

Comment sauvegarder tous ses dépôts Github ?

Objectif

L'objectif de cet article est d'expliquer comment sauvegarder l'ensemble de ses dépôts Github avec un script utilitaire.

J'ai écrit cet script dans le but de pouvoir :

- récupérer tous mes dépôts Github ainsi que toutes les branches associées,
- récupérer tous les dépôts dont j'ai fait un fork ainsi qu'une mise à jour du fork avant récupération,
- récupérer tous les dépôts de mes organisations,
- récupérer tous les dépôts de mes ascendants.

Quand le script est exécuté pour la première fois, il va récupérer tous les dépôts si les dépôts ont été précédemment récupérer il fera une mise à jour.

Important: Les dépôts avec des espaces dans le nom seront récupérés avec remplacement de l'espace par -. Par exemple le dépôt avec nom **repo 1** sera récupérer avec le nom **repo-1**.

Installation

Avant de commencer, vous devez installer **Python** et **Git** sur votre machine :

[Python >=3.9](#)

[Git](#)

Après installation, vous pouvez cloner le script utilitaire à l'adresse suivante: [Github Clone Repo](#)

Préparation

Création des tokens

Allez sur votre compte **Github** pour générer un token avec l'autorisation de la lecture uniquement comme ci-dessous :

- Si vous utilisez l'option [Fine-grained personal access tokens](#) vous devez fournir les autorisations suivantes :

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

- Si vous utilisez l'option [Personal access token \(classic\)](#) vous devez fournir les autorisations suivantes :

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: Si vous voulez mettre à jour les dépôts fork, vous devez donner l'autorisation workflow pour permettre la mise à jour des dépôts. Activez l'option comme ci-dessous :

☒ 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

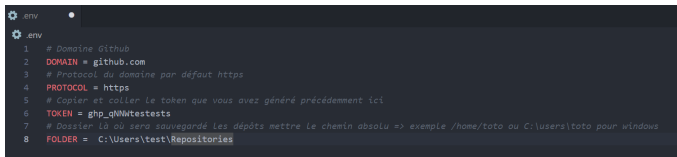
Initialisation des variables d'environnement

Dans le dossier du dépôt récupéré, créer le fichier **.env** avec le contenu suivant :

```
# Domaine Github
DOMAIN = github.com
# Protocol du domaine par défaut https
PROTOCOL = https
# Copier et coller le token que vous avez généré précédemment ici
TOKEN = le_token_genere_ici
# Dossier là où sera sauvegardé les dépôts mettre le chemin absolu => exemple
```

```
/home/toto ou C:\users\toto pour windows  
FOLDER = chemin_absolu_du_dossier
```

Vous devez mettre les valeurs spécifiques comme l'exemple ci-dessous :



```
.env  
1 # Domaine Github  
2 DOMAIN = github.com  
3 # Protocol du domaine par défaut https  
4 PROTOCOL = https  
5 # Copier et coller le token que vous avez généré précédemment ici  
6 TOKEN = ghp_ghMtestests  
7 # Dossier là où sera sauvegardé les dépôts mettre le chemin absolu => exemple /home/toto ou C:\users\toto pour windows  
8 FOLDER = C:\Users\toto\Repositories
```

Information: Lors de l'exécution, les dossiers de destination seront créés s'ils n'existent pas.

Exécution

1. Lancez la commande suivante dans le dossier du projet (doit être lancé une fois) :

```
python -m venv env
```

ou

```
python3 -m venv env
```

ou pour ubuntu

```
virtualenv env
```

Information: Si vous rencontrer une erreur, veuillez vérifier sur le lien suivant:
<https://gist.github.com/frfahim/73c0fad6350332cef7a653bcd762f08d>

2. Lancez la commande suivante pour activer l'environnement :

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

ou sur windows

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

3. Installez les modules à partir de la commande :

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

4. Lancez le script de sauvegarde avec la commande : `python main.py` or `python3 main.py`

Et voilà à la fin vous aurez tous les dépôts dans le dossier de destination.

```
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: Notez que pour chaque exécution, vous avez le journal généré dans le dossier du script utilitaire dans le sous-dossier **logs**.