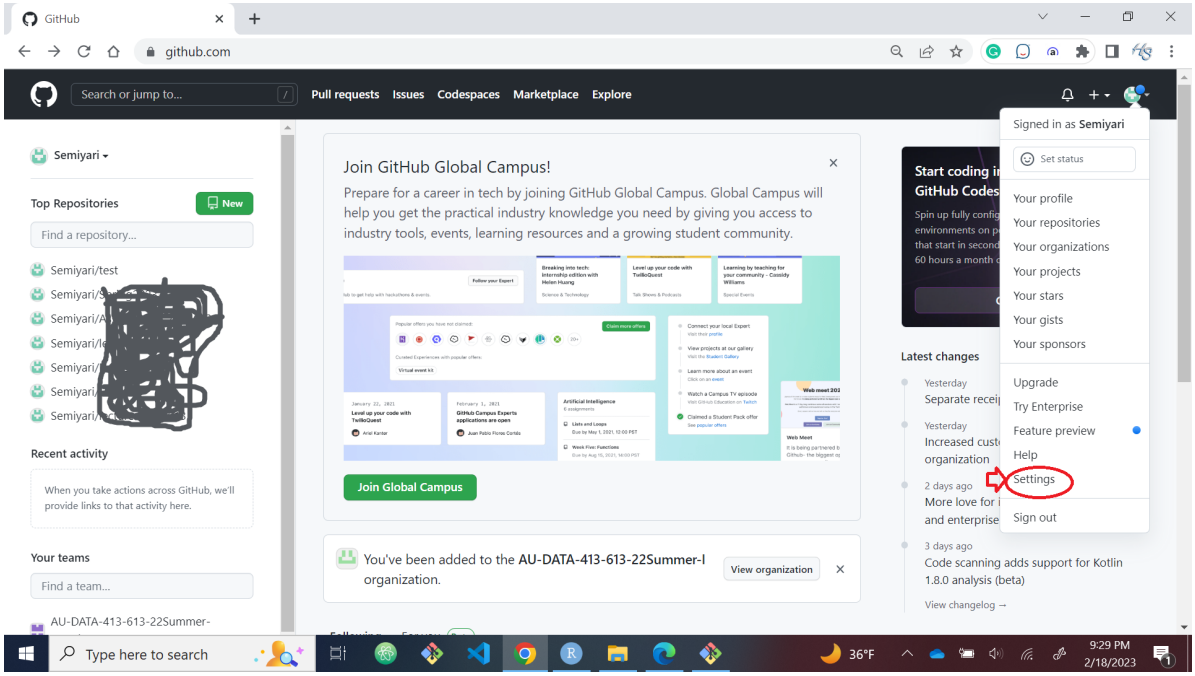
Personal Access Token, PAT

Hamid Semiyari

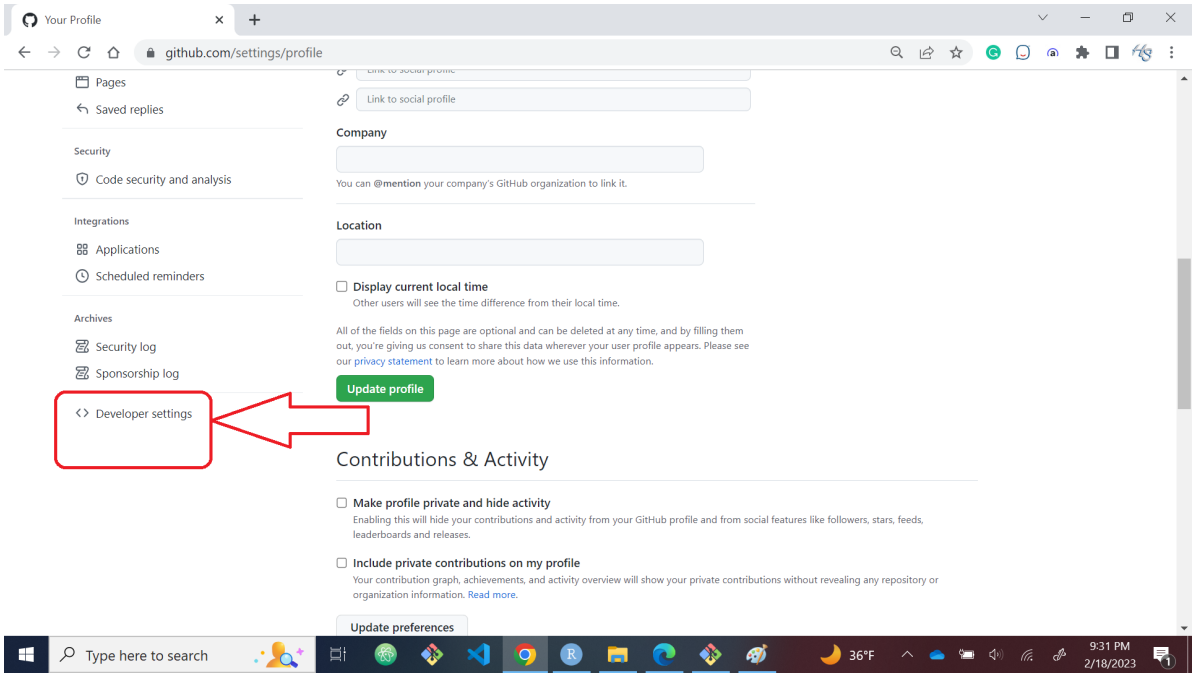
Generating Personal Access Token

1. **Fine-grained personal access tokens**

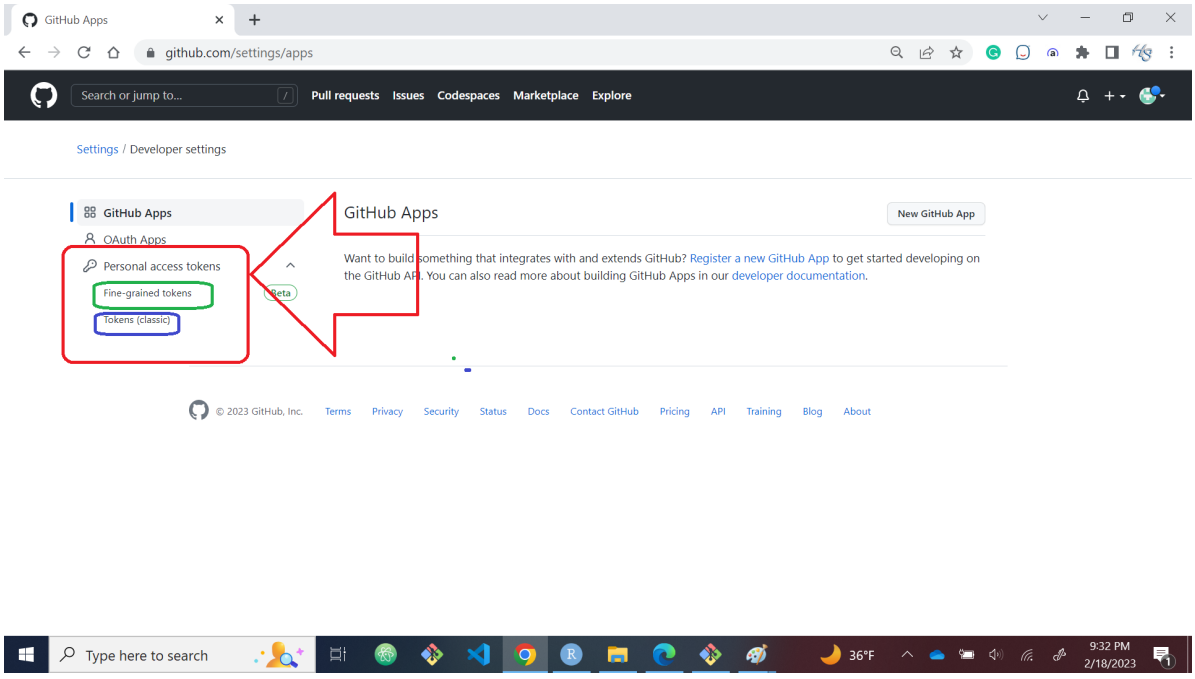
- Step 1. In the upper-right corner of any page, click on your avatar (or your profile photo), then click *Settings*.



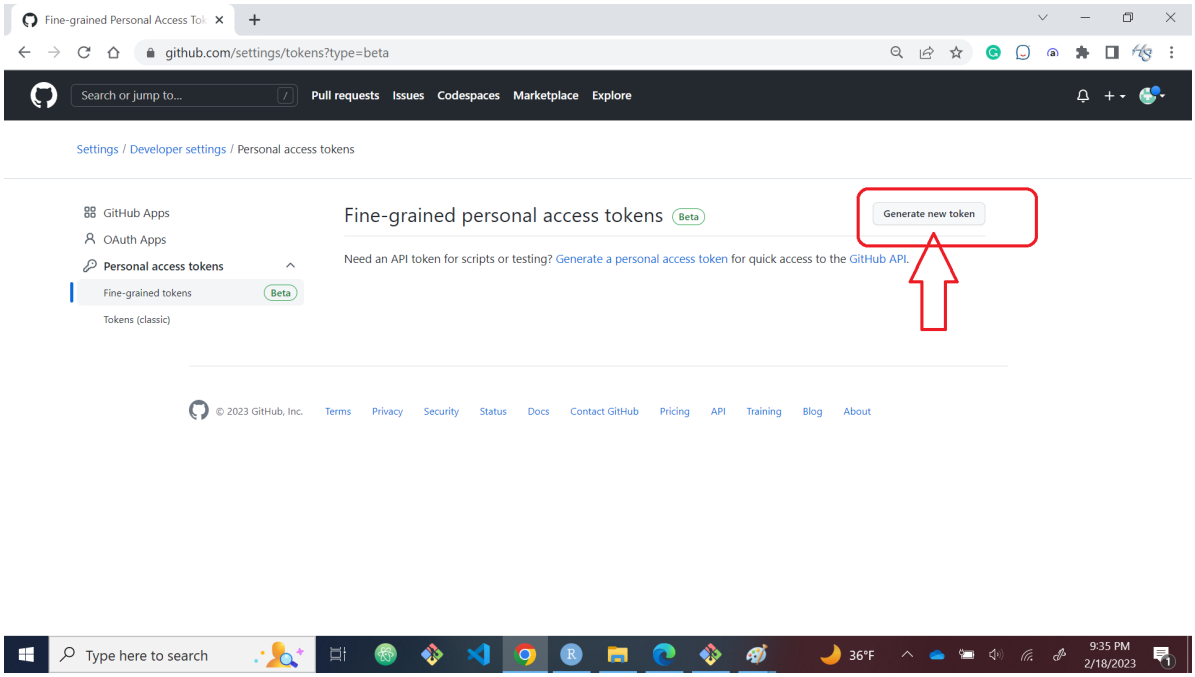
- Step 2. In the left sidebar, click on *Developer settings*. This should be at the end of left sidebar.



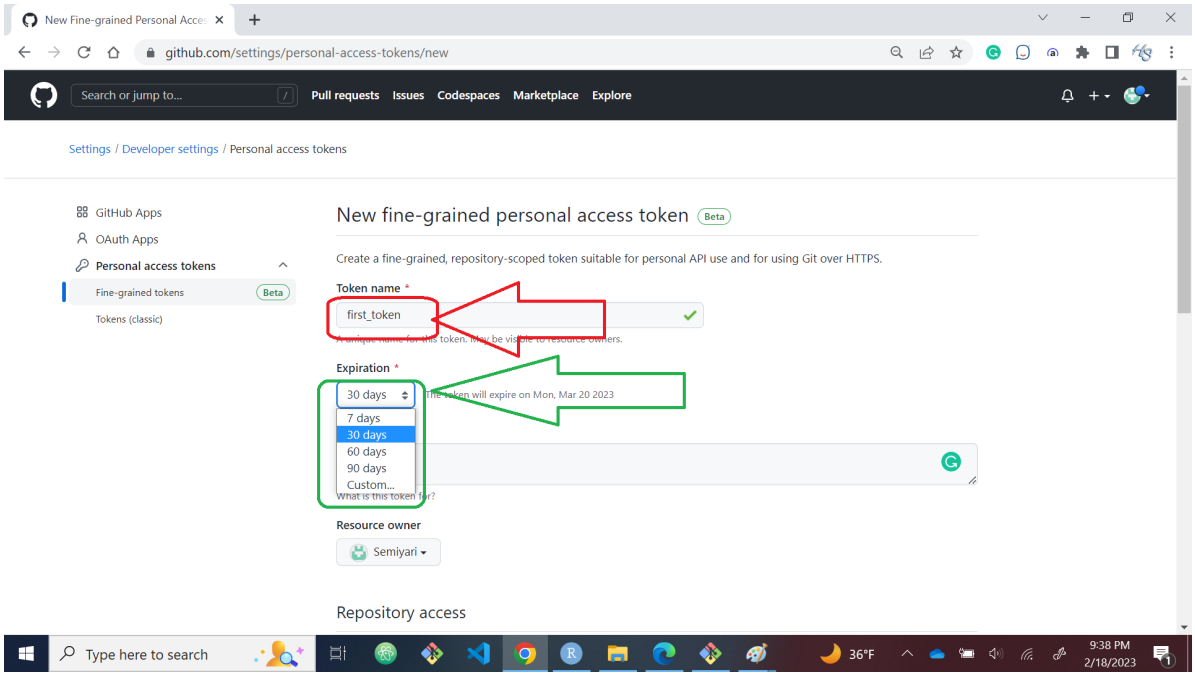
- Step 3. In the left sidebar, under *Personal access tokens*, click *Fine-grained tokens*.



- Step 4. Click *Generate new token*.

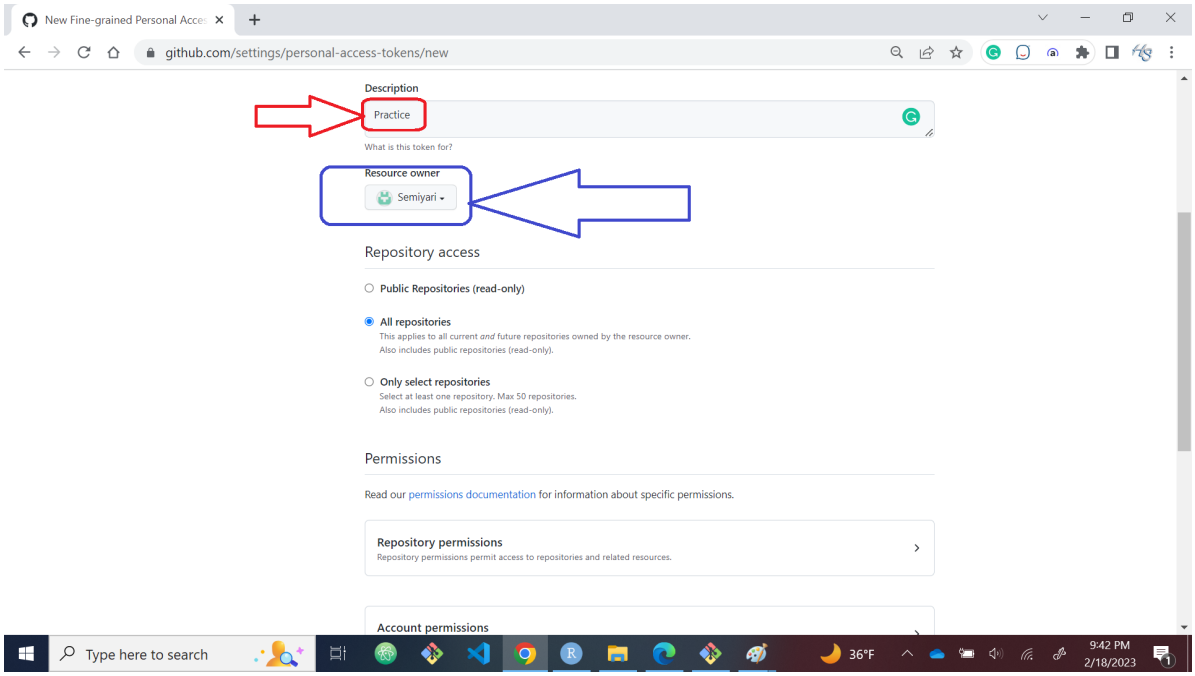


- Step 5. Under *Token name*, enter a name for the token. Under *Expiration*, select an expiration for the token.

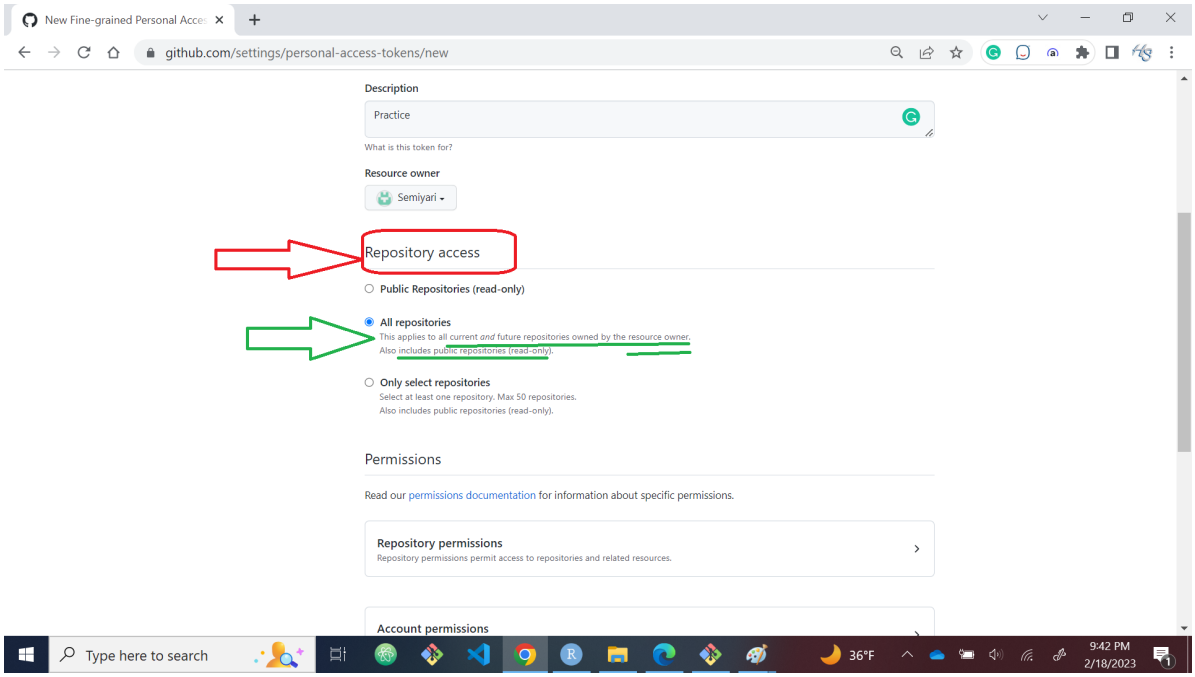


- Step 6. Optionally, under *Description*, add a note to describe the purpose of the token. Under Resource owner, select a resource owner. The token will only be able to access

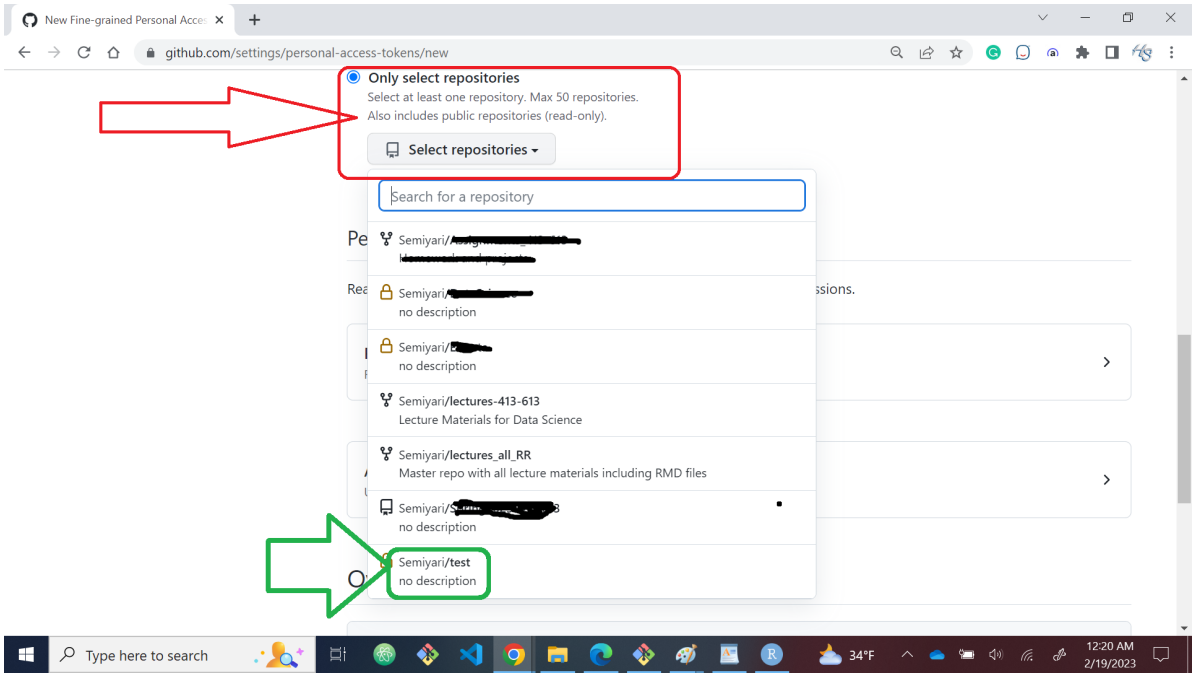
resources owned by the selected resource owner. Organizations that you are a member of will not appear unless the organization opted in to fine-grained personal access tokens. For more information, see Setting a personal access token policy for your organization.
Optionally, if the *resource owner* is an organization that requires approval for fine-grained personal access tokens, below the resource owner, in the box, enter a justification for the request.



- Step 7. Under *Repository access*, select which repositories you want the token to access. You should choose the minimal repository access that meets your needs. Tokens always include read-only access to all public repositories on GitHub.

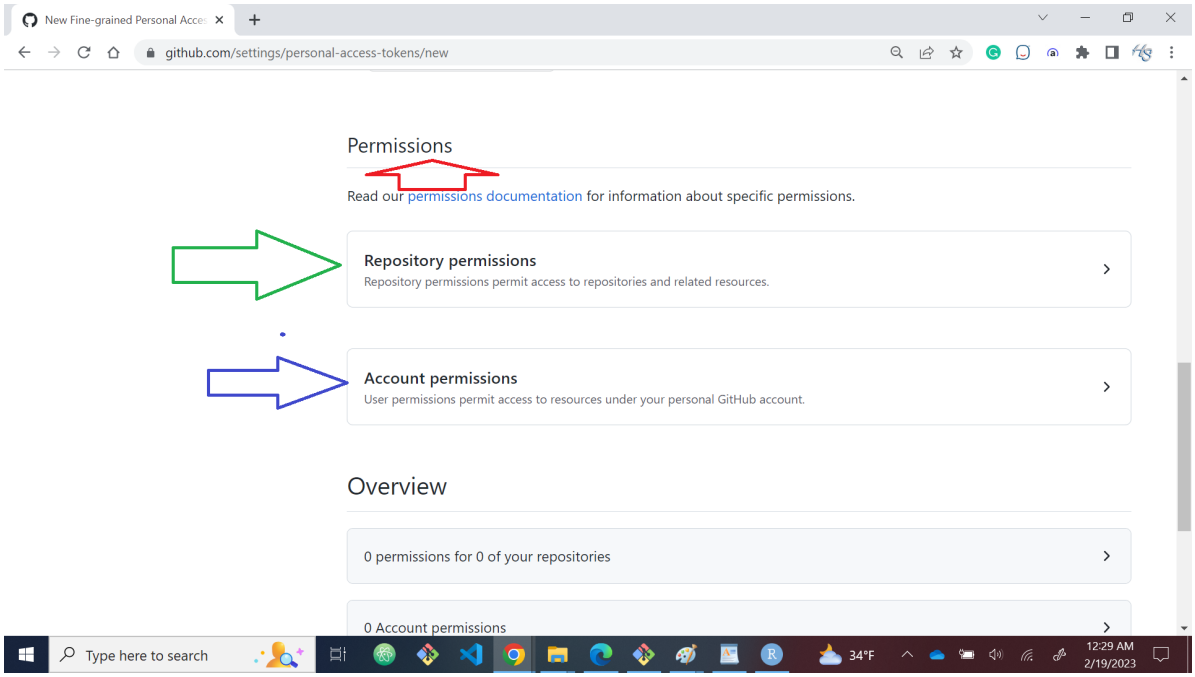


- Step 8. If you selected *Only select repositories* in the previous step, under the Selected repositories dropdown, select the repositories that you want the token to access.

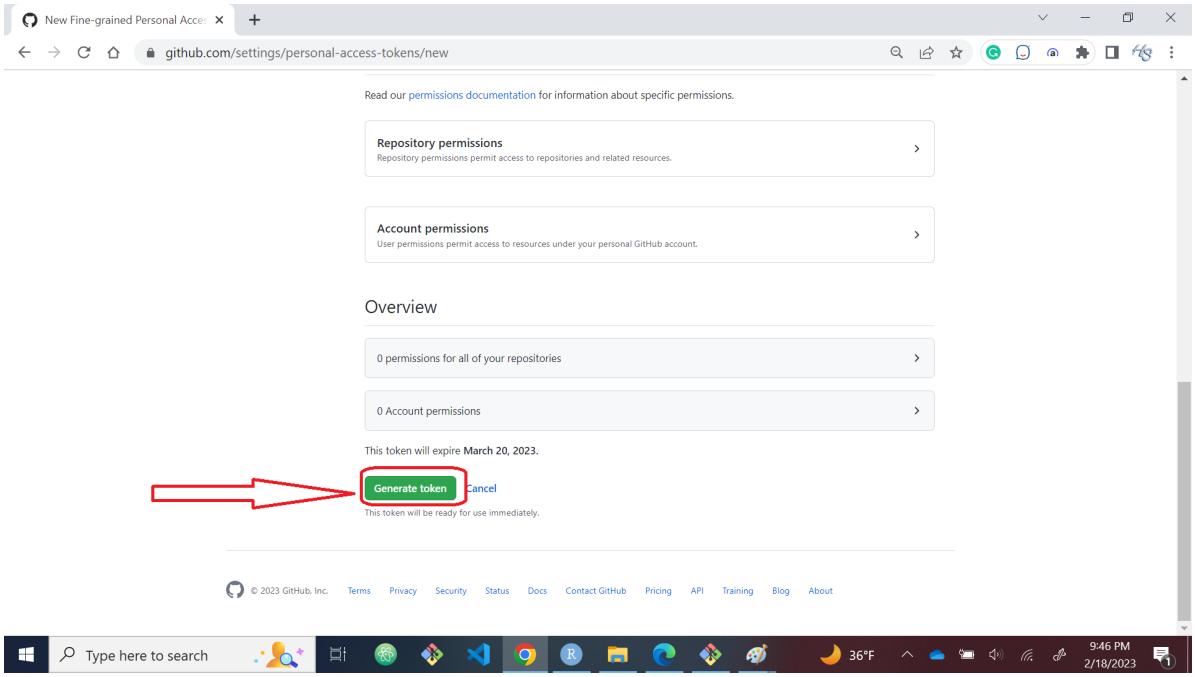


- Step 9. Under *Permissions*, select which permissions to grant the token. Depending

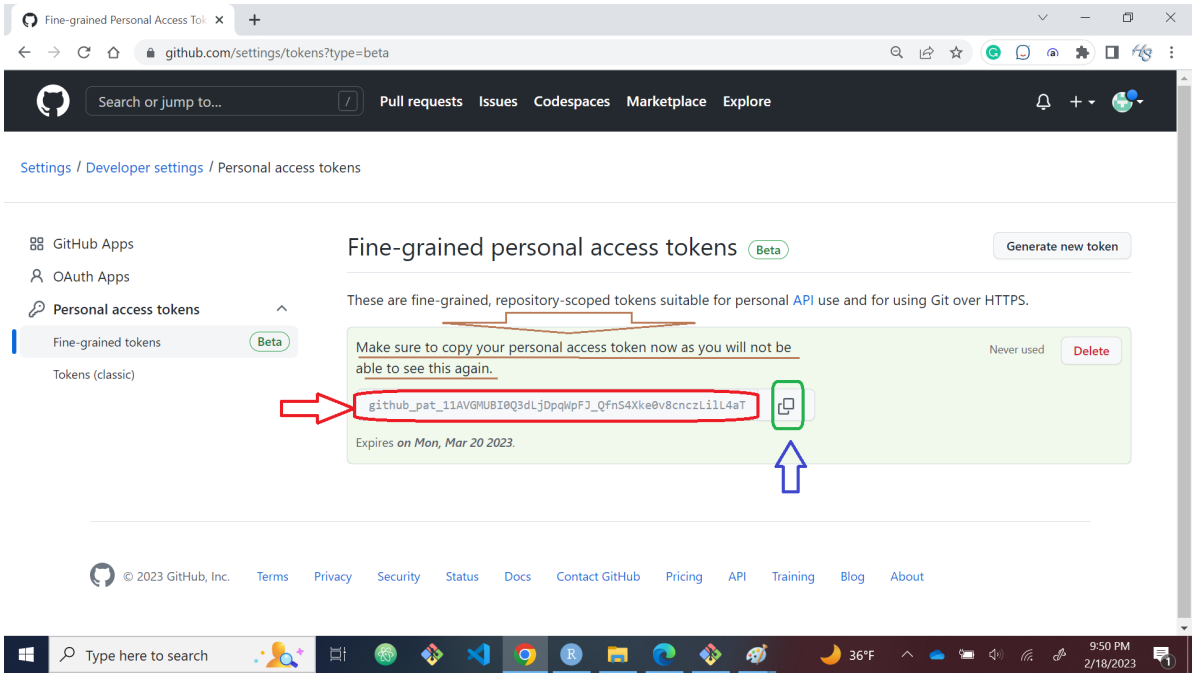
on which resource owner and which repository access you specified, there are repository, organization, and account permissions. You should choose the minimal permissions necessary for your needs. For more information about what permissions are required for each REST API operation, see Permissions required for fine-grained personal access tokens.



- Step 10. Click *Generate token*. If you selected an organization as the resource owner and the organization requires approval for fine-grained personal access tokens, then your token will be marked as `pending` until it is reviewed by an organization administrator. Your token will only be able to read public resources until it is approved. If you are an owner of the organization, your request is automatically approved. For more information, see Reviewing and revoking personal access tokens in your organization.



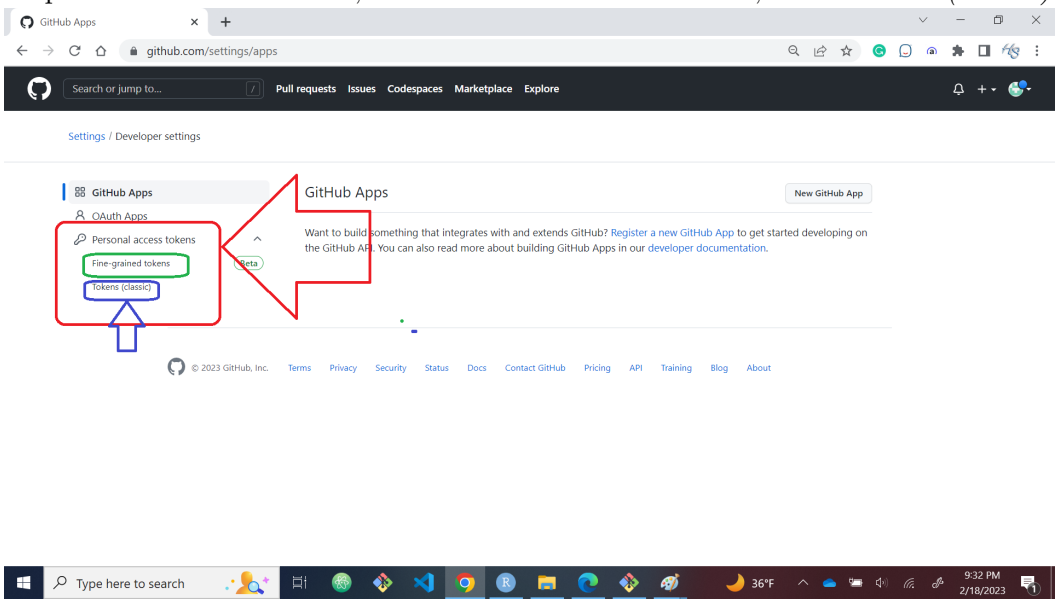
- Step 11. Copy and temporarily save the token before closing the window. You most likely will not be able to retrive it if you refresh or move away from the page.



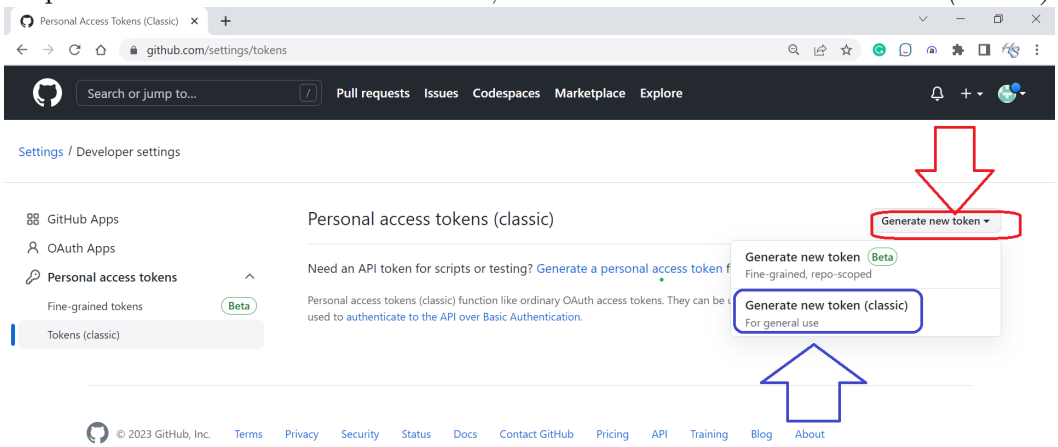
- Once you have a token, you can enter it instead of your password when performing Git operations over HTTPS. - Personal access tokens can only be used for *HTTPS* Git operations.

2. **Creating a personal access token (classic)**

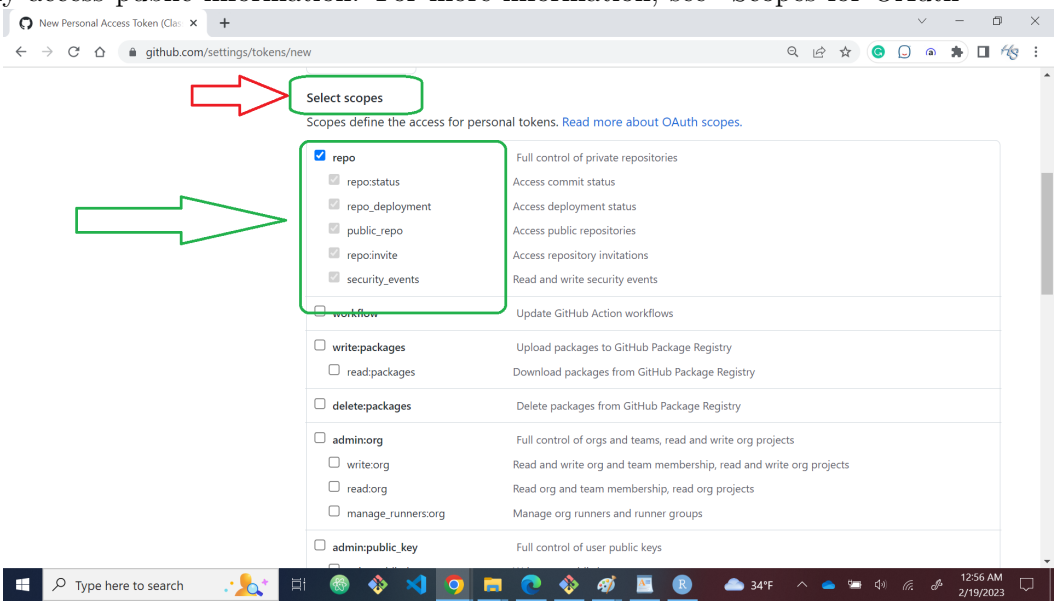
- The process is the same except in *Step 3*, *Step 4* and *Step 7*
  - Step 1. In the upper-right corner of any page, click on your avatar (or your profile photo), then click *Settings*.
  - Step 2. In the left sidebar, click on *Developer settings*.
  - Step 3. In the left sidebar, under *Personal access tokens*, click *Tokens (classic)*.



  - Step 4. Select *Generate new token*, then click *Generate new token (classic)*.



- Step 5. Give your token a descriptive name.
- Step 6. To give your token an expiration, select the Expiration drop-down menu, then click a default or use the calendar picker.
- Step 7. Select the *scopes* you'd like to grant this token. To use your token to access repositories from the command line, select repo. A token with no assigned scopes can only access public information. For more information, see "Scopes for OAuth Apps".



- Step 8. Click *Generate token.*

Authenticating with the command line

- You can access repositories on GitHub from the command line in two ways,
  1. HTTPS and
  2. SSH, and both have a different way of authenticating. The method of authenticating is determined based on whether you choose an HTTPS or SSH remote URL when you clone the repository. For more information about which way to access, see About remote repositories.
  3. *HTTPS*: You can work with all repositories on GitHub over HTTPS, *even if you are behind a firewall or proxy.*
  4. *SSH*: You can work with all repositories on GitHub over SSH, although *firewalls and proxies might refuse to allow SSH connections.*

Revisiting PAT

- Once you have a token, you can enter it instead of your password when performing Git operations over HTTPS.
  - Personal access tokens can only be used for *HTTPS* Git operations. If your repository uses a *SSH* remote URL, you will need to switch the remote from SSH to HTTPS.

witch the remote from SSH to HTTPS

1. Open Git Bash.
2. Change the current working directory to your local project.Command Line 101
3. List your existing remotes in order to get the name of the remote you want to change.

```
git remote -v

> origin  git@github.com:USERNAME/REPOSITORY.git (fetch)
> origin  git@github.com:USERNAME/REPOSITORY.git (push)
```

4. Change your remote's URL from SSH to HTTPS with the `git remote set-url` command.

```
git remote set-url origin https://github.com/USERNAME/REPOSITORY.git
```

5. Verify that the remote URL has changed.

```
git remote -v

# Verify new remote URL
> origin  https://github.com/USERNAME/REPOSITORY.git (fetch)
> origin  https://github.com/USERNAME/REPOSITORY.git (push)
```

The next time you `git fetch`, `git pull`, or `git push` to the remote repository, you'll be asked for your GitHub username and password. When Git prompts you for your password, enter your personal access token.

- **Question: How about if we want to switch from HTTPS to SHH?**
  The process is similar except in *part 4*, we do as follow
- Change your remote's URL from HTTPS to SSH with the git remote set-url command.

```
git remote set-url origin git@github.com:USERNAME/REPOSITORY.git
```