

How to Delete Commit History in Github



By Rahul — Updated: July 15, 2021 ⌚ 2 Mins Read — July 15, 2021

This tutorial will help you to delete commit history from your local git repository and remote git repository hosted on Github, Gitlab, Bitbucket, or any other git hosting. In this repository, you will remove the local master branch and create a new master branch. Finally, push changes to the remote git repository.

Delete Commit History in Github Repository

Follow the below instruction to completely delete the commit history of the Github repository.

Warning: This will remove your old commit history completely, You can't recover it again.

1. **Create Orphan Branch** – Create a new orphan branch in git repository. The newly created branch will not show in 'git branch' command.

```
$ git checkout --orphan temp_branch
```

2. **Add Files to Branch** – Now add all files to newly created branch and commit them using following commands.

```
$ git add -A  
$ git commit -am "the first commit"
```

3. **Delete master Branch** – Now you can delete the master branch from your git repository.

```
$ git branch -D master
```

4. **Rename Current Branch** – After deleting the master branch, let's rename newly created branch name to master.

```
$ git branch -m master
```

5. **Push Changes** – You have completed the changes to your local git repository. Finally, push your changes to the remote (Github) repository forcefully.

```
$ git push -f origin master
```

That's it. You have successfully removed the commit history on a Git repository.

Conclusion

In this tutorial, we have described to you to remove the commit history of a Github repository. The same instruction can be followed on other upstream Git repository providers like Gitlab, BitBucket etc.