

# How do I back up all my Github repositories?

---

## Goal

The goal of this article is to explain how to back up all your Github repositories with a utility script.

I wrote this script in order to be able to:

- the personal user repositories private and public,
- the user organization repositories,
- the ancestor repositories,
- the fork repositories.

When the script is run for the first time, it will retrieve all the repositories if the repositories have been previously retrieved it will do an update.

**Important:** The repository name with space will be replace in clone folder by -. Example the repository name like **repo 1** will be clone with name **repo-1**.

---

## Installation

Before getting started, you'll need to install **Python** and **Git** on your machine:

[Python >=3.9](#)

[Git](#)

After installation, you can clone the utility script at the following address: [Github Clone Repo](#)

---

## Preparation

### Token creation

Go to your **Github** account to generate a token with read-only authorization as below::

- If you use the [Fine-grained personal access tokens](#) you must provide the following authorizations:

Repository permissions 2 Selected

Repository permissions permit access to repositories and related resources.

Account permissions

User permissions permit access to resources under your personal GitHub account.

Overview

2 permissions for all of your repositories

Contents

Access: Read-only

Metadata

Access: Read-only

0 Account permissions

Update

Cancel

- If you use the [Personal access token \(classic\)](#) you must provide the following authorizations:

Expiration \*

30 days

The token will expire on Thu, Aug 17 2023

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

☐ delete:packages

Delete packages from GitHub Package Registry

☐ admin:org

Full control of orgs and teams, read and write org projects

☐ write:org

Read and write org and team membership, read and write org projects

☐ read:org

Read org and team membership, read org projects

☐ manage\_runners:org

Manage org runners and runner groups

**Important:** If you want to update fork repositories, you need to give workflow authorization to allow repositories to be updated. Activate the option as shown below:

☒ 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

☐ delete:packages

Delete packages from GitHub Package Registry

☐ admin:org

Full control of orgs and teams, read and write org projects

☐ write:org

Read and write org and team membership, read and write org projects

☐ read:org

Read org and team membership, read org projects

☐ manage\_runners:org

Manage org runners and runner groups

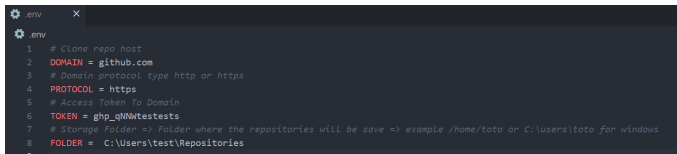
Initializing environment variables

In the clone utility project repository folder, create the file `.env` with the following contents:

```
# Clone repo host
DOMAIN = github.com
# Domain protocol type http or https
PROTOCOL = https
# Access Token To Domain
TOKEN = your_generate_token_here
# Storage Folder => Folder where the repositories will be save => example
```

```
/home/toto or C:\users\toto for windows  
FOLDER = folder_path_here
```

You must enter specific values as shown in the example below:



```
1 # Clone repo host  
2 DOMAIN = github.com  
3 # Domain protocol type http or https  
4 PROTOCOL = https  
5 # Access Token To Domain  
6 TOKEN = ghp_gMteStests  
7 # Storage Folder -> Folder where the repositories will be save -> example /home/toto or C:\users\toto for windows  
8 FOLDER = C:\Users\toto\Repos
```

**Information:** During execution, destination folders will be created if they do not exist.

## Execution

1. Run the following command in the project folder (must be run once):

```
python -m venv env
```

or

```
python3 -m venv env
```

or for ubuntu

```
virtualenv env
```

**Information:** If you encounter an error, please check the following link:  
<https://gist.github.com/frfahim/73c0fad6350332cef7a653bcd762f08d>

2. Run the following command to activate the environment: Lancez la commande suivante pour activer l'environnement:

```
source env/bin/activate
```

or on windows

```
env\Scripts\activate.bat
```

3. Install the modules from the command:

```
pip install -r requirements.txt
```

4. Run the backup script with the command: `python main.py` or `python3 main.py`.

At the end you'll have all the repositories in the destination folder.

```
The summary of actions are:  
Number of new repository clones: 3  
Number of repository updates: 0  
Number of failures: 0  
Number of successes: 3  
  
This programme takes: 0 year(s) and 0 month(s) and 0 day(s) and 0 hour(s) and 0 minute(s) and 14 second(s)
```

**Information:** Note that for each execution you have the log generated inside utility script folder inside subfolder **logs**.