UDACITY

Nanodegree   Catalog   Karen West

How to Use Git and GitHub

DASHBOARD

≡ Lesson 3   >   Updating Your Local Repository Quiz

CLASSROOM

MATERIALS

DISCUSSION

OVERVIEW

Updating pull request to resolve conflicts

Repeat these steps yourself:

- Simulate Sarah's changes
- Update your different-oil branch with Sarah's changes
- Push the different-oil branch

The following should be true:

- ☑ Your local master contains Sarah's changes
- ☑ Your local different-oil contains the merge commit
- ☑ The pull request contains the merge commit
- ☑ The "merge" button is green

👍 **Correct!**

Good job!

Get Help

Discussion Topics

Report an Issue

Review Instructions   Submit Answer   Continue to Answer

Instructor Notes   ## Downloadables

Simulate Sarah's changes

To simulate Sarah's changes, first download the file
`sarah_changes_2.sh` from the Downloadables section.
Then, using Git Bash or your terminal, `cd` to the directory
where you've saved the file. Then type
`bash sarah_changes_2.sh` followed by a space, followed by
the url to your fork. For example, if Caroline were running
the code, she would type
`bash sarah_changes_2.sh https://github.com/cbuckey-uda/recipes.git` ,
but you should use the url for your fork, not Caroline's
fork. If you haven't set up password-caching, then you will
be prompted to enter your GitHub username and
password.

Note: This code will not actually create a pull request.
Instead, it will update the commit history on GitHub to
make it look as though the pull request has already been
merged. You will not need to merge a pull request.
Instead, check for a commit on the master branch on
GitHub with the message "Merge pull request".

Make sure you are on the master branch

You'll want to make sure you check out the master branch
before following the steps in the instructions video.

Use `git add` on the conflicting files before committing

SECTION

sarah_changes_2   ⊕

LESSON

Videos   ⊕

Transcripts   ⊕

Just as in lesson 2, before running `git commit` to create the merge commit, you'll need to use `git add` to add any files that had conflicts to the staging area.

INFORMATION

Nanodegree Credentials

Georgia Tech Program

Udacity for Business

Udacity for Veterans

Help and FAQ

Feedback Program

COMMUNITY

Blog

News & Media

Developer API

UDACITY

About

Jobs

Contact Us

Legal

Service Status

FOLLOW US ON

MOBILE APPS

Nanodegree is a trademark of
Udacity