MSP360
MANAGE. SUPPORT. PROTECT.

GUIDE

MSP360 RMM API Integration Scenarios Guide

1-866-4-MSP360
1-415-301-7773
sales@msp360.com
6425 Living Place Suite 200
Pittsburgh, PA 15206
msp360.com

This guide is a technical reference for administrators who build on the MSP360 RMM public API. It covers the ten integration scenarios MSPs request most often, ordered by customer demand, and maps each one to the exact API calls behind it. Where the MSP360 RMM API alone is not enough, the scenario says so and points to the right tool.

IN THIS GUIDE, RANKED BY CUSTOMER DEMAND

Before your first API call

One thing to sort out before any of the scenarios below: who can actually generate a token, and where.

Access requirements

- The MSP360 RMM API is available on provider accounts with an RMM license. API tokens are created and managed at the provider level. Each token is issued for a specific administrator who holds a valid RMM license; administrators cannot generate or manage tokens themselves
- RMM Community Edition has no API access. To test the RMM API on Community Edition, request a trial of MSP360 RMM.

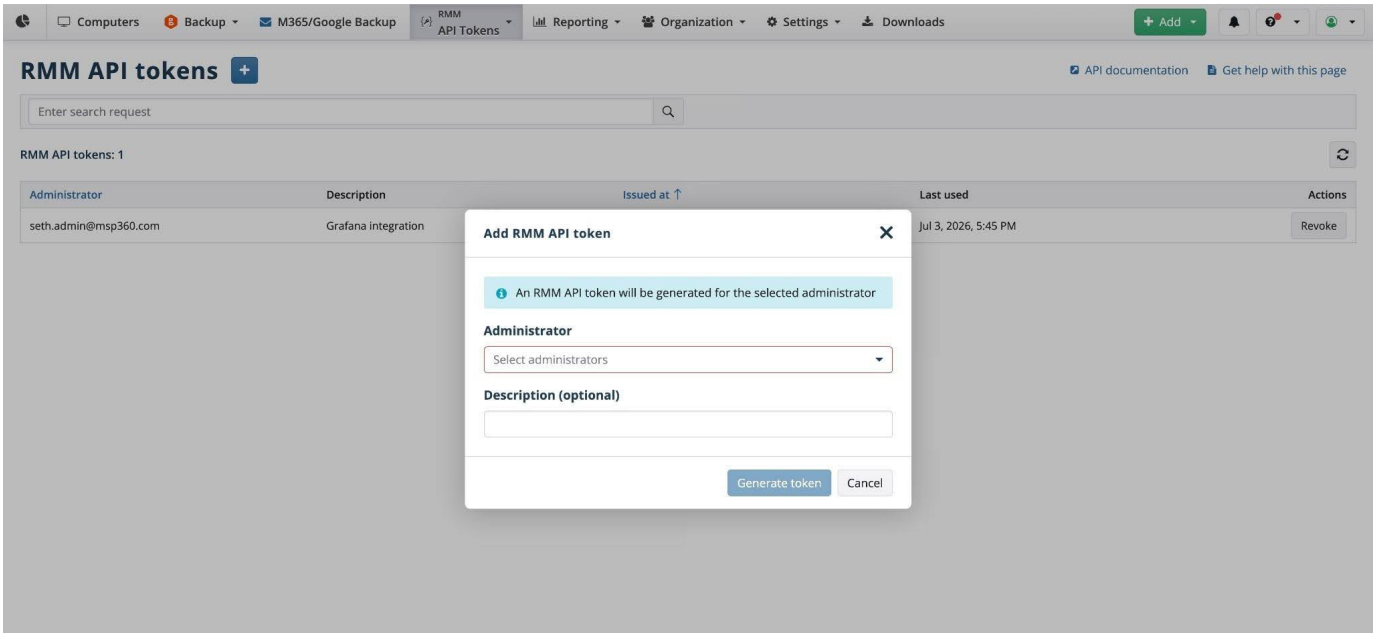
How to generate a token

1. Go to **RMM > API tokens**, or **Settings > General > RMM API tokens**.

2. Click **Add RMM API token** and select the administrator the token will be issued for. The token carries that administrator's access, so the administrator must hold a valid RMM license.

3. Generate the token and store it securely. Every call in this guide authenticates with it.

If you don't see this option at all, that's usually the license-assignment requirement above, not a bug — check the account type before troubleshooting further.
Full endpoint reference: https://api.rmm.mspbackups.com/swagger/index.html



Fewer calls, fresher data: 3 important things to know

The structural facts that shape how every scenario should actually be built:

- Every stat type has both a fleet-wide endpoint (all computers, paginated) and a per-device endpoint (one hid). Most scenarios below should be built on the fleet-wide version — looping over individual devices is no longer necessary and wastes calls.
- Data refreshes on two different clocks: alert data every 10 minutes, everything else every 60 minutes. Polling faster than that returns no new data — it just adds load onto MSP360's infrastructure. Each scenario below states its recommended interval.
- Update-status endpoints cover system/OS patches only. Third-party software patch status (Winget on Windows, Homebrew on macOS) lives in the software inventory endpoint instead — covered in scenario 2.

Endpoint pattern, at a glance

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/{type}/latest	Returns {type} data for every computer in the account, paginated. This is the default choice for dashboards, sync jobs, and anything fleet-wide.
GET [SINGLE DEVICE] /api/v1/computers/{hid}/stat/{type}/latest	Returns {type} data for one computer only. Reach for this when drilling into a specific machine flagged by the fleet-wide view — not as your primary polling pattern.

Refresh cadence, at a glance

ENDPOINT	WHAT IT GIVES YOU
Alert data (the "alert" attribute in summary/latest)	Refreshes every 10 minutes. Poll at this cadence for anything alert-driven — faster polling won't surface anything new.
Everything else (CPU, memory, disk, antivirus, updates, hardware, software…)	Refreshes every 60 minutes. Polling more often than hourly returns the same data and adds unnecessary load.

Note: the alert attribute lives inside the same summary/latest payload as the hourly fields. If a scenario needs fast alert visibility, it's fine to poll the whole summary endpoint every 10 minutes — the non-alert fields simply won't have changed between most of those calls, which is harmless.

1. Asset and inventory sync (CMDB and PSA)

This scenario helps syncing asset data or configuration items to CMDB and PSA. These external tools turned out to need the exact same data, just written to different places. Build the feed once and fan it out to both.

Endpoints to use

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/host/latest	Core system identity for every computer — OS, manufacturer, model, serial number, IP/MAC address, time zone, and location. Paged.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/hardware/latest	Every installed hardware device, fleet-wide. Paged.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/software/latest	Full installed software list for every computer, including names, architecture, and version. Paged.

For reconciling a single record

ENDPOINT	WHAT IT GIVES YOU
GET [SINGLE DEVICE] /api/v1/computers/{hid}/stat/host|hardware|software/latest	Same fields, one machine — use this only when chasing down a specific mismatch, not as your regular sync mechanism.

How it comes together:

- Page through all three fleet-wide endpoints on one scheduled job.
- Normalize each record and fan it out to destinations: your CMDB or asset tool or your PSA's configuration-item / asset object.
- Use the hid as the join key across every downstream system — one identity, multiple consumers.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes — matches the underlying refresh cadence for hardware, software, and host data.**

2. Patch and Windows update status monitoring

The update endpoint gives you real patch data — with one important gap. It covers system/OS updates only. Third-party packages managed through Winget (Windows) or Homebrew (macOS) don't appear here at all, and neither does the summary endpoint's update flag. You need a second call to see the full patching picture.

System updates

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/update/latest	System/OS update status for every computer, paged. Does not include Winget or Homebrew packages.
GET [SINGLE DEVICE] /api/v1/computers/{hid}/stat/update/latest	Same data, single machine — for drilling into one host's update history.

Third-party package updates (Winget / Homebrew)

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/software/latest	Check the availableVersion attribute per package — if it's populated, an update is available. This is the only place Winget and Homebrew package updates show up.

How it comes together:

- Poll the bulk update endpoint for OS-level patch compliance across the fleet.
- Separately poll the bulk software endpoint and flag any package where availableVersion is populated — that's your third-party patch backlog.
- Report the two compliance numbers side by side. A single "% patched" figure built only from the update endpoint will look healthier than it actually is.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes for both endpoints — matches the underlying refresh cadence.**

3. Live dashboard (Grafana)

This is the integration customers describe with the most enthusiasm. The fleet-wide summary endpoint hands you a health snapshot for every computer in a handful of paginated calls — no need to loop through hid values one at a time.

Headline endpoint

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/summary/latest	CPU, memory, disk usage, antivirus status, system update status, SMART status, and the alert attribute for every computer, paged.

Drill-down, per machine

ENDPOINT	WHAT IT GIVES YOU
GET [SINGLE DEVICE] /api/v1/computers/{hid}/stat/runtime/latest	Process tree with live CPU and memory — for a machine flagged as an outlier.
GET [SINGLE DEVICE] /api/v1/computers/{hid}/stat/eventtotal/latest	Event log record count from the latest agent pull — a quick noise/activity indicator.

Remember: the update status inside summary/latest is system-only. If your dashboard shows a patch-compliance tile, pull it from scenario 2's two-endpoint check, not from summary alone — otherwise third-party patch gaps (Winget/Homebrew) won't show up.

How it comes together:

- Page through the fleet-wide summary endpoint on each poll — this single paginated call sequence covers the entire device fleet.
- Land results in InfluxDB or TimescaleDB and build the color-coded status grid and outlier panels from there.
- Drill into runtime and eventtotal per hid only for machines the dashboard flags as needing attention.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes for the general health tiles. If the same dashboard also needs fast alert visibility, poll summary every 10 minutes instead — the other fields just won't change between most of those calls, which is fine.**

4. BI / data warehouse export (PowerBI)

There's no one-click export here — you're assembling the pipeline yourself. The fleet-wide endpoints make that straightforward: they paginate cleanly into fact and dimension tables without any per-device looping.

Endpoints to feed your warehouse

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/summary/latest	Your fact table — CPU, memory, disk, antivirus, system update status, and alerts, refreshed hourly, paged.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/host/latest	Your dimension table — stable identity data that changes rarely.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/software/latest	Include this for full patch-compliance reporting — track availableVersion over time to report third-party patch backlog, which summary/latest can't show.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/hddsmart/latest	SMART health history, paged — useful for hardware-failure trend reports.

How it comes together:

- Land raw paginated responses in a staging table (a simple Bronze layer) exactly as the API returns them.
- Transform them into fact and dimension tables keyed on hid for a proper star schema.
- Publish to Power BI through a scheduled dataset refresh (Gateway), or push rows with the Power BI REST API for near-real-time reports.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes — a nightly full pull plus hourly incremental refresh matches the data's own cadence without over-polling.**

5. Auto-create ticket in PSA upon RMM alert

There's still no webhook or push-based alerts feed — this remains a polling pattern. But it's a well-defined one: the alert attribute inside the fleet-wide summary endpoint refreshes every 10 minutes, which gives you a clean, efficient polling target instead of guessing at an interval.

Endpoint to use

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/summary/latest	Check the alert attribute per computer, fleet-wide, in one paginated pass — no need to loop through hid values individually.

How it comes together:

- Poll the fleet-wide summary endpoint every 10 minutes and keep a last-seen alert set per computer.
- On any new alert, call your PSA's ticket-creation endpoint and attach the alert detail plus a host/latest snapshot for context.
- Track which alerts already have an open ticket to avoid duplicate tickets on a flapping condition.

RECOMMENDED POLLING INTERVAL: **Every 10 minutes — matches the alert refresh cadence exactly. Polling faster adds load without surfacing anything new; polling slower delays ticket creation.**

6. Security Operations Center live view

A security-filtered version of the dashboard in scenario 3 — same fleet-wide endpoints, narrower focus. If the SOC wants a dedicated screen instead of a shared general-health board, this is a light lift of scenario 3.

Endpoints to use

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/antivirus/latest	Live antivirus status for every computer, paged — protected, out of date, or definitions stale.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/summary/latest	Check the alert attribute here alongside antivirus status for a combined security snapshot.

How it comes together:

- Reuse scenario 3's polling job — just filter the Grafana panel to antivirus and alert fields.
- Color-code green / amber / red per machine, filterable by client, for a fast SOC scan.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes for antivirus status; every 10 minutes if this view is also the SOC's primary alert feed.**

7. Native report publishing for clients

There's still no dedicated "reports" endpoint in the public API — no report list, no report-run, no export-to-PDF call. A "native report" here would really be you re-assembling the fleet-wide stat endpoints into your own layout — the same job as scenario 4, styled for a client instead of a warehouse.

Endpoints to use

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/summary/latest	The closest thing to a pre-built report — a rolled-up snapshot, not an exportable document.

Before investing engineering time

- Check whether CSV data export from the MSP360 web console can help here.
- If not, treat this as scenario 4's output styled into a client-facing PDF or portal page — not a new integration.

> Recommendation: don't build a separate "report engine." Point this need at the same pipeline you're already building for scenario 4, and add a client-facing export layer on top.

8. IT Ops Center TV wallboard

Still not a new integration — it's a display choice on top of the dashboard already built in scenario 3.

How it comes together:

- Take the Grafana dashboard from scenario 3 and open it in Grafana's kiosk mode on the Ops Center TV.
- Simplify the panel set for a TV — large status tiles and a client-filtered alert feed read better from across a room than dense tables.

- Set panels to auto-refresh and auto-cycle between clients if one screen needs to cover more accounts than fits at once.

RECOMMENDED POLLING INTERVAL: **Matches scenario 3 — no separate polling job.**

9. SLA / QBR client-facing dashboard

The core of an SLA story is ticket response and resolution time, and that still doesn't exist anywhere in this RMM API — it lives in your PSA. The RMM side supplies the uptime, patch, and security columns; it can't supply the SLA columns on its own.

What the RMM API can supply

ENDPOINT	WHAT IT GIVES YOU
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/host/latest	Uptime and identity context, fleet-wide.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/update/latest	System patch compliance for the QBR patching section — pair with software/latest for third-party coverage.
GET [FLEET-WIDE, PAGED] /api/v1/computers/stat/antivirus/latest	Security status for the QBR security section.

Before investing engineering time

- Pull ticket volume, response time, and resolution time from your PSA's own API — not from MSP360 RMM.
- Join the two data sets by client ID on a shared per-client dashboard.

> Recommendation: plan this as a two-source build from day one — RMM API plus PSA API. Treating it as an RMM-only integration will leave the most important column on the dashboard empty.

RECOMMENDED POLLING INTERVAL: **Every 60 minutes for the RMM-side data; match your PSA's own recommended cadence for ticket data.**

10. Backup job status monitoring (MSP360 Backup)

This is the RMM API — backup job status lives in MSP360's separate Backup product, which has its own management API. Nothing in the RMM endpoint set (antivirus, hardware, software, updates, and system stats) covers backup jobs, schedules, or success/failure state.

Before investing engineering time

1. Check MSP360's documentation for a Backup management API, separate from the RMM spec - see https://api.mspbackups.com/Help.

Before you start polling

With fleet-wide, paginated endpoints now the default pattern, call volume at 300–500 endpoints is dramatically lower than looping per device — a handful of paginated requests per stat type covers the whole fleet. Two rules keep it that way:

- Never poll faster than the data actually refreshes: 10 minutes for alert data, 60 minutes for everything else. Faster polling adds load to MSP360's infrastructure without returning anything new.
- Reach for the per-hid endpoints only when drilling into a specific machine — they're a diagnostic tool, not your primary sync mechanism.
- Build scenarios 1 and 3 first — nearly every other scenario in this guide reuses their endpoints or their polling job.

About MSP360

Established in 2011 by a group of IT professionals, MSP360™ provides simple and reliable cutting-edge Backup and IT management solutions for MSPs and IT departments worldwide.

MSP360™ platform combines the number one easy-to-use backup solution to deliver best-in-class data protection, secure remote access software to provide support to customers or team members, and painless RMM to handle all aspects of IT infrastructures, all under a single pane of glass.