

Fabric Notebook
Custom Agent
Private Preview Docs

Overview

What is Fabric Notebook custom agent?

The [Fabric Data Engineering VS Code extension](#) provides Fabric users with Pro-dev experience within Visual Studio Code. This free extension enables data engineers working with Fabric to write notebook code and execute their Fabric Notebooks directly inside VS Code, enhancing workflow efficiency and integration with familiar development tools.

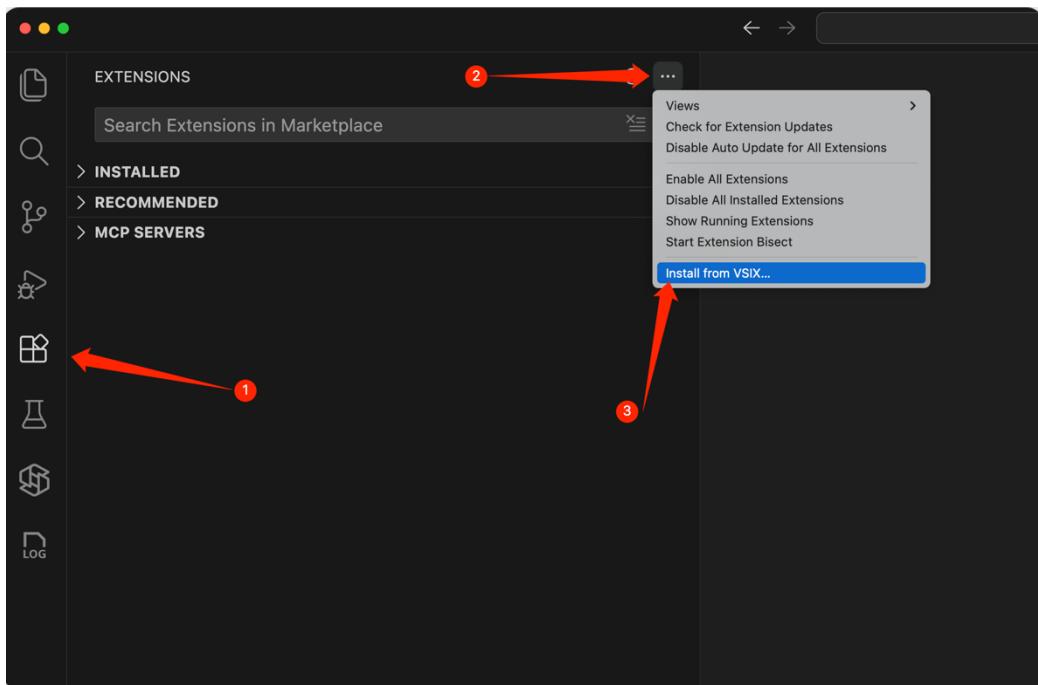
One advancement in this extension is the introduction of the Fabric Notebook custom agent with this preview update. This new feature is designed to enhance the coding experience specifically for notebooks, especially when using GitHub Copilot. The custom agent streamlines the development process, allowing for a smoother and more productive coding environment within Fabric Notebooks.

With this custom agent, GitHub Copilot will be set up to generate notebook code that adheres to Fabric's best practices and coding standards. For example, using relative path to access the file and table from the default Lakehouse of the notebook and using full ABFSS path for the regular Lakehouse.

In summary, this custom agent supplies the essential Fabric context required to guide GitHub Copilot in generating appropriate Fabric notebook code.

Pre-Requisite

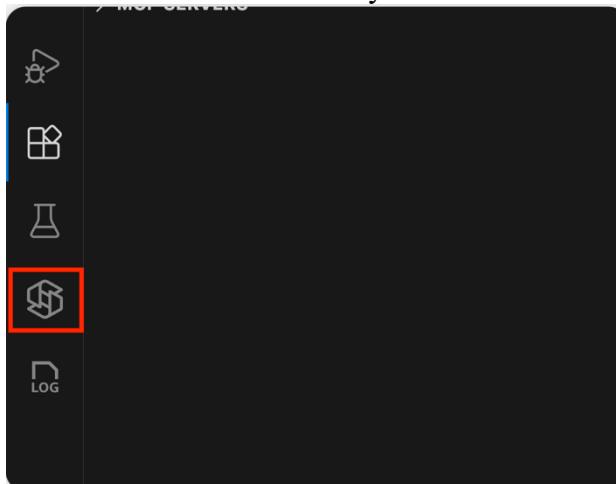
1. Download and install VS Code(<https://code.visualstudio.com/download>)
2. Install GitHub Copilot Chat extension to access the GitHub Copilot in VS Code (<https://code.visualstudio.com/docs/copilot/overview>)
3. Install the private preview version of the extension with the offline VSIX file, the file should be located inside the same folder of this doc.



Click the “Extensions” icon in the Activity Bar on the side of VS Code, click the “...” option and select “Install from VSIX...”

You can also use the Extensions: Install from VSIX command in the **Command Palette**, point to the .vsix file.

Once the installation is completed, you should be able to see the Fabric Data Engineering extension icon in the Activity Bar

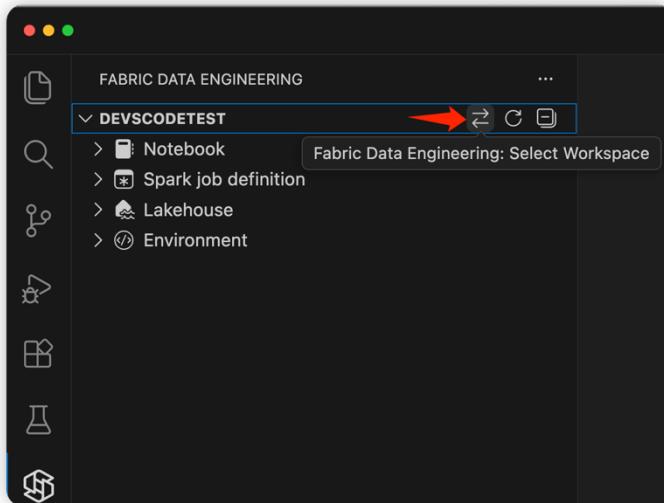


Get Started

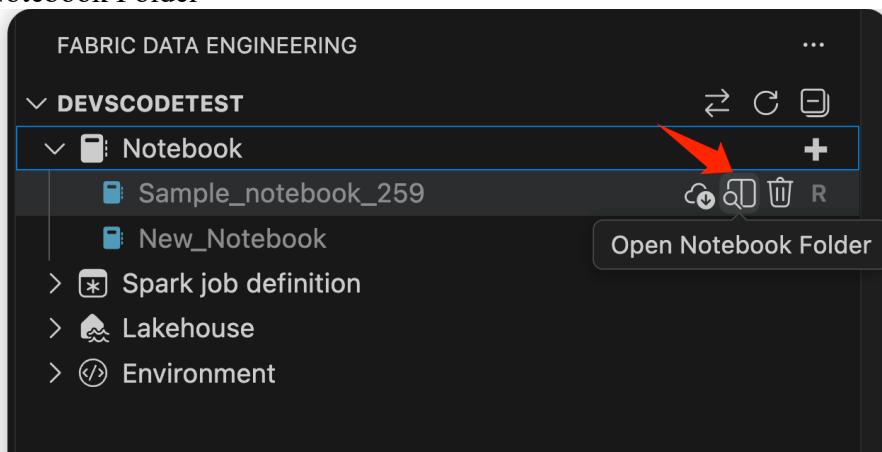
Open notebook under VFS mode

With VFS(Virtual File System) support in VS Code, you can directly open the Fabric Notebook inside VS Code without downloading it.

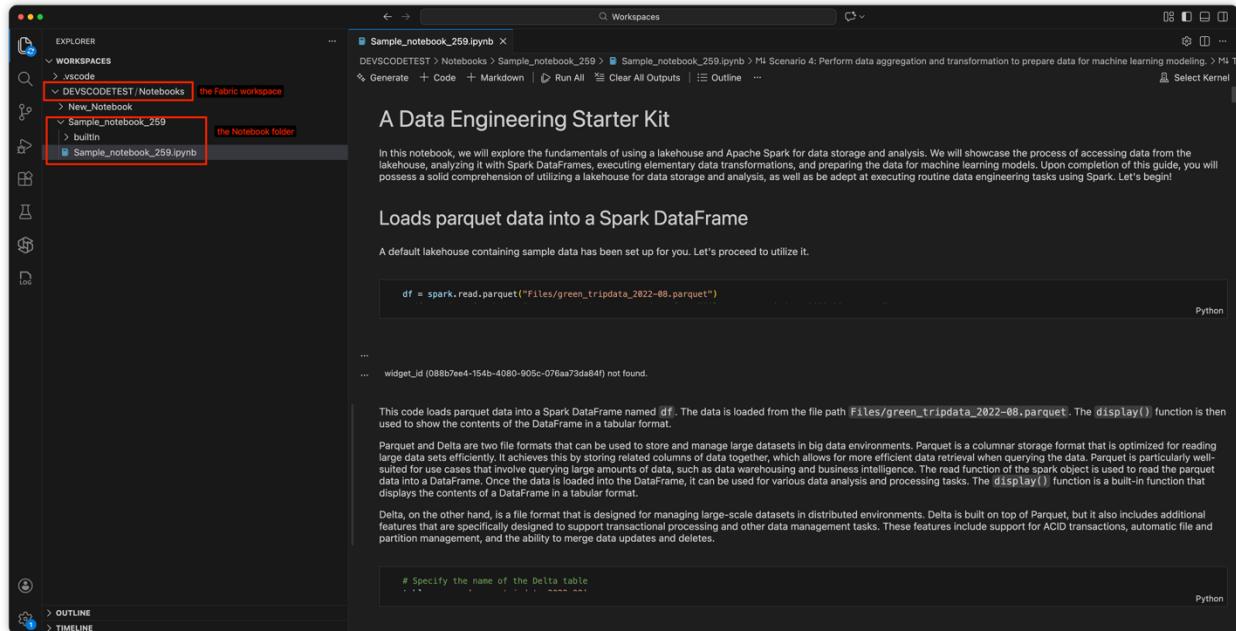
1. Switch to the Fabric workspace inside VS Code



2. Expand the Notebook node, and locate the notebook item to be open, click “Open Notebook Folder”



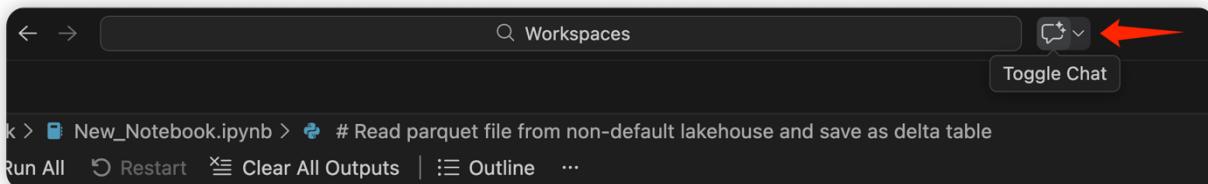
VS Code Explorer view should be activated, and the Notebook content is open. The VS Code Explorer view should list the fabric workspace and all the notebook item within this workspace.



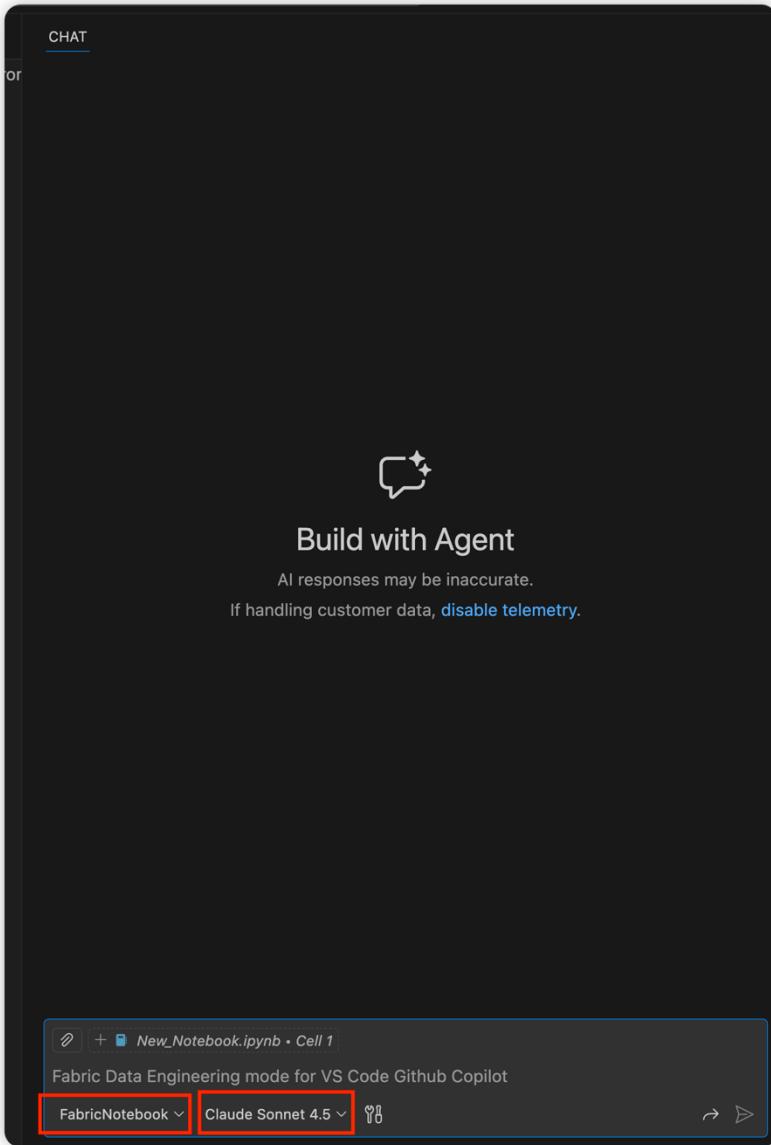
Once you update the notebook content and save it, the change should be synced back to the remote workspace automatically.

Switching Fabric Notebook custom agent

Click the “Toggle Chat” to open the VS Code Chat windows with GitHub Copilot

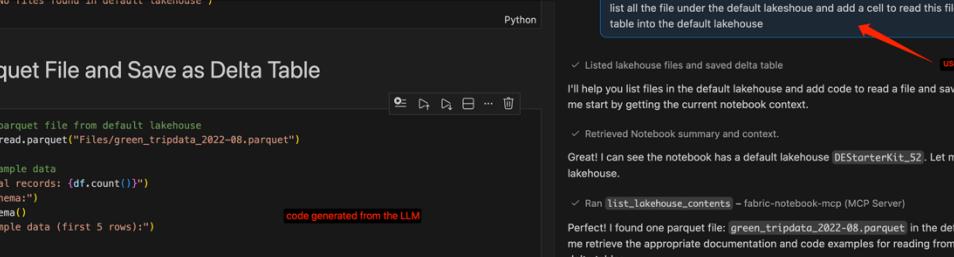


The “FabricNotebook” custom agent should be selected automatically when the notebook is open, and Claude Sonnet 4.5 is selected as the model



With this Agent, you can start writing your prompt to let the LLM generate the notebook code for you. Following is a sample prompt, and the expected result from the LLM

Following is another sample of the prompts and its result from LLM



Sample_notebook_259.ipynb

DEVSODETEST > Notebooks > Sample_notebook_259 > Sample_notebook_259.ipynb > M4 A Data Engineering Starter Kit > M4 Read

Generate + Code + Markdown | Run All | Clear All Outputs | Outline | Select Kernel

Python

Read Parquet File and Save as Delta Table

```
# Read the parquet file from default lakehouse
df = spark.read.parquet("files/green_tripdata_2022-08.parquet")

# Display sample data
print(f"Total records: {df.count()}")
print("nSchema:")
df.printSchema()
print("nSample data (first 5 rows):")
df.show(5)

# Save as Delta table in the default lakehouse
table_name = "green_tripdata"
df.write.mode("overwrite").format("delta").saveAsTable(table_name)

print(f"\nSuccessfully saved data as Delta table: {table_name}")
```

code generated from the LLM

Python

CHAT

list all the file under the default lakeshore and add a cell to read this file then save it as delta table into the default lakehouse

user prompt

- ✓ Listed lakehouse files and saved delta table
- I'll help you list files in the default lakehouse and add code to read a file and save it as a delta table. Let me start by getting the current notebook context.
- ✓ Retrieved Notebook summary and context.
- Great! I can see the notebook has a default lakehouse `DEStarterKit_52`. Let me now list the files in this lakehouse.
- ✓ Ran `list_lakehouse_contents` - fabric-notebook-mcp (MCP Server)
- Perfect! I found one parquet file: `green_tripdata_2022-08.parquet` in the default lakehouse. Now let me retrieve the appropriate documentation and code examples for reading from lakehouse and writing delta tables.
- ✓ Ran `get_fabric_doc` - fabric-notebook-mcp (MCP Server)
- ✓ Ran `query_code_examples` - fabric-notebook-mcp (MCP Server)

Now I'll add a cell to read the parquet file and save it as a delta table. Based on the documentation, since we're working with the default lakehouse, I'll use relative paths.

- ✓ Edit
- ✓ Edit
- ✓ Edit
- ✓ Edit

Perfect! I've added cells to your notebook that will:

Files found in default lakehouse `DEStarterKit_52`:

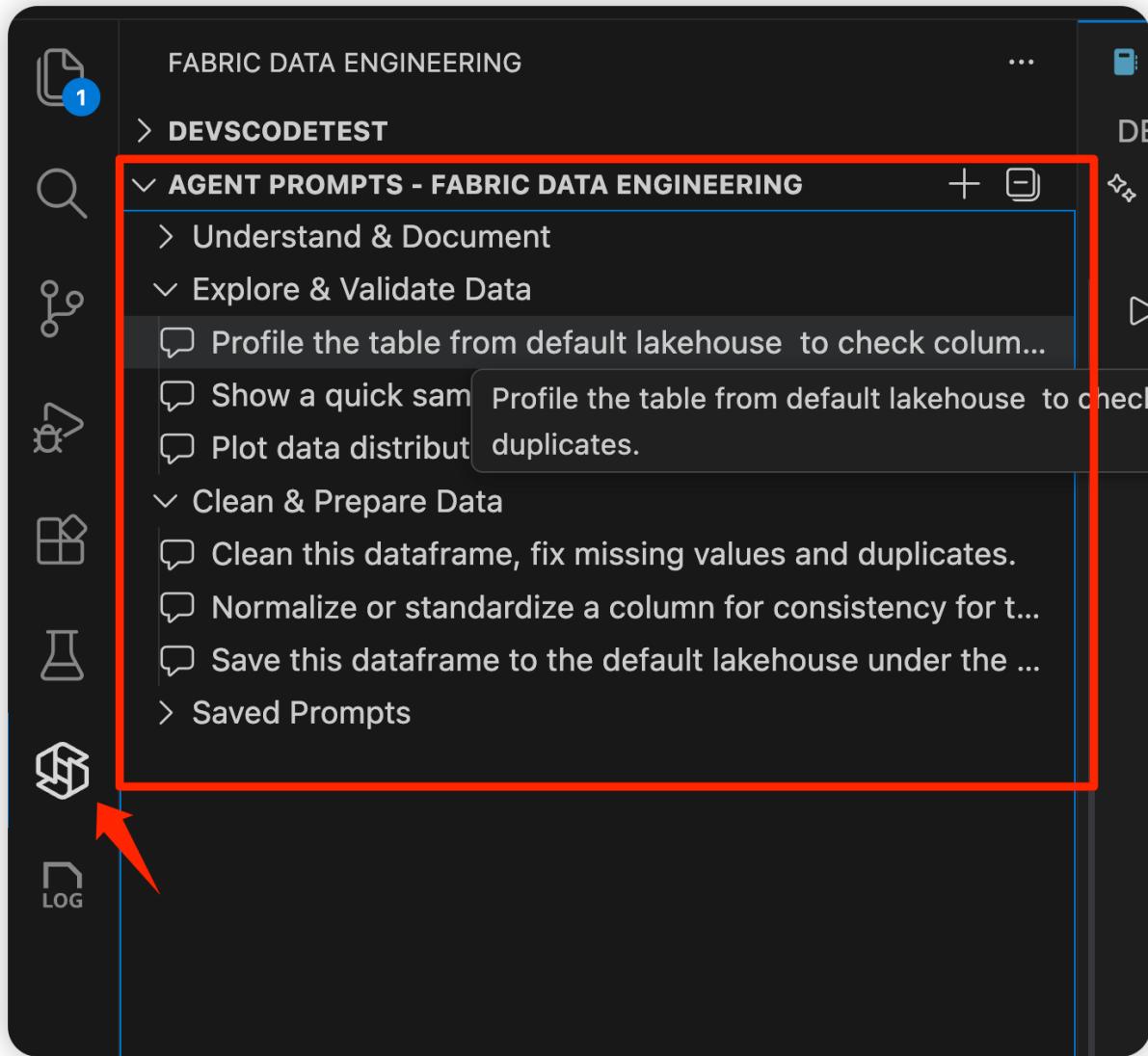
- `green_tripdata_2022-08.parquet` (1.3 MB)

Sample_notebook_259.ipynb - Cell 6

Fabric Data Engineering mode for VS Code Github Copilot

FabricNotebook Claude Sonnet 4.5

The extension provides sample prompts to get you started, you can find these sample prompt in the extension view.



Known Issue:

1. Claude Sonnet 4.5 and GPT-5.2 are the two models supported by this preview. please stick with these two models.