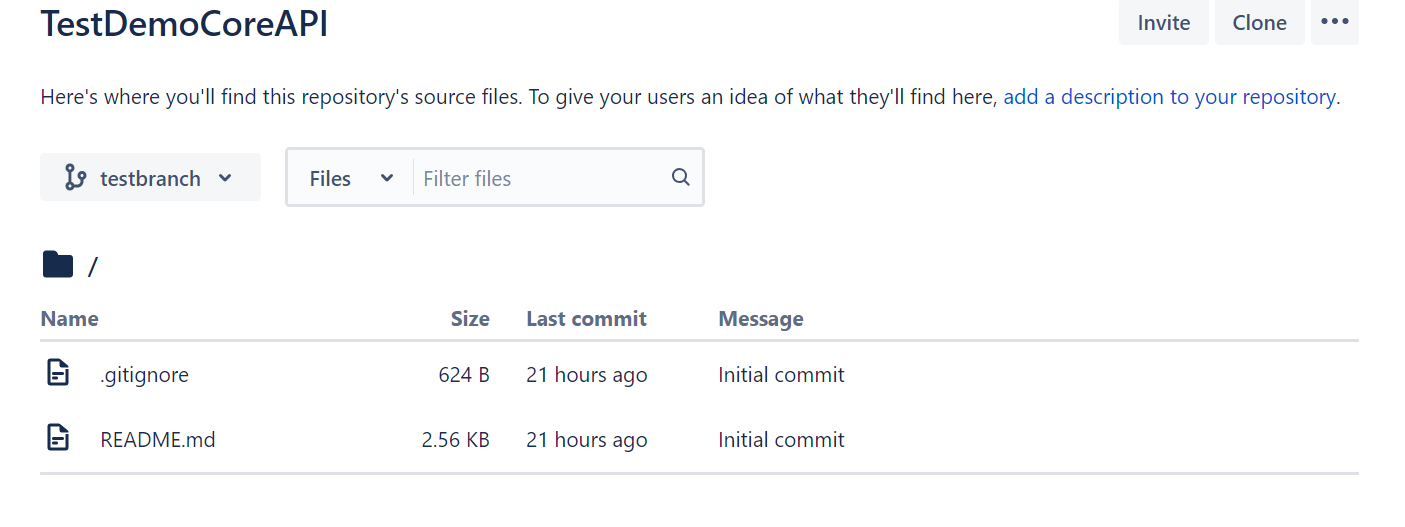
**Revert the commit using Git and Bitbucket**

Created new branch from master branch. Eg: testbranch



**git clone**

Usage: git clone [url]

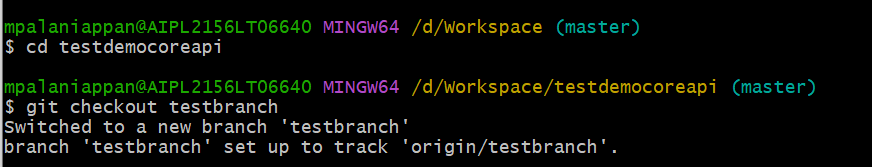
This command is used to obtain a repository from an existing URL.

**git checkout**

Syntax: git checkout [branch name]

This command is used to switch from one branch to another.

Example:

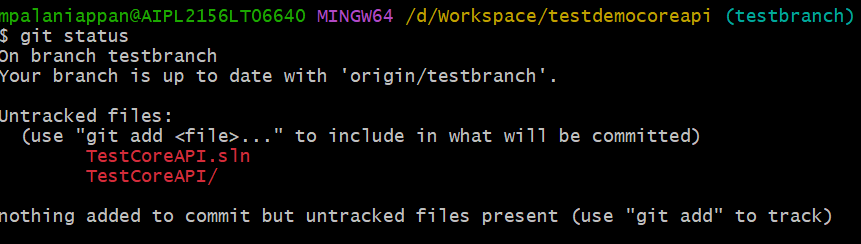


Copy the project and paste in the local repo path. check the status

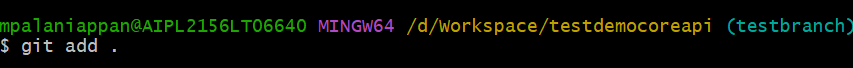
**git status**

Syntax: git status

This command lists all the files that have to be committed.



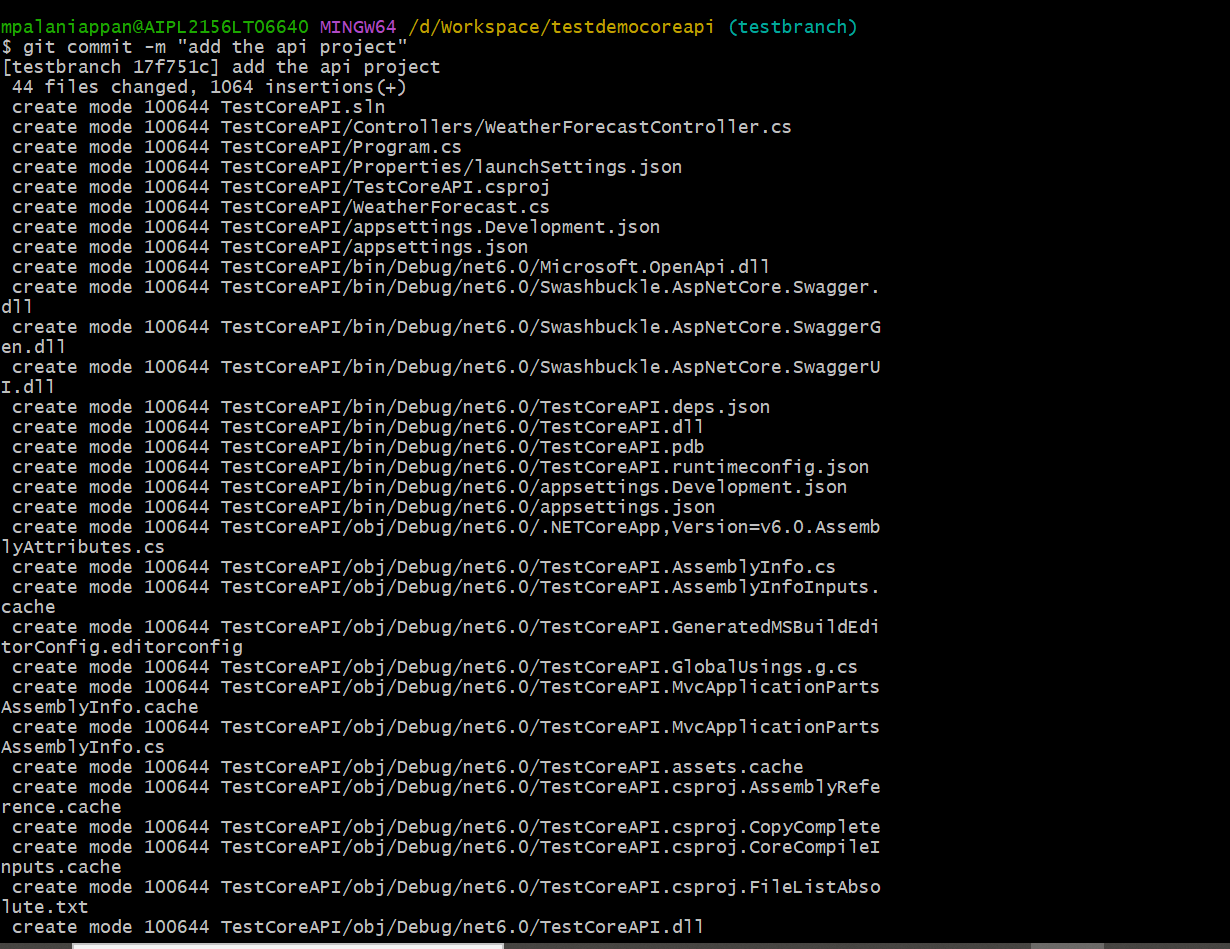
Add all the file changes to stage.



**git commit**

Syntax: git commit -m “[ Type in the commit message]”

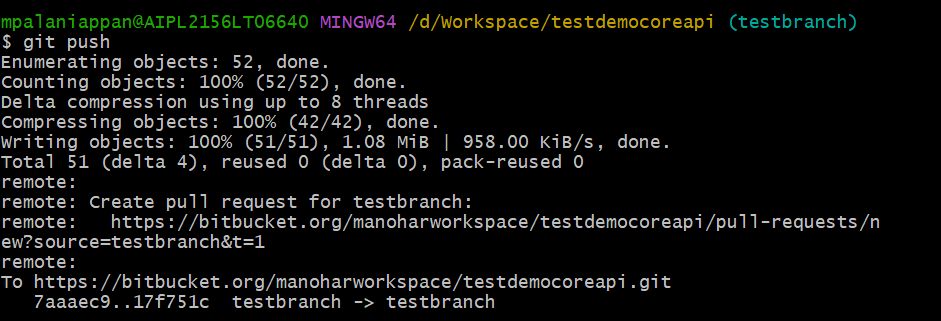
This command records or snapshots the file permanently in the version history.



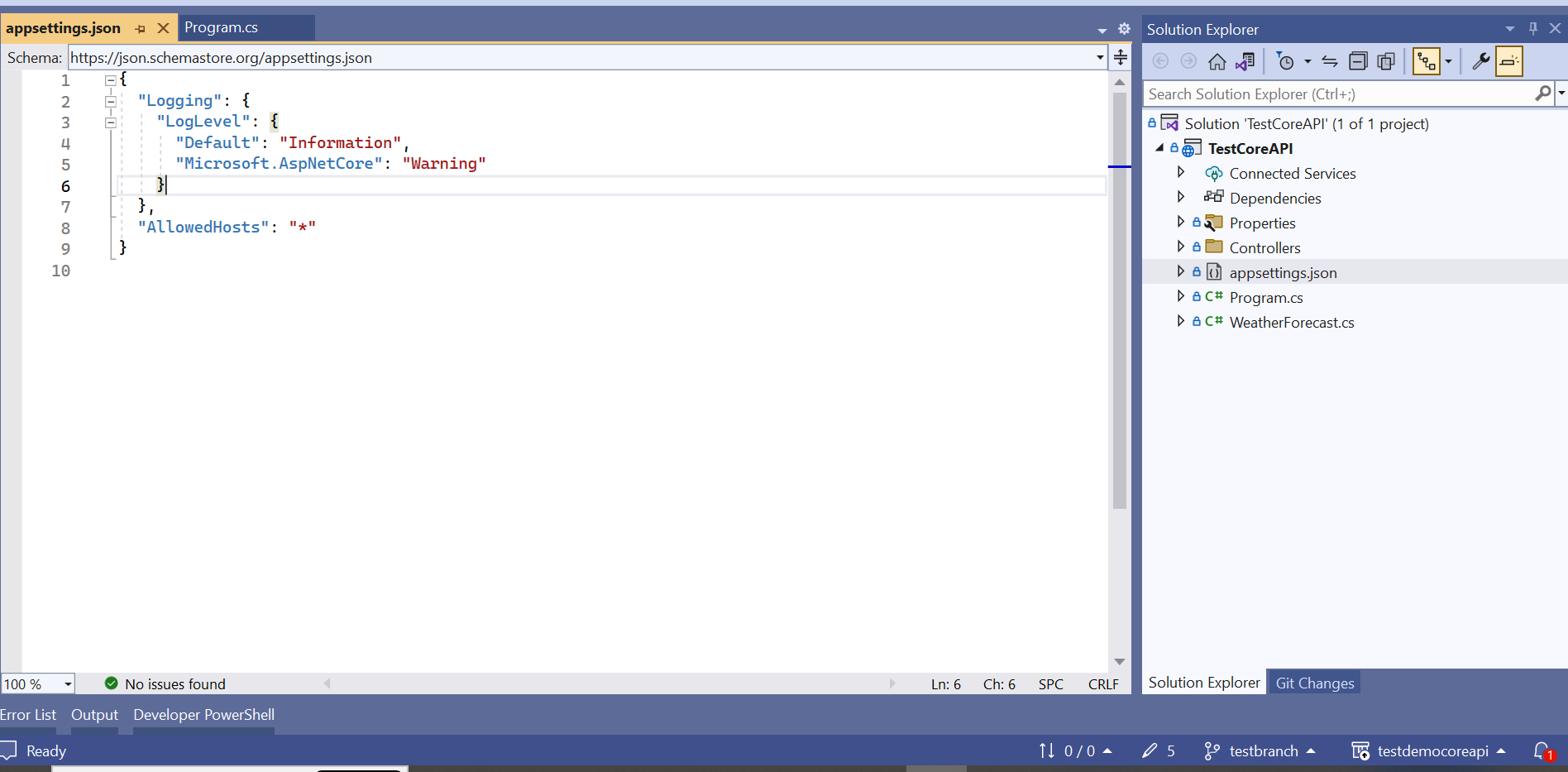
**Git push**

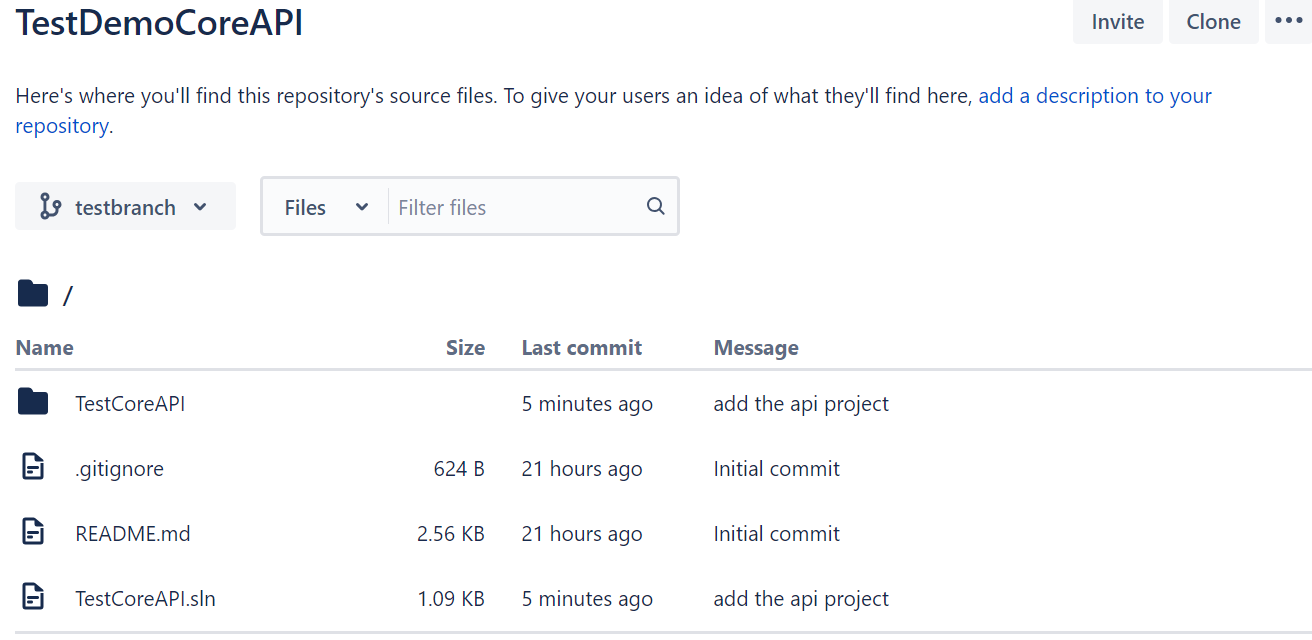
Syntax: git push [variable name] [branch]

This command sends the branch commits to your remote repository.

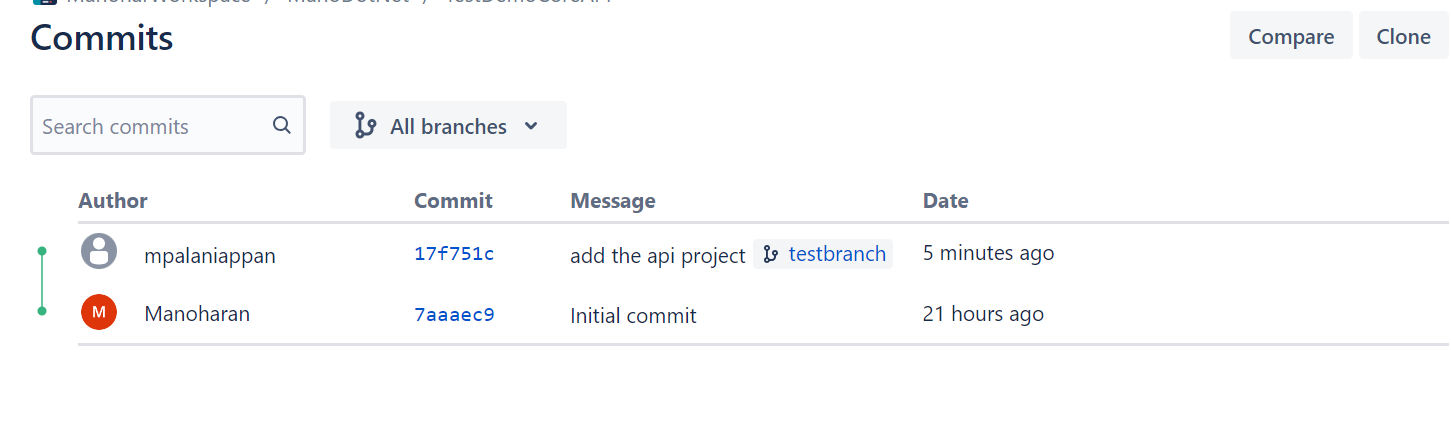


After push, the code changes added in remote repo.





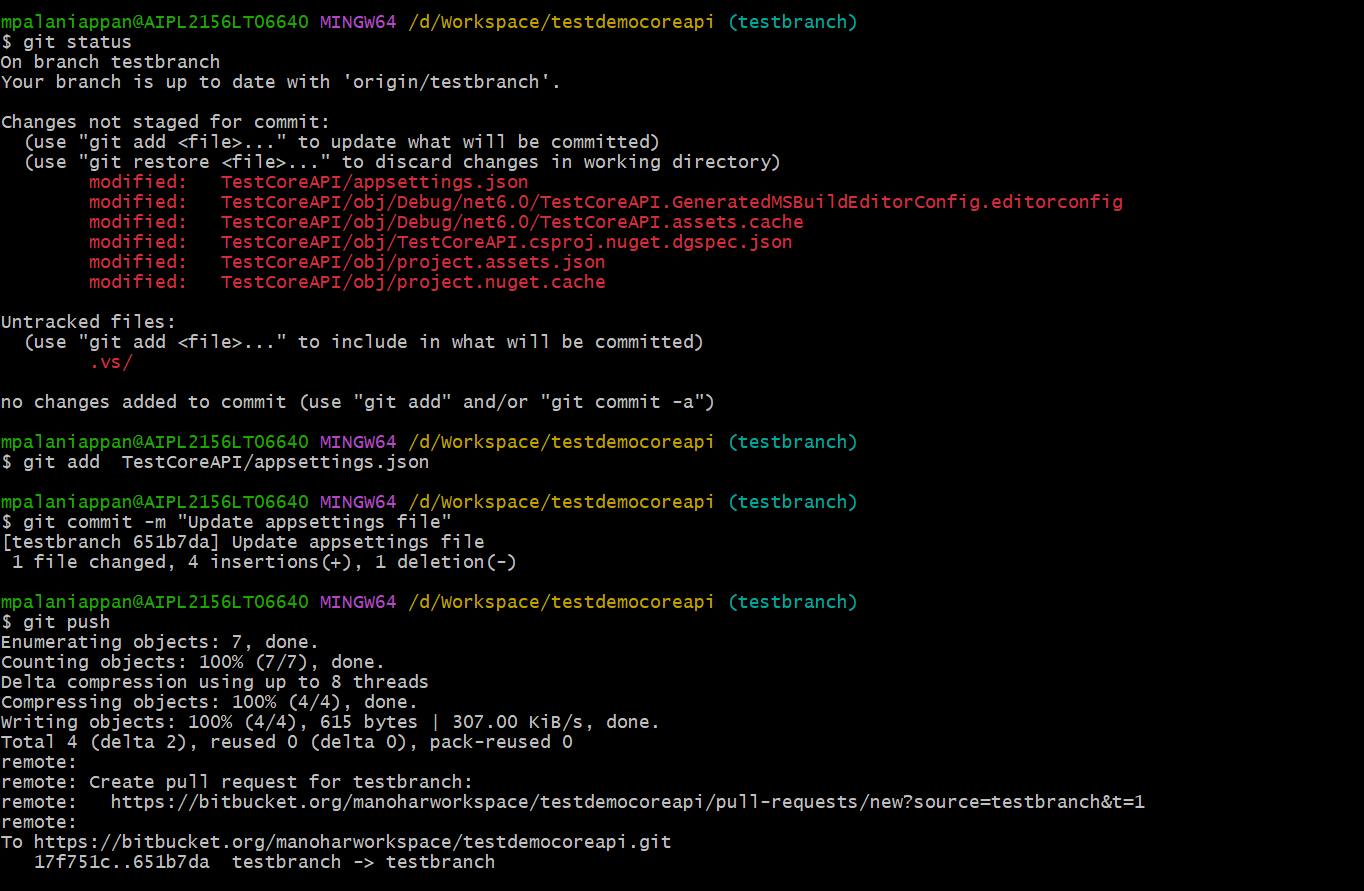
Branch commits list:

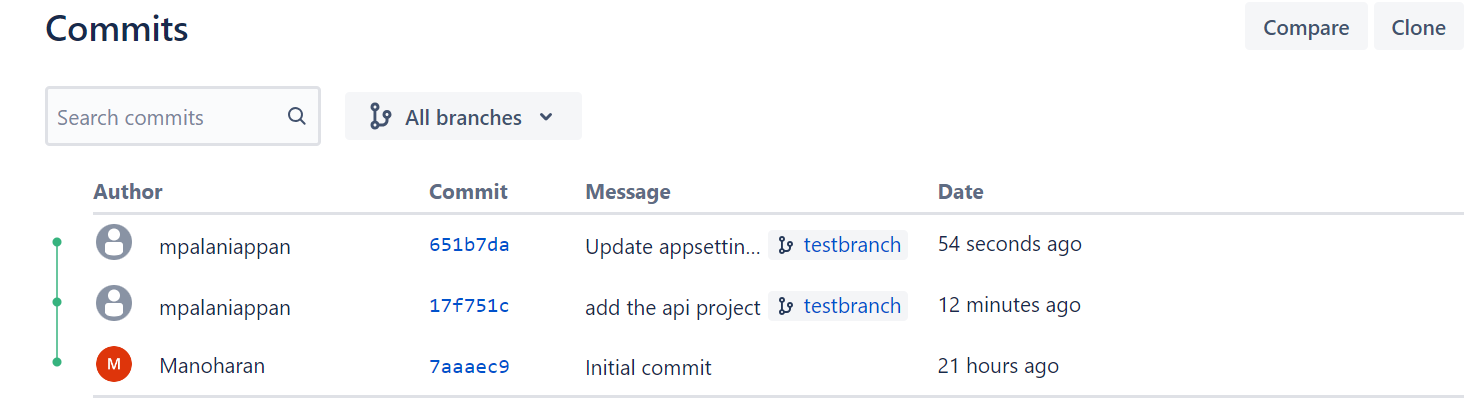


**Second commit:**

Now we do the changes in existing appsettings.json file and commit and push the code changes.

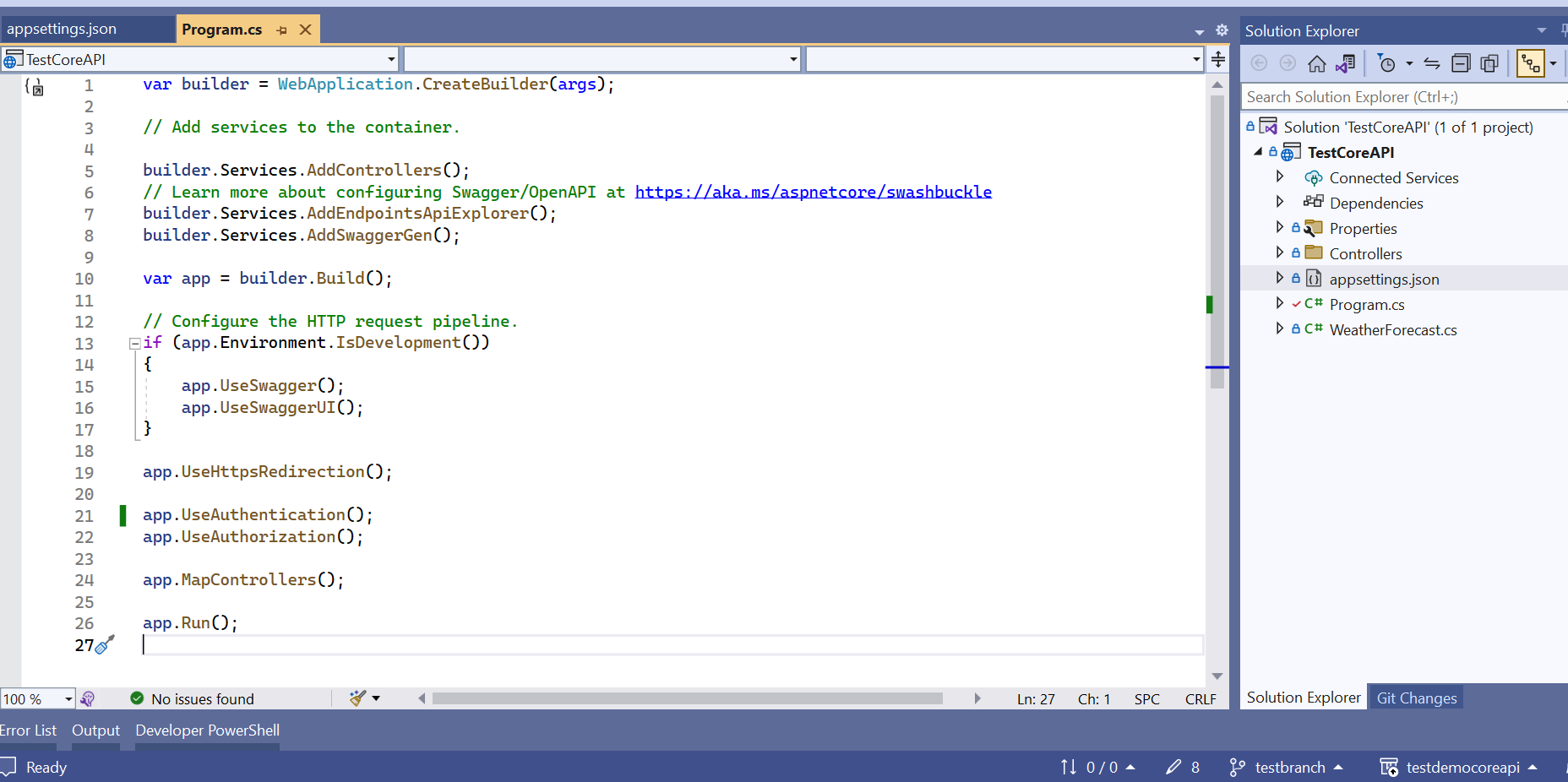


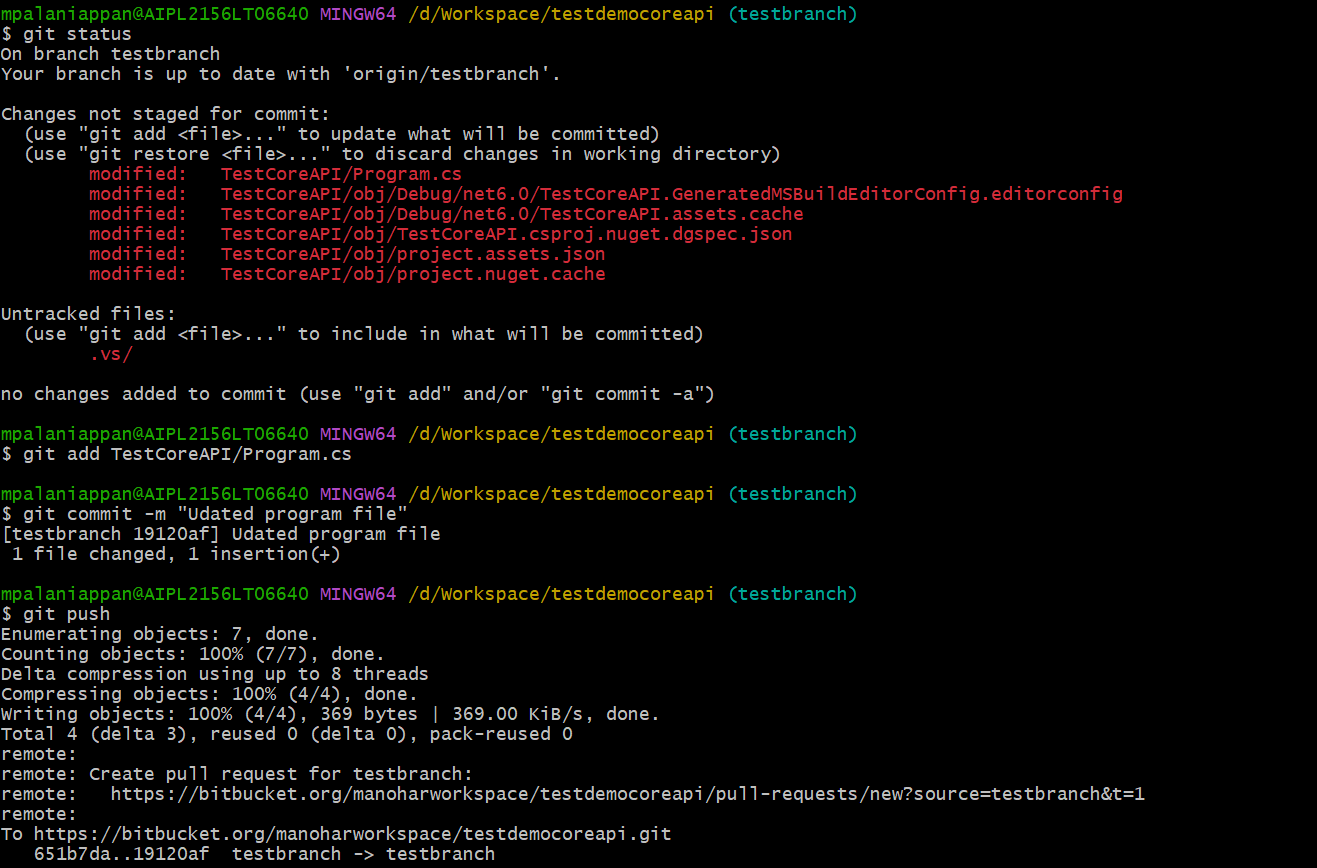


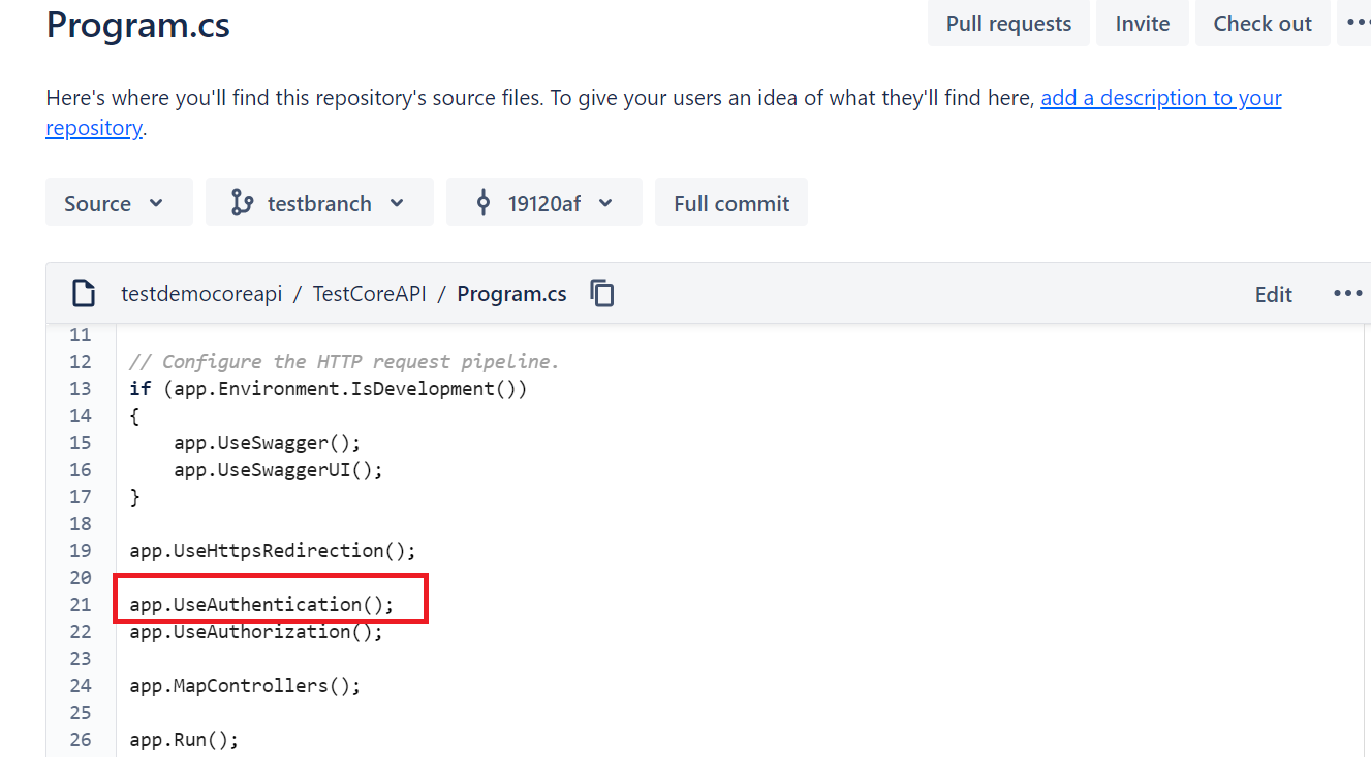


**Third Commit:**

Now we do the changes in existing program.cs file and commit and push the code changes.







Those changes reflected in bitbucket repo.

**How to revert the commit**

The purpose of the git revert command is to remove all the changes a single commit made to your source code repository.

**For example,** if a past commit updated a file named appsettings.json to the repo, a git revert on that commit will remove the appsettings.json file code changes from the repo. If a past commit added a new line of code to appsettings file, a git revert on that commit will remove the added line.

When you revert a Git commit, the changes from the targeted commit are removed from your local workspace. A new commit is also created to reflect the new state of your repository.

Now we going to revert the commit, for that first need to check log.

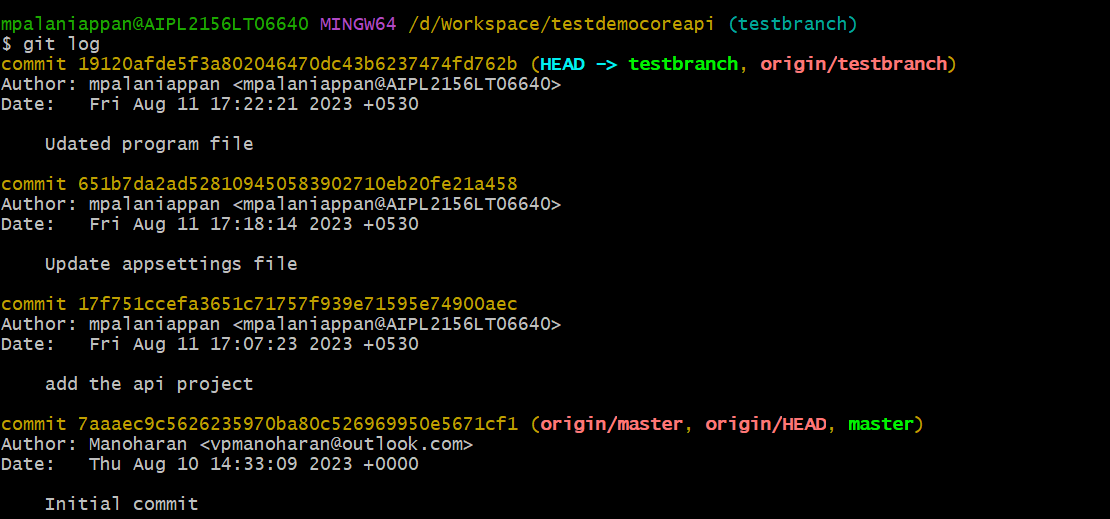
**git log**

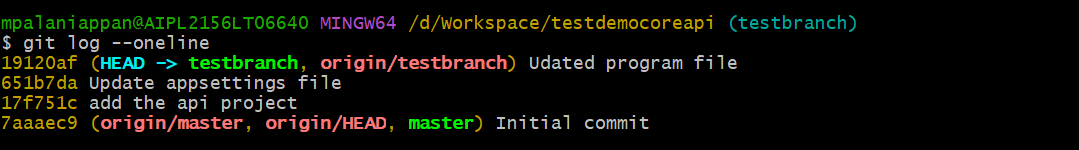
Syntax: git log

This command is used to list the version history for the current branch.

(or)

Syntax: git log --oneline





Above log list show the branch commit list, use this commit id we revert the appsettings commit.

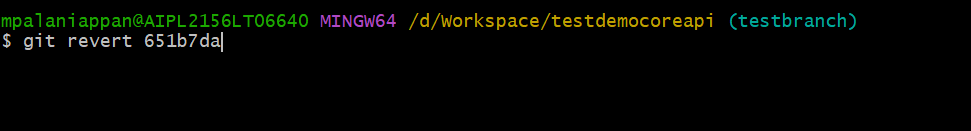
**Git revert**

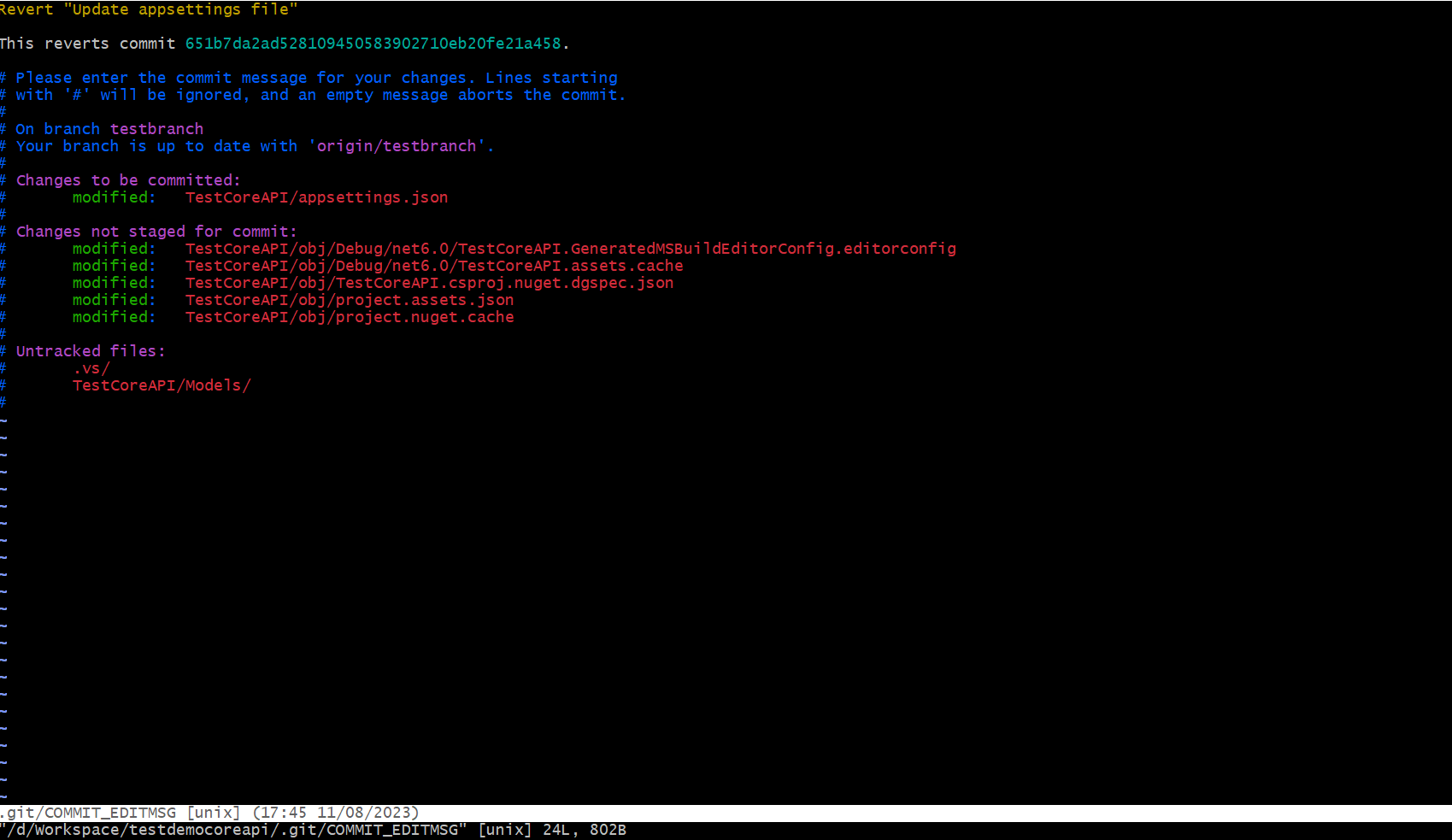
Syntax: git revert <commit-ish>

Use the revert command to undo the changes made in commits pushed to shared branches.

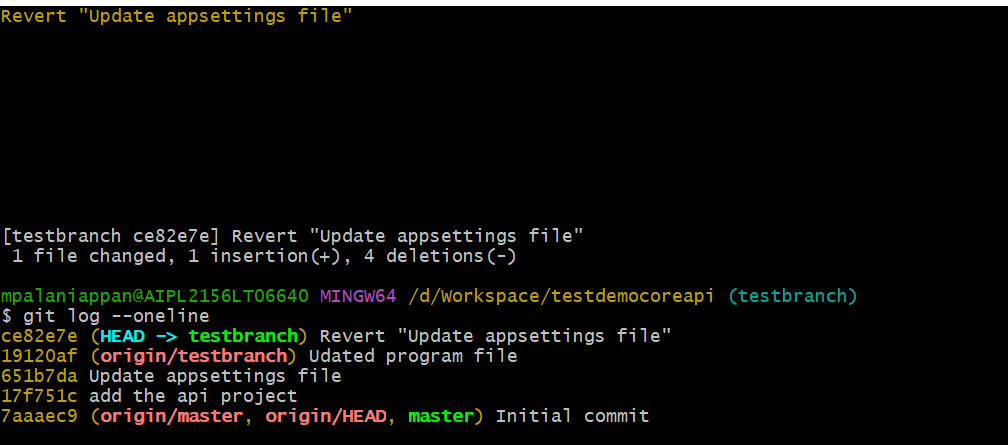
The revert command creates a new commit that undoes the changes made on a previous commit.

The revert command doesn't rewrite the repository history, which makes it safe to use when you're working with others.

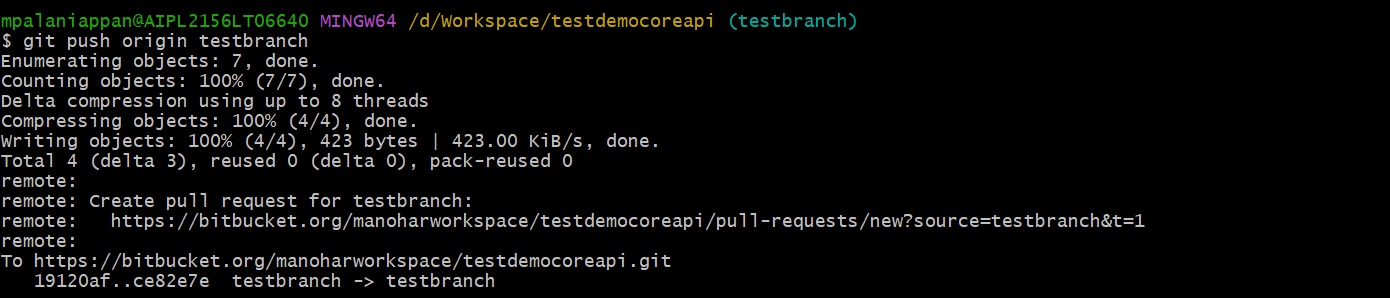




Appsetting commit is reverted in local repo and check in log.

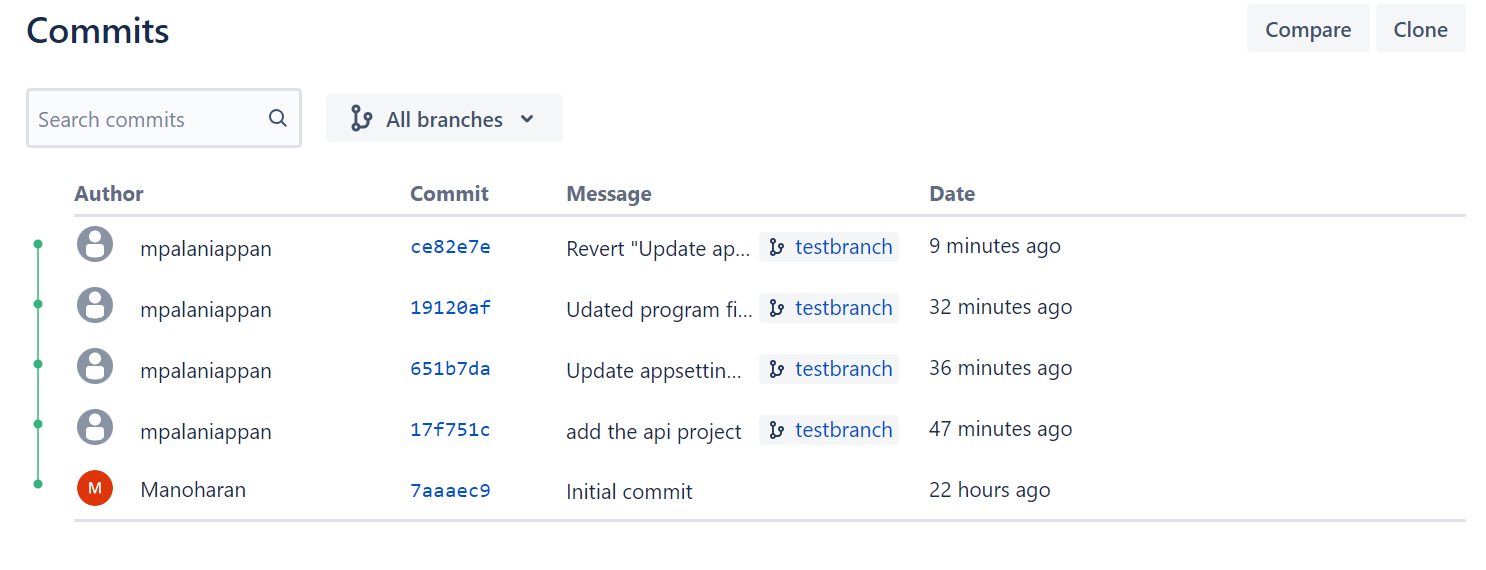


Push the revert commit to remote repo.



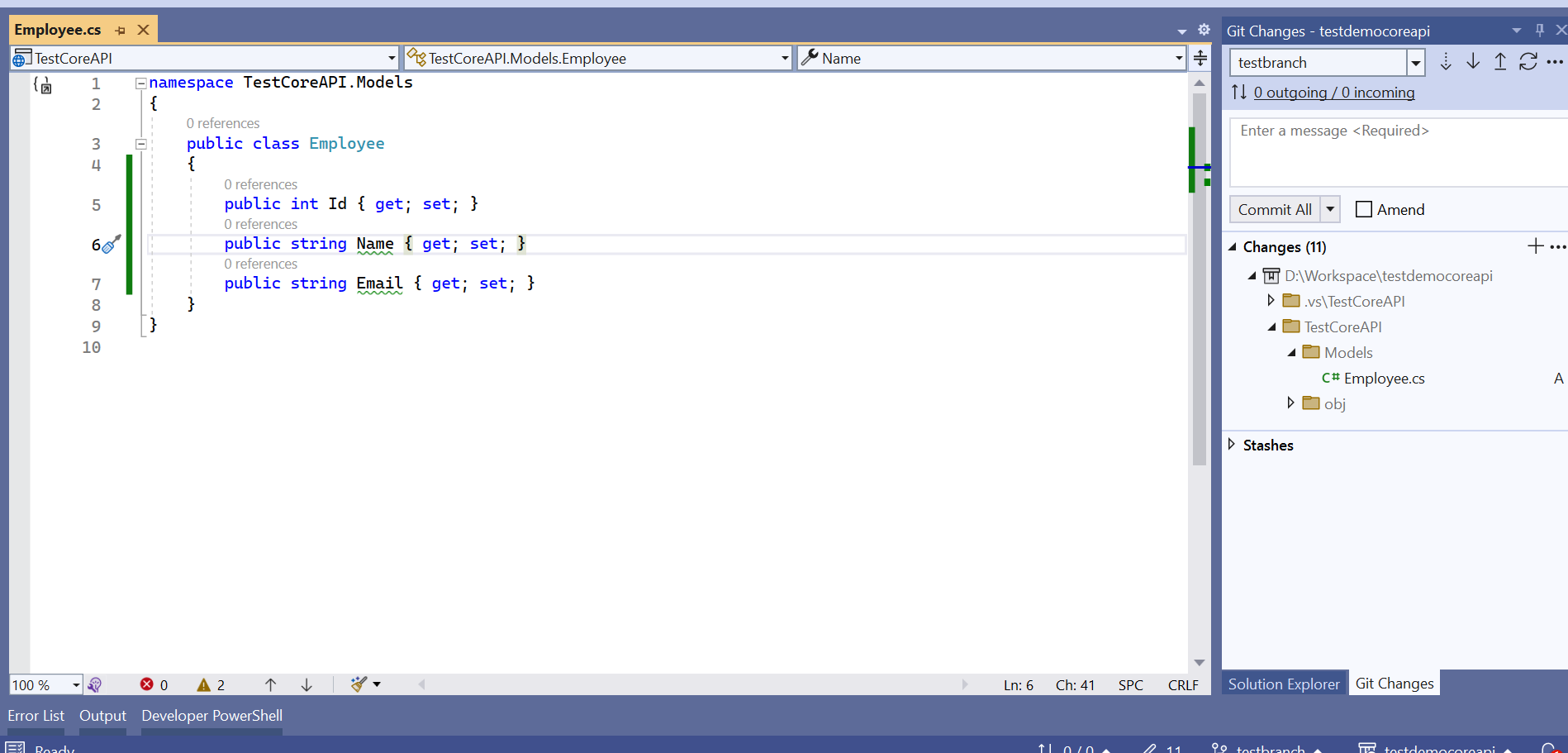
Revert changes reflected in bitbucket.

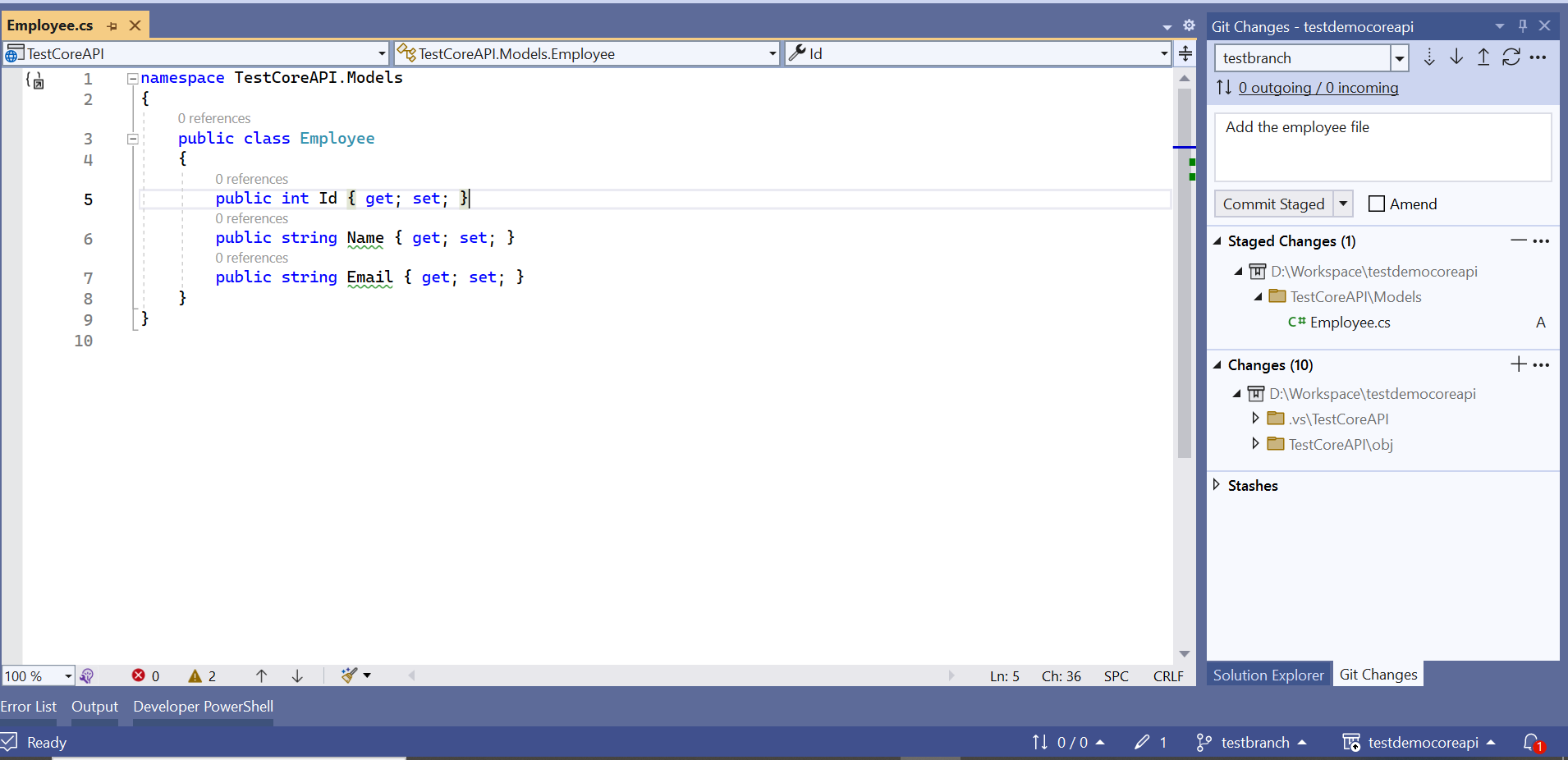


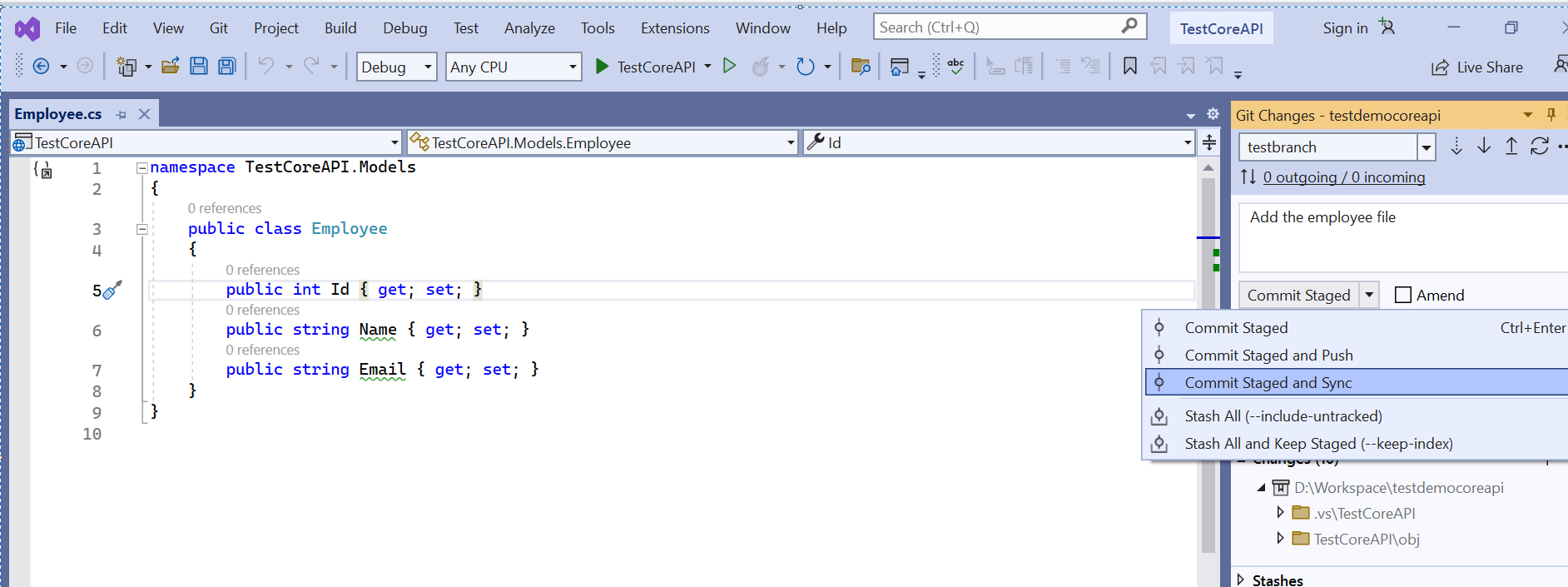


**Revert using Visual Studio IDE**

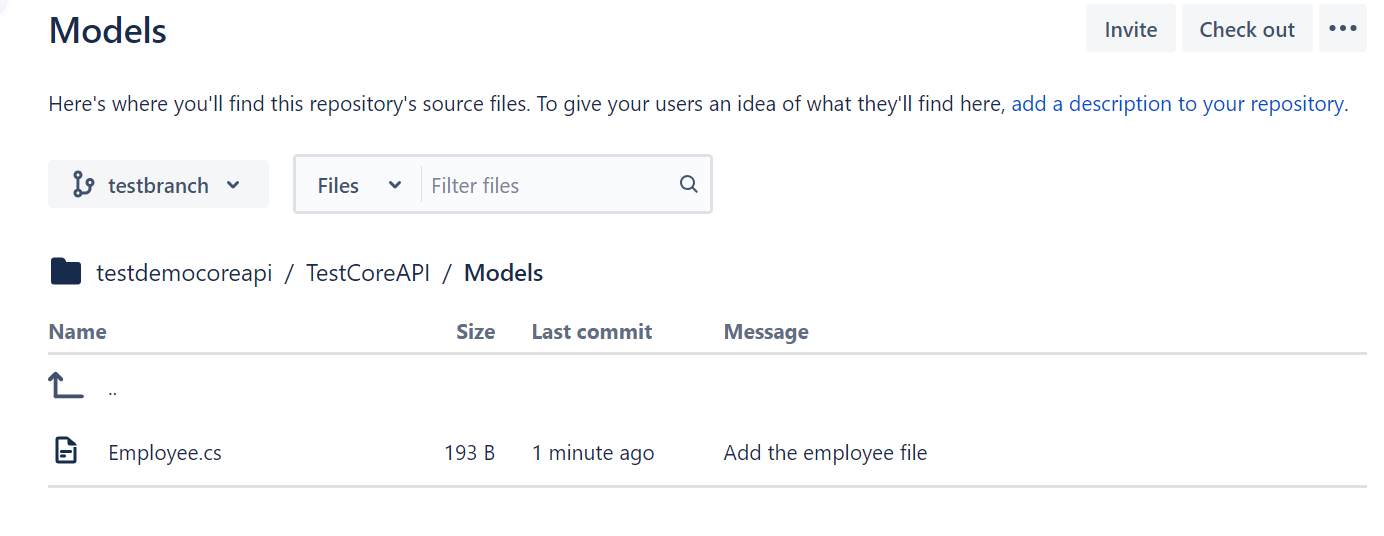
Add the new Employee file under the models folder, commit and push the changes to remote repo using git in visual studio IDE.

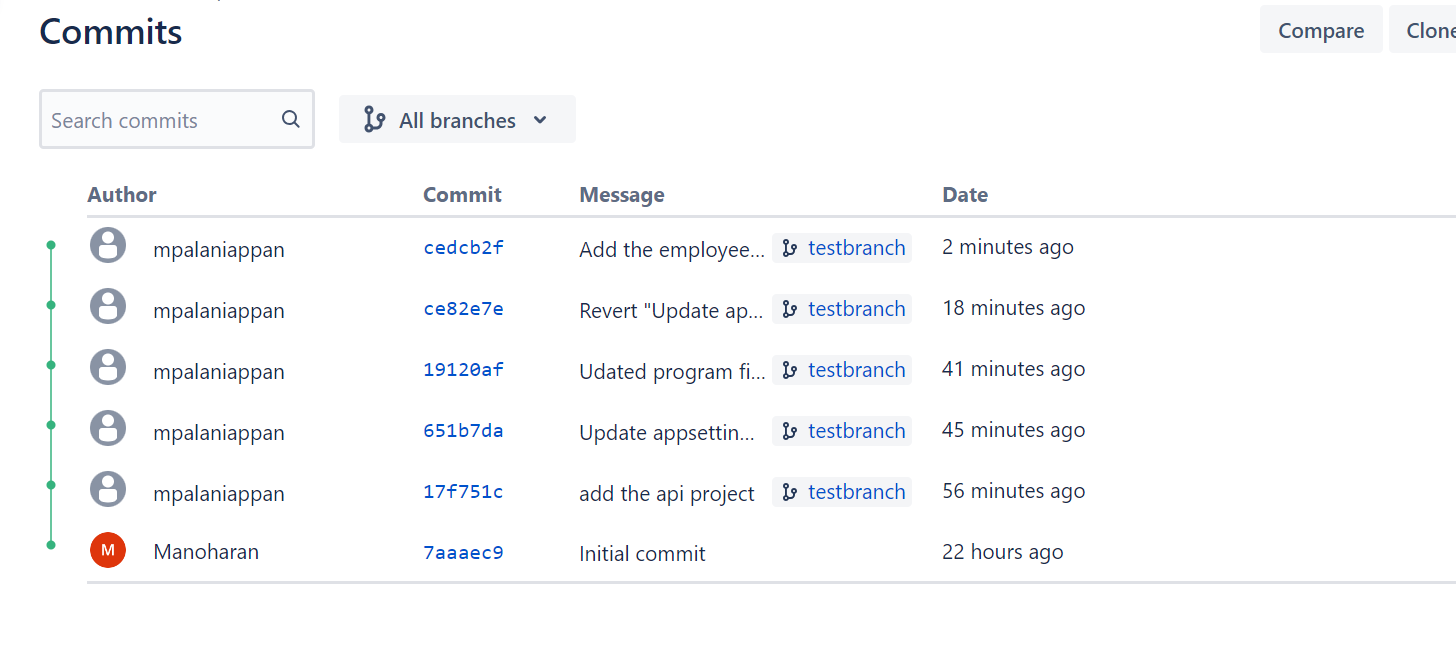






New changes reflected in bitbucket repo.

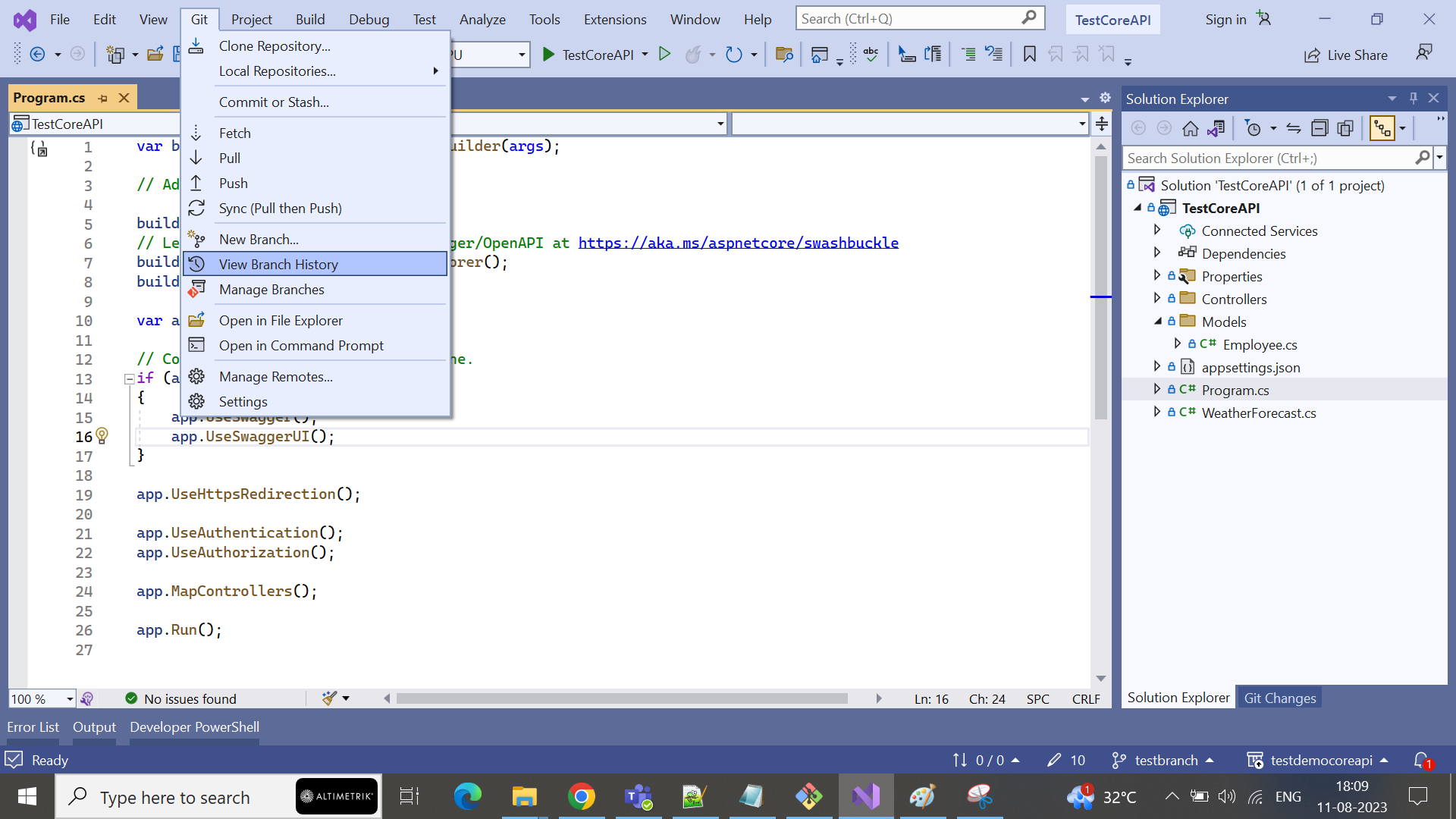


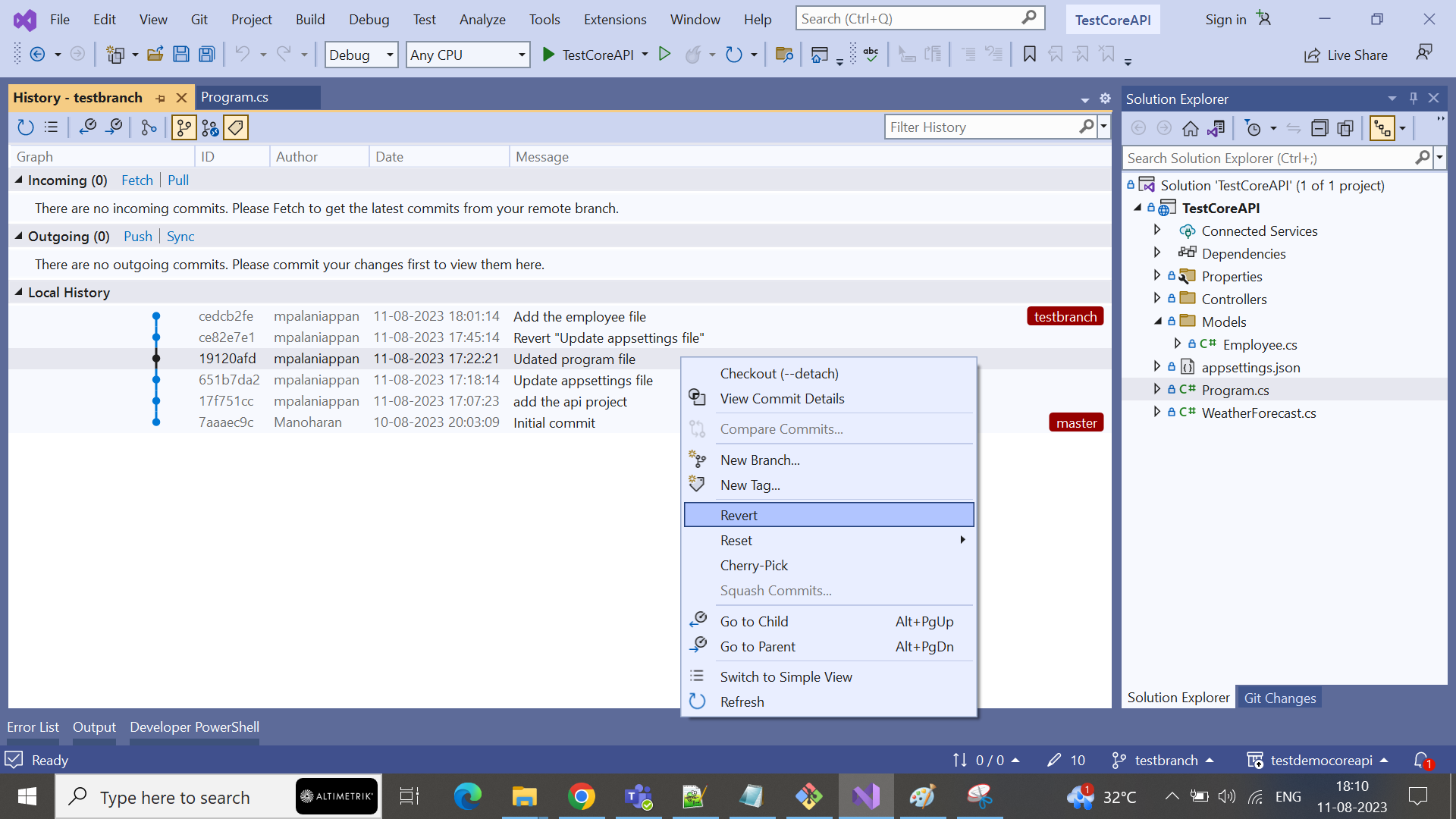


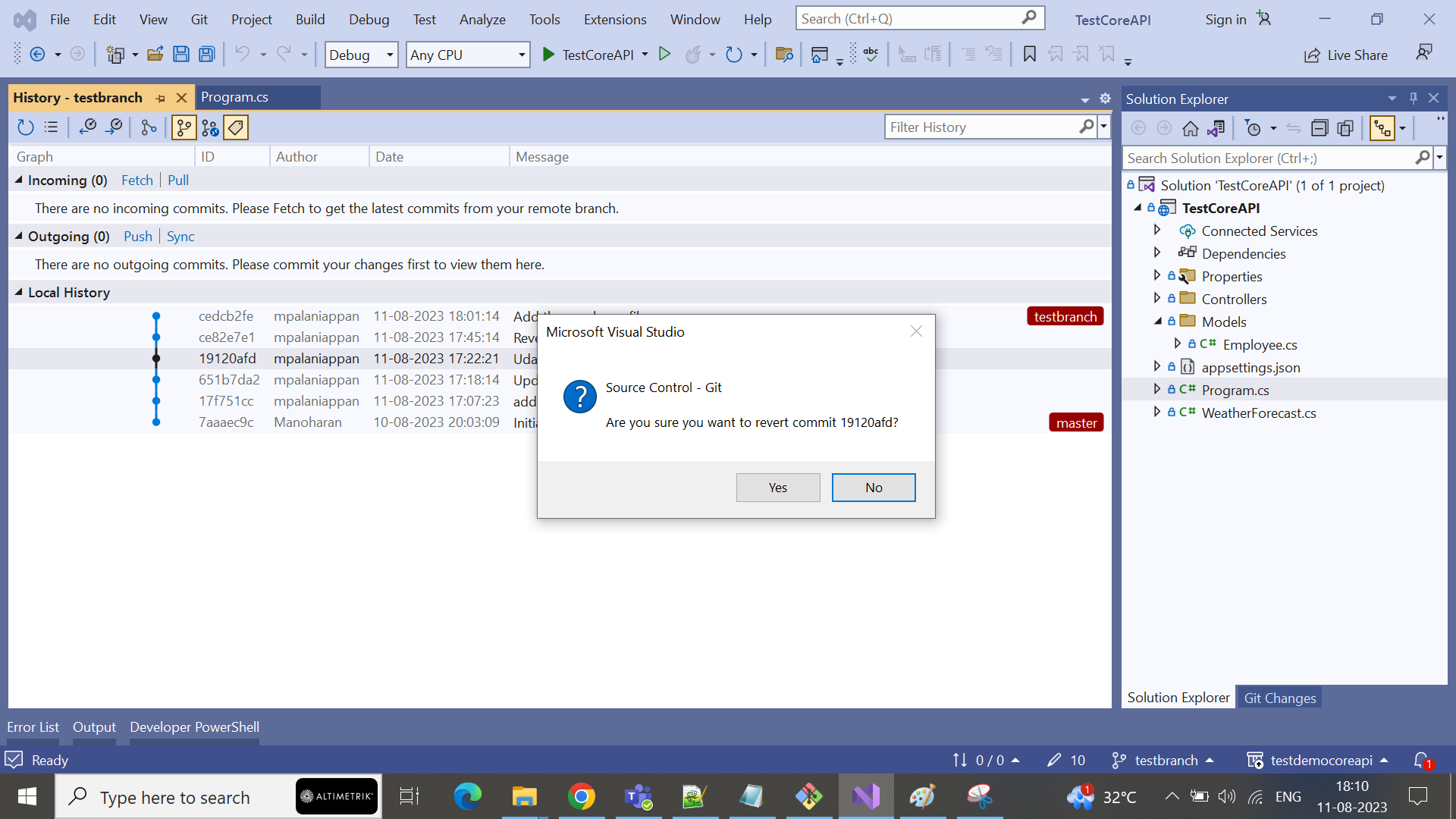
Revert the program changes commit in visual studio.

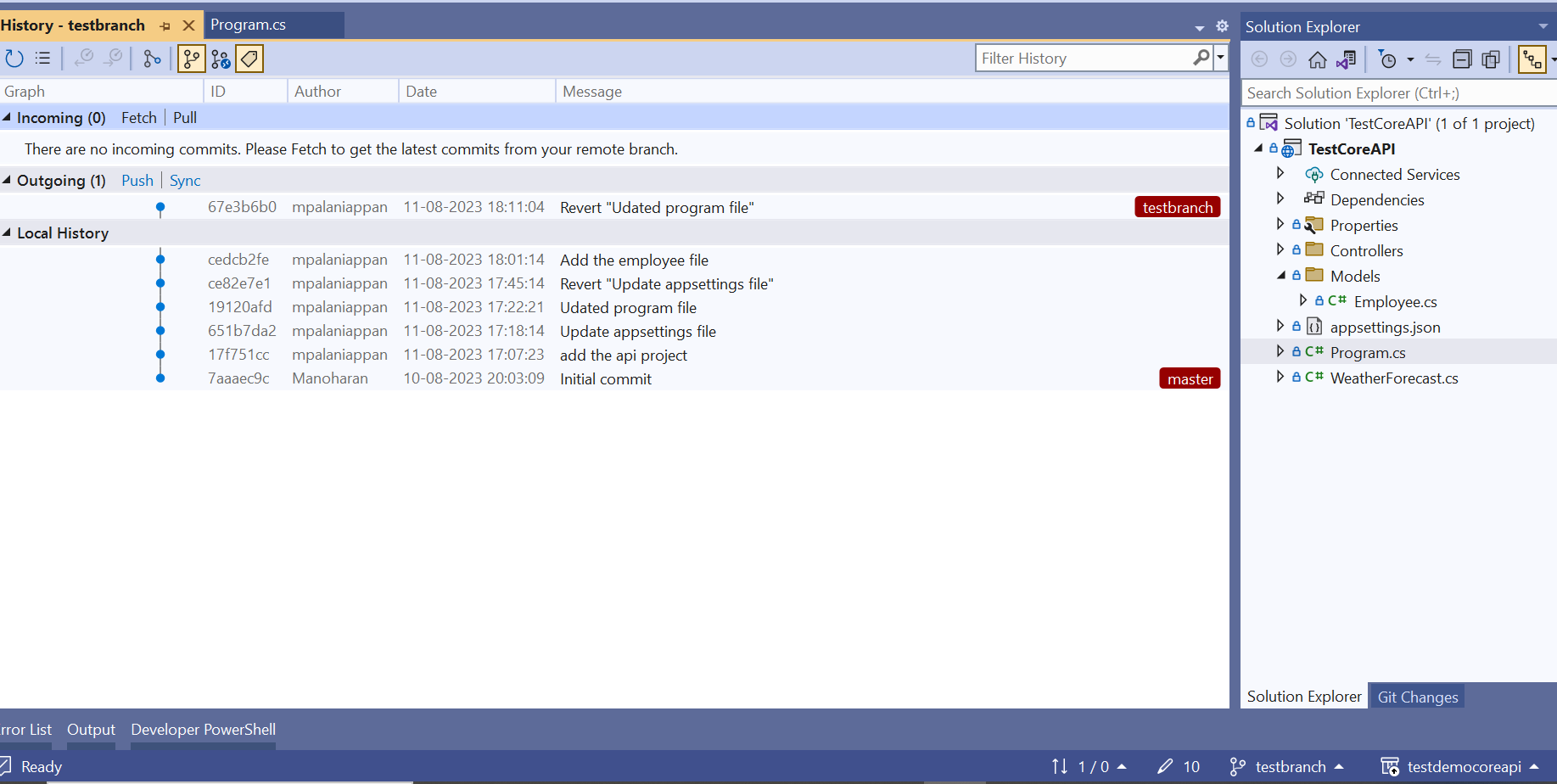
The commands will undo the changes made in commit **19120af** and create a new commit on the branch. The original commit is still in the Git history.

To do the same in Visual Studio, right-click the commit you want to revert and then select Revert. After you confirm your action and the operation is complete, Visual Studio displays a success message and a new commit appears in the **Outgoing** section.









Now click the push option in outgoing section. Changes reflected in remote repo.



