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.

Objectives

After completing this lab, you will have demonstrated 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: Throughout this lab you will be prompted copy and paste URLs into a notepad and save the notepad on your own device. These URLs will be uploaded for peer review in the Final Submission section of the course. You can use any notepad app to keep note of your URLs.

Task 1: Fork the repository

1. Fork the project's source repository.
2. **Save the URL of your forked repository in a notepad to submit later for peer review.**

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 your to the `bug-fix-typo` branch. You will need to generate a personal access token from GitHub.com to use as your password in this step. Follow the instructions in the lab named **Generate GitHub personal access token**.
6. Switch to the main branch. Merge the `bug-fix-typo` branch back into your `main` branch. Take a screenshot showing the current branch and successful the merge operation with the file that has changed and save it as `merge_branches.png` or `merge_branches.jpg`.

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. As before, your second PR is closed automatically. **Note the URL of this PR in a notepad to submit later for peer review**

Task 4: Check the status of your branches

1. Navigate to the `Branches` section within the GitHub UI on your page. Its link will be in the following format:

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

2. Within this section, you will find the branch names along with their current status.

3. Make a note of the URL of this page in a notepad for later submission during peer review.

Checklist

After completing this part of the final project, you should have:

1. The forked repository URL.
2. The pull request URL.
3. The screenshot showing the merge operation of `bug-fix-typo` into `main` named `merge_branches` in png of jpg format.
4. The URL of the `Branches` page showing the branches of the repository and 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 that repository 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)

Upkar Lidder