

# Final Project: Part 2 - Git CLI

**Estimated Time:** 60 minutes

## Important notice about this lab environment:

Please be aware that sessions for this lab environment are not persisted. Every time you connect to this lab, a new environment is created for you. Any data you may have saved in the earlier session would get lost. Plan to complete this lab in a single session, to avoid losing your data.

## Scenario

Congratulations on starting the journey with your company by creating an open-source Simple Interest Calculator bash script on GitHub. Your changes have been accepted and merged and the company has created a new global [repository](#) for the teams to collaborate. Other developers have contributed to this repository over time. Your team has found a mistake in one of the markdown files. You are asked to fork this repository and fix the mistake by using Git CLI in the provided lab environment and open a pull request (PR).

## Objectives

After completing this lab, you will be able to demonstrate that you can:

1. Fork the upstream repository into your own account.
2. Clone the code locally in the lab environment.
3. Create a branch in the repository.
4. Make changes in the branch and commit it.
5. Merge the branch back into the main branch.
6. Create a Pull Request from the forked repository to the upstream repository.
7. Revert a change that you made earlier.

## Note:

1. As mentioned in the **Final Project Overview Criteria**, you can submit your project deliverables through either Option 1: AI-Graded Submission and Evaluation or Option 2: Peer-Graded Submission and Evaluation.
2. **For Option 1: AI-Graded Submission and Evaluation:**
  - Submission requires the **curl commands and the terminal output** for Task 6 to 9.
3. **For Option 2: Peer-Graded Submission and Evaluation:**
  - Submission requires the **URLs** and the screenshot for Task 6 to 10.

## Task 1: Fork the repository

1. Fork the project's [source repository](#). And make sure that it is public and ensure the repository name must be exactly the same as given.

**For Option 1: AI-Graded Submission and Evaluation:**

2. Run the curl command that verifies your pull request is valid and shows the forked repository from which the pull request was created. This confirms your repository has been forked from the repository: - ibm-developer-skills-network/mcino-Introduction-to-Git-and-GitHub

```
curl -s https://api.github.com/repos/<username>/mcino-Introduction-to-Git-and-GitHub | jq -r '.parent.clone_url'
```

**Note:** Replace the <username> in the above command with your GitHub username.

- Copy and save the curl command along with its terminal output in the text file named **forked-repo** for the final project submission and evaluation.

**For Option 2: Peer-Graded Submission and Evaluation:**

- Copy and save the GitHub URL of the forked repository named as **mcino-Introduction-to-Git-and-GitHub** in the text file ensuring it contains all the required contents for Peer Assignment.

## Task 2: Fix the typo and merge with main

1. Clone the forked repository in the lab environment.
2. Create a branch named bug-fix-typo.
3. Change the footer in the main README.md file from

2022 XYZ, Inc.

to

2023 XYZ, Inc.

4. Add the file with your fix and commit it with a meaningful message.
5. Push your fix to the `bug-fix-typo` branch. In this step, you will need to generate a personal access token from GitHub.com to use as your password. Follow the instructions in the lab [Generate GitHub personal access token](#).
6. Switch to the main branch. Merge the `bug-fix-typo` branch back into your main branch.

**For Option 1: AI-Graded Submission and Evaluation:**

- Copy and save the terminal output of the first merge in the text file named **merge\_branches**, showing the commands `git checkout main` and `git merge bug-fix-typo` for the final project submission and evaluation. The output should clearly indicate that 1 file was changed.

**For Option 2: Peer-Graded Submission and Evaluation:**

- Take a screenshot showing the current branch and successful merge operation with the file that has changed. Save the screenshot as `merge_branches.png` or `merge_branches.jpg` for Peer Assignment.

## Task 3: Revert the typo and submit a pull request

1. Check the content of `README.md` in the main branch. The file should now read:

```
2023 XYZ, Inc.
```

2. Create a new branch named `bug-fix-revert`.
3. Revert back the change you implemented in the previous task using the `git revert` command. The file should now read:

```
2022 XYZ, Inc.
```

4. Push the revert to your repository in the `bug-fix-revert` branch. Please ensure you use the personal access token that you generated on Github for your account as the password (and not your actual git password), when prompted.
5. Go to the GitHub UI. Create a new pull request from the `bug-fix-revert` branch of your repository to the main branch of the [original repository](#). This PR will be closed automatically.

#### For Option 1: AI-Graded Submission and Evaluation:

- Run the curl command that verifies your pull request is valid and shows the forked repository from which the pull request was created.

```
curl -s https://api.github.com/repos/ibm-developer-skills-network/mcino-Introduction-to-Git-and-GitHub/pulls/<Pull-Request-ID> | jq -r '.head.repo.clone_url'
```

**Note:** Replace the `<Pull-Request-ID>` in the above command with your GitHub Pull Request ID.

- Copy and save the curl command along with its terminal output in the text file named **bug-fix-revert** for the final project submission and evaluation.

#### For Option 2: Peer-Graded Submission and Evaluation:

- Copy and save the GitHub repository URL of this Pull Request in a text file to submit later for Peer Assignment.

## Task 4: Check the status of your branches

#### For Option 1: AI-Graded Submission and Evaluation:

- Open your terminal and navigate to your cloned repository directory.
- Run the following command to view all branches along with their current status:

```
git branch -vv
```

This command displays all the branches you have created and their respective status.

- Copy and save the terminal output along with the command in the text file named **github-branches** showing the branch names and their status in the repository for the final project submission and evaluation.

### For Option 2: Peer-Graded Submission and Evaluation:

Navigate to the Branches section within the GitHub UI on your page. It will be in the following format:

<https://github.com/<Your Github username>/mcino-Introduction-to-Git-and-GitHub/branches>

- Within this section, you will find the branch names along with their current status.
- Copy and save the GitHub repository URL of this page in a text file to submit during Peer Assignment.

## Checklist for submission

Follow the checklist below to verify that your project meets all requirements before submission.

**Submit your work through either Option 1: AI-Graded Submission and Evaluation or Option 2: Peer-Graded Submission and Evaluation, depending on the submission path you choose for project evaluation.**

### Checklist for Option 1: AI-Graded Submission and Evaluation

After completing the final project, you should have saved:

1. The GitHub repository URL named as **github-final-project** where the updated README.md file is located.
2. The GitHub repository URL that redirects to the **Apache 2.0 license** file.
3. The GitHub repository URL where the **CODE\_OF\_CONDUCT.md** file is present.
4. The GitHub repository URL where the **CONTRIBUTING.md** file is present.
5. The GitHub repository URL where the **simple-interest.sh** file is present.
6. The curl command along with the terminal output in the file named **forked-repo** that verifies the forked repository.
7. The terminal output of the first merge in the file named **merge\_branches**, including the commands `git checkout main` and `git merge bug-fix-typo`.
8. The curl command along with the terminal output in the file named **bug-fix-revert** validating the pull request.
9. The terminal output of the **Branches** using the command `git branch -vv` in the file named **github-branches** showing the branches of the repository and their status.

### Checklist for Option 2: Peer Review Evaluation

After completing the final project, you should have:

1. The GitHub repository URL of repository named 'github-final-project' ensuring it contains all the required files.
2. The URL of the **LICENSE file** (Apache 2.0 License) present in the repository.
3. The URL of **README.md** file present in the repository.
4. The URL of **CODE\_OF\_CONDUCT.md** file present in the repository.

5. The URL of **CONTRIBUTING.md** file present in the repository.
6. The URL of the file **simple-interest.sh** present in the repository.
7. The URL of the forked repository named as **mcino-Introduction-to-Git-and-GitHub** ensuring it contains all the required contents.
8. A screenshot of the first merge, showing that the current branch is main and the changes made to the README.md file in the bug-fix-typo branch have been successfully merged into main.
9. The URL of the pull request that has been accepted.
10. The URL of the Branches page in your repository, showing all three branches along with their status.

## Summary

Congratulations! You have completed both parts of the final project. You have demonstrated that you know how to create an open-source project in Git, make changes to that project, and make it available to the community. You can fork a GitHub repository, clone it to your local system, make changes to the local repository, commit the changes locally, push it back to your GitHub fork, and create a pull request to add your update to the original repository.

## Author(s)

**Md Haroon Hussain**

**© IBM Corporation. All rights reserved.**