

GIT COMMANDS

1) check log for process result

git log

2) To show particular file

git show HEAD:<File name with full path> | cat

3) To rollback particular commit

git revert <commit key>

4) Revert Changes to File

- Revert (reset) changes to a file if they haven't been committed yet:

git checkout -- <file>

- Revert File to Previous Commit

Revert (reset) a single file to a specific revision:

git checkout <previous commit_hash> -- <file>

5) To change last commit message

git commit --amend

6) To show committed object file list

git show <Object> --name-only

7) To remove git local cache

git fetch --prune --all

8) Search commit object by message

git log --grep="{MESSAGE_TEXT}" --pretty=format:"%cn | %H | %s | %cd"

git log --grep="Email sequence issue" --pretty=format:"%cn | %H | %s | %cd"

where,

%cn = Committer name

%H = Commit hash

%s = Commit message

%cd = Commit date

9) Rename branch

- If you want to rename a branch while pointed to any branch, do :

git branch -m <oldname> <newname>

- If you want to rename the current branch, you can do:

git branch -m <newname>

10) To set your pointer to particular commit

git reset --hard 91b756827c45b36583acbbfa6b12bc28db30b698

11) To change origin path https:// to ssh://

git remote -v

git remote set-url origin git@bitbucket.org:netclues/royal-fidelity-crm.git

git remote -v

12) To get particular commit

git cherry-pick <commit id>

i.e. git cherry-pick 64887d15b2613fe171ee5c4f73337f9d5e31e504

13) To add all modified file with one command except untracked file

git add -u

14) Save changes before move to another branch and apply back when you come again to this branch

- Show all stash list

git stash list

- Save changes before move

git stash save <stash name>

i.e. git stash save dfp_changes

- Apply stash to branch - Not remove stash from list

git stash apply <stash id>

i.e. git stash apply stash@{0}

- Apply stash to branch - Remove stash from list if conflict is not occurs

git stash pop <stash id>

i.e. git stash pop stash@{0}

- Drop particular stash from stash list

git stash drop <stash id>

i.e. git stash drop stash@{0}

15) Delete local branch

git branch -D <branch name>

i.e. git branch -D www-gaurangg-dfp-changes

16) Recover branch after its deletion

- If you don't know the 'sha' off the top of your head, you can:

//to see full <sha - commit object> that is being abbreviated by default

```
git reflog --no-abbrev
git checkout -b <branch name> <sha - commit object>
```

i.e.

```
git reflog --no-abbrev
git checkout -b branch-1 81aca269d325ca849993ecaafbccd785b4302611
```

- If your commits are not in your reflog:

1) You can try recovering a branch by resetting your branch to the sha of the commit found using a command like:

```
git fsck --full --no-reflogs --unreachable --lost-found | grep commit | cut -d\ -f3 | xargs -n 1 git log
-n 1 --pretty=oneline > .git/lost-found.txt
```

2) You can then display each commit using one of these:

```
git log -p <commit object>
git cat-file -p <commit object>
```

i.e.

```
git log -p 81aca269d325ca849993ecaafbccd785b4302611
git cat-file -p 81aca269d325ca849993ecaafbccd785b4302611
```

3) Once you found your commit, then create a branch on this commit with:

```
git branch <branch name> 81aca269d325ca849993ecaafbccd785b4302611
```

17) Recover a dropped stash

- Find out your stash name after apply below command:

```
gitk --all $( git fsck --no-reflog | awk '/dangling commit/ {print $3}' )
```

- Apply stash using below command:

```
git stash apply $stash_hash
```

i.e. git stash apply 6180e95ecfbf87043fee62da1907f2ab461ea6f0

18) .gitignore file ignore untracked file. If you want to add/ignore tracked file then use below commands:

- First change the file you do not want to be tracked and use the following command:

```
git update-index --assume-unchanged FILE_NAME
```

- If you want to track the changes again use this command:

```
git update-index --no-assume-unchanged FILE_NAME
```

19) Delete remote branch

```
git push origin --delete <branch name>
```

20) How to merge two repository in one repository

<https://blog.doismellburning.co.uk/merging-two-git-repositories/>

21) When .git/refs/origin/master has been corrupt OR 'origin/master' upstream is gone

Below commands creates the missing master branch in the remote repo.

```
rm .git/refs/remotes/origin/master
git fetch
git branch --set-upstream-to=origin/master
```

GIT Settings Commands

```
// It will set your name
git config --global user.name "Your Name"
```

```
// It will set your email
git config --global user.email "you@example.com"
```

```
// If you want git to do a rebase instead of a merge when pulling you can tell git to automatically
rebase when pulling
git config --global pull.rebase true
```

```
// It will only push the current branch when you do git push
git config --global push.default current
```

File commit process as it is. After that follow below steps

1) git checkout master
-> it will take you to master branch
-> make sure you don't have any modified files in your branch, new file will not create any issue

2) git pull
-> it will take latest update from remote master to local master branch

3) git checkout BRANCH NAME
-> you will move back to your working branch

4) git rebase master
-> it will apply master branch changes to local branch
-> solve any conflict if its there

5) git checkout master

6) git rebase BRANCH NAME

-> it will merge your branch changes to local master branch

7) git push

-> it will push local master changes to remote master branch

For more details, please refer below link:

<https://git-scm.com/docs>