

Part 10 - Git Rename, Move and Delete Files



GIT - RENAMING FILES

Q. How to Rename the files in Git ?

METHOD 1 - (Using git commands)

- Let us rename the file states.txt to states_data.txt

Syntax :

git mv <current_filename> <new_filename>

```
[root@localhost projects]# ls
globe      s22  sample1      shadow_file  start3.txt  start5.txt  start7.doc  start.txt  states.txt
README.md  s33  shadow_dump  start2.txt   start4.txt  start6.txt  start.pem   startup.txt
[root@localhost projects]# git mv states.txt states_data.txt
```

Now, execute the ls and git status commands to verify the rename

```
[root@localhost projects]# ls
globe      s22  sample1      shadow_file  start3.txt  start5.txt  start7.doc  start.txt  states_data.txt
README.md  s33  shadow_dump  start2.txt   start4.txt  start6.txt  start.pem   startup.txt

[root@localhost projects]# git status
# On branch master
# Your branch is ahead of 'origin/master' by 5 commits.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       renamed:    states.txt -> states_data.txt
#
```

Commit the changes to Local Repository -

```
[root@localhost projects]# git commit -m "renaming the file states.txt to states_data.txt"
[master f58ec31] renaming the file states.txt to states_data.txt
1 file changed, 0 insertions(+), 0 deletions(-)
rename states.txt => states_data.txt (100%)
[root@localhost projects]#
```

METHOD 2 - (using OS Commands)

- Let us try to rename file states_data.txt to states.txt

mv <current_filename> <new_filename>

STEP 1 : # ls

```
[root@localhost projects]# ls
globe      s22  sample1  shadow_file  start3.txt  start5.txt  start7.doc  start.txt  states_data.txt
README.md  s33  shadow dump  start2.txt  start4.txt  start6.txt  start.pem  startup.txt
```

STEP 2: # mv states_data.txt states.txt

- Execute the command mv
- Then verify using ls command

```
[root@localhost projects]# mv states_data.txt states.txt
[root@localhost projects]# ls
globe      s22  sample1  shadow_file  start3.txt  start5.txt  start7.doc  start.txt  states.txt
README.md  s33  shadow_dump  start2.txt  start4.txt  start6.txt  start.pem  startup.txt
```

STEP 3 - Check the git status

```
[root@localhost projects]# git status
# On branch master
# Your branch is ahead of 'origin/master' by 7 commits.
#   (use "git push" to publish your local commits)
#
# Changes not staged for commit:
#   (use "git add/rm <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    states_data.txt
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       states.txt
no changes added to commit (use "git add" and/or "git commit -a")
[root@localhost projects]#
```

Note - * git sees this as, we have deleted a file and added new file

STEP 4 - Fix this problem -

- To fix the problem, we have to use below command

git add -A

- git add -A flag helps to append the local Repository if there is any change in a file of rename, update, delete.
- Now, let us execute command # git add -A

```
[root@localhost projects]# git add -A
```

Check the git status

```
[root@localhost projects]# git status
# On branch master
# Your branch is ahead of 'origin/master' by 7 commits.
#   (use "git push" to publish your local commits)
#
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       renamed:    states_data.txt -> states.txt
#
[root@localhost projects]#
```

STEP 5 - Commit changes to git Local Repository using git commit command

```
[root@localhost projects]# git commit -m "renaming the file states-data.txt to states.txt"
[master 59b4ecc] renaming the file states-data.txt to states.txt
1 file changed, 0 insertions(+), 0 deletions(-)
```

- Verify using git status command

```
[root@localhost projects]# git status
# On branch master
# Your branch is ahead of 'origin/master' by 8 commits.
#   (use "git push" to publish your local commits)
#
nothing to commit, working directory clean
```

GIT - MOVING FILES

Q. How to move the files in GIT ?

Case Study : Let us move states.txt file into new folder called test

STEP 1 - Let us create folder test and verify it using ls command

```
[root@localhost projects]# mkdir test
[root@localhost projects]# ls
globe      s22  sample1      shadow_file  start3.txt  start5.txt  start7.doc  start.txt  states.txt
README.md  s33  shadow_dump  start2.txt   start4.txt  start6.txt  start.pem   startup.txt test
```

STEP 2 - Execute the below command to move the file states.txt to new folder test

```
[root@localhost projects]# git mv states.txt test/
```

STEP 3 - verify the movement of file states.txt file into new folder test

```
[root@localhost projects]# ls test/  
states.txt
```

STEP 4 - check with git status

```
[root@localhost projects]# git status  
# On branch master  
# Your branch is ahead of 'origin/master' by 8 commits.  
#   (use "git push" to publish your local commits)  
#  
# Changes to be committed:  
#   (use "git reset HEAD <file>..." to unstage)  
#  
#       renamed:    states.txt -> test/states.txt  
#
```

STEP 5 - Now commit the changes to Local Repository

```
[root@localhost projects]# git commit -m "moved the file states.txt to folder test"
[master claf200] moved the file states.txt to folder test
1 file changed, 0 insertions(+), 0 deletions(-)
rename states.txt => test/states.txt (100%)
```

STEP 6 - verify using git status

```
[root@localhost projects]# git status
# On branch master
# Your branch is ahead of 'origin/master' by 9 commits.
#   (use "git push" to publish your local commits)
#
nothing to commit, working directory clean
```

GIT - DELETING FILES

Let us delete the file which is not yet tracked

- Use below command to delete the files that are not yet tracked

rm <file_name>

```
[root@localhost projects]# touch rmfile
```

```
[root@localhost projects]# rm rmfile
rm: remove regular empty file ârmfileâ? y
[root@localhost projects]#
```

CASE STUDY -

- Below error message is replicated when we try to delete the file using git rm command when the file itself is not tracked by git

```
[root@localhost projects]# touch rmfile
[root@localhost projects]# git rm rmfile
fatal: pathspec 'rmfile' did not match any files
[root@localhost projects]#
```

Q. How to delete the file when it is staged -

- Use below command to delete the files in staging area only.

git rm -f <file_name>

Example :

- Created file rmfile and verified by ls

```
[root@localhost projects]# touch rmfile
[root@localhost projects]# ls
globe      rmfile    s33       shadow_dump  start2.txt  start4.txt  start6.txt  start.pem  startup.txt
README.md  s22      sample1   shadow file  start3.txt  start5.txt  start7.doc  start.txt  test
```

Demonstration of deleting file that is staged.

```
[root@localhost projects]# git add rmfile
[root@localhost projects]# git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   rmfile
#
[root@localhost projects]# git rm -f rmfile
rm 'rmfile'
```

Q. How to delete the files that are already tracked in Local Repository ?

- Use below command to delete the files that are in Local Repository

```
# git rm <file-name>
```

- Then commit the changes .

Note - ** Once the file is committed, we cannot recover the files.

Expected Interview Questions -

Q. How do you move or rename the files/folders in Git ?

Q. What command do you use to rename the files in Git ?

Q. What command do you use to move the files in Git ?

Q. How do you delete the files in git that are tracked and in staging area ?

THANK YOU