

# github

Repository from command line:

<https://gist.github.com/alexpchin/dc91e723d4db5018fef8>

<https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line>

## Create repo on github

```
curl -H 'Authorization: token my_access_token'
https://api.github.com/user/repos
curl -u 'AnonTester' https://api.github.com/user/repos -d
'{"name":"context.youtube.dl.menu"}'

curl -u USERNAME:PERSONAL_ACCESS_TOKEN https://api.github.com/user/repos -d
'{"name":"myDirName"}' #will create the repo in github.
```

## Create repository

```
cd context.youtube.dl.menu/
git config --global init.defaultBranch main
git init
Initialised empty Git repository in
/home/kodi/wuffrepo/context.youtube.dl.menu/.git/
git add .
git commit -m "initial commit v1.2.0"
git remote add origin git@github.com:AnonTester/context.youtube.dl.menu.git
git push -u origin master
```

## Generate SSH key for github access

```
eval $(ssh-agent -s)
ssh-keygen -t rsa -b 4096 -C "key comment"
## you'll be prompted to a couple of times. Press enter for the first
prompt. choose a passphrase for the second prompt, or press enter twice for
no passphrase
ssh-add ~/.ssh/id_rsa #this is your private key
cat ~/.ssh/id_rsa.pub # copy the output of this command. this is your SSH
public key
curl -u USERNAME:PASSWORD https://api.github.com/user/keys -d
'{"title":"KEY_NAME", "key":"YOUR_RSA_PUBLIC_KEY_HERE"}' #the value you
copied earlier and your keyname. I recommend using a combination of machine
```

name and app (My-laptop (Git CLI))

## Creating a personal access token

NOTE: As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year.

<https://docs.github.com/en/github/authenticating-to-github/creating-a-personal-access-token>

1. In the upper-right corner of any page, click your profile photo, then click Settings.
2. In the left sidebar, click Developer settings.
3. In the left sidebar, click Personal access tokens.
4. Click Generate new token.
5. Give your token a descriptive name.
6. Select the scopes, or permissions, you'd like to grant this token. To use your token to access repositories from the command line, select repo.
7. Click Generate token.
8. Click to copy the token to your clipboard. For security reasons, after you navigate off the page, you will not be able to see the token again.

Warning: Treat your tokens like passwords and keep them secret. When working with the API, use tokens as environment variables instead of hardcoding them into your programs.

## Using a token on the command line

Once you have a token, you can enter it instead of your password when performing Git operations over HTTPS.

For example, on the command line you would enter the following:

```
$ git clone https://github.com/username/repo.git
Username: your_username
Password: your_token
```

## Removing git history

```
git checkout --orphan newBranch
git add -A # Add all files and commit them
git commit
git branch -D master # Deletes the master branch
git branch -m master # Rename the current branch to master
git push -f origin master # Force push master branch to github
git gc --aggressive --prune=all # remove the old files

#rename url on github.com if desired, then update the url:
```

```
git remote set-url origin newURL
```

## Rebase with fork master

Example fork of repo mainuser/interestingrepo to myuser/interestingrepo

```
git clone git@github.com:myuser/interestingrepo.git
cd interestingrepo
git remote add upstream git@github.com:mainuser/interestingrepo.git
```

To update:

```
git fetch upstream
git rebase upstream/master
```

## Change last commit message

```
git commit --amend -m "New message"
git push --force repository-name branch-name
```

## Git Credentials

```
git config credential.helper store
```

then:

```
git push http://example.com/repo.git
Username: <type your username>
Password: <type your password>

[several days later]
git push http://example.com/repo.git
[your credentials are used automatically]
```

Passwords stored in plain text, each on their own line in URL format:

```
https://user:pass@example.com
```

## Forceful undo last commit

```
git reset --hard HEAD^
git push origin -f
```

From:  
<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:  
<http://wuff.dyndns.org/doku.php?id=howto:github>

Last update: **2025/10/21 20:53**

