

---

ELLM

# Coding with ELLM (the code API)

*The eLLM code API offers developers control over coding assistants by centralising access, logging usage, and allowing local model use to keep code secure. It provides dedicated coding modes and special API keys for streamlined and secure operations.*

AUTHOR

Paul Flanders

PUBLISHED

9 Jun 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from EssingtonITS

# Coding with ELLM (the code API)

---

Paul Flanders · 9 Jun 2026

## What this is for

Coding assistants work as agents: they read your code, plan changes, call tools (read a file, run a search), and edit. When these tools talk to a normal AI service, each developer needs a provider key, and the organisation has little visibility of what is being sent or spent. eLLM's code API puts that traffic back under your control: one place to issue and revoke access, a record of who used what, control of which model answers, and the option to use a local coding model so code never leaves the building.

## Main features

- A dedicated coding mode, **ellm-code**, that behaves the way coding agents expect: it does not search your documents, and it passes the agent's tools straight through.
- **Code keys** special API keys that an administrator issues just for coding tools, kept separate from ordinary keys.
- **Choice of coding model** answers come from a model you have marked as a "code" model, whether local or cloud, chosen automatically by cost or pinned to a specific one.
- **Observe-only safeguarding** coding prompts are recorded but never blocked, so a legitimate request that happens to mention words like "kill a process" or "exploit" is not refused.
- **Usage logging** every coding request is recorded, with the model used and the tokens it cost.

# How to set it up

## For the administrator

1. **Mark a coding model.** On the Models page, tag the model you want coders to use with the **code** capability. You can tag more than one (for example a local model and a cloud fallback); eLLM prefers a local model first, then the lowest-cost option. If a model should be preferred when an agent uses tools, also tag it tools.
2. **Issue a code key.** On the API keys page, tick "**Code key**" when you generate the key, and assign it to the right groups. A code key carries a small code badge in the list.
3. **Check the routing.** Under Settings → Test routing, switch the mode selector to Code (ellm-code) to see which model a coding request would use, and why.

## For the developer

1. Point your coding tool at eLLM's API address, ending in /v1, using your OpenAI-compatible settings.
2. Use the code key your administrator gave you as the credential.
3. Set the model field to **ellm-code** (or **ellm-code:** followed by a specific model name to target one).

## Things to know

- **A code key is required.** Only a key marked as a code key can use ellm-code. An ordinary key that tries it is refused. This is deliberate: code mode trades question-blocking for logging, so it is kept to keys an administrator has chosen for it.
- **No document grounding.** Coding mode never searches your company documents — coding agents bring their own context. If you want grounded, cited answers, that is the ordinary chat API, not coding mode.

- **Safeguarding is observe-only here.** Across the rest of eLLM, harmful questions can be blocked. In coding mode they are only recorded, never blocked, because normal code legitimately contains words a safeguarding filter would otherwise flag. The recording still happens, so you keep a trail.
- **Spending is still capped.** If the chosen coding model is a cloud model and your monthly cloud budget is exhausted, the request is refused — the cost guard still applies.
- **Access still follows groups.** A code key's groups work exactly like a person's, so the usual access rules hold.
- **It is stateless.** As with the rest of the external API, eLLM does not remember earlier turns; the coding tool sends the conversation each time.

## Troubleshooting

- **My request is refused with "not authorised for code mode":** the key is an ordinary key. Ask an administrator for a code key, or to tick "Code key" on the key you are using.
- **My coding tool gets document-style answers:** the model field is probably set to ellm-rag or a plain model. Set it to ellm-code.
- **Routing falls back to the default model:** no model is tagged with the code capability. Tag one on the Models page.
- **A cloud coding request is refused:** the monthly cloud budget may be reached. Use a local coding model, or ask an administrator to raise the budget.
- **The tool's tools aren't being used:** confirm the tool is sending its tool definitions and that the model tagged for code supports tool calling (tag it tools so it is preferred).

## Frequently asked questions

### Which coding tools work?

Any OpenAI-compatible agent works today (for example Cline). Codex CLI and Claude Code are designed for and will be supported without changing how keys and models are set up.

### **Can coding stay entirely on our servers?**

Yes. Tag a local coding model with the code capability and eLLM will prefer it, so code never leaves the building.

### **Why doesn't coding mode block harmful prompts like the rest of eLLM?**

Because real code routinely contains words a safeguarding filter would flag, blocking would break ordinary work. Coding requests are still recorded, so you keep oversight without false refusals.

### **How do we see what coding cost?**

Every coding request is in the audit log with the model used and its token usage, and cloud spend counts towards your monthly budget.

## **Summary**

The code API lets your developers' AI coding tools run through eLLM using a dedicated eLLM-code mode and a special code key. You choose which model answers (local or cloud), keep a full usage record, and let safeguarding observe rather than block so coding is centrally governed and logged while staying out of the developer's way.

Need assistance navigating the complexities of eLLM? EssingtonITS offers expert guidance and tailored IT solutions to help you succeed. Visit [EssingtonITS.co.uk](https://EssingtonITS.co.uk) for comprehensive support.