Hands-on Lab: Create GitHub Repository for your Project

Estimated time needed: **30** minutes

Introduction

In this lab, you will set up the online repository for the project.

Objectives

After completing this lab, you will be able to:

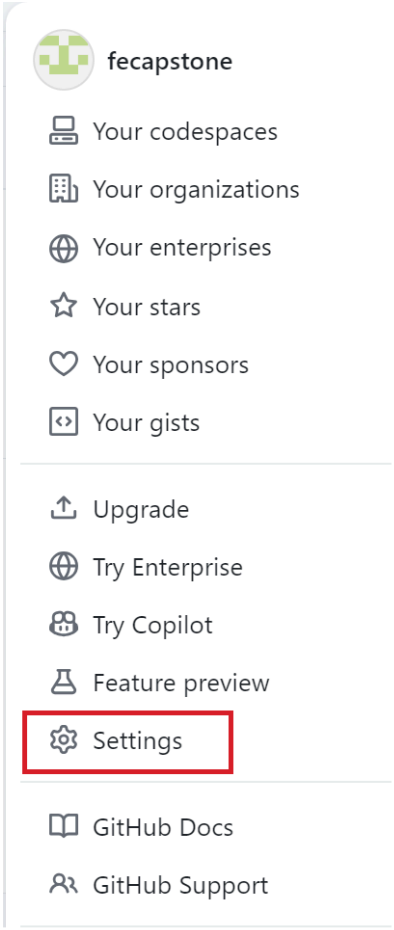
- Generate a Personal Access Token (PAT) for the GitHub repository
- Fork the source GitHub repository
- Clone your project GitHub repository in the lab environment
- Create a README.md file
- Perform `git push`

Prerequisites

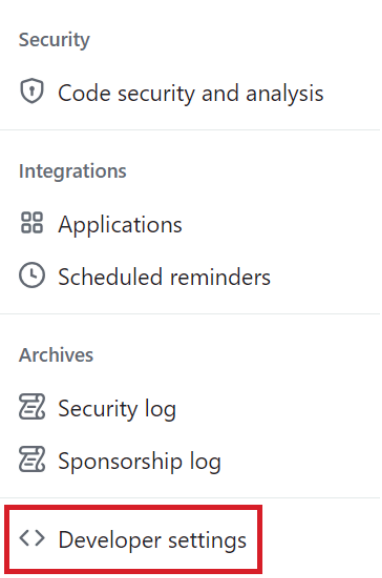
- You must have completed the prerequisite courses, especially the **Getting Started with Git and GitHub** course.
- You must have a GitHub account. If you want to set up a GitHub account, refer to the **GitHub Sign Up and Create Repo** lab for detailed steps.

Exercise 1: Create PAT to authenticate the `git commit` and `git push` commands

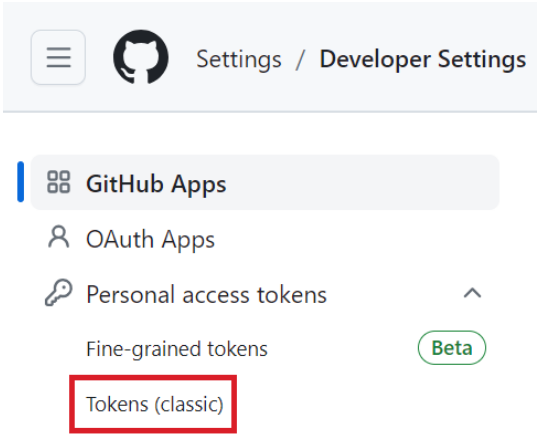
1. To begin, go to your GitHub account and click your profile icon located in the top-right corner. Then, click **Settings**.



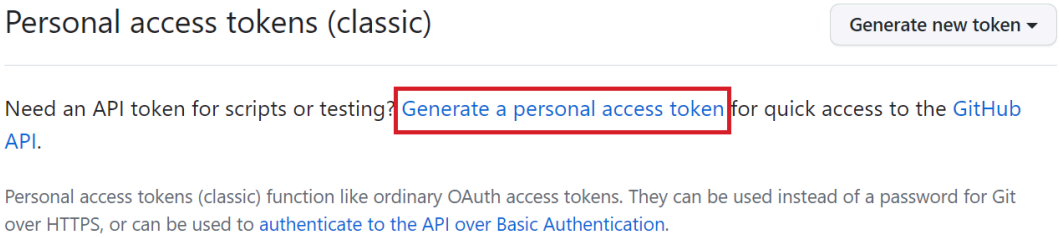
2. Next, select **Developer settings**. This option is typically available towards the bottom of the window.



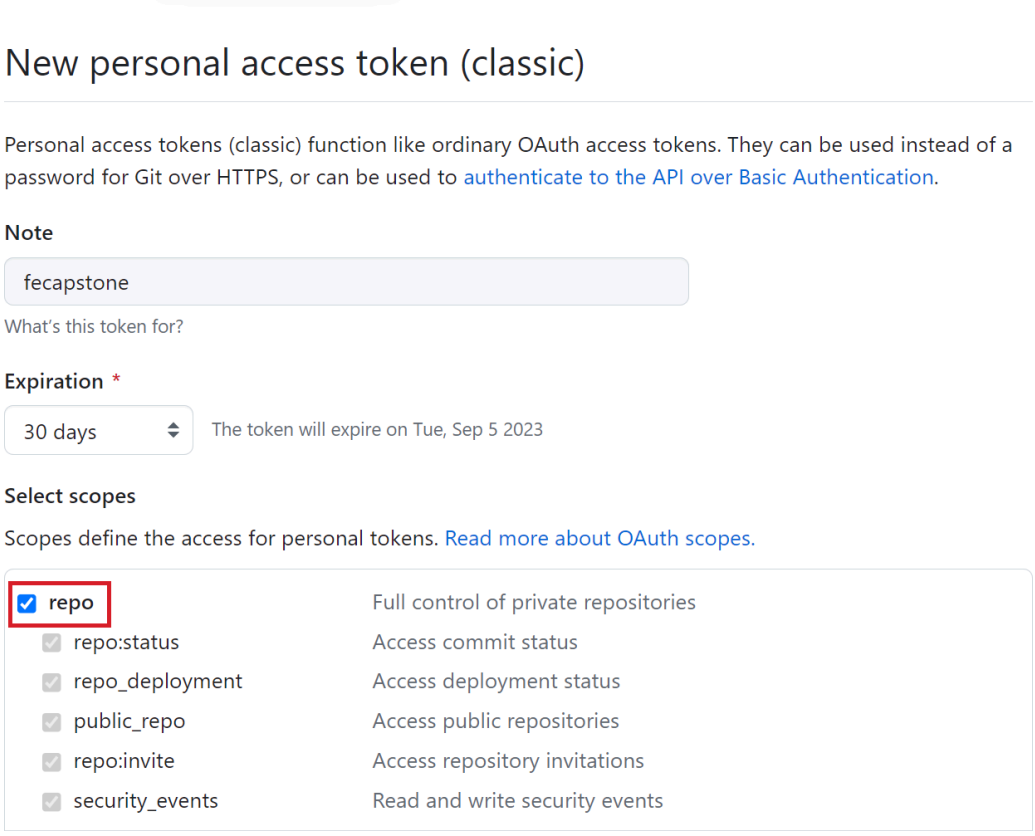
3. Navigate to **Tokens (classic)** under **Personal access tokens**.



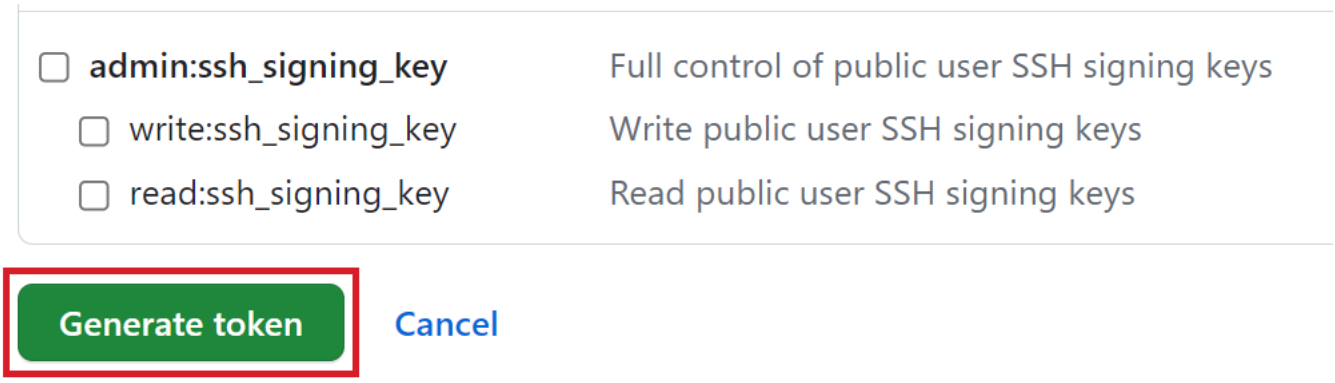
4. To generate an access token, click **Generate a personal access token**.



5. In the **Generate token** page, fill in the required details and click the **repo** checkbox to enable access for `git` commands.



6. Then, click **Generate token**.



7. Your personal access token will be generated. The token is only valid for **30 days**. You will need to generate a new token once the current token expires.

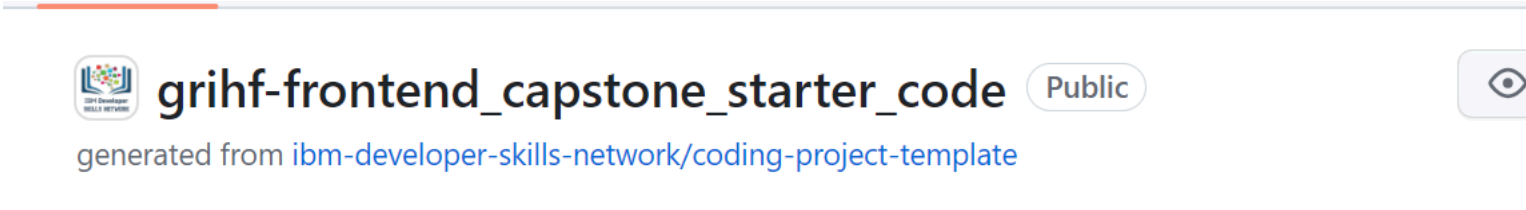
   REMEMBER: Make sure to copy your personal access token now. You won't be able to see it again!

Exercise 2: Fork the GitHub repository

1. Navigate to the source GitHub repository. This repository contains the prototype for the website's landing page in the **Landing** folder.

   ```
   https://github.com/ibm-developer-skills-network/grihf-frontend_capstone_starter_code
   ```

2. Click the **Fork** button at the top-right corner of the repository page.



Note: Ensure your forked repository is **Public**. The forked repository will be visible to anyone with the right access, including your peers during the peer review. This action will create a copy of the forked repository within your GitHub account.

Exercise 3: Clone your repository

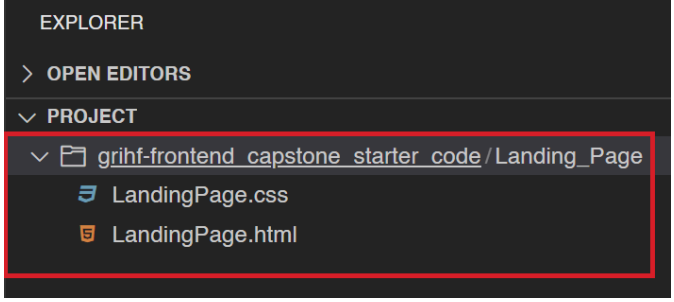
1. In the Skills Network lab environment, start a new terminal.

   1. Click the **Terminal** menu option.
   2. Then, click **New Terminal**.

2. Run the `git` command to clone the your project repository within the lab environment.

▶ Click here for a **hint**.

3. After cloning, verify that the **Landing** folder contains two files, **LandingPage.html** and **LandingPage.css**, in the lab environment.



4. You can then customize the layout design of the Landing page based on the theme of your website.

Exercise 4: Create README.md

1. Create a file named **README.md** within the **grihf-frontend_capstone_starter_code** folder. Then, add details to maintain general instructions and notes about the **Medical Appointment Booking** website in the **README.md** file.

   Note: Save the changes.

   2. **Take a screenshot** of the **README.md** file and save it as **readme.md_file.png**.

2. Navigate to the **grihf-frontend_capstone_starter_code** folder using the `cd` command in the terminal.

3. Then, set the `global commit` with your GitHub email address and username.

   ▶ Click here for **hints.**

4. Next, add and commit changes.

   ▶ Click here for **hints.**

5. Then, perform `git push` to update your forked repo with the latest changes. When prompted, enter your GitHub account username and the PAT key that you generated in Exercise 1.

   Refer to the **Capstone Project Reference: Git Commands** reading for details about Git command syntax and use relevant to the project.

7. Then, press **Enter**.

   Note: Do not worry if the PAT key is not visible when you paste it in the terminal. The key not being visible is a security feature. As soon as you press Enter, the system will start pushing the latest changes to the repository.

Screenshot checklist

You should have taken the following screenshot as part of this lab:

- *readme.md_file.png*

Note about data management and persistence

To ensure the proper management and persistence of your data in a GitHub repository, it is crucial to follow a few essential steps:

- **Regular Updates:** Whenever you make changes or add new components to your project, it is essential to add, commit, and push the updates to your GitHub repository. This ensures that your latest work is safely stored and accessible to collaborators.
- **Session Persistence:** During an active session, your data remains accessible. However, it's important to note that if your session expires or you log out, you will need to clone the repository again to resume work.
- **Ignoring node modules:** When pushing data to GitHub, it's best practice to exclude the node modules folder from both your server and client directories. This folder contains external dependencies and can be quite large, making the repository heavy and slowing down the process. By adding it to the .gitignore file, you prevent it from being pushed to the repository, keeping your commits cleaner and more focused.

By adhering to these guidelines, you can maintain a well-organized and efficient GitHub repository, ensuring that your work is securely stored and easily accessible to you and your collaborators.

Author(s)

Richa Arora

Other Contributor(s)

Rav Ahuja