

# Checking out a commit in GitHub Desktop

## In this article

About checking out a commit

Checking out a commit

Troubleshooting

Further reading

You can use GitHub Desktop to checkout a previous commit in your repository.

## About checking out a commit

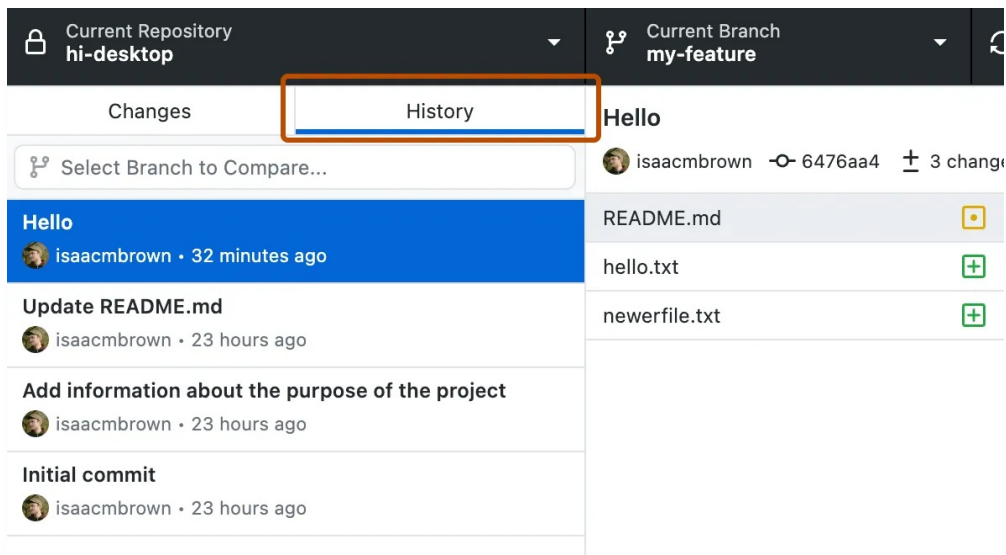
Checking out a commit allows you to view your repository in a previous state without needing to create a new branch or modify an existing branch. This can be helpful when debugging since it allows you to see if a bug exists in your repository at a previous commit.

Checking out a commit puts your repository in a "detached HEAD" state. In Git terminology "HEAD" is the reference that points to the tip, or latest commit, of a named branch in your repository. A "detached HEAD" state means that HEAD refers to a specific commit, but not on a named branch in your repository.

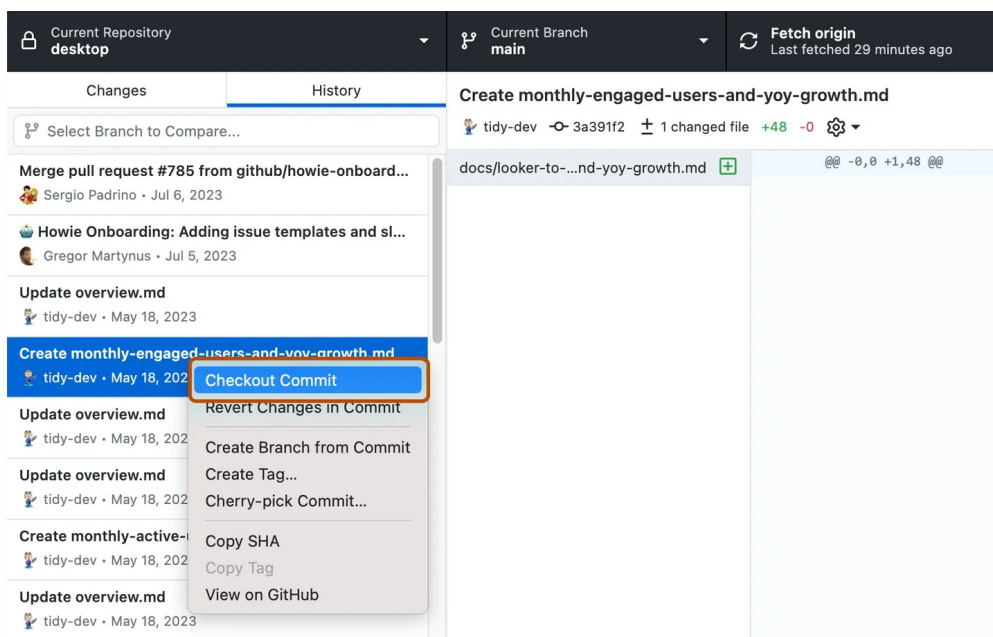
**Note:** Any commits made in a "detached HEAD" state will be lost when switching branches, since these commits have not been made on a named branch. If you need to recover the lost commits, see "[Troubleshooting](#)."

## Checking out a commit

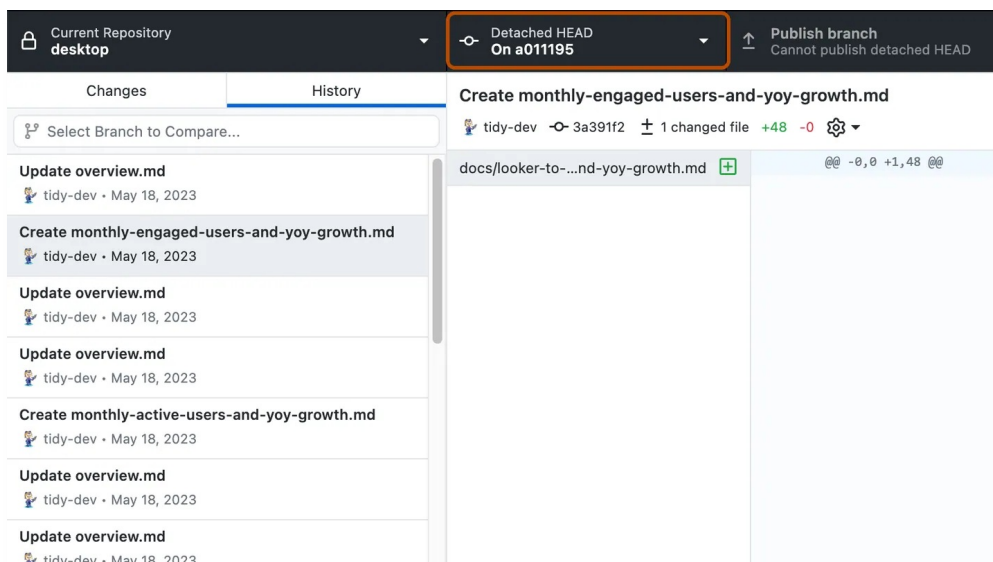
- 1 In the left sidebar, click **History**.



- Right-click on the commit you would like to checkout and select **Checkout commit**.



- The **Current Branch** item in the repository bar will now show "Detached HEAD", along with the SHA of the commit that was checked out.



- 4 To exit the "detached HEAD" state you will need to switch branches. For more information, see "[Managing branches in GitHub Desktop](#)."

## Troubleshooting

---

You can recover commits that have been made in a "detached HEAD" state using the `git reflog` command from the Git command line. You can open your repository in the command line from GitHub Desktop by going to the menu bar, selecting **Repository**, and clicking **Open in command line**.

The `git reflog` command will show the output of events that have happened in your repository, including commits. Here is a sample output of the `git reflog` command:

```
81fa9136f8 (HEAD -> main) HEAD@{0}: checkout: moving from
8bd5e736a27a52a7e36a856b30e6f0582d341aa1 to main
8bd5e736a2 HEAD@{1}: commit: testing out a feature
22fa76c125 HEAD@{2}: checkout: moving from main to
22fa76c1250a2847305b9325752d941dbaa55983
```

The `8bd5e736a2 HEAD@{1}: commit: testing out a feature` line is the commit that we want to recover, since it was made while the repository was in a "detached HEAD" state. To recover it you can run `git cherry-pick 8bd5e736a2` to apply the commit to the current branch in your repository.

## Further reading

---

- [Detached HEAD](#) in the Git documentation
- [Git cherry-pick](#) in the Git documentation

### Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)