

Exercise: Committing the project

Overview

So far, you've set up your project using `create-react-app`. You've also set up the repository on GitHub so that you can push your commits to it, making them visible to the world or to a select few people, based on whether your repository is public or private. Additionally, you've added some VS Code extensions to help you streamline your workflow and simplified your starter app in the previous exercise. Specifically, you've:

- Removed all the code in between the returned `div` with the `className` of `App`.
- Added some text in between the opening and closing `div` tags.
- Deleted the `logo.svg` file from the app, as well as deleted its import inside the `App.js` file.

However, you haven't been instructed to commit these updates to your project yet. In this exercise, you will:

- Add your changes as another commit
- Push those changes to your remote origin (GitHub)
- Inspect the updates on GitHub

Scenario

When you add new features to the React app you are building for Little Lemon as you progress through the various course exercises and save the changes to your app's code, you need to commit and push those changes to GitHub. This makes it possible to track changes and for other developers to access your code if need be. You should then verify that you've pushed those changes by inspecting the newly-added commit on the GitHub website. Committing progress you've made with your project will feature throughout the development process.

Note: To recap key topics related to committing your project, you can revisit the [Version Control](#) course.

Instructions

Step 1: Add your changes as another commit

- Save all your updates in your local project. Make sure to save each file individually or click the **File > Save All** command in VS Code's top-level menu.
- Once that's done, open the built-in terminal and run the `git add --all` command, followed by the `git commit -m "Some commit message of your choice goes here"` command.

Tip: Make sure that the commit message is sensical, such as **Simplify the boilerplate app**.

Step 2: Push your changes to your remote origin (GitHub)

Now that you have saved and committed the changes, you can push them to GitHub.

- Use the `git push` command to push the saved and committed changes to GitHub.

Step 3: Inspect the updates on GitHub

- Open your browser and navigate to GitHub.
- Make sure you're logged in.
- Locate your project's repository.
- Ensure that your most recently pushed commit is displayed.

Tip: Sometimes it takes a few seconds for GitHub to update with your changes. Refreshing the GitHub web application might help show the updates.

Conclusion

By completing this exercise, you now have the ability to commit your changes to GitHub as you make them to your project.