### …or create a new repository on the command line

echo "# CodeFristDemo" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/acheetah20/CodeFristDemo.git

git push -u origin master

### …or push an existing repository from the command line

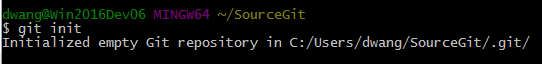
git remote add origin https://github.com/acheetah20/CodeFristDemo.git

git push -u origin master

### …or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Create the GitHub account and Create the folder in the local computer



Create a new C++ console project called HelloUSA



Here is the process for the new project

**// HelloUSA.cpp : this is created to try out the github different operations**

**// Setup new project**

**// Commit**

**// Sync**

**// Push to create new repo**

**// Create new branch if there is only master branch**

**// Working on branch**

**// Commit the change**

**// Sync**

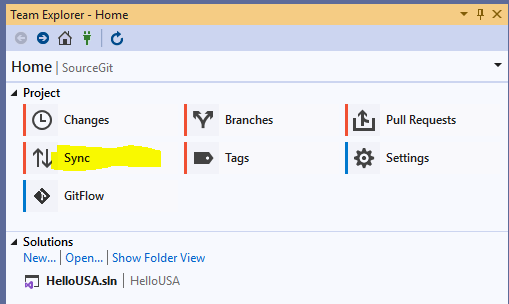
**// push**

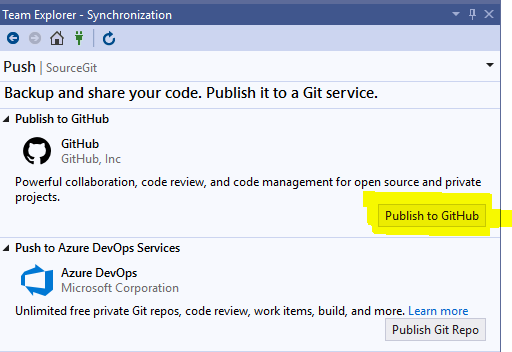
**// review the code and merge the change**

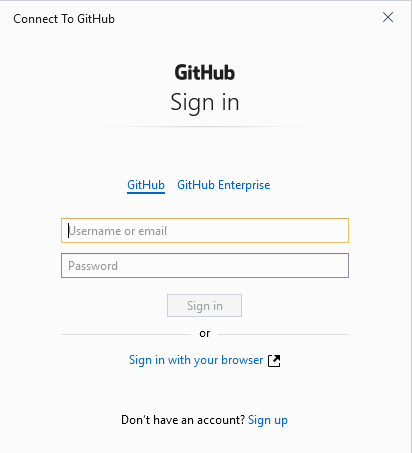
**// Pull the codeto your branch**

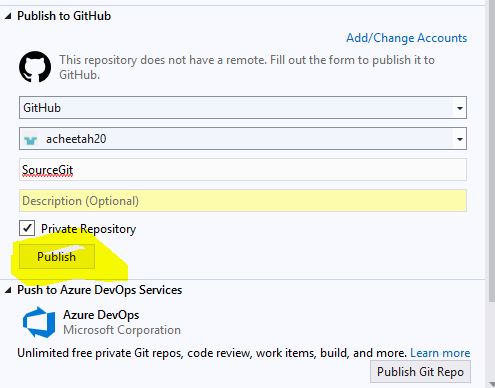
**// end of change**

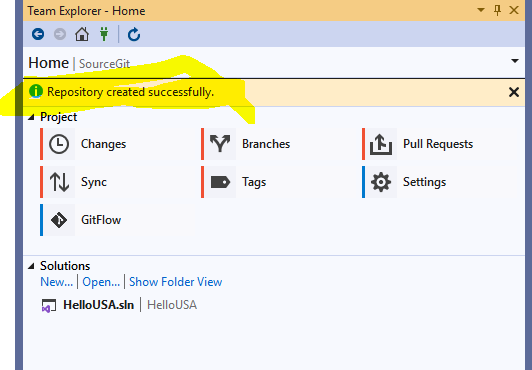
//









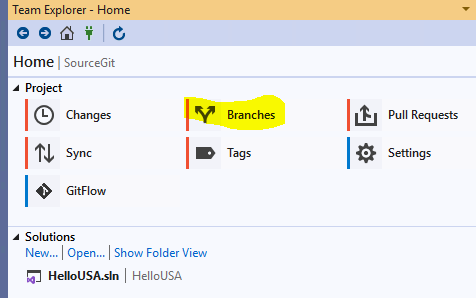


**Create a branch**

On the Team Explorer tab and use the main navigation dropdown to open the Branches view.

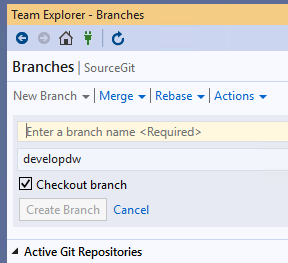
Usually the **1st contributions** repo is the default branch, which is called master.

Right-click on master and choose New Local Branch From....



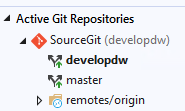
You could name your branch like develop-<your\_name\_here>, for example: develop-dwang.

Leave the Checkout branch box checked and click the Create Branch button

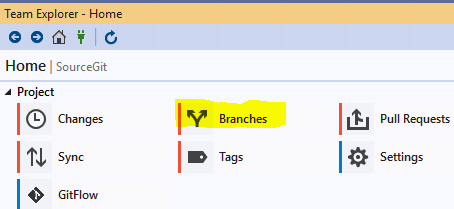


Click back to the Team Explorer tab and use the main navigation dropdown to open the Branches view.

You should see your new branch below



Now make the change on the HelloUSA project



## Commit & Push changes to GitHub

Make the change on the code.

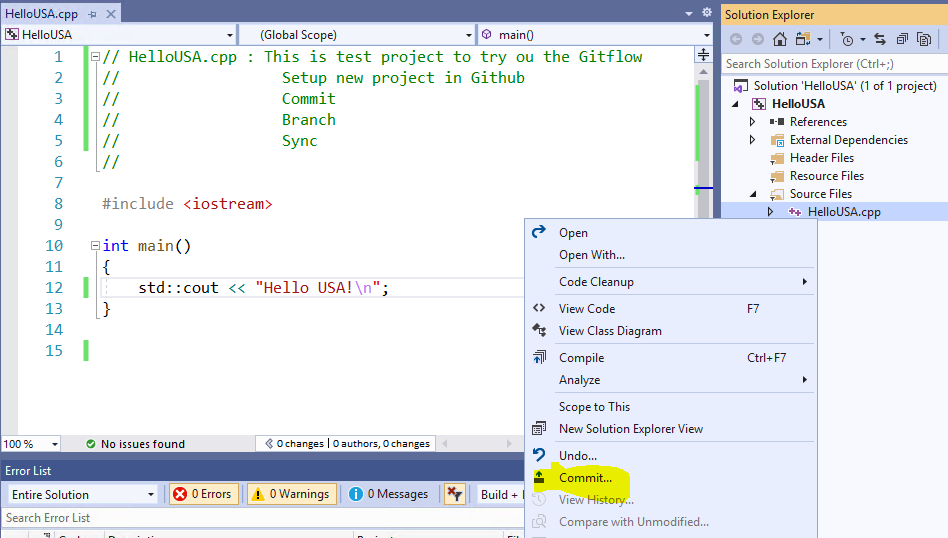
Switch back to Team Explorer and navigate to the Changes view.

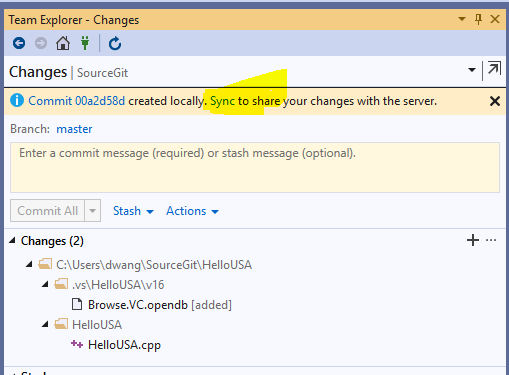
Now you chould see a list of changed files.

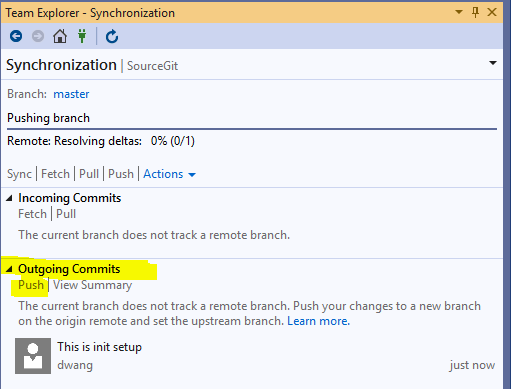
Comments should be in brief but thorough.

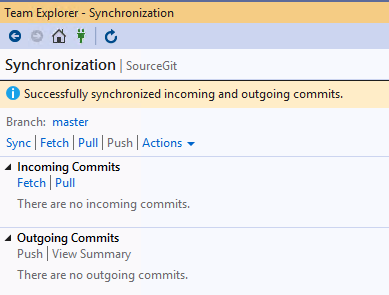
You could outline your commit. This is very important so you and team member will know what it is.

Then Commit All to perform a local commit and Push  to push your changes back up to your repo

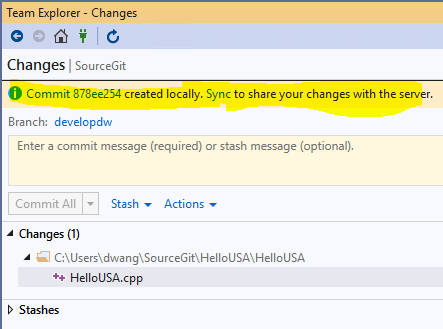
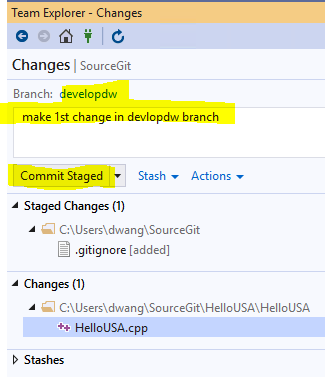
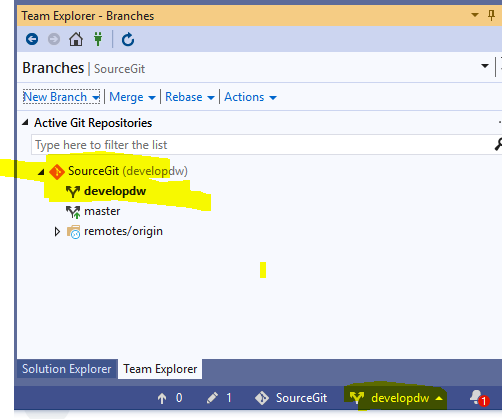


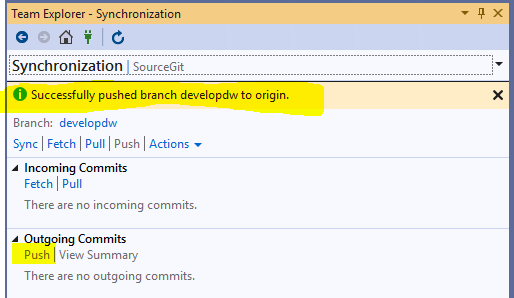


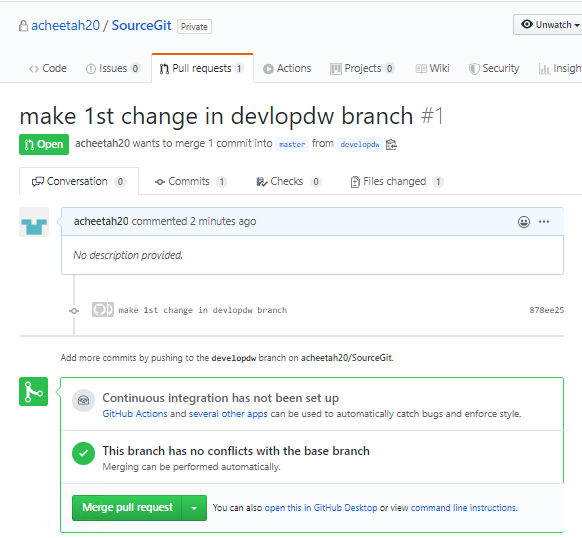


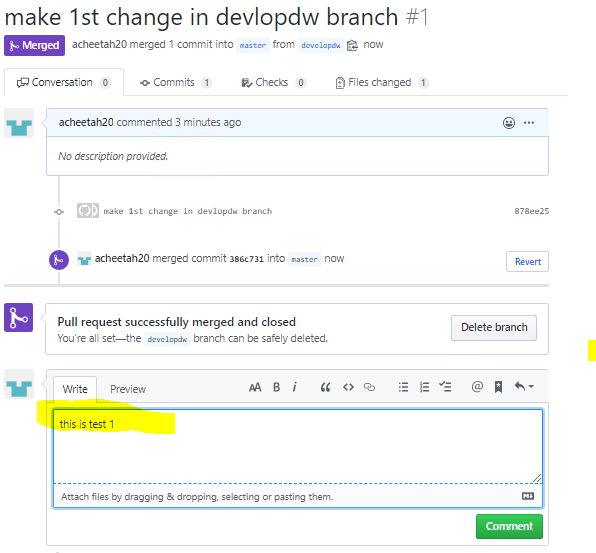


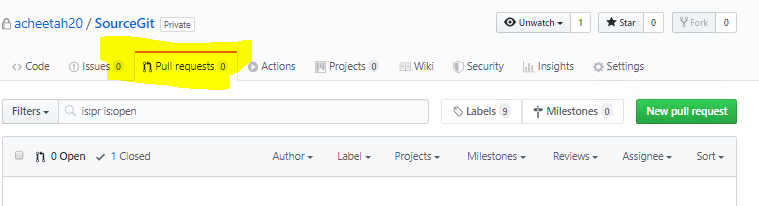
Create new branch.

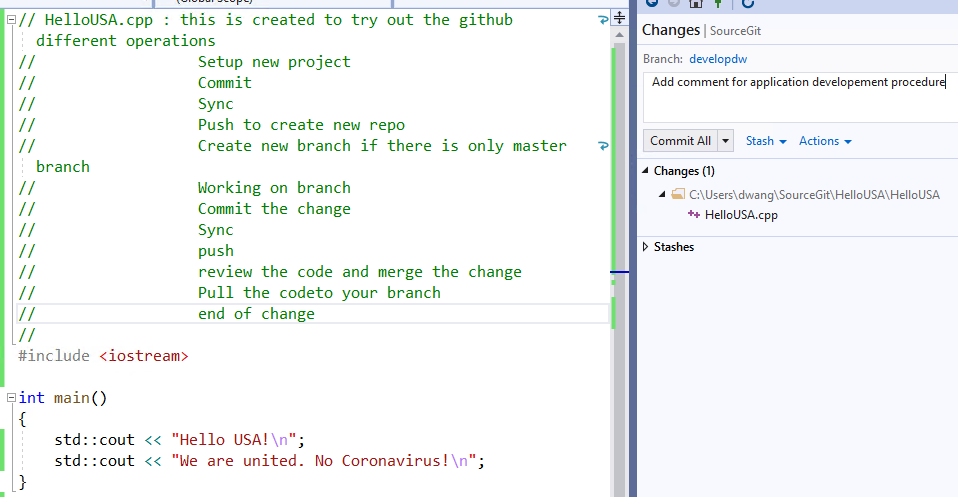


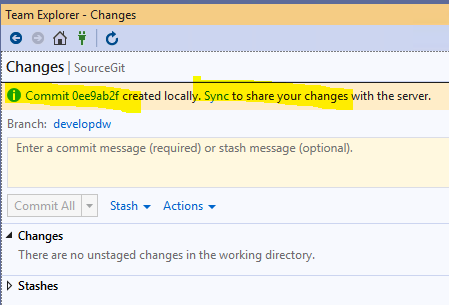


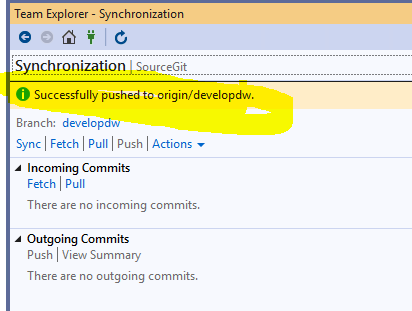
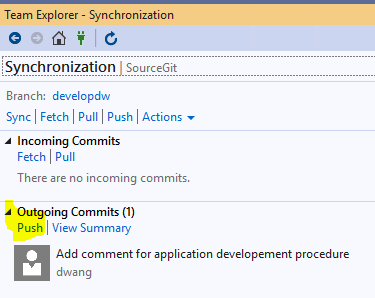








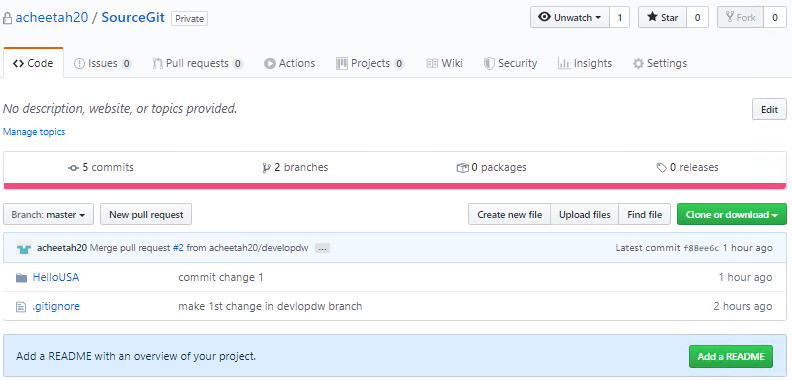


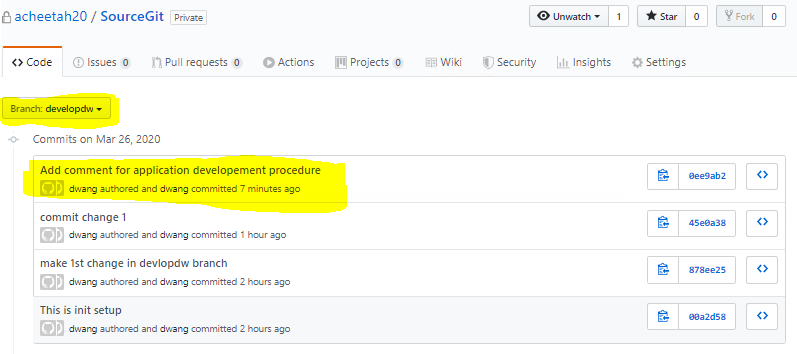


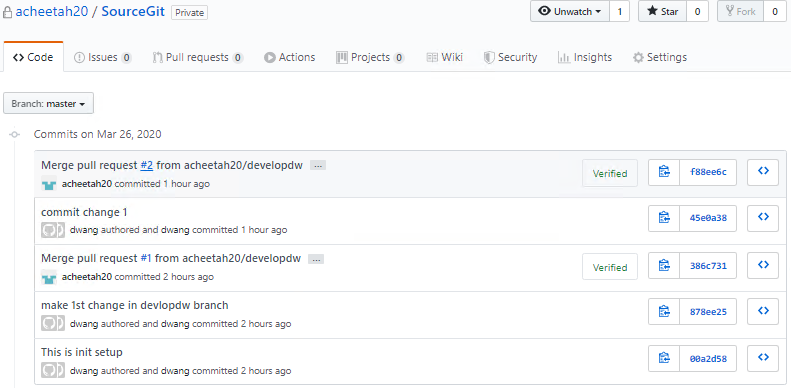
After the Push operation completes, open your repo in GitHub and you should see a message indicating a recently pushed branch.

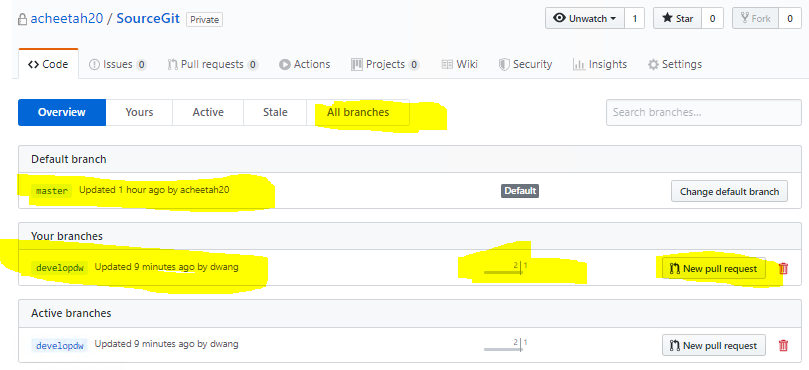
You could see your change in the github <https://github.com/acheetah20/SourceGit>

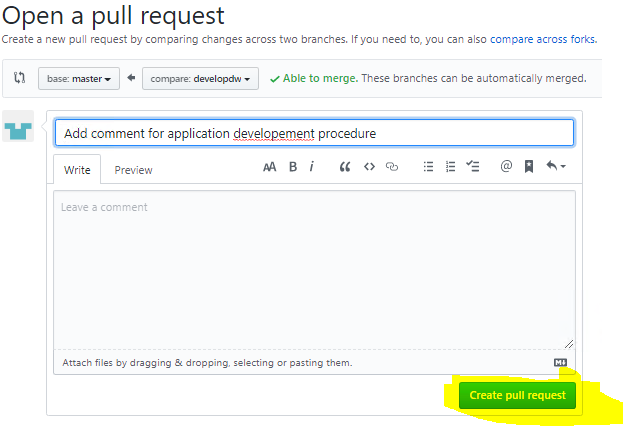
In the repo on GitHub you'll see the Compare & pull request button next to the new branch notification.

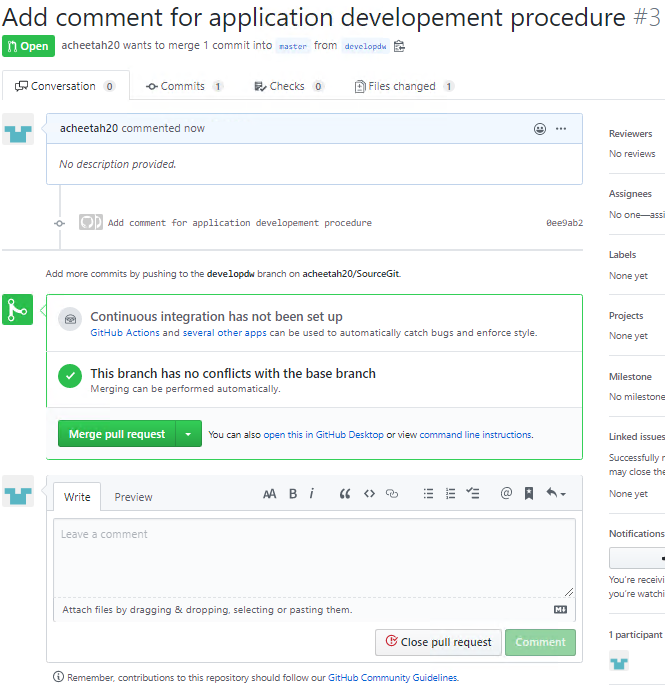


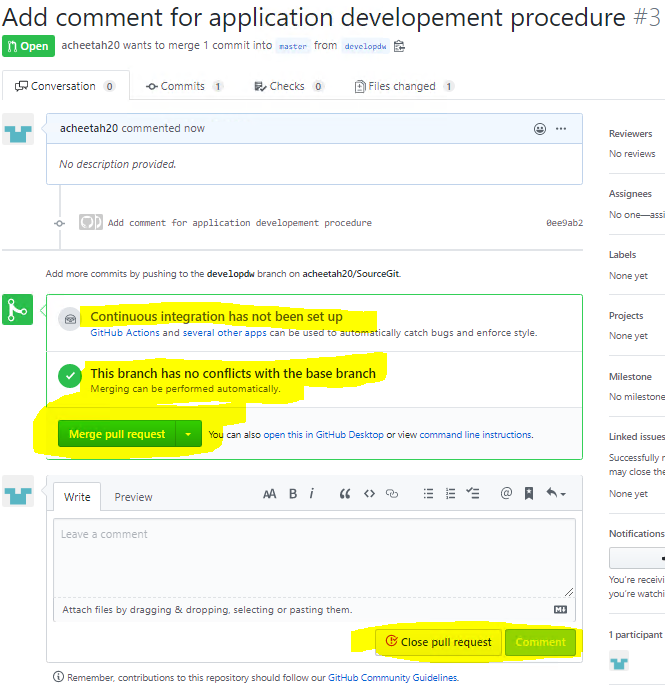


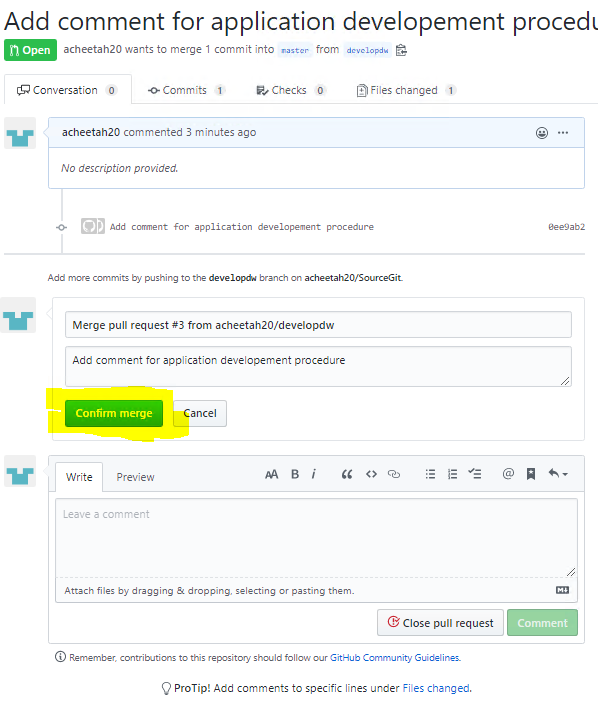


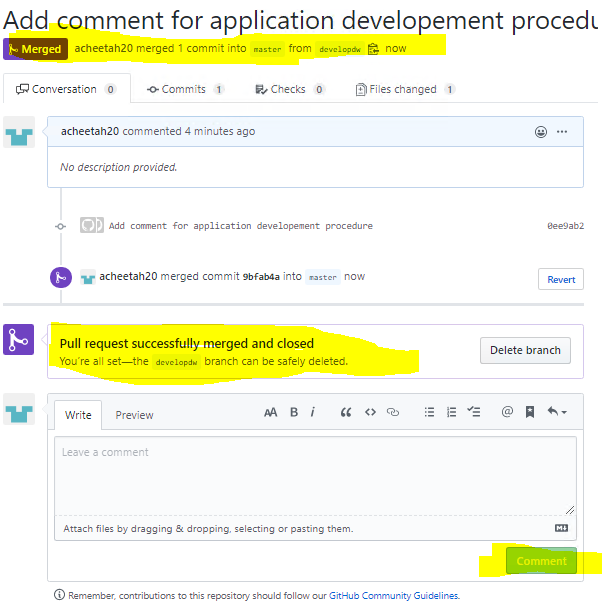


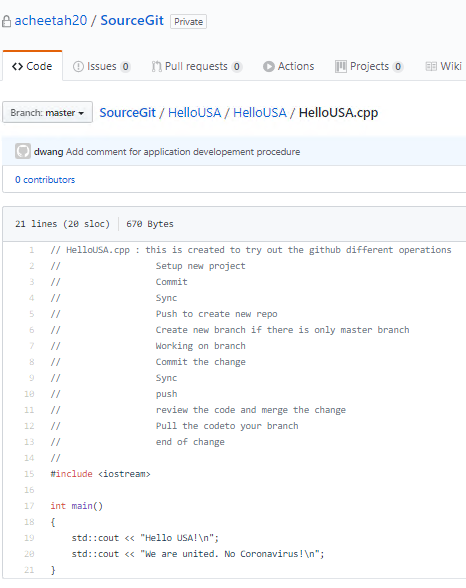




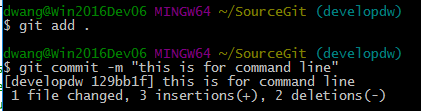
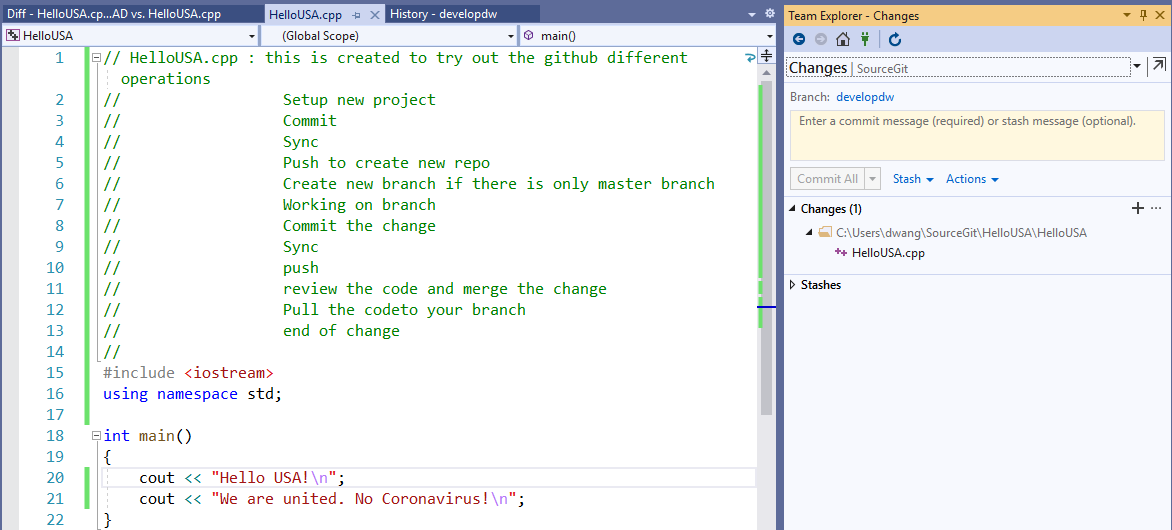








**Using Command line for Github**

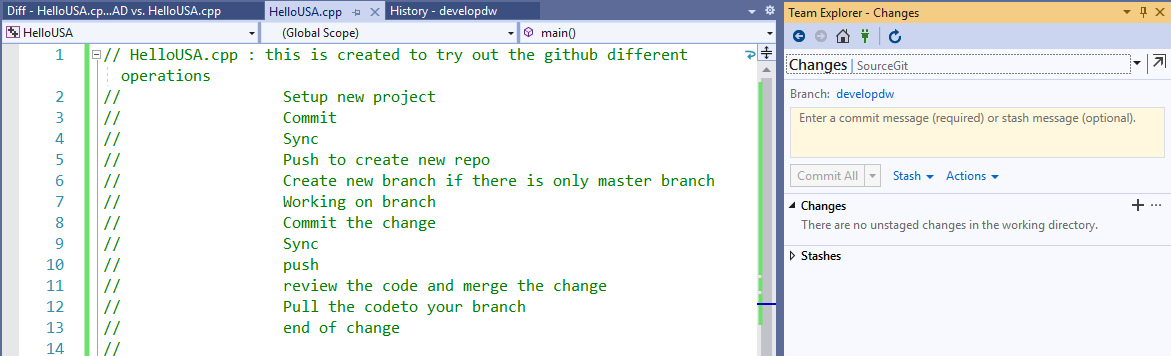


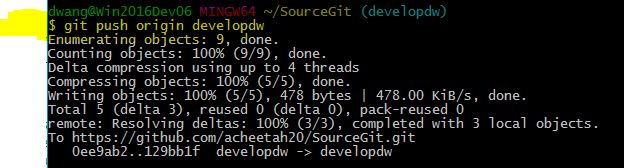
dwang@Win2016Dev06 MINGW64 ~/SourceGit (developdw)

$ git add .

dwang@Win2016Dev06 MINGW64 ~/SourceGit (developdw)

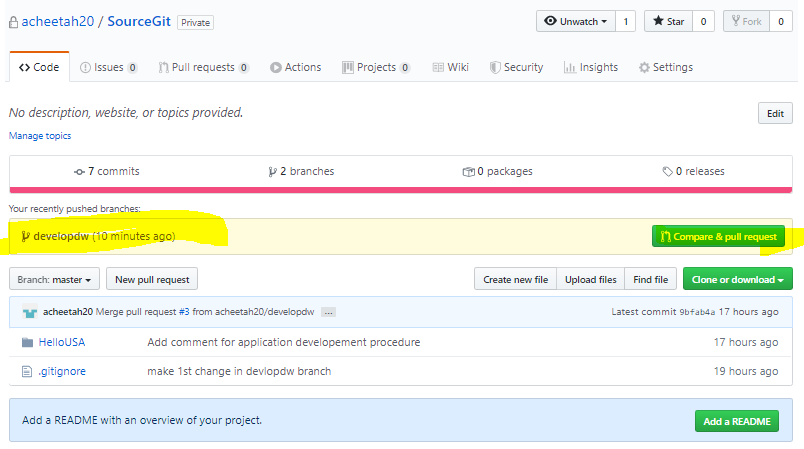
$ git commit -m "this is for command line"

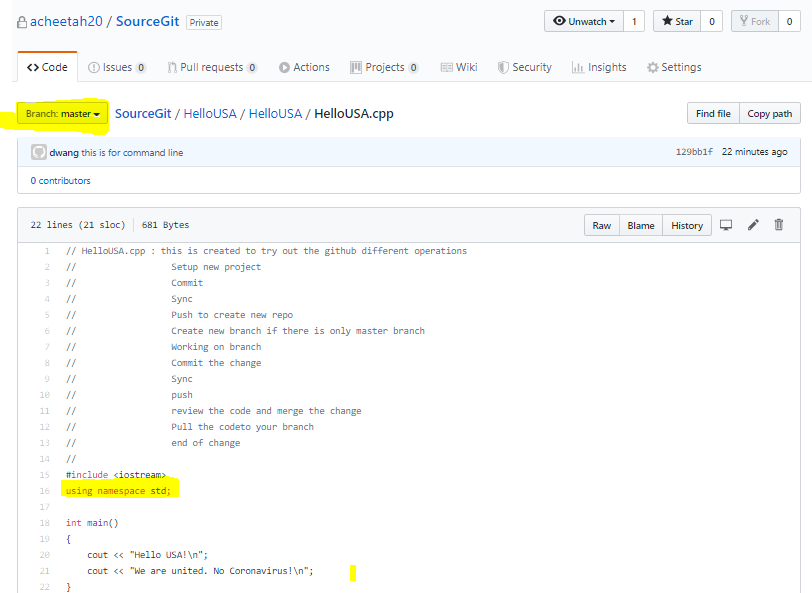
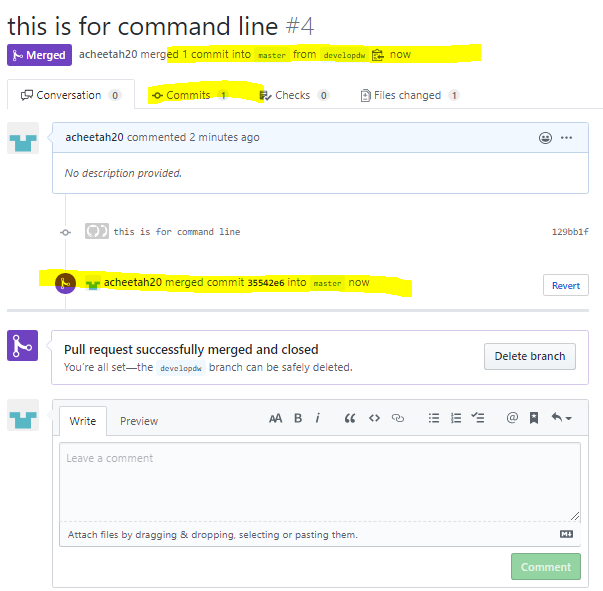
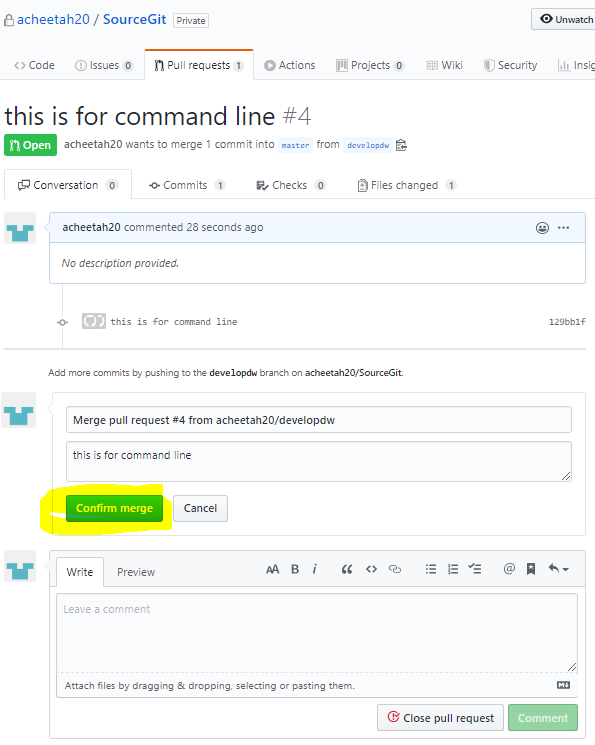
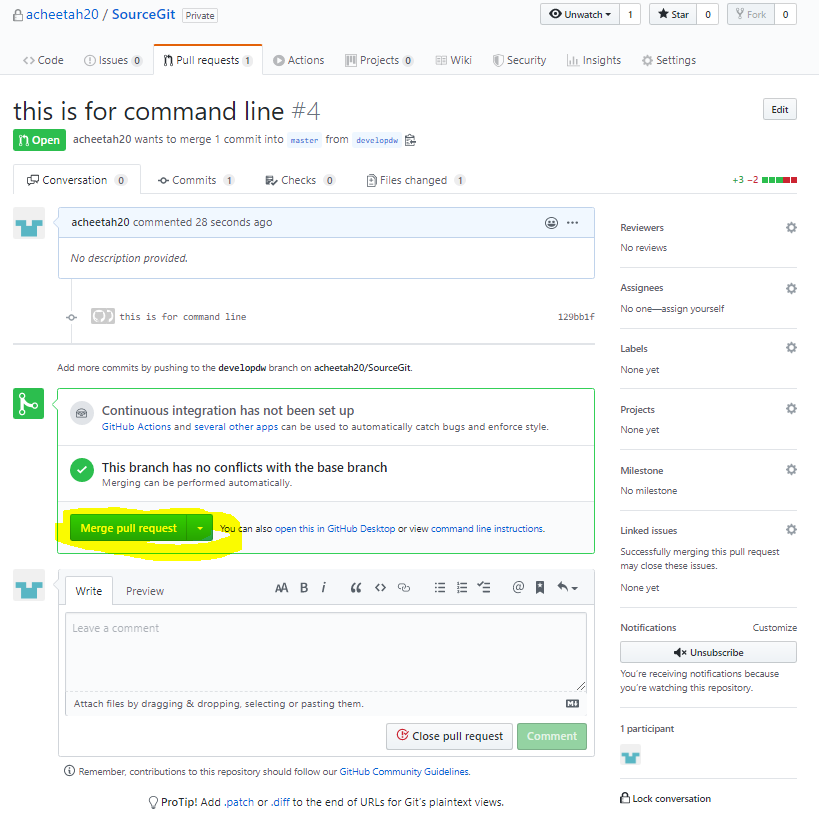
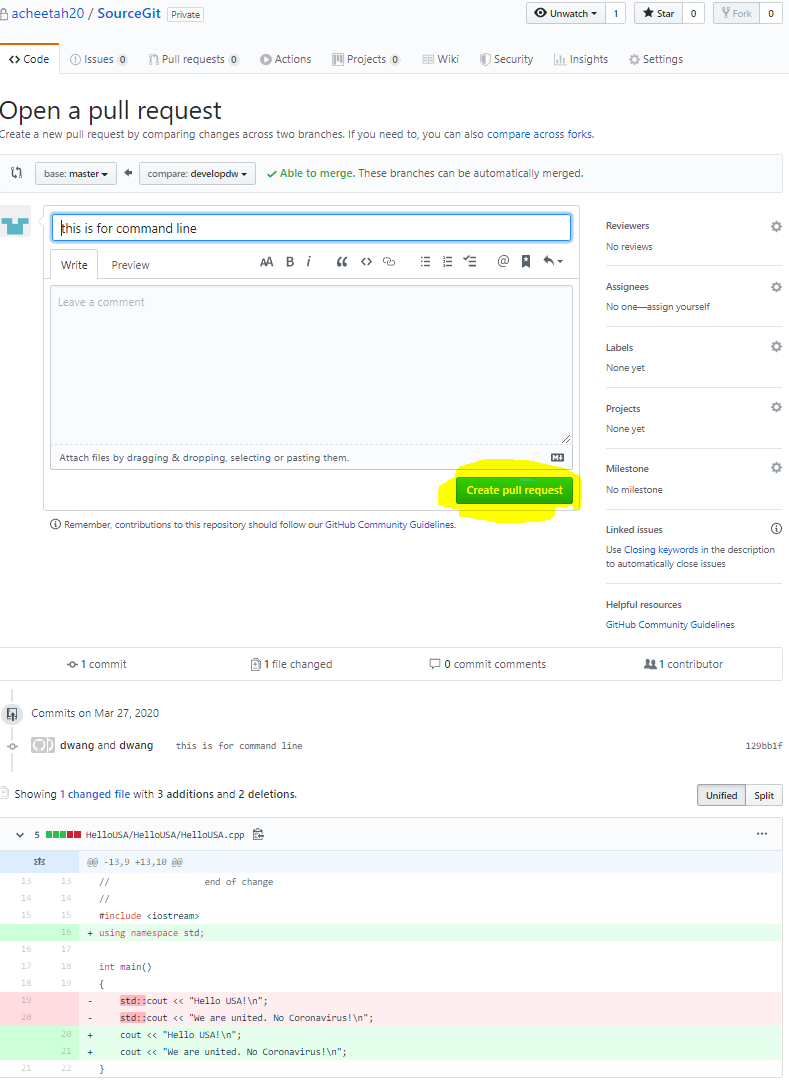




dwang@Win2016Dev06 MINGW64 ~/SourceGit (developdw)

$ git push origin developdw





Following command could add doc folder and all the files within the doc folder.

Then commit the change and push to the git

After that we could use git to review and merge and publish the change

