



Giorgos Myrianthous

Follow

Dec 20, 2021 · 4 min read

Listen



How to fix support for password authentication was removed on GitHub

Understanding how to configure and use Access Tokens in GitHub



Photo by [Twitter @ethmessages](#) on [Unsplash](#)

Introduction

In July 2020, [GitHub announced their intent](#) to require users to use **token-based authentication** in order to perform certain (authenticated) Git operations. Going forward and as of the 13th of August 2021, **account passwords are no longer accepted when authenticating with the REST API**.

For instance, if you attempt to `push` on the remote server using password authentication the operation will fail with the following message:

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

The recent changes affect the command line access to Git as well as any services accessing GitHub repositories directly with the use of password. On the other hand, if you have already enabled the two-factor authentication you are required to use a token-based authentication (or SSH-based authentication) and therefore you shouldn't be seeing the error mentioned above.

In today's article we will discuss we will go through a quick step by step guide that will help you configure Access Tokens on GitHub that we'll allow you to perform token-based authentication when executing Git operations that require you doing so.

Reproducing the error

Now let's assume that you've initialised a Git repository (`git init`), you've done some work and created a commit and finally you want to push the changes made to the remote host.

```
$ git push -U origin main  
Username for 'https://github.com': <username>  
Password for 'https://your-username@github.com':  
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-
```

Given that you are attempting to perform password-based authentication, the push command will fail with the authentication fatal error shown above.

Fixing the error: A step by step guide

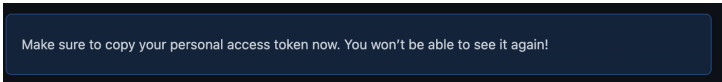
In the following sections we will go through a step by step guide that will help you configure Access Tokens on GitHub so that you can perform token-based authentication when performing Git operations requiring authentication.

Step 1: Create Access Token on GitHub

First of all, you must create a personal Access Token on GitHub. To do so follow the steps below:

- Click on your **GitHub profile icon** on the top right corner
- Click **Settings**
- From the menu shown on the left, click **Developer Settings**
- Click **Personal access tokens**
- Click **Generate new token**
- Add a note that will help you identify the scope of the access token to be generated
- **Choose the Expiration period** from the drop down menu (Ideally you should avoid choosing the No Expiration option)
- Finally, **select the scopes you want to grant the corresponding access to the generated access token**. Make sure to select the minimum required scopes otherwise you will still have troubles performing certain Git Operations.
- Finally click **Generate Token**

By now, you should have generated your personal access token successfully and the following message should be visible on your screen.



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

Source: [Author](#)

Underneath that message, you should also be able to see your personal access token. **Make sure to copy it** as we will need it in the following step(s).

Step 2: Change the remote URL

Now that we have created our personal access token, we need to run the following command:

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

In the above command make sure to replace:

- <githubtoken> with the personal access token you have copied in the previous step
- <username> with your GitHub username
- <repositoryname> with the name of your GitHub repository

Step 3: Test that it works

Now we have successfully configured token-based authentication for a specific repository on GitHub. To test that everything went to plan, simply try to push the changes you've made locally to the remote host. For example,

and the Git operation should be executed with no issues.

Final Thoughts

In today's short guide we discussed about the recent changes made on GitHub that now require users to perform token-based authentication as password-based authentication are no longer accepted.

Additionally, we went through a step-by-step guide that will help you configure your GitHub Access Tokens that will allow you to properly authenticate Git operations.

Become a member and read every story on Medium. Your membership fee directly supports me and other writers you read. You'll also get full access to every story on Medium.

Join Medium with my referral link — Giorgos Myrianthous

As a Medium member, a portion of your membership fee goes to writers you read, and you get full access to every story...

gmyrianthous.medium.com

You may also like

How To Undo The Last Local Commit in Git

Understanding how to undo the most recent local commit in Git

towardsdatascience.com

How To Merge Other Git Branches Into Your Own

Quickly merge branches using the command line or GitHub Desktop

betterprogramming.pub

Level Up Coding

Thanks for being a part of our community! Level Up is transforming tech recruiting. [Find your perfect job at the best companies.](#)

Level Up — Transforming the Hiring Process

- 🔥 Enabling software engineers to find the perfect role that they love
- 🔴 Finding talent is the most painful part of...

jobs.levelup.dev



67




2



Sign up for Top Stories





 [Get this newsletter](#)

Emails will be sent to bd97450@gmail.com.
[Not you?](#)

