Github Usage using Eclipse

(Author: Tapas Chakraborty)

To make it more collaborative development work, EZTO development team has decided to go for pull request and gets it reviewed by peer developer or technical lead or director of EZTO. This all can be achieved by command line tool. I came up with one approach where you can avoid command line pain and can complete all the works which is very much necessary for development work. In this document I have clearly explained the different processes (like Create a New Branch, Checkout, Rebase, Squash multiple commits) using eclipse plugin.

Eclipse plugin

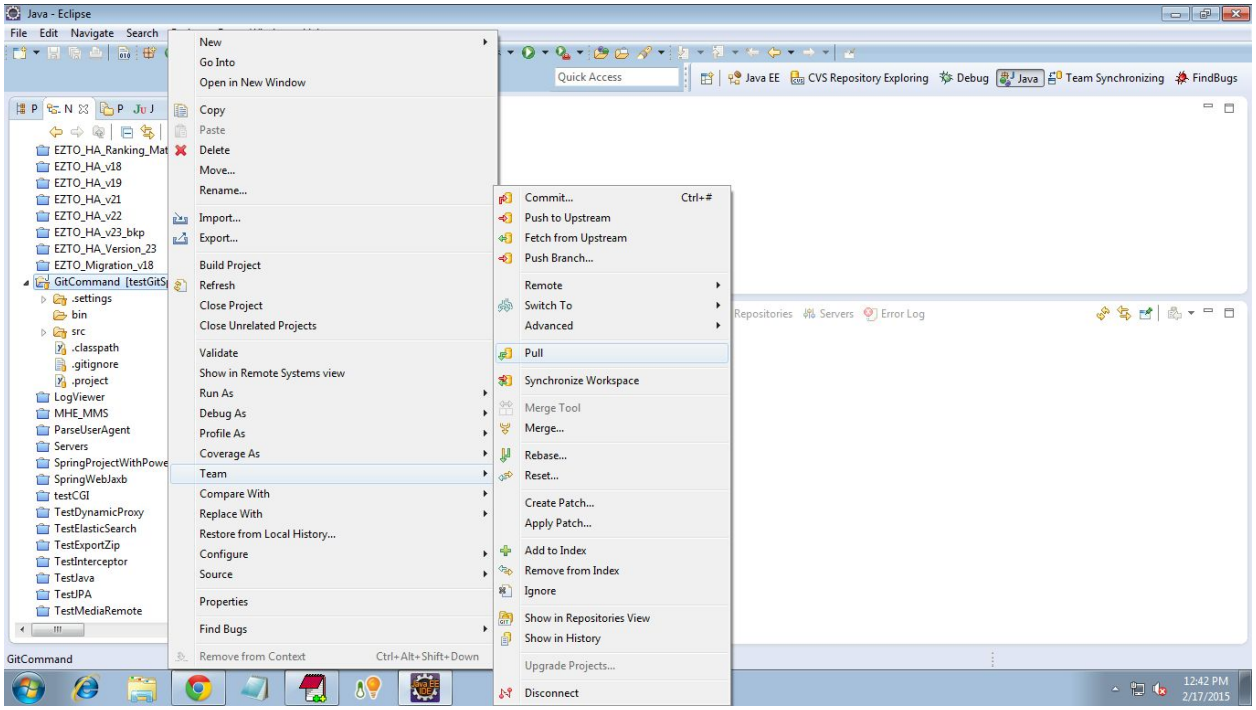
This document is meant to address how can we work with github with the help of eclipse plugin.

1. Download the github plugin from http://download.eclipse.org/egit/updates

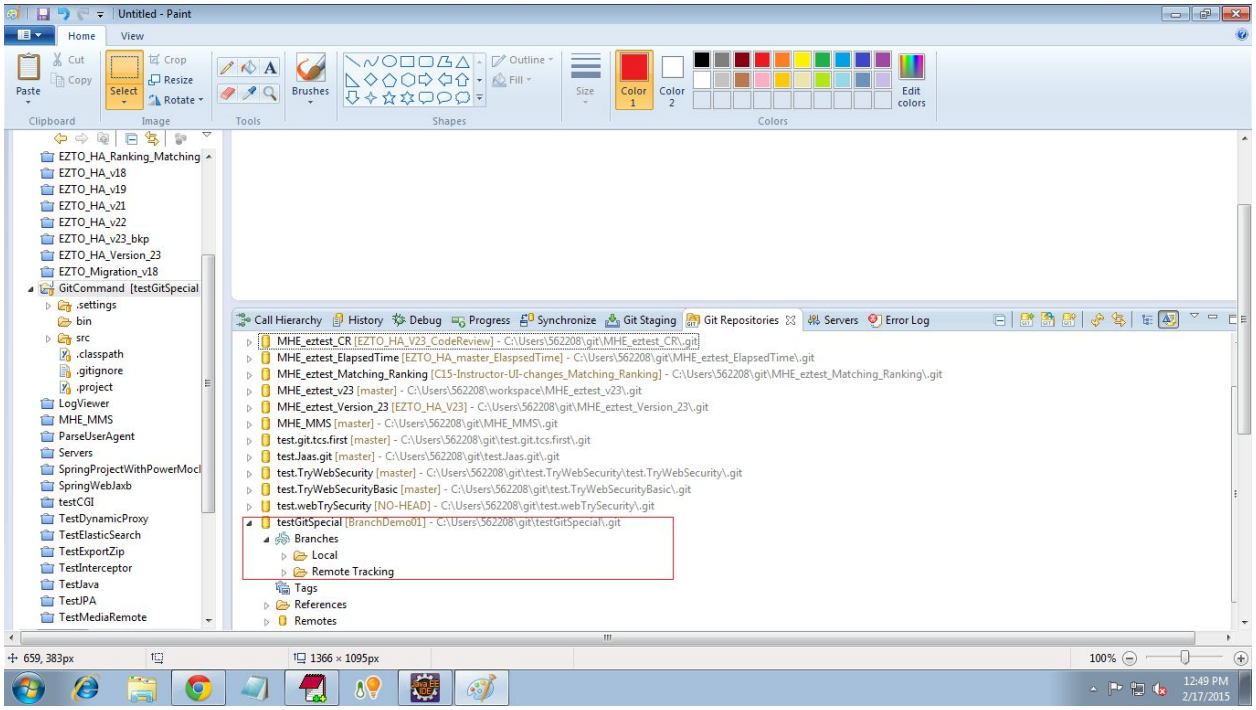
Code development

1. Get the master branch (Or related branch) from github to eclipse.

2. Before start working on your changes you should get latest code from your main branch that could be master or other relevant branch.
To get the latest code, Right click on the project => team => pull

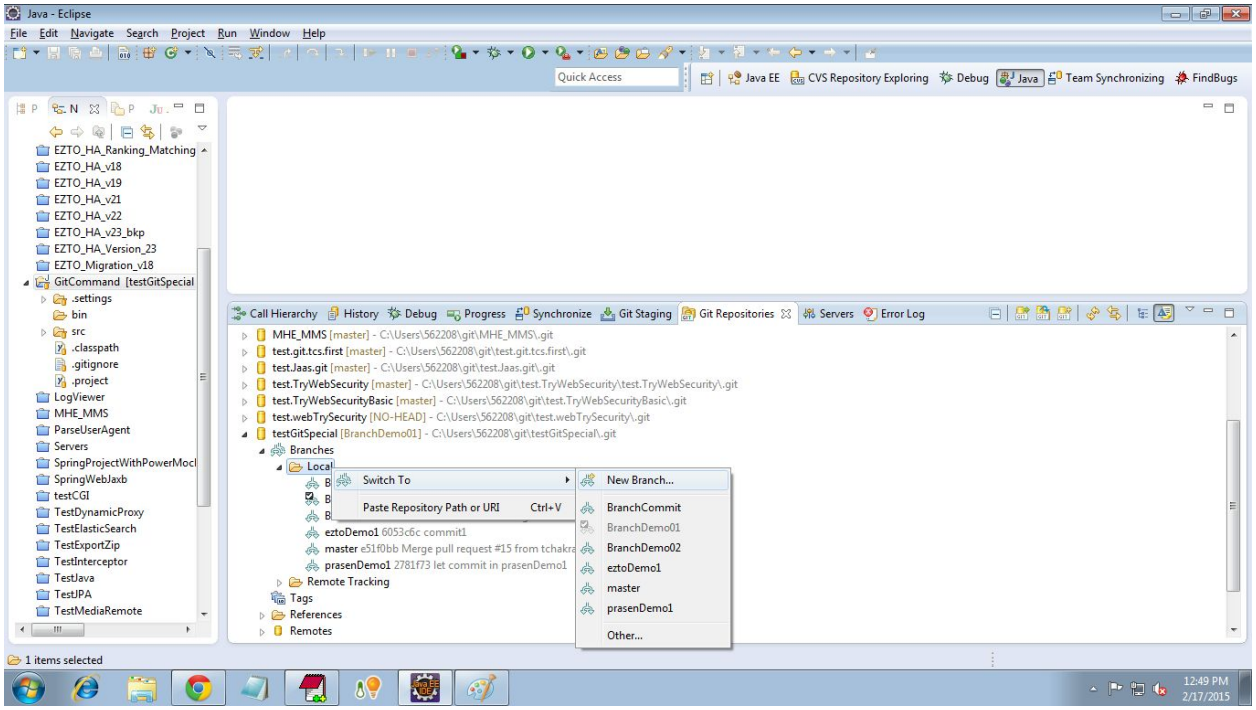

Java - Eclipse
File Edit Navigate Search
Quick Access
Java EE
CVS Repository Exploring
Debug
Java
Team Synchronizing
FindBugs
EZTO_HA_Ranking_Mat
EZTO_HA_v18
EZTO_HA_v19
EZTO_HA_v21
EZTO_HA_v22
EZTO_HA_v23_bkp
EZTO_HA_Version_23
EZTO_Migration_v18
GitCommand [testGitS
.settings
bin
src
.classpath
.gitignore
.project
LogViewer
MHE_MMS
ParseUserAgent
Servers
SpringProjectWithPowe
SpringWebJaxb
testCGI
TestDynamicProxy
TestElasticSearch
TestExportZip
TestInterceptor
TestJava
TestJPA
TestMediaRemote
GitCommand
New
Go Into
Open in New Window
Copy
Paste
Delete
Move...
Rename...
Import...
Export...
Build Project
Refresh
Close Project
Close Unrelated Projects
Validate
Show in Remote Systems view
Run As
Debug As
Profile As
Coverage As
Team
Compare With
Replace With
Restore from Local History...
Configure
Source
Properties
Find Bugs
Remove from Context
Ctrl+Alt+Shift+Down
Commit...
Ctrl+#
Push to Upstream
Fetch from Upstream
Push Branch...
Remote
Switch To
Advanced
Pull
Synchronize Workspace
Merge Tool
Merge...
Rebase...
Reset...
Create Patch...
Apply Patch...
Add to Index
Remove from Index
Ignore
Show in Repositories View
Show in History
Upgrade Projects...
Disconnect
Repositories
Servers
Error Log
12:42 PM
2/17/2015

3. Open Git Repository in Eclipse. Here, under Branches you can see Local and Remote

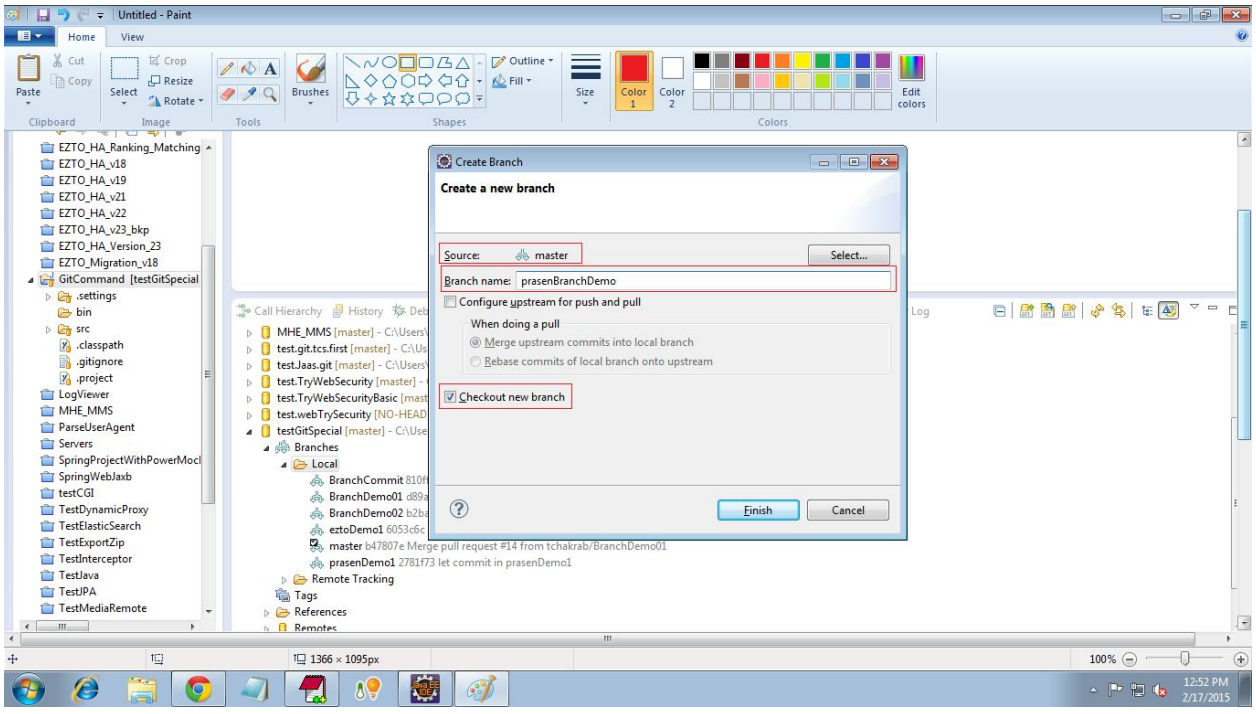


New Branch

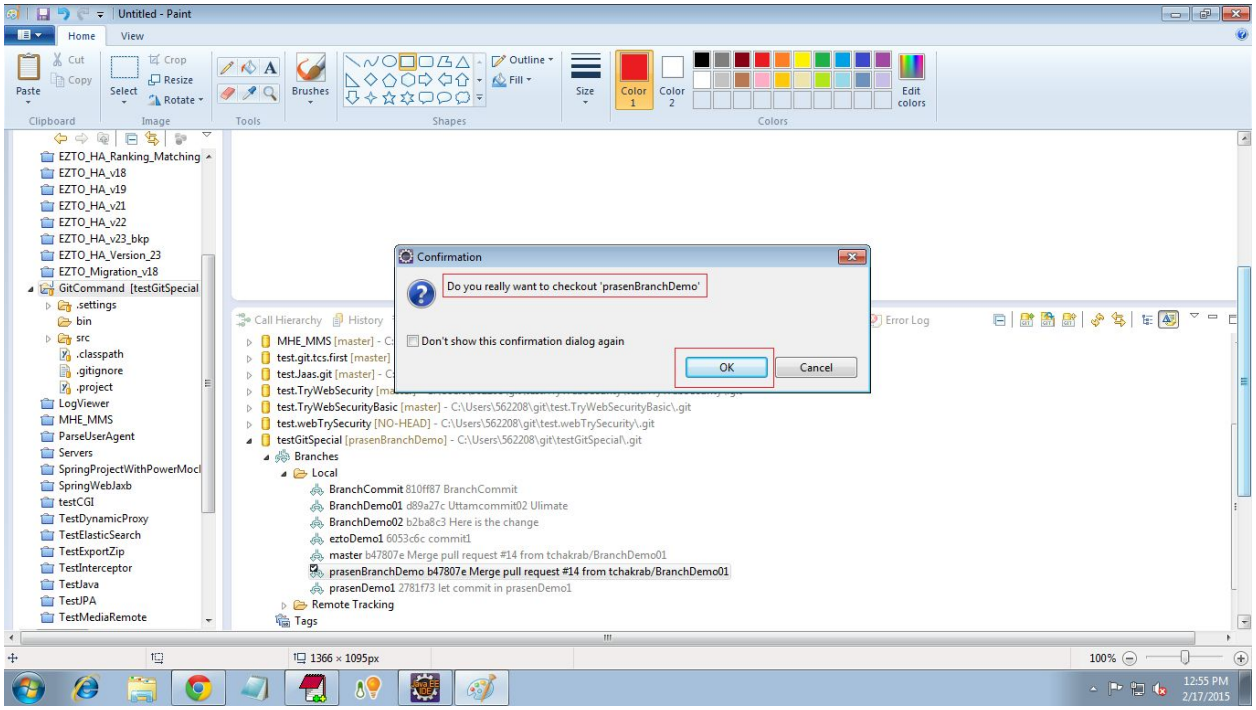
Now, Create a new Branch on your local git repository. Right click Local => Switch To => New Branch



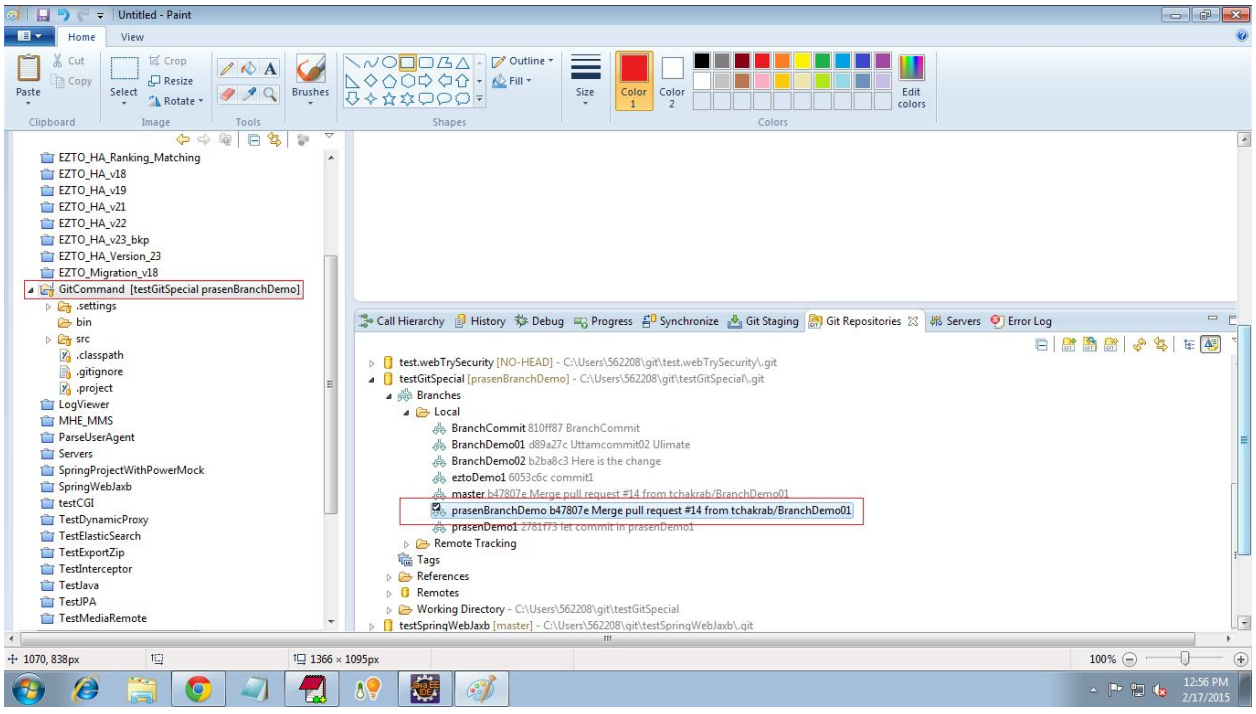
This will open a dialogue box, provide the branch name. Here “checkout new branch” check box should be checked. Then go for finish. This dialogue box also show from which branch you are creating the new branch in source tag. for example here this new branch is getting created from master



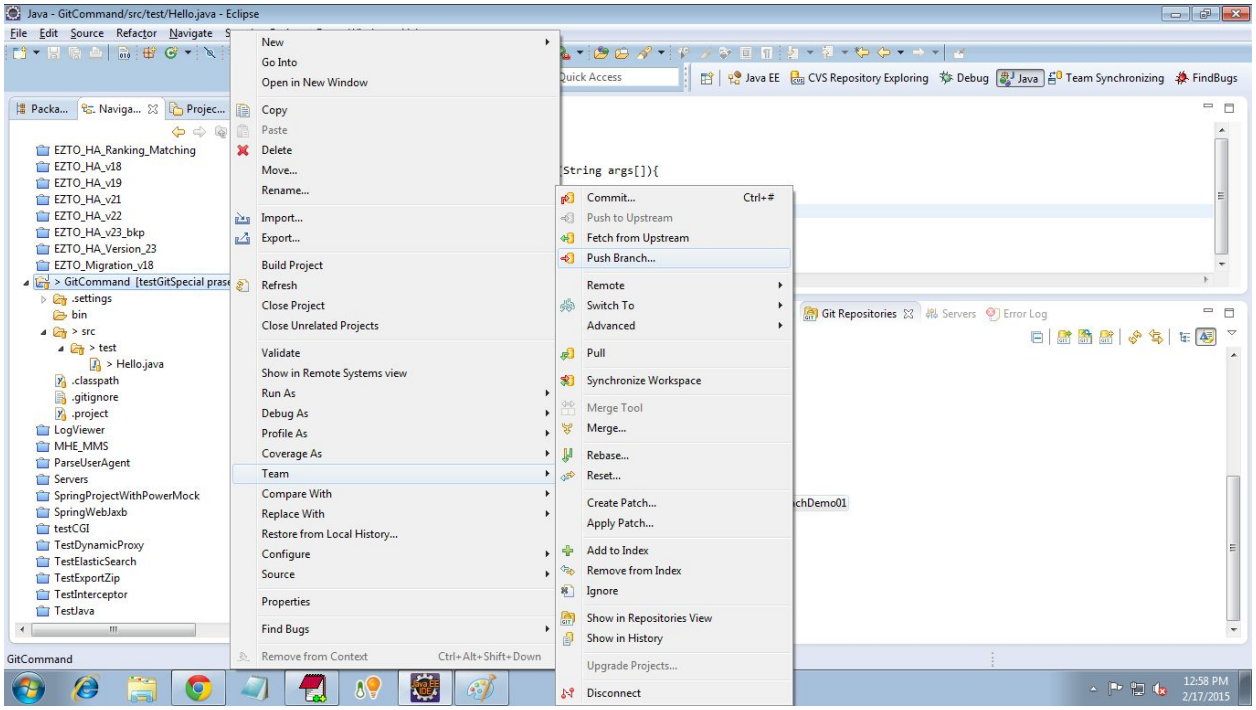
At this point, new branch has been created. Double click on the newly created branch on the localgit repository. It will ask for “Do you really want to checkout '<your branch name>'”. Click OK.This will set your eclipse branch to your newly created local branch



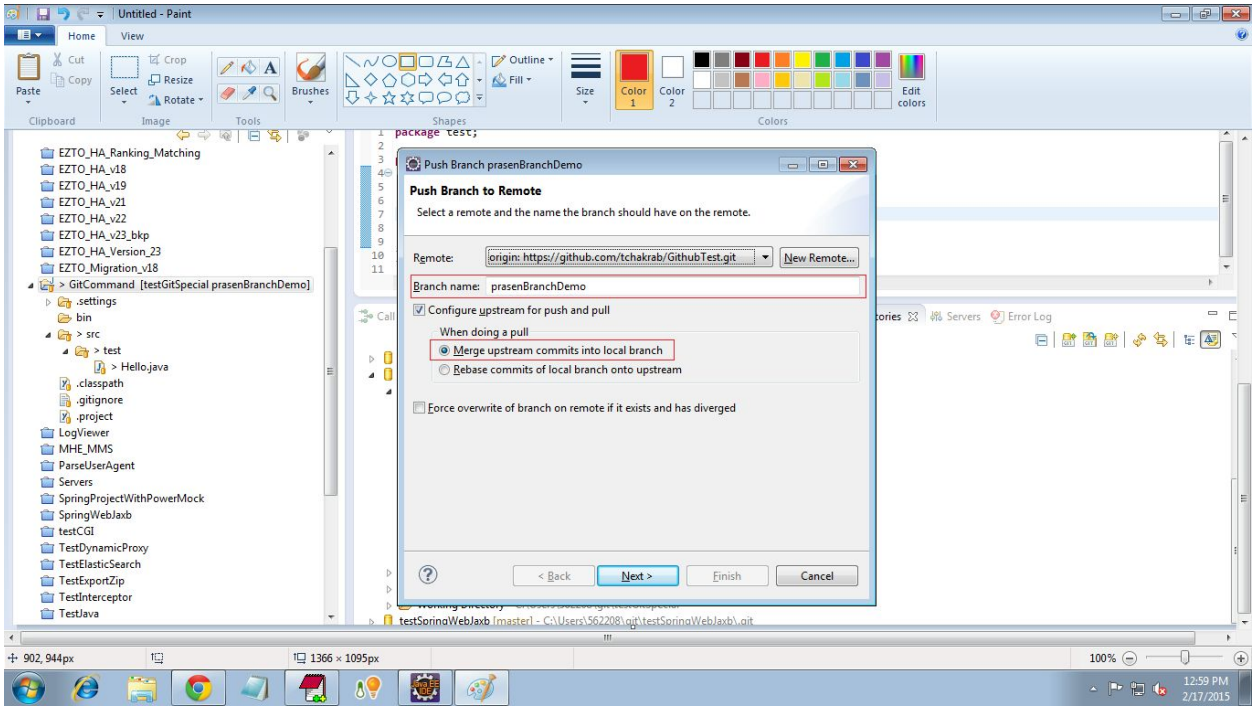
You can view your branch in your local repository also.



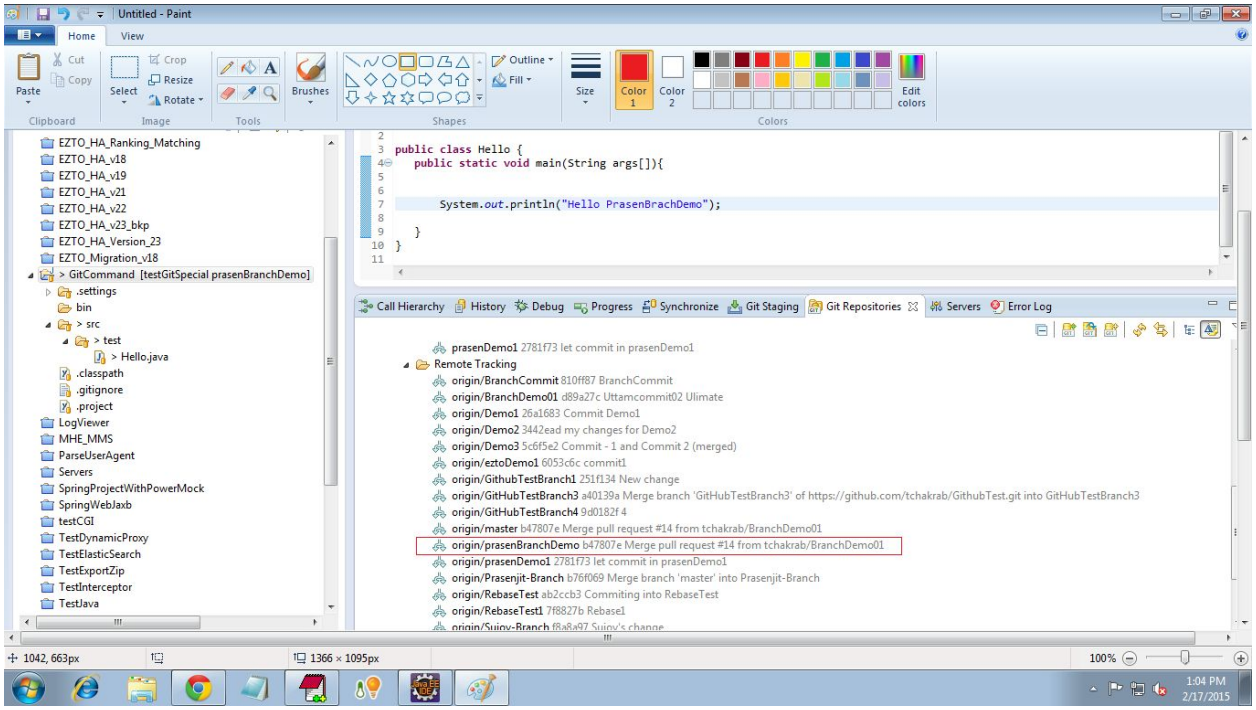
Now, you can go for your changes. That will be done only in your local branch. When you are done with your changes, push the branch to remote git repository.
To push the Branch to remote repository, Right click on the project => team => Push Branch



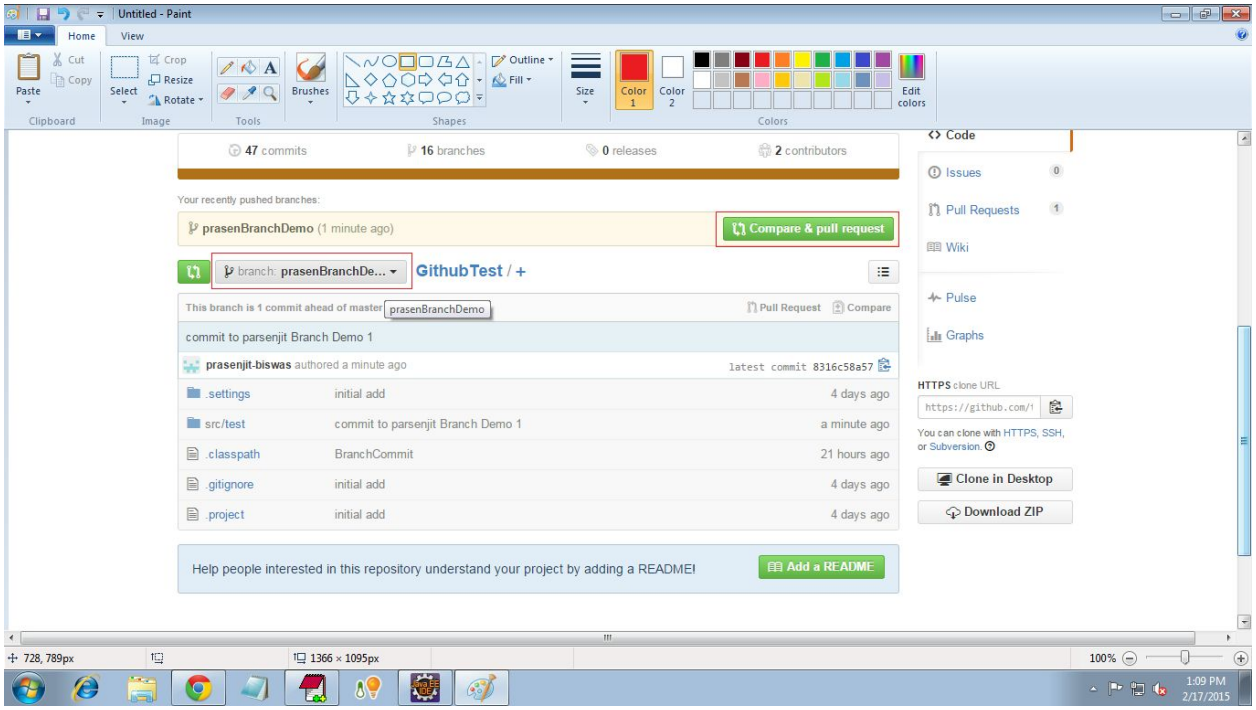
It will open a dialogue box, you can see your branch name in Text field. Click next. Here Merge upstream commit option should be selected.



Once you are done with pushing your branch successfully in remote repository, you can view the branch in remote repository in eclipse.

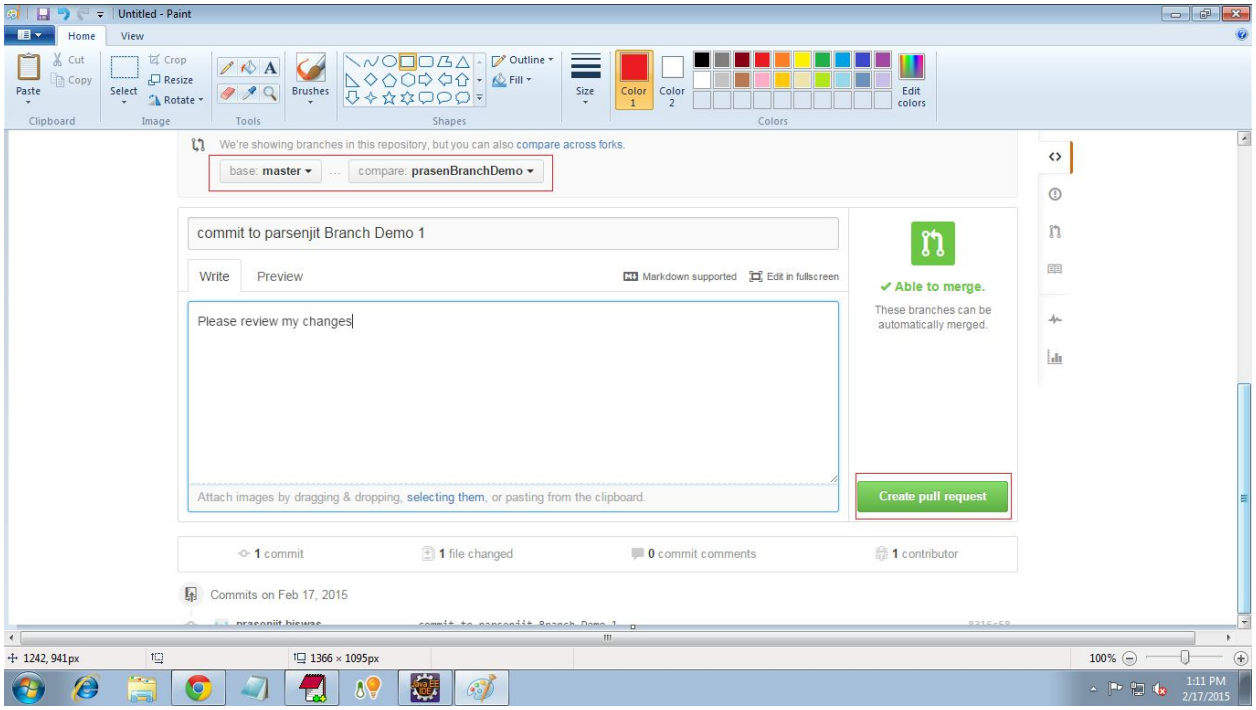


In Git User interface you can also view your Branch.

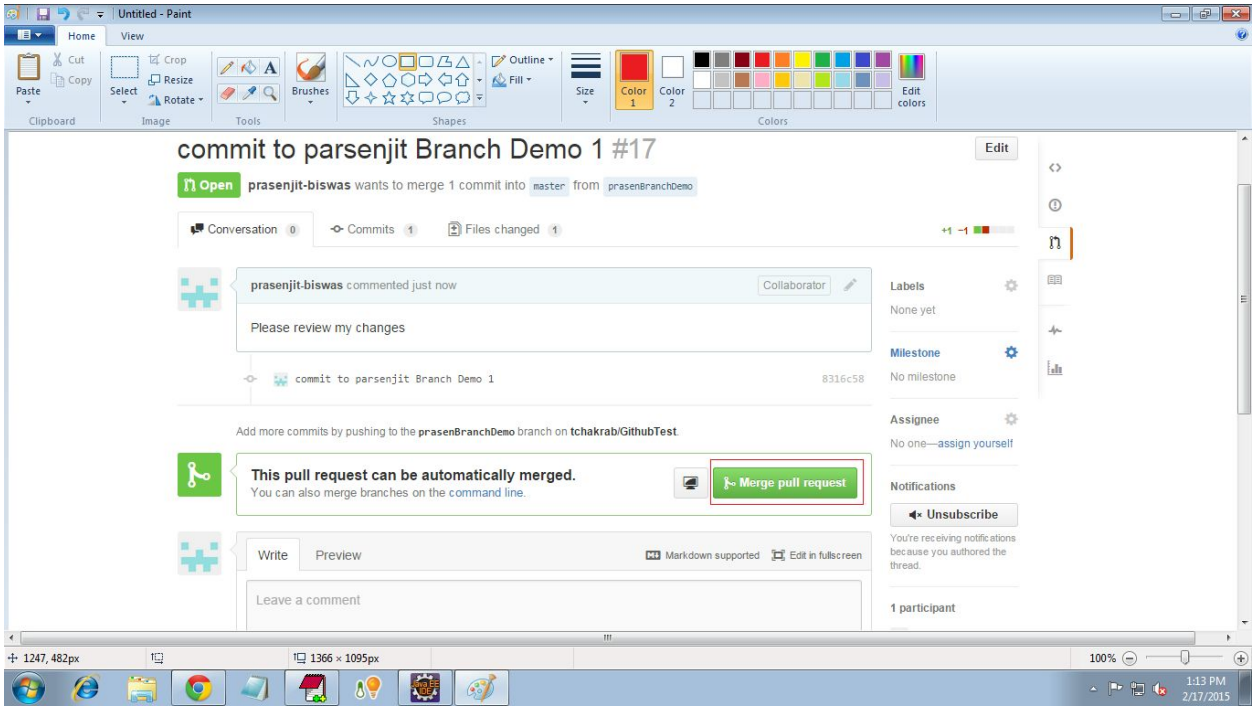


Pull Request

Now, it is time to do a pull request. Click on the “compare & pull request” button shown in previous screen. In the next screen, click on the “Create pull request” button. This screen also shows from which branch we are raising the PR.

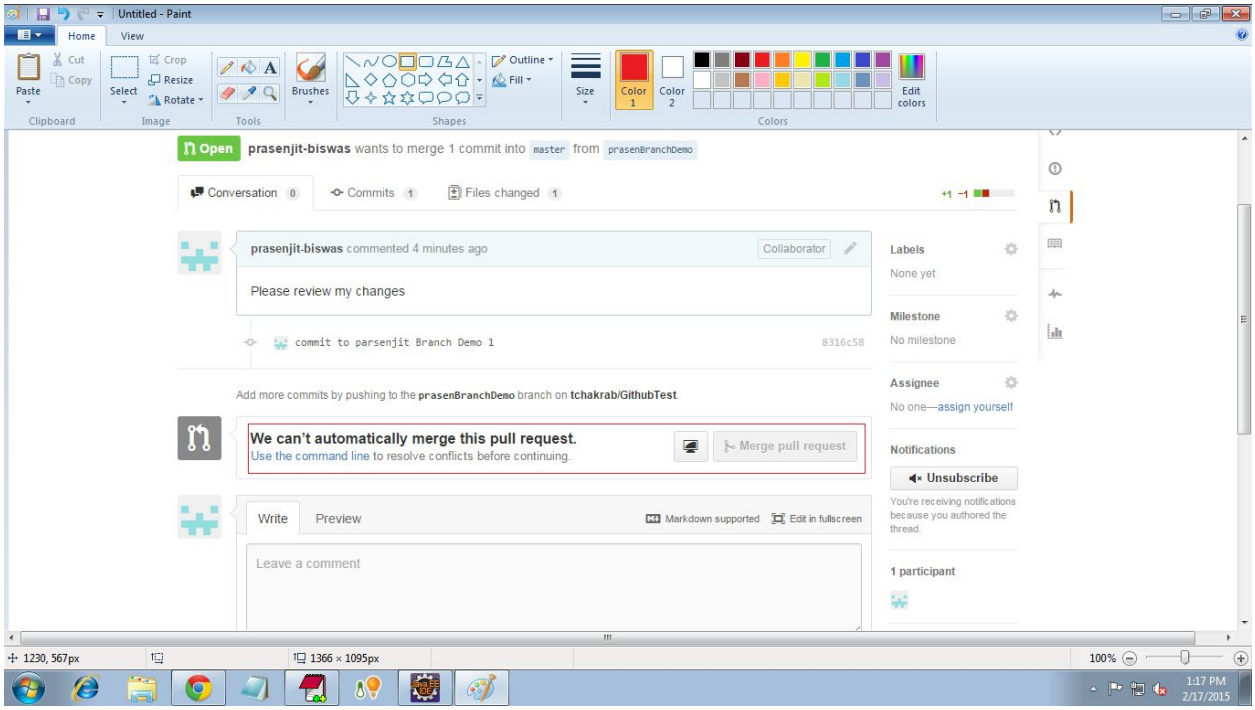


This will create a Pull request. In the pull request screen you can see “Merge pull request” button.If you can see this button in green color, it means the changes what you have done can be merged to the main branch automatically. But do not Merge your change by clicking this button.

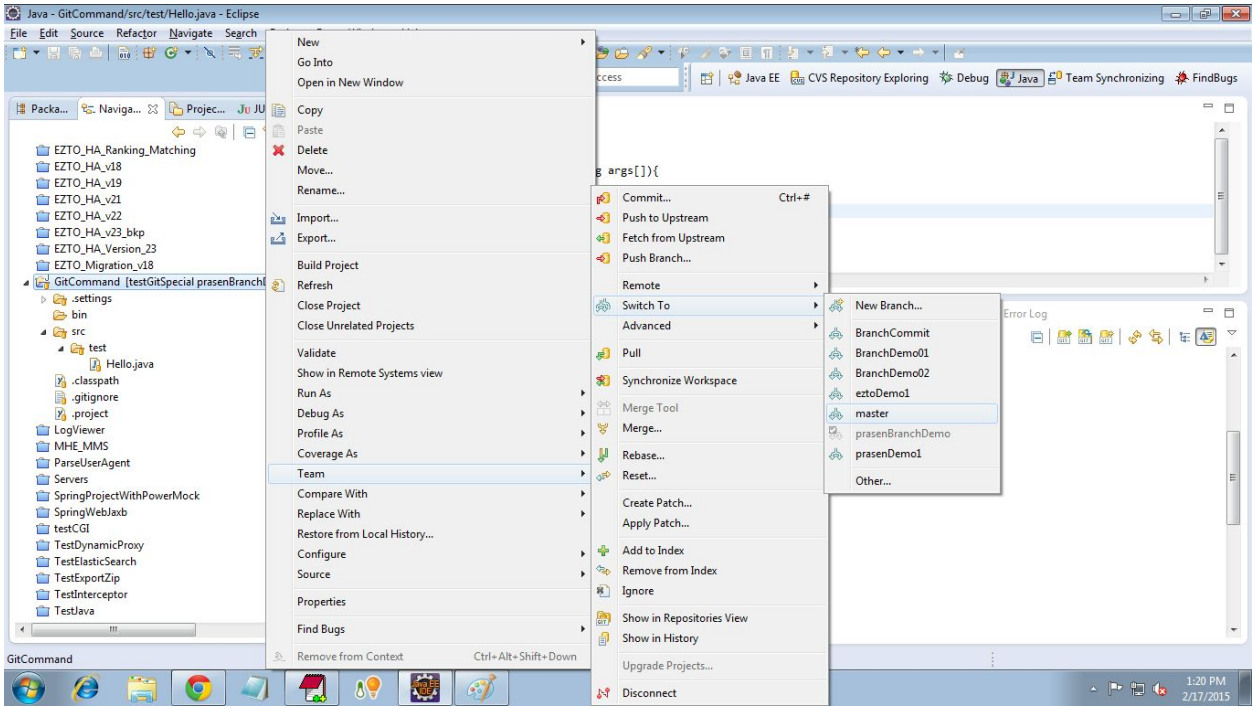


Rebase

Now we will deal with the case where your changes can not be merged automatically. It means it has some code conflict.

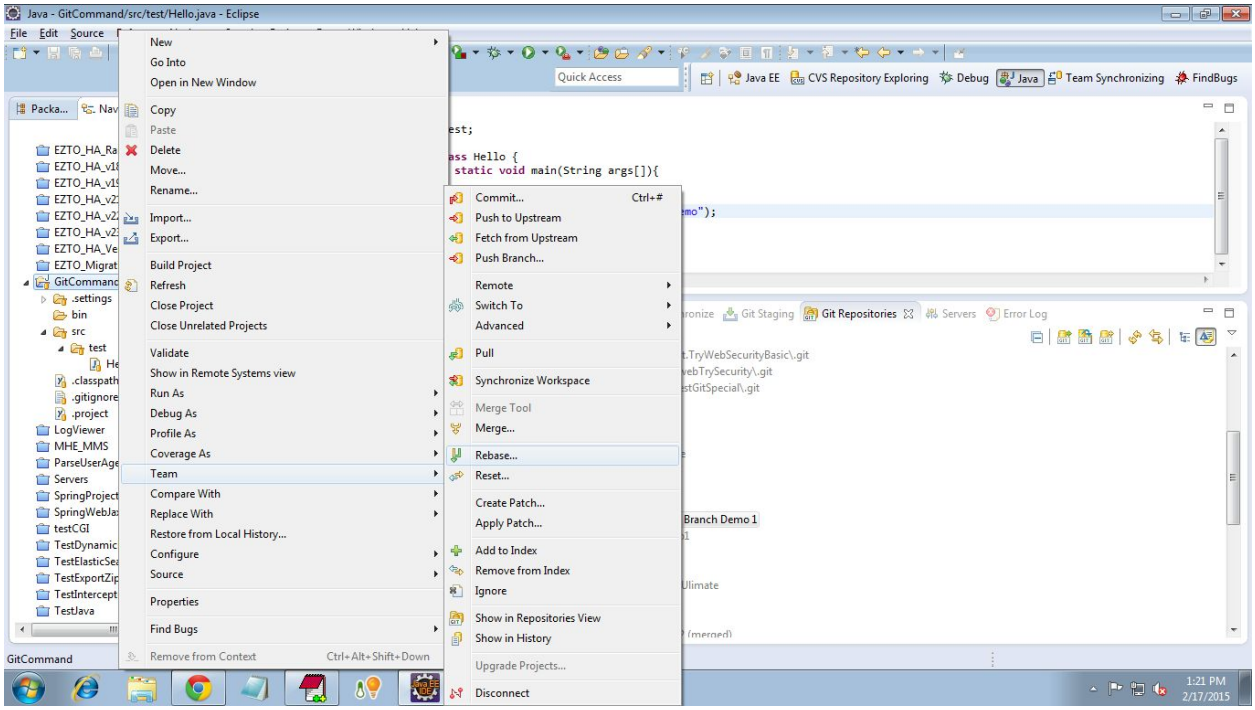


i. Switch you project to main branch where you would like to merge your code.
Right click on the project => team => Switch To => select your main branch

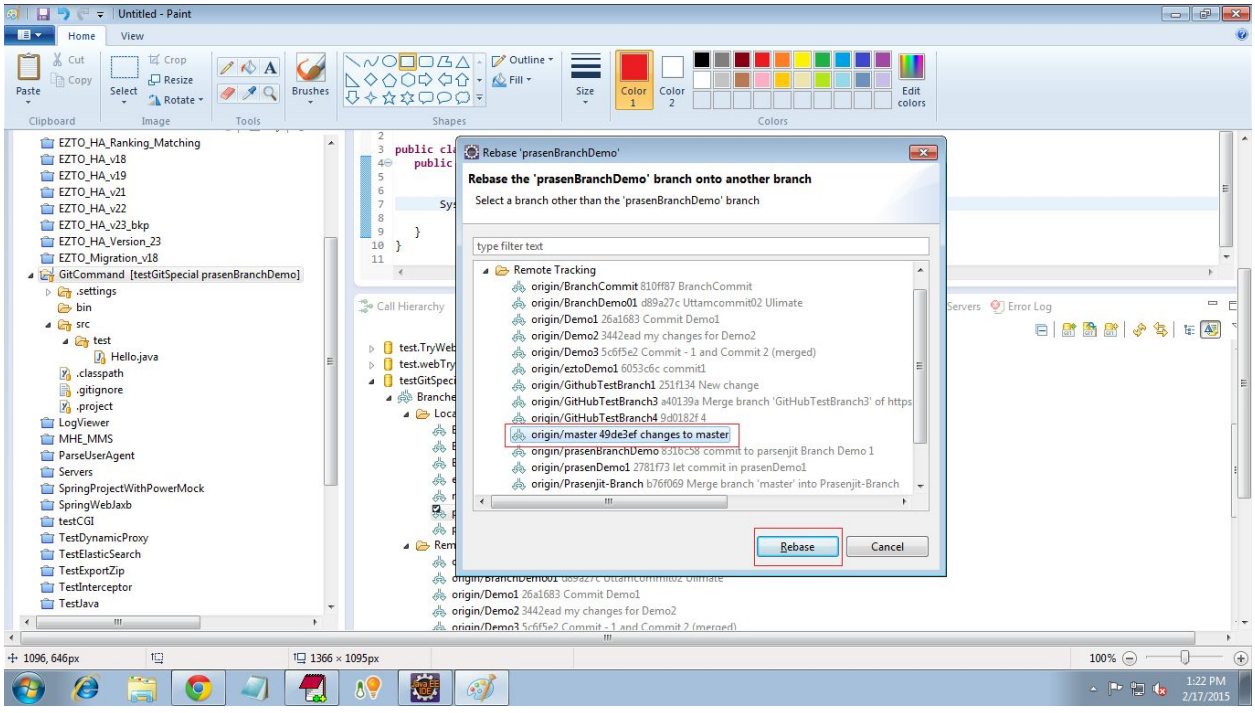


ii. Get the latest code from main branch by pulling the code from git
iii. Switch Back to your Branch again.

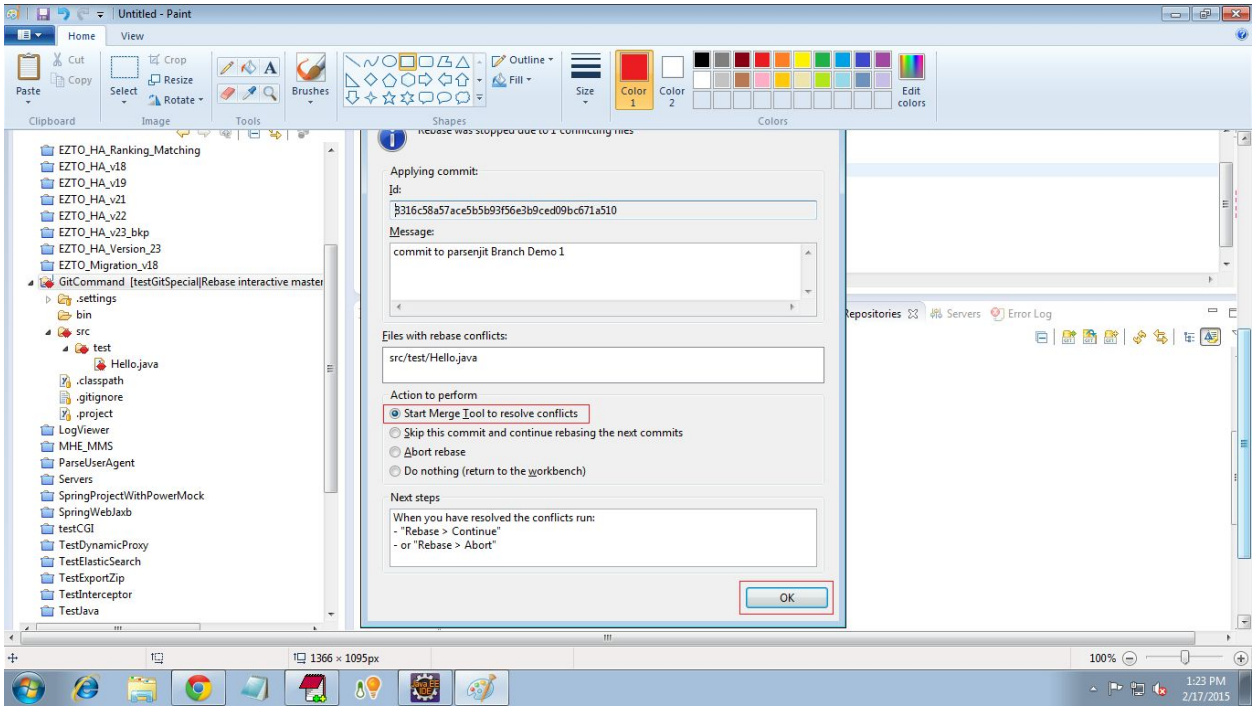
IV. Now, go to rebase option. Right click on the project => team => Rebase



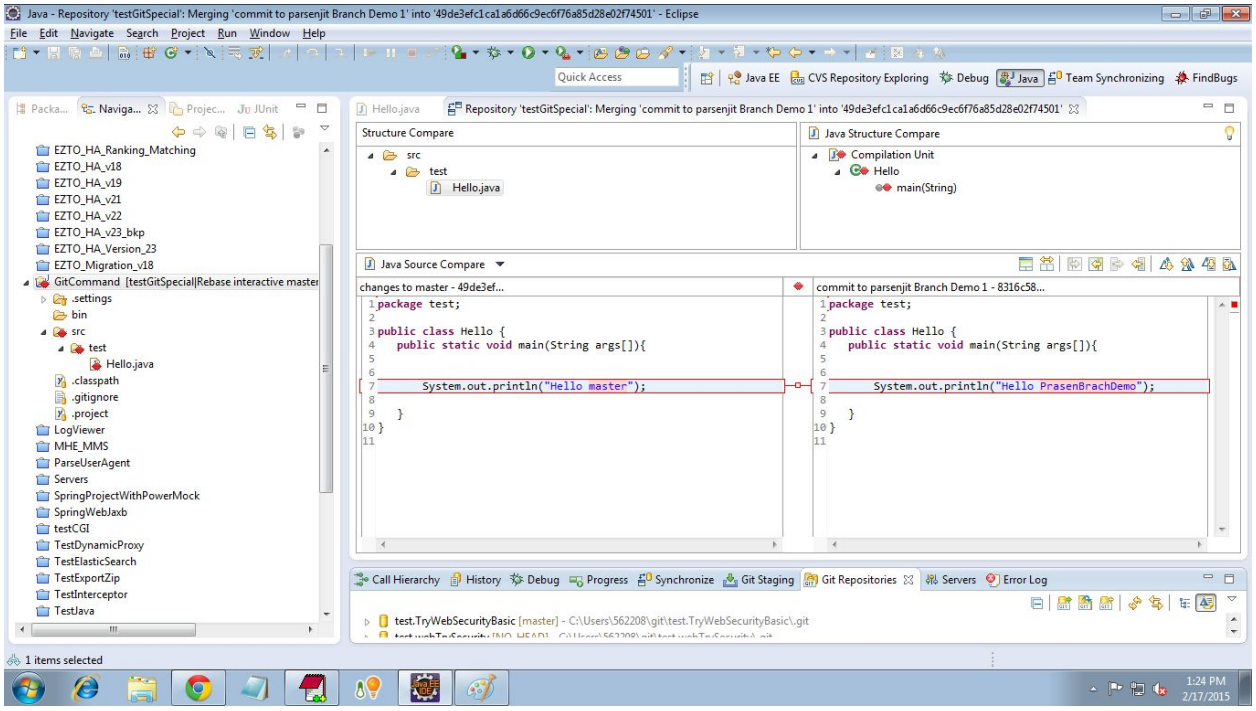
iv. This will open a dialogue box, select the main branch from the remote repository. And click on Rebase button.



V. Clicking on the Rebase button will open another window, By clicking ok here it will open Eclipse merge tool

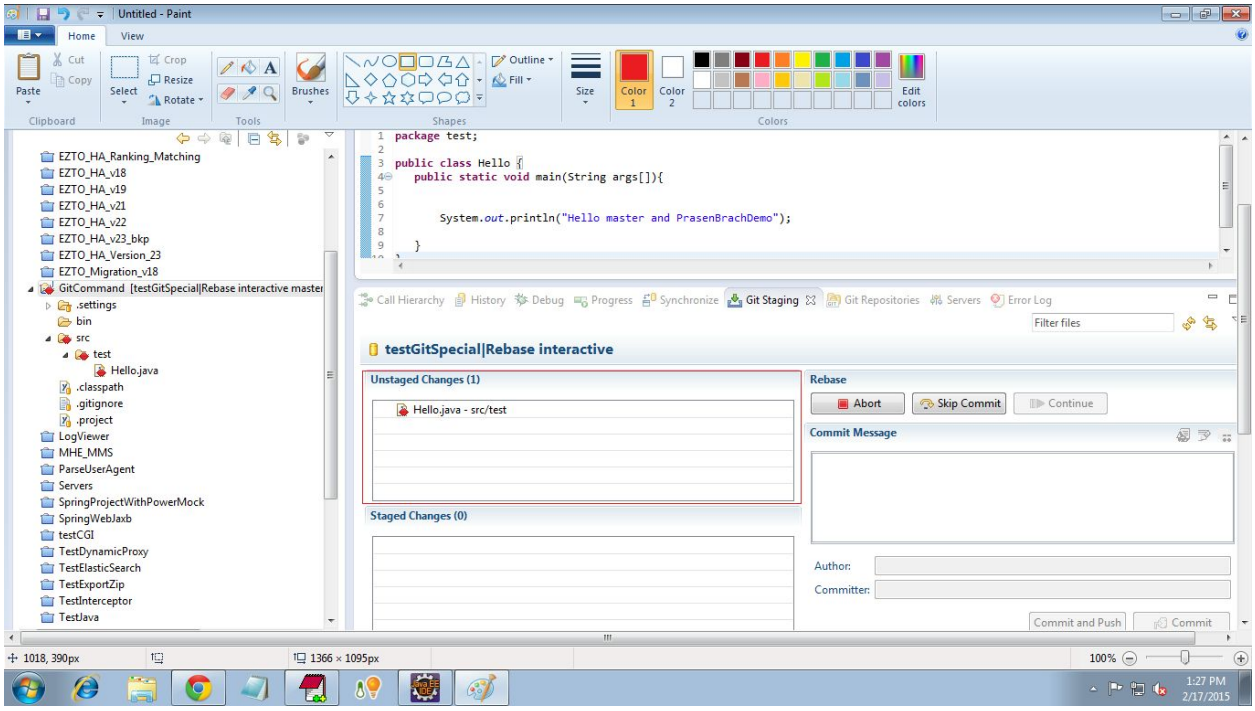


VI. The conflicting code now you can see in your eclipse.

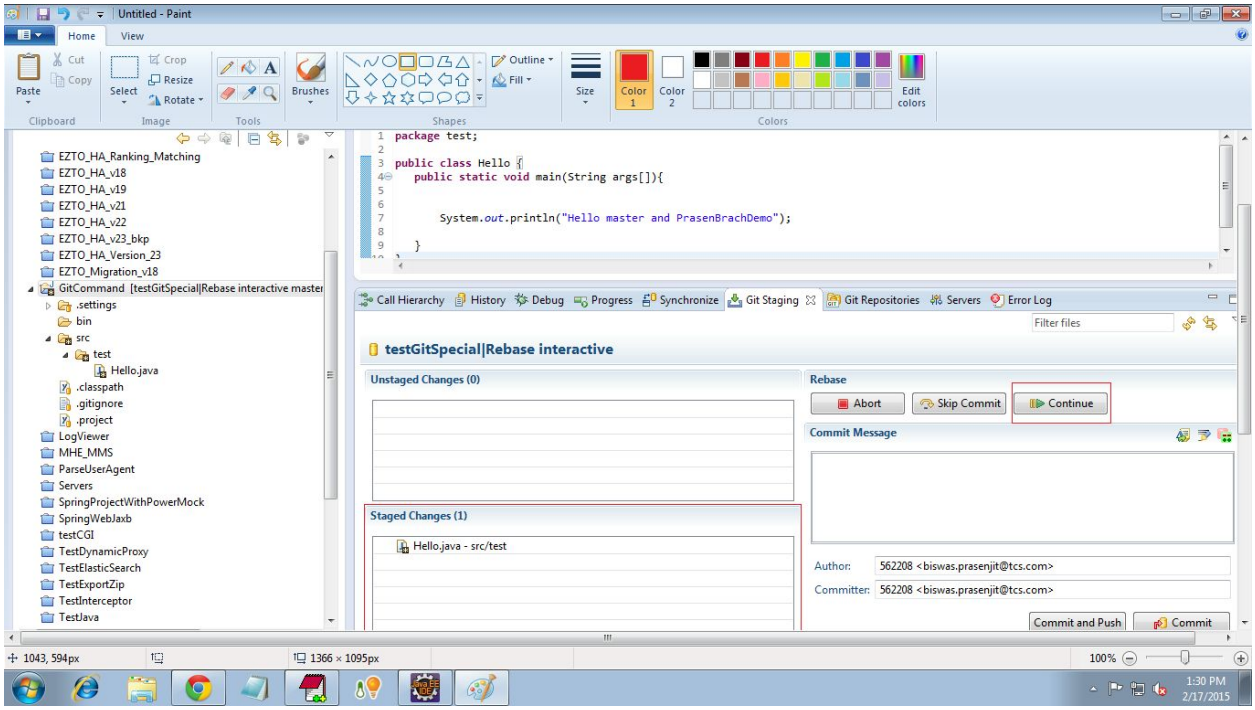


VII. Now Open your code and merge it manually.

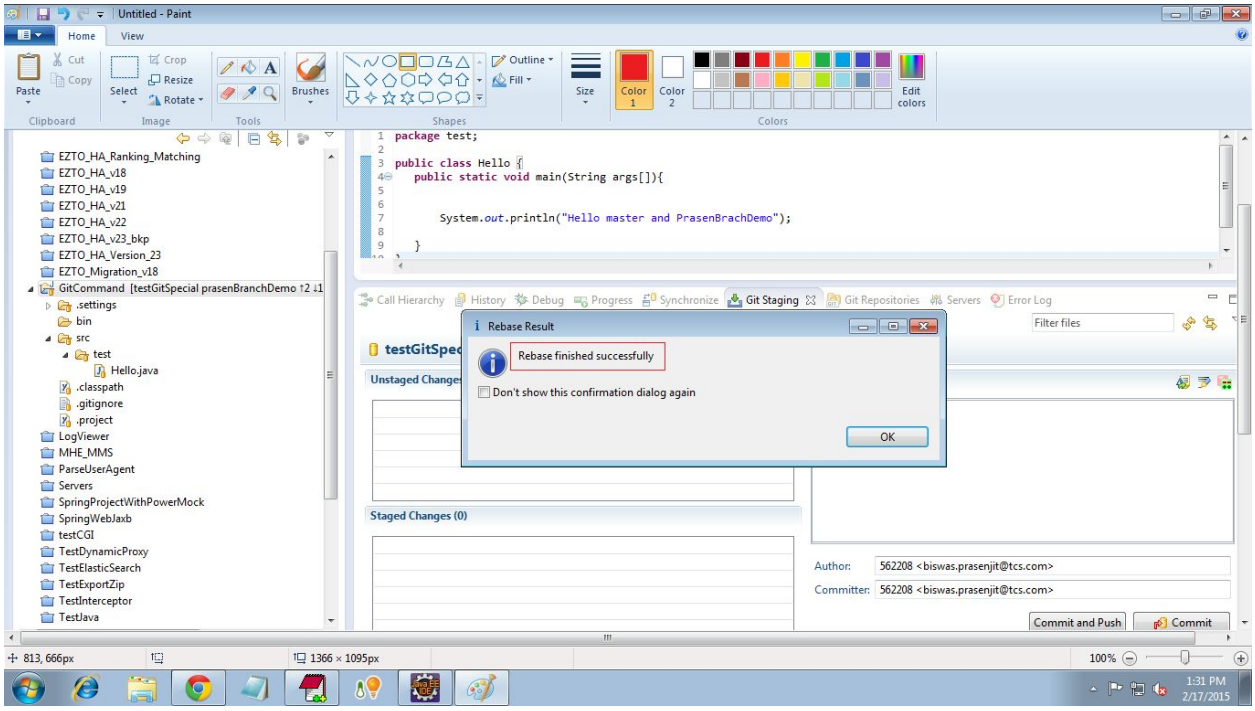
VIII. Open Git Staging view in your eclipse. We should be able to see your updated source file in “Unstaged Changes” section



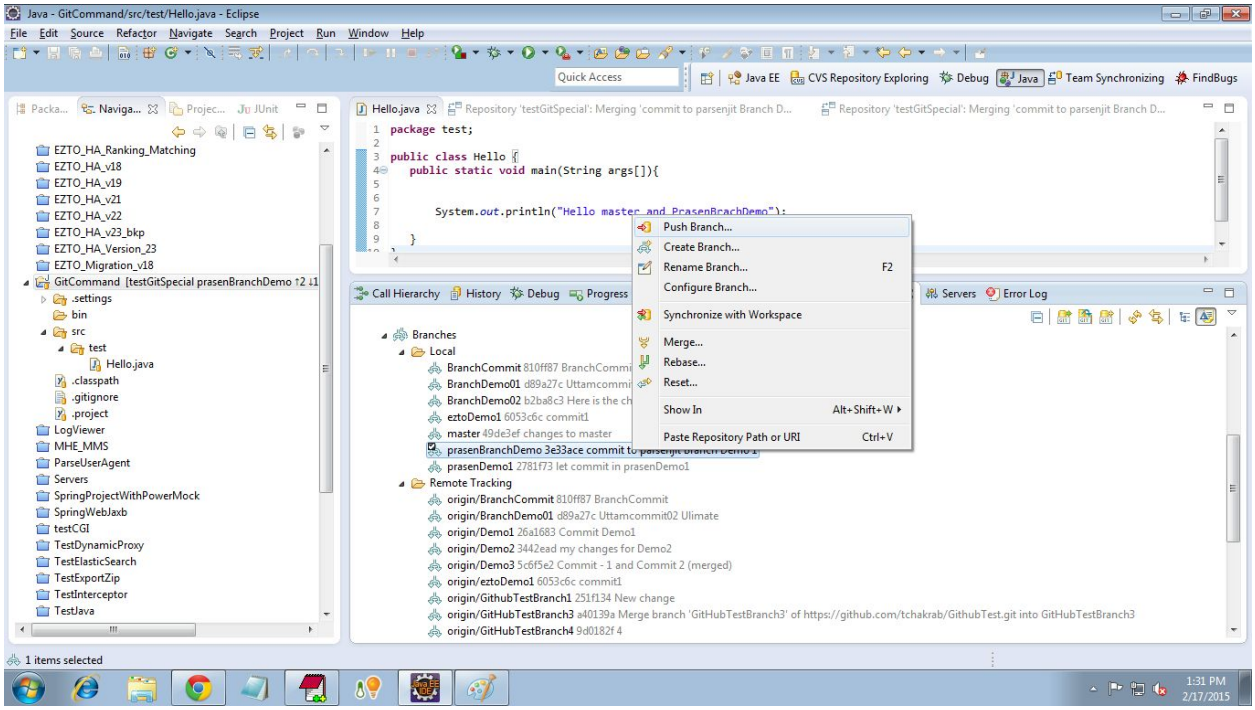
IX. Drag the Updated file from “Unstaged Changes” section to “Staged changes” section And click the continue button.



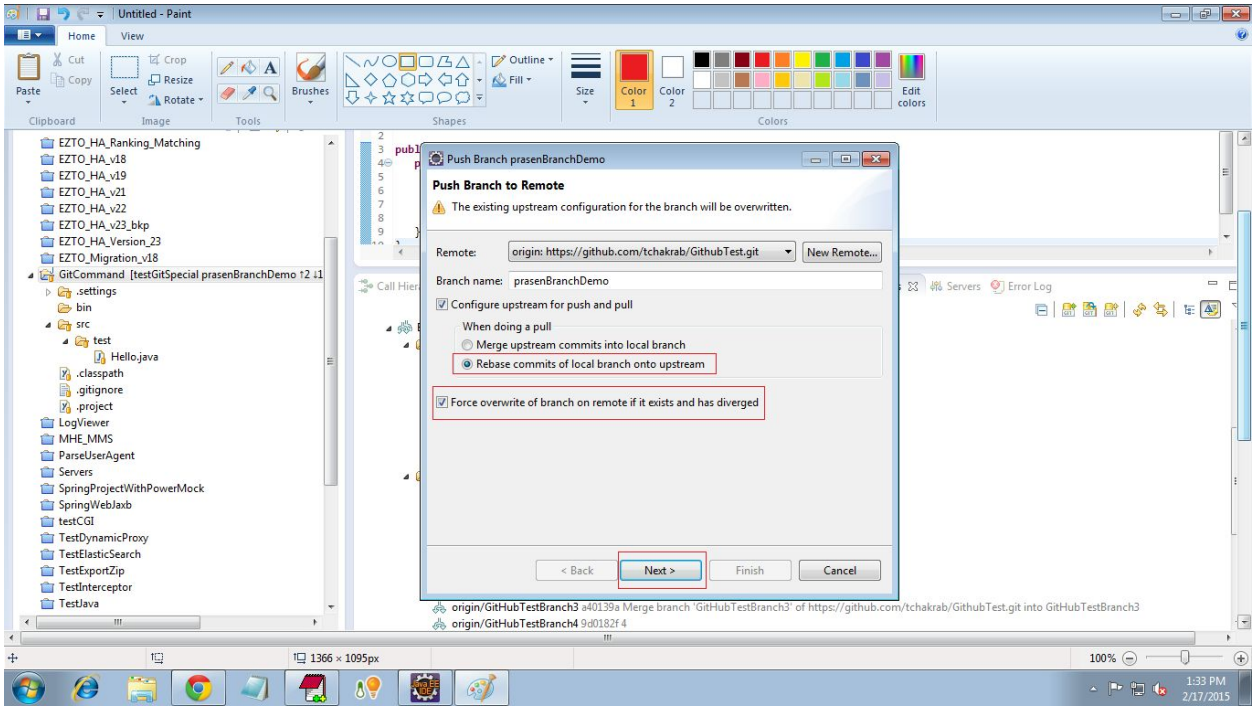
X. It will show Reset successfully completed.



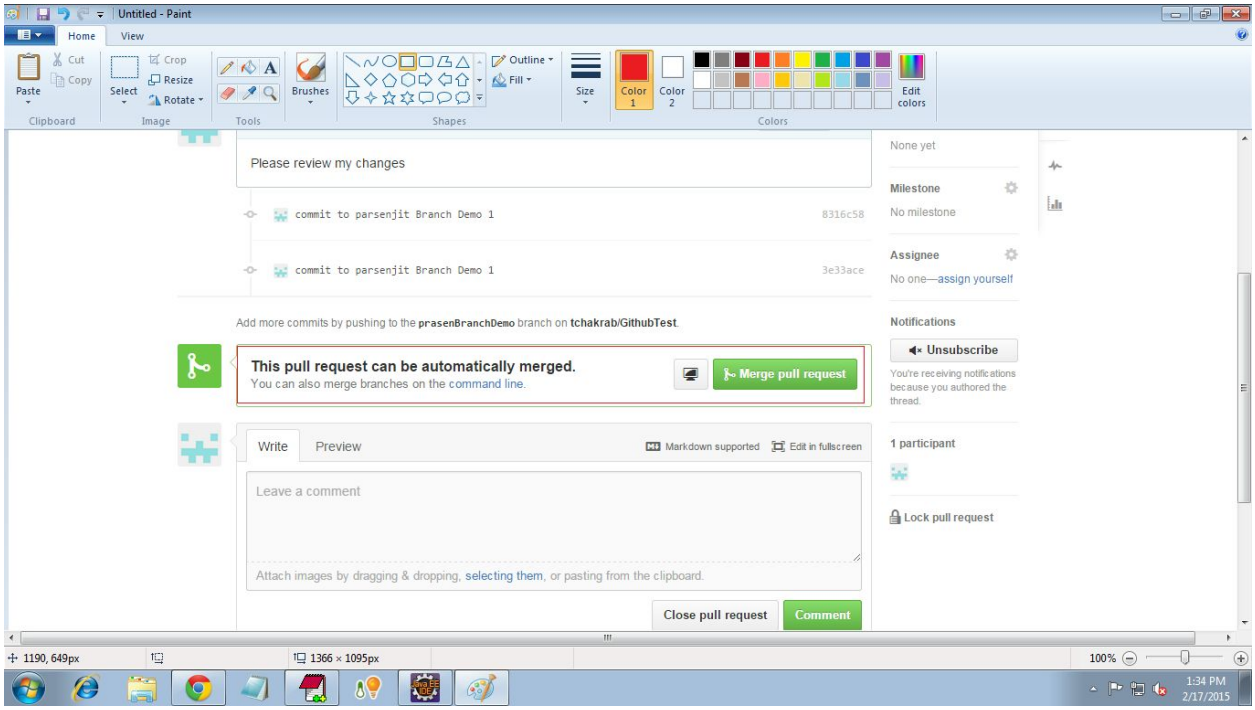
XI. Now, go to git repository view in eclipse. Right click on the branch and go to Push Branch.



XII. This will open a dialogue box, here rebase option should be selected and check the option “Force overwrite of branch on remote” and click the next button.



You are done with rebase. Now go to Git hub user interface your changes should be coming with green merge pull request. That means, your change which has conflict, now can be merged automatically.



But as told you before, please do not merge your code directly to main Branch. It is another one's responsibility. Let him do his work.

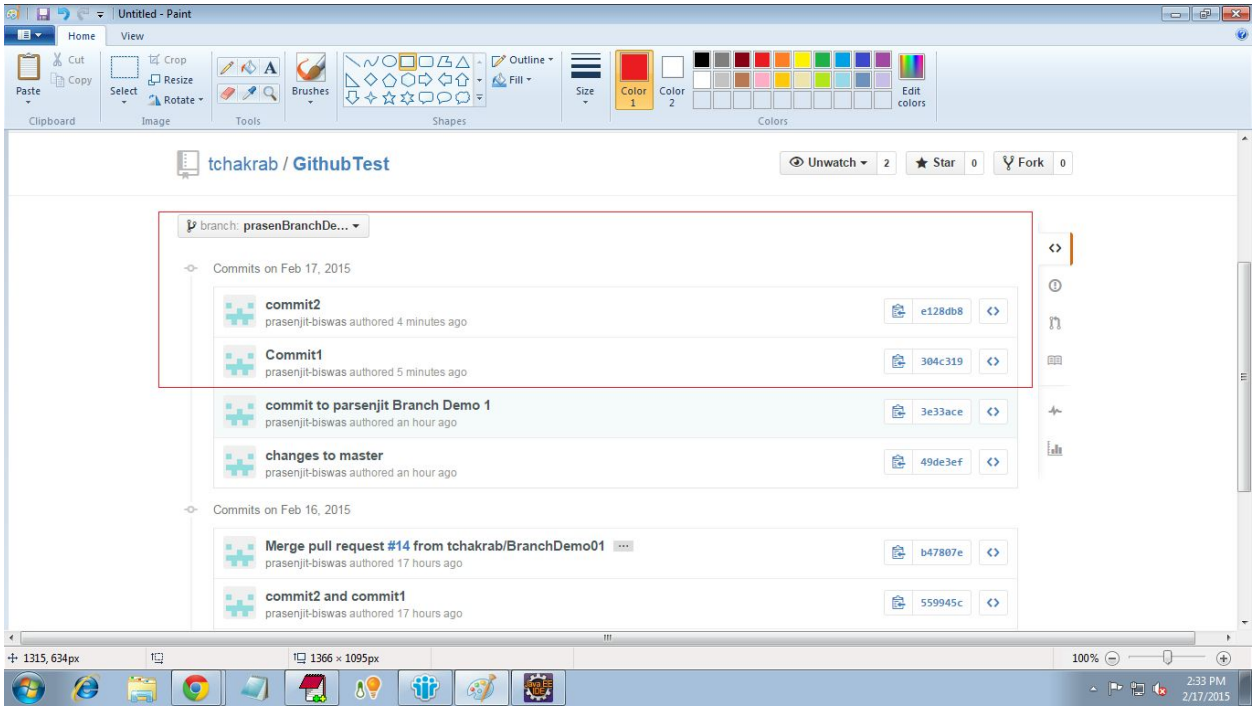
SQUASH

Consider one situation, you have committed your changes and have raised one pull request. Now, One reviewer come up and raised some issues in your code. As a result you need to change your code and go for commit and so on.

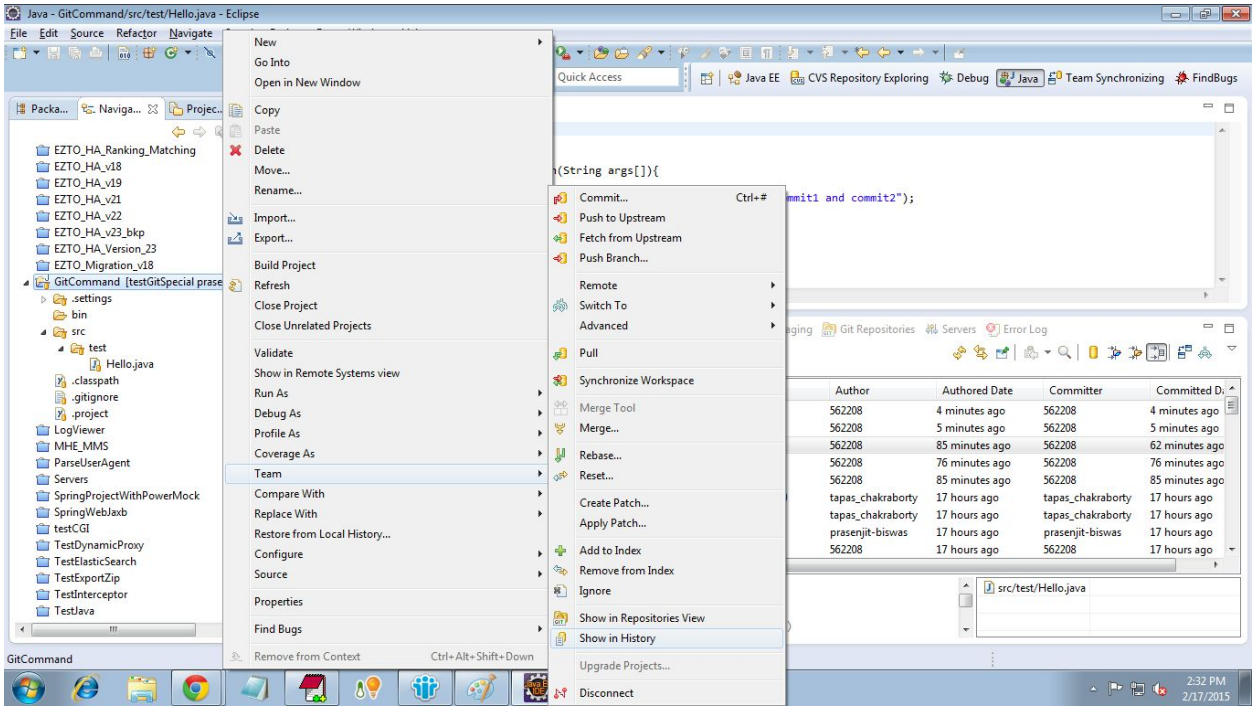
Now the problem is, how to make those multiple commits in a single commit. so that it can be helpful for the reviewer to track your changes.

To Help this situation, Git Hub provides one feature call Squash commit.

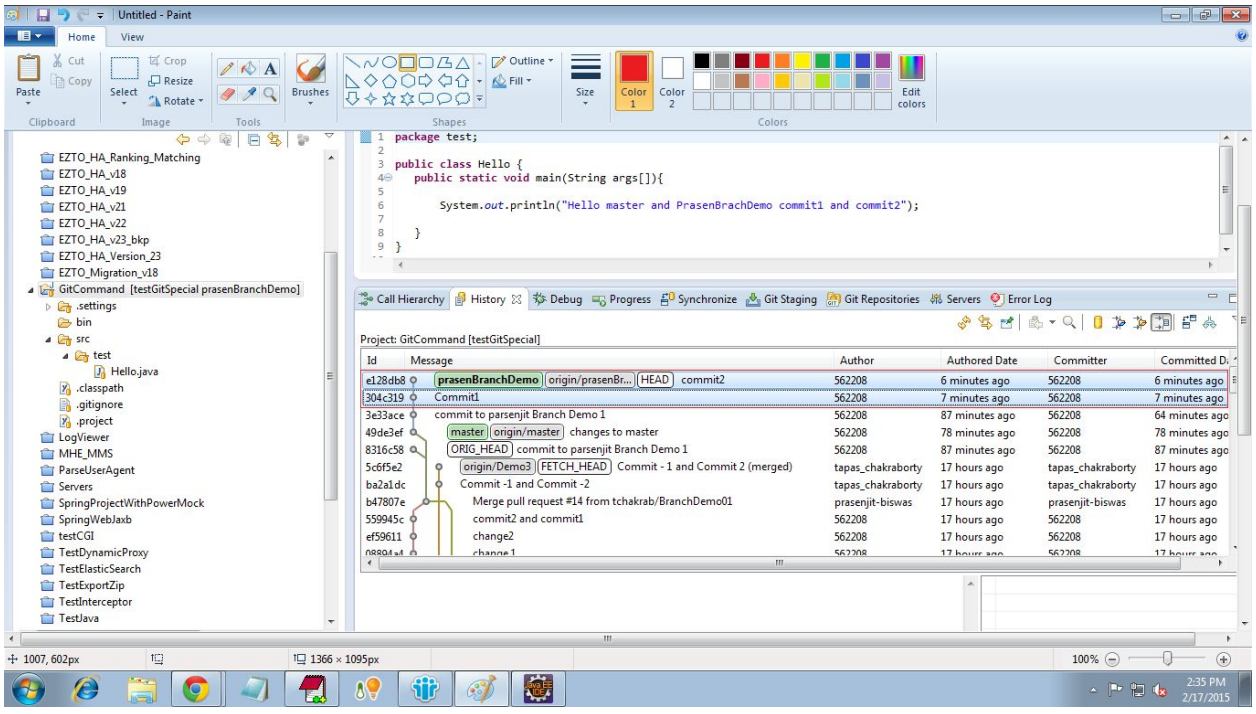
Lets see how can we do squash commit using eclipse.
Say, you have committed 2 times in your branch. you can view that from git user interface



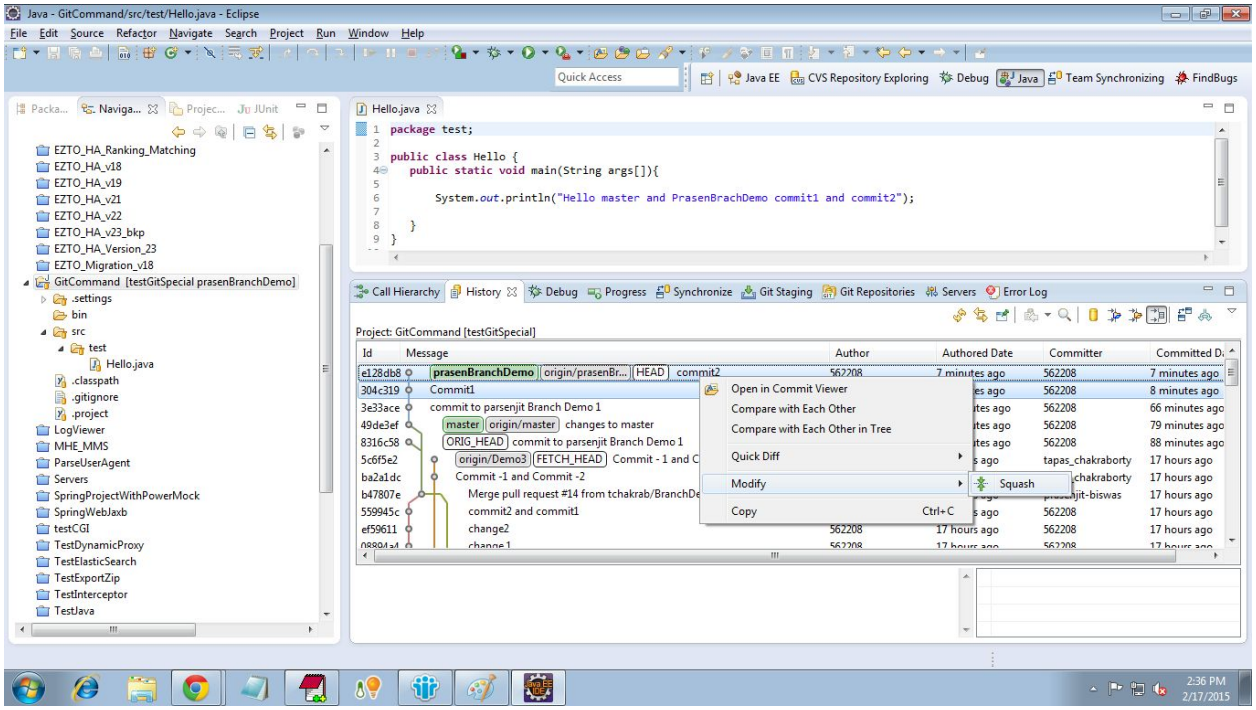
Now, go to your eclipse and open History view.To open history view, Right click on project => Team => Show in History



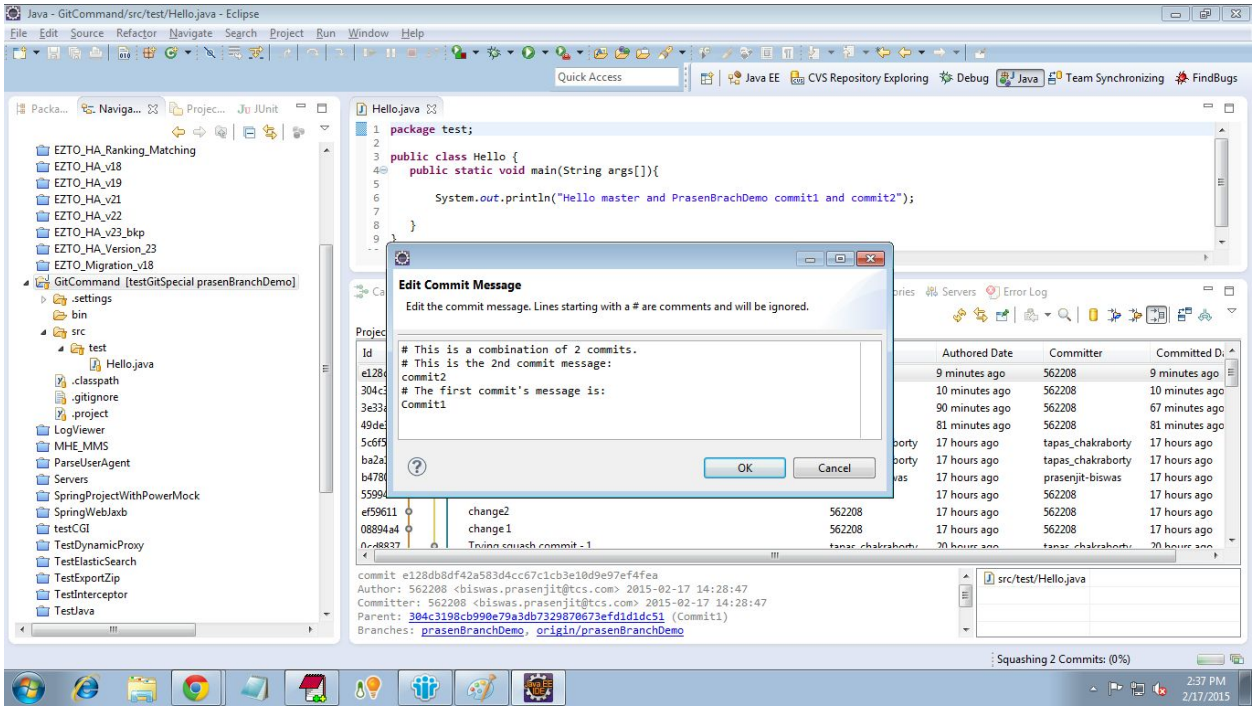
In History view, you can also see your commits



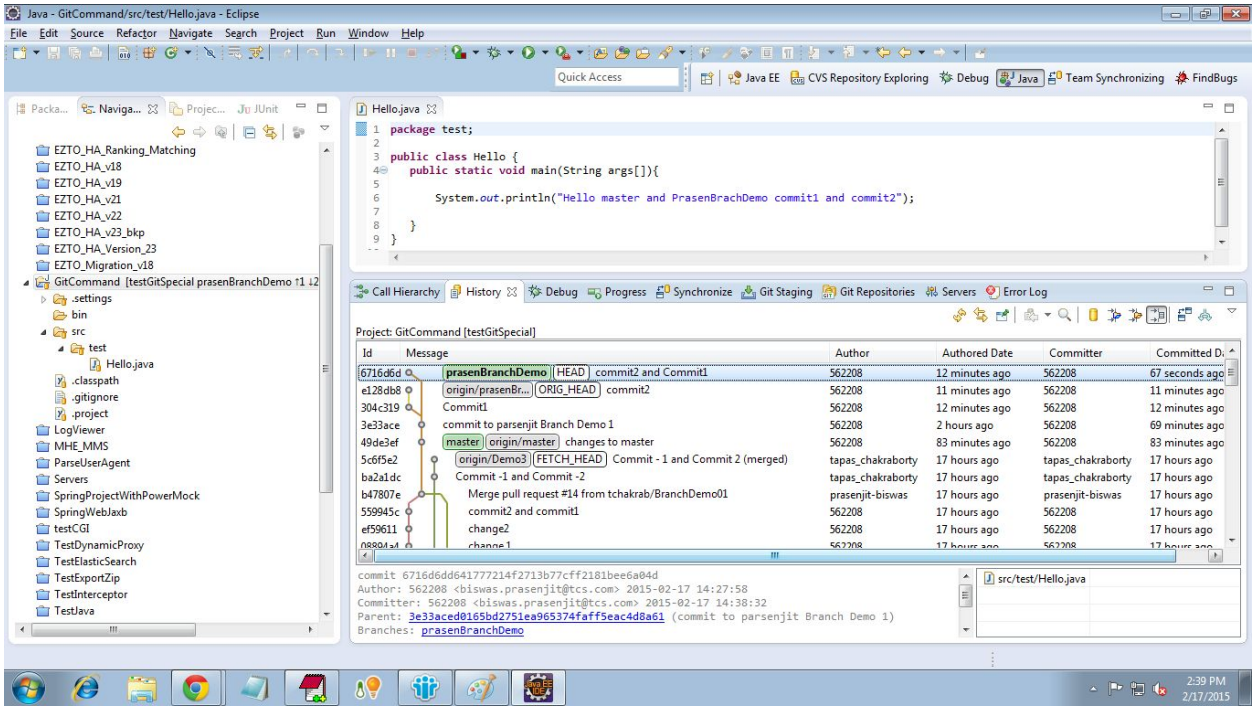
Now, select those commits what you like to squash and then right click on selected commits => Modify => Squash



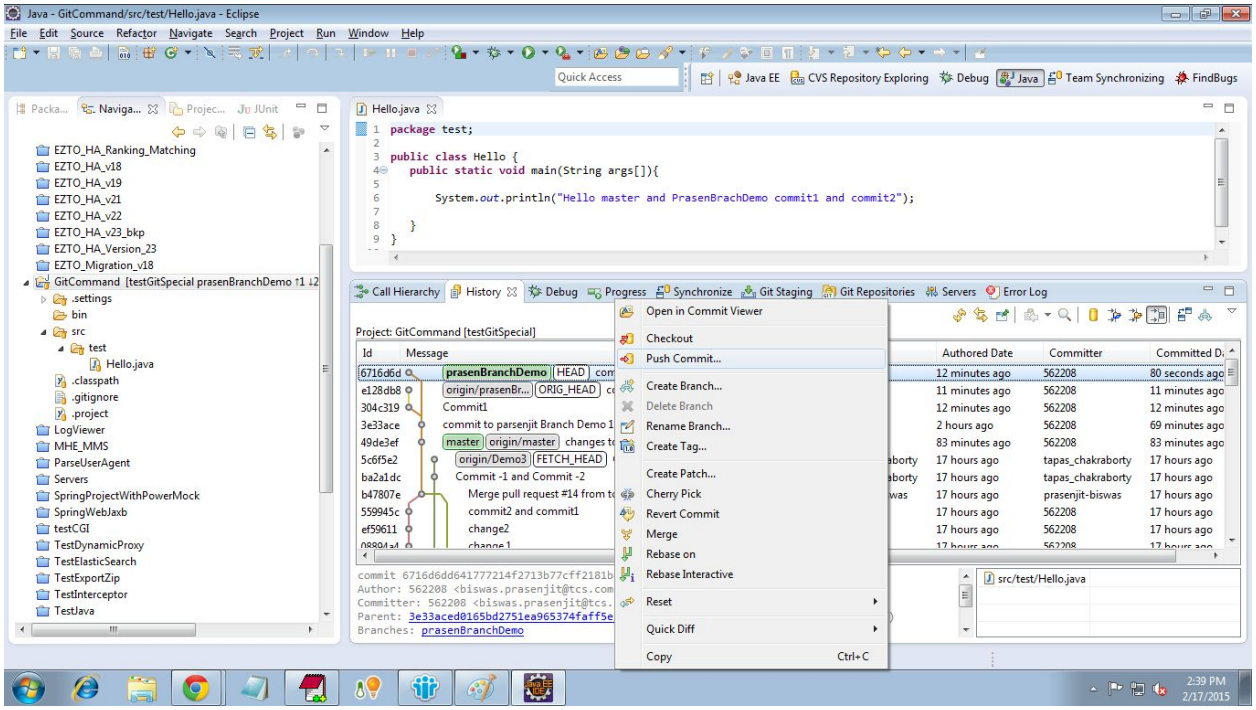
When you select the squash, it will open a dialogue box for editing commit message



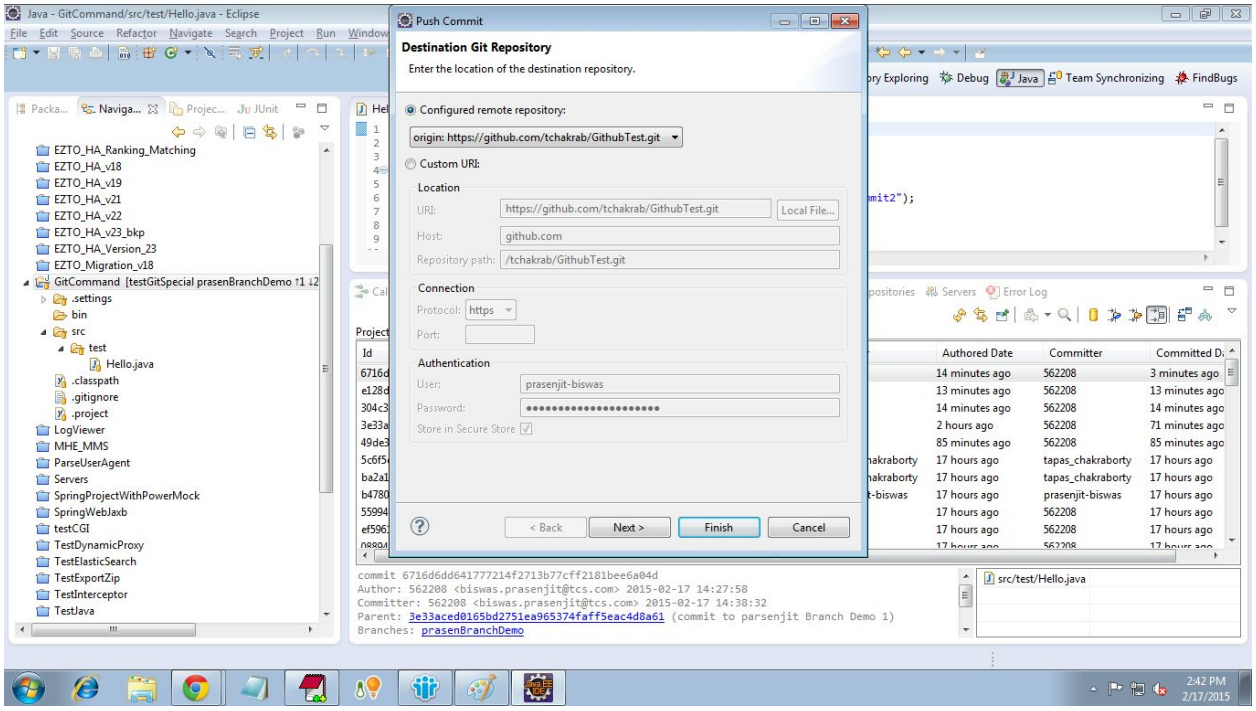
Edit the commit message and click OK. Now in History you can see your two commits are replaced with single commit with your modified commit message.



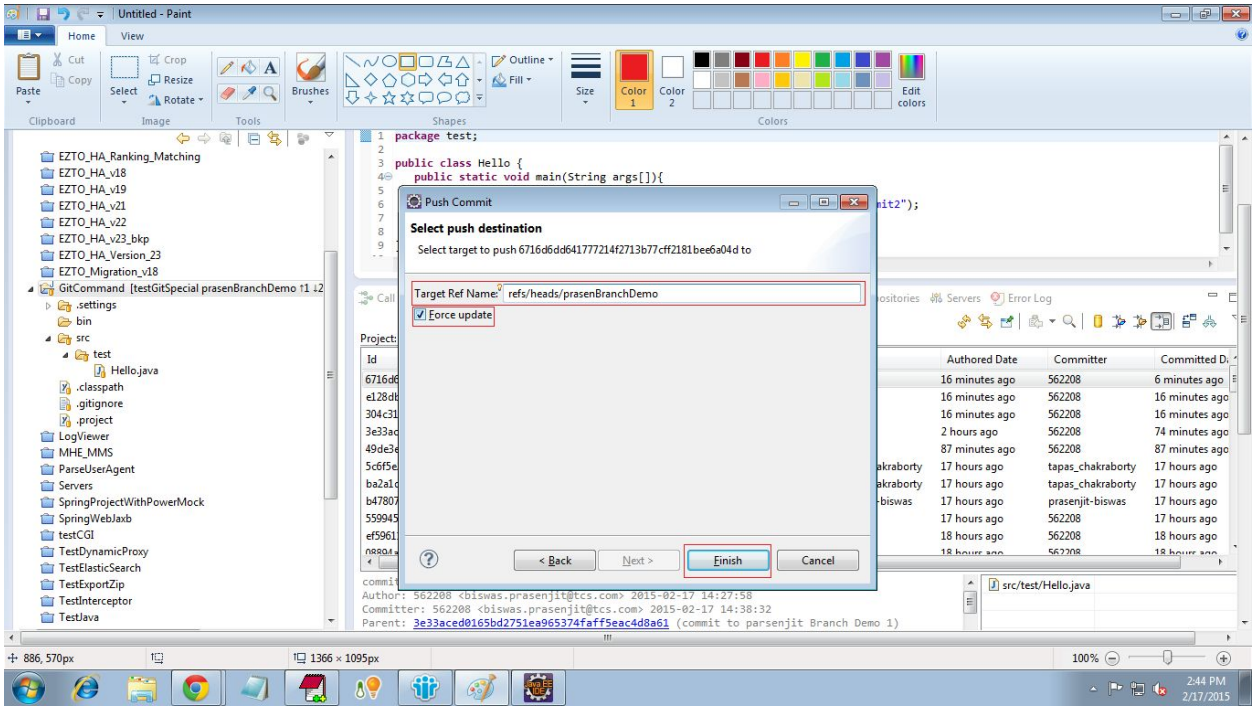
Select that commit from History and go for Push Commit



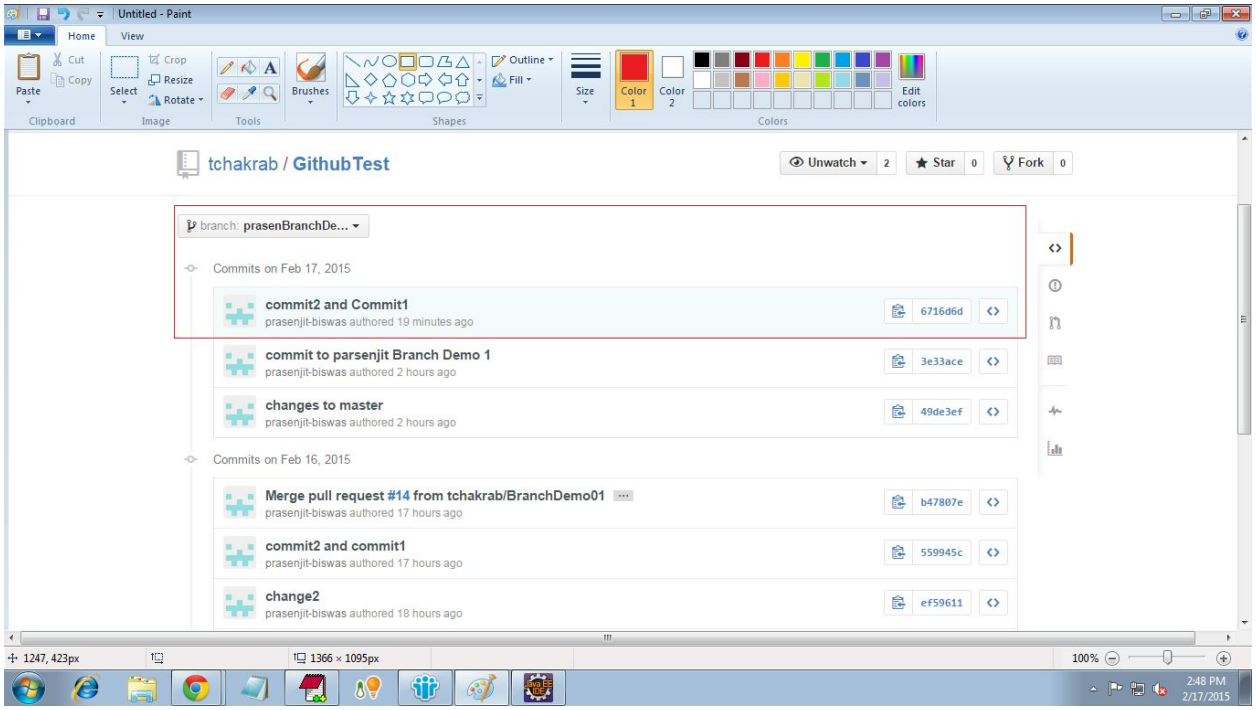
when you do Push Commit, it will open you a dialogue box. here go for Next



In the next screen, select your branch in Target Ref Name field. Force Update checkbox should be checked and then click finish button



And you are done with Squash commit. You can validate the same from Git User Interface also


Untitled - Paint
Home
View
Paste
Cut
Copy
Select
Crop
Resize
Rotate
Brushes
Outline
Fill
Size
Color 1
Color 2
Edit colors
Clipboard
Image
Tools
Shapes
Colors
tchakrab / GithubTest
Unwatch 2
Star 0
Fork 0
branch: prasenBranchDe...
Commits on Feb 17, 2015
commit2 and Commit1
prasenjit-biswas authored 19 minutes ago
6716d6d
commit to parsenjit Branch Demo 1
prasenjit-biswas authored 2 hours ago
3e33ace
changes to master
prasenjit-biswas authored 2 hours ago
49de3ef
Commits on Feb 16, 2015
Merge pull request #14 from tchakrab/BranchDemo01
prasenjit-biswas authored 17 hours ago
b47807e
commit2 and commit1
prasenjit-biswas authored 17 hours ago
559945c
change2
prasenjit-biswas authored 18 hours ago
ef59611
1247, 423px
1366 × 1095px
100%
2:48 PM
2/17/2015