
ELLM

How to Install eLLM Code and Connect It to Your Endpoint

Learn how to install the eLLM Code extension in Visual Studio Code and connect it to your organisation's endpoint. This guide covers installation steps, configuration, and troubleshooting to ensure seamless integration and usage.

AUTHOR

Paul Flanders

PUBLISHED

10 Sep 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from
EssingtonITS

How to Install eLLM Code and Connect It to Your Endpoint

Paul Flanders · 10 Sep 2026

Estimated time: 4 minutes

Difficulty: Beginner

Why you'd use this

eLLM Code is a coding assistant that lives inside Visual Studio Code and talks only to your organisation's own eLLM endpoint. Before you can ask it anything, the extension has to be installed and pointed at that endpoint with a model name and, usually, an API key. Once this is done, every developer on the team gets the same chat panel, with source code and prompts never leaving your environment.

Before you start

Permissions required:

- Permission to install extensions in your VS Code (some managed installs block VSIX files).
- An API key for the eLLM endpoint, if your gateway requires one.

You'll need:

- VS Code 1.85 or newer.
- The `e11m-code-<version>.vsix` file supplied by your platform team.
- The endpoint URL, which ends in `/v1/chat/completions`, and the model identifier your gateway expects (the default is `e11m-code`).

Steps

1. Open a terminal and run `code --install-extension e1lm-code-<version>.vsix` .
Alternatively, open the Extensions view in VS Code, click the ... menu at the top of the view, choose **Install from VSIX...** and select the file.
2. Open any project folder. An **eLLM** item appears in the status bar and a chat icon appears in the editor title bar.
3. Click **eLLM** in the status bar, or run **eLLM: Open Chat** from the Command Palette (`Ctrl+Shift+P` / `Cmd+Shift+P`). The chat panel opens beside your code.
4. If the endpoint is not yet configured, a banner reads "Set your eLLM endpoint and API key to start." Click **Configure**, or click the ⚙ button in the panel header.
5. In the settings panel, fill in the **eLLM endpoint** field with the full chat-completions URL.
6. Fill in the **Model** field with the model identifier your gateway expects.
7. Paste your key into the **API key** field. The key is stored in VS Code's secret storage, not in a settings file, so the field shows blank the next time you open the panel.
8. Click **Save**, then **Close**.
9. Type a short question in the composer, for example "What does this project do?", and press **Enter**. A streamed answer confirms the connection.

What you should see

The setup banner disappears, the composer becomes active, and the status row under the composer shows a timer and token count while a reply streams. Running **eLLM: About (version)** from the Command Palette shows the installed version, the model and the endpoint in one line, which is a quick way to confirm the configuration on a colleague's machine.

Troubleshooting

- **No reply, or an error mentioning 401 or 403:** the key is missing or wrong. Re-enter it via **eLLM: Set API Key**.
- **An error mentioning "not found" or a 404:** the endpoint URL is incomplete. It must include the `/v1/chat/completions` path.
- **Reply arrives but from an unexpected model:** your gateway routes by model name. Check the Model field matches what the gateway publishes.
- **The status bar item is missing:** reload the window (**Developer: Reload Window**) after installing the VSIX.

Related guides

- How to Set or Clear the API Key
- How to Configure a Reasoning Model and Native Tool Calling
- How to Ask Questions About Your Codebase