ELLM

How to Run the Benchmark Suite

This article provides a detailed guide on running a benchmark suite to test model configurations, including setup requirements, steps for execution, and troubleshooting tips. It's aimed at advanced users looking to evaluate model capabilities before deployment.

AUTHOR	PUBLISHED	DOCUMENT
Paul Flanders	11 Sep 2026	InfoHub — Knowledge, news, and updates from EssingtonITS

How to Run the Benchmark Suite

Paul Flanders · 11 Sep 2026

Estimated time: 15 to 60 minutes, mostly unattended

Difficulty: Advanced

Why you'd use this

Before rolling a new model or gateway configuration out to a team, you want to know whether it follows the action format, recalls context deeply enough, writes files reliably and can complete a plan-driven build. The benchmark drives a set of scripted scenarios through the whole pipeline against your endpoint, scores each pass or fail, and writes a report with a measured capability profile of the routed model. Note that a full run sends many requests to the gateway and consumes credits accordingly.

Before you start

Permissions required:

- Approval to spend gateway credits on a test run.

You'll need:

- An empty folder open in VS Code. The benchmark creates and edits files freely, and runs commands automatically inside it.
- The endpoint, model and API key configured.
- Node.js on PATH for the scenarios that run tests, and Chromium for the browser scenario.

Steps

1. Open an empty folder.
2. Run **eLLM: Run Benchmark Suite** from the Command Palette.
3. Pick **Quick suite** (capability probes and core tasks, roughly 10 to 20 minutes) or **Full suite** (adds Teaching Mode and an Autopilot Plan and Build, roughly 30 to 60 minutes).
4. Leave the chat panel open. Each scenario is announced with a progress line and its result as it finishes. You can work in another window meanwhile.
5. When the run ends, a note reads "Benchmark complete: N/M passed" with the path to the report.
6. Open `bench-<timestamp>/report.md` in the folder. It lists each scenario, the assertions that failed with details, and the measured profile: protocol fidelity, context-recall depth and decode speed. `summary.json` holds the same data for scripts.
7. To compare two configurations, run the suite again in a fresh empty folder after changing the model or settings, and diff the two reports.

What you should see

Any settings the benchmark changes for a scenario are restored when it finishes. A scenario that overruns its time limit is scored as failed with "finished within time limit" marked false, rather than blocking the run. Each scenario's trace is kept so a failure can be diagnosed from the debug log.

Troubleshooting

- **"open a folder to run the benchmark in":** the command needs a workspace folder, ideally empty.
- **"configure the endpoint first":** set the endpoint in ⚙ before running.

- **The browser scenario fails with "no headless browser":** install Chromium or set `ELLM_CHROMIUM`.
- **You need to stop early:** click **Stop** in the chat. Scenarios that already finished are still written to the report.

Related guides

- How to Configure a Reasoning Model and Native Tool Calling
- How to Tune Generation Settings for Your Model