

## Hands-on Lab: Cloning, committing and pushing your GitHub repo from the command line.



Effort : 30 mins

### Objectives

After completing this lab you will be able to:

1. Clone your GitHub repository locally.
2. Make changes to the cloned files.
3. Add a new file.
4. Check the status.
5. Commit changes.
6. Push the changes back to GitHub.

### Pre-requisites

GitHub account, with a project in it, as illustrated in the [this lab](#).

GitBash or git installed on your local desktop, as in [this lab](#).

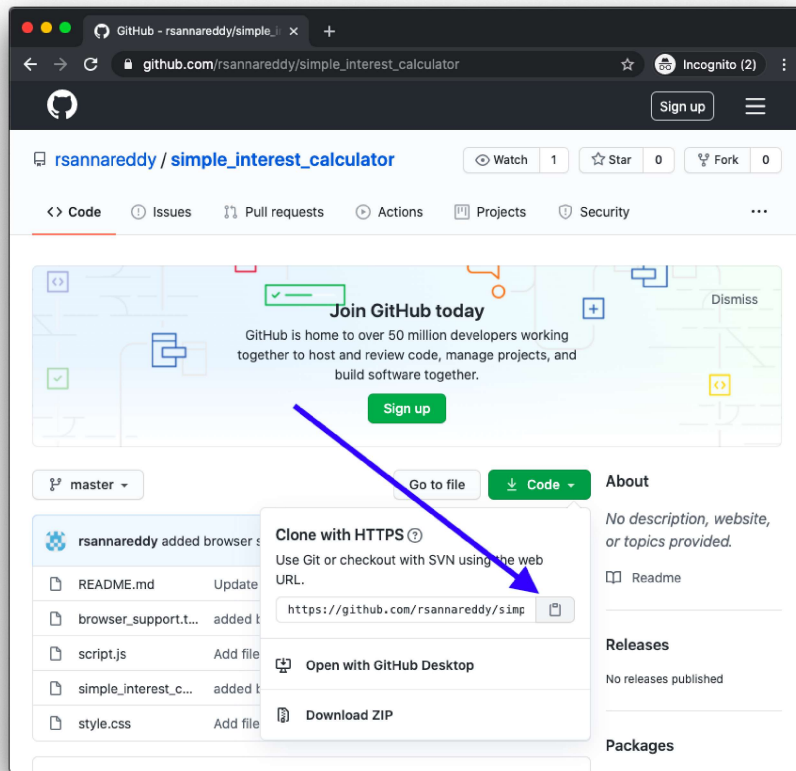
Create ssh keys, as in [this lab](#)

Add SSH Key to GitHub, as in [this lab](#)

### Exercise 1: Clone a repo

To clone a repo, you need the ssh url of the repo.

1. To get the ssh url, login into GitHub.
2. Navigate to the repo you wish to clone.
3. Click on the 'Code' button.
4. Click on the 'clipboard icon' to copy the url. Paste this url where you can access it later.



5. On your desktop open a terminal.(gitbash if you are using windows os)
6. Navigate to a directory where you wish to clone the repo.
7. Run the command "git clone <your repo ssh url>"

```
(base) sr@rameshs-air work % git clone git@github.com:rsannareddy/simple_interest_calculator.git
```

8. This will clone the repo on GitHub into your current directory.
9. You can see all the downloaded files under a directory named as your repo name.

```
(base) sr@rameshs-air work % git clone git@github.com:rsannareddy/simple_interest_calculator.git
Cloning into 'simple_interest_calculator'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
(base) sr@rameshs-air work %
```

change to the simple\_interest\_calculator directory and list the files to verify all the files got downloaded

## Exercise 2: Make changes to cloned files.

Using your favourite editor make the changes to the html file.

```
simple_interest_calculator — vi simple_interest_calc.html — 99x17
<!doctype html>
<html>
  <head>
    <title>Web App to compute Simple Interest</title>
    <script src="script.js"></script>
    <link rel="stylesheet" href="style.css">
  </head>

  <body>
    <h1>Simple Interest Calculator</h1>

    <input type="number" id="principal"> Amount <br/>
    <input type="number" id="rate"> Rate <br/>
    <input type="number" id="years"> No. of Years <br/>
    Interest : <span id="result"></span><br>

"simple_interest_calc.html" 21L, 542C written
```

git status will show all the modified files.

```
simple_interest_calculator — -zsh — 72x17
(base) sr@rameshs-air simple_interest_calculator % ls
README.md          simple_interest_calc.html
script.js          style.css
(base) sr@rameshs-air simple_interest_calculator % git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   simple_interest_calc.html

no changes added to commit (use "git add" and/or "git commit -a")
(base) sr@rameshs-air simple_interest_calculator %
```

Exercise 3: Add a new file to the local repo

Let us add a new file to the local repo.

Using a text editor, create a new file “browser-support.txt”.

Add “Chrome, Firefox, Edge” into the file.

Save the file.

Exercise 4: Check the status

Run “git status” to see info on the modified files.

Let us add the file for committing.

Run “git add browser-support.txt”

Exercise 5: Commit the changes

git commit will record all the changes into the local stating area.

To commit the changes you have made. Run git commit with a message like this.

git commit -m ‘added a new file browser-support.txt’

Now all the changes you have made thus far, get committed locally.

Excercise 6: Generate Personal Access Token

1. 1

1. Verify your email address if it hasn’t been verified on Github.

Copied!
1. 1

1. In the upper-right corner of any page, click your profile photo, then click Settings.

Copied!

Search or jump to... Pull requests Issues Marketplace Explore

**css-circle**  
forked from rsannareddy/css-circle

Code Pull requests Actions ZenHub Projects Wiki Security Insights Settings

master 1 branch 0 tags

This branch is even with rsannareddy:master. Contribute Fetch upstream

rsannareddy	Rename circle.html to index.html	95ef9d7 on 9 Sep 2020	6 commits
index.html	Rename circle.html to index.html	12 months ago	
style.css	Add files via upload	12 months ago	
supported-browsers.txt	changes made from theia lab environment	12 months ago	

About  
Create a new repository  
Releases  
No releases  
Create a new release  
Packages

Signed in as [username]  
Set status  
Your profile  
Your repositories  
Your codespaces  
Your organizations  
Your projects  
Your stars  
Your gists  
Upgrade  
Feature preview  
Help  
**Settings**  
Sign out

1. In the left sidebar, click Developer settings.
- Copied!

github.com/settings/profile

Billing & plans

Security log

Security & analysis

Emails

Notifications

Scheduled reminders

SSH and GPG keys

Repositories

Packages

Organizations

Saved replies

Applications

Developer settings

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

URL

Twitter username

Company

You can @mention your company's GitHub organization to link it.

Location

4. 1. 1  
1. In the left sidebar, click Personal access tokens and click on 'Generate Tokens'

Copied!

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

## Personal access tokens

Generate new token

Need an API token for scripts or testing? [Generate a personal access token](#) for quick access to the [GitHub API](#).

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

5. 1. 1  
1. Give your token a descriptive name. To give your token an expiration, select the Expiration drop-down menu, then click a default or use the calendar picker. Select the scopes, or permissions, you'd like to grant this token. To use your token to access repositories from the command line, select repo.
- Copied!

Settings / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

## New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

ibm\_cloudcert\_projects

What's this token for?

**Expiration \***

30 days ▾ The token will expire on Sun, Sep 19 2021

**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

☒ repo

Full control of private repositories

☒ repo:status

Access commit status

☒ repo\_deployment

Access deployment status

☒ public\_repo

Access public repositories

☒ repo:invite

Access repository invitations

☒ security\_events

Read and write security events

6. 1. 1  
1. Click Generate token and make a note of it.

Copied!

<input type="checkbox"/> <b>notifications</b>	Access notifications
<input type="checkbox"/> <b>user</b>	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> <b>delete_repo</b>	Delete repositories
<input type="checkbox"/> <b>write:discussion</b>	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> <b>admin:enterprise</b>	Full control of enterprises
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> <b>admin:pgp_key</b>	Full control of public user GPG keys ( <a href="#">Developer Preview</a> )
<input type="checkbox"/> write:pgp_key	Write public user GPG keys
<input type="checkbox"/> read:pgp_key	Read public user GPG keys

Generate token

Cancel

7. Make sure you copy the token and keep it safe. It is not visible to you again.



## Personal access tokens

[Generate new token](#)[Revoke all](#)

Tokens you have generated that can be used to access the [GitHub API](#).

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

[Delete](#)

Treat your tokens like passwords and keep them a secret.

Once you have a token, you can enter the Personal Access Token as password when performing Git operations.

### Exercise 7: Push the code to GitHub

The git push command will enable you to sync all the changes made locally to the GitHub web repository.

1. 1
  2. 2
  3. 3
  4. 4
  5. 5
  6. 6
  7. 7
1. Run the following command with your actual HTTPS link:  
2.  
3. ``git push [HTTPS link]``  
4.  
5. You will be prompted by git for your username and password.  
6.  
7. 2. Type your GitHub username and for the password, enter the personal access token you generated in the previous task. When you are authenticated, all committed changes are synced with your GitHub repository.

Copied!

```
simple_interest_calculator — -zsh — 71x11
(base) sr@rameshs-air simple_interest_calculator % git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 304 bytes | 304.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:rsannareddy/simple_interest_calculator.git
 2858800..bcf175b master -> master
(base) sr@rameshs-air simple_interest_calculator %
```

You can now visit the GitHub repository page and check to ensure that the revised and newly added files are in place.

### Summary

In this lab, you have learned how to clone a GitHub repository, make changes to it, commit the changes locally, and push it back to GitHub.

Author(s)

Ramesh Sannareddy

Other Contributor(s)

Rav Ahuja

Changelog

Date	Version	Changed by	Change Description
2023-05-11	1.4	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	1.3	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	1.2	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-10	1.1	Eric Hao & Vladislav Boyko	Updated Page Frames
2020-08-23	1	Ramesh Sannareddy	Initia version created.