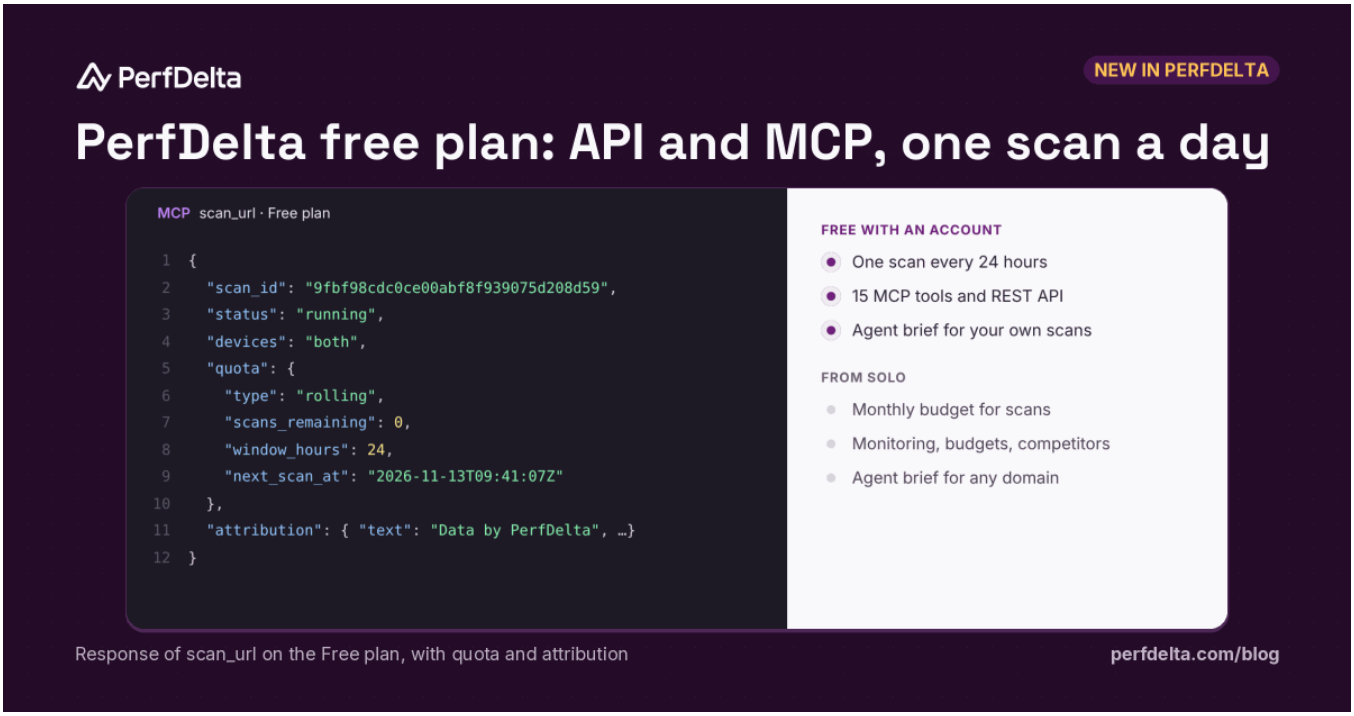
PerfDelta Free Plan: REST API, MCP Tools and Agent Brief for Your Own Pages

Published November 12, 2026 by Christian Schreiber



The PerfDelta free plan now includes the REST API and the MCP server. With a free account you run one scan every 24 hours from Claude Code, Cursor, ChatGPT or a shell script, read the full report and hand the agent brief for that scan to your coding agent. You need no credit card, only a confirmed email address or a Google sign-in. Until now, the MCP server started with the Solo plan and the REST API with the Agency plan. With this release the REST API is also included from Solo.

In short: one scan every 24 hours via REST API or MCP, 15 of 25 MCP tools, 30 requests per minute, the agent brief for your own scans, no credit card.

Developers had to buy a plan before they could find out whether the loop of measuring, letting an agent fix things and measuring again fits their repository. The free plan lets you test that loop on your own site.

Free plan vs. Google's PageSpeed Insights API: fixes instead of raw Lighthouse JSON

Google's PageSpeed Insights API is free with an API key and allows 25,000 requests per day (Google, PageSpeed Insights API documentation, retrieved 12 November 2026). It returns the Lighthouse result of one run and the Chrome UX Report field data of the URL and the origin. If you need many measurements and want to build the evaluation yourself, that API is the better choice.

PerfDelta uses the same Lighthouse data and adds what a coding agent needs to act on it:

- prioritised fixes with stable IDs, affected files and an acceptance criterion per fix,
- the history of your page across scans and what changed between two scans (`get_changes`),
- a feedback loop: the agent reports per fix, and the next brief shows whether a later scan still detects it,
- a ready MCP interface, so the agent needs no parser of its own.

The technical reference for both interfaces is in the PageSpeed API documentation.

Free plan scope: 15 of 25 MCP tools and the agent brief for your own scans

The free plan covers measuring, reading reports and fixing. Anything that measures on a schedule or watches several URLs stays with the paid plans.

	Free plan	**From Solo**
Scans via API and MCP	One scan every 24 hours, smartphone and desktop, one URL	Monthly budget, shared with the web app and monitoring
MCP tools	15 of 25: measure, read reports, fixes, agent brief, notes, read projects	All 25, including budgets, monitoring and competitor management
REST API	Start scans, add notes, read usage, projects, URLs, reports, details, history, fixes and changes	Additionally budgets, competitors and changes to projects
Agent brief	For your own scans, via MCP or as a download in the report	For every domain PerfDelta has measured
Notes per scan	Up to 50 on your own scans	Up to 50
History of a project URL	The last three scans; `older_runs_stored` counts the rest	Complete history
Requests per minute	30	60

Reading tools such as `get_report`, `get_report_details` and `get_fixes` never use the daily scan. The REST API has no endpoint for the agent brief. Every scan, whether started in the web app, via REST or via MCP, appears under "My scans" in your account, grouped by domain.

Free plan setup: create an API key and connect your client

1. Create a free account and confirm your email address, or sign in with Google.
2. Create a key on the API keys page. A free account has one active key.
3. Connect your client. In Claude Code this is a single command:

```
claude mcp add --transport http perfdelta https://perfdelta.com/api/mcp.php \
  --header "Authorization: Bearer YOUR_API_KEY"
```

Without an MCP client, the REST API takes the same key:

```
curl -X POST -H "Authorization: Bearer YOUR_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{"url": "https://example.com/"}' \
  "https://perfdelta.com/api/v1/scans.php"
```

On the free plan the response carries the state of your quota and the attribution (shortened):

```
{
  "scan_id": "ddfec74eaf652c105187ed5c3c0fc3f3",
  "status": "running",
  "devices": "both",
  "quota": {
    "type": "rolling",
    "scans_used": 1,
    "scans_limit": 1,
    "scans_remaining": 0,
    "window_hours": 24,
    "next_scan_at": "2026-11-13T09:12:40Z"
  },
  "attribution": { "text": "Data by PerfDelta", "url": "https://perfdelta.com/?utm_source=api&utm_medium=fre
}
```

Settings for Cursor, VS Code and the connectors in Claude.ai and ChatGPT are in the API documentation. The article on the PageSpeed MCP server describes every tool.

Agent workflow on the free plan: fix today, confirm in the lab tomorrow

With one scan a day, a full round on one page takes two days: scan and fix on the first, rescan on the second.

1. **Scan:** the agent calls `scan_url`, polls `get_scan` until the status is `done` and reads the report with `get_report_details`.
2. **Brief:** `get_agent_brief` returns the fixes as Markdown. Each fix has a stable ID, the affected files and an acceptance criterion.

3. **Fix:** the coding agent works through the brief in your repository. Before deploying, it can check the change locally with Lighthouse CI or the Chrome DevTools MCP server, because PageSpeed Insights only reaches public URLs. It compares audits there, not absolute milliseconds, since hardware and throttling differ from PageSpeed Insights. Then it reports per fix with `add_note`: one note per fix, at most 280 characters, with the status `done`, `skipped`, `needs_approval` or `in_progress`.
4. **Rescan the next day:** once `next_scan_at` has passed, the agent scans the page again. The new brief marks each reported fix as resolved or still detected.

This is an excerpt from a real brief for perfdelta.com:

```
### PD-render-blocking-resources: Unblock the first render

- Affects: LCP
- Evidence: lab test
- Impact: estimated time saving 151 ms
- Approval needed: no

[…]

**Acceptance criterion:** The listed files no longer block first render
(defer/async, media attribute or critical CSS inlined); Lighthouse reports
no render-blocking resources for them.
```

The agent takes the ID from the brief and writes for example `PD-render-blocking-resources done: critical CSS inlined (src/header.php:12)`. IDs that name a resource, a file or a third-party vendor, are longer, so the agent should copy them and not build them itself.

The rescan confirms lab results only: an audit that no longer appears in Lighthouse. A single run per day has no median, and values such as LCP vary noticeably between runs (Lighthouse documentation on score variability). Lighthouse does not measure INP; for INP fixes, Total Blocking Time is only a proxy. Field data from the Chrome UX Report covers a rolling 28-day window: an improvement at the 75th percentile appears gradually, is complete after 28 days and, on pages with little traffic, shows only at origin level. The article website speed optimization with AI coding agents explains how the brief is built and how the feedback works.

Free plan limits: rolling 24-hour quota and shared daily capacity

- **Rolling 24 hours:** the next scan becomes available exactly 24 hours after the last one, not at midnight. Every started or reused scan includes `quota` with `scans_remaining` and `next_scan_at`.
- **Headers for scripts:** REST responses carry `X-RateLimit-Limit`, `X-RateLimit-Remaining`, `X-RateLimit-Reset` and `X-Quota-Scans-Remaining`, and once the quota is used up also `X-Quota-Next-Scan-At`. Scans declined with 429 or 503 add `Retry-After` in seconds.

- **Error codes:** `429 quota_exhausted` before the next scan is due, `402 daily_limit` when the daily scans of the account are used up, `503 free_capacity_exhausted` when the shared capacity is used up and `422 free_tier_single_url` for more than one URL per scan.
- **Same URL within an hour:** if an agent asks for the same URL again within an hour, the API returns the existing scan with `reused: true`, whatever devices you asked for, and does not use your quota. The response names the devices that scan measured. A retry loop therefore cannot use up the daily scan by accident.
- **Daily scans of the account:** every API or MCP scan also counts against the daily scans of your free account, the same ones the web app uses, one per device. If you have used them up in the web app on the same day, the API scan is declined with `daily_limit`.
- **Shared capacity:** all free scans from the web app, REST and MCP share one daily capacity. On a day with unusually high demand, a scan may be declined with `free_capacity_exhausted` until midnight Pacific Time, when Google resets the PageSpeed Insights quota. Reading tools are not affected.
- **Public URLs only:** PageSpeed Insights can only reach public http(s) addresses, not localhost or a staging site behind a login.
- **Attribution:** if you publish results from the free plan, please credit PerfDelta as the source. The `attribution` field in each scan response contains text and link.

With a free key, MCP lists only the 15 free tools. REST endpoints outside the free plan and `scan_url` with a `project_id` answer `plan_required` with `_meta.upgrade`, which names the reason and the plan that unlocks it. The agent can then tell you instead of failing silently.

Solo plan: monthly budget, monitoring and agent briefs for any domain

You reach the limits of the free plan once you check more than one page a day or want PerfDelta to measure without being asked. From Solo you get a monthly scan budget, daily monitoring with alerts, performance budgets that let a build fail on a crossed limit, competitor management via API and MCP, the waterfall in `get_report_details`, the complete history and the agent brief for every domain PerfDelta has measured, including your clients' sites. Scans and notes from the free plan carry over when you upgrade, and older scans that the free plan only counts appear in the history right away. All plans are compared on PerfDelta pricing.

To start, create a free account, add the MCP server and ask your agent: "Scan my homepage, work through the agent brief and report each fix." If you want to see what a report contains first, run the website speed test without an account.