Leibniz Institute for the Social Sciences

Tools and Workflows for Reproducible Research in the Quantitative Social Sciences

Using Git and GitHub

Bernd Weiß

2022-11-18

Leibniz Association

Overview

Recap

- Basic introduction to Git
- Git workflow
- Important Git concepts/commands (`git init`, `git add`, `git commit`, `git log`, `git show`, `git clone`, `git checkout`, `git branch`, ...)
- ...

Please, let me in (authentication)

Local or remote & HTTPS or SSH?

- A Git project is stored in a repository, which can be local or remote
- When using Git to access a remote repository (for backup or collaborative work) on a remote server, you need to authenticate yourself to the server
- There are two ways of authentication: HTTPS or SSH

Choose a remote server

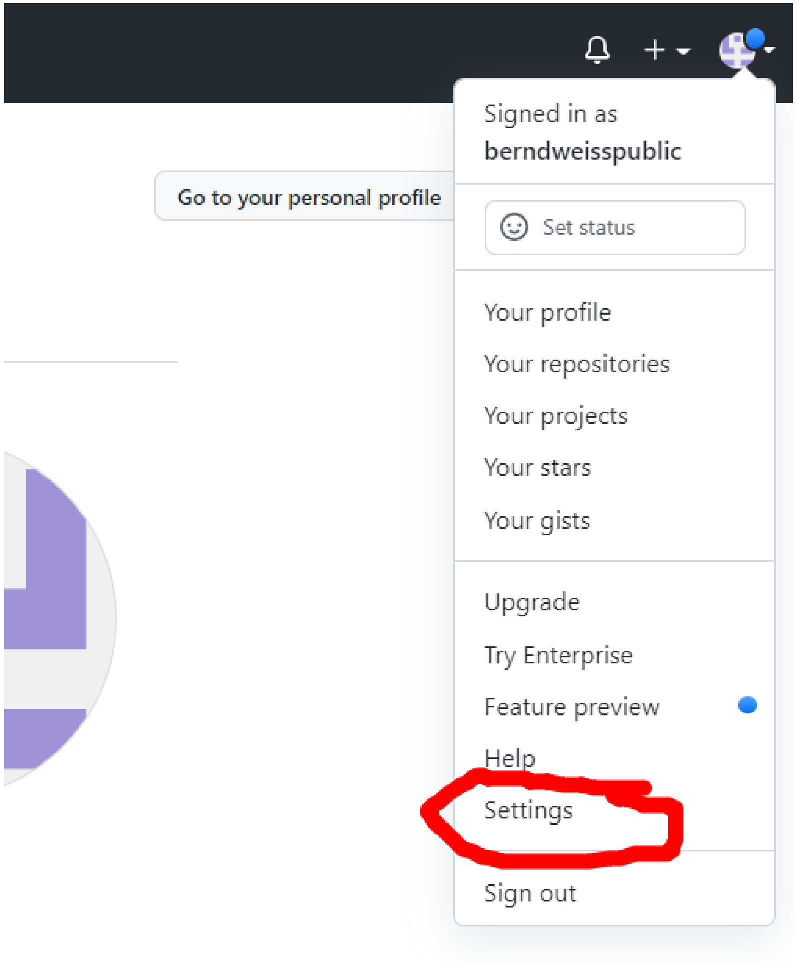
- GitHub (which will be using)
- GitLab
- ...

GitHub: Using personal access tokens

- These days, authentication via personal access tokens (PAT) (and https) seems the way to go when using GitHub
- In the following, I will illustrate the process using multiple screenshots
- Note that my explanation does not include any R/RStudio-related processes. Johannes will talk about these things in more detail

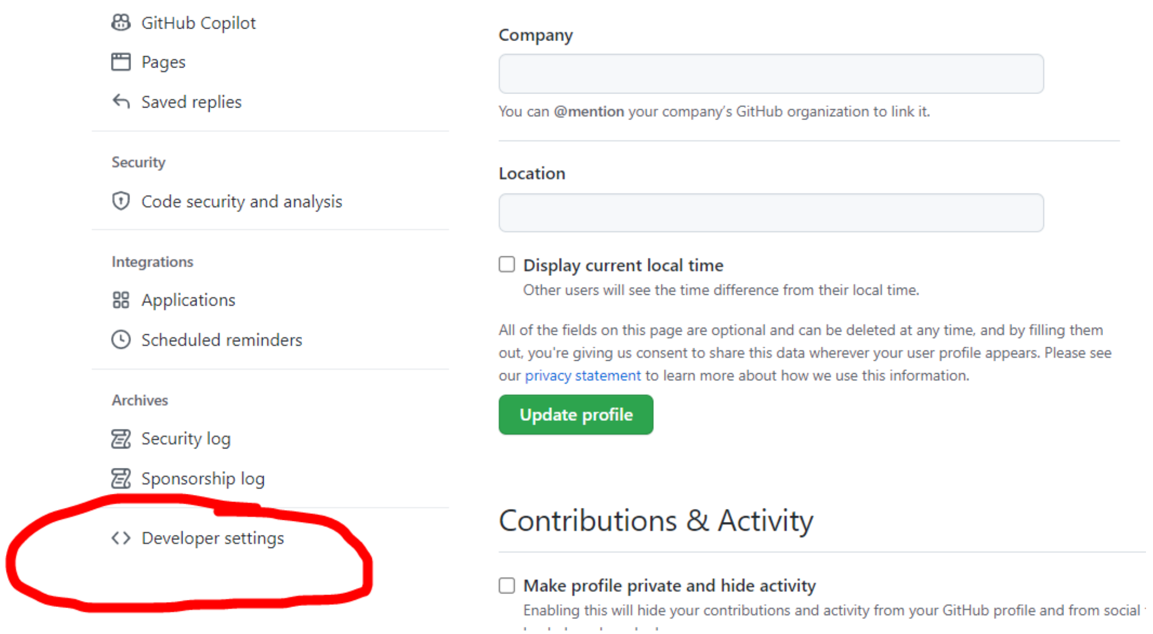
gesis Leibniz Institute for the Social Sciences

In GitHub, go to the `Settings` website:



gesis Leibniz Institute for the Social Sciences

Next, go to the `Developer Settings` entry:


GitHub Copilot
Pages
Saved replies

Security
Code security and analysis

Integrations
Applications
Scheduled reminders

Archives
Security log
Sponsorship log

Developer settings

Company

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

Location

☐ Display current local time
Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our privacy statement to learn more about how we use this information.

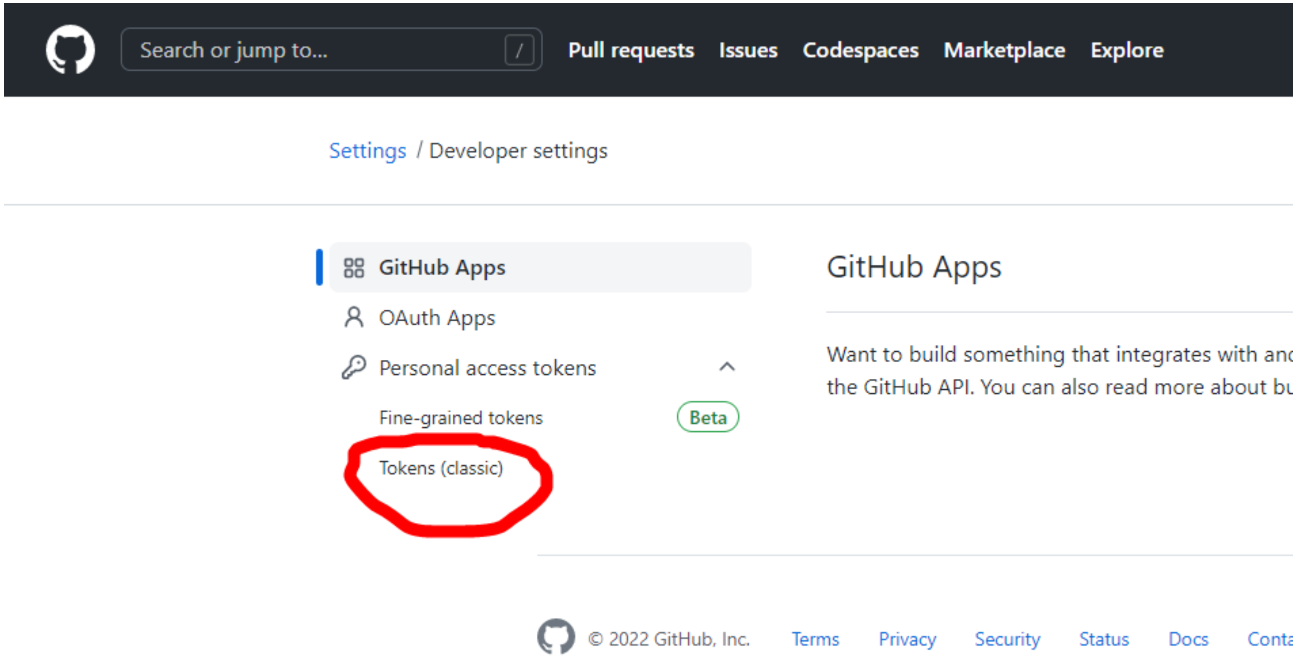
Update profile

Contributions & Activity

☐ Make profile private and hide activity
Enabling this will hide your contributions and activity from your GitHub profile and from social

Leibniz Institute for the Social Sciences

Then, choose `Tokens (classic)`:

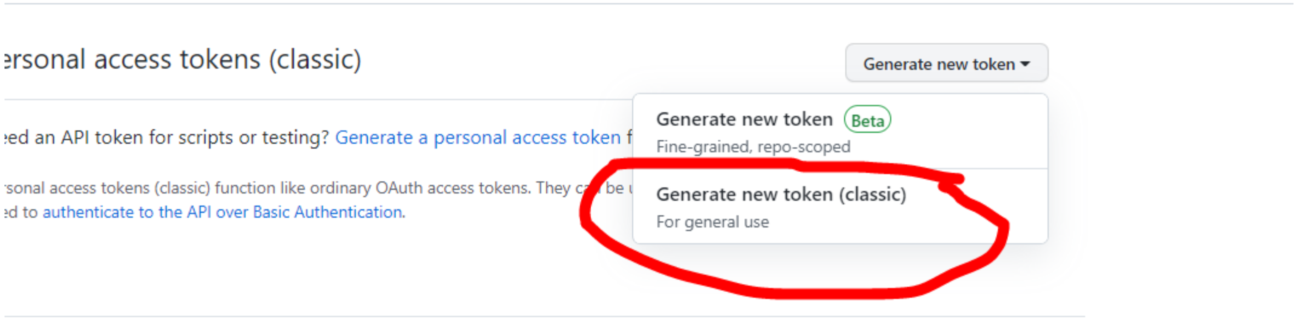


GitHub Apps

Want to build something that integrates with anc
the GitHub API. You can also read more about bu

© 2022 GitHub, Inc.

And, generate a new token; important, save this Token (e.g., `ghp_FqamoGsx6PcEnpwnXyh...`):



ersonal access tokens (classic)

Generate new token

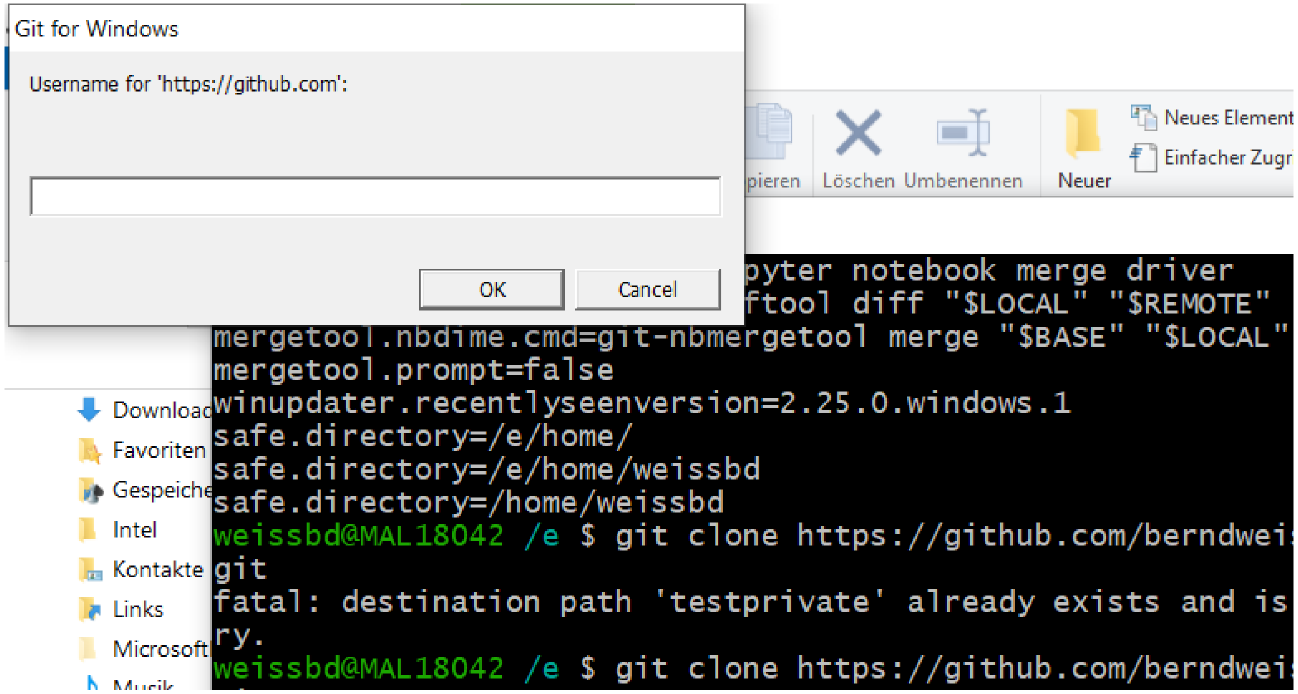
ed an API token for scripts or testing? Generate a personal access token f

sonal access tokens (classic) function like ordinary OAuth access tokens. They ca be u
ed to authenticate to the API over Basic Authentication.

Generate new token Beta
Fine-grained, repo-scoped

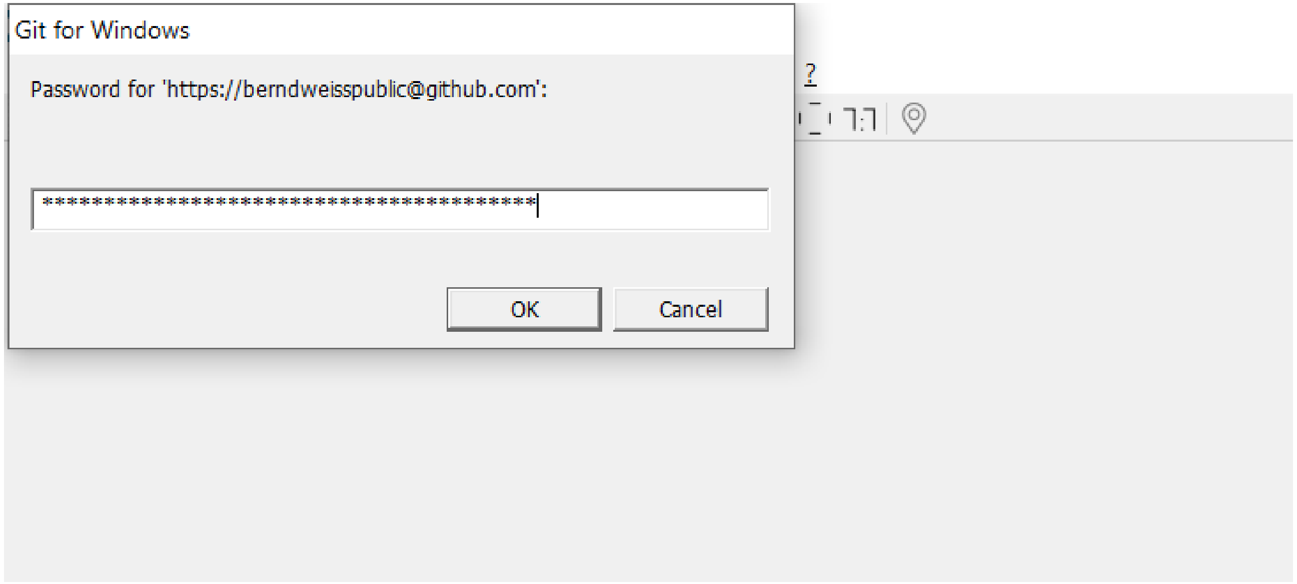
Generate new token (classic)
For general use

When you are now cloning a new repository (or pushing/pulling for the first time), you will be asked once to enter your username...



```
pyter notebook merge driver
ftool diff "$LOCAL" "$REMOTE"
mergetool.nbdime.cmd=git-nbmergetool merge "$BASE" "$LOCAL"
mergetool.prompt=false
winupdater.recentlyseenversion=2.25.0.windows.1
safe.directory=/e/home/
safe.directory=/e/home/weissbd
safe.directory=/home/weissbd
weissbd@MAL18042 /e $ git clone https://github.com/berndwei
git
fatal: destination path 'testprivate' already exists and is
ry.
weissbd@MAL18042 /e $ git clone https://github.com/berndwei
```

... and your Token (even though it says "Password"):


Git for Windows

Password for 'https://berndweisspublic@github.com':

****

If, for whatever reason, you decide to reset/remove your credentials, you can do so using the Windows Credentials Manager (in German: "Anmeldeinformationsverwaltung")

Setting up SSH

- SSH is a network protocol that comes in handy, when you work with remote repositories and when you do not want to type-in your passwort every time you pull (fetch) or push (send) from a remote repository. You still need to authenticate yourself, though.

- To work with Git on you local computer, you do not need SSH (= Secure Shell).

- Authentication in SSH (which is also the name of the program) works by using a private and a public key (usually the public key has the file extension `.pub`, e.g., my public key is `id_rsa.pub`). When you start working with SSH for the very first time, you have to create both keys.

- The private key remains on your local computer and you have to make sure that it is safe -- it is a simple text file and it is your password now, and everyone who has your private key can access your files. Again, everyone who has your private key has your password!

This is what my *public key* looks like:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAyOQ9RT6TkfgkdO2NspzdVJE5CZ03yYAhVwLGo
CrI3E9/Ix0MAySunXExjhsQi2XkhPBjLOEahYuuLaAWHuBc7apUPRNSBy+mdUHnH3
0BdTQijQ6vj3RL99HO4yrZnipIlkS5ufw/+hpbXXOzSOqTvyGtL9ygm3eA2HDSQtz
2ptFq8anODJDKrgTbNLb/YZ9KDIcpdO/Sfk4LtvaGF3tIFlyE+pogNmN4eWiYg9Xv
25BhVVxWMHadRFLeDastWO4SedriEHzQYaNgxVNTufqolJ0nbg4R//fVDxjR2SbzV
AHLZ+eVPUx+vzcPVMP9wYPcnii9YLiSRy+hlUAOR/kXeQ== berndweiss
```

The *public key* (not the private key!) has to be stored at the GitHub/GitLab/... website. Now, everyone who has your public key can encrypt files (that are sent to you via the internet) but only you (or anyone else who has your private key) can decrypt the files. And, for that reasons you do not have to login everytime you push/pull files from the remote repository.

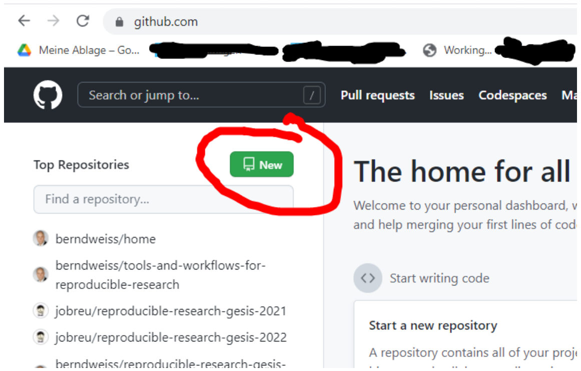
- How to setup SSH on you computer is explained on this website: https://docs.gitlab.com/ce/ssh/README.html ("Generate an SSH key pair")
- The most important point is that `ssh` is able to find your key pair, i.e., it needs to be located in your HOME folder

If everything works well, you should receive the following friendly welcome message after typing in the command `ssh -T git@github.com`:

> Hi berndweiss! You've successfully authenticated, but GitHub does not provide shell access.

Exercise

- In a previous exercise, you have created your own repository (let's call it `your-new-repo`)
- Now, go to your GitHub account and create a new repository on GitHub
- Startpage -> tab "Repositories" -> green button "New"


github.com
Meine Ablage – Go...
[redacted]
[redacted]
Working...
[redacted]
Search or jump to...
Pull requests
Issues
Codespaces
Top Repositories
New
Find a repository...
berndweiss/home
berndweiss/tools-and-workflows-for-reproducible-research
jobreu/reproducible-research-gesis-2021
jobreu/reproducible-research-gesis-2022
berndweiss/reproducible-research-gesis-
The home for all
Welcome to your personal dashboard,
and help merging your first lines of cod
Start writing code
Start a new repository
A repository contains all of your proj

- Enter a new "Repository name"
- Make it "Private" (unless you have something important to share)
- Do **not** check any of the "Initialize this repository with" boxes
- Hit the green "Create repository" button

- Choose SSH or HTTPS as protocol ("Quick setup — if you’ve done this kind of thing before")
- Look for "...or push an existing repository from the command line"
- SSH:
    - Copy the line `git remote add origin git@github.com:berndweiss/your-repo-name.git`
    - Execute the command `git remote add origin git@github.com:berndweiss/your-repo-name.git` in the Git Bash in your local repository (`your-new-repo`)

- HTTPS:
    - Copy the line `git remote add origin https://github.com/berndweiss/your-repo-name.git`
    - Execute the command `git remote add origin https://github.com/berndweiss/your-repo-name.git` in the Git Bash in your local repository (`your-new-repo`)
- Make sure that `git status` shows a clean repository
- Now you can run your first `git push origin main`
- Reload the GitHub page via F5; you now should see the content of your local repo `your-new-repo`
- You can delete a GitHub repository via the tab "Settings" -> "Options", then scroll down -> "Danger Zone" -> "Delete this repository"