

## Prerequisites

- You should have made the necessary changes cited in the task's step-by-step guide

### What is a patch file

- You should know by now that git is a way for developers to manage code in a project especially if there's other developers collaborating in that project too.
- A git patch file is just a file that you can apply to a repository to get the changes / modifications / additions another developer did on his / her machine onto your local machine. This isn't the only way to do that of course but this is a viable method if you don't have push access to a repository

### How to make a patch file

Scenario 1: You only made one commit for all the required changes

- Fire up a terminal, enter the repository via the terminal you opened (via the `cd <repo_name_here>` aka change directory command) and run the command below

`git format-patch -1 HEAD`

- After executing the command, a .patch file will be produced in the directory where you executed the command. You will upload this as your submission to the task

```
b6a0efa (HEAD -> main) Apply updates to meet requirements
379221f (origin/main, origin/HEAD) Re-add sample unit test
24707a7 Rename client and server files
bd3f50e Added Requirements
bf70ef0 initial commit
5ae952a Update README.md
fc80355 Initial commit
```

← In this example, this top-most commit is the only commit made on top of existing commits to apply all changes for the task

## Scenario 2: You made multiple commits for all the required changes

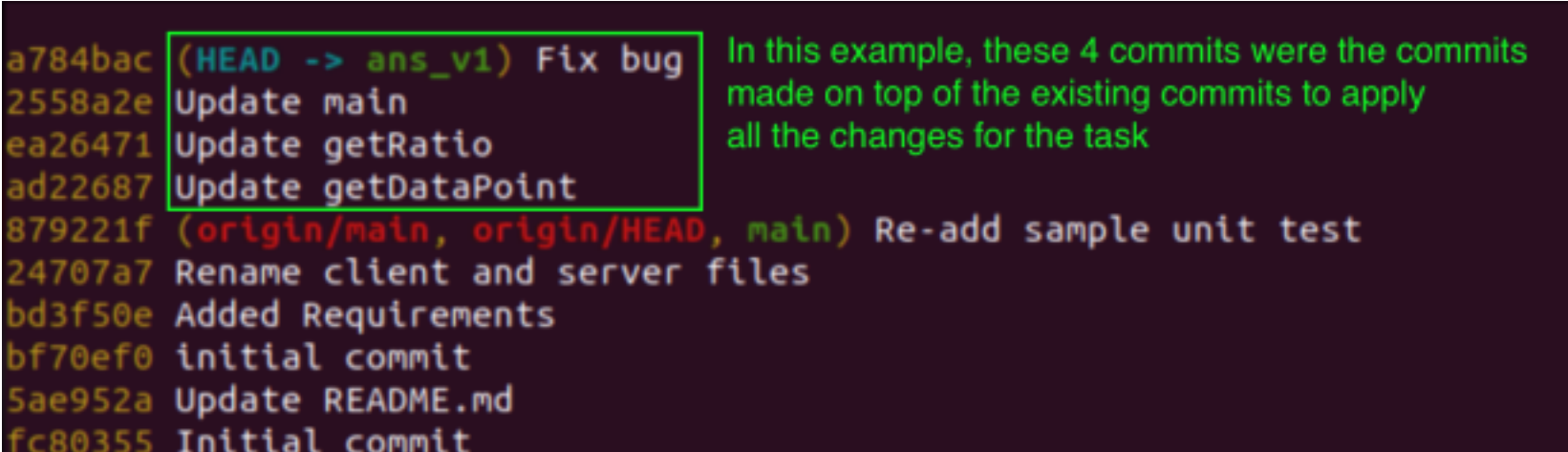
- Fire up a terminal, enter the repository via the terminal you opened (via the **cd** <repo\_name\_here> aka [change directory command](#)) and run the command below

```
git format-patch -n -stdout > multi_commit.patch
```

*note: the **n** in **-n** must be replaced with a number which represents the number of commits you made for the task. So the real command if you made 4 commits on top of the old commits should be*

```
git format-patch -4 --stdout > multi_commit.patch
```

- After executing the command, a .patch file will be produced in the directory where you executed the command. You will upload this as your submission to the task



```
a784bac (HEAD -> ans_v1) Fix bug
2558a2e Update main
ea26471 Update getRatio
ad22687 Update getDataPoint
879221f (origin/main, origin/HEAD, main) Re-add sample unit test
24707a7 Rename client and server files
bd3f50e Added Requirements
bf70ef0 initial commit
5ae952a Update README.md
fc80355 Initial commit
```

In this example, these 4 commits were the commits made on top of the existing commits to apply all the changes for the task