



Every Site in Your Account From One API or MCP Call

The new mySites.guru site summary endpoint returns versions, updates, vulnerabilities, backups and a needs-attention flag for every site in one API or MCP call.

Phil E. Taylor | 24 September 2026

Since we published the [mySites.guru API](#) and the [MCP server that lets Claude manage your sites](#), many hundreds of digital agencies have created clients that read from their mySites.guru accounts. Some feed an internal dashboard, some post a morning report into Slack, and a lot of them are Claude conversations asking which sites need work today.

Most of those integrations are well behaved. A few are not. We give every access token a generous 1,000 requests an hour, and some clients spend it by walking through every site in the account, one request per site, every minute, around the clock, and getting back the same answer each time because nothing has changed since the last lap.

So today we are adding a **site summary endpoint** to the API, and the matching `list_site_summaries` tool to the MCP server. One call returns a complete summary of every site in your account. Where an integration used to make thousands of calls, it now makes one.

Why polling every site every minute tells you nothing new

The data behind the API does not change every minute. mySites.guru collects it on a schedule: each connected site is snapshotted twice a day, audits run daily, weekly or monthly depending on how you have set them up, and backups run on whatever backup schedules you have created. Between those points, asking again returns what you already have.

Take an agency with 200 sites that loops through them every minute. That is 12,000 requests an hour against a 1,000 an hour allowance, so the loop hits the limit about five minutes into each hour, starts collecting `429 Too Many Requests` responses, and spends the next 55 minutes locked out, including for the requests that would have done something useful, such as triggering a backup. Slow the loop to every fifteen minutes and it fits, just, at 800 requests an hour, with almost nothing left for anything else.

Either way it asks about each site somewhere between 96 and 1,440 times a day, for data that a snapshot refreshes twice.

Those requests are not free on our side either. Every one of them is authenticated, checked against your team permissions and served from the same database the dashboard uses, for every mySites.guru customer at once.

Snapshots run twice a day. Poll accordingly.

One summary call every few hours sees every change a per-site loop would, for a fraction of the requests.

What the site summary endpoint returns

`GET /api/v1/sites/summary` returns one row per site, built entirely from data mySites.guru already holds. It makes no live call to any of your sites, so it is quick even on the largest accounts, and it costs one request from your hourly allowance whether you manage five sites or five hundred.

Each row answers the questions agencies were making one request per site to ask:

Identity

The site hash_id, URL, friendly name, platform, your tags, and a manage_url that opens the site in the mySites.guru dashboard.

Versions

The CMS, PHP and database versions, and the mySites.guru connector version, as recorded at the last snapshot.

State

Whether the site is connected, paused or flagged as hacked.

Snapshot

When the site was last snapshotted, how many days ago that was, and a handful of facts from that snapshot: debug mode, caching, user registration, offline mode, super admin count, admins without two-factor authentication, and malicious cron jobs.

Audit

When the last audit ran, its id (ready to pass to the audit endpoint for the full result), and whether you have looked at it yet.

Backups

When the most recent backup started and, separately, when the most recent backup finished.

Updates

How many updates are waiting and whether a core update is available.

Vulnerabilities

How many installed extensions or plugins have known vulnerabilities, and how many known vulnerabilities affect the core version.

SSL

The certificate expiry date and the days remaining, which goes negative once it has lapsed.

Attention

A `needs_attention` flag and the list of reasons behind it.

There are two backup times because the start time is written when a backup begins, so on its own it cannot tell you whether the backup finished. `last_backup_completed` comes from the backup records themselves and is the one to trust when a client asks whether last night's backup worked.

Sites that are not WordPress or Joomla, the ones connected as general websites, return `null` for updates, vulnerable extensions and snapshot facts, because there is no CMS to count them on.

A request and a response

```
GET https://manage.mysites.guru/api/v1/sites/summary?needs_attention=true
Authorization: Bearer eyJ...
```

```
{
  "data": [
    {
      "hash_id": "aB3xY9",
      "url": "https://acme.example.com",
      "friendly_name": "Acme Corp",
      "platform": "Joomla",
      "manage_url": "https://manage.mysites.guru/en/sites/manage/aB3xY9",
      "tags": [{ "slug": "retainer", "name": "Retainer", "color": "#0e1116" },
      "version": "5.3.3",
      "php_version": "8.3.24",
      "db_version": "8.0.43",
      "connector_version": "2026.09.1",
      "is_connected": true,
      "is_paused": false,
      "is_hacked": false,
      "last_snapshot": "2026-09-24T06:12:40+00:00",
```

```

    "snapshot_age_days": 0,
    "last_audit": "2026-09-21T02:03:11+00:00",
    "last_audit_id": "k9Qm2T",
    "audit_unread": true,
    "last_backup_started": "2026-09-24T01:00:02+00:00",
    "last_backup_completed": "2026-09-23T01:04:57+00:00",
    "updates_available": 4,
    "core_update_available": false,
    "vulnerable_extensions": 1,
    "core_vulnerability_count": 0,
    "ssl_expiration": "2026-11-02T23:59:59+00:00",
    "ssl_days_remaining": 39,
    "needs_attention": true,
    "attention_reasons": ["updates_available", "vulnerable_extensions"],
    "snapshot": {
      "id": "Zp7Lr4",
      "debug_enabled": false,
      "cache_enabled": true,
      "user_registration_enabled": false,
      "offline_mode": false,
      "super_admin_count": 2,
      "non_2fa_admins": 1,
      "malicious_cron_jobs": 0
    }
  },
  "meta": {
    "total": 1,
    "scope_total": 87,
    "generated_at": "2026-09-24T09:30:00+00:00",
    "stale_snapshot_days": 2,
    "counts": {
      "hacked": 0,
      "updates_available": 1,
      "core_update_available": 0,
      "vulnerable_extensions": 1,
      "core_vulnerabilities": 0,
      "disconnected": 0,
      "stale_snapshot": 0,

```

```

    "paused": 3,
    "needs_attention": 1
  },
  "summary": "1 of 87 sites need attention: 1 with updates, 1 with vulne
}
}

```

Look at the **meta** block before the rows. **counts** totals every reason across the whole account, and **summary** puts it in a sentence you can post straight into a Slack channel or a morning email. The acme row also shows why the two backup times are separate: last night's backup started but has not finished, so the last completed one is from the night before.

How do I narrow the summary down?

Every parameter is optional, and with none of them the summary returns every site in your account, paused ones included. They work the same way on the REST endpoint and on the MCP tool.

needs_attention

Set to true to return only the sites that need attention. The meta counts still cover the whole account, so you still get "4 of 120 sites need attention".

platform

Joomla, Wordpress or Generic.

tag

One of your tag slugs, for example a client or a retainer tier.

hash_ids

A comma-separated list of up to 100 site hash_ids, for when you only care about a handful of sites.

include_paused

Defaults to true. Set it to false to leave paused sites out altogether.

sort

name (the default), last_snapshot or last_backup_completed (oldest first, with never-run sites at the top), or updates (most first).

sort=last_backup_completed puts the sites with the oldest completed backup, and any that have never completed one, at the top. That is the list to read before a client asks about their backups.

What counts as needing attention?

A site needs attention when it is not paused and at least one of seven things is true. Each row lists the reasons that applied in **attention_reasons**, so you can route them differently: a hacked site pages someone, four pending updates can wait for the Tuesday maintenance window.

hacked

mySites.guru has flagged the site as hacked.

updates_available

At least one extension, plugin or theme update is waiting.

core_update_available

A newer WordPress or Joomla release is available.

vulnerable_extensions	An installed extension or plugin has a known vulnerability.
core_vulnerabilities	The installed WordPress or Joomla version has known vulnerabilities.
disconnected	mySites.guru cannot currently reach the site through its connector.
stale_snapshot	The site has never been snapshotted, or its last snapshot is more than two days old.

Paused sites never count as needing attention. You paused them on purpose, usually because the client has left or the site is mid-rebuild, and a report that shouts about them every morning trains everyone to ignore the report. They still appear in the list unless you pass `include_paused=false`, and `meta.counts.paused` tells you how many there are.

Using the summary from Claude and other MCP clients

The MCP server exposes the same thing as the `list_site_summaries` tool, so any AI client connected to your mySites.guru account can call it. Nothing needs setting up: the next time your client lists the server's tools, it is there.

AI clients gain the most from it. Ask Claude "which of my sites need attention?" with only `list_sites` and `get_site` available, and it does what the looping scripts do: fetch the list, then open each site in turn. On a large account that is hundreds of tool calls, a long wait, a rate limit that runs out halfway through, and a conversation stuffed with detail it did not need. With `list_site_summaries` it is one call.

The tool's own description tells the client to start with `needs_attention=true` on accounts with a lot of sites and then narrow down by platform, tag or specific sites. Questions like these now take a single call:

- "Which of my sites need attention, and why?"
- "Which client sites tagged retainer have not completed a backup this week?"
- "List every Joomla site with a vulnerable extension."
- "Which sites have an SSL certificate expiring in the next 30 days?"

From there the client can open the full audit for any site that looks wrong, or trigger the backups and updates it finds are missing, and it still has almost all of its hourly allowance left to do it.

A polling pattern that fits inside the rate limit

If you are running one of the loops described above, here is the replacement. It uses a fraction of the allowance and tells you more.

1. Call `GET /api/v1/sites/summary?needs_attention=true` on a schedule. Every two to four hours is plenty, given snapshots run twice a day.
2. Read `meta.summary` and `meta.counts` for the headline, and post it wherever your team looks. It is the API version of the morning routine we recommend for checking sites.
3. For each row that needs attention, act on the reason: open the audit with `last_audit_id`, send a person to `manage_url`, or trigger the update or backup through the API.
4. Only fetch full site detail for the handful of sites you are acting on.

In a shell, the whole morning report is two commands:

```
curl -s -H "Authorization: Bearer $TOKEN" \
  "https://manage.mysites.guru/api/v1/sites/summary?needs_attention=true&
```

```
| jq -r '.meta.summary, (.data[] | "\(.friendly_name): \(.attention_reas
```

That is one request, where the same report built from per-site calls would cost one request per site. Each response also has `generated_at`, so if you cache the result in your own dashboard you can show your team how old the numbers are.

The rate limit has not changed

Each access token still gets 1,000 requests an hour, and a summary call costs one of them. Go over and the API returns `429 Too Many Requests` with a `Retry-After` header telling you how many seconds to wait. Well-behaved clients read that header and back off rather than retrying straight away.

What the site summary leaves out

The summary only returns what mySites.guru already has on file, and that shapes what is not in it.

- Live data. Every value is as fresh as the last snapshot, audit or backup, and no less. If you need to know what a site looks like this second, [trigger a snapshot](#) and read the result when it finishes.
- Uptime status. Whether a site is up right now comes from [uptime monitoring](#), which lives outside the data the summary reads, so it would need a live lookup per site, the cost this endpoint exists to avoid.
- Paging. You always get the full filtered set, with `meta.total` telling you how many rows came back. The filters are how you keep a response small.
- Any change to `list_sites`. `GET /api/v1/sites` and the `list_sites` tool behave as they did yesterday, so integrations you have already built keep working.

The summary follows the same permissions as everything else in the API. A token only sees the sites its user can see in the dashboard, and a team member restricted to

certain sites gets a summary of those sites and nothing else, as described in our post on [managing sites with your whole team](#).

The API is still in beta

We reserve the right to make changes to the API while it is still in beta, and we may increase or decrease API rate limits as we see different traffic patterns. Any client that is constantly maxed out or abusing its permissions will be terminated instantly.

The dashboard already knows all of this

Every field in the summary is something mySites.guru collects for you anyway, unattended, on each connected site, and the API returns it in one piece. If you are weighing up building your own monitoring, the loop that asks for the data is the easy part. Collecting it in the first place is where the work is.

That work is all part of the subscription. The twice-daily snapshots record every version, setting and extension. The [vulnerable extension list](#) turns those versions into a warning the morning a flaw is published. Scheduled audits scan for [hacked files and backdoors](#), [real-time alerting](#) tells you when a file changes or an unfamiliar admin logs in, and [scheduled backups](#) run whether anyone remembers them or not. The site summary is the one call that reads all of it back.

If you already have an API client, the endpoint is live on your account today. If you do not have an account yet, [run a free audit](#) on one of your sites, or [see the plans](#) and connect all of them.

Further reading

- [How to access your mySites.guru account using our secure API](#) - OAuth2 with PKCE, scopes and the rest of the endpoints

- Manage every site from Claude Desktop or other AI tools - connecting the MCP server that exposes `list_site_summaries`
- Snapshot vs audit: what's the difference? - why the data behind the summary refreshes when it does
- RFC 6585, section 4 - the standard that defines HTTP 429 Too Many Requests
- Retry-After on MDN - how a client should read the wait time after a 429
- MCP specification: tools - how MCP servers publish tools such as `list_site_summaries` to AI clients

Frequently Asked Questions

What is the mySites.guru site summary endpoint?

It is a single mySites.guru API request, GET /api/v1/sites/summary, and a matching MCP tool, list_site_summaries, that return every site in your account in one response. Each row has the CMS, PHP and database versions, the last snapshot, audit and backup times, cached update and vulnerability counts, hacked, connected and paused state, SSL expiry, and a needs_attention flag with the reasons behind it.

Does the summary endpoint contact my sites?

No. The mySites.guru site summary is built entirely from data mySites.guru already holds from its twice-daily snapshots, your audits and your backups. It makes no live call to any site, which is why it is fast and why it is cheap enough to return hundreds of sites in one response.

Does a summary call count against the mySites.guru API rate limit?

Yes, as one request. The mySites.guru API allows 1,000 requests an hour per access token, and a summary call costs one of them whether your account has five sites or five hundred. Reading the same data one site at a time costs one request per site.

How often should I call the site summary endpoint?

Every few hours is plenty. mySites.guru refreshes each site's snapshot twice a day, and audits and backups run on the schedules you set, so calling the summary every minute returns the same answer over and over. Call it on a schedule, or when you are about to act, and use the response's generated_at time to see how fresh it is.

What does needs_attention mean in the site summary?

A site needs attention in the mySites.guru site summary when it is not paused and at least one of these is true: it is flagged as hacked, it has updates available, it has a core update available, it runs a vulnerable extension or plugin, its core version has known vulnerabilities, it is disconnected, or its last snapshot is missing or stale. Each row lists which of those reasons applied.

Can I use the site summary from Claude or another AI client?

Yes. The mySites.guru MCP server exposes it as the list_site_summaries tool, so any MCP client connected to your account can call it. Ask something like which of my sites need

attention, and the client gets the whole picture in one call instead of walking your site list one site at a time.

Has the existing list sites endpoint changed?

No. GET /api/v1/sites and the list_sites MCP tool work exactly as before. The site summary is a separate endpoint for the job of checking on every site at once, and it returns more per site than list_sites does.

AI MODIFIED

Written and edited by a human, with AI assistance. [Our approach to AI](#)

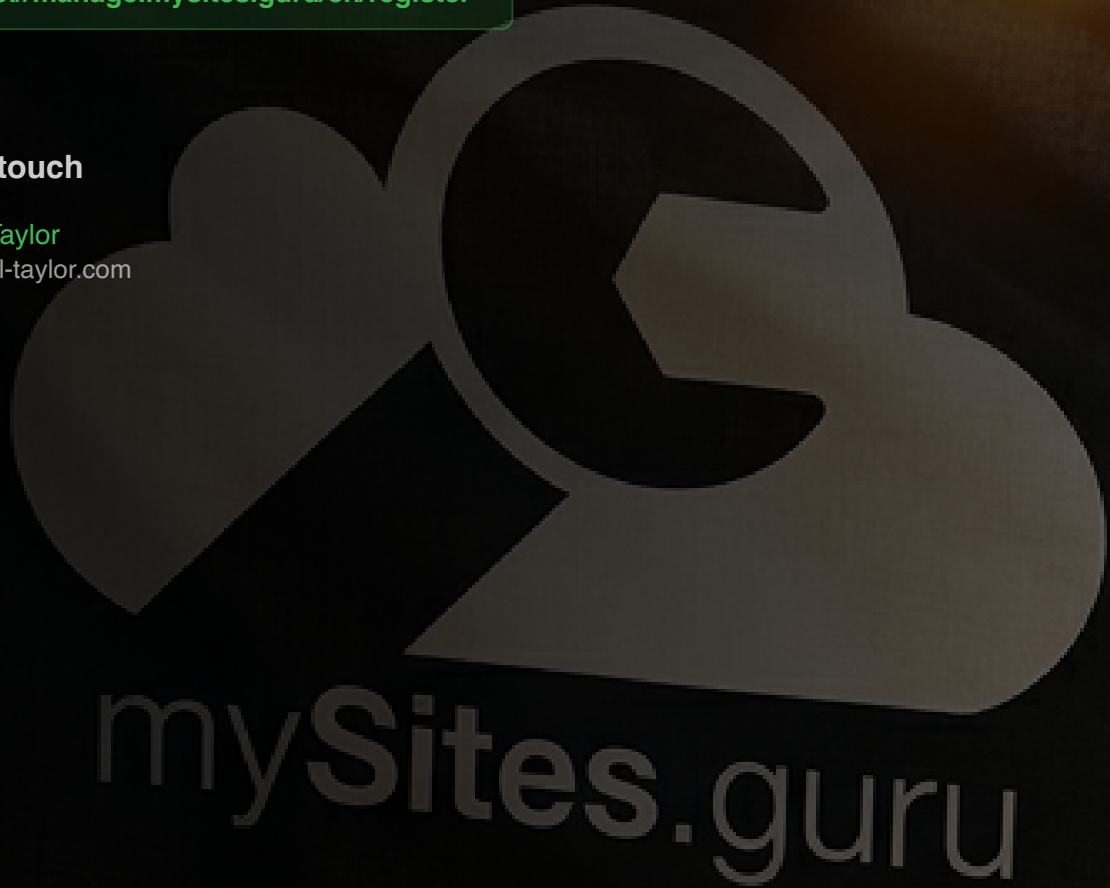
Get Your Free Site Audit

See how your WordPress and Joomla sites measure up.
No credit card required.

<https://manage.mysites.guru/en/register>

Get in touch

Phil E. Taylor
phil@phil-taylor.com



© 2026 Blue Flame Digital Solutions Limited. All rights reserved.

mysites.guru

