git remote Support for password authentication was removed on August 13, 2021

子云之风

一 问题描述

git push 上报时提示
remote: Support for password authentication was removed on August 13, 2021. Please use a personal access token instead.

原因是：github 从 2021 年 8 月 13 日开始就不能用了，必须使用个人访问令牌（personal access token），需要把你的密码替换成 token！

二 解决办法

1.去 github 个人右侧界面，选择 Settings

ziyunzhifeng

☺ Set status

- Your profile
- Add account

- Your repositories
- Your projects
- Your Copilot
- Your organizations
- Your enterprises
- Your stars
- Your sponsors
- Your gists

- Upgrade
- Try Enterprise

🧪 Feature preview

⚙️ Settings

📖 GitHub Docs

👥 GitHub Support

Sign out

2.点击右侧的 Developer settings

🖥️ Codespaces

📦 Packages

🤖 Copilot

Pages

Saved replies

Security

Code security and ana

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

3.选择 Personal access tokens

Settings / **Developer Settings**

▢▢ **GitHub Apps**

人 OAuth Apps

🔑 Personal access tokens ⌃

Fine-grained tokens (Beta)

Tokens (classic)

4.Generate new token

Q Type [?] to search

Personal access tokens (classic)   Generate new token ▾   Revoke all

Tokens you have generated that can be used to access the

Generate new token (Beta)
Fine-grained, repo-scoped

Generate new token (classic)
For general use

▬▬▬ — repo   Delete
Expires **on Wed, Apr 17 2024**.

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over

5.配置 token 选项内容

设置一个名字，选择过期时间

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

What is this token for?

Expiration *

| 30 days ⇕ | The token will expire on Wed, Apr 17 2024

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

☐ **admin:public_key**　　　　Full control of user public keys
　☐ write:public_key　　　　　Write user public keys

下面选项内容建议全选，防止之后出现莫名其秒问题

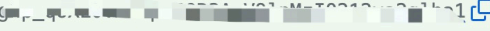| ☑ **delete_repo** | Delete repositories |
☑ **write:discussion**	Read and write team discussions
☑ read:discussion	Read team discussions
☑ **admin:enterprise**	Full control of enterprises
☑ manage_runners:enterprise	Manage enterprise runners and runner groups
☑ manage_billing:enterprise	Read and write enterprise billing data
☑ read:enterprise	Read enterprise profile data
☑ **audit_log**	Full control of audit log
☑ read:audit_log	Read access of audit log
☑ **codespace**	Full control of codespaces
☑ codespace:secrets	Ability to create, read, update, and delete codespace se
☑ **copilot**	Full control of GitHub Copilot settings and seat assignm
☑ manage_billing:copilot	View and edit Copilot Business seat assignments
☑ **project**	Full control of projects
☑ read:project	Read access of projects
☑ **admin:gpg_key**	Full control of public user GPG keys
☑ write:gpg_key	Write public user GPG keys
☑ read:gpg_key	Read public user GPG keys
☑ **admin:ssh_signing_key**	Full control of public user SSH signing keys
☑ write:ssh_signing_key	Write public user SSH signing keys
☑ read:ssh_signing_key	Read public user SSH signing keys

Generate token   Cancel

8 产生密钥
点击上面"Generate token"之后，得下面图片位置的 token
内容，保存起来避免遗忘

Tokens you have generated that can be used to access the GitHub API.

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

✓ g▓_▓_▓▓ ▓▓▓▓ ▓▓▓▓▓▓▓▓ ▓▓▓▓1 ⧉   Delete

▓▓▓ ▓ — *repo*   Last used within the last week   Delete

Expires *on Wed, Apr 17 2024.*

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over

三 执行push

在 git init，git commit， git add remote 执行之后
修改现有的url

```
git remote set-url origin  https://
<your_token>@github.com/<USERNAME>/<REPO>.git
```

将<your_token>换成你自己得到的令牌。是你自己github的
用户名，是你的项目名称

换成你自己得到的令牌。是你自己github的用户名，`是你
的项目名称

然后再次执行 pull push 操作，大功告成。

如果之后出现失败，大概率是你的 token 过期 了，需要重新
生成

今天的问题就记录到这里。欢迎关注我的公众号，获得更多

内容。