

How do I back up all my Gitlab repositories?

Goal

The goal of this article is to explain how to back up all your Gitlab 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 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: [Gitlab Clone Repo](#)

Preparation

Token creation

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

Token name

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

Expiration date

YYYY-MM-DD

Select scopes

Scopes set the permission levels granted to the token. [Learn more.](#)

☐ api
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.

☒ read_api
Grants read access to the API, including all groups and projects, the container registry, and the package registry.

☐ read_user

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:

Token name

For example, the application using the token or the purpose of the token. Do not give sensitive information for the name of the token, as it will be visible to all project members.

Expiration date

2023-10-23

Select scopes

Scopes set the permission levels granted to the token. [Learn more.](#)

- ☐ api
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ read_api
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ☒ read_user
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- ☐ create_runner
Grants create access to the runners.
- ☐ read_repository
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ☒ write_repository
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Initializing environment variables

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

```
# Clone repo host
DOMAIN = gitlab.com
# Domain protocol type http or https
PROTOCOL = https
# Access Token To Domain
TOKEN = your_generate_token_here
# Username gitlab
USERNAME = your_gitlab_username
# 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:

```
.env
1 # Clone repo host
2 DOMAIN = gitlab.com
3 # Domain protocol type http or https
4 PROTOCOL = https
5 # Access Token To Domain
6 TOKEN = ay5tpondzvz7BvnduK
7 # User name
8 USERNAME = john
9 # Storage Folder -> Folder where the repositories will be save -> example /home/toto or C:\users\toto for windows
10 FOLDER = C:\Users\john\repositories
```

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 venv
```

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**.