
G6Solver User Manual

The hosted reasoning toolkit for your AI coding agent

Edition 1.0 · September 2026

Contents

Welcome to G6Solver

- What G6Solver gives your agent
- What G6Solver is honest about

Getting started

1. Create an account
2. Get an API key
3. Connect your MCP client
4. Verify the connection

Plans and pricing

- What each plan includes
- The 30-day trial
- Two things you are paying for — subscription vs. spend

Using the tools

- Orientation and guidance
- Core solving and composition
- The job workbench (Researcher and Builder)
- Navigator (Researcher and Builder)
- Budgets and approvals
- The reliability gate (Builder)
- Diagnostics and invariants (all plans)

Budgets and spend controls

- Seeing your usage
- Paused runs
- Bring your own model key

Understanding the reliability gate

Privacy and your data

- What is stored
- What leaves your machine
- Third parties
- Deleting or exporting your data

Legal grounding and its limits

Licence and terms

Support and troubleshooting

- Rate limits by plan
- Common messages and what they mean

Frequently asked questions

Welcome to G6Solver

G6Solver is a **hosted service** that gives your AI coding agent a large toolkit of reasoning, solving, and reliability tools over the **Model Context Protocol (MCP)**. You connect Claude Code — or any MCP-compatible client — to G6Solver with a single command, and your agent gains new tools it can call during its work.

There is **nothing to install and nothing to run on your own machine**. The tools run on G6Solver's servers; your agent calls them the same way it calls any other MCP tool.

What G6Solver gives your agent

- **Goal decomposition and planning** — break a large task into structured, ordered steps.
- **Solvers and optimisation** — Bayesian optimisation, AutoML, algorithm recommendation, and multi-stage pipelines.
- **A job “workbench”** — a guided, multi-step workflow for running and refining a piece of work end to end.
- **Component discovery** — search and inspect a library of specialised components and invoke them individually.
- **Grounding and retrieval** — pull supporting evidence for a domain.
- **Budget and spend controls** — see and cap what a run may spend on paid models, and approve or resume paused work.
- **A deterministic reliability gate** — a check that flags unsafe shortcuts, missing evidence, and unsupported claims, and labels a piece of work *verified*, *qualified-draft*, or *blocked-escalated*.

What G6Solver is honest about

G6Solver is a **toolkit and a reliability process**, not an oracle. In particular, its reliability gate is a deterministic *linter* — it does not prove your code correct. The section “**Understanding the reliability gate**” explains exactly what it does and does not check. We would rather you trust the tool for what it is than be surprised later.

A note on names. “G6Solver” is the product and the service at <https://g6solver.com>. The controlling legal entity and its obligations are stated in the [Licence](#) and the [Privacy Policy](#); please refer to those documents for the definitive entity name and contact details.

Getting started

You need three things: an account, an API key, and an MCP client (such as Claude Code). It takes a few minutes.

1. Create an account

Go to <https://g6solver.com> and sign up. You can sign up with:

- **Google or GitHub** (single sign-on), or
- **email and a password**.

Email sign-ups (and social sign-ups) require **email verification** — you will receive a message with a confirmation link before your account is active.

2. Get an API key

Once signed in, open **Billing** → **API keys** (<https://g6solver.com/billing/api-keys/>) and create a key. Treat this key like a password:

- It authenticates every tool call your agent makes.
- Do not share it or commit it to a repository.
- Your licence does not permit sharing or transferring keys (see **Licence & terms**).

3. Connect your MCP client

From Claude Code, add G6Solver as an MCP server over SSE:

```
claude mcp add g6 --transport sse \  
  --url https://g6solver.com/mcp/sse \  
  --header "Authorization: Bearer YOUR_API_KEY"
```

Replace `YOUR_API_KEY` with the key you created. The endpoint is `https://g6solver.com/mcp/sse` — the `/mcp/sse` path matters; a bare `/mcp/` will not complete the connection.

Any MCP client that supports SSE transport with a bearer header can connect the same way.

4. Verify the connection

Ask your agent to call the first-call help tool, or the guide:

- `g6_first_call_help_tool` — a short orientation and your current entitlements.
- `guide_system_overview` — a plain-language tour of what is available to you.

If those return content, you are connected. If a call is refused, see **Support & troubleshooting**.

Plans and pricing

G6Solver has three plans. All prices are in **Australian dollars (AUD)** per month.

Plan	Price / month	Who it's for
Free	A\$0	Trying the core tools
Researcher	A\$20	Individuals running real workflows
Builder	A\$50	Power users who want the full toolkit and controls

What each plan includes

Free (A\$0)

- 30 core tools (solve, guide, invariants, diagnostics).
- Up to 10 requests per minute.
- Community support.

Researcher (A\$20/month)

- The full component library for individual workflows.
- Hosted service — connect Claude Code or any MCP client; nothing to install.
- Ongoing improvement: G6Solver's tools are refined over time under human supervision (see the honesty note under **Frequently asked questions**).
- Up to 100 tool calls per minute.
- Community support.
- **30-day free trial** (see below).

Builder (A\$50/month)

- Everything in Researcher.
- Multi-device licence (use on up to 3 devices).
- A usage and benchmark dashboard.
- Human-approval-gate configuration (require sign-off before certain actions).
- Up to 500 tool calls per minute.
- Priority support.

The 30-day trial

A trial gives you the **Researcher tool surface for 30 days at zero platform spend** — you can use every tool that does not cost money to run on paid models. To run *paid* models during a trial, bring your own model key (see **Budgets & spend controls**). When the trial ends, you continue on the plan you choose.

Two things you are paying for — subscription vs. spend

It helps to think of two separate meters:

1. **Your subscription** (A\$20 or A\$50) — this is your *access* to the toolkit and its rate limit. It does not, by itself, pay for calls to paid AI models.
2. **A platform spend allowance** — a separate, monthly limit on how much G6Solver will spend on paid models *on your behalf*. This is configured by the operator of your deployment. Typical defaults are around **A\$15/month for Researcher** and **A\$37.50/month for Builder**; it can also be set to unlimited. **Free and trial accounts have a zero platform-spend allowance.**

If you never need G6Solver to call a paid model for you — for example because you drive everything from your own Claude Code session, or use a locally served model — the platform spend allowance never comes into play. When it does, and a call would exceed it, the call is **refused rather than billed** and you are told why (see **Support & troubleshooting**).

There is also a **per-run cost ceiling**: a single run will pause rather than spend more than a configured amount at once, and you can approve extending a paused run up to your allowance.

Using the tools

Your agent discovers G6Solver's tools automatically once connected. The tools available to you depend on your plan. Below are the main groups.

Orientation and guidance

- `g6_first_call_help_tool` — start here; shows what you can do.
- `guide_ask`, `guide_how_to`, `guide_find_tool`, `guide_plan_workflow`, `guide_system_overview` — ask, in plain language, how to do something or which tool to use.

Core solving and composition

- `decompose_goal` — turn a goal into structured sub-steps.
- `run_bayesian`, `run_automl`, `recommend_algo` — optimisation and algorithm-selection tools.
- `run_pipeline`, `compose_and_run` — run multi-stage pipelines.
- `list_components`, `describe_component`, `invoke_component`, `invoke_component_typed` — discover and call individual components.
- `system_status` — check what the service can currently do.

The job workbench (Researcher and Builder)

A guided flagship workflow for taking a task from start to finished result:

- `job_submit`, `job_get_result`, `job_refine`, `job_export`, `job_feedback`.
- `job_workbench_start`, `job_workbench_plan`, `job_workbench_run`, `job_workbench_select`, `job_workbench_list`, `job_workbench_explain`.

Navigator (Researcher and Builder)

- `nav_discover`, `nav_search`, `nav_recommend`, `nav_inspect`, `nav_stats`, `nav_health` — explore the component library and get recommendations.

Budgets and approvals

- `budget_status`, `budget_usage_history`, `budget_telemetry` — see your spend and usage.
- `budget_approve_spend`, `budget_resume_task`, `budget_paused_tasks` — manage paused runs.
- `approval_disclosure`, `approval_grant` — the human-approval surface.
- `tier_get_status` — see your current plan and entitlements.

The reliability gate (Builder)

- `coding_agent_gate` — a pre-flight check on a proposed piece of work.
- `coding_agent_verify_answer` — a check on completed work.

See the next section for exactly what these do.

Diagnostics and invariants (all plans)

- `diagnostics_collect`, `diagnostics_export`, `diagnostics_list`, `diagnostics_status`, and related tools — report and inspect errors.
- `invariants_check`, `invariants_verify`, and related tools — check component compliance.

Your plan decides which tools you can call. If your agent calls a tool your plan does not include, G6Solver refuses it and names the lowest plan that includes it. If a tool name is not recognised at all, your client reports an unknown tool — that tool is not part of the hosted service.

Budgets and spend controls

G6Solver never spends money on a paid model without a way for you to see and cap it.

Seeing your usage

- `budget_status` — your current spend against your allowance for the period.
- `budget_usage_history` — a history of what has been spent.

The platform spend allowance is a **rolling window** (typically the last 30 days), not a calendar month, so there is no fixed monthly “reset date”; older spend simply ages out of the window.

Paused runs

If a run reaches the per-run cost ceiling, it **pauses** instead of overspending. You can then:

- `budget_paused_tasks` — list paused runs.
- `budget_approve_spend` — approve additional spend for one.
- `budget_resume_task` — resume it.

Bring your own model key

You can supply **your own OpenRouter API key** at `https://g6solver.com/billing/llm-key/`. When you do:

- Paid model calls are billed to *your* provider account, not to G6Solver.
- G6Solver’s platform spend caps do not apply to those calls.
- This is the way to run paid models during a free trial or on a zero-allowance account.

Only OpenRouter keys are accepted, and this feature is available where the operator of your deployment has enabled it. If it is unavailable, storing a key is refused with a clear message rather than silently failing.

Understanding the reliability gate

This is the most important thing to understand correctly, because it is easy to over-trust.

The coding-agent gate is a deterministic keyword *linter*, not a semantic code verifier. It reads the description and the claims your agent supplies and applies consistent, auditable rules. It is deliberately simple so that its verdict is predictable and explainable.

What it does:

- Flags **unsafe shortcuts** described in the work (for example, skipping or disabling tests, forcing a push, bypassing a check).
- Notes **missing evidence** and **claims that were made but not substantiated**.
- Applies a label: **verified**, **qualified-draft**, or **blocked-escalated**, and shows the evidence behind that label.
- May consult an optional AI safety reviewer that can only ever make the verdict *stricter* — it can never relax a verdict or remove a finding.

What it does **not** do — in the tool's own words:

"The gate counts the claims the caller supplied. It does not check them against the repository, so a claim listed here is a claim MADE, never a claim VERIFIED."

"The coding-agent gate is a deterministic keyword floor. It performs no retrieval and no grounding ... The absence of a score here is NOT evidence that the answer is grounded."

So a **gate "pass" is a qualified draft, never a proof of correctness**. Treat a `verified` label as "this cleared a consistent set of reliability checks," not "this code is guaranteed correct." The gate's value is that it is honest, repeatable, and shows its reasoning — use it to catch the obvious reliability problems, and keep your own review for correctness.

Privacy and your data

The authoritative statement is the **Privacy Policy** at <https://g6solver.com/privacy/>. This is a plain-language summary of the parts that matter most when using the tools.

What is stored

- **Account information** — your name, email, and a hashed password.
- **Billing information** — your Stripe customer id and subscription status. Card details are handled by Stripe, not stored by G6Solver.
- **Usage data** — IP address, browser/device, session logs, and tool-invocation counts.
- **API-key metadata** — key id, plan, and when it was created and last used.
- **The data you send to tools** — **the request text and any context you pass to solver tools are retained as part of your run history**, along with run status, timing, token counts, and cost. A separate cross-session learning record keeps only a one-way digest of your request, not its text.

There is currently no automatic expiry on solver run history or the learning digest — they persist until removed. If ephemeral handling matters to you, pass only what a task needs, and request deletion (below).

What leaves your machine

Only **what you explicitly pass to a tool** leaves your machine. Your source files are not read or uploaded unless you include their contents in a tool request. Data you pass to tools may be forwarded to the AI model providers that run the models.

Third parties

G6Solver relies on: AI model providers (Anthropic, OpenAI, OpenRouter), Stripe (payments), Google Cloud (servers and databases, in the United States) and Cloudflare (delivery and DDoS protection), Zoho (account email, sent through Zoho's Australian service), Sentry (error reports from our servers), and Google and GitHub if you choose to sign in with them. See the Privacy Policy for details.

Deleting or exporting your data

To request **deletion** of your data, or a **copy of your data** in a machine-readable format, email **hello@g6solver.com**. We aim to respond within 30 days. After deletion we retain only the minimal records the law requires us to keep (sign-up details and date, payment records, and your deactivation date). Tool-invocation metadata (which tool, when, and the billing record) is kept with your account. A scheduled 365-day clean-up of that metadata is being built; until it ships, nothing removes it automatically, and this manual will say so for as long as that is true.

Legal grounding and its limits

Some G6Solver tools can ground an answer in real Australian legislation, drawn from the **Open Australian Legal Corpus** (CC BY 4.0). When they do, the answer cites the instrument it relied on. Four limits matter, and none of them is hidden:

- **It is a point-in-time snapshot, not the current law.** Amendments, repeals and commencements since the corpus was collected are not captured. Confirm any provision against the official source before relying on it.
- **Legislation only — no case law.** How a court has read a provision is not grounded here.
- **A bill is not law.** 2,589 of the 43,344 indexed documents are bills, which may never have been enacted. Check the citation before treating a result as law.
- **Some jurisdictions are absent entirely.** The corpus covers the Commonwealth, New South Wales, Queensland, Western Australia, South Australia, Tasmania and Norfolk Island. **Victoria, the ACT and the Northern Territory are not included at all.**

Grounding is **reference material, not legal advice**, and a legal answer names the licensed human who has to verify it before you act on it. The full measurement record behind these statements is `docs/trust/legal-grounding-corpus.md`, held to the frozen figures in `docs/trust/legal_grounding_measurement_2026-08-28.json`.

Licence and terms

Your use of G6Solver is governed by the **Licence** (bundled with the Software and summarised here), the **Terms of Service** at <https://g6solver.com/terms/>, and the **Privacy Policy** at <https://g6solver.com/privacy/>. This is a summary; the Licence text controls.

Nature. G6Solver is proprietary software, provided as a hosted service. It is **licensed, not sold**.

What you may do (grant). With an active subscription, you have a limited, non-exclusive, non-transferable, revocable licence to **access and use** the Software through the hosted service at <https://g6solver.com> (and, where G6Solver makes one available, a licensed desktop application on devices you own or control). Your right to use ends when your subscription ends.

What you may not do (restrictions). You may not:

- copy, reproduce, or distribute the Software's source code;
- reverse engineer, decompile, or disassemble the Software;
- sublicense, lease, or lend it to a third party;
- use it to build a competing product or service;
- remove proprietary notices; or
- share or transfer your licence key or activation credentials.

Termination. The licence terminates automatically if you breach its terms or cancel your subscription; on termination you must stop using the Software.

"As is." The Software is provided without warranty, and liability is limited, as set out in the Licence.

Governing law. Queensland, Australia.

Publisher. G6Solver is operated by Daniel Busch Dental Care Pty Ltd (ABN 71 616 422 229), Queensland, Australia.

Availability today. G6Solver is currently delivered as a **hosted service**. There is no customer download or desktop installer to purchase or install today; connect over MCP as described in **Getting started**.

For licensing enquiries: legal@g6solver.com.

Support and troubleshooting

Contact: email hello@g6solver.com for support, account, privacy, and data requests.

Rate limits by plan

Plan	Requests / minute
Free	10
Researcher	100
Builder	500

If you exceed your rate limit, calls are temporarily refused; wait and retry, or move to a higher plan.

Common messages and what they mean

G6Solver returns clear, coded messages (`G6_E_...`) rather than raw errors. The most common:

Code	Meaning	What to do
G6_E_MISSING_PARAM	A required field was not supplied.	Provide the field and retry.
G6_E_BILLING_REQUIRED	A paid model call needs a valid billing context.	Ensure you have an active subscription (or credits), or run in a mode that doesn't spend platform credit.
G6_E_SPEND_CAP_UNSET	Your tier has no paid-model spend allowance configured on this deployment.	Ask the deployment operator to set an allowance for your tier, use a locally served model, or bring your own key. Your own budget can only tighten the allowance, never raise it.
G6_E_TIER_NO_PLATFORM_SPEND	Your plan (e.g. free or trial) never spends platform credit.	Move to a paid plan, bring your own key, or use a tool/model that costs nothing to run.
G6_E_CALL_COST_CEILING	One call was priced above the whole per-run ceiling.	Lower <code>max_tokens</code> , split the work, or ask the operator to raise the ceiling. Price is estimated from what you <i>ask</i> for.
G6_E_VARIABLE_PRICE_ROUTER	You named an OpenRouter routing alias (e.g. <code>openrouter/auto</code>), which has no fixed price to cap.	Name a specific model, or add your own OpenRouter key at <code>/billing/llm-key/</code> .
G6_E_PAID_CAPABILITY_NOT_ENTITLED	A paid third-party feature requires a paid plan.	Subscribe to a paid plan.
G6_E_BYO_KEY_UNSEALABLE	Your stored provider key could not be decrypted (often after a key rotation).	Re-enter your key at <code>/billing/llm-key/</code> .

For any code not listed here, the message includes a **suggested action** — read it first; it is written to tell you the next step. If you are stuck, email support and quote the code.

Frequently asked questions

Is my source code sent to G6Solver? Only what you explicitly pass into a tool. Your files are not read or uploaded on their own. Anything you do pass may be forwarded to the AI model provider that runs the model.

Can I run G6Solver on my own infrastructure? No. G6Solver is a hosted service; there is no self-hosted or source-checkout install for customers. Connect over MCP as described above.

Does G6Solver “get smarter with every use”? Is it self-modifying? G6Solver improves over time, but that improvement is **supervised and human-in-the-loop**, and confined to safe-to-change surfaces (such as prompts, routing choices, retry policies, and examples). Core governance, security boundaries, billing, and human-approval rules are **off-limits** to any automatic change.

Does a gate “pass” mean my code is correct? No. See **Understanding the reliability gate** — a pass is a *qualified draft* from a deterministic linter, not a proof of correctness.

How do I cancel? Manage your subscription from your account’s billing area, or email **hello@g6solver.com**. Your licence and access end when the subscription ends.

Which model does G6Solver use? You choose the model where a tool takes one. Name a specific model so its cost can be priced and capped; routing aliases that pick a model per request cannot be priced against a platform cap (see `G6_E_VARIABLE_PRICE_ROUTER`). On your own OpenRouter key, routing aliases work as OpenRouter intends.

G6Solver User Manual · Edition 1.0 · September 2026. This manual summarises the service as delivered; the Licence, Terms of Service, and Privacy Policy at <https://g6solver.com> are the controlling documents.