



2. Click on Developer Settings



Your personal account

[Go to your personal profile](#)

Account settings

Profile

Account

Appearance

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

SSH and GPG keys

Repositories

Packages

Organizations

Saved replies

Applications

Developer settings

Public profile

Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

You can @mention other users and organizations to link to them.

URL

Twitter username

Company

You can @mention your company's GitHub organization to link it.

Profile picture



3. Click on Personal Access Tokens

[Settings](#) / Developer settings

GitHub Apps

OAuth Apps

Personal access tokens

GitHub Apps

[New GitHub App](#)

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

4. Click on Generate new token

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Generate new token

Tokens you have generated that can be used to access the [GitHub API](#).

Personal access tokens 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](#).

5. Now type in the name of the token and select the scopes, or permissions, you'd like to grant this token. Make sure you select repo to use your token to access repositories from the command line. Click Generate token.

Note

What's this token for?

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<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"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input checked="" type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects

Note:

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

How to git push using GitHub token on the command line

Note:

Personal Access Tokens (PAT) can only be used for HTTPS git operations. Switch your repository to HTTPS if it is using SSH.

Once you have a token, you can use it instead of your password when performing git push operations over HTTPS as below:

```
git push https://<GITHUB_ACCESS_TOKEN>@github.com/<GITHUB_USERNAME>/<REPOSITORY_NAME>
```

Note: Remember to replace <GITHUB_ACCESS_TOKEN> , <GITHUB_USERNAME> , <REPOSITORY_NAME> with your token, username and repository.

Tagged: [git passwordless](#) [git push](#) [GitHub](#)

About The Author



Ramya Santhosh

is a Web Designer and content creator. A freelance writer on latest trends in technology, gadget reviews, How to's and many more.

Comments

Leave a Comment

Noname

3 weeks ago

Thanks a lot fot it