

# how to delete all commit history in github? [duplicate]

Asked 11 years, 1 month ago   Modified 8 days ago   Viewed 571k times

▲

606

▼

🔖

🕒

This question already has answers here:

[Make the current commit the only \(initial\) commit in a Git repository?](#) (18 answers)

Closed 8 years ago.

I want to delete all commit history but keep the code in its current state because, in my commit history, there are too many unused commits.

How can I do it?

Is there any git command can do this?

```
git filter-branch ?
git rebase ?
...
```

My code is hosted on github.com.

git   github

Share   Improve this question   Follow

edited Feb 11, 2019 at 17:30

 Christopher Bradshaw

2,665   4   24   39

asked Dec 5, 2012 at 5:05

 Chinaxing

8,294   4   30   39

- 2   1) Delete all .git files and .gitignore files in parent directory as well as subdirectory that might have separate .git/.gitignore files. In order to do so, run : `rm -rf .git` command which will delete any file ending with .git. 2) Back out to parent directory and run `git init` which will initialize .git file by creating a new blank .git file without history 3) run `git add .` or `git add *` 4) run `git commit --all -m "initial commit"` 5) run `git --set-upstream origin <git-url>` 6) run `$ git push --mirror <git-repository-path>` This process is going to re write history. – [surendrapanday](#) Jul 8, 2019 at 3:11
- Followed the page and worked like a charm! [docs.github.com/en/authentication/...](#) – [Jugal Panchal](#) Sep 8, 2022 at 22:15

## 2 Answers

Sorted by: Highest score (default) ▼

▲

1979

▼

🔖

✅

🕒

Deleting the .git folder may cause problems in your git repository. If you want to delete all your commit history but keep the code in its current state, it is very safe to do it as in the following:

1. Checkout/create orphan branch (this branch won't show in `git branch` command):  

```
git checkout --orphan latest_branch`
```

2. Add all the files to the newly created branch:  

```
git add -A
```

3. Commit the changes:  

```
git commit -am "commit message"
```

4. Delete `main` (default) branch (this step is permanent):  

```
git branch -D main
```

5. Rename the current branch to `main` :  

```
git branch -m main
```


6. Finally, all changes are completed on your local repository, and force update your remote repository:  

```
git push -f origin main
```

PS: This will not keep your old commit history around. Now you should only see your new commit in the history of your git repository.

Share   Improve this answer   Follow

edited Dec 31, 2023 at 1:10

 Audwin Oyong

answered Sep 23, 2014 at 16:43

