

We changed our privacy policy. [Read more.](#)

Support for password authentication was removed. Please use a personal access token instead

Asked 1 month ago Active yesterday Viewed 290k times



I got this error on my console when I tried to use `git pull` :

523



remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see <https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/> for more information.
fatal: unable to access "...": The requested URL returned error: 403



177



It's very weird, because I just follow the documentation and created a token two weeks ago on GitHub that expires on Tue, Oct 26, 2021. Why has this been removed today?

[git](#) [github](#) [token](#) [github-api](#) [git-pull](#)

Share Improve this question Follow

edited 2 days ago



[user202729](#)

2,482 3 15 28

asked Aug 13 at 16:49



[Daemes](#)

4,091 3 3 8

5 I don't understand the question. Literally from the link in the error message "Beginning August 13, 2021, we will no longer accept account passwords when authenticating Git operations on GitHub.com." – [njzk2](#) Aug 14 at 21:42

5 I get this stupid error even though I already have an ssh key – [munchschair](#) Aug 18 at 16:45

3 @JuanC here is quick tutorial for mac youtu.be/iKf8-hhdWjs – [VeKe](#) Aug 20 at 2:18

1 Thank you for the tutorial @VeKe! I was able to fix it but it's a bit misleading when I am asked for the "password" but then it works with the token. – [Juan C](#) Aug 20 at 11:46

1 @JuanC glad it helped, are you working with windows or mac? – [VeKe](#) Aug 20 at 12:13

28 Answers

Active

Oldest

Votes



659



From August 13, 2021, GitHub is no longer accepting account passwords when authenticating Git operations. You need to add a **PAT (Personal Access Token)** instead, and you can follow the below method to add a PAT on your system.

Create Personal Access Token on GitHub

From your GitHub account, go to **Settings => Developer Settings => Personal Access Token => Generate New Token** (Give your password) => **Fillup the form** => click **Generate token** => **Copy the generated Token**, it will be something like

```
ghp_sFhFsSHhTzMDreGRLjmks4Tzuzgthdvfsrta
```

Now follow below method based on your machine:

For Windows OS

Go to **Credential Manager** from **Control Panel** => **Windows Credentials** => find `git:https://github.com` => **Edit** => On Password replace with with your **GitHub Personal Access Token** => You are Done

If you don't find `git:https://github.com` => Click on **Add a generic credential** => Internet address will be `git:https://github.com` and you need to type in your username and password will be your **GitHub Personal Access Token** => Click Ok and you are done

For macOS

Click on the Spotlight icon (magnifying glass) on the right side of the menu bar. Type **Keychain access** then press the Enter key to launch the app => In Keychain Access, search for `github.com` => Find the **internet password** entry for `github.com` => Edit or delete the entry accordingly => You are done

For a Linux-based OS

For Linux, you need to configure the local GIT client with a username and email address,

```
$ git config --global user.name "your_github_username"
$ git config --global user.email "your_github_email"
$ git config -l
```

Once GIT is configured, we can begin using it to access GitHub. Example:

```
$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY
> Cloning into `Spoon-Knife`...
$ Username for 'https://github.com' : username
$ Password for 'https://github.com' : give your personal access token here
```

Now cache the given record in your computer to remembers the token:

```
$ git config --global credential.helper cache
```

If needed, anytime you can delete the cache record by:

```
$ git config --global --unset credential.helper
$ git config --system --unset credential.helper
```

Now try to pull with `-v` to verify

```
$ git pull -v
```

Linux/Debian (Clone as follows):

```
git clone https://<tokenhere>@github.com/<user>/<repo>.git
```

For PhpStorm


If you are using PhpStorm, go to menu **Git** => **pull** and select authentication via **Personal Access Token**. Enter your PAT it will allow to pull/push the changes.

Share Improve this answer Follow

edited Sep 27 at 15:33

answered Aug 14 at 6:58

STA sta
15.9k 8 31 49

-
- 2 Be sure to save the token on some local file or in some cloud.Upvoted. – [Goran_Ilic_Ilke](#) Aug 14 at 9:37
-
- 2 Given that the OP states they already created a token two weeks ago, this doesn't address the actual problem. – [9769953](#) Aug 14 at 11:55
-
- 3 Ummmm, it isn't asking for a token when I clone my repo, how should I proceed then? – [Prhyme](#) Aug 14 at 15:16
-
- 3 Same experience as @Prhyme. I am not able to get git to prompt me for username and password, even trying to clone private repos, push to repos, etc. When I go to push to a repo, it continues to give the "Support for password authentication was removed" message, without prompting for anything. – [9gt53wS](#) Aug 14 at 19:08 
-
- 2 For mac first time, when you are prompted for a password when you pushing the code to your repo, enter the access token, It will add the token in the keychain and it won't prompt again. – [Dhiraj Himani](#) Aug 23 at 5:38
-

▲ If you're using macOS, just simply follow these steps:

255



1. Go to this link: <https://github.com/settings/tokens> (Profile -> settings -> developers setting -> personal access tokens). (don't go to repository setting; it's your profile setting)
2. Generate a new token and copy-paste it somewhere safely.
3. Now search for an app in your Mac, named `Keychain Access`.

4. Search for `github.com` (if there are multiple GitHub logins then choose `Kind: Internet password`), double-click it.
5. Click on show password, then enter your Mac's password and hit .
6. Password should be visible by now. Now, just paste the token you generated in step 2 and click .

And that's it. Enjoy!

If you're using Windows:

1. Follow steps 1 and 2 as above.
2. Search for an application in your Windows OS, named *Credential Manager* → then *Windows Credentials*.
3. Search for `github.com` and *edit* the password with the `token` you have generated on GitHub. Now enjoy!

Developer's hack (shortcode):

```
git remote set-url origin https://<githubtoken>@github.com/<username>/<repositoryname>.git
```

While cloning:

```
git clone https://<username>:<githubtoken>@github.com/<username>/<repositoryname>.git
```

It will work on every OS (Mac, Windows, or Linux).

Cons: You have to remember or should need to do to each repository in your local. So I'll prefer everyone to use above mentioned steps.

NOTE:

For those who don't have this entry: it could be made. one way to do it is- to clone a project. then it will ask for your username and password. instead of password give it the token and then the entry would be made.

Share Improve this answer Follow

edited Oct 16 at 5:06

answered Oct 14 at 14:16



17 There's no such entry for me (Mac) – [DarkNeuron](#) Aug 14 at 16:01

1 thanks, this works for Linux in my case – [Arun Maurya](#) Aug 15 at 6:11

1 definitely! thanks @ClaudH . I updated the answer. – [kartik tyagi](#) Aug 15 at 14:04 ✎

2 I found that on Windows, it was necessary to *delete* the existing credential in the credential manager. Then, the next time I tried to do a `git push`, it prompted for the Personal Access Token. Simply resetting the existing credential did not work. – [J. Peterson](#) Aug 16 at 7:48

1 saved my day 🙏 developer heck shortcut was great for me, just removed old remote url and added new one (mac, but i believe this is the case for all OS) – [Inzamam Malik](#) Aug 16 at 8:52 ✎



Use *My Account* → *Settings* → *Developer settings* → *Personal access tokens* → *Generate new token*.

170

```
git remote set-url origin https://<token>@github.com/<username>/<repo>
```



Share Improve this answer Follow



7 Note that this seems to require running this command in each repo for that repo to be reset with the new credentials. Hopefully there is a global solution. – [9gt53wS](#) Aug 14 at 19:08

8 This one perfectly worked for me. The above one with maximum vote didn't work. – [Kapilfreeman](#) Aug 14 at 20:55

3 Thanks! Using token as password not work, but adding this work awesome – [Syorito Hatsuki](#) Aug 16 at 4:28

5 This is very insecure, not only does it allow anyone on the local system with read access to the local repo control over that repo, `git remote -v` simply discloses the key giving the attacker account wide access to your GitHub account. – [Joshua Kolden](#) Aug 17 at 20:48

4 @Bălăji OS key managers and browsers take grate pains to secure credentials. Normal git workflow do not expose passwords or ssh private keys to other users with read access to a repo. The above advice does. It allows anyone with read access to a copy of a local repo, including for example a shared work or school filesystem, to see the user's clear text Personal Access Token. This extends to archived copies as well, such as source tarballs that retain the `.git` directory.



For Linux these simple steps can solve your problem

66

1. If your Git password is cached in *credential.helper*, then unset it:



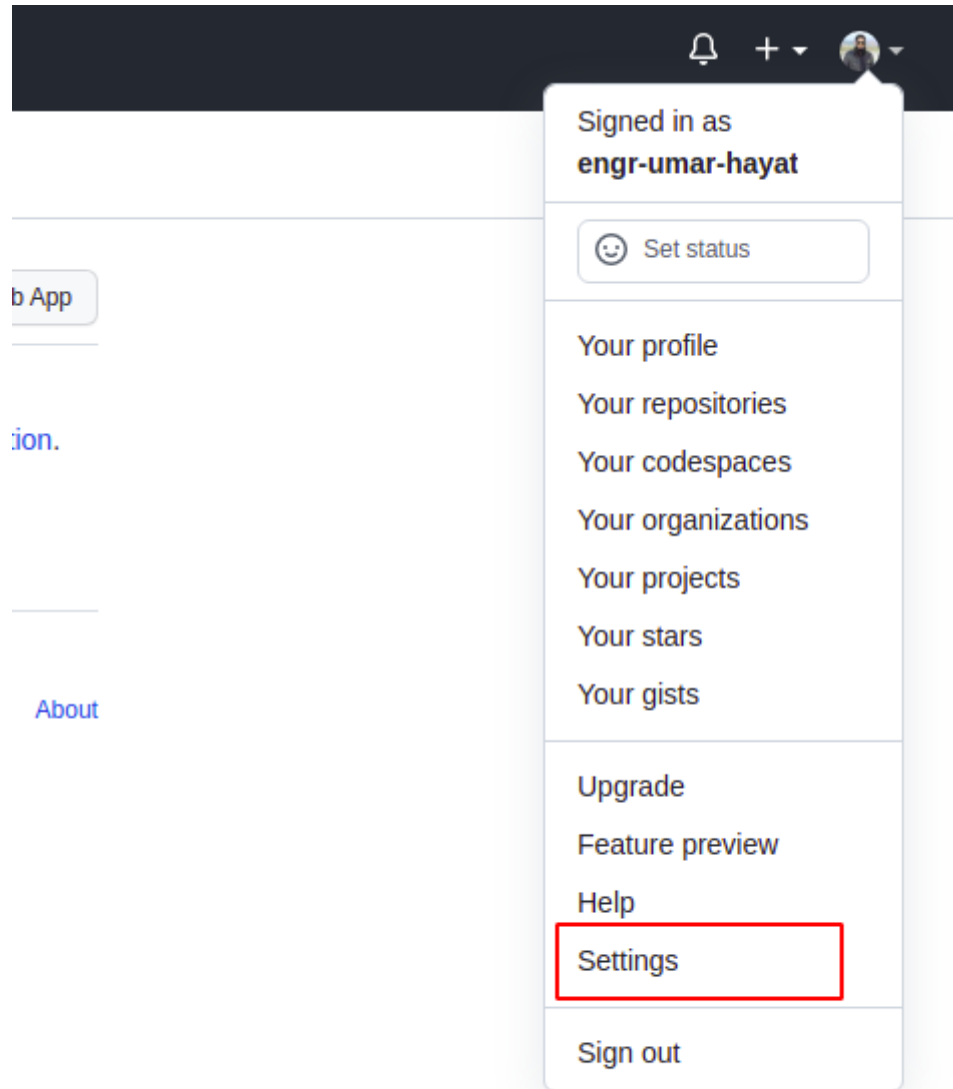
```
git config --local --unset credential.helper
```



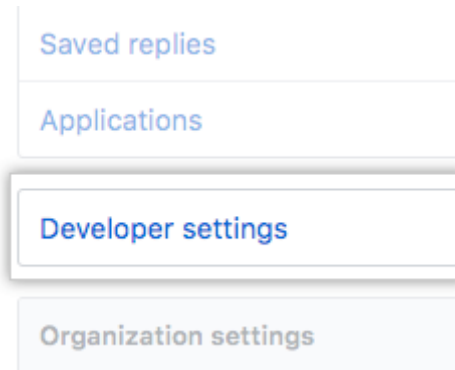
Or, If you have set your credentials globally, then:

```
git config --global --unset credential.helper
```

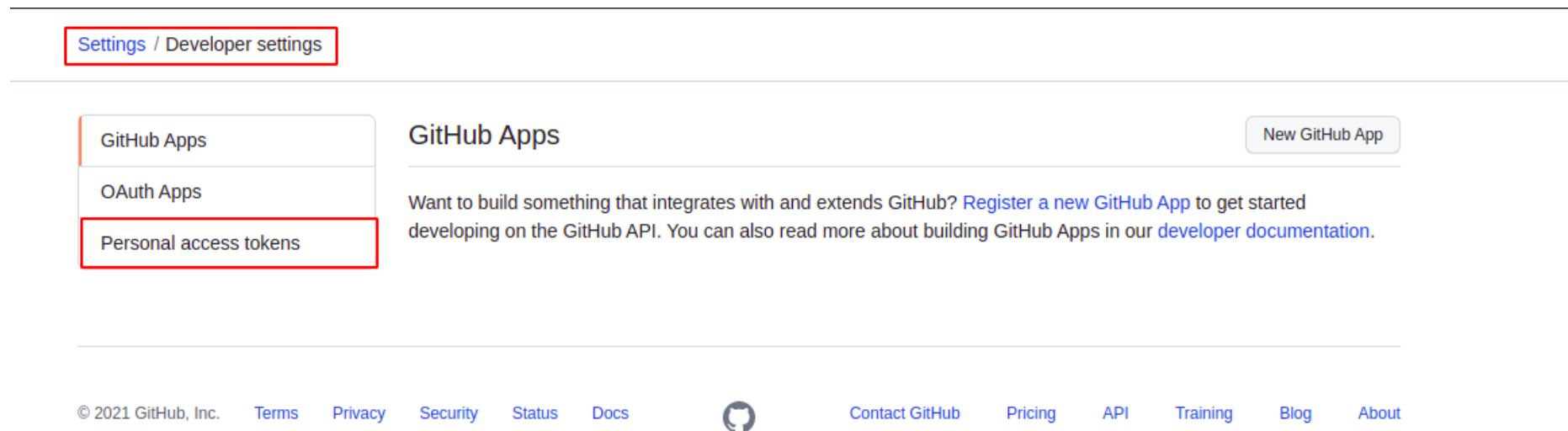
2. Now go to your GitHub *Account settings*



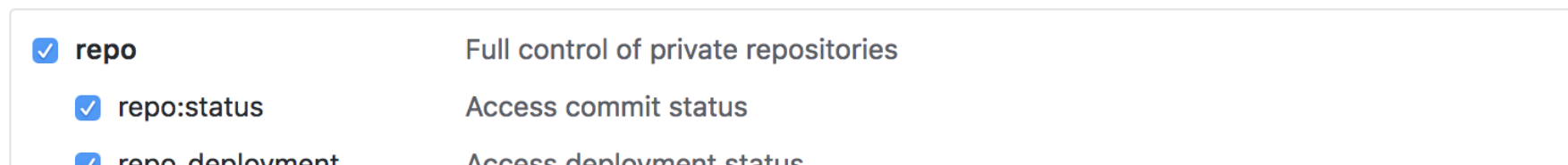
3. Click *Developer Settings*



4. Select *Personal Access*



5. Generate a token with the given permissions, e.g.,



<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> admin:org	Full control of orgs and teams
<input checked="" type="checkbox"/> write:org	Read and write org and team membership
<input checked="" type="checkbox"/> read:org	Read org and team membership
<input checked="" type="checkbox"/> admin:public_key	Full control of user public keys
<input checked="" type="checkbox"/> write:public_key	Write user public keys
<input checked="" type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input checked="" type="checkbox"/> admin:org_hook	Full control of organization hooks
<input type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input checked="" type="checkbox"/> user	Update all user data
<input checked="" type="checkbox"/> read:user	Read all user profile data
<input checked="" type="checkbox"/> user:email	Access user email addresses (read-only)
<input checked="" type="checkbox"/> user:follow	Follow and unfollow users
<input type="checkbox"/> delete_repo	Delete repositories

<input type="checkbox"/>	write:discussion	Read and write team discussions
<input type="checkbox"/>	read:discussion	Read team discussions
<input checked="" type="checkbox"/>	admin:gpg_key	Full control of user gpg keys (Developer Preview)
<input checked="" type="checkbox"/>	write:gpg_key	Write user gpg keys
<input checked="" type="checkbox"/>	read:gpg_key	Read user gpg keys

6. Now `git pull` inside your Git repository

7. Provide a username and the generated token as a password

```
Username for 'https://github.com': engr-umar-hayat
Password for 'https://engr-umar-hayat@github.com': GENERATED TOKEN HERE
remote: Enumerating objects: 1814, done.
remote: Counting objects: 100% (949/949), done.
remote: Compressing objects: 100% (200/200), done.
```

That is a straightforward solution step by step.

PS: If you are annoyed by Git asking for username/token again and again, follow these three simple steps

- Run `nano ~/.git-credentials`. Remove the GitHub line and save it.
- `git config --global credential.helper store`
Risky as physically the token is saved in file `~/.git-credentials`
- Run `git pull` and provide the username and password only once

It will not ask for the username and access token again and again now!

Share Improve this answer Follow

edited Sep 5 at 9:14



Peter Mortensen

28.9k 21 96 123

answered Aug 14 at 8:51



Umar Hayat

1,809 6 21

1 Those are a *lot* of permissions. What are you using that token for? – [Bergi](#) Aug 14 at 13:15

It all depends on the use case you want to use that token for. So select those permissions according to your use case. – [Umar Hayat](#) Aug 14 at 15:37



62



This message means that you're using a password instead of a personal access token to access GitHub over HTTPS, and that's no longer allowed. GitHub has disabled password authentication because it's common for people to accidentally leak their passwords, and while a personal access token can be restricted to limit the damage, a password cannot.

If you haven't explicitly entered your password at a prompt, then it's likely you have a credential manager which is saving your password and sending it without prompting you.

You can follow the directions for clearing your credential manager [listed in the Git FAQ](#):

```
$ echo url=https://account@github.com | git credential reject
```

You should use this same URL, but replace `account` with your own username (e.g., in my case, it would look like `echo url=https://bk2204@github.com`).

Then, the next time you try to push or pull, Git will prompt you for a username and password. For the username, enter your GitHub username, and for the password, generate a new personal access token [on the appropriate settings page](#) and paste it into the password field. If you're working from the command line, you may not see any indication that the password was successfully pasted; this is normal, so just hit afterwards.

That will save the personal access token in your credential manager for the next time, assuming you have one set up. If you're not sure if you have one set up, run `git config credential.helper` and see if it outputs anything.

If you don't have one set up, you should add one so that you don't have to memorize your token. Run one of the following commands,

depending on operating system:

- `git config --global credential.helper manager` on Windows;
- `git config --global credential.helper osxkeychain` on macOS;
- `git config --global credential.helper libsecret` on Linux (if available); or
- `git config --global credential.helper store` on Linux if `libsecret` isn't available.

Then you can try the push or pull again until you're no longer prompted.

Share Improve this answer Follow

edited Sep 4 at 22:28



Peter Mortensen

28.9k 21 96 123

answered Aug 13 at 17:01



bk2204

37.8k 3 30 50

- 1 Thanks for the info, I just tested `git config --global credential.helper osxkeychain` and `git config --global credential.helper` but nothing is happening in my console, so I will try the first solution. Just one question, what is the `url=https://account@github.com` ? The link of my repot or my github?
– Daemes Aug 13 at 17:22
 - 3 check out stackoverflow.com/questions/68776300/... to change the password on a mac – Evan Aug 13 at 19:22
 - 2 The technique I've mentioned above will clear the password on any system so Git will prompt for it again. That's why we documented it in the FAQ, because the directions are different for each individual OS and credential helper and it's too hard to write a single answer that applies to all systems when it involves the GUI.
– bk2204 Aug 13 at 19:56
 - 2 `echo url=https://account@github.com | git credential reject` was just what I needed. Thank you! – Nolan Strait Aug 17 at 18:39
- this was extremely clear and helpful, thanks a lot! – baskcat Aug 19 at 13:18



42



GitHub has made changes in password authentication. If you are trying to access Git by username and password then it does not allow you. So use a **personal access token** instead of a password to access Git everywhere.

Here are the steps to generate **Personal access tokens**.

Click here - <https://github.com/settings/tokens>



Step 1 - Open GitHub and log in with your credentials

Step 1 - Open GitHub and log in with your credentials.

Step 2 - Click on the **Setting** menu.

Step 3 - From the **Setting** menu click on **Developer Settings**

Step 4 - From the **Developer Settings**, menu click on **Personal access token**

Step 5 - From the **Personal access token** click on the **Generate new Token** button.

Step 6 - Now fill up required details like **Note, Expiration, Select scopes**. And then click on **Generate Token** button.

Step 7 - After that new token has been generated. Copy that generated token and use this token to access Git with username and token.

If you are using Windows then please follow the below step.

Open **Control Panel** => **User Accounts** => **Manage your credentials** => **Windows Credentials**.

It will show all generic credentials. Find your GitHub URL and click on that. Now click on the **edit** button. And then add the **personal access token** generated from **GitHub** into the password field. And click on the button.

Now you can access Git.

If you are accessing Git in Android Studio, if asked for a password then add the GitHub **personal access token** instead of your password everywhere.

Share Improve this answer Follow

edited yesterday

answered Aug 14 at 11:03




Jaydip Meghapara

1,019 12 13

1 You are my hero. At first it did not work because for me the correct GitHub URL in the credential manager was git:github.com and not github.com/username. Maybe it helps someone. Also I am of the opinion the hatefulness of GitHub knows no bounds. – Lars Wissler Aug 14 at 11:31

There's a alternative way as well to do that: you can use direct ssh key as well: youtu.be/egdJy_FWUh4 – Code Cooker Aug 21 at 16:38

1 So basically password should be replaced by git token everywhere. – Aditya Kapre Shrewsbury Aug 21 at 20:45

And, if I am using Linux command line ? Edit: answer: just use the token as password... – 3xCh1_23 Aug 24 at 18:45 

1 @Lars Wissler Thank you so much, you saved me! git:github.com is the correct address indeed. – Blue Phoenix Sep 2 at 1:39



Generate an access token in GitHub from *Settings* → *Developer settings*.

30

If you have cloned your repository in the past and made it as *origin*, then you can change the authentication so,



```
git remote set-url origin https://<token>@github.com/<username>/<repo>.git
```



If you are going to clone repository after 13 August 2021, then you can use the normal clone command and set the generated access token in the password field.

Share Improve this answer Follow

edited Sep 5 at 9:16



Peter Mortensen

28.9k 21 96 123


answered Aug 14 at 14:41



Abdelrahman Elayashy

345 3 10

And it should be "<repo>.git". – G.Vanem Aug 15 at 12:04

If you're like me and have years worth of git repos that you used password based auth on. I have made this: `find . -type f -path '*/.git/*' -name 'config' -exec sed -i 's%https://github.com/%git@github.com:%g' {} \;` It will recursively search in the `.git` folder for any file called config and find and replace the https URL with the git@github one. I'd recommend only doing this on simple repos where you know you have plain vanilla remotes. I haven't tested how it will work on different setups. YMMV! – robmsmt Aug 17 at 2:05 

I tried removing the username and it worked: `git remote set-url origin https://<token>@github.com/<repo>.git` – doubleo46 Aug 20 at 7:07



For **Ubuntu**, use the following steps

16



- At <https://github.com/settings/tokens>, go and generate a token.
- `git push`
- `username: user_github_username`



- password: add_generated_token instead of the password.

Share Improve this answer Follow

edited Sep 5 at 9:09



Peter Mortensen

28.9k 21 96 123

answered Aug 14 at 17:29



Muhammad Javed

Baloch

430 5 17

worked perfectly, simple and effective, thanks! – Mathiasfc Sep 3 at 18:15

▲ A one-command simple solution to solve it

12 *If your computer has no SSH key added to the GitHub account, I add information for you to do it at the end of the answer. You should do it first.*

▼ After push failed, then do this:



```
git remote set-url origin git@github.com:{user_id}/{project_name}.git
```

And push again. Then it works.

Let me show my case in the following.

(And I will guide you on how to do your case.)

At the first, when I add, commit, and push, then I meet this issue:

```
miloo@MacBook-Pro ~/git/open-idea-pool (master) $ touch test.md
miloo@MacBook-Pro ~/git/open-idea-pool (master) $ git add .
miloo@MacBook-Pro ~/git/open-idea-pool (master) $ git commit -m "Add test.md"
[master 55b6486] Add test.md
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.md
miloo@MacBook-Pro ~/git/open-idea-pool (master) $ git push -u origin master
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.
remote: Please see https://github.blog/2020-12-15-token-authentication-requirements-for-git-operations/ for more information.
fatal: unable to access 'https://github.com/milochen0418/open-idea-pool/': The requested URL returned error: 403
```

And then, my current Git log is the following.


```

milo@MacBook-Pro ~/git/open-idea-pool (master) $ git log
commit 55b6486471282a9840d00d1ad6a76a035a8ee3dc (HEAD -> master)
Author: milochen0418 <milo_chen@mail2000.com.tw>
Date: Sat Aug 14 21:35:57 2021 +0800

    Add test.md

commit 142716a28ce3b81066ad3e4a123a0a41e2949742 (origin/master, origin/HEAD)
Author: HackMD <no-reply@hackmd.io>
Date: Wed Aug 7 18:40:00 2019 +0000

```

In the final, this is my way to solve the issue.

```

milo@MacBook-Pro ~/git/open-idea-pool (master) $ git remote set-url origin git@github.com:milochen0418/open-idea-pool.git
milo@MacBook-Pro ~/git/open-idea-pool (master) $ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 272.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:milochen0418/open-idea-pool.git
    142716a..55b6486  master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
milo@MacBook-Pro ~/git/open-idea-pool (master) $ git log
commit 55b6486471282a9840d00d1ad6a76a035a8ee3dc (HEAD -> master, origin/master, origin/HEAD)
Author: milochen0418 <milo_chen@mail2000.com.tw>
Date: Sat Aug 14 21:35:57 2021 +0800

    Add test.md

```

In my case,

1. **{project_name}** <-> open-idea-pool
2. **{user_id}** <-> milochen0418 is the
3. **{branch_name}** <-> master

(your *branch_name* maybe is *main*, but not *master*)

When I push failed, the only thing I need is this one command:

```
git remote set-url origin git@github.com:{user_id}/{project_name}.git
```

Then I push it again by:

```
git push -u origin {branch_name}
```

For the example of my case,

```
git remote set-url origin git@github.com:miлоchen0418/open-idea-pool.git
git push -u origin master
```

It works.

-- By the way, this is the process of creating an SSH key to GitHub account.

You can refer to these two links to do it. The information here supports Mac, Windows, and Linux.

1. [Issue on adding SSH key to GitHub](#)
2. [Adding a new SSH key to your GitHub account](#)

Furthermore, if you want to clone a new project, you can do the following command:

```
git clone git@github.com:{user_id}/{project_name}.git
```

Share Improve this answer Follow

edited Sep 13 at 11:37

answered Aug 14 at 14:07



Milo Chen

2,222 2 12 25

2 I bet this works only because you have an SSH key added to the github account. – [HolyBlackCat](#) Aug 14 at 15:17

@HolyBlackCat Yes, you are right – [Milo Chen](#) Aug 14 at 15:29

HolyBlackCat Yes, you are right. – [Milo Chen](#) Aug 14 at 15:29

1 Maybe add the process of creating an SSH key to the answer? – [HolyBlackCat](#) Aug 14 at 15:32

Thanks a lot. I will add some simple processes of creating an SSH key tomorrow. GitHub official doc also introduces how to do it, but that's is too complicated for junior people. – [Milo Chen](#) Aug 14 at 15:35

I've done to add the process of creating an SSH key to github account by refer two excellent links. – [Milo Chen](#) Aug 15 at 4:57



Generating a GitHub personal access token (PAT)

1. Log in to your GitHub account and open *Settings* → *Developer Settings* Personal Access Tokens.
2. Click on *Generate New Token* .



Do not forget to copy and the token after generation. It is accessible only once.

Windows

1. Open *Control Panel* → *User Accounts* → *Credential Manager*.
2. Now, look for `git:https://github.com` .
3. Click on *Edit*.
4. Replace the password with the GitHub Personal Access Token.

MAC

1. Click on Spotlight Icon and search for `Keychain Access` .
2. Inside Keychain Access, search for `github.com`.
3. Now, look for the internet password entry for github.com.
4. Update the entry.

Share Improve this answer Follow

edited Sep 4 at 22:55



[Peter Mortensen](#)

answered Aug 14 at 11:40



[Anurag](#)



Peter Mortensen
28.9k 21 96 123



Ayush
111 5

I found that on Windows, it was necessary to *delete* the existing credential in the credential manager. Then, the next time I tried to do a `git push`, it prompted for the Personal Access Token. Simply resetting the existing credential did not work. – J. Peterson Aug 16 at 7:51

Your solution is the simplest and fastest. Thanks – ekkee Aug 16 at 15:04

▲ Mac users -- a simple solution.

7 Set up the personal access token in GitHub:

▼ *Settings → Developer Settings → Personal access token*

🕒 Open *Keychain* → type "GitHub"

Change the password to the personal access token.

Try commit/pushing.

Select "Always allow" and business is as usual.

Share Improve this answer Follow

edited Sep 5 at 8:56



Peter Mortensen
28.9k 21 96 123

answered Aug 14 at 17:14



Vasilios Kambouras
79 4

▲ I was using `git pull` on Linux with a password before.

7 If that is the case, the only thing you need to do is use **token** in place of **password**.

▼ Generate an access token by going to [Personal access tokens](#).

🕒 Here is example of `git pull`

```
git pull
```

```
Username for 'https://github.com': gitusername
Password for 'https://yourusername@github.com': //Then enter generated token
```

Share Improve this answer Follow

edited Sep 5 at 22:54

answered Sep 2 at 9:28



VK321

4,988

3

35

43

In my case after doing "git pull", just do "git pull" a second time and you should get prompted for the username and password automatically. It shouldn't require you go to delete your password in the OS settings. – [P.Brian.Mackey](#) Sep 9 at 15:27



In Linux, you can generate a new access token in GitHub and directly replace it in place of the password in the `~.git-credentials` file.

5

The password section starts with `:` and ends with `@`.



Share Improve this answer Follow

edited Sep 5 at 7:48

answered Aug 14 at 17:02



Peter Mortensen

28.9k

21

96

123



Shriraj Hegde

91

5



First create the token on GitHub:

5

1. Click on the profile picture and in the menu select **Settings**



2. Select **Developer Settings** in the left menu at the bottom



3. Select **Personal Access tokens** in the left menu (third option)

4. Select **Generate new token** and follow the next steps.

If you have not installed GitHub CLI, you would not find it in your keychain Access. Therefore you should install it first.

1. Install GitHub CLI for macOS

```
brew install gh
```

For Windows:

```
winget install gh  
scoop install gh  
choco install gh
```

2. On the command line, enter `gh auth login`, follow the prompts, and provide the following answers

? What account do you want to log into? **GitHub.com**

? What is your preferred protocol for Git operations? **HTTPS**

? Authenticate Git with your GitHub credentials? **Yes**

? How would you like to authenticate GitHub CLI? Paste an authentication token

3. Paste the token you created on GitHub and press .

Share Improve this answer Follow

edited Sep 5 at 10:08



Peter Mortensen

28.9k 21 96 123

answered Sep 4 at 10:01



Pascal Nitchou

355 3 4



For Mac users

4

Generate a token and set it in the keychain.



[This quick video explains it well.](#)



For Windows Users

Just instead of the keychain, you need to set up [Credential Manager](#) with the newly generated token.

[Here is a quick video for Windows](#)

Share Improve this answer Follow

edited Sep 5 at 9:21



Peter Mortensen

answered Aug 20 at 5:10



Pascal Nitchou



Peter Mortensen

28.9k 21 96 123



Snagayag

81 12



Go to *Settings* → *Developer settings* → generate a new token if it does not exist.

4

Or copy the token if it exists.



Settings / Developer settings



GitHub Apps

OAuth Apps

Personal access tokens

Edit personal access token

Make sure to copy your personal access token now. You won't be able to see it again!

[Redacted token]



Note

test

What's this token for?

Expiration

This token expires **on Tue, Sep 21 2021**. To set a new expiration date, you must [regenerate the token](#).

Then set the token:

```
git remote set-url origin https://<token>@github.com/<username>/<repo>.git/
```

Share Improve this answer Follow

edited Sep 5 at 9:30



Peter Mortensen

28.9k 21 96 123

answered Aug 22 at 13:53



FBA Gimhana

61 1



As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year. To provide additional security, we highly recommend adding expiration to your personal access tokens.

3

Just follow [Creating a token](#).



Share Improve this answer Follow

edited Sep 5 at 9:06



Peter Mortensen

28.9k 21 96 123

answered Aug 19 at 11:30



vaibhav

131 1 2



Run these commands inside your project directory if the project was cloned before 13 Aug 2020:

3

```
git remote set-url <remote_name> <remote_url>
git pull
```



Type your *username* and *personal access token* as the *password*.

Share Improve this answer Follow

edited Sep 5 at 9:17



Peter Mortensen

28.9k 21 96 123

answered Aug 14 at 18:17



Jirawat Yuktawathin

39 2

Example - git remote set-url origin github.com/myName/GitTest.git – Akitha_MJ Aug 17 at 17:05



A quick solution for Windows users

3

Open [Credential Manager](#) and just remove the already-saved credentials for Git: <https://github.com>.



Control Panel > User Accounts > Credential Manager

Control Panel Home

DESKTOP-JJQ0L68 Modified: 12/30/2020

Certificate-Based Credentials [Add a certificate-based credential](#)

No certificates.

Generic Credentials [Add a generic credential](#)

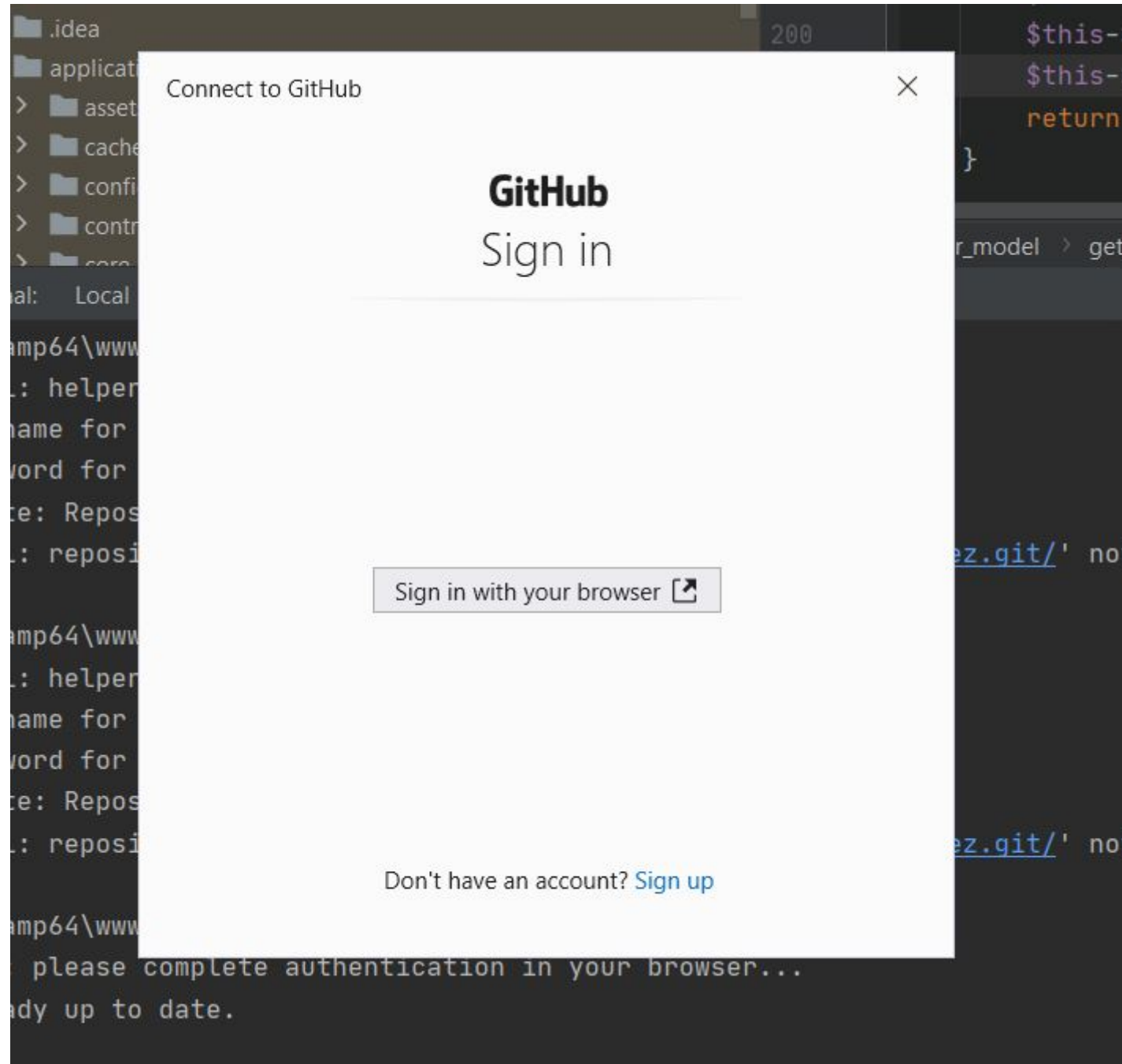
MSIX-Skype for Desktop MSAv2/live:ahmadkhan12345...	Modified: 8/19/2021
Termius/localKey	Modified: 4/23/2021
git:https://github.com	Modified: Today
Internet or network address: git:https://github.com User name: Ahmadkhan12345566 Password: Persistence: Local computer Edit Remove	
MicrosoftAccount:user=ahmadkhan12345566@outloo...	Modified: 8/16/2021
MSIX-Skype for Desktop MSA/live:ahmadkhan12345566	Modified: 6/15/2021
MSIX-Skype for Desktop/live:ahmadkhan12345566	Modified: 6/15/2021
virtualapp/didlogical	Modified: 8/5/2021
SSO_POP_Device	Modified: Today
XboxLive	Modified: Today

See also

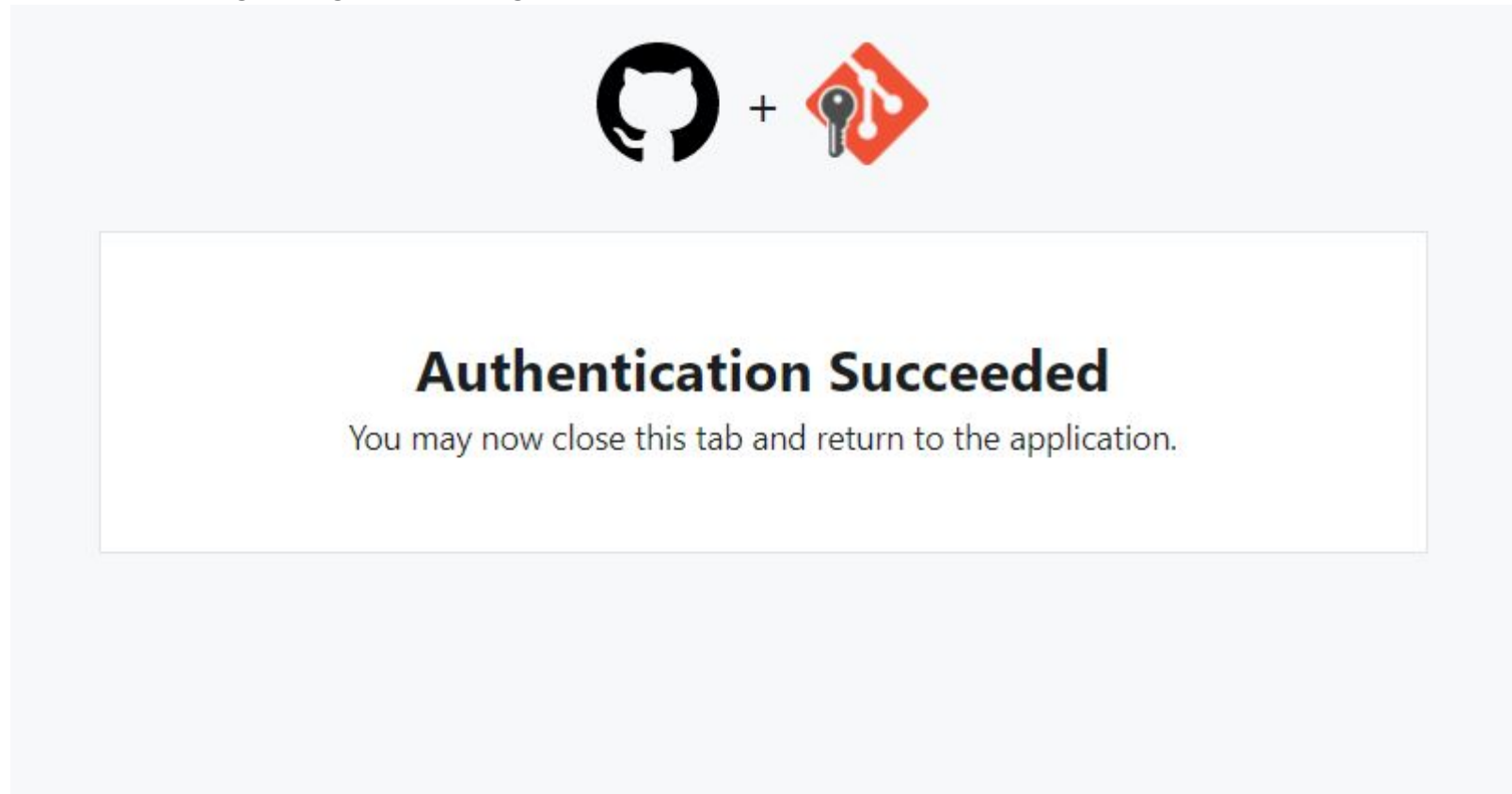
User Accounts

After this step, run the command again for pull/push in the terminal. GitHub will ask to log in with your default browser automatically (**make sure you are logged in with GitHub in the default browser**)

Make sure you are logged in with GitHub in the default browser.



After successful login, we got this message:



For more help related to this, you can also watch [this video](#).

Share Improve this answer Follow

edited Sep 5 at 9:28



Peter Mortensen

28.9k 21 96 123

answered Aug 20 at 14:56



Arslan Ahmad Khan

4,068 1 22 31

▲ Android Studio Arctic Fox Solution (Windows 10)

- 2 I was authenticated in Android Studio Arctic Fox (latest version) using a personal access token, but still, I was getting this error on August 14, 2021:

remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.



I generated a new access token on github.com, removed my GitHub account from Android Studio, and added the account again with the new access token. But still, it was showing the same error.

Reading other solutions on Stack Overflow, I downloaded the GitHub CLI, and added my GitHub account in the Windows command prompt with the access token successfully, and tried to push the repository from Android Studio again, which again failed.

Then following [this article](#), I did the following at the command line:

```
cd <project-directory>
git remote set-url origin https://<TOKEN>@github.com/<user_name>/<repo_name>.git
```

I pushed again from Android Studio, and it worked finally!

Share Improve this answer Follow

edited Sep 4 at 22:46



Peter Mortensen

28.9k 21 96 123

answered Aug 14 at 9:31



Raw Hasan

315 1 11



In three commands:

2

First create a token as indicated in the official documentation: [Creating a personal access token](#)



Then reset the Git credentials



```
git config --global --unset credential.helper
```

Clone or pull your repositories:

```
`git pull` or `git clone`
```

Then enter the token as your password.

If you want to save your token, you can use:

```
git config --global credential.helper cache
```

Share Improve this answer Follow

edited Sep 5 at 9:08



Peter Mortensen

28.9k 21 96 123

answered Aug 19 at 19:00



Badr Bellaj

7,588 1 33 31



2



Recently GitHub updated their *remote: Support for password authentication*. Which was removed on August 13, 2021 to use PAT (personal access token) instead for accessing the GitHub API.

You can read about in the blog post [here](#).



Now to enable PAT

- Log in to your GitHub account.
- Follow [this write-up](#) for the next actions.

Now to use the token on the command line, *do* this:

```
git clone https://github.com/username/repo.git
```

```
Username: your_username
```

```
Password: your_token
```

Note that your token has an expiry time and must be kept privately and used whenever you need to *authenticate* on the command line.

Heads up

If your are cloning into a repository that is owned by an:

Organisation

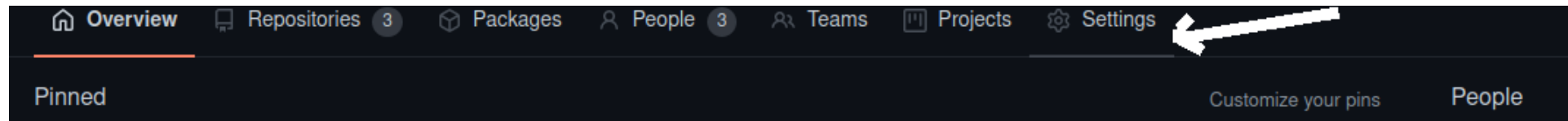
- Make sure you are **member** of that organisation

Or if some personal repository,

- Make sure you have **enabled** the following

The organisation/personal will need to ***enable remote cloning and forking*** like so:

- Go to organisation settings:



- Enable member privileges:

Account settings

- Profile
- Billing & plans
- Member privileges**
- Organization security
- Security & analysis
- Verified & approved domains
- Audit log
- Sponsorship log
- Webhooks
- Third-party access
- Installed GitHub Apps
- Scheduled reminders
- Repository topics

Member repository permissions

Base permissions

Base permissions to the organization's repositories apply to all members and excludes outside collaborators. Since organization members can have permissions from multiple sources, members and collaborators who have been granted a higher level of access than the base permissions will retain their higher permission privileges.

Write ▾

Repository creation

Members will be able to create only selected repository types. Outside collaborators can never create repositories.

- ☒ **Public**
Members will be able to create public repositories, visible to anyone. [Why is this option disabled?](#)
- ☒ **Private**
Members will be able to create private repositories, visible to organization members with permission.

Save

Repository forking

- ☒ **Allow forking of private repositories**
If enabled, forking is allowed on private and public repositories. If disabled, forking is only allowed on public repositories. This setting is also configurable per-repository.

Save

Share Improve this answer Follow

edited Sep 5 at 9:48



Peter Mortensen

28.9k 21 96 123

answered Sep 2 at 10:41



Leo Sammy

383 2 7



For people who want to use [Sourcetree](#):

2

Just copy like this at the *source* path:



`https://<token>@github.com/<username>/<repo>`



Clone

Cloning is even easier if you set up a [remote account](#)

[Browse](#)

Repository Type: [?](#) No path / URL supplied

[Browse](#)

Local Folder:

➤ Advanced Options

[Clone](#)

Share Improve this answer Follow

edited Sep 5 at 9:51

answered Sep 4 at 7:28



Peter Mortensen

28.9k 21 96 123



MHL007

31 2

▲ For those who followed the instruction to generate the **Personal access tokens**, but forgot to select suitable scopes:

- 1 Please do remember to choose suitable scopes like repositories. You won't get a different error prompt whether you use the previous password or a new token without a suitable scope.



Share Improve this answer Follow

edited Sep 5 at 9:23



Peter Mortensen

28.9k 21 96 123

answered Aug 20 at 9:51



Eugene

8,887 4 38 59

Re "choose suitable scopes like repositories" (seems incomprehensible): Do you mean "choose suitable scopes **for** repositories"? Or something else? Please respond by [editing \(changing\) your answer](#), not here in comments (**without** "Edit:", "Update:", or similar - the answer should appear as if it was written today).

– Peter Mortensen Sep 5 at 9:25

▲ For anyone who uses [Sourcetree](#) as the Git client, open *Preferences* → *go to the Accounts tab* → *Add GitHub account* → *Auth type = basic* → fill in your GitHub account and the access tokens from GitHub you created as a password → *Use HTTPS protocol* → done!

1



Share Improve this answer Follow



edited Sep 5 at 9:33



Peter Mortensen

28.9k 21 96 123

answered Aug 23 at 6:06



gogoqaz

172 2 10

▲ The best solution I have found so far:

- 1 1. Install GitHub CLI, `brew install gh` or [check how to install for other OSes](#)
2. Type `gh auth login` in your terminal
3. Follow through like this:



```
What account do you want to log into?
Choose GitHub.com
```

What is your preferred protocol for Git operations?

Choose HTTPS

Authenticate Git with your GitHub credentials?

Choose Yes

How would you like to authenticate GitHub CLI?

Choose Login with a web browser

(Copy the one-time code and paste in the browser)

Press 'Enter' to open github.com in your browser

Share Improve this answer Follow

edited Sep 5 at 10:10



Peter Mortensen

28.9k 21 96 123

answered Sep 4 at 8:48



ashdaily

1,259 15 24

In case someone is looking for way to install brew on Arch based distros: gist.github.com/fardjad/114ebf50a0dd031418bb63b3b134db51 – Harry Adel Sep 7 at 21:32 ✎

Thank you, Peter! Your solution has worked for me – Monero Jeanniton Sep 24 at 22:22

▲ Yesterday I had some issues trying to find GitHub on the Keychain in my Macbook. I've tried a lot of ways with no success.

0 From my perspective, the workaround I've used to be able to pull and push from my GitHub account fast, was doing it by SSH.

▼ This allowed me to do it with no further configuration.



1. You can do it going here: <https://github.com/settings/keys>
2. Then New SSH key
3. If you already created a ssh locally just run `pbcopy < ~/.ssh/id_rsa.pub` in your terminal
4. Then pull, push, clone or whatever you need from your GitHub repo and it's done

I know this is not the solution to the issue, but it worked for me since I needed to do it fast and to me honest there is no reason that I need to

use https instead of SSH

Hope this information is useful for someone!

Cheers!

Share Improve this answer Follow

answered Sep 19 at 20:09



That did not work for me. – Daniel Sep 21 at 20:47

Can you provide more details? Do you already have a ssh generated? Also, make sure when you are cloning the repo you change. HTTPS to SSH – undefinedsoul
Sep 22 at 18:46 ✎



0



I was not able to clone the project was always giving this error. The solution I encouraged was, after having generated the Token as the post was spoken in the post, I did as described below.

Using personal access token for the **cloning** new project

For cloning also you just need to modify the url as you have done instep 2

Older way to clone a repository

git clone repository_url folder_name

```
git clone https://github.com/<user>/<repository> my_project
```

New way of clone with personal access token

```
git clone https://<token>@github.com/<user>/<repository> my_project
```

How to solve, I want to share my solution.

Share Improve this answer Follow

answered Sep 22 at 15:46



This does not provide an answer to the question. Once you have sufficient [reputation](#) you will be able to [comment on any post](#); instead, [provide answers that don't require clarification from the asker](#). - [From Review](#) – [Daly](#) Sep 22 at 20:54



Highly active question. Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.