



≡ Experiments > Cron Add Hardening

Experiments

Cron Add Hardening

Context

Recent gateway logs show repeated `cron.add` failures with invalid parameters (missing `sessionTarget` , `wakeMode` , `payload` , and malformed `schedule`). This indicates that at least one client (likely the agent tool call path) is sending wrapped or partially specified job payloads. Separately, there is drift between cron provider enums in TypeScript, gateway schema, CLI flags, and UI form types, plus a UI mismatch for `cron.status` (expects `jobCount` while gateway returns `jobs`).

Goals

Stop `cron.add` `INVALID_REQUEST` spam by normalizing common wrapper payloads and inferring missing `kind` fields.

Align cron provider lists across gateway schema, cron types, CLI docs, and UI forms.

Make agent cron tool schema explicit so the LLM produces correct job payloads.

Fix the Control UI cron status job count display.

Add tests to cover normalization and tool behavior.

Non-goals



Change cron scheduling semantics or job execution behavior.

Add new schedule kinds or cron expression parsing.

Overhaul the UI/UX for cron beyond the necessary field fixes.

Findings (current gaps)

`CronPayloadSchema` in gateway excludes `signal` + `imessage`, while TS types include them.

Control UI `CronStatus` expects `jobCount`, but gateway returns `jobs`.

Agent cron tool schema allows arbitrary `job` objects, enabling malformed inputs.

Gateway strictly validates `cron.add` with no normalization, so wrapped payloads fail.

What changed

`cron.add` and `cron.update` now normalize common wrapper shapes and infer missing `kind` fields.

Agent cron tool schema matches the gateway schema, which reduces invalid payloads.

Provider enums are aligned across gateway, CLI, UI, and macOS picker.

Control UI uses the gateway's `jobs` count field for status.

Current behavior

Normalization: wrapped `data` / `job` payloads are unwrapped; `schedule.kind` and `payload.kind` are inferred when safe.

Defaults: safe defaults are applied for `wakeMode` and `sessionTarget` when missing.



Providers: Discord/Slack/Signal/iMessage are now consistently surfaced across CLI/UI.

See [Cron jobs](#) for the normalized shape and examples.

Verification

Watch gateway logs for reduced `cron.add INVALID_REQUEST` errors.

Confirm Control UI cron status shows job count after refresh.

Optional Follow-ups

Manual Control UI smoke: add a cron job per provider + verify status job count.

Open Questions

Should `cron.add` accept explicit `state` from clients (currently disallowed by schema)?

Should we allow `webchat` as an explicit delivery provider (currently filtered in delivery resolution)?

[◀ Onboarding and Config Protocol](#)

[Telegram Allowlist Hardening ▶](#)

Powered by [mintlify](#)