

Connecting to GitHub via HTTPS

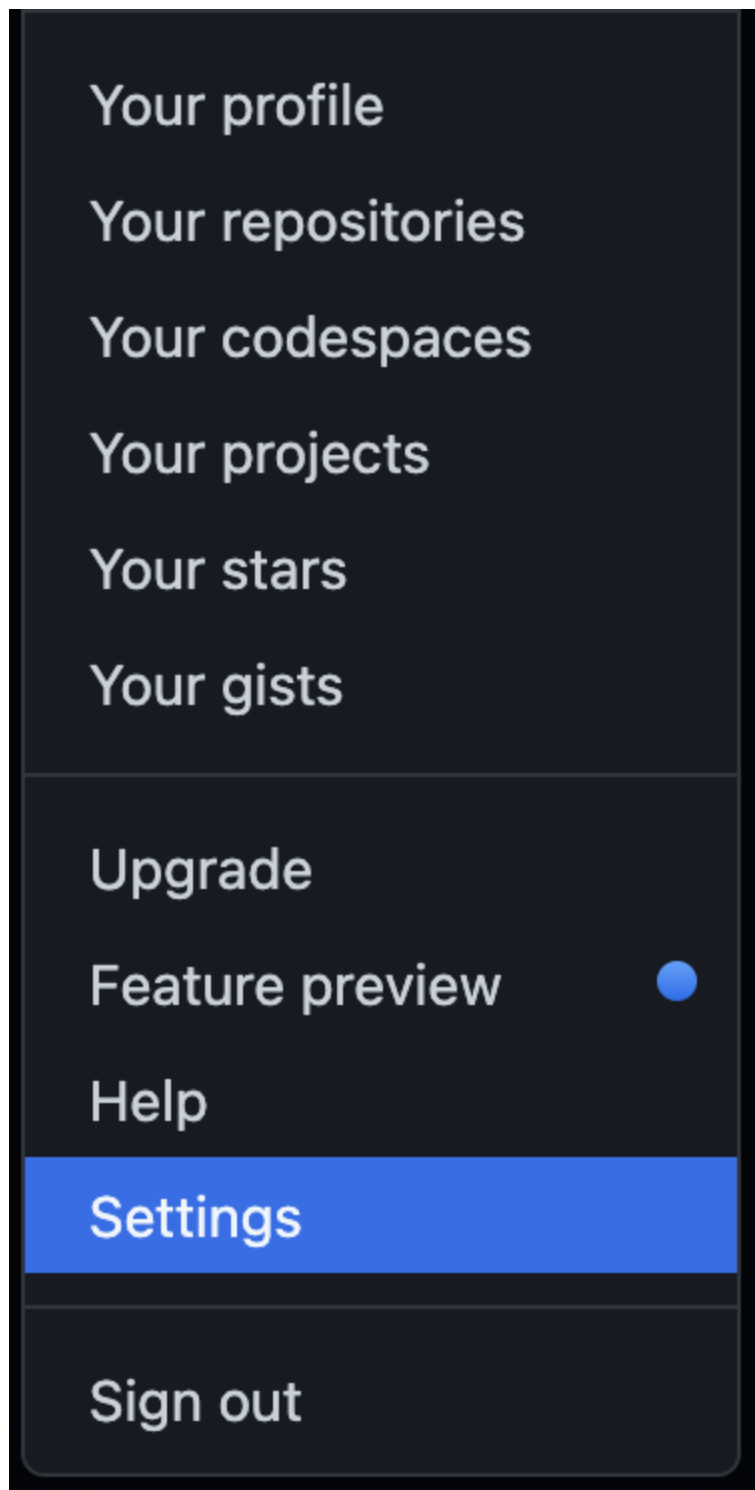
When using Github via the Coursera platform, it is required to authenticate using a Personal Access Token over HTTPS. A Personal Access Token is a special password that you use instead of your actual account password. When you're finished using the token, you can revoke it so that it can no longer be used. It is also possible to set an expiry time for the token. This helps to keep your account secure.

Generate a Personal Access Token


We now need to set up our Personal Access Token.


Step 1: Log in to Github


Step 2: Click on the profile icon in the top right of the screen and select Settings.



Step 3: On the Settings screen, on the left-hand side click Developer Settings.

 SSH and GPG keys


 Organizations


 Moderation




Code, planning, and automation


 Repositories

 Packages

 Pages


 Saved replies

Security

 Code security and analysis

Integrations


 Applications

 Scheduled reminders

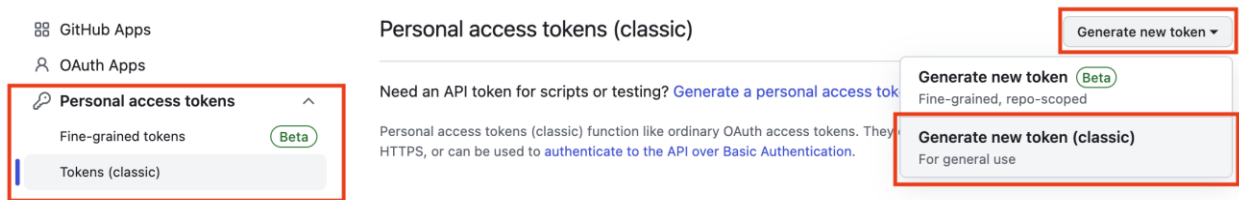
Archives

 Security log

 Sponsorship log

 Developer settings

Step 4: On the Developer Settings screen, click **Personal access tokens**. Under that, click on **Tokens (classic)**. Then, click the **Generate new token** button and select **Generate new token (classic)**.



Step 5: On the New Personal Access Token page, enter a token name and an expiry time. If you wish to manually revoke the token, set the expiry time to **No Expiration**.

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

courseira

What's this token for?

Expiration *

No expiration

The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

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

- | | |
|--|--------------------------------------|
| <input type="checkbox"/> repo | Full control of private repositories |
| <input type="checkbox"/> repo:status | Access commit status |
| <input type="checkbox"/> repo_deployment | Access deployment status |
| <input type="checkbox"/> public_repo | Access public repositories |
| <input type="checkbox"/> repo:invite | Access repository invitations |
| <input type="checkbox"/> security_events | Read and write security events |
| <hr/> | |
| <input type="checkbox"/> workflow | Update GitHub Action workflows |

Step 6: Under scopes, select **repo**.

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

coursera

What's this token for?

Expiration *

No expiration  The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure.

[Learn more](#)

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 type="checkbox"/> workflow | Update GitHub Action workflows |

Step 7: Scroll to the end of the page and click **Generate token**.

| | |
|---|--|
| <input type="checkbox"/> admin:ssh_signing_key | Full control of public user SSH signing keys |
| <input type="checkbox"/> write:ssh_signing_key | Write public user SSH signing keys |
| <input type="checkbox"/> read:ssh_signing_key | Read public user SSH signing keys |

Generate token

[Cancel](#)


Step 8: The token is now generated. Make sure to copy and keep note of the token as it will be hidden when you leave the page. This token can now be used when connecting to a repository over HTTPS.

Personal access tokens

[Generate new token](#)[Revoke all](#)

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!

✓ 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 

[Delete](#)

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](#).

Note: If you lose the token, you can delete the old token and create a new one.

Accessing Repositories

When accessing a repository and using HTTPS authentication, make sure you have access/permission to connect, and then just use the HTTPS address for the Git repository itself.

