Exercise: Setting up the repository

Overview

In earlier courses in the Front-End Developer program, including React Basics and Advanced React, you used create-react-app to build your own starter, boilerplate React projects. This means that, by now, you should be quite familiar with setting up a React project using VS Code and the create-react-app npm package. Furthermore, you should also understand version control best practices and be able to track and manage changes to your code. In this exercise and exercises to come, you will integrate this understanding and further expand your skillset, starting with setting up a repository with GitHub.

The following resources may help you complete this exercise:

- React Basics - Setting up a React project in VS Code
- Version Control - What is version control?
- Version Control - Create your GitHub account

Scenario

It's time for you to start building the web application for the Little Lemon website. You'll begin by setting up the project on your local machine and tracking it with Git.

Instructions

Step 1: Setup a new React app using create-react-app

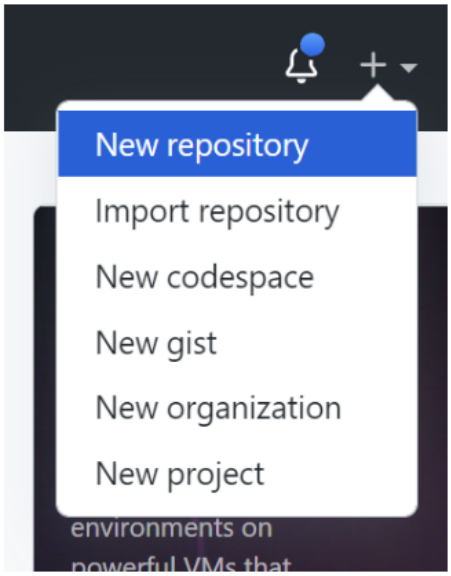
- Navigate to a folder on your machine where you want Little Lemon restaurant's web app to live.
- Start a new React project inside the chosen folder using the `npm init react-app my-app` command, where `my-app` is the name of your choice.
- For assistance, refer to the "Opening the built-in VS Code terminal and running `npm init` react-app command" section in the Setting up a React project in VS Code lesson item from the React Basics course.

Step 2: Add a new repository on GitHub

- Navigate to your GitHub account.

Tip: If you need to create an account, refer to the Create your GitHub account lesson item.

- Inside the GitHub account, create a new repository.



- Once the new repository window displays, note the URL of the repository.

Step 3: Use the `git` command to push to your repository

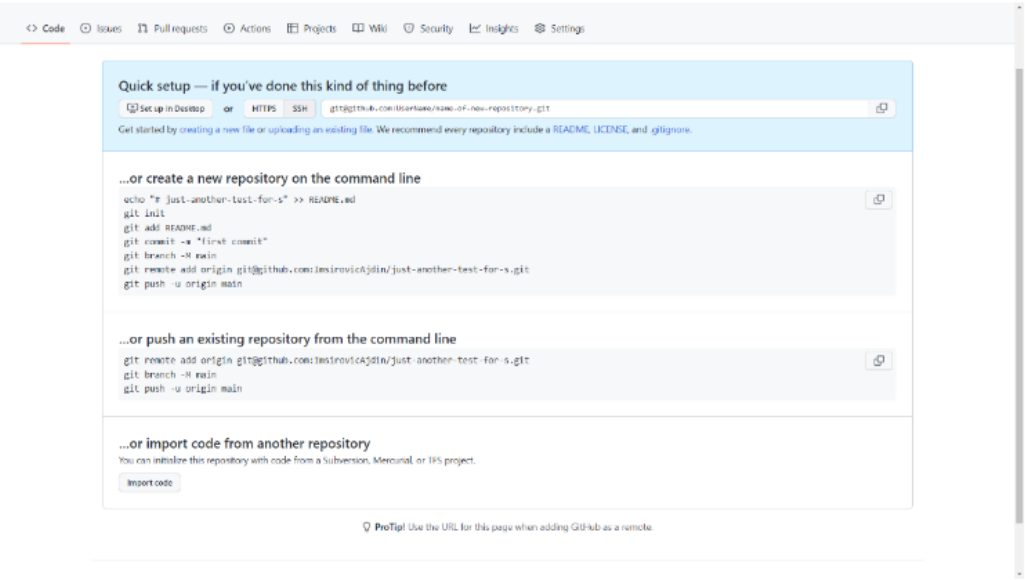
Once you've built the starter app using create-react-app, navigate to the directory using the command line. You will need to push your files to the repository on GitHub.

- Run the `git init` command to initialize the repository.
- Use the `git add` command to stage your files for commit.
- Use the `git commit` command to commit your staged changes.

Tip: You may refer to the Add and commit lesson item in the Version Control course for further assistance.

- Use the `git remote` command to add a remote for the repository on GitHub.

Tip: If you're unsure of the remote URL, view the repository on GitHub and it will provide you with the relevant information.



You may find it helpful to revisit the Connecting to GitHub via HTTPS, Connecting to GitHub via SSH and Remote vs. local lesson items in the Version Control course for additional guidance.

- Use the `git push` command to push the commits to the remote repository.

Tip: Refer to the Push and pull lesson item for further assistance.

- Run the `git status` command, then the `git log` command to confirm that Git is tracking your project locally and that you have the correct Git user set up.
- Log on to GitHub and check that your commits are visible on the repository

Conclusion

You have completed all the steps necessary to set up your project's Git. You can now push every new commit to the remote origin and share your project's code with the world.

Exercise: Setting up the React project

Overview

You previously discovered that instead of building a new React project from scratch, you can cut down on the initial steps by starting from a boilerplate project. However, you'll first want to set up a few things to make this a streamlined process. In this exercise, you will set up a React project and tweak the resulting default boilerplate.

Scenario

So far, you have been preparing to create the reserve-a-table function for the Little Lemon website. As a part of this process, it is now time to set up your React project properly.

Note: Before you begin, this exercise assumes that you are using the VS Code editor as your editor of choice. However, it is possible to use other editors to achieve similar results. If you are using another editor, you may use this exercise as a pointer and demonstration of how things are done in VS Code.

VS Code comes with many extensions that can help you when developing applications in any kind of technology and React is no different. There are numerous extensions that you have available at your disposal to make your React development a smoother and more efficient experience. In the steps that follow, you'll set up a few React-specific extensions, including:

- **ES7+ React/Redux/GraphQL/React-Native**
- **Auto Rename Tag**
- **Trailing Spaces**

You'll also clean up your starter app's code.

Instructions

Step 1: Setup the ES7+ React/Redux/GraphQL/React-Native VS Code extension

Add the ES7+ React/Redux/GraphQL/React-Native VS Code extension. This extension provides you with JavaScript and React/Redux snippets in ES7+ with Babel plugin features for VS Code.

Step 2: Setup the Auto Rename Tag VS Code extension

Add the Auto Rename Tag VS Code extension. This extension makes it possible to automatically rename the paired HTML/XML tag as Visual Studio IDE does.

Step 3: Setup the Trailing Spaces VS Code extension

Add the Trailing Spaces VS Code extension. This extension allows you to highlight and delete trailing spaces. It's also useful when working with Git to help you spot accidental whitespace addictions more easily and prevent unnecessary commits.

Step 4: Create the React project

Use `create-react-app` to create the Little Lemon project.

Tip: Refer to the previous recap reading for guidelines.

Step 5: Remove the starter app's boilerplate code from App.js

- Open the project in Visual Studio Code. `create-react-app` generates some code in the App.js file that we will not need for the project.
- Remove all the code in between the returned div with the `classname` of `App`.
- Add some text in between the opening and closing `div` tags, such as `Homepage`, for the time being.

Step 6: Remove the starter app's logo.svg file

Delete the `logo.svg` file from the app and delete its import from the App.js file.

Conclusion

By completing this exercise, you have set up and altered the default boilerplate React project in preparation to build the full application. Well done.

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.