

AI Access via MCP

Point any AI assistant at your Stratum data — read-only, tenant-scoped, self-hosted.

What it is

Stratum ships a **Model Context Protocol (MCP) server** — the open standard for connecting AI assistants to live systems. It lets a client company or your own PEO staff point an AI client (Claude Desktop, an internal copilot) at their Stratum data and ask questions in plain language: *“how many employees does Acme have, when’s their next payday, who’s out this week?”* It complements the data connectors (QuickBooks/Xero push payroll data *out*; MCP lets an AI read *in*) — a different, additive integration surface.

Security — the same guardrails as the UI

- **No shared credential.** Each connection presents its own revocable personal access token (PAT); the server never holds a master key.
- **Read-only by design.** Tokens are read-scoped and the gateway blocks every write (POST/PUT/DELETE) — a read token is a read-only *credential*, not just a read-only tool.
- **Tightly scoped.** Every request flows through the same API gateway as the portals — tenant isolation and role limits all apply, enforced closed-by-default. The AI sees exactly what the token’s owner can see, and nothing in any other client’s data.
- **Self-hosted.** The MCP endpoint runs on your own infrastructure alongside the platform; token material never leaves your environment.

Two modes, one endpoint

Client	A client’s admin or employee	Their own company (an employee sees only their own pay, PTO, benefits).
Operator	Your PEO staff	Across the whole book — list clients, then drill into any one — still limited to that operator’s console permissions.

What it can answer today (read-only)

- Employee roster and individual profiles · pay periods, next/open run, pay stubs
- Pending time-off approvals and who’s out · benefit enrollment windows
- Open support/HR cases · workers'-comp summary (experience mod, claims, YTD premium)
- **QuickBooks / accounting sync** — connection status, the GL-account → QuickBooks-account mapping, and the GL journal (debits/credits) that posts to QuickBooks or Xero for a given payroll run.
- **Any connected integration** — one call reports every integration (BambooHR, Greenhouse, Lever, Slack, Deputy, Clockify, tax engines, ...) with its connection status, last sync, and any errors.

Write actions (e.g. “request time off”) are a deliberate later step behind human confirmation — not v1.

Connecting

- Mint a read token in the portal — **Integrations** → **API & MCP access** (client) or **API & MCP Access** (PEO staff). Shown once, revocable anytime.
- Point the MCP client at the hosted endpoint with that token as a bearer header (or run it locally against the gateway for a single user). Standard MCP — works with any compliant client.