Pre-training actions

Before September 22, please take the actions below. Do not worry if you have any issues or if you are unable to install something. If that is the case, please join the pre-training office hour on Monday, September 20 at 3pm (BST) and we will get you sort it out. The zoom links to join all sessions will be provided via email.

Table of Contents

Sign up for a Zoom account

All sessions and office hours will be hosted on Zoom, a popular video conferencing platform. Thus, if you do not have a Zoom account, you can sign up for free here. To join the Zoom video conferences, first log in to your account and then click on the link provided via email.

Instructions to install Git in MACs and WINDOWS

FOR MACS:

1) Create an account on Github.com.
    a. Please remember your username and password, we are going to need this info later.

2) Go to Homebrew and read the instructions given. More details in the steps below:

3) Go to the terminal (black box) in your Mac, copy and paste the following:

$ /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

It might ask you for the password you use when apps are installed. Example:

```
Last login: Sun Jul 25 10:07:18 on ttys000
(base) nas-10-240-185-251:~ andrea$ /bin/bash -c "$(curl -fsSL https://raw.githu
busercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password).
Password:
==> You are using macOS 10.13.
```

This installation will take around 2-4 minutes, at some point it will ask you to press RETURN to continue, so press RETURN

```
==> The Xcode Command Line Tools will be installed.

Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/chmod u+rwx /usr/local/bin /usr/local/include /usr/local/
lib /usr/local/share /usr/local/lib/pkgconfig /usr/local/share/info /usr/local/s
hare/man /usr/local/share/man/man1 /usr/local/share/man/man3 /usr/local/share/ma
n/man5 /usr/local/share/man/man7
```

4) Once you have Homebrew installed, you can download Git by typing the following into the command line: **(the command line is sensitive to spaces)**

brew install git

```
(base) nas-10-240-185-251:~ andrea$ brew install git
Warning: You are using macOS 10.13.
```

It will take around three minutes to get everything sorted.

5) You can check whether git was properly installed by typing (after "git" there is a space):
git --version

```
(base) nas-10-240-185-251:~ andrea$ git --version
git version 2.15.0
```

6) Now you need to link your account, in the terminal, execute the following commands, replacing your_*username* with the git username you just created and the email that is linked to your account:

git config --global user.name "your_username"
git config --global user.email "your_email_address@example.com"

```
(base) nas-10-240-185-251:~ andrea$ git config --global user.name "aso111"
(base) nas-10-240-185-251:~ andrea$ git config --global user.email"andrea.s@cisi
dat.org.mx"
```

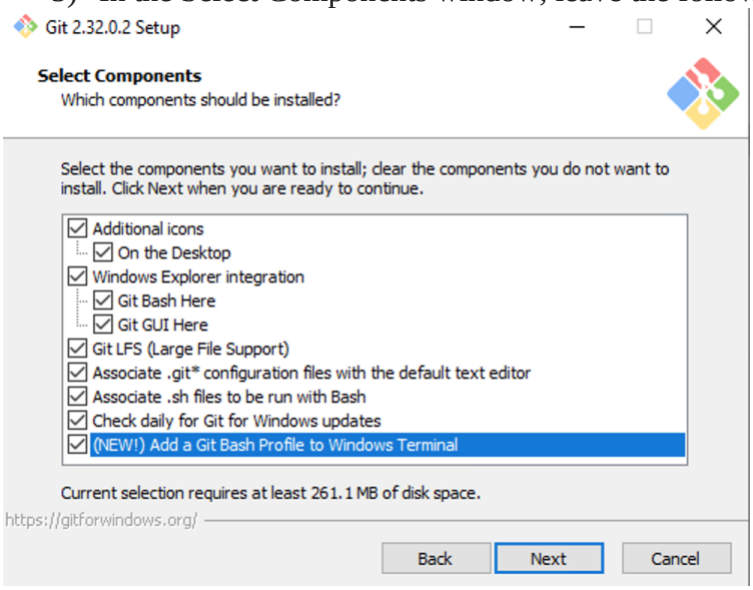
7) Test that you have linked your account by typing:
git config --global --list

```
(base) nas-10-240-185-251:Test_Jul25 andrea$ git config --global --list
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=aso111
user.email=andrea.s@cisidat.org.mx
```

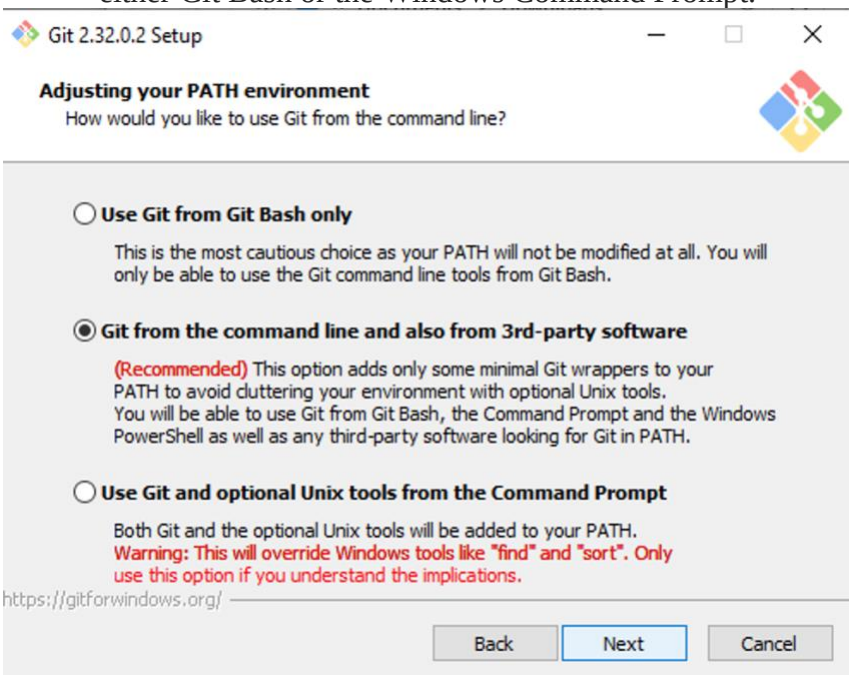
If you can see your username and user email, it is done ☺

For WINDOWS

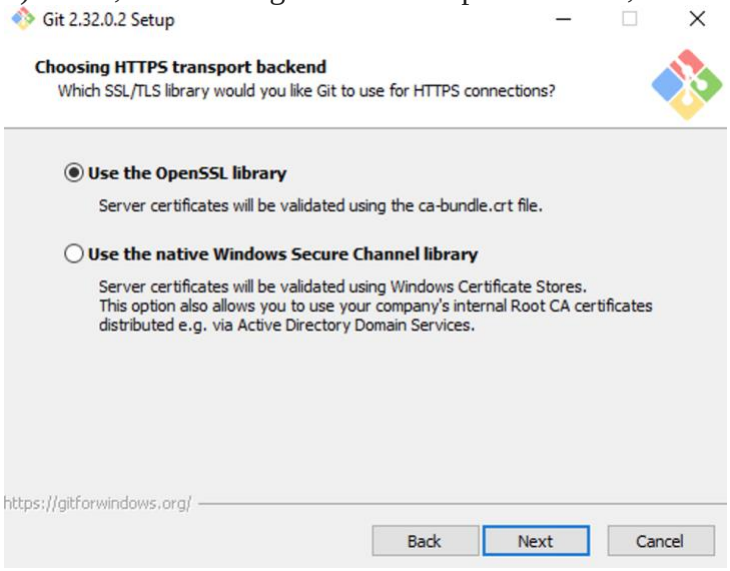
1) Create an account on Github.com.
    a. Please remember your username and password, we are going to need this info later.
2) Go to https://git-scm.com/download/win
3) Click the Download link to download Git. The download should automatically start.
4) Once downloaded, start the installation from the browser or the download folder.
5) In the Select Components window, leave the following options:



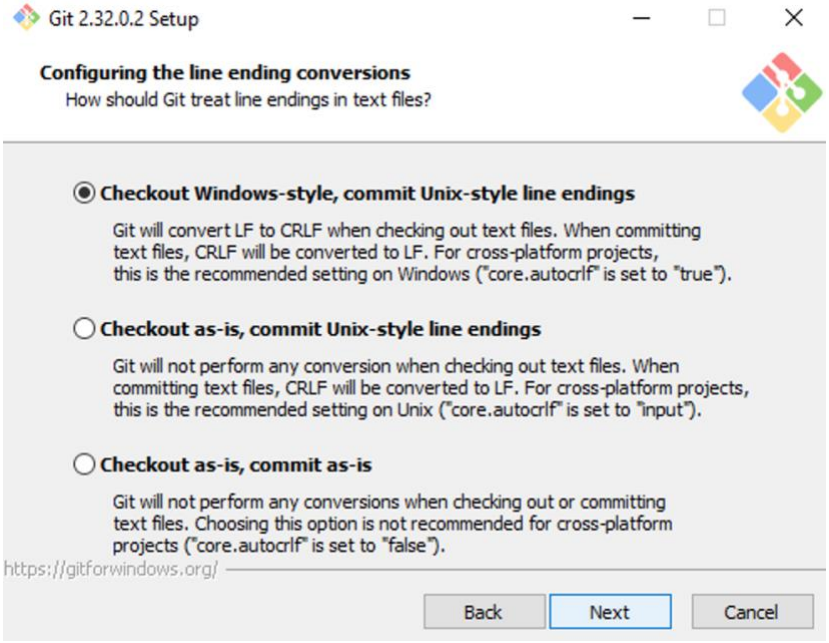
6) Next, in the Adjusting your PATH environment, we recommend keeping the default Use Git from the command line and also from 3rd-party software as shown below. This option allows you to use Git from either Git Bash or the Windows Command Prompt.



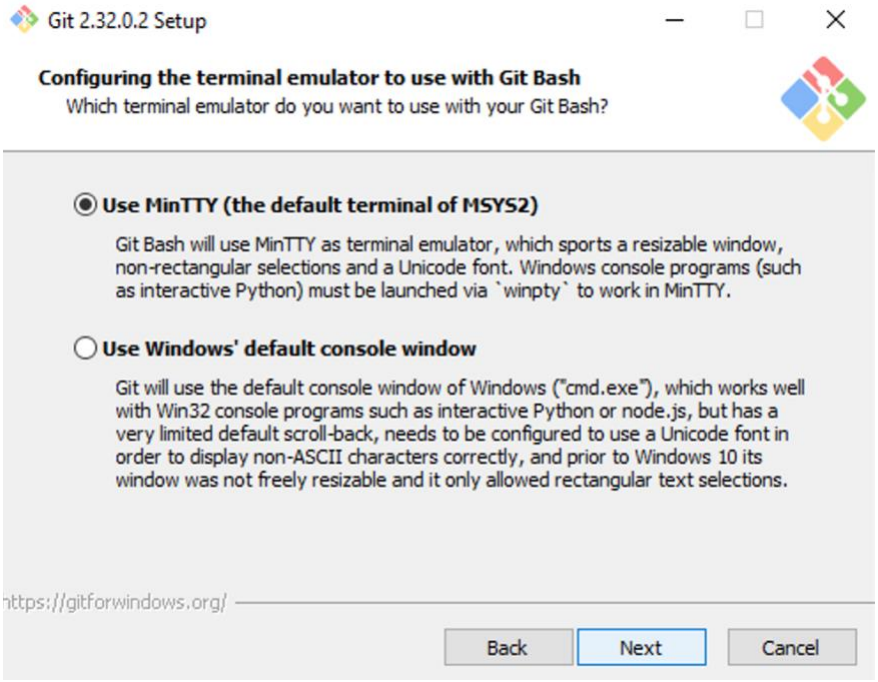
7) Next, in Choosing HTTPS transport backend, leave the default Use the OpenSSL library selected.



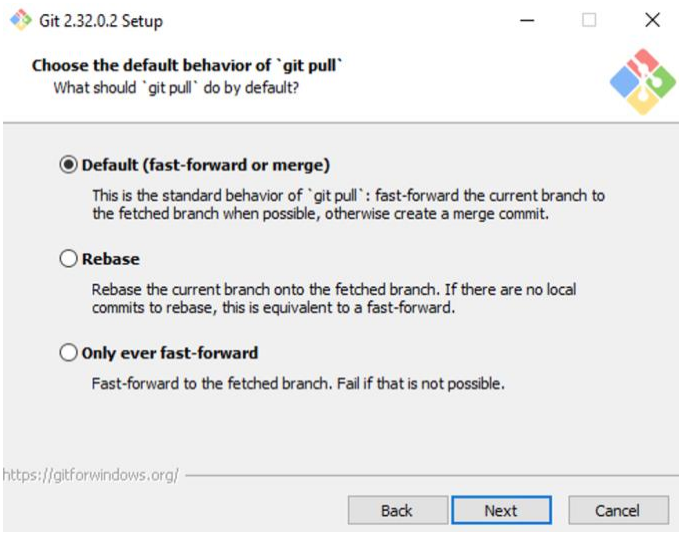
8) In the Configuring the line ending conversions, select Checkout Windows-style, commit Unix-style line endings unless you need other line endings for your work.



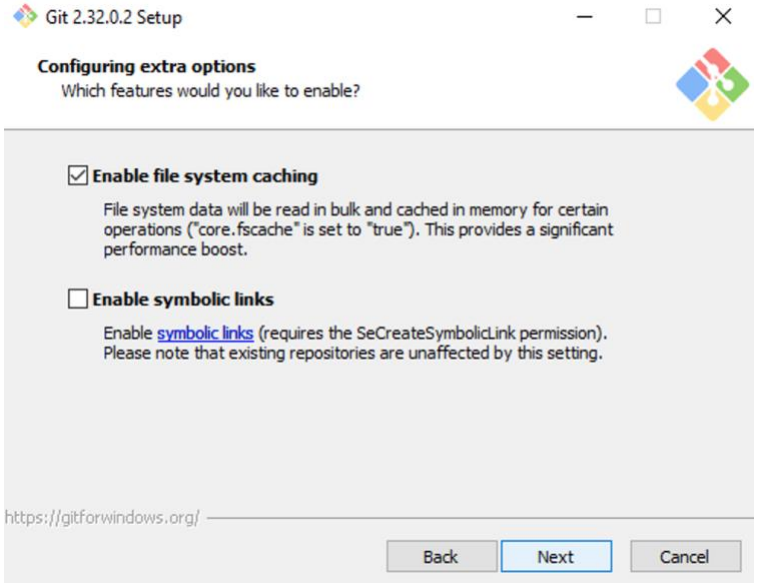
9) In the Configuring the terminal emulator to use with Git Bash window, select Use MinTTY (the default terminal of MSYS2).



10) What about git pull

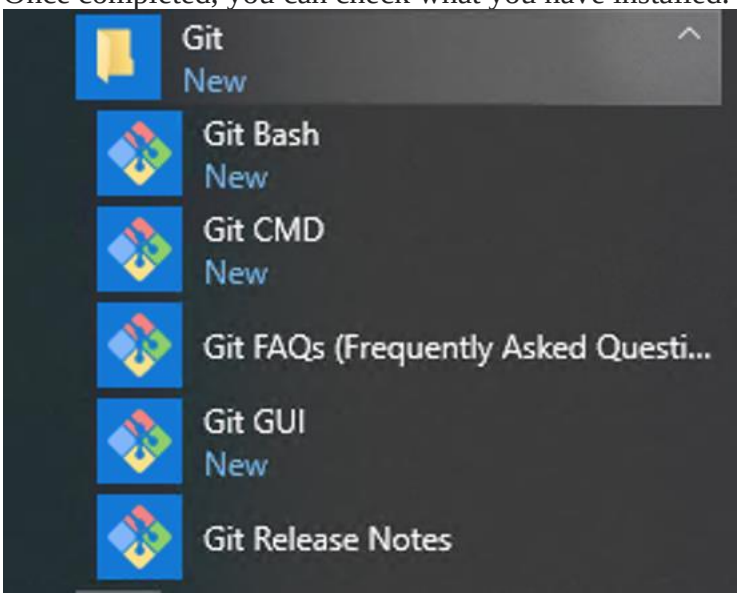


11) On the Configuring extra options window, leave the default options



12) Click the Install button

13) Once completed, you can check what you have installed.



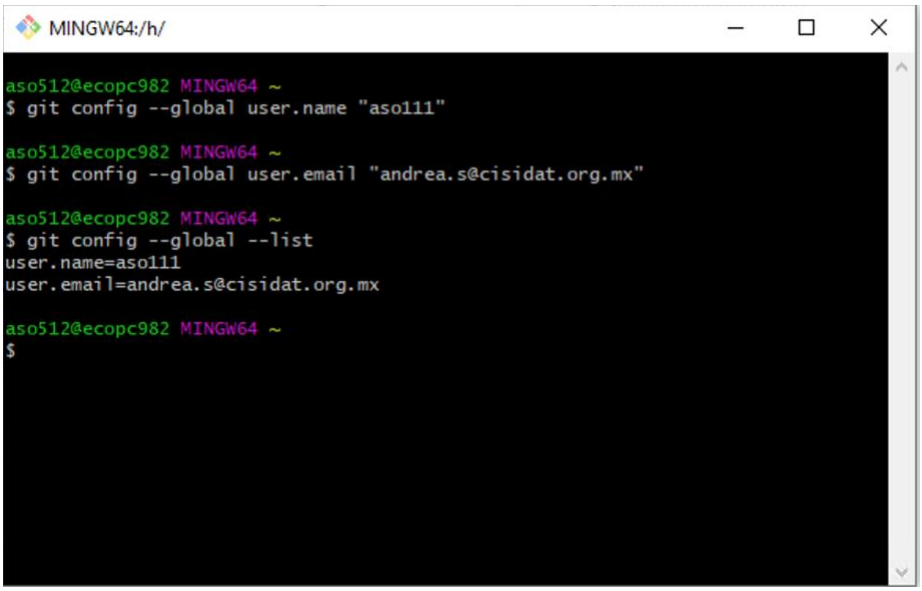
14) To link your account, open Git Bash and execute the following commands, replacing your_*username* with the git username you just created and the email that is linked to your account:

```
git config --global user.name "your_username"
git config --global user.email "your_email_address@example.com"
```

15) Test that you have linked your account by typing:

```
git config --global –list
```

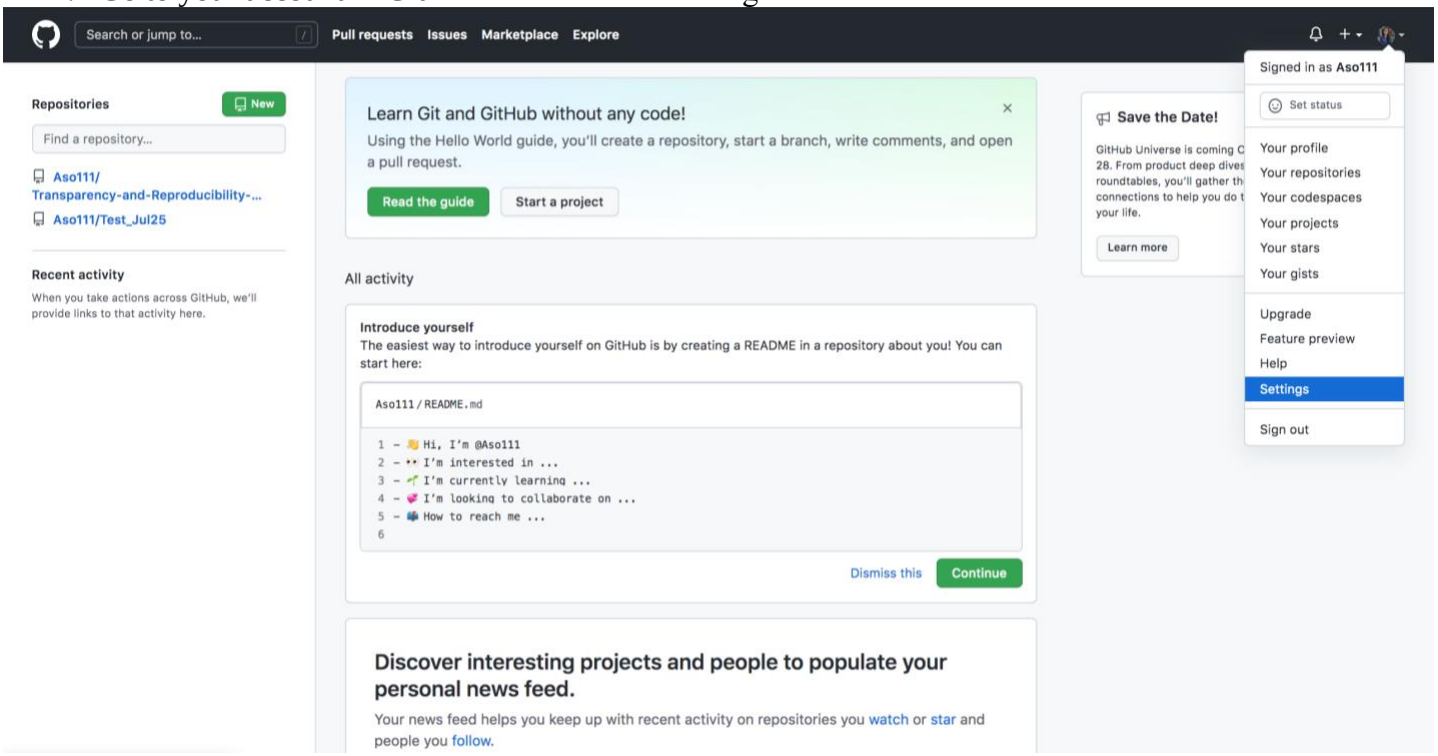
16) And once you can see your username and user email, it is done ☺


MINGW64:/h/
aso512@ecopc982 MINGW64 ~
$ git config --global user.name "aso111"
aso512@ecopc982 MINGW64 ~
$ git config --global user.email "andrea.s@cisidat.org.mx"
aso512@ecopc982 MINGW64 ~
$ git config --global --list
user.name=aso111
user.email=andrea.s@cisidat.org.mx
aso512@ecopc982 MINGW64 ~
$

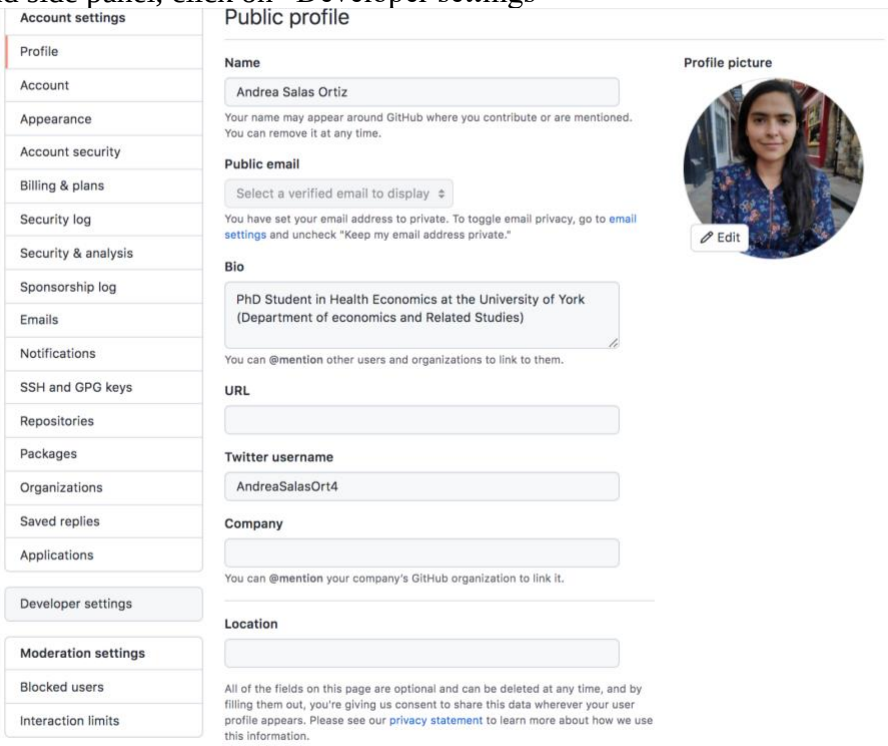
Getting a TOKEN from GitHub

This must be done during the workshop. This is just to show you what you will need to do.

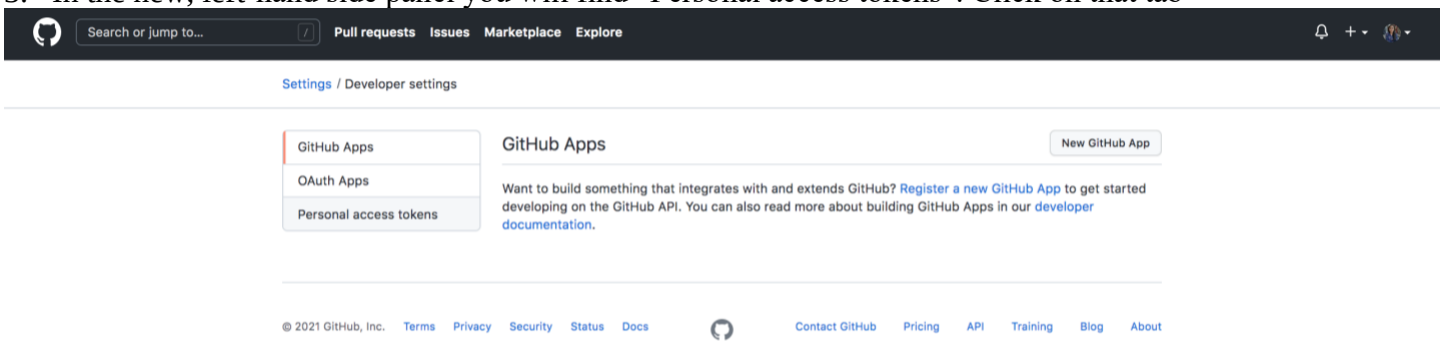
1. Go to your account in GitHub and click on “Settings”



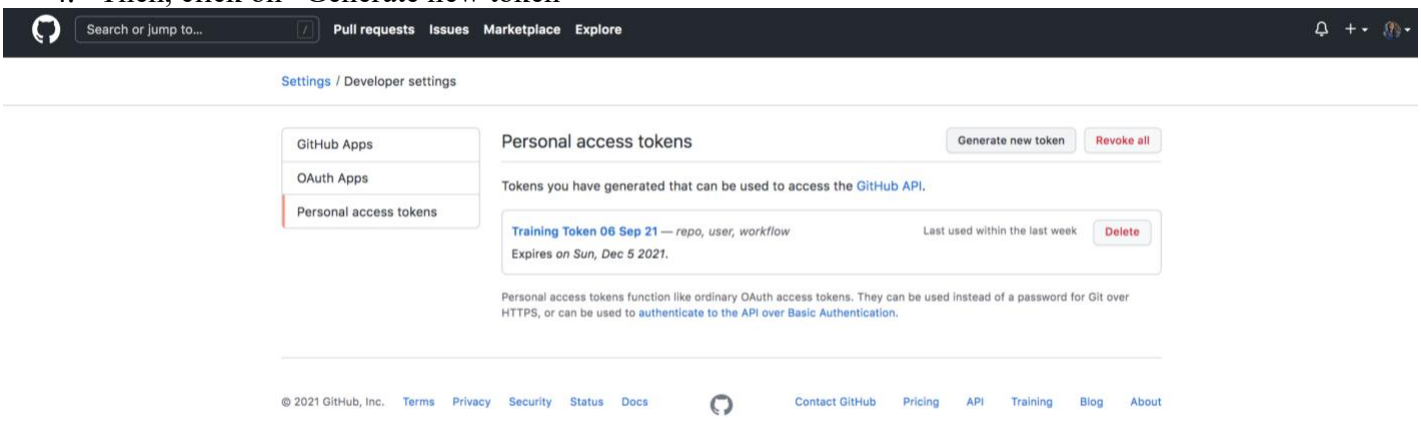
2. In the left-hand side panel, click on “Developer settings”



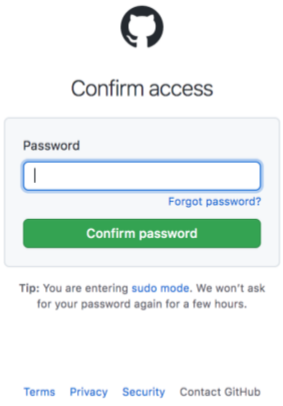
3. In the new, left-hand side panel you will find “Personal access tokens”. Click on that tab



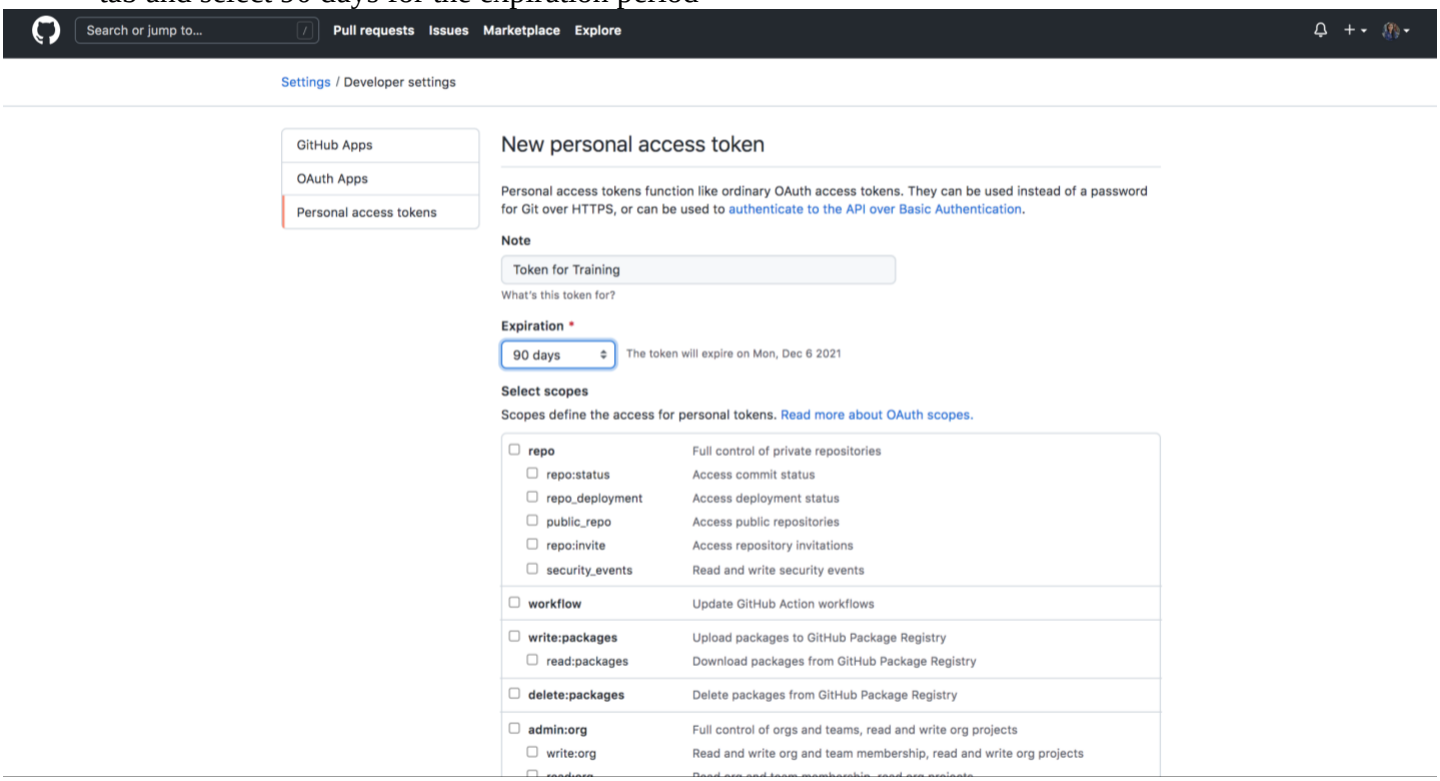
4. Then, click on “Generate new token”



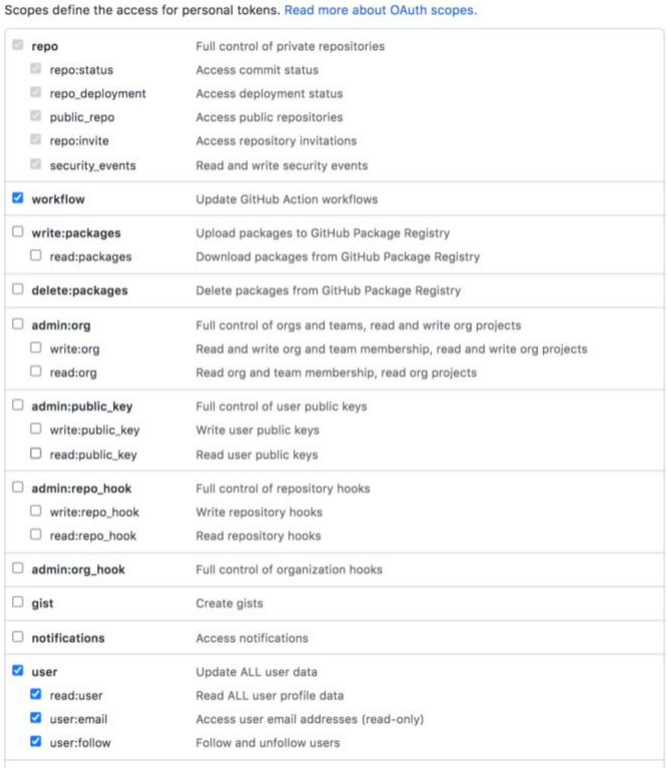
5. This will need you to insert your password



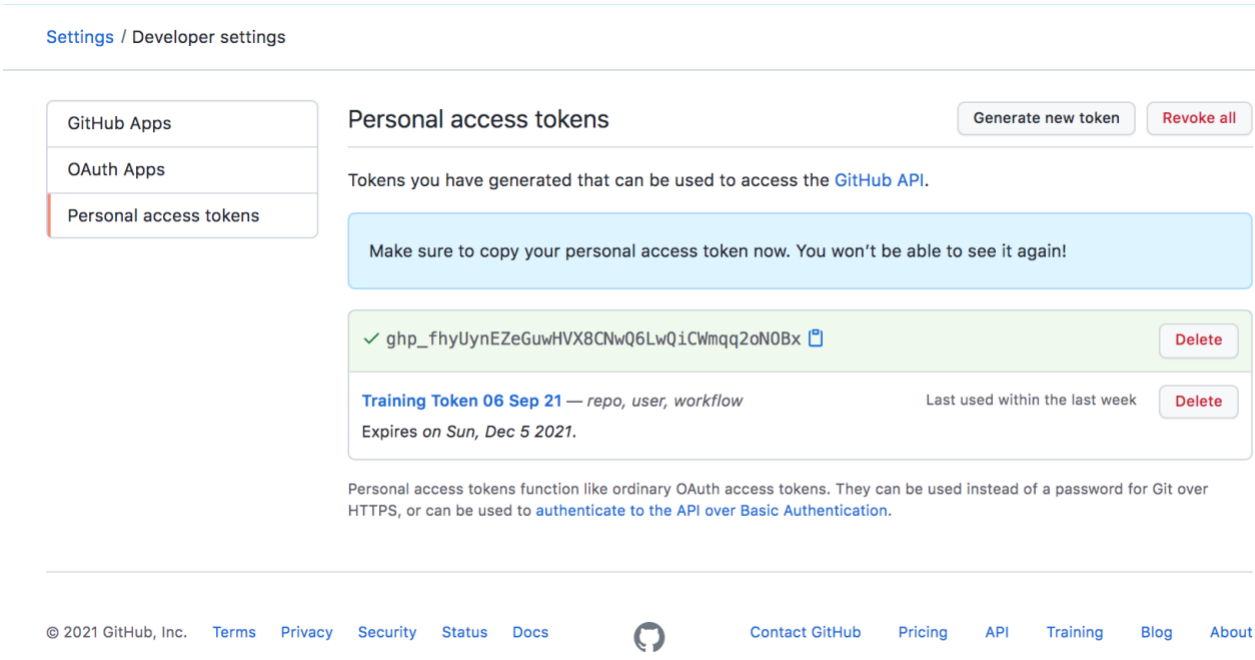
6. The new TOKEN must be given a purpose and an expiration period. Please add the reason in the Note tab and select 90 days for the expiration period



7. Then, select the: “repo”, “workflow” and “user” tabs



8. The personal TOKEN has been created. Please copy the TOKEN and paste it in a safe place, as the TOKEN will be displayed one time only



Instructions for the "Dynamic Documents in R and Stata" sessions

Dataset to be used in the exercises

The aim of these sessions is to show you how to create "Dynamic documents". We will do the **same** exercise using the same dataset, but different software (R and Stata). You can attend both sessions if you want.
To get the dataset:

1) Please go to the HEDG webpage (https://www.york.ac.uk/economics/hedg/software/) and download the "AHE_2ed_Ch_3.dta" dataset
2) We recommend creating a new folder and saving this dataset there. This folder will be the directory to be changed in both, the R-script and Stata-Dofile that will be shared with you

Dynamic documents in R

To use R Markdown, you will need R and RStudio installed. You can download and install them using the following links:

1. R here: https://cran.r-project.org/
2. RStudio here: https://www.rstudio.com/products/RStudio/

Please make sure you have a version >= 4.0.5.

Once you've got RStudio open, you will need to install the following packages. To get them, type the following into the console:

```
install.packages("rmarkdown")
install.packages("foreign")
install.packages("readstata13")
install.packages("ggplot2")
install.packages("stargazer")
install.packages("coefplot")
install.packages("doBy")
install.packages("kableExtra")
```

Dynamic documents in STATA

For the Stata session, you will be required to already have access to Stata. Unfortunately, we cannot provide a temporarily access, sorry! ☹

To be able to create dynamic documents in Stata, you will need to install the following commands:

- *ssc install coefplot*
- *ssc install outreg2*
- *ssc install codebookout*

We will also need the *markdoc* package, but this is in a Git repository. So, in order to get it, we will first install a command that will allow us to get packages hosted in Git **from** Stata.

1) In the command tab in your Stata type:
net install github, from("https://haghish.github.io/github/")

2) To install a package, all you need is the GitHub username and the name of the repository. The combination of username and repository name - seperated by a slash - provides the needed URL to the repository.
To install the markdoc package, which is hosted on https://github.com/haghish/markdoc, you need to type:
github install haghish/markdoc, force

- To make sure that you have installed *markdoc*, please run the following into your STATA's console:
markdoc example, test

- Click on "(markdoc created **example.pdf**)" and if installed correctly, you should get a pdf document that runs the following heading "Testing markdoc Package".

Hopefully, you were able to install and do all smoothly, otherwise please come to the "pre-training office hour" and I can help!