

Red Pen for Express

User manual - review notes on every page your Express app serves, in one middleware line.

Version 0.3.1 · August 2026 · by Lincoln Tracy · redpen.tools

What it does

Mount one middleware and every HTML page your app serves gets a review overlay: a red button in the corner, a notes panel, and numbered pins on page elements. Notes save to a file in your project - which means they live in your repo, travel with your branches, and show up in your diffs like any other project artifact.

Zero dependencies. Off automatically in production, and only reachable from the machine it runs on.

Install and mount

```
npm install red-pen-express
```

```
const redPen = require('red-pen-express');  
app.use(redPen()); // mount early - before your routes
```

That's the whole integration. The middleware injects its script into HTML responses; JSON, assets, and streams pass through untouched. Options if you need them:

Option	Default	Meaning
<code>mount</code>	<code>/__redpen</code>	URL prefix for the overlay script, API and board
<code>file</code>	<code>.redpen/notes.json</code>	store location - the log lives beside it
<code>enabled</code>	on unless <code>NODE_ENV=production</code>	master switch
<code>inject</code>	<code>true</code>	auto-inject the overlay into HTML responses
<code>localOnly</code>	<code>true</code>	serve only connections from this machine
<code>author</code>	<code>git config user.name</code>	name notes are signed with

Leaving notes

[Home](#) [About](#)

Red Pen Express

This page is served by a plain Express app. The Red Pen middleware injected the overlay - look for the red button in the bottom-right corner.

A feature block

Pretend this is something you want to leave a note about.

Another section

Notes are scoped per page and saved to `.redpen/notes.json`.

A page served by the demo app: the red button bottom-right, and pin 1 anchored to the feature block.

Click the red button and the panel opens. Type the note, pick a type (Note, Idea, Problem, Question) and a priority, and Add - or Ctrl+Enter. **Pin** lets you click the element the note is about; it becomes a numbered marker on the page, and Locate flashes it later. Notes are scoped per page, and each note carries its own reply thread.

[Home](#) [About](#)

Red Pen Express

This page is served by a plain Express app. The Red Pen middleware injected the overlay - look for the red button in the bottom-right corner.

A feature block

1

Pretend this is something you want to leave a note about.

Another section

Notes are scoped per page and saved to `.redpen/notes.json`.

The Red Pen overlay panel is a red-themed interface for managing notes. It features a header with a pin icon, the text 'Red Pen', and a link to 'All notes'. Below the header are two tabs: 'Open' (with a count of 3) and 'Resolved' (with a count of 1). The main area displays a list of notes, each with a title, priority, date, and actions. The first note is a 'Question' with 'low' priority, dated 'Jul 12 5:10 AM', with the text 'Should we link the API docs from here?'. The second note is a 'Problem' with 'normal' priority, dated 'Jul 12 5:02 AM', with the text 'Spacing collapses under this block on mobile widths' and a sub-note 'Confirmed on a 375px viewport Jul 12 5:05 AM'. The third note is an 'Idea' with 'high' priority, dated 'Jul 12 5:00 AM', with the text 'The hero heading could be bolder - try 800 weight'. Each note has 'Locate', 'Edit', 'Reply', 'Resolve', and 'Delete' actions. At the bottom, there is a text input field 'Leave a note on this page...', a 'Pin' button, a 'Note' dropdown, a 'Normal' dropdown, and an 'Add' button. A red circular button with a pin icon and a count of 3 is located at the bottom right of the page.

The panel: status tabs, typed notes with priorities, a reply thread, and the add form.

Who wrote it

Every note and reply is signed. The name comes from `git config user.name` - the identity your project already has - falling back to your OS username. The chip in the panel header shows the name you are signing with; click it to use a different one, which is remembered in your browser and sent with your notes. There are no accounts and nothing to configure.

Open, In Progress, Resolved

A note is **Open** when you leave it. **Start** moves it to **In Progress** - its card and its pin turn amber, so a page mid-fix does not look untouched - and **Stop** puts it back. **Resolve** closes it and drops its pin; **Reopen** brings it back. The panel and the board both have a tab per state with live counts, and the same three states are what the Red Pen Hub reads when it aggregates this project with your others.

The All-notes board

The panel's **All notes** link opens the board - every note across every page of the app, with status tabs, per-row resolve and delete, and links that jump to the note's page.

Open Resolved All						
Type	Priority	Note	Page	When	Status	
Idea	high	About page needs a contact section	/about	7/12/2026, 5:20:00 AM	Open	Resolve Delete
Question	low	Should we link the API docs from here?	/	7/12/2026, 5:10:00 AM	Open	Resolve Delete
Problem	normal	Spacing collapses under this block on mobile widths Confirmed on a 375px viewport	/	7/12/2026, 5:02:00 AM	Open	Resolve Delete
Idea	high	The hero heading could be bolder - try 800 weight	/	7/12/2026, 5:00:00 AM	Open	Resolve Delete

The board at `/__redpen/repo`.

Where your notes live

`.redpen/notes.jsonl` in your project root: an append-only log, one JSON record per line, created on the first note. **Commit it.** Notes are about the code, so they belong on the branch the code is on - the review you left on a feature branch arrives with the feature and disappears when the branch does.

Committing shared working data usually means merge conflicts, so the middleware writes two files into `.redpen/` when it opens the store, to make sure it does not:

```
.redpen/.gitattributes    notes.jsonl merge=union text eol=lf
.redpen/.gitignore       notes.json  (and the other derived files)
```

`merge=union` tells git to keep both sides of a concurrent change instead of raising a conflict. That is only safe because nothing in the log is ever rewritten in place: every change appends a record, and reading replays the log taking the newest write per field. Two people on two branches - one edits a note's text, the other resolves it - merge with no conflict and both changes land on the right fields.

`.redpen/notes.json` beside it is a derived snapshot, kept for tools that read notes off disk (the Red Pen Hub does). It is regenerated from the log, so it is gitignored rather than committed.

Prefer to keep review notes out of the repo? Gitignore the whole `.redpen/` directory. Everything works the same; the notes just stay on your machine.

If git is already ignoring them, Red Pen says so. Red Pen used to tell you to gitignore `.redpen/`, and a rule left over from that advice buries the log: git will not re-include a path whose parent directory was excluded, so the `.redpen/.gitignore` above cannot rescue it. On startup the store asks git whether the log is ignored and, if it is, prints the offending file, line number and pattern once on your terminal with what to change. It never edits your `.gitignore` - that file is yours, and you may be ignoring the notes on purpose.

Upgrading from 0.2.x: the first run converts an existing `notes.json` automatically. The original is copied to `notes.pre-jsonl-<timestamp>.json` first, and the new log is read back and checked note-for-note before it is accepted - if anything failed to round-trip, the log is discarded and the old file keeps serving. If `notes.json` was committed, run `git rm --cached .redpen/notes.json` once.

Production, and who can reach it

When `NODE_ENV=production` the middleware does nothing at all - no injection, no routes, no file access. Deploy with it mounted and it stays silent; you can also force it off with `enabled: false`.

Outside production it answers only connections from the machine it runs on. A visitor from anywhere else gets your app with no overlay, and the API answers them with a 403 - Red Pen has no login, so anything that can reach it can read and write every note in the project. For a genuinely shared dev box, `localOnly: false` opts out. One caveat: behind a reverse proxy every request arrives from the proxy, which is loopback, so that check cannot help you there - keep Red Pen off in production instead.

If something breaks

Symptom	Likely cause and fix
No red button on any page	<code>NODE_ENV</code> is production, the middleware is mounted after the route that served the page, or the response isn't <code>text/html</code> .
Button on some pages, missing on others	Those responses aren't HTML, or another middleware (compression, custom writes) altered the response before injection. Mount Red Pen first.
No red button, but only for one person	They are not on the machine the app runs on. Pass <code>localOnly: false</code> if that is deliberate.
Notes vanish between branches	<code>.redpen/notes.jsonl</code> differs per branch - that's the committed-notes workflow doing its job. Merge or cherry-pick the file like any other.
A merge conflicted the notes file	<code>.redpen/.gitattributes</code> is missing or was not committed. Restore the <code>notes.jsonl</code> <code>merge=union</code> line, commit it, and merges stop conflicting. Union merge only applies once that file is in the branch you are merging into.
A note you resolved came back open	Someone else's branch resolved or reopened it later; the newest write per field wins. Check the log - every change is a line with a timestamp and a name.

<code>git add -A</code> stages nothing from <code>.redpen/</code>	A rule in a <code>.gitignore</code> higher up the tree is excluding the directory - most likely the <code>.redpen/</code> line Red Pen's own 0.2.x README told you to add. The terminal warning names the file, line and pattern; remove that rule.
---	---

The rest of the family

Red Pen also runs on WordPress, static sites, Blender and Ableton Live, with the Red Pen Hub collecting them onto one board. Each surface is its own port of one shared model, so they do not all have the same features at the same time. The **Surface matrix** at `docs.redpen.tools/matrix.html` is the per-feature answer for which surface has what today.

Red Pen for Express 0.3.1 - built by Lincoln Tracy. Part of the Red Pen family: WordPress, Express, static sites, Blender, Ableton Live, and the Red Pen Hub. `redpen.tools`