
ELLM

How to Preview a Web App and Capture Runtime Errors

Learn how to preview a static web app locally to capture runtime errors using a headless browser. This guide explains the steps to set up the preview, troubleshoot common issues, and ensure your web app runs smoothly before deployment.

AUTHOR

Paul Flanders

PUBLISHED

10 Sep 2026

DOCUMENT

InfoHub — Knowledge, news, and updates from
EssingtonITS

How to Preview a Web App and Capture Runtime Errors

Paul Flanders · 10 Sep 2026

Estimated time: 3 minutes

Difficulty: Intermediate

Why you'd use this

A static web app can pass every syntax check and still throw the moment it loads, for example when a script calls a function that another file never defines. The preview serves the workspace on a local port, loads the entry page in a headless browser and collects console errors, so the assistant sees what a user would see. You can trigger it by hand after a change, and Plan Mode runs it automatically before declaring a web build complete.

Before you start

Permissions required:

- Ability to bind a local port on the machine.

You'll need:

- An `index.html` in the workspace root or in `public/`, `src/`, `dist/` or `www/`.
- Chrome or Chromium installed, or `ELLM_CHROMIUM` set to the path of a browser binary.

Steps

1. Press `/` and choose **Preview web app (capture errors)**, or run **eLLM: Preview Web App**.

2. A "Preview: starting" line appears, then the page is loaded headlessly for a few seconds.
3. Read the note that follows. It names the entry file, the local URL, and either "loaded with no runtime errors" or a list of the console errors captured.
4. If errors are listed, ask "Fix the preview errors". The assistant reads the files involved and proposes fixes, then previews again.
5. To look at the page yourself, open the local URL from the note in your normal browser while the preview server is running.

What you should see

The preview server only serves files from the workspace, refuses paths outside it, and shuts down with the session. Each load carries a unique marker so an error report is matched to the exact load that produced it.

Troubleshooting

- **"no HTML entry point (index.html) was found"**: add an `index.html`, or tell the assistant the entry path so it can be previewed directly.
- **"the browser did not run"**: no Chromium was found. Install one or set `ELLM_CHROMIUM`, then reload the window.
- **A page that needs a backend shows fetch errors**: the preview is for static front-ends. Start the backend yourself first, or ask the assistant to run it with a background command.

Related guides

- How to Understand the Verification Gates
- How to Plan and Build a Feature with Plan Mode