



Model Context Protocol

A protocol for seamless integration between LLM applications and external data sources

The Model Context Protocol (MCP) is an open protocol that enables seamless integration between LLM applications and external data sources and tools. Whether you're building an AI-powered IDE, enhancing a chat interface, or creating custom AI workflows, MCP provides a standardized way to connect LLMs with the context they need.

GitHub MCP Server

MCP Server for the GitHub API, enabling file operations, repository management, search functionality, and more.

Features

- Automatic Branch Creation: When creating/updating files or pushing changes, branches are automatically created if they don't exist
- Comprehensive Error Handling: Clear error messages for common issues
- Git History Preservation: Operations maintain proper Git history without force pushing
- Batch Operations: Support for both single-file and multi-file operations
- Advanced Search: Support for searching code, issues/PRs, and users

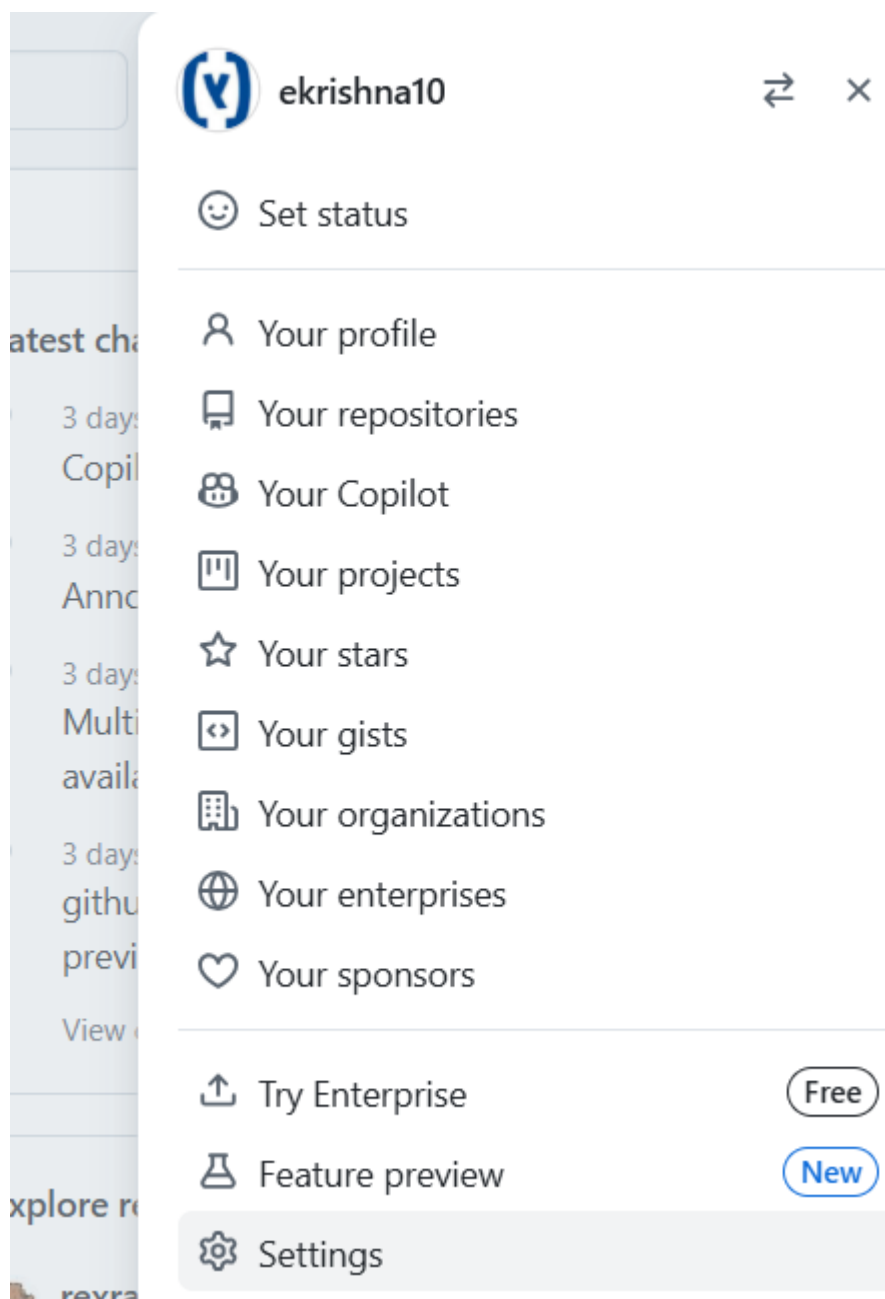
Source: all operations mentioned below are possible

<https://github.com/modelcontextprotocol/servers/tree/main/src/github>

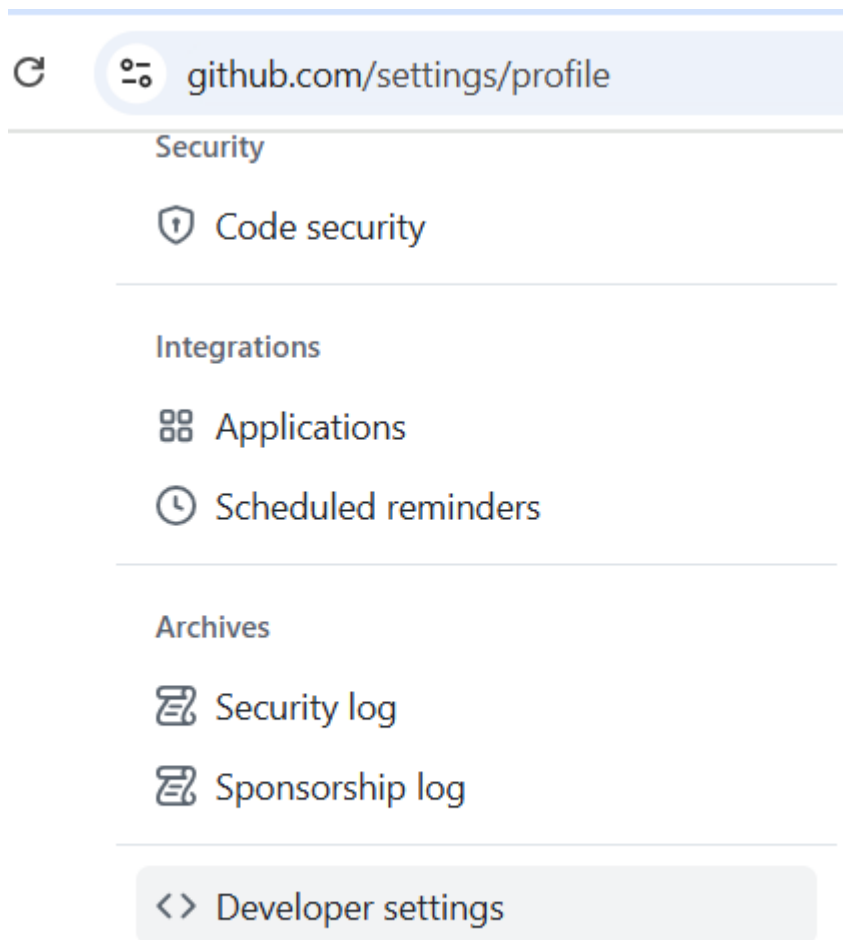
Personal Access Token is needed to connect to GitHub :

Steps to Generate a GitHub Personal Access Token:

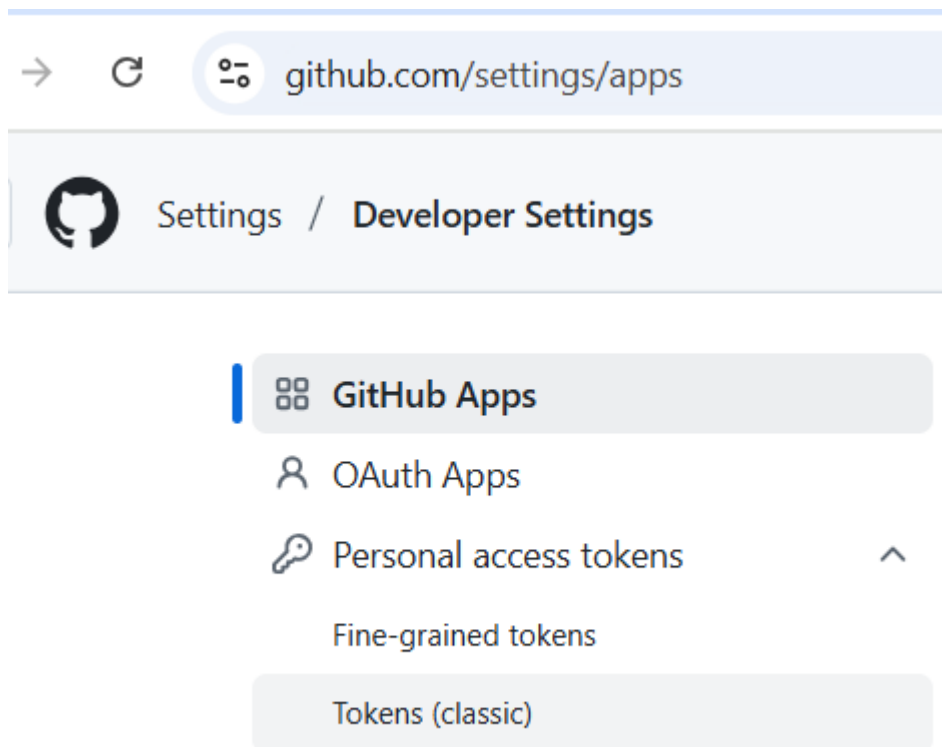
1. **Sign in to your GitHub account .**
2. **Go to your profile settings:** Click on your profile image in the upper right corner, then click "Settings".



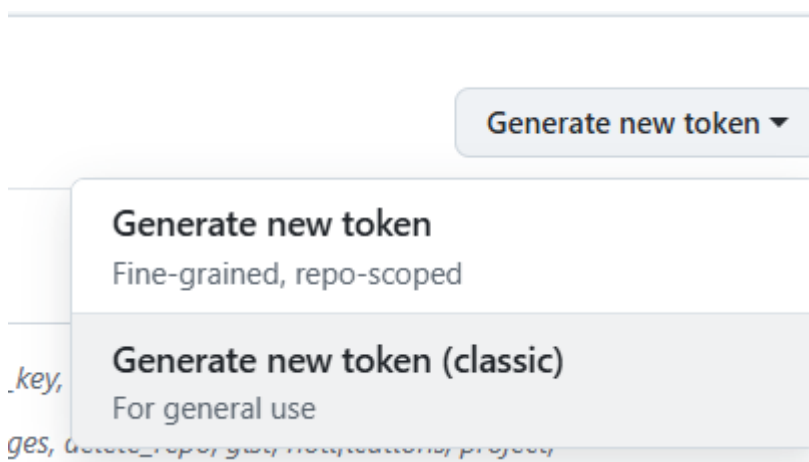
3. **Navigate to Developer settings:** In the left sidebar, click "Developer settings".



4. **Access Personal access tokens:** In the left sidebar, under "Personal access tokens", click "Tokens (classic)".



5. **Generate a new token:** Click "Generate new token".



6. **Provide a token description:** Enter a descriptive name for your token to help you remember its purpose.
7. **Set token expiration (optional):** Choose an expiration time for the token (e.g., 30 days, 60 days, 90 days, or no expiration). Note: It's recommended to set an expiration for security reasons.
8. **Select scopes/permissions:** Choose the scopes or permissions you want to grant this token. For example, select "repo" if you need the token for repository operations.

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

mcpgithub

What's this token for?

Expiration

 30 days (May 07, 2025) ▾

The token will expire on the selected date

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 <input checked="" type="checkbox"/> read:packages | Upload packages to GitHub Package Registry Download packages from GitHub Package Registry |
| <input checked="" type="checkbox"/> delete:packages | Delete packages from GitHub Package Registry |
| <input checked="" type="checkbox"/> admin:org <input checked="" type="checkbox"/> write:org <input checked="" type="checkbox"/> read:org <input checked="" type="checkbox"/> manage_runners:org | Full control of orgs and teams, read and write org projects Read and write org and team membership, read and write org projects Read org and team membership, read org projects Manage org runners and runner groups |
| <input checked="" type="checkbox"/> admin:public_key <input checked="" type="checkbox"/> write:public_key <input checked="" type="checkbox"/> read:public_key | Full control of user public keys Write user public keys Read user public keys |
| <input checked="" type="checkbox"/> admin:repo_hook <input checked="" type="checkbox"/> write:repo_hook <input checked="" type="checkbox"/> read:repo_hook | Full control of repository hooks Write repository hooks Read repository hooks |
| <input checked="" type="checkbox"/> admin:org_hook | Full control of organization hooks |
| <input checked="" type="checkbox"/> gist | Create gists |
| <input checked="" type="checkbox"/> notifications | Access notifications |
| <input checked="" type="checkbox"/> user <input checked="" type="checkbox"/> read:user <input checked="" type="checkbox"/> user:email <input checked="" type="checkbox"/> user:follow | Update ALL user data Read ALL user profile data Access user email addresses (read-only) Follow and unfollow users |

9. **Generate the token:** Click "Generate token".

| | |
|---|---|
| <input checked="" type="checkbox"/> codespace | Full control of codespaces |
| <input checked="" type="checkbox"/> codespace:secrets | Ability to create, read, update, and delete codespace secrets |
| <input checked="" type="checkbox"/> copilot | Full control of GitHub Copilot settings and seat assignments |
| <input checked="" type="checkbox"/> manage_billing:copilot | View and edit Copilot Business seat assignments |
| <input checked="" type="checkbox"/> write:network_configurations | Write org hosted compute network configurations |
| <input checked="" type="checkbox"/> read:network_configurations | Read org hosted compute network configurations |
| <input checked="" type="checkbox"/> project | Full control of projects |
| <input checked="" type="checkbox"/> read:project | Read access of projects |
| <input checked="" type="checkbox"/> admin:gpg_key | Full control of public user GPG keys |
| <input checked="" type="checkbox"/> write:gpg_key | Write public user GPG keys |
| <input checked="" type="checkbox"/> read:gpg_key | Read public user GPG keys |
| <input checked="" type="checkbox"/> admin:ssh_signing_key | Full control of public user SSH signing keys |
| <input checked="" type="checkbox"/> write:ssh_signing_key | Write public user SSH signing keys |
| <input checked="" type="checkbox"/> read:ssh_signing_key | Read public user SSH signing keys |

Generate token
Cancel

10. Securely store the token: Copy and store the generated token in a secure location, as you will not be able to view it again.

✓ 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
Delete

mcp — *admin:enterprise, admin:gpg_key, admin:org, admin:org_hook, admin:public_key, admin:repo_hook, admin:ssh_signing_key, audit_log, codespace, copilot, delete:packages, delete_repo, gist, notifications, project, repo, user, workflow, write:discussion, write:network_configurations, write:packages*
Last used within the last week
Delete

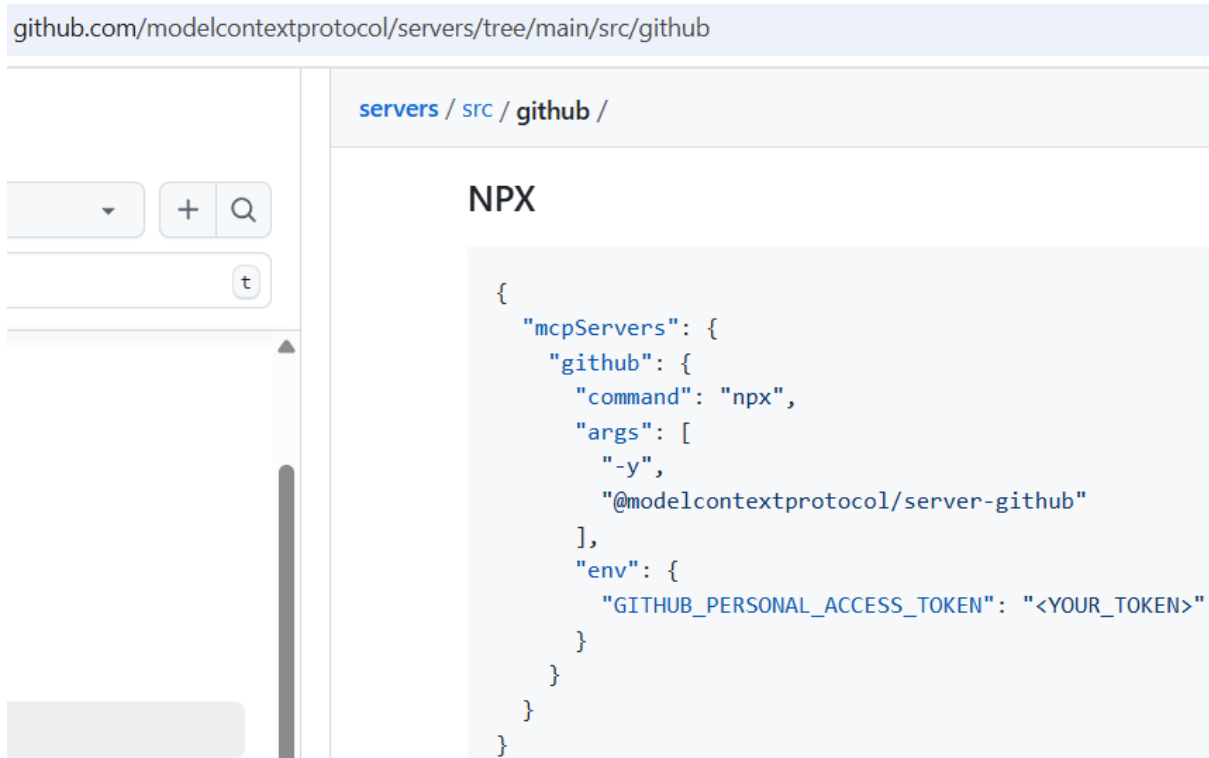
Expires **on Wed, May 7 2025.**

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](#).

Make sure you not down the token .

Token : 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

Replace the token in NPX of github:



Goto > claudes desktop > File > Settings > Developer > Edit Config>

"C:\Users\ekris\AppData\Roaming\Claude\claude_desktop_config.json"

And paste the below content to the config.json file

```
{
  "mcpServers": {
    "github": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-github"
      ],
      "env": {
        "GITHUB_PERSONAL_ACCESS_TOKEN": "ghp_qkBNMQG0mtQJx8IyboSNi19n06yc3L3K2FKa"
      }
    }
  }
}
```



```
{ } claude_desktop_config.json ●
C: > Users > ekris > AppData > Roaming > Claude > { } claude_desktop_config.json > ...
1 {
2   "mcpServers": {
3     "github": {
4       "command": "npx",
5       "args": [
6         "-y",
7         "@modelcontextprotocol/server-github"
8       ],
9       "env": {
10        "GITHUB_PERSONAL_ACCESS_TOKEN": "ghp_qkBnMQG0mtQJx8IyboSni19n06yc3L3K2FKa"
11      }
12    }
13  }
14 }
```

Restart the claude desktop and check the settings



GitHub account repositories :


Top repositories

 New

Find a repository...

-  ekrishna10/Python
-  ekrishna10/seaborn
-  ekrishna10/Numpy
-  ekrishna10/EDA
-  ekrishna10/Pandas
-  ekrishna10/matplotlib
-  ekrishna10/one

Create new repository called as mcpgithub

 create repository **mcpgithub**

I'll create a new GitHub repository called "mcpgithub" for you.

View result from **create_repository** from **github** (local) >

Great! I've successfully created a new GitHub repository:

Repository name: **mcpgithub**

Owner: ekrishna10

Description: My GitHub project repository

URL: <https://github.com/ekrishna10/mcpgithub>

The repository has been initialized with a README.md file and is public. Your repository is now ready to use! Would you like to add any specific files or make any other changes to the repository?

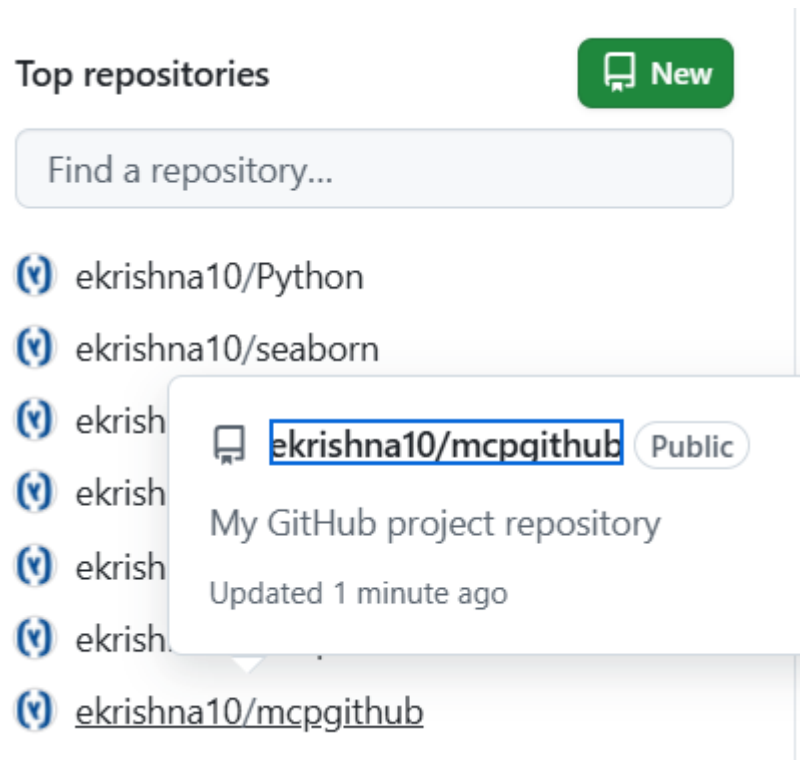
 Claude does not have internet access. Links provided may not be accurate or up to date.

   Retry



Claude can make mistakes. Please double-check responses.

Goto github and check now for repository:



mcpgithub repository is created successfully



Goto to repository and check for One.java

mcpgithub Public

Pin Unwatch 1

main 1 Branch 0 Tags

Go to file

Add file

Code

ekrishna10 Add One.java

c64884d · 1 minute ago 1 Commit

One.java

Add One.java

1 minute ago

mcpgithub

+ Search

t

mcpgithub / One.java

ekrishna10 Add One.java

Code

Blame

5 lines (5 loc) · 123 Bytes

Code 55% faster with GitHub Copilot

```
1 public class One {
2     public static void main(String[] args) {
3         System.out.println("Hello from One.java!");
4     }
5 }
```