Configuring git & GitHub to communicate with RStudio

You should have already created a GitHub account, and checked for git on your computer, in Assignment 1. If you have *not*, you'll need to do that before starting here.

1. Configure git

- Open RStudio, and open the 'Terminal' tab (next to the Console tab). Run the following commands in the Terminal (type in, then press Enter after each line), **replacing the username and email (keep the quotation marks for each) with <u>your username and email used for your GitHub account</u>**.

  ```
  git config --global user.name "Jane Doe"

  git config --global user.email "janedoe@example.com"
  ```

- Next, in the Terminal run the following, carefully checking that the name and email returned match your GitHub information. If your username and email are returned, great! You've configured git successfully. If not, come see us in office hours.

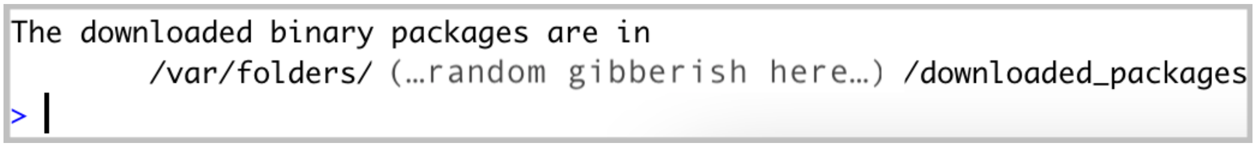
  ```
  git config --list --global
  ```

2. Store your GitHub personal access token (PAT)

- Once you have git configured successfully in the steps above, install the `usethis` package in R by running the following in the RStudio Console:

  ```
  install.packages("usethis")
  ```

A lot of scary looking red text will show up while this is installing - don't panic. If you get to the end and see something like below (with no error) it's installed successfully.

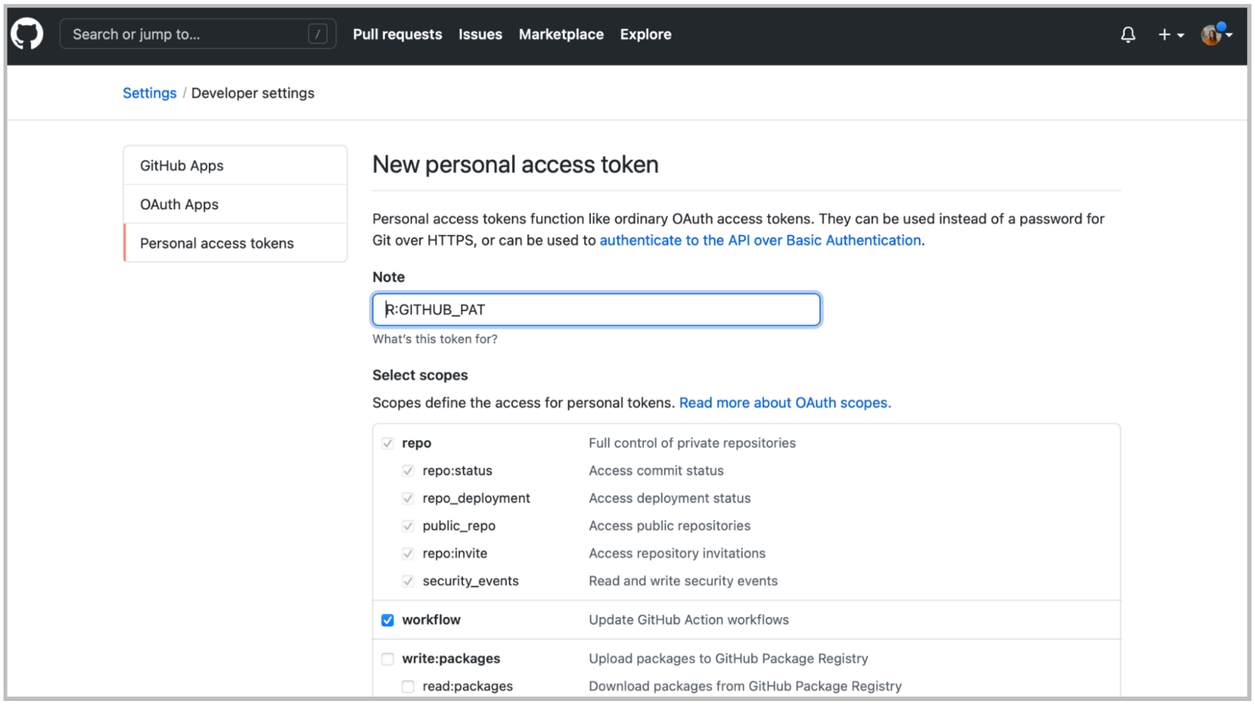


```
The downloaded binary packages are in
        /var/folders/ (…random gibberish here…) /downloaded_packages
> |
```

- Run the following in the RStudio Console:

  ```
  usethis::create_github_token()
  ```

- This should automatically take you back to GitHub. Enter your GitHub password when prompted. You'll be taken to a screen that looks like this, with the right options already auto-selected for you:



Settings / Developer settings

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

R:GITHUB_PAT

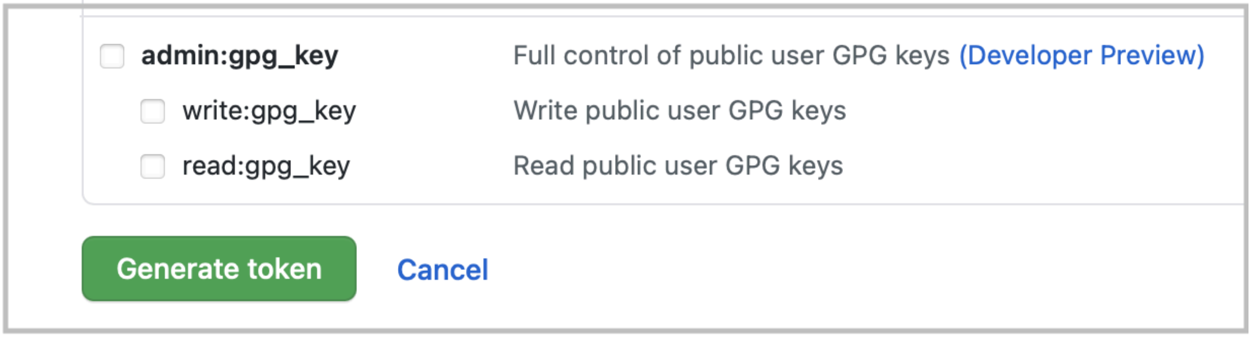
What's this token for?

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
☑ workflow	Update GitHub Action workflows
□ write:packages	Upload packages to GitHub Package Registry
□ read:packages	Download packages from GitHub Package Registry

- Do not change anything. Just scroll down to the bottom of that page and click the green 'Generate token' button:

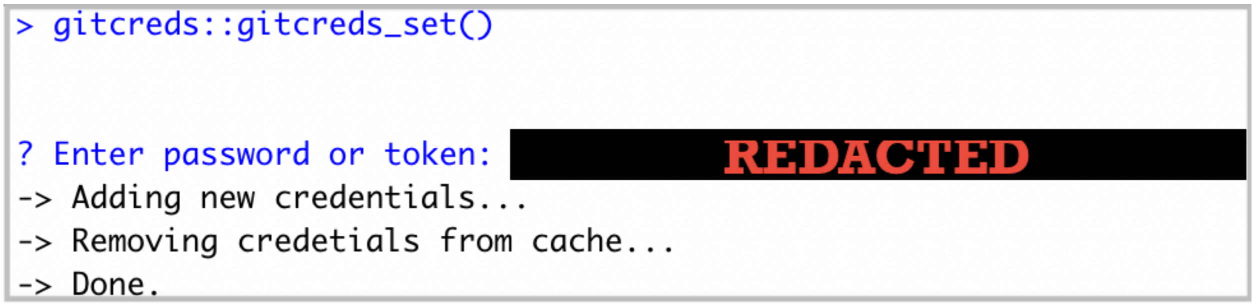


□ admin:gpg_key	Full control of public user GPG keys (Developer Preview)
□ write:gpg_key	Write public user GPG keys
□ read:gpg_key	Read public user GPG keys

- Copy the generated token to your clipboard
- Back in RStudio, in the Console run the following command:

```
gitcreds::gitcreds_set()
```

This will prompt you to paste the token you just copied from GitHub. Paste the PAT, press Enter. You should see something like this show up if all is well so far (you'll have pasted your PAT where mine says "REDACTED"):



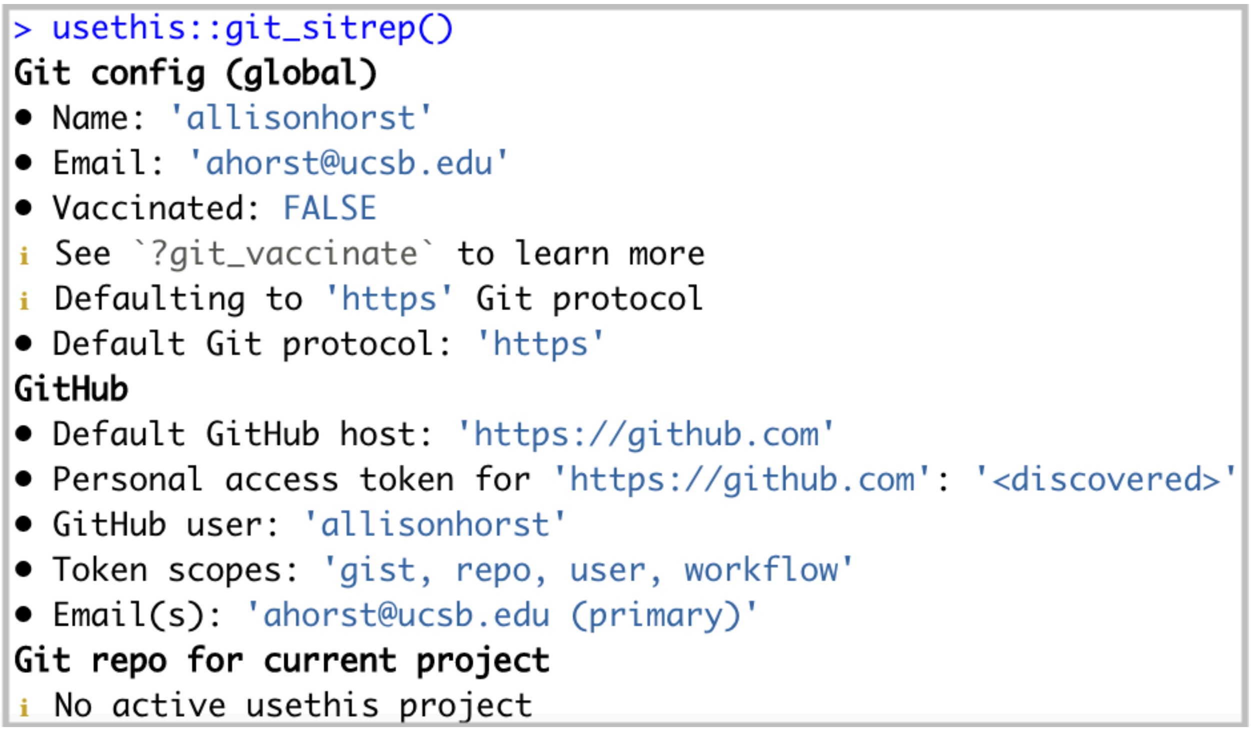
```
> gitcreds::gitcreds_set()


? Enter password or token: REDACTED
-> Adding new credentials...
-> Removing credetials from cache...
-> Done.
```

- In the RStudio Console, run:

```
usethis::git_sitrep()
```

Does it return information about your connected GitHub account that looks something like below? Great! You've configured git and successfully stored your PAT.



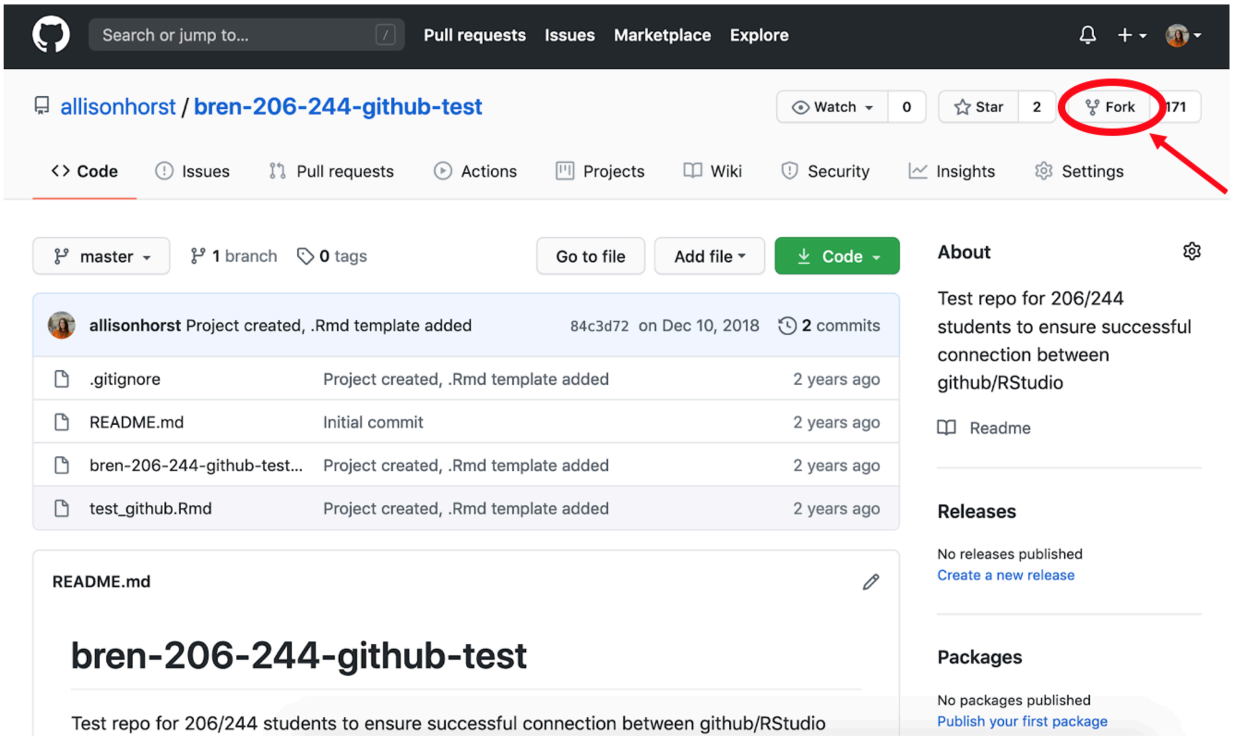
```
> usethis::git_sitrep()
Git config (global)
• Name: 'allisonhorst'
• Email: 'ahorst@ucsb.edu'
• Vaccinated: FALSE
ℹ See `?git_vaccinate` to learn more
ℹ Defaulting to 'https' Git protocol
• Default Git protocol: 'https'
GitHub
• Default GitHub host: 'https://github.com'
• Personal access token for 'https://github.com': '<discovered>'
• GitHub user: 'allisonhorst'
• Token scopes: 'gist, repo, user, workflow'
• Email(s): 'ahorst@ucsb.edu (primary)'
Git repo for current project
ℹ No active usethis project
```

3. Test that everything is working

1. Log-in to your GitHub account

2. Go to my repository **bren-206-244-github-test**. Notice that there are some files in the repo.

3. Click ‘Fork’ to create your own copy of the repo in your account (you don’t have permissions to change anything in mine, so don’t worry about wrecking it):



allisonhorst Project created, .Rmd template added 84c3d72 on Dec 10, 2018 2 commits

.gitignore	Project created, .Rmd template added	2 years ago
README.md	Initial commit	2 years ago
bren-206-244-github-test...	Project created, .Rmd template added	2 years ago
test_github.Rmd	Project created, .Rmd template added	2 years ago

Test repo for 206/244 students to ensure successful connection between github/RStudio

No releases published
Create a new release

No packages published
Publish your first package

README.md

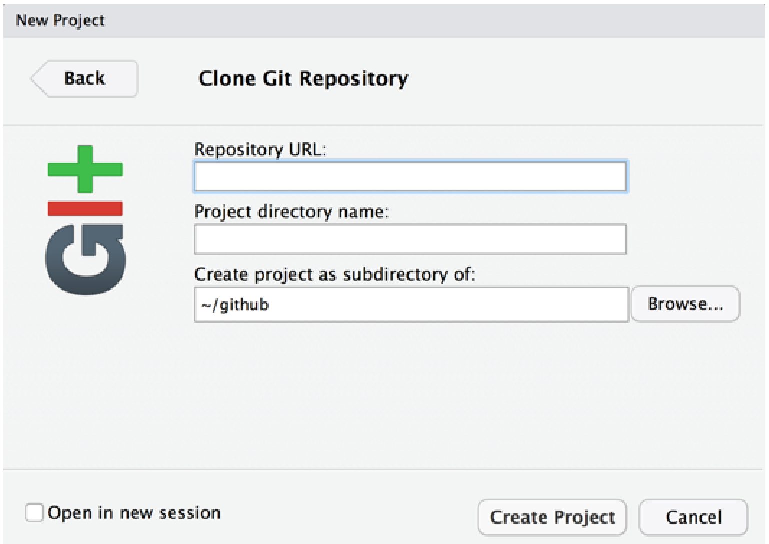
bren-206-244-github-test

Test repo for 206/244 students to ensure successful connection between github/RStudio

4. Once you fork it, now you have your own copy of that repo. But you still need to get it talking with RStudio so you can actually work with it. After forking, **click on the green ‘Code’ button**, then follow these instructions **carefully:**

a. When you click ‘Code’, you should see a URL in the drop-down info. That URL is going to be the pathway that github and RStudio use to talk to each other**. Copy that link.**

b. Open (or go back to) RStudio. Choose: File > New Project > Version Control > Git, which gets you to a screen like this:



Clone Git Repository

Repository URL:

Project directory name:

Create project as subdirectory of: ~/github

☐ Open in new session

d. Paste the URL (that you copied from github when you pressed ‘Code’) into the **Repository URL:** section, then…

e. If it didn’t already auto-populate, **Press TAB** to populate the **Project directory name:** section with the correct (matching) project named linked to the github repo, then…

f. Choose where you want the version controlled project to live on your computer (probably within your ESM 206 folder somewhere…)

g. Press **Create Project** (optional: select ‘Open in new session’ if you’re already working on stuff that you don’t want to close)

h. What you’ve just created is a git-enabled project that contains all of the files contained in the forked repo, now sitting right in your working directory waiting for you to work with them.

6. Make some changes to the markdown document

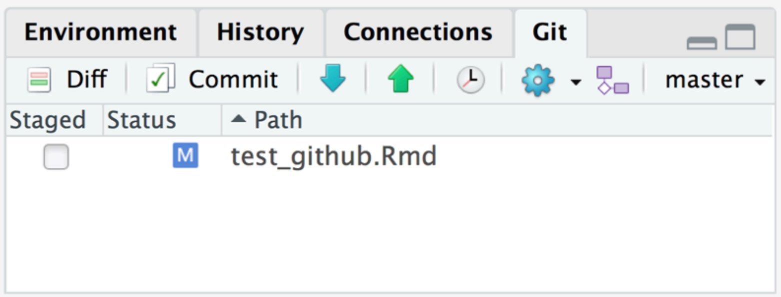
   a. In RStudio, now click on the file ‘test_github.Rmd’ in the “Files” tab to open it

   b. Make minor changes to the markdown document (add your name, change some variable names where prompted, etc.)

   c. Save the updated .Rmd

7. Go to the Git tab in RStudio (next to the Environment tab). Any updated files/projects will show up in the ‘Git’ tab in RStudio (probably just the .Rmd this time).

- To **stage** the updated file, select the empty box (under 'Staged') next to any file that appears there
- **Commit** the staged files by pressing the "Commit" button, then entering a brief commit message (e.g. "testing connection"), then press 'Commit.' No error = working. Close that window.
- After committing, press the upward green arrow to **Push** changes back to GitHub. No error = working. You **may** need to enter your github username/password to do this.



Staged	Status	Path
□	M	test_github.Rmd

8. Ensure that you successfully pushed back to github: Go to your GitHub account, and refresh the repo. Do you see recent commits show up? Cool! That means they've been pushed back successfully. Now, click on the test_github.Rmd file in your GitHub repo to see the code contained – does it contain your updates? GREAT. That's it!

NOT WORKING? Visit us in office hours next week.