

GitLab - Squashing Commits

Description

Squashing is a way of combining all commits into one when you are obtaining a merge request.

Steps for Squashing Commits

Step 1 – Go to your project directory and check out a new branch with the name *squash-chapter* by using the *git checkout* command –

```
C:\first-gitlab-prjt>git checkout -b squash-chapter
Switched to a new branch 'squash-chapter'
```

The flag *-b* indicates new branch name.

Step 2 – Now, create a new file with two commits, add that file to working directory and store the changes to the repository along with the commit messages as shown below –

```
C:\first-gitlab-prjt>echo "message1" >> README.md
C:\first-gitlab-prjt>git add .
C:\first-gitlab-prjt>git commit -a -m "message1 committed"
[squash-chapter 771bb9a] message1 committed
1 file changed, 1 insertion(+)
C:\first-gitlab-prjt>
```

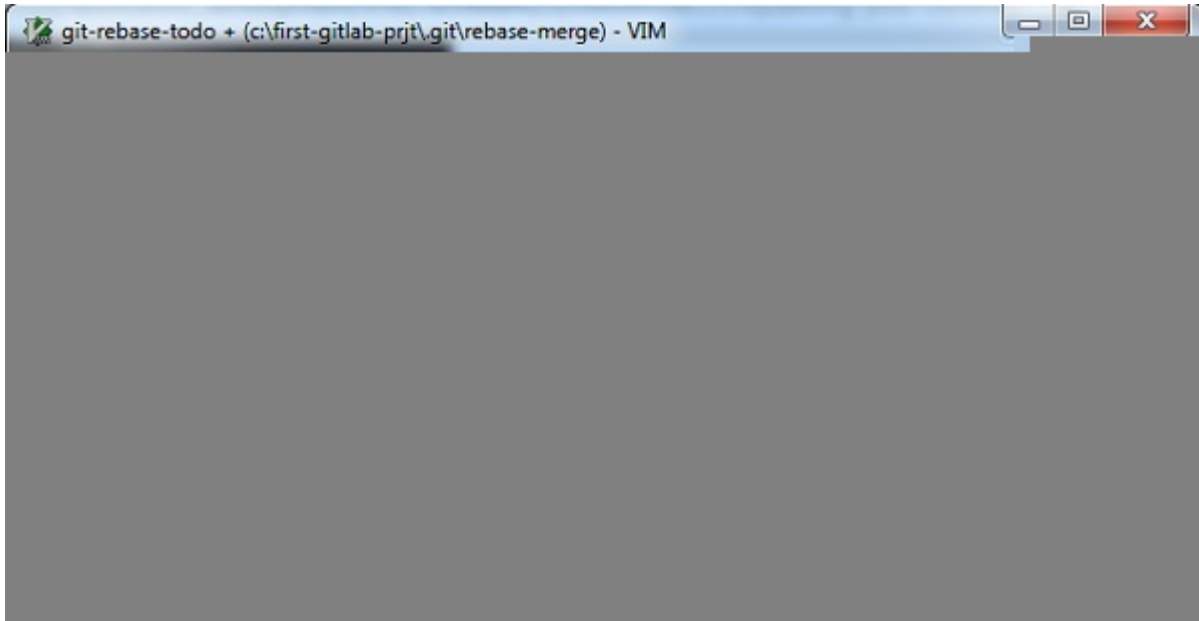
```
C:\first-gitlab-prjt>echo "message2" >> README.md
C:\first-gitlab-prjt>git add .
C:\first-gitlab-prjt>git commit -a -m "message2 committed"
[squash-chapter 6b67004] message2 committed
1 file changed, 1 insertion(+)
C:\first-gitlab-prjt>_
```

Step 3 – Now, squash the above two commits into one commit by using the below command –

```
$ git rebase -i HEAD~2
```

Here, *git rebase* command is used to integrate changes from one branch to another and *HEAD~2* specifies last two squashed commits and if you want to squash four commits, then you need to write as *HEAD~4*. One more important point is, you need atleast two commits to complete the squash operation.

Step 4 – After entering the above command, it will open the below editor in which you have to change the *pick* word to *squash* word in the second line (you need to squash this commit).



Now press the *Esc* key, then colon(:) and type *wq* to save and exit from the screen.

Step 5 – Now push the branch to remote repository as shown below –