GIT Basics - My First Repo In GIT || PUSH & PULL In GIT

First we need to create the project in our eclipse at location we wanted .Now if we wan to push this code to the remote git repo, we need to perform the following things:

Ex:

Created a Demo project in eclipse at the location:

J:\MyEclipse\DemoGITRepo

Now from this we need to put the files into our remote gr repo which is at :

Git username :

FJaveed1 / fouzia01@gmail.com

pswd: fjGithub84

In the same way we need to create the new repositrory in git so that we can send our local project to the remote repo.So created a new repository called DemoRepofj.

git remote add origin <https://github.com/FJaveed1/DemoRepofj.git>

Steps to do the git :



we need to do cmd promt from the location of our local project :

ex : go to J:\MyEclipse\DemoGITRepo

cd J:\MyEclipse\DemoGITRepo

J:\MyEclipse\DemoGITRepo>dir

Volume in drive J is J

Volume Serial Number is 7A77-8427

Directory of J:\MyEclipse\DemoGITRepo

07/08/2020 05:45 PM <DIR> .

07/08/2020 05:45 PM <DIR> ..

07/08/2020 05:22 PM 1,210 .classpath

07/08/2020 05:07 PM 563 .project

07/08/2020 05:07 PM <DIR> .settings

07/08/2020 05:21 PM 3,015 pom.xml

07/08/2020 05:07 PM <DIR> src

07/08/2020 05:07 PM <DIR> target

3 File(s) 4,788 bytes

5 Dir(s) 418,433,753,088 bytes free

From this location in cmd , we need to do

git init

J:\MyEclipse\DemoGITRepo>git init

Initialized empty Git repository in J:/MyEclipse/DemoGITRepo/.git/

At this point the hidden directory .git will be created in the folder.

Now to connect the local dir to the remote repo , we need to type in the following : This link will be available at our rmote git repo

git remote add origin <https://github.com/FJaveed1/DemoRepofj.git>

J:\MyEclipse\DemoGITRepo>git remote add origin https://github.com/FJaveed1/DemoRepofj.git

Now we need to push our local project to the remote, we need to add the complete project folder .For this we need to chk the current status to see the pending files.

git remote set-url origin  https://github.com/FJaveed1/POMfj.git

J:\MyEclipse\DemoGITRepo>git status

On branch master

No commits yet

Untracked files:

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

.classpath

.project

.settings/

pom.xml

src/

target/

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

---------------------------------------

Here we can see that the above files are red in color which are pending .So to add all of them into git we need to do the following

J:\MyEclipse\DemoGITRepo>git add .

J:\MyEclipse\DemoGITRepo>git status

On branch master

No commits yet

Changes to be committed:

(use "git rm --cached <file>..." to unstage)

new file: .classpath

new file: .project

new file: .settings/org.eclipse.core.resources.prefs

new file: .settings/org.eclipse.jdt.core.prefs

new file: .settings/org.eclipse.m2e.core.prefs

new file: pom.xml

new file: src/main/java/com/qa/libs/TestBase.java

new file: src/main/java/com/qa/pages/HomPage.java

new file: src/main/java/com/qa/pages/LoginPage.java

new file: src/main/java/com/qa/util/TestUtil.java

new file: src/test/java/com/qa/tests/LoginTests.java

new file: target/classes/com/qa/libs/TestBase.class

new file: target/classes/com/qa/pages/HomPage.class

new file: target/classes/com/qa/pages/LoginPage.class

new file: target/classes/com/qa/util/TestUtil.class

new file: target/test-classes/com/qa/tests/LoginTests.class

---------------------------------------------------

Now after adding, and then checking the status we see all files are added and are green in color. But these files are just added .Now we need to commit all .For this we need the below command :

J:\MyEclipse\DemoGITRepo>git commit -m "First Git commit for my demo project"

[master (root-commit) 69edb55] First Git commit for my demo project

16 files changed, 203 insertions(+)

create mode 100644 .classpath

create mode 100644 .project

create mode 100644 .settings/org.eclipse.core.resources.prefs

create mode 100644 .settings/org.eclipse.jdt.core.prefs

create mode 100644 .settings/org.eclipse.m2e.core.prefs

create mode 100644 pom.xml

create mode 100644 src/main/java/com/qa/libs/TestBase.java

create mode 100644 src/main/java/com/qa/pages/HomPage.java

create mode 100644 src/main/java/com/qa/pages/LoginPage.java

create mode 100644 src/main/java/com/qa/util/TestUtil.java

create mode 100644 src/test/java/com/qa/tests/LoginTests.java

create mode 100644 target/classes/com/qa/libs/TestBase.class

create mode 100644 target/classes/com/qa/pages/HomPage.class

create mode 100644 target/classes/com/qa/pages/LoginPage.class

create mode 100644 target/classes/com/qa/util/TestUtil.class

create mode 100644 target/test-classes/com/qa/tests/LoginTests.class

---------------------------

Now the code is committed but not pushed.

To do this we need to do :

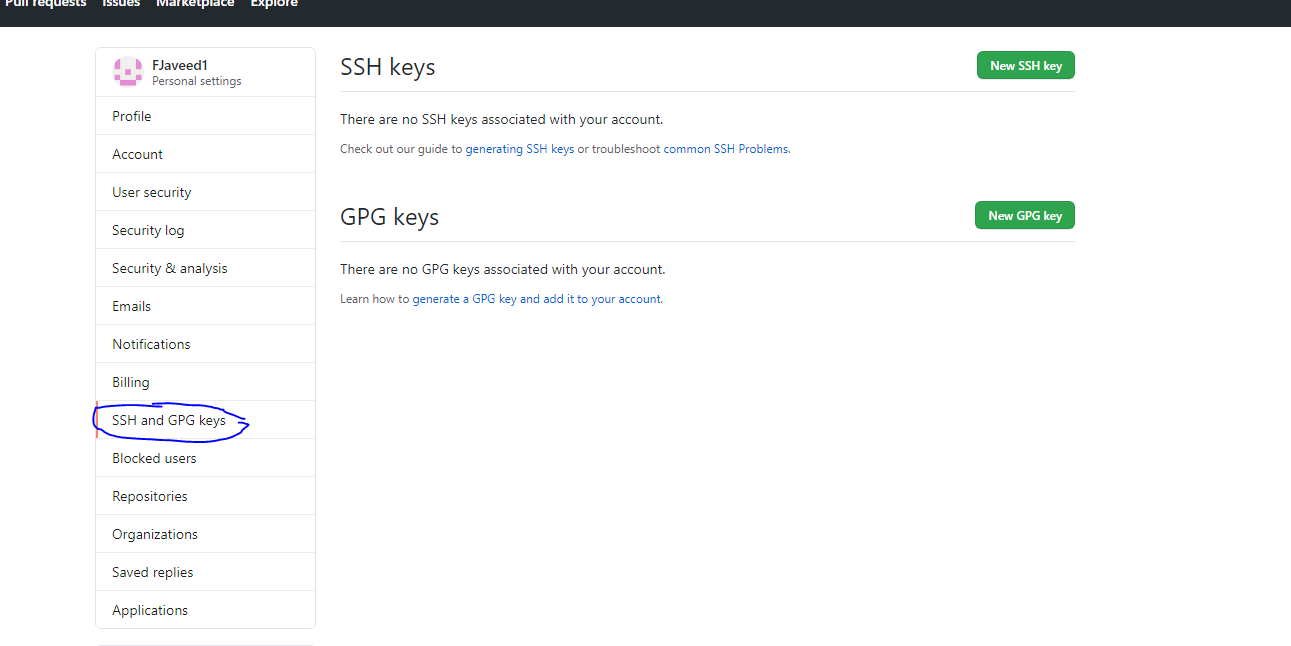
J:\MyEclipse\DemoGITRepo>git push origin master

As soon as u do this , it will give u the error : permission denied



WE get this error due to authorization as anyone can push the code if they have the login email-id.

For this we need to have the SSH or public key .Now to set up this permission, we need to go to the account settings



Unable to set the SSH key .

So its asking for the GUI password .

After entering the password and checking it gives the following :

J:\MyEclipse\DemoGITRepo>git push origin master

Enumerating objects: 43, done.

Counting objects: 100% (43/43), done.

Delta compression using up to 8 threads

Compressing objects: 100% (25/25), done.

Writing objects: 100% (43/43), 4.93 KiB | 631.00 KiB/s, done.

Total 43 (delta 3), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (3/3), done.

To https://github.com/FJaveed1/DemoRepofj.git

\* [new branch] master -> master

--------

J:\MyEclipse\DemoGITRepo>

J:\MyEclipse\DemoGITRepo>git status

On branch master

nothing to commit, working tree clean

J:\MyEclipse\DemoGITRepo>git status

On branch master

nothing to commit, working tree clean

J:\MyEclipse\DemoGITRepo>

So everything of the locval project is added now to the remote repo.

This Is how we push the first time

Now if we want to update or add anything to the project folder.

We can do that in our local directory and then check again the status in git cmd prompt:

J:\MyEclipse\DemoGITRepo>git status

On branch master

Untracked files:

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

src/main/java/com/qa/pages/SearchPage.java

target/classes/com/qa/pages/SearchPage.class

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

J:\MyEclipse\DemoGITRepo>git status

On branch master

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git restore <file>..." to discard changes in working directory)

modified: src/main/java/com/qa/pages/LoginPage.java

modified: target/classes/com/qa/pages/LoginPage.class

Untracked files:

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

src/main/java/com/qa/pages/SearchPage.java

target/classes/com/qa/pages/SearchPage.class

no changes added to commit (use "git add" and/or "git commit -a")

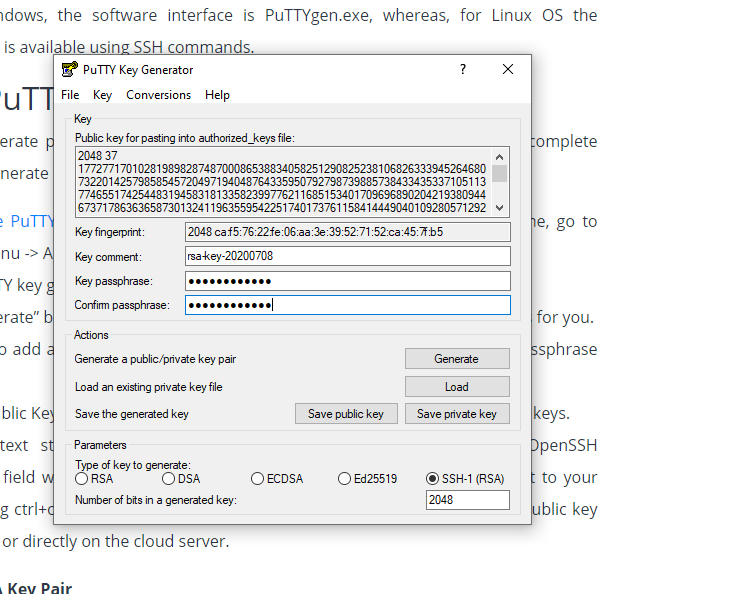
------

So here we have the unsaved uncommitted files in dir which are not pushed in remote git repo.Fo this we need to again do the same process.

Putty :

SSH;

Public key :



Key passphrase:

Ahmfoualeinu

Public Key :

2048 37 17727717010281989828748700086538834058251290825238106826333945264680732201425798585457204971940487643359507927987398857384334353371051137746551742544831945831813358239977621168515340170969689020421938094467371786363658730132411963559542251740173761158414449040109280571292396984521618282057382463564282461531807518788069821202758390717960470146308135092365676114785162485731665905196454737608183860201701903648886670775628527809064158674169589017247574077211678640184688740365375395476528163084438657949659966784295371152850629080255654741385088088070812262558708075162775982790089812780069396129977857722942413793621 rsa-key-20200708

Through GITBASH:

A@DESKTOP-JG740F6 MINGW64 /j/MyEclipse/DemoGITRepo (master)

$ ls -al

total 21

drwxr-xr-x 1 A 197121 0 Jul 8 17:45 ./

drwxr-xr-x 1 A 197121 0 Jul 8 17:07 ../

-rw-r--r-- 1 A 197121 1210 Jul 8 17:22 .classpath

drwxr-xr-x 1 A 197121 0 Jul 8 18:31 .git/

-rw-r--r-- 1 A 197121 563 Jul 8 17:07 .project

drwxr-xr-x 1 A 197121 0 Jul 8 17:07 .settings/

-rw-r--r-- 1 A 197121 3015 Jul 8 17:21 pom.xml

drwxr-xr-x 1 A 197121 0 Jul 8 17:07 src/

drwxr-xr-x 1 A 197121 0 Jul 8 17:07 target/

A@DESKTOP-JG740F6 MINGW64 /j/MyEclipse/DemoGITRepo (master)

$ ls -al ~/.ssh

ls: cannot access '/c/Users/A/.ssh': No such file or directory

A@DESKTOP-JG740F6 MINGW64 /j/MyEclipse/DemoGITRepo (master)

$ $ ssh-keygen -t rsa -b 4096 -C "fouzia01@gmail.com"

bash: $: command not found

A@DESKTOP-JG740F6 MINGW64 /j/MyEclipse/DemoGITRepo (master)

$ ssh-keygen -t rsa -b 4096 -C "fouzia01@gmail.com"

Generating public/private rsa key pair.

Enter file in which to save the key (/c/Users/A/.ssh/id\_rsa):

Created directory '/c/Users/A/.ssh'.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /c/Users/A/.ssh/id\_rsa

Your public key has been saved in /c/Users/A/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:v6F9q3hBmBvtG19O8fh0+ZNyNZrEOdmErrJMdBndzPs fouzia01@gmail.com

The key's randomart image is:

+---[RSA 4096]----+

| |

| . + |

| + . ..+ |

| + o o....|

| S\* oo ==.|

| o.= B++\*|

| .o=o++oE|

| o=o+o+.+o|

| o+\*o..o .|

+----[SHA256]-----+

A@DESKTOP-JG740F6 MINGW64 /j/MyEclipse/DemoGITRepo (master)

PULL THE CODE:

1. CLONE
2. PULL

Now when we want to pull the code from the remote repo to our local then we need to clone first and then :

Git clone <link/website name>

git clone https://github.com/naveenanimation20/SeleniumJavaCourse.git

Directory of C:\Users\A\Desktop\JavaSeleniumCourse

07/08/2020 09:04 PM <DIR> .

07/08/2020 09:04 PM <DIR> ..

0 File(s) 0 bytes

2 Dir(s) 82,435,612,672 bytes free

C:\Users\A\Desktop\JavaSeleniumCourse>git clone https://github.com/naveenanimation20/SeleniumJavaCourse.git

Cloning into 'SeleniumJavaCourse'...

remote: Enumerating objects: 243, done.

remote: Total 243 (delta 0), reused 0 (delta 0), pack-reused 243

Receiving objects: 100% (243/243), 1.68 MiB | 1.52 MiB/s, done.

Resolving deltas: 100% (72/72), done.

If anything has changed in remote repo , and we need to update in our local also .It should be just pull not clone again

git pull origin master

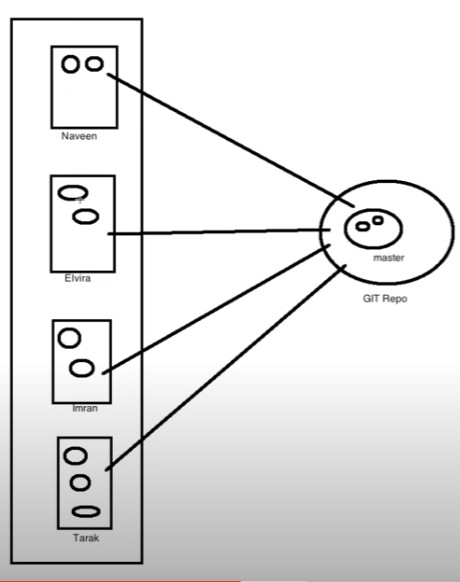
GIT PULL request:

git push origin master

<https://www.youtube.com/watch?v=iJNuZw0ur2E>

GIT Branching concept :

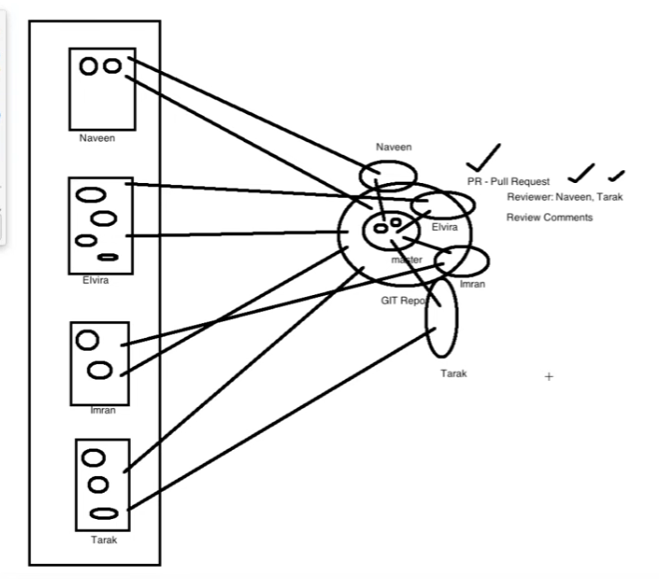
The team lead Naveen is implementing the code and pushed into the main master branch in git and giving to other team members who can pull from that.



Each team member is working on their code and creating their own branches to review their code by team lead. Every time a code is checked in there will be a PR(pull reqest) raised, also a reviewer name.

This PR and the reviewer process is called code versioning tool.

Without any code review all the process will be a buggy and not working .The reviewer is important because each individual will not find any bugs in his own code.



Now of there is another team who wants the framework from the Naveen team .They will create the following file from the master branch and give to the other team.

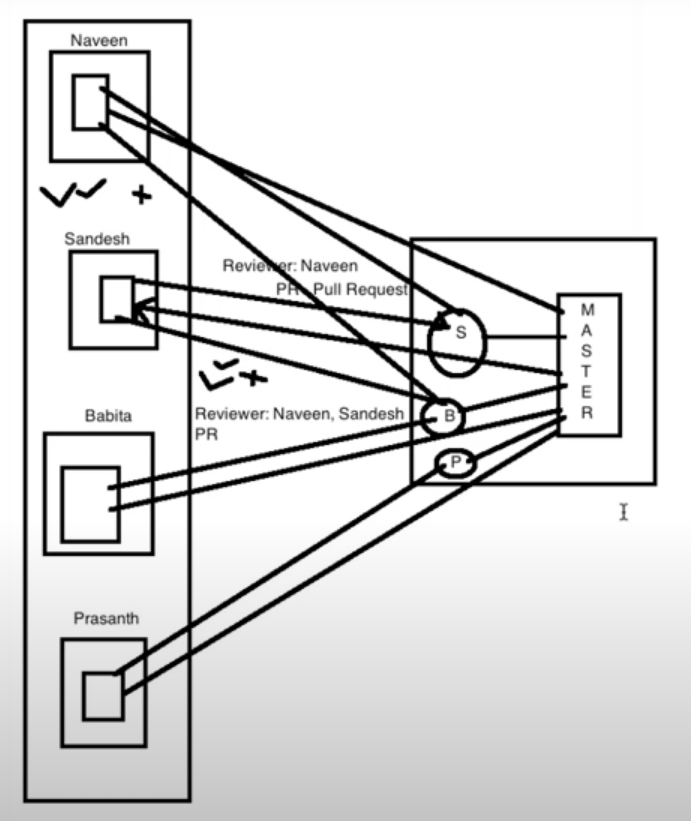
1. Jar
2. Artifact
3. Maven/gradle dependency
4. Version 1.11

# <https://www.youtube.com/watch?v=-LfyYDuryT4&list=TLPQMDgwNzIwMjCcsTr1N9jNwA&index=2>

# What is GIT PULL Request || Commit|| PUSH || PULL || CheckOut Branch || Merge || Code Review Process

When a team project starts , all the code is kept in the git repo from where all the team can asscess the project info.Now when one of the individual finishes his parft of work and wants to send his code to the repo, first he will create his own branch raising the PR(pull-request) and need to give the reviewer name. So whoever is the reviewer , he will chk the code from his branch and confirm the code to merge to the master branch.

Here master branch is the main branch where all the code should be pushed only if all reviewers agree with the code. We cannot put the buggy code into the master branch.

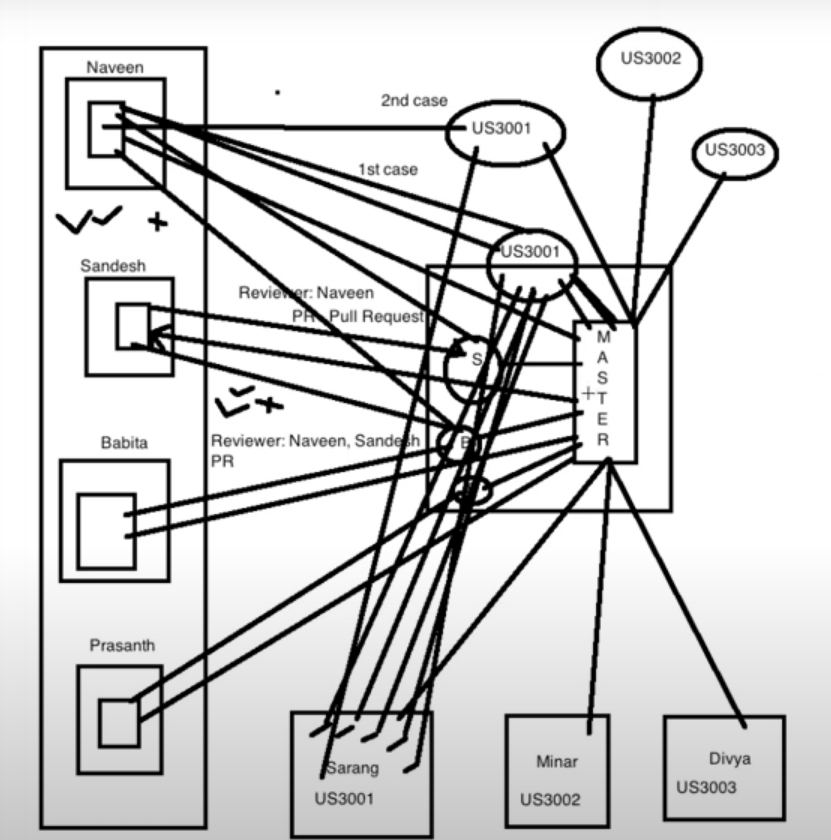


Now for the next team , some new joined Sarang, minar and divya .They all are working on the user stories of the agile US3001 , US3002, US3003.

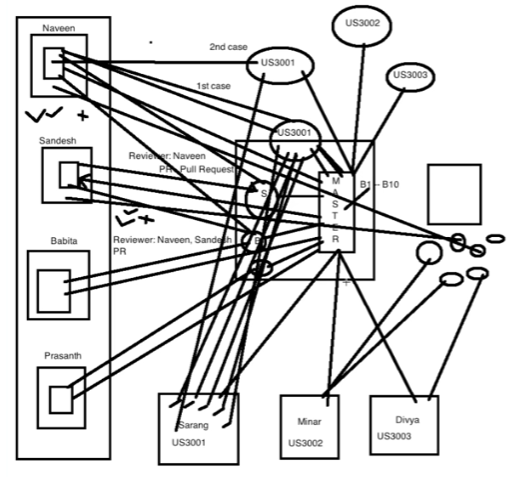
So sarang has the user story US3001 of 5 days.So each day he will raise a new PR(for 5 days) and put his code daily to review for Naveen and them merge onto the master branch.

In second case , he will raise only one PR with his 1000 lines of code to review which is very tedious kob for Naveen and then merge into master. But here the second case is not practical as there might be some imp points which will be ignored. So its always a good practice to merge small codes daily and raise small PR’s to work and then merge to the master.

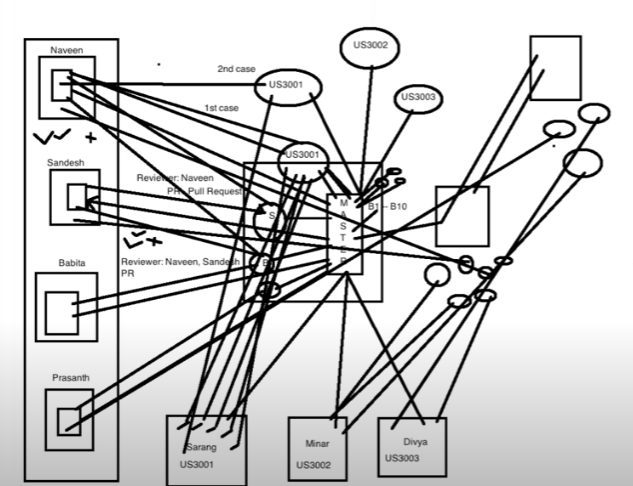
If anyone is reviewing the code, It’s a good practice to take the code, compile it and execute ion their own machine and then put it in the master branch. But many times (almost 80%) the reviewer just checks the code directly in the git repo, modifies and checks and then give review there itself without executing the code.



After the sprint is completed.This is the development phase where development team has completed the code . Now after all the user stories are created by all the team mates in a team and reviewed, the code is finally merged to the main branch and then given to the QA team for testing further. When they test and raises the bugs like B1-B10 .Here the same process will be created for the QA team as well .For the bugs,B1-B10 , ten branches will be created and 10 PRS will be rasised then its again given back to the development team to fix the bugs nd merged to the master branch with fixed bugs, and then return to the QA team again for testing. This process will be same as before. The concept is that whenever there is a problem/update in the code , we need to raise a PR and work on the code.

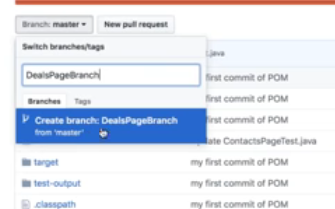


After the QA team tests and everything Is confirmed , then same master branch is passed to the production team. Now the production is tested by the users and there will be production issue. Again the bugs will be raised and given back to the development team where they again raise the PRS into the master branch .They work on the PRS and then give back to QA team where they test and merge to the main branch and give to the production with error free branch.

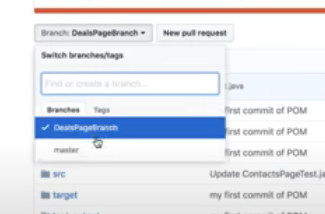


Creating the branch:

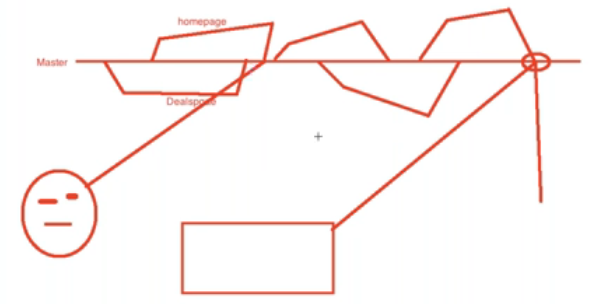
We can directly create the branch in the git hub where there is mater branch tab.



Now there are two branches : master and dealsPage



As soon as we create the new branch , the entire code present in master branch will be merged with the new DealsPage branch .

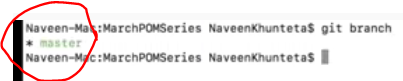


This concept is creating and merging the branch.

Now when we want to modify the code from the DealsPage bramch , first we need to go to the DealsPage branch in eclipse .For this we need to do the following

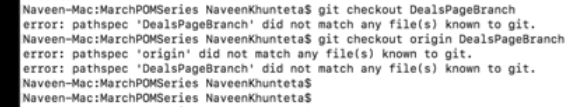
git checkout branch.

For this first we need to check the branch by the following command:

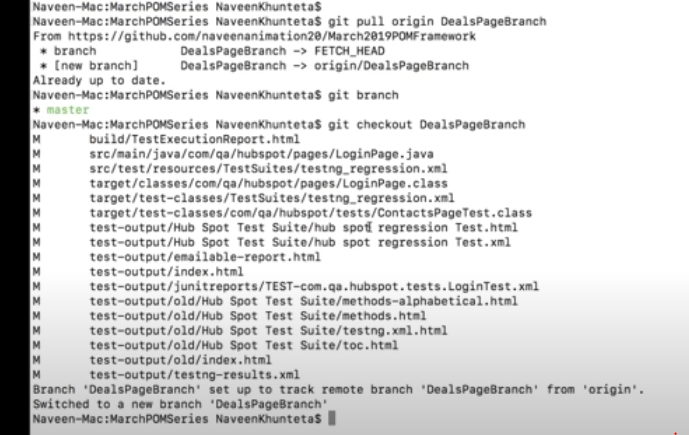


Here its pointing to the master branch .

Now if we give the command git checkout DelasPageBranch, it gives the following error.



So before this command we need to do the pull for DelasPageBranch and then the chkout as below :



Now its pointing to the DelasPageBranch:



Ex:

Cloning and pulling and switching to other branches:

D:\GitTest>git clone https://github.com/FJaveed1/DemoRepofj

Cloning into 'DemoRepofj'...

remote: Enumerating objects: 60, done.

remote: Counting objects: 100% (60/60), done.

remote: Compressing objects: 100% (30/30), done.

remote: Total 60 (delta 7), reused 60 (delta 7), pack-reused 0

Unpacking objects: 100% (60/60), 6.28 KiB | 27.00 KiB/s, done.

D:\GitTest>dir

Volume in drive D is New Volume

Volume Serial Number is 34D2-C33E

Directory of D:\GitTest

07/10/2020 08:57 AM <DIR> .

07/10/2020 08:57 AM <DIR> ..

07/10/2020 08:57 AM <DIR> DemoRepofj

0 File(s) 0 bytes

3 Dir(s) 100,886,335,488 bytes free

After modifying the code in the git repo , we are using the pull request so that it updates in the local too.

D:\GitTest\DemoRepofj>git pull origin master

remote: Enumerating objects: 17, done.

remote: Counting objects: 100% (17/17), done.

remote: Compressing objects: 100% (5/5), done.

remote: Total 9 (delta 3), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (9/9), 1017 bytes | 21.00 KiB/s, done.

From https://github.com/FJaveed1/DemoRepofj

\* branch master -> FETCH\_HEAD

a224064..5064c69 master -> origin/master

Updating a224064..5064c69

Fast-forward

src/main/java/com/qa/util/TestUtil.java | 5 +++++

1 file changed, 5 insertions(+)

Second modificarion :

D:\GitTest\DemoRepofj>git pull origin master

remote: Enumerating objects: 28, done.

remote: Counting objects: 100% (28/28), done.

remote: Compressing objects: 100% (11/11), done.

remote: Total 18 (delta 5), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (18/18), 2.06 KiB | 25.00 KiB/s, done.

From https://github.com/FJaveed1/DemoRepofj

\* branch master -> FETCH\_HEAD

5064c69..b6a88e4 master -> origin/master

Updating 5064c69..b6a88e4

Fast-forward

src/main/java/com/qa/pages/LoginPage.java | 3 +++

src/main/java/com/qa/util/TestUtil.java | 1 +

2 files changed, 4 insertions(+)

Now creating new branch in the git repo.

And then switching it in local

D:\GitTest\DemoRepofj>git branch

\* master

D:\GitTest\DemoRepofj>git checkout MyOtherBranch

error: pathspec 'MyOtherBranch' did not match any file(s) known to git

D:\GitTest\DemoRepofj>git pull origin MyOtherBranch

From https://github.com/FJaveed1/DemoRepofj

\* branch MyOtherBranch -> FETCH\_HEAD

\* [new branch] MyOtherBranch -> origin/MyOtherBranch

Already up to date.

D:\GitTest\DemoRepofj>git branch

\* master

D:\GitTest\DemoRepofj>git checkout MyOtherBranch

Switched to a new branch 'MyOtherBranch'

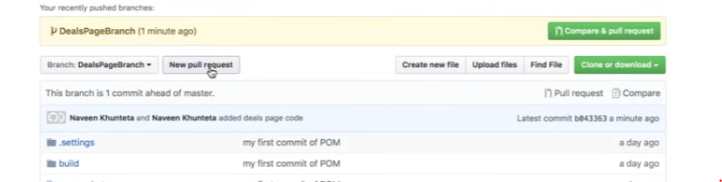
M target/classes/com/qa/pages/LoginPage.class

M target/classes/com/qa/util/TestUtil.class

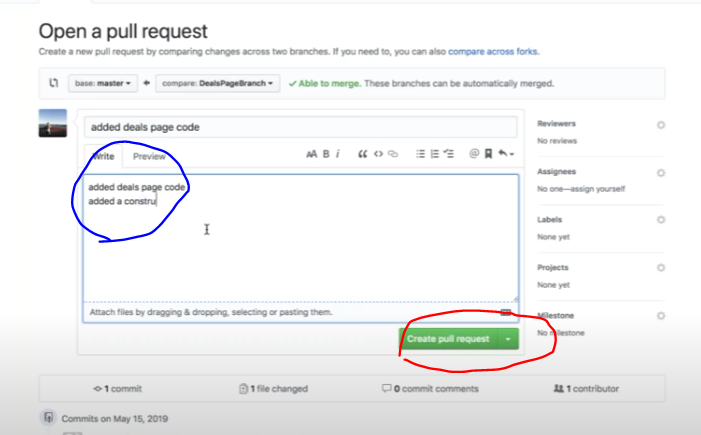
Branch 'MyOtherBranch' set up to track remote branch 'MyOtherBranch' from 'origin'.

When we create another branch , it gets updated only in that branch but not in themaster brach.

Now to merge this branch code to them aster branch , we need to do the pull request

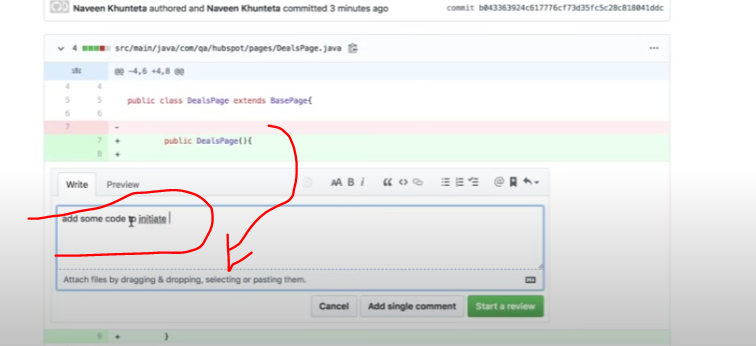


Where we can write the commnet to the reviewer to view the commnets :



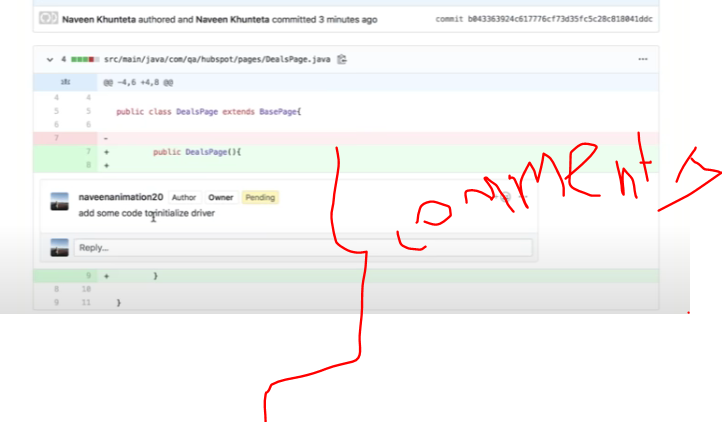
As soon as we create the pull request it will be notified to the reviewer which is added.

And then the reviewer will review the code and add a comment to improve the code or not

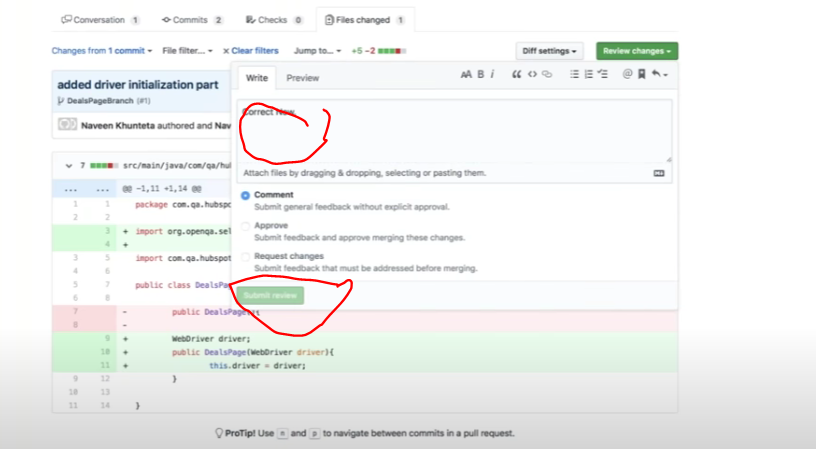


We will get the commnest for our coide like the below:

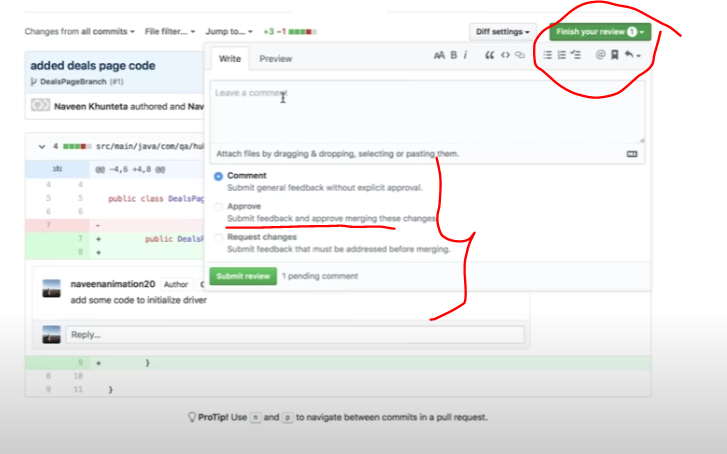
We need to make sure we don’t get many commnet else our code is not good



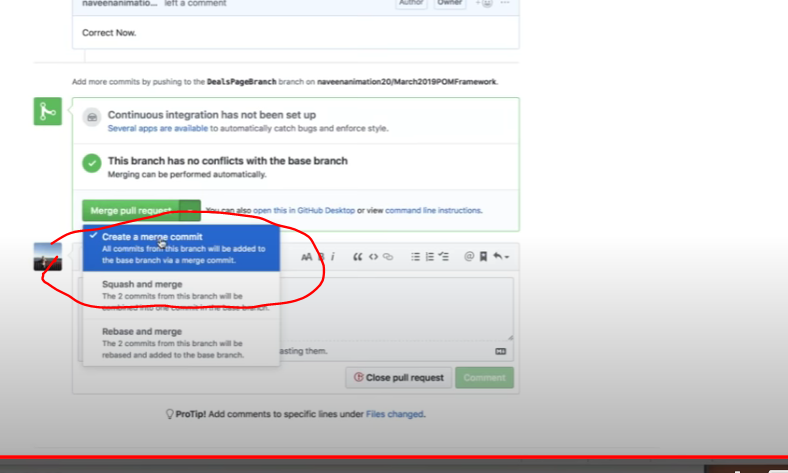
After uodating the code in eclipse , and the changes are made.



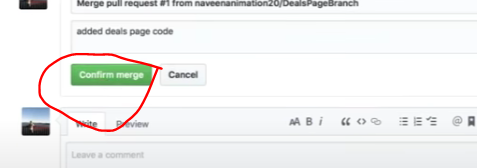
The reviewer will finish the review once the code is completed and then approve



After approving then the person is notified to merge :



Merge and confirm



After merging and confirming the master branch will updated to the changes which were in other branch.