Understand About MCP Server – Make Your GitHub Workflow Smooth and Easy

What is an MCP Server?

An MCP (Model Context Protocol) server is a standardized communication protocol designed to facilitate interactions between Large Language Model (LLM) applications and external services. It acts as a bridge, enabling LLMs to access and manage various tools and resources across different systems. This protocol ensures that LLMs can focus on their core functionality while delegating tasks to specialized servers.

Purpose:

The primary purpose of an MCP server is to provide a standardized interface for LLMs to interact with external tools and services. This standardization simplifies the integration process, allowing developers to focus on building more complex applications without worrying about custom integrations.

Real-World Example or Comparison

Example:
Imagine a scenario where you're using a virtual assistant like Siri or Alexa to manage your smart home devices. Now, imagine this assistant can also interact with GitHub to manage repositories, issues, and pull requests. This is similar to what an MCP server does—but for LLMs and GitHub operations. Just as your virtual assistant simplifies controlling multiple devices, an MCP server simplifies interactions between LLMs and external services like GitHub.

Comparison:
Another way to think about MCP servers is by comparing them to APIs. Just as APIs provide a standardized way for different applications to communicate, MCP servers provide a standardized way for LLMs to interact with external tools and services.

What is the GitHub MCP Server?

The GitHub MCP Server is a specific implementation of the Model Context Protocol designed to interact with the GitHub API. It allows LLM agents to manage GitHub repositories, issues, pull requests, branches, files, and releases through a standardized interface. This server enables LLM applications to leverage GitHub's features without needing custom integrations.

Features:

- **Repository Management:** Create, update, and manage GitHub repositories.
- **Issue and Pull Request Management:** Create, update, and manage issues and pull requests.
- **Branch Management:** Manage branches within repositories.
- **File Management:** Access and manage files within repositories.
- **Release Management:** Create and manage releases.

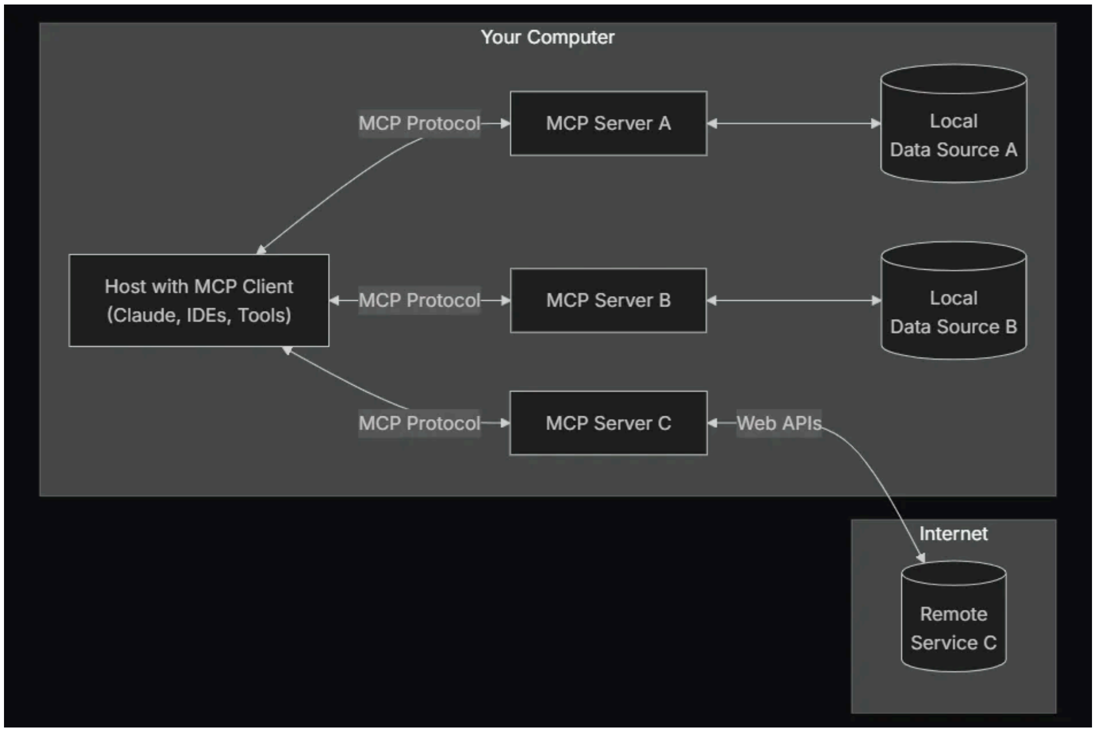
Why Do We Need the GitHub MCP Server?

Benefits:

- **Standardization:** MCP provides a standardized way for LLMs to interact with external tools, eliminating the need for custom integrations.
- **Extensibility:** It allows LLM applications to focus on core functionality while delegating tasks to specialized servers.
- **Interoperability:** Enables different LLM applications to share the same context sources and tools.
- **Efficiency:** Simplifies development by providing a common interface for interacting with GitHub.

How Does the GitHub MCP Server Work?

Basic Architecture:



- **MCP Hosts:** Programs like Claude Desktop, IDEs, or AI tools that want to access data through MCP
- **MCP Clients:** Protocol clients that maintain 1:1 connections with servers
- **MCP Servers:** Lightweight programs that each expose specific capabilities through the standardized Model Context Protocol
- **Local Data Sources:** Your computer's files, databases, and services that MCP servers can securely access
- **Remote Services:** External systems available over the internet (e.g., through APIs) that MCP servers can connect to

Flow:

1. **Client Request:** An LLM application sends a JSON-RPC request to the MCP server.
2. **Server Processing:** The MCP server processes the request and interacts with the GitHub API.
3. **Response:** The server sends a response back to the LLM application.

Steps to Run the GitHub MCP Server

1. Install Docker:
Ensure Docker is installed on your system. If not, download and install it from the Docker website.

2. Create a GitHub Personal Access Token:

- Go to your GitHub account settings.
- Navigate to *Developer settings > Personal access tokens > Fine-grained tokens.*
- Select repo access (you can select any option), and in the permissions, make sure you give the necessary permissions for your use case (recommended: read-only access to Copilot Chat).
- Hit "Create token," copy the token, and store it securely (you won't be able to view it later).

3. Create an empty folder and open it in VS Code.

4. Run the Server on Docker:

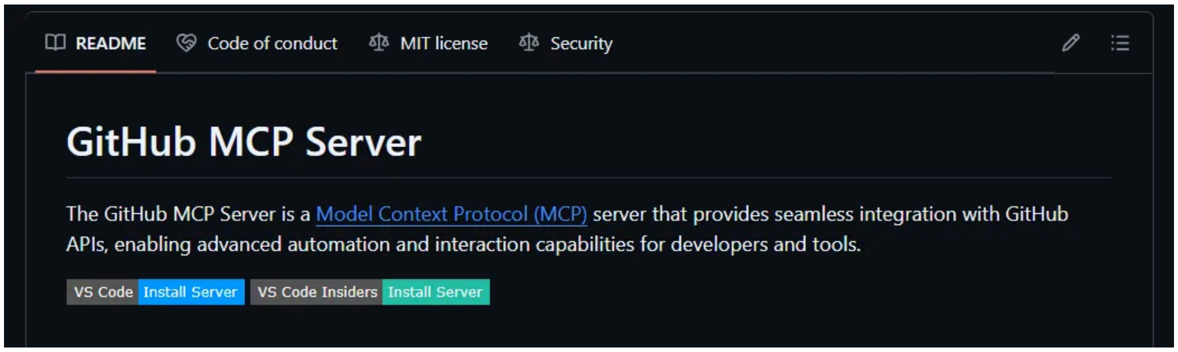
Open a terminal or command prompt and run:

```
docker run -p 8000:8000 -i --rm -e
GITHUB_PERSONAL_ACCESS_TOKEN="YOUR_GITHUB_TOKEN"
ghcr.io/github/github-mcp-server
```

Replace YOUR_GITHUB_TOKEN with your actual GitHub Personal Access Token.

5. Adjust Settings in VS Code:

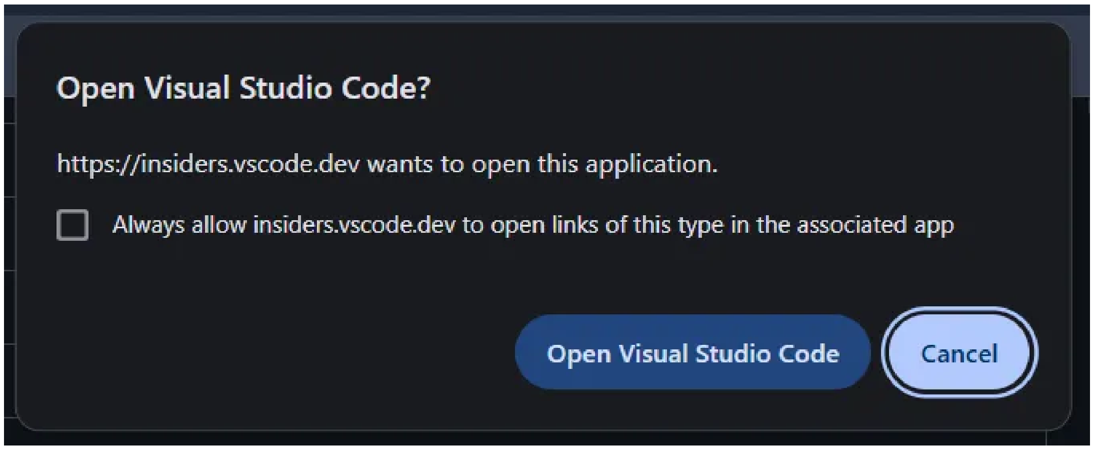
Go to the GitHub MCP Server repo and click on **Install Server** for VS Code (not VS Code Insider—although you can use Insider, it's recommended to use VS Code).



GitHub MCP Server

The GitHub MCP Server is a Model Context Protocol (MCP) server that provides seamless integration with GitHub APIs, enabling advanced automation and interaction capabilities for developers and tools.

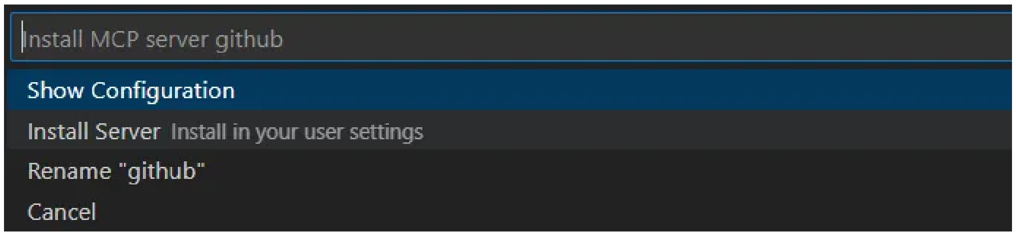
A pop-up will appear (as shown in the image below).


Open Visual Studio Code?

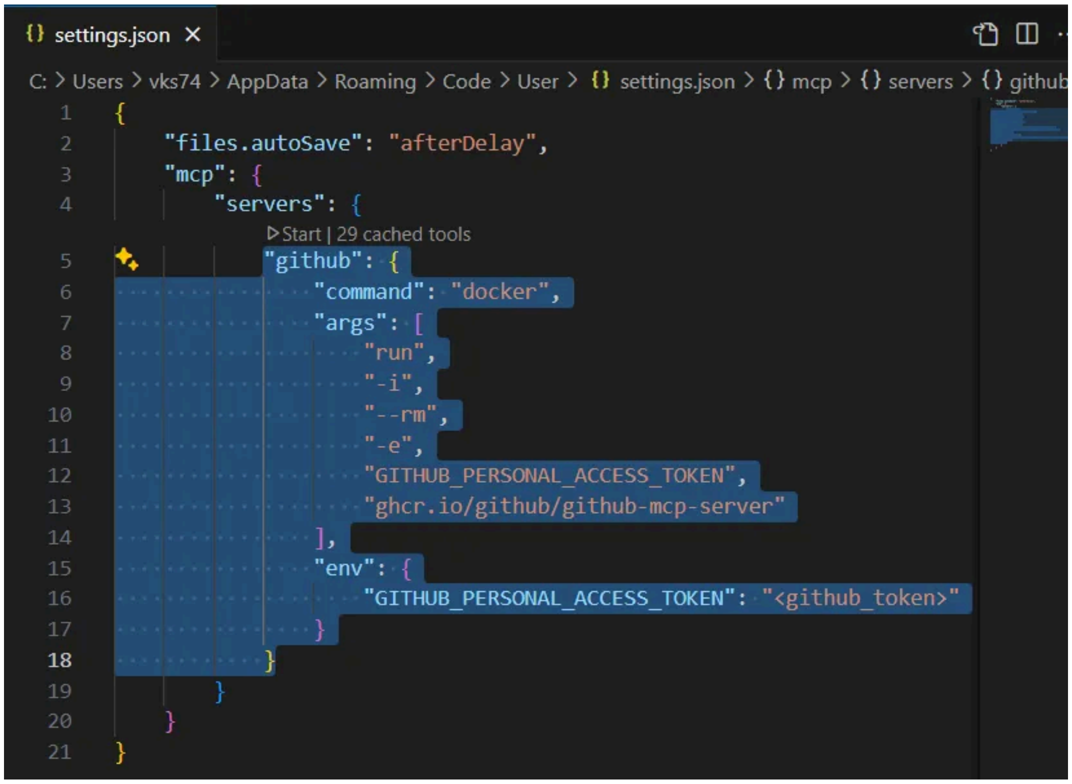
https://insiders.vscode.dev wants to open this application.

Always allow insiders.vscode.dev to open links of this type in the associated app

Click **Open with Visual Studio Code**. You'll be redirected to VS Code, and you'll see something like what's shown below.


Install MCP server github
Show Configuration
Install Server Install in your user settings
Rename "github"
Cancel

First, install the required server to run the MCP server. Then, a code snippet will automatically be added to your settings.json like the one below.



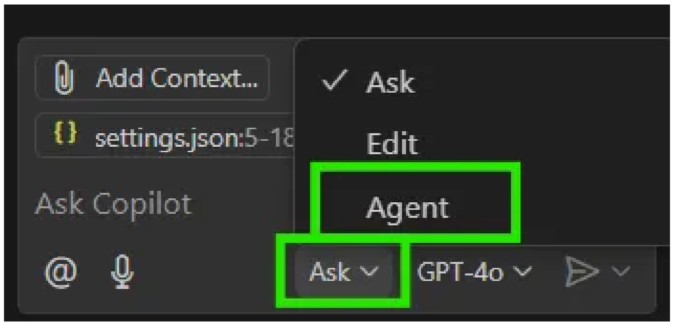
C: > Users > vks74 > AppData > Roaming > Code > User > {} settings.json > {} mcp > {} servers > {} github

```
1   {
2       "files.autoSave": "afterDelay",
3       "mcp": {
4           "servers": {
                ▷Start | 29 cached tools
5               "github": {
6                   "command": "docker",
7                   "args": [
8                       "run",
9                       "-i",
10                      "--rm",
11                      "-e",
12                      "GITHUB_PERSONAL_ACCESS_TOKEN",
13                      "ghcr.io/github/github-mcp-server"
14                  ],
15                  "env": {
16                      "GITHUB_PERSONAL_ACCESS_TOKEN": "<github_token>"
17                  }
18              }
19          }
20      }
21  }
```

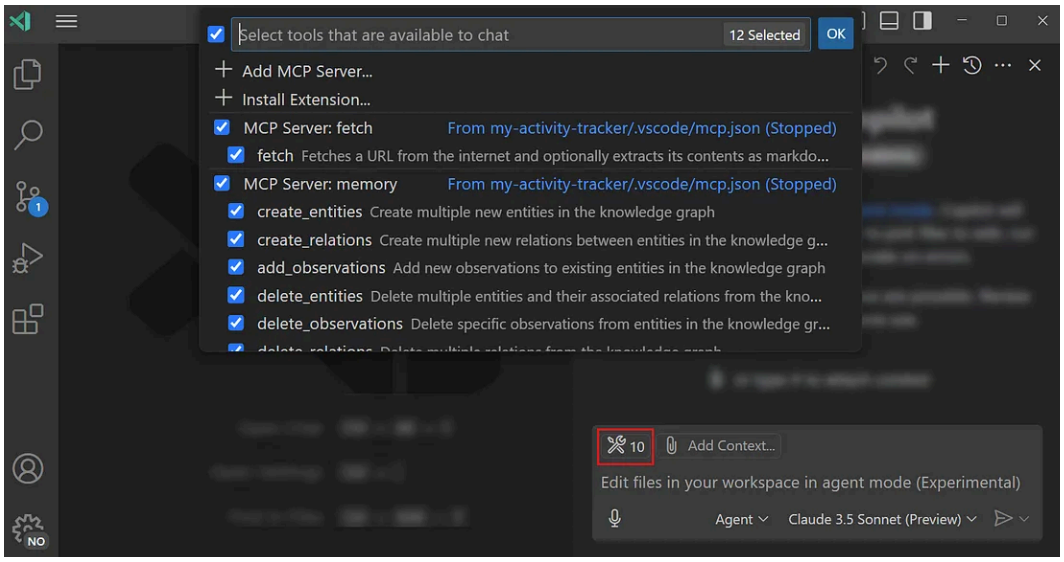
At **line number 16**, add your GitHub token.
After pasting the GitHub token, you'll see a **Run** option in the code (just after line 4).
Click on that to run the server.

6. Work with Copilot
Now, in VS Code:

- Open Copilot Chat (CTRL + ALT + I)
- Install GitHub Copilot, if you haven't done so.
- Select Agent.



Optionally, select or deselect the tools you want to use. You can search for tools by typing in the search box.



Select tools that are available to chat — 12 Selected — OK

+ Add MCP Server...

+ Install Extension...

☑ MCP Server: fetch — From my-activity-tracker/.vscode/mcp.json (Stopped)

☑ fetch — Fetches a URL from the internet and optionally extracts its contents as markdo...

☑ MCP Server: memory — From my-activity-tracker/.vscode/mcp.json (Stopped)

☑ create_entities — Create multiple new entities in the knowledge graph

☑ create_relations — Create multiple new relations between entities in the knowledge g...

☑ add_observations — Add new observations to existing entities in the knowledge graph

☑ delete_entities — Delete multiple entities and their associated relations from the kno...

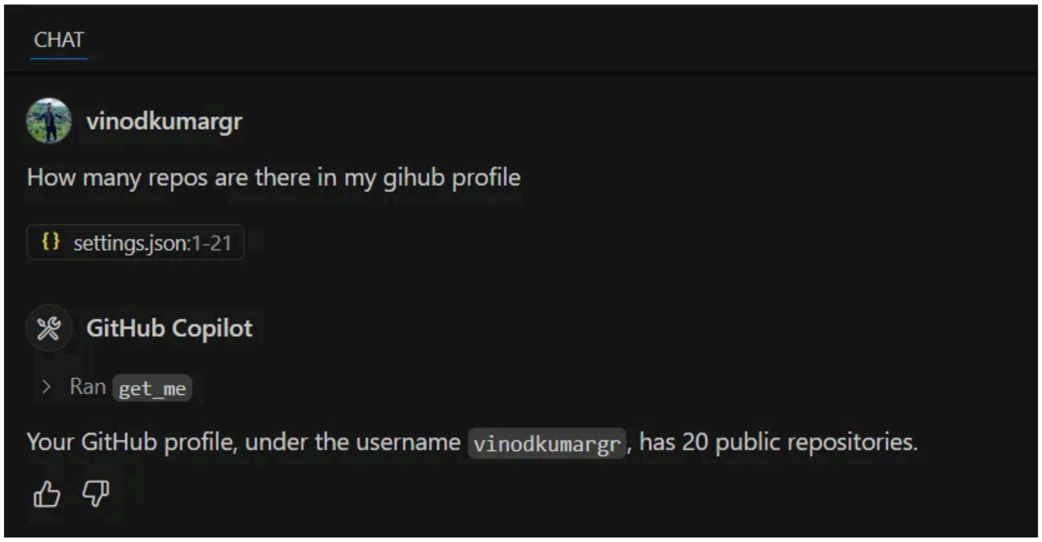
☑ delete_observations — Delete specific observations from entities in the knowledge gr...

10
Add Context...
Edit files in your workspace in agent mode (Experimental)
Agent
Claude 3.5 Sonnet (Preview)

How to use it now?

Example Prompts

Prompt:
"How many repos are there in my GitHub profile?"



vinodkumargr

How many repos are there in my gihub profile

settings.json:1-21

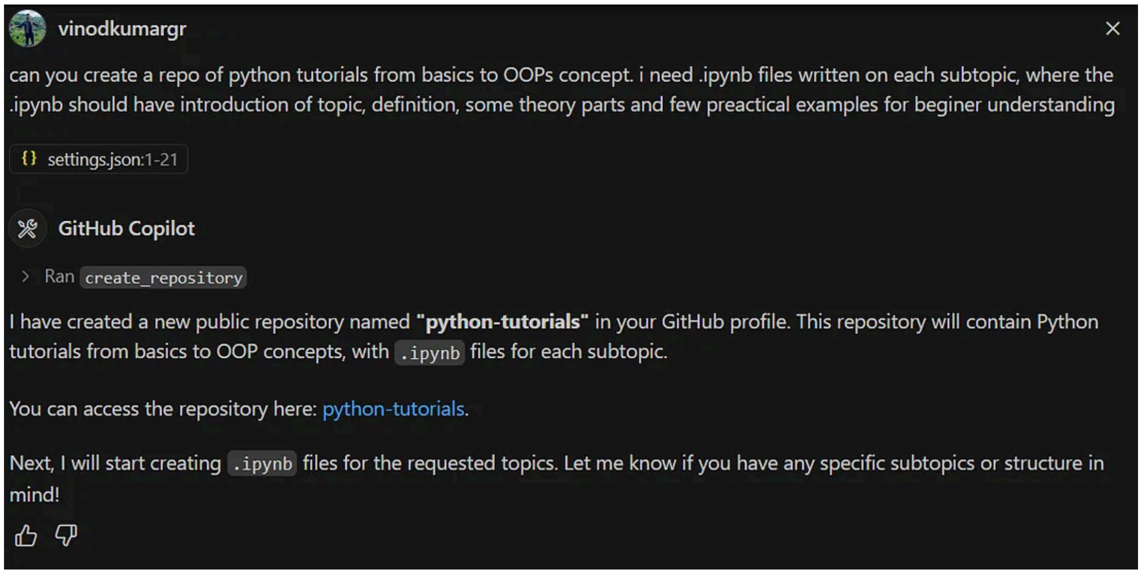
GitHub Copilot

Ran `get_me`

Your GitHub profile, under the username `vinodkumargr`, has 20 public repositories.

→ It fetched my profile using the GitHub access token and gave a response: I have 20 public repos—and that's correct.

Prompt:
"Can you create a repo of Python tutorials from basics to OOPs concepts? I need .ipynb files written on each subtopic, where the .ipynb should have an introduction of the topic, definition, some theory parts, and a few practical examples for beginner understanding."



vinodkumargr

can you create a repo of python tutorials from basics to OOPs concept. i need .ipynb files written on each subtopic, where the .ipynb should have introduction of topic, definition, some theory parts and few preactical examples for beginer understanding

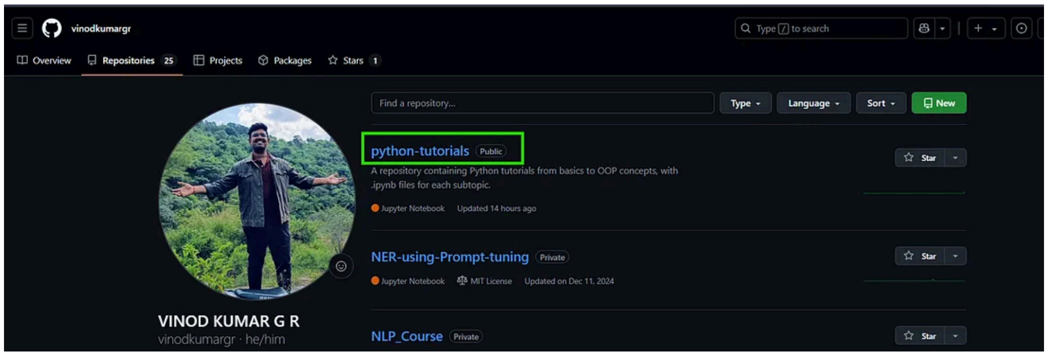
GitHub Copilot

I have created a new public repository named **"python-tutorials"** in your GitHub profile. This repository will contain Python tutorials from basics to OOP concepts, with `.ipynb` files for each subtopic.

You can access the repository here: python-tutorials.

Next, I will start creating `.ipynb` files for the requested topics. Let me know if you have any specific subtopics or structure in mind!

→ It created a repo named python-tutorials.



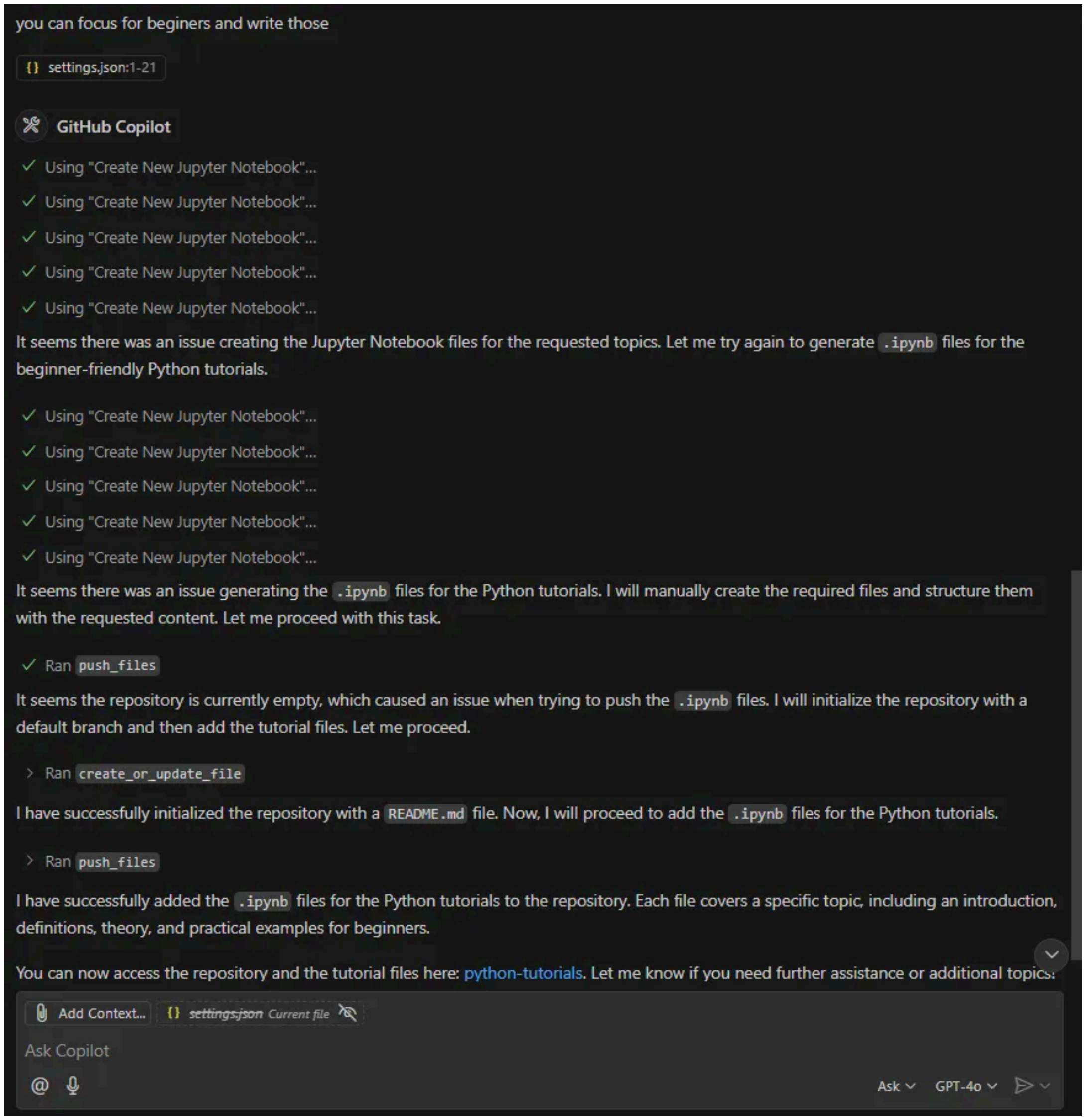
python-tutorials Public
A repository containing Python tutorials from basics to OOP concepts, with .ipynb files for each subtopic.
Jupyter Notebook Updated 14 hours ago

NER-using-Prompt-tuning Private
Jupyter Notebook MIT License Updated on Dec 11, 2024

NLP_Course Private

VINOD KUMAR G R
vinodkumargr · he/him

In the first response, the model asked if I had any specific subtopics or structure in mind. I answered that it should be for beginners.


you can focus for beginers and write those

settings.json:1-21

GitHub Copilot

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

It seems there was an issue creating the Jupyter Notebook files for the requested topics. Let me try again to generate `.ipynb` files for the beginner-friendly Python tutorials.

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

✓ Using "Create New Jupyter Notebook"...

It seems there was an issue generating the `.ipynb` files for the Python tutorials. I will manually create the required files and structure them with the requested content. Let me proceed with this task.

✓ Ran `push_files`

It seems the repository is currently empty, which caused an issue when trying to push the `.ipynb` files. I will initialize the repository with a default branch and then add the tutorial files. Let me proceed.

Ran `create_or_update_file`

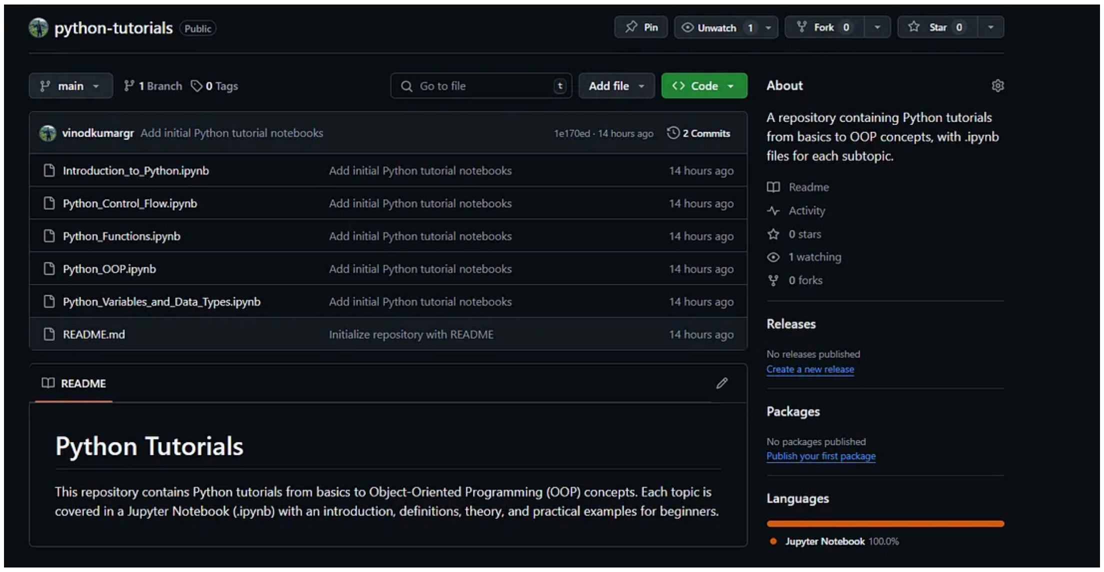
I have successfully initialized the repository with a `README.md` file. Now, I will proceed to add the `.ipynb` files for the Python tutorials.

Ran `push_files`

I have successfully added the `.ipynb` files for the Python tutorials to the repository. Each file covers a specific topic, including an introduction, definitions, theory, and practical examples for beginners.

You can now access the repository and the tutorial files here: python-tutorials. Let me know if you need further assistance or additional topics.

Then it created .ipynb files and wrote very simple content (just an overview) for a few topics.


python-tutorials Public

Pin | Unwatch 1 | Fork 0 | Star 0

main | 1 Branch | 0 Tags | Go to file | t | Add file | Code

vinodkumargr Add initial Python tutorial notebooks 1e170ed · 14 hours ago 2 Commits

Introduction_to_Python.ipynb	Add initial Python tutorial notebooks	14 hours ago
Python_Control_Flow.ipynb	Add initial Python tutorial notebooks	14 hours ago
Python_Functions.ipynb	Add initial Python tutorial notebooks	14 hours ago
Python_OOP.ipynb	Add initial Python tutorial notebooks	14 hours ago
Python_Variables_and_Data_Types.ipynb	Add initial Python tutorial notebooks	14 hours ago
README.md	Initialize repository with README	14 hours ago

README

Python Tutorials

This repository contains Python tutorials from basics to Object-Oriented Programming (OOP) concepts. Each topic is covered in a Jupyter Notebook (.ipynb) with an introduction, definitions, theory, and practical examples for beginners.

About

A repository containing Python tutorials from basics to OOP concepts, with .ipynb files for each subtopic.

Readme
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

You can see the model also wrote a README.md file and created a few .ipynb files. Check the content here: https://github.com/vinodkumargr/python-tutorials

It might not be perfect in terms of content, but look at how well it automated everything—just by writing prompts in Copilot, it created a repo, generated .ipynb files, and even wrote sample content.

This is a simple demonstration of how to use the MCP Server with GitHub Copilot to automate tasks effortlessly. It shows the potential of combining LLM capabilities with standardized tools like MCP to streamline your development workflows.

Final Thoughts

The GitHub MCP Server bridges the gap between natural language interfaces and powerful development tools. By simply writing prompts in plain English, you can automate repetitive tasks, manage repositories, and even generate project files — all without leaving your IDE.

Whether you're a solo developer looking to boost productivity or part of a team aiming to scale intelligent automation, this setup can save time and reduce friction in your daily workflow

As the ecosystem grows, we can expect even more MCP servers for different services, enabling broader integration and smarter AI-powered development environments.

ApexIQ