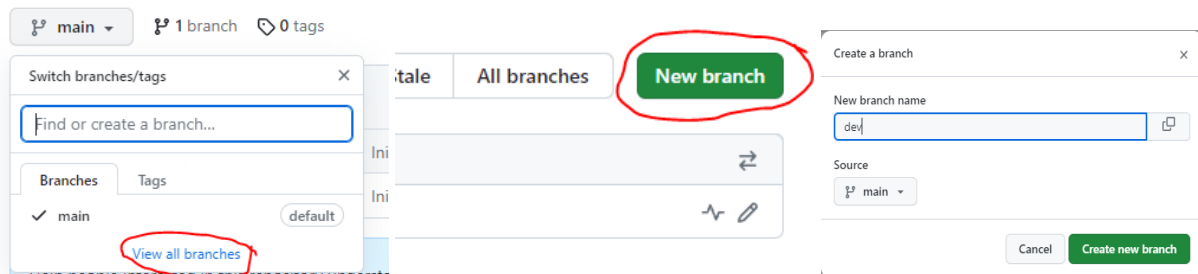


git branch

git branch -r

2. Synchronize- Online Branch to Local

Create a branch on web

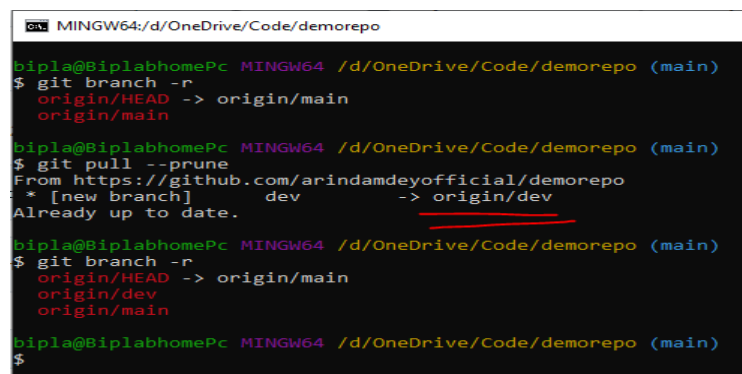


Now in your local repo folder "Git bash here"

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

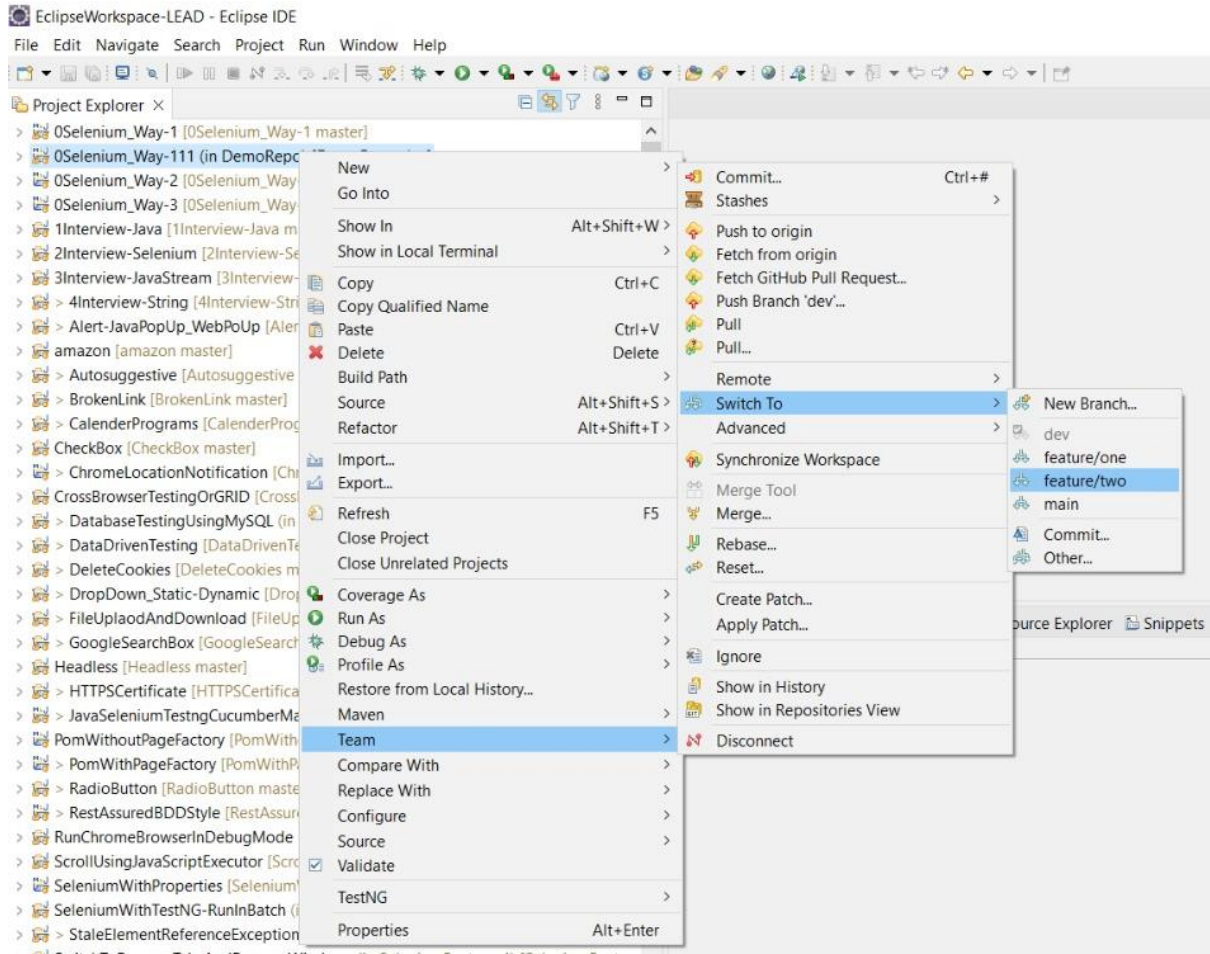
Now run command "git pull --prune"

Check remote branch now, it's synced

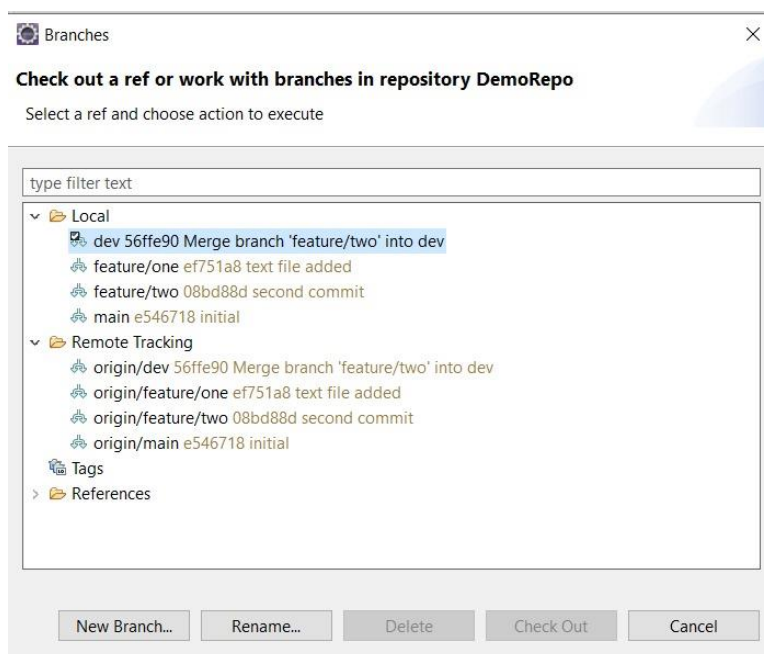


3. Change Branch – Preferably use your IDE

1. Eclipse

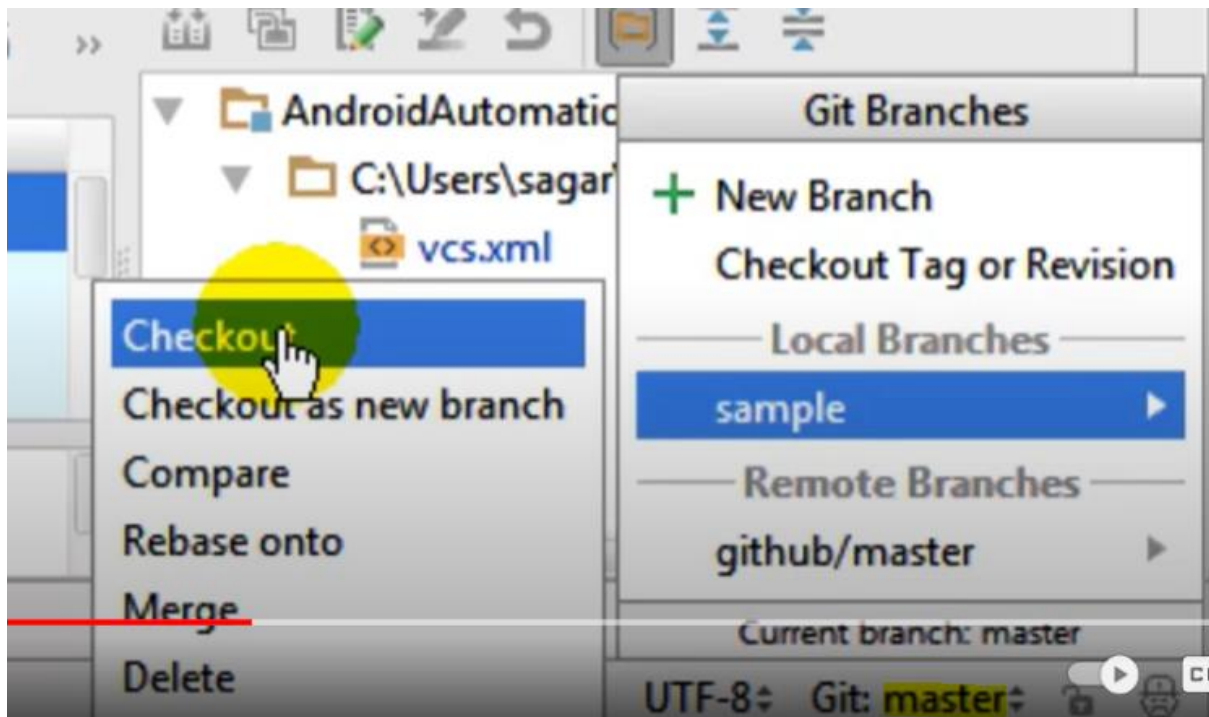


If branch not coming then go to **Team** → **Switch To** → **Others** and double click on branch to map it local



2. IntelliJ

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

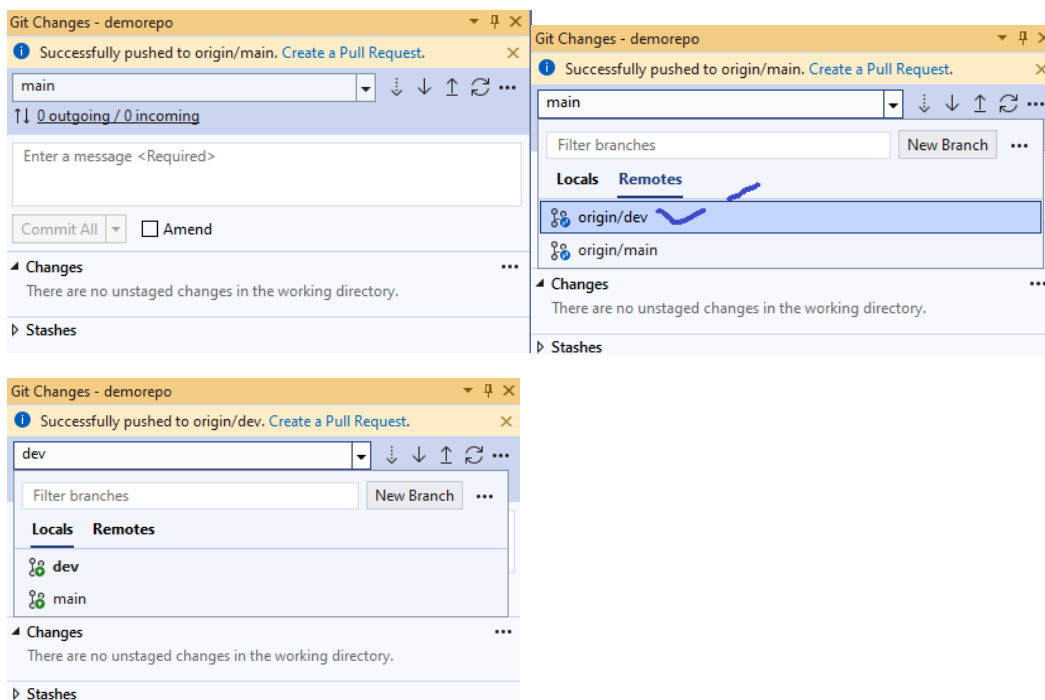


3. Visual Studio

Simply double click on branch name under local

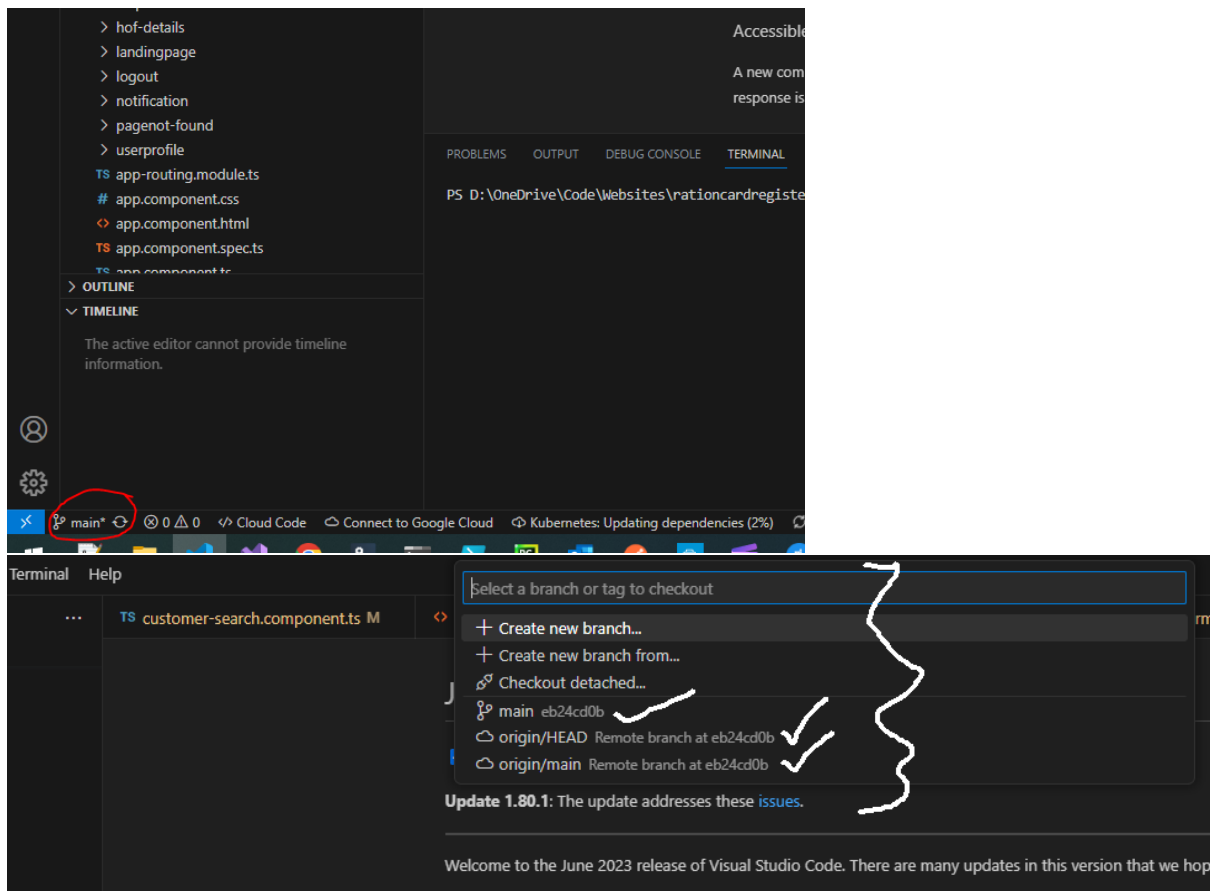
If not under local go to remote and double click on it

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



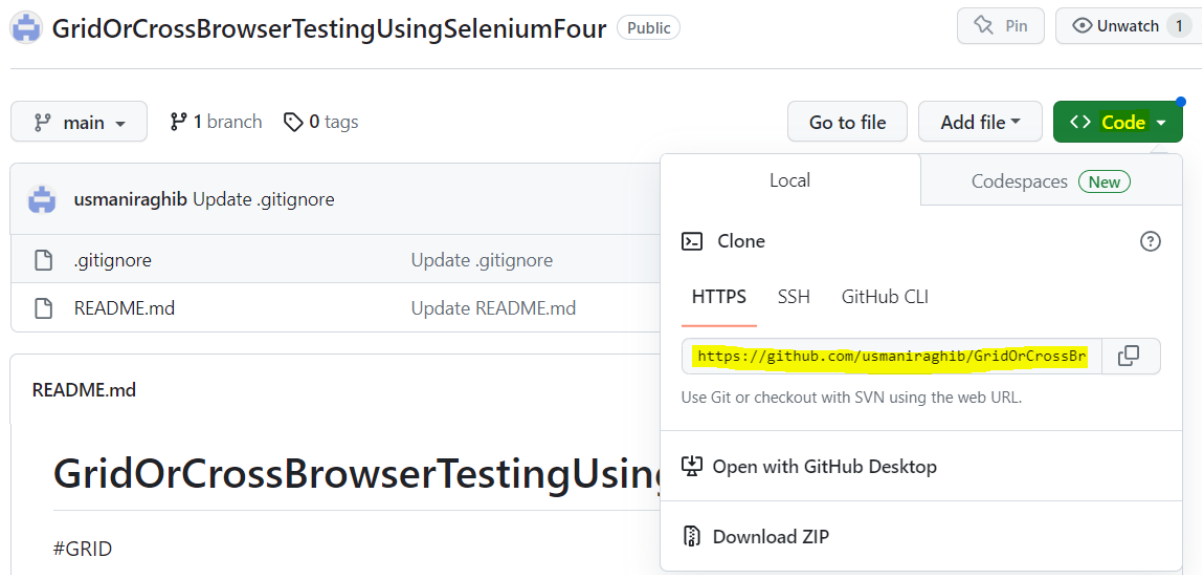
4. VS Code

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



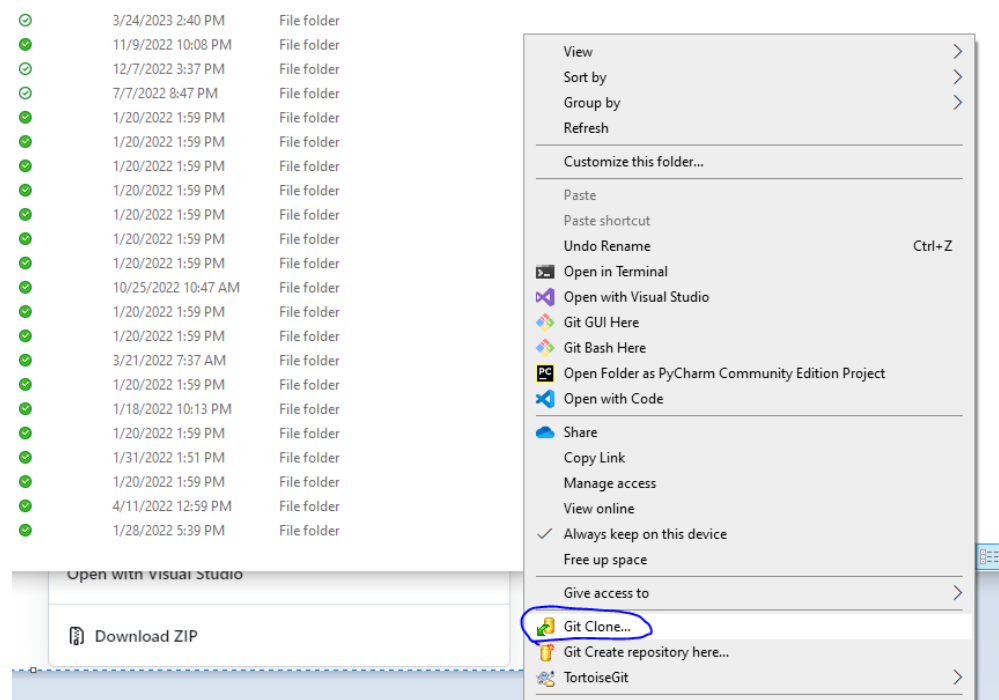
Git Clone – Preferably use TortoiseGit

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

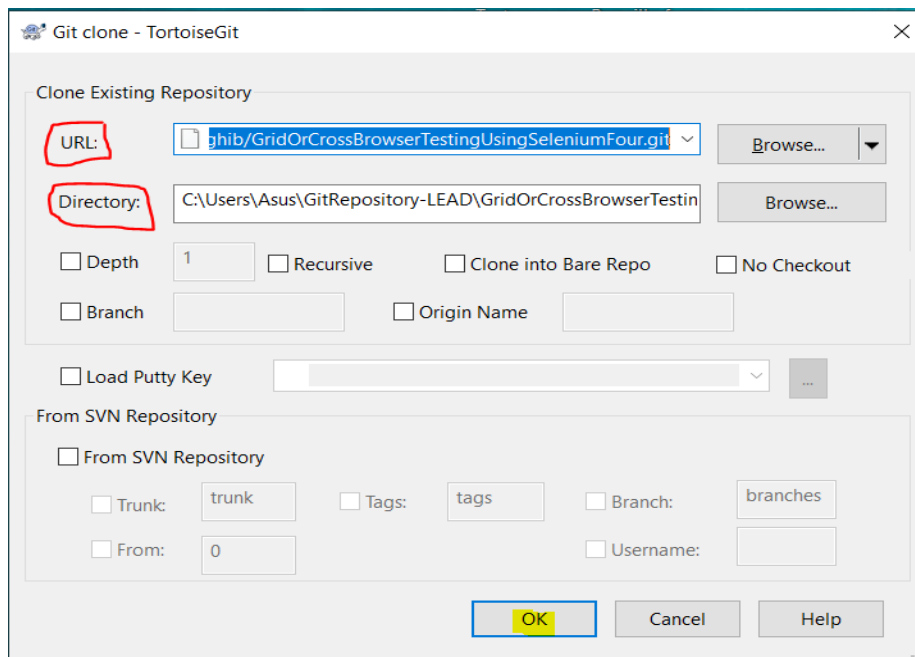


Right click on blank space inside the folder path where you want to clone the repository or download the code.

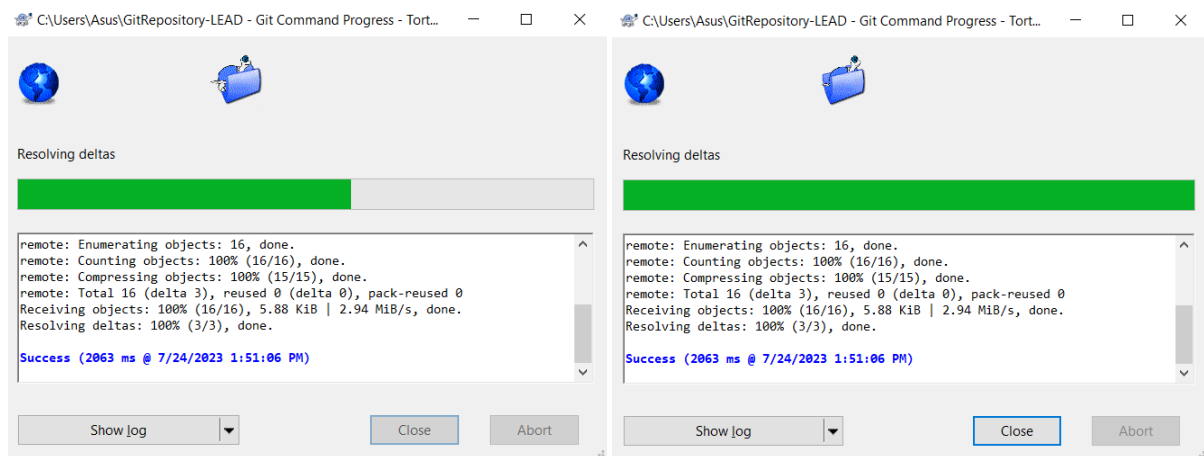
Click on “Git Clone”



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



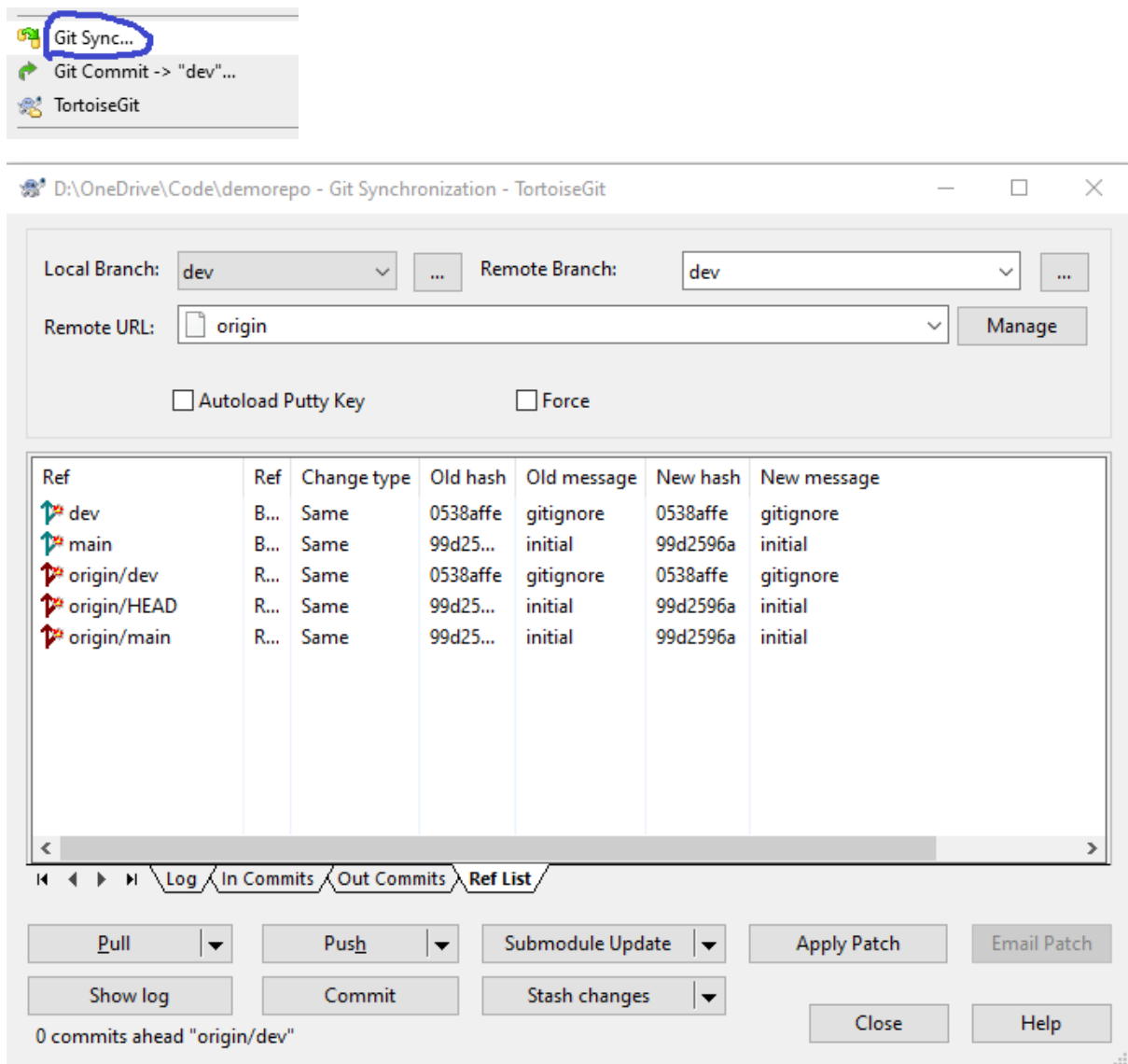
It downloads the code.



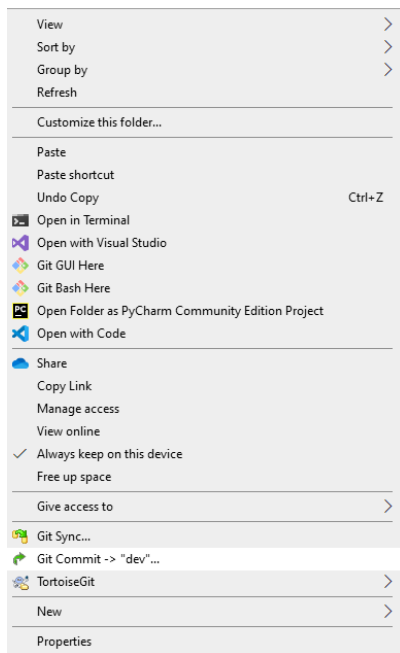
Git Pull – Preferably use TortoiseGit

Go to “Git Sync...” and “Pull”

Pull = Fetch + Merge

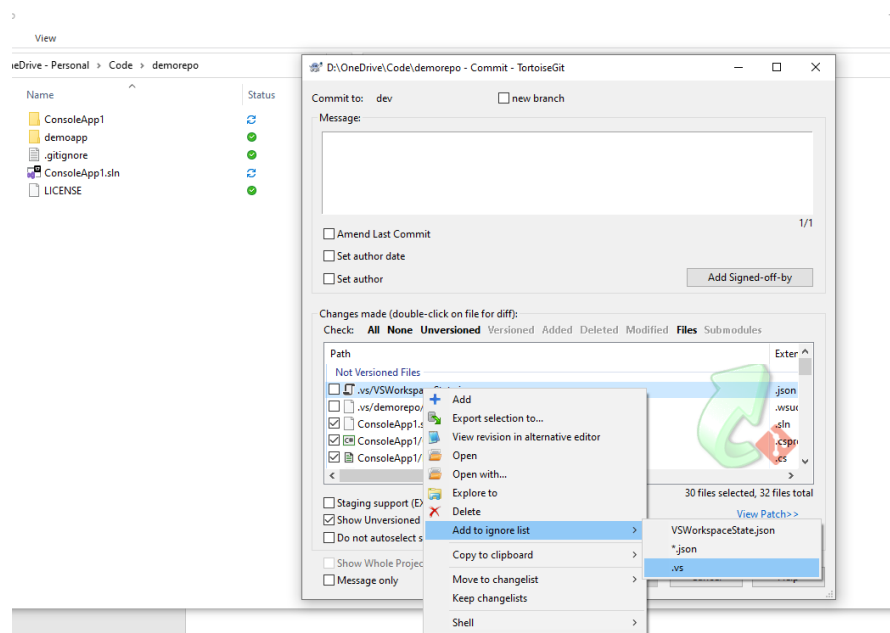


Git Commit – Preferably use TortoiseGit

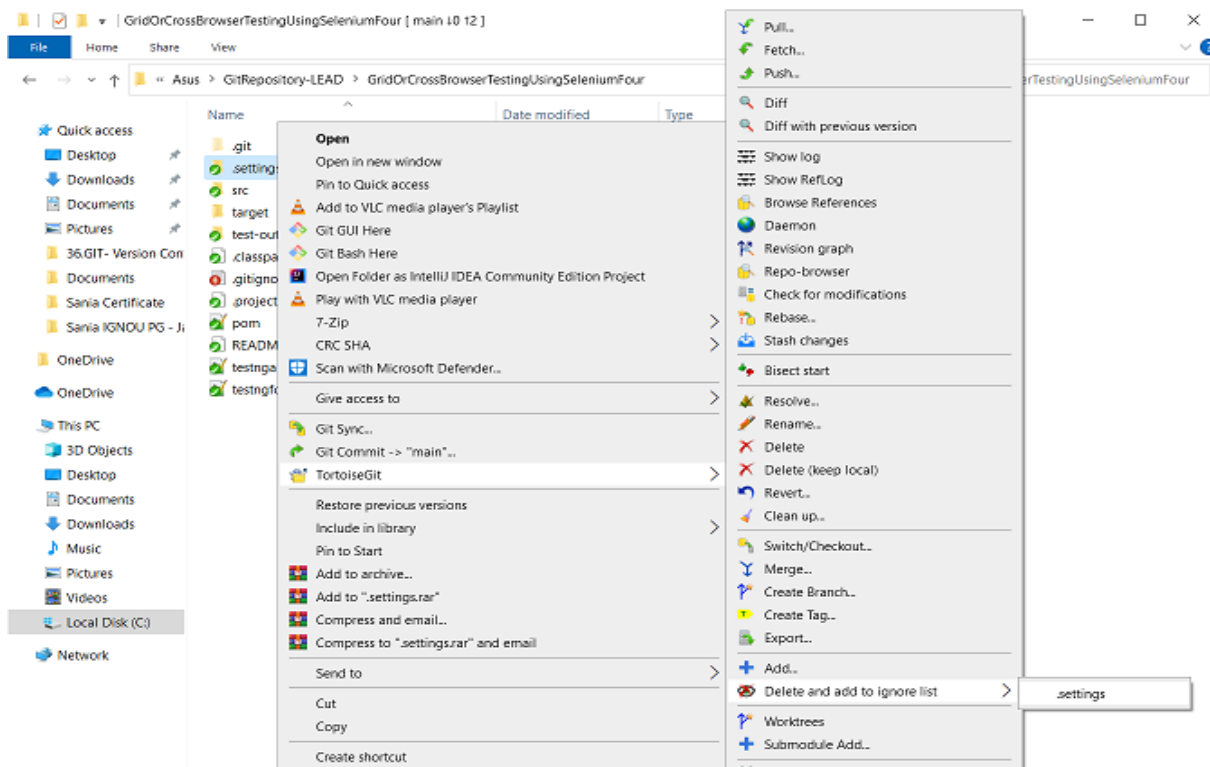


Add unnecessary files to ignore list

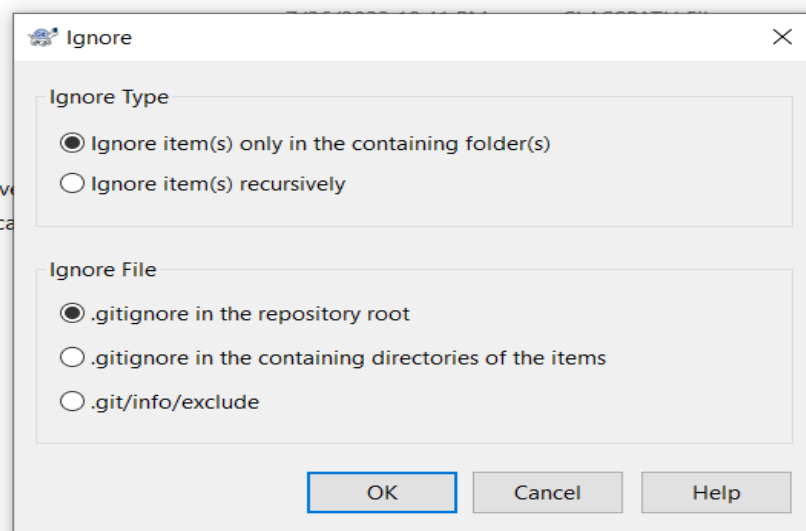
WAY-1



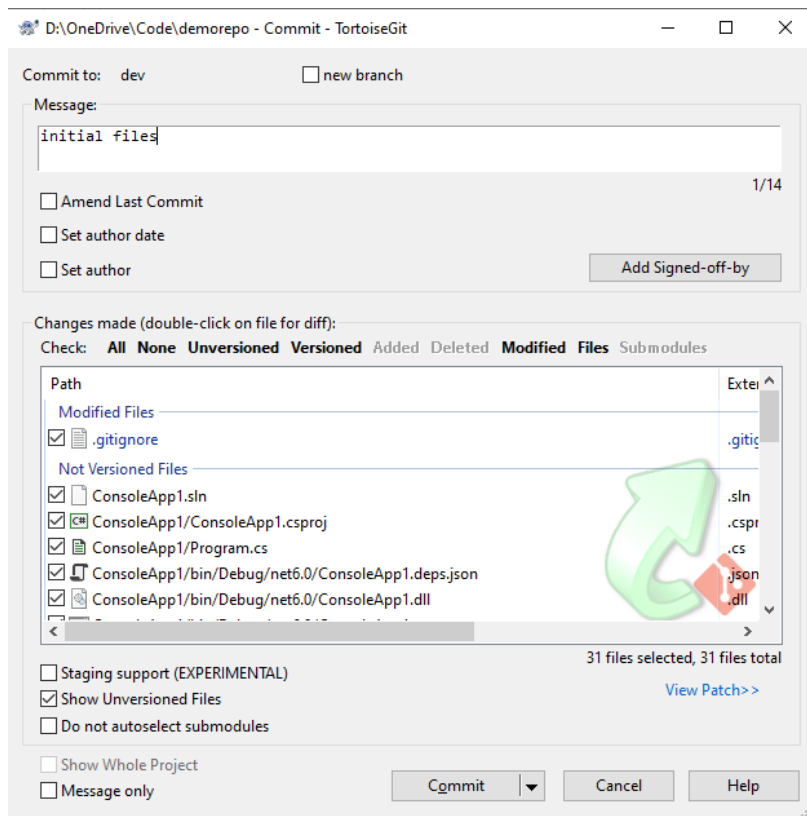
WAY-2



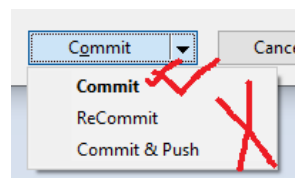
Name	Date modified	Type	Size
.git	7/27/2023 10:39 AM	File folder	
.settings	7/26/2023 10:31 PM	File folder	
src	7/26/2023 10:31 PM	File folder	
target	7/27/2023 10:53 AM	File folder	
test-output	7/26/2023 10:31 PM	File folder	
.classpath	7/26/2023 10:31 PM	File	1 KB
.gitignore	7/26/2023 10:31 PM	File	1 KB
.project	7/26/2023 10:31 PM	File	1 KB
pom	7/26/2023 10:31 PM	File	2 KB
README.md	7/26/2023 10:31 PM	File	7 KB
testngabstractdriver	7/26/2023 10:31 PM	File	1 KB
testngfordesiredcapabilities	7/26/2023 10:31 PM	File	1 KB



Then select the files and add a message



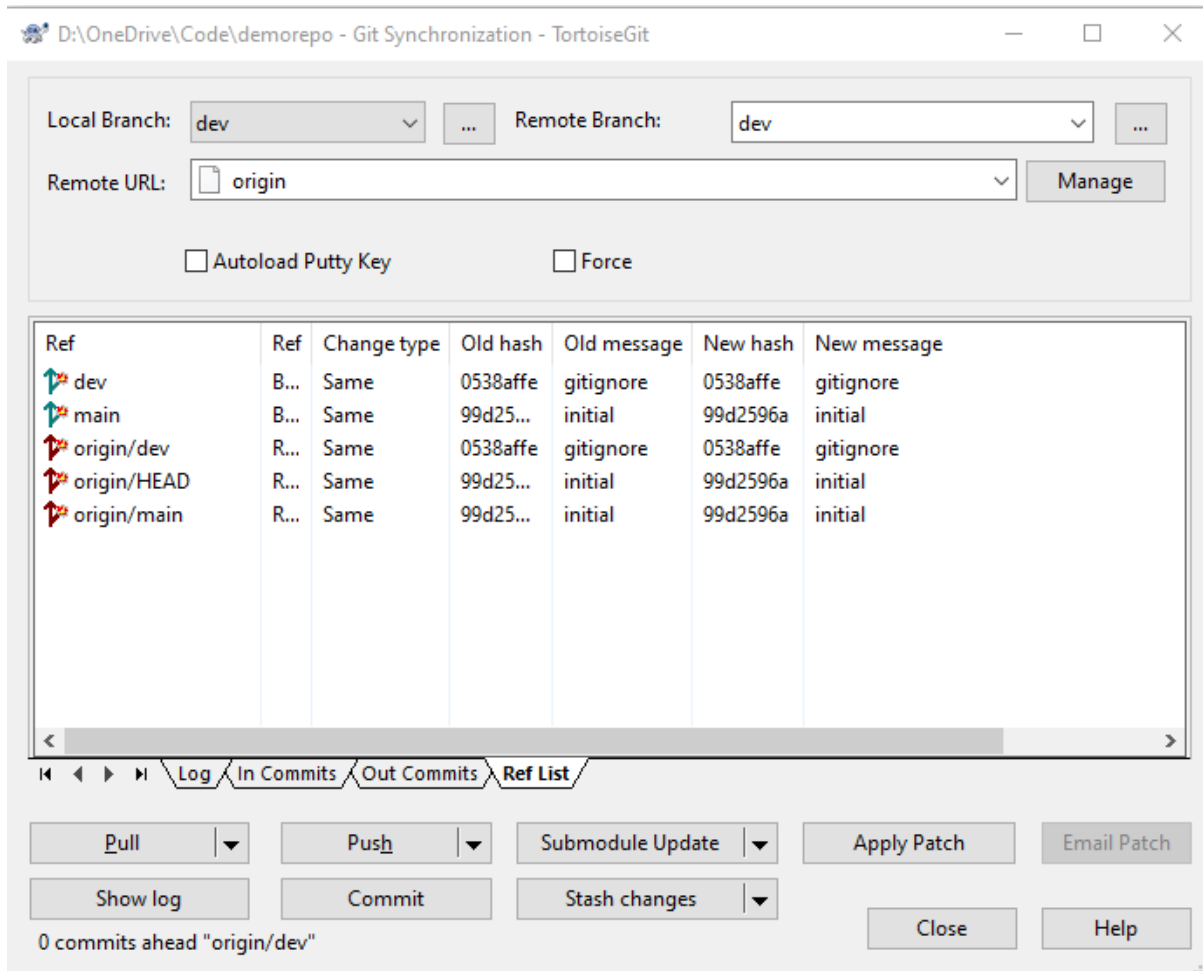
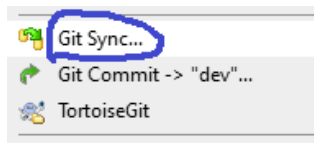
Click on commit



Don't select any other type of commit

Git Push – Preferably use TortoiseGit

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



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

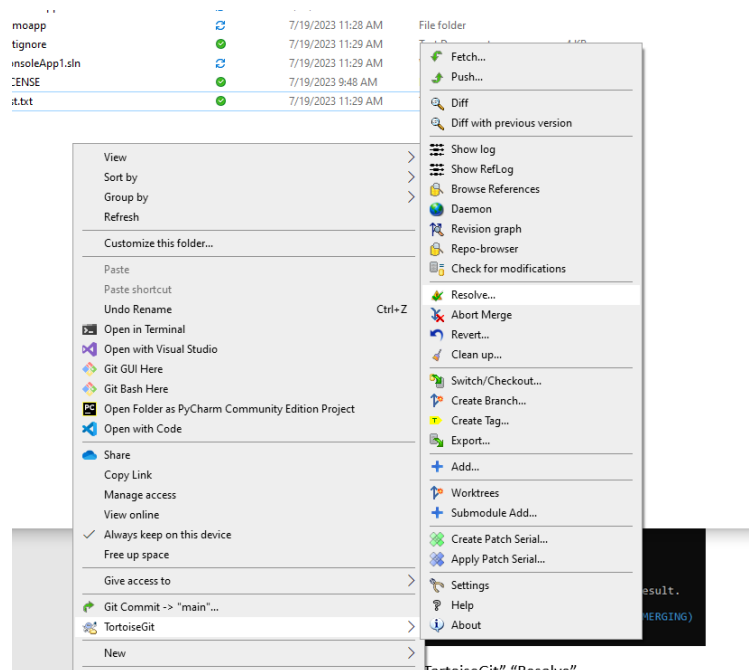


```
MINGW64:/d/OneDrive/Code/demorepo

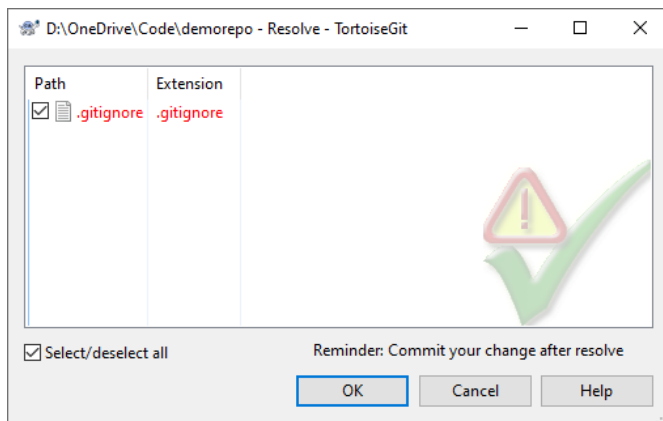
bipla@BiplabhomePc MINGW64 /d/OneDrive/Code/demorepo (main)
$ git merge dev
Auto-merging .gitignore
CONFLICT (content): Merge conflict in .gitignore
Automatic merge failed; fix conflicts and then commit the result.

bipla@BiplabhomePc MINGW64 /d/OneDrive/Code/demorepo (main|MERGING)
$
```

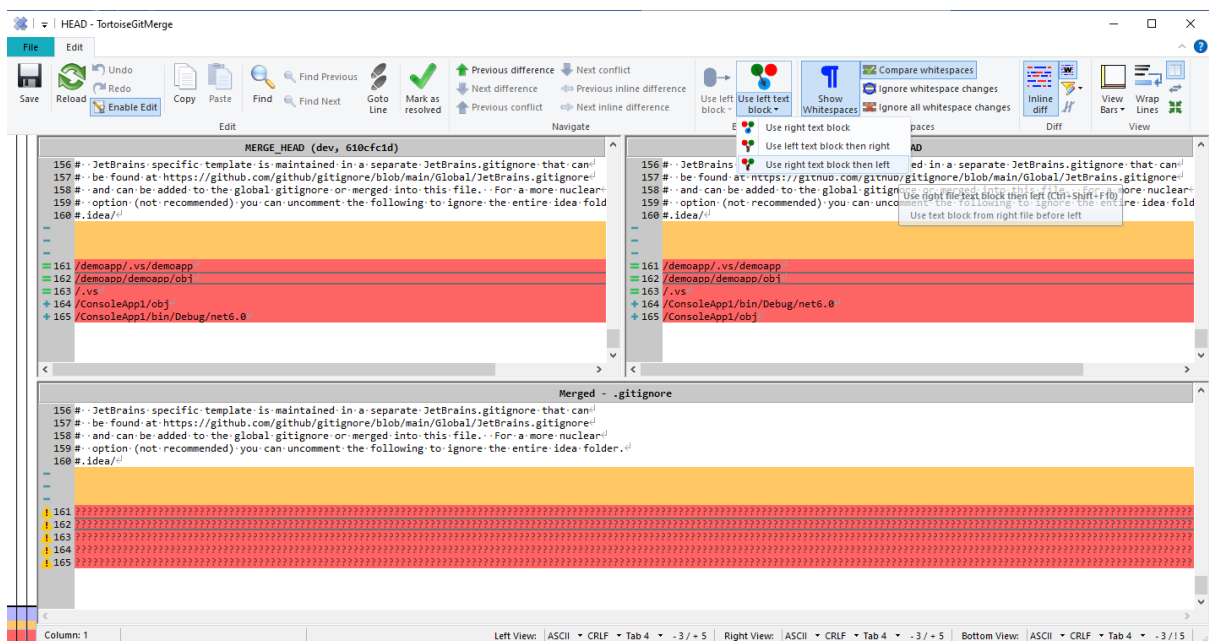
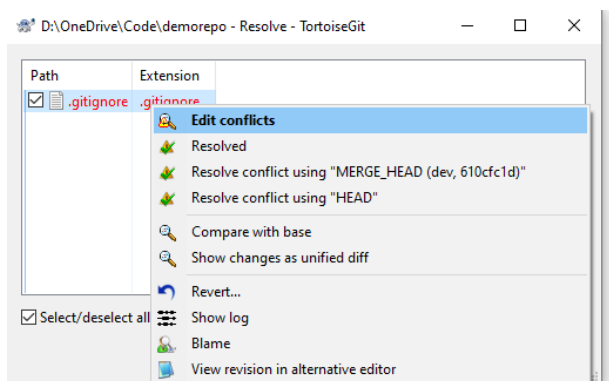
Open right click menu “TortoiseGit” “Resolve”



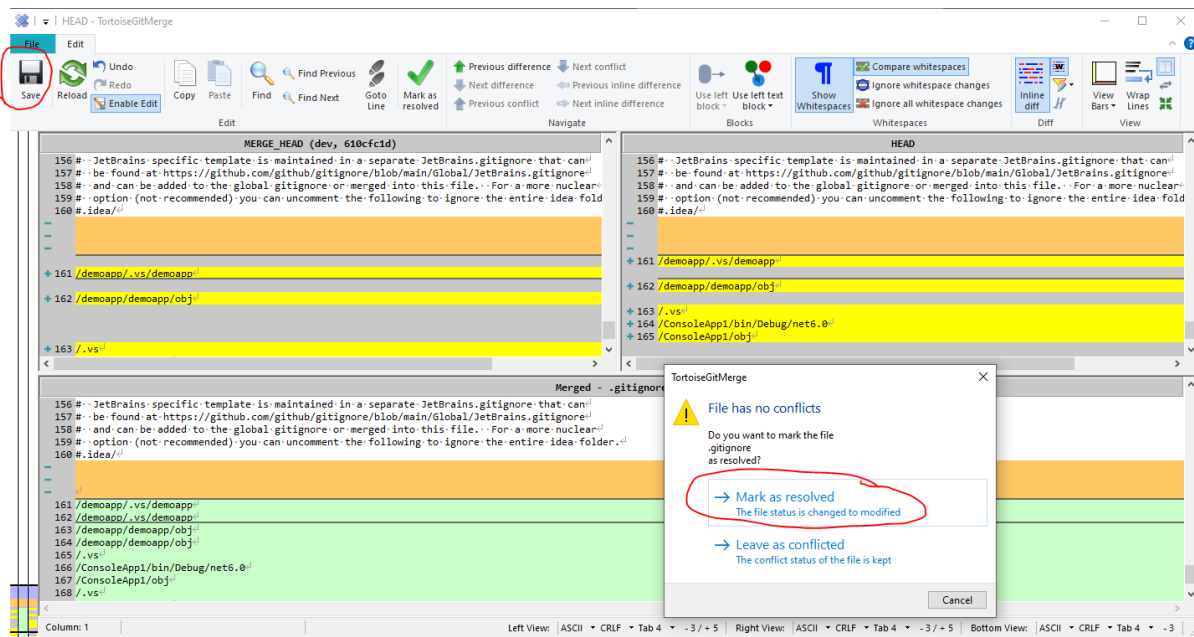
List of files appear with conflict



Edit Conflict



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



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