Evolve Full Stack Developer

Tech Appendix 2 - Git Collaboration and Merge Conflicts

INCEPTIONU

What do you mean there is only one user?

Activities

Collaborating with Git

- Activity: Create a merge conflict (on purpose)
- Activity: Commit Catch

Setting up SSH keys

GitHub has recently made changes to their authentication policy and they may force you to connect to your repos via SSH (instead of logging in via your GitHub account using HTTPS). If this happens, follow the steps below to set up your SSH keys (it's a pain but only has to be done once per machine).

- About SSH
- Check for existing SSH keys
- Generate a new SSH key and add it to the ssh-agent
- Add your SSH key to your GitHub account
- Test your connection

Setup and Prerequisites

Prerequisites

- Quality of Life Tips for a Developer
- Files, directories and naming conventions
- Command Line Basics
  - Command line practice: Follow the white rabbit

Git and GitHub Setup

- Installing Git for the first time
- Setting your Identity
- Publish a webpage with Git and GitHub Pages

Terminology

- **Automatic Merge:** When Git attempts to merge the changes of two commits into one with an overwriting commit. This is usually successful unless there are two changes to the same line(s) of code. At that point, a merge conflict must be resolved manually.
- **Merge Conflict:** When two developers, systems or branches make different changes to the same line(s) of code.
- **Resolved Conflict:** When two conflicting commits are merged by creating a new commit that overwrites the other two.
- **Unmerged Paths:** When a conflicting commit has been pulled, or merged from a branch, but has not yet been resolved with a new commit.
- **Incoming Change:** A change from a commit that has been pulled from a remote repo.
- **Current Change:** A change from a commit that has been created on the local repo.

Key Takeaways

Some things to think about when collaborating with a team member or when pushing commits from two systems:

- When working with another developer(s), the owner of the remote repo will need to add the other team member(s) as a Collaborator before they can push changes.
- You have to pull any new changes from the remote repo before you can push your own changes. If there are no new changes on the remote repo (such as when you're working alone from one machine), you can push without pulling.
- The last person/system to push code is the one that must resolve any merge conflicts. Try to push (working) code daily so that your teammates can pull up-to-date code regularly.
- Editing a file directly in the GitHub web interface counts as a different system according to Git and may create a merge conflict.
- After choosing between an Incoming and Current Change in VS Code, you still need to save the file, `add`, and `commit` these changes before you can `push` them to the remote repo. You will continue to see an "unmerged paths" error until you do.

Common Merge Errors

“failed to push some refs”

```
→  hello-conflict (main) git push
To github.com:acidtone/hello-conflict.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'github.com:acidtone/hello-conflict.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
→  hello-conflict (main)
```

Problem: Tried to push before pulling new changes from the remote repo.

Solution: `git pull` before you `git push`

INCEPTIONU

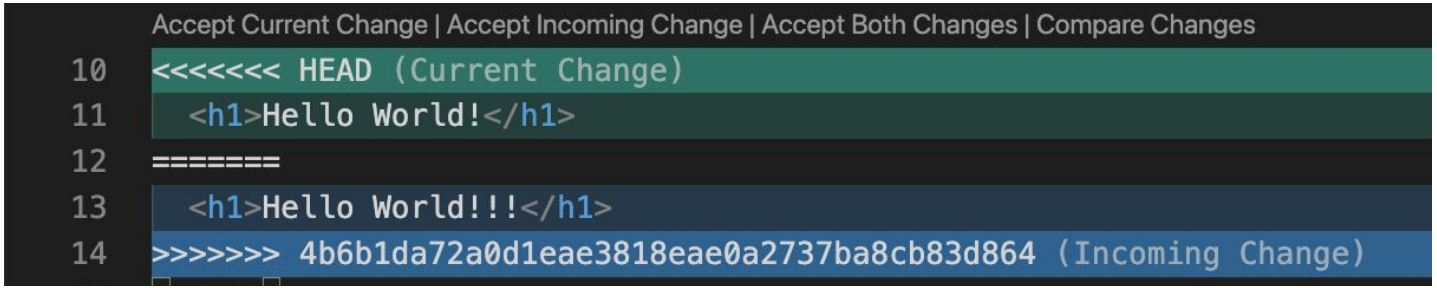
Common Merge Errors

“Automatic merge failed; fix conflicts and then commit the result.”

```
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
→  hello-conflict (main)
```

Problem: You've created a local commit that changes the same line(s) of code as an incoming commit pulled from the remote repo.

Solution: View the offending files in VS Code and resolve the conflicts using its UI.



INCEPTIONU

Common Merge Errors

“You have unmerged paths”

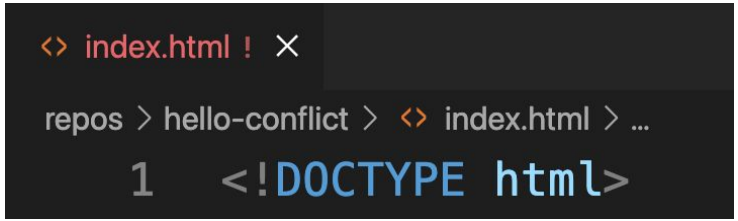
```
→  hello-conflict (main) git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
→  hello-conflict (main)
```

VS Code will display a red ! symbol next to files that have unmerged paths.



Problem: You've (probably) resolved a conflict by accepting or rejecting incoming changes but you haven't committed your new changes.

Solution: Save the offending files and `add` / `commit` these new changes as normal.

INCEPTIONU