Chameleon
Smarter World

City of Melbourne
Open Data

DATA SCIENCE Team

DEAKIN
UNIVERSITY

Reverting Changes in GitHub: A General Guide

This document provides a step-by-step guide on how to revert problematic changes in a GitHub repository, such as accidentally committed files or issues introduced by a pull request. It includes methods to identify and revert the changes using GitHub Desktop and best practices to prevent similar issues in the future.

Step-by-Step Guide to Reverting Changes

1. *Identify the Problematic Pull Request*

- Go to the "Pull Requests" tab on the GitHub web interface.
- Look for recent pull requests merged into the master branch. Review pull requests that involve unexpected changes or files, such as .icloud files, code errors, or large modifications.
- Use filters like *'is:merged'* to see only merged pull requests or *author:<username>* to find pull requests by a specific user.
- If the pull request has already been merged into the master branch, you'll need to manually revert the changes to ensure that no new modifications are lost, especially if you are behind on some commits and ahead on others.
- You can achieve this by reviewing and reverting specific commits individually.

2. *Review the Comments Inside the Pull Request*

- Open the pull request that you suspect caused the issue.
- Review any comments from team members or automated checks that might indicate concerns or issues with the changes.
- Look for comments that point out specific problems or note any unexpected files or changes.

3. *Create a New Branch:*

- Open GitHub Desktop and switch to the master branch.
- Click on "Branch" in the top menu and select "New Branch" to create a branch (e.g., revert-icloud-files).
- In the "History" tab of GitHub Desktop, right-click on each problematic commit and select "Revert This Commit".
- After selecting the erroneous commits and reverting them, go back to GitHub Desktop.
- In the "Changes" tab, review the changes to ensure they correctly undo the problematic commits.
- Commit these changes with a message like "Revert .icloud files and restore original content."

4. Push the Revert Branch to Remote Repository:

- Push the changes to your remote repository by clicking the "Push origin" button.

5.Merge the Revert Branch:

- After pushing your changes, open a pull request from the revert branch (revert-icloud-files) on GitHub.
- Review the pull request carefully to ensure all unwanted changes are reverted.
- Merge the revert branch into the master branch.

Best Practices for Identifying and Reverting Changes

1. Use GitHub Desktop and Web Together:

- Use GitHub Desktop for detailed inspection of commit history and reverting specific commits.
- Use the GitHub web interface for reviewing pull requests, comments, and overall project activity.

2. Review Comments Carefully:

- Always check for comments on pull requests that might highlight potential issues or concerns before merging.

3. Communicate with Your Team:

- Inform your team about the problematic pull request, the revert action taken, and any steps needed to prevent future issues.

Further reading

- GitHub's Search Documentation

Author

Dinuk

v1.0 2024