
SETUP GUIDE · EVERY PLATFORM

SAM.gov Multi-Platform

Install the SAM.gov MCP server on whichever AI client you already use. Entity registration, exclusions and debarment, contract opportunities, contract awards, federal hierarchy, and FFATA subawards, as 19 tools you can ask for in plain English. About 10 minutes from API key to first query.



Claude
via Desktop



ChatGPT
via Codex



Gemini
via CLI



Copilot
via VS Code

Tool

sam-gov-mcp, version 1.0.0
github.com/1102tools/federal-contracting-mcps

Date

August 2026
1102tools.com

Updated August 2026

sam-gov-mcp is now at version 1.0.0, rebuilt on version 2 of the MCP Python SDK. Tool names, parameters, and responses are unchanged. Two things differ from the April edition of this guide. The Claude Desktop `.mcpb` one-click bundles have been discontinued, so every platform now installs from a config block. And `lookup_psc_code` no longer describes a retired PSC as invalid; it now says the code is retired and points you at the flag that reveals it.

Part 1: Get your free SAM.gov key

1. Go to `https://sam.gov` and sign in or create a free account.
2. Click your username in the upper right, then **Account Details**.
3. In the API Key section, request or view your Public API Key.
4. Copy it somewhere safe.

The key rotates every 90 days. Plan to refresh on that cadence; an expired key shows up as a 403.

Part 2: Install uv (required runtime)

The server runs as a Python package launched by `uvx`, which ships with `uv`. Most users do not have it preinstalled.

Check first. `which uvx` (macOS, Linux) or `where uvx` (Windows). If a path comes back, skip ahead. Otherwise:

- macOS: `brew install uv`
- macOS or Linux (no Homebrew): `curl -LsSf https://astral.sh/uv/install.sh | sh`
- Windows: `winget install astral-sh.uv`
- Windows (no winget): `powershell -c "irm https://astral.sh/uv/install.ps1 | iex"`

Close and reopen your terminal, then fully quit and reopen your AI client so it inherits the new PATH. Forgetting that restart is the most common reason a correct config appears not to work.

Part 3: Pick your platform

PLATFORM	INSTALL PATH
Claude Desktop	JSON config file
Claude Code	<code>claude mcp add</code>

PLATFORM	INSTALL PATH
Codex Desktop / CLI	<code>~/.codex/config.toml</code> or <code>codex mcp add</code>
Gemini CLI	<code>gemini mcp add</code> or <code>~/.gemini/settings.json</code>
Copilot	<code>mcp.json</code> or <code>/mcp add</code>

Claude Desktop extensions (.mcpb) are discontinued

Earlier editions offered a one-click `.mcpb` download. Those bundles were retired in August 2026. They could not be signed in a way Claude Desktop recognizes, so every install raised an untrusted-developer warning with no way to clear it, and they re-resolved dependencies on each launch instead of pinning them. Existing installs keep working but will not update.

Claude Desktop

Open `~/Library/Application Support/Claude/claude_desktop_config.json` (macOS) or `%APPDATA%\Claude\claude_desktop_config.json` (Windows) and merge:

```
{
  "mcpServers": {
    "sam-gov": {
      "command": "uvx",
      "args": ["--refresh-package", "sam-gov-mcp", "--from", "sam-gov-mcp", "sam-gov-mcp"],
      "env": { "SAM_API_KEY": "your-key-here" }
    }
  }
}
```

Save, fully quit Claude Desktop (Cmd+Q on macOS), and reopen. `--refresh-package` makes `uv` check for a newer release each launch so fixes reach you automatically.

Claude Code

```
claude mcp add sam-gov --scope user --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
```

Verify with `claude mcp list`, or `/mcp` inside a session.

Codex Desktop and Codex CLI

Both read `~/.codex/config.toml` (macOS/Linux) or `%USERPROFILE%\codex\config.toml` (Windows). Configuring one configures both.

```
[mcp_servers.sam-gov]
command = "uvx"
args = ["sam-gov-mcp"]
startup_timeout_sec = 60
[mcp_servers.sam-gov.env]
SAM_API_KEY = "your-key-here"
```

Or from the CLI:

```
codex mcp add sam-gov --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
```

The `--env` flag must come before the `--` separator. `startup_timeout_sec` covers uvx's cold start on first launch.

Gemini CLI

```
gemini mcp add --scope user sam-gov --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
```

`--scope user` writes to `~/.gemini/settings.json` and makes it global. Verify with `gemini mcp list`. Note the consumer `gemini.google.com` chat does not support custom MCP servers; the CLI does.

Copilot

VS Code uses the top-level key `servers` and a `type` field, unlike the other clients. Open `.vscode/mcp.json` or your user config via Command Palette, MCP: Open User Configuration:

```
{
  "servers": {
    "sam-gov": {
      "type": "stdio",
      "command": "uvx",
      "args": ["sam-gov-mcp"],
      "env": { "SAM_API_KEY": "${input:sam_api_key}" }
    }
  },
  "inputs": [
    { "id": "sam_api_key", "type": "promptString", "description": "SAM.gov API key",
      "password": true }
  ]
}
```

The `${input:...}` pattern prompts for the key instead of storing it in plaintext. Reload VS Code (Command Palette, Developer: Reload Window), then open Copilot Chat in Agent mode.

Visual Studio 2026 / 2022 17.14+. Same `servers` schema, in `%USERPROFILE%\mcp.json` (global) or `<SOLUTIONDIR>\mcp.json` (per-solution). Or click the green + in the chat tool picker, then Add MCP

Server, then switch the chat dropdown from Ask to Agent.

JetBrains IDEs (IntelliJ, PyCharm, GoLand, WebStorm, Rider). Click the Copilot status-bar icon, then Edit Settings, Model Context Protocol, Configure. Paste the same `mcpServers` JSON shape used for Claude Desktop above. Requires the Copilot plugin updated to latest. MCP support is in public preview.

Copilot CLI. Run `copilot`, then `/mcp add` interactively (Type: STDIO, Command `uvx`, Args `sam-gov-mcp`, Env `{"SAM_API_KEY": "your-key-here"}`). Or edit `~/.copilot/mcp-config.json` directly. Note it uses `mcpServers` (not `servers`) and `type: "local"` (not `stdio`). Manage with `/mcp show`, `/mcp edit`, `/mcp disable`.

Part 4: Verify it works

Restart your client, then try:

"Look up the SAM registration status for UEI PJM8HKWCB5S3."

"Show me sources sought notices for small business set-asides posted this week."

"Is there an active exclusion for this vendor?"

Real records back means you are set. 19 tools are available covering Entity Management, Exclusions, Opportunities, Contract Awards (the FPDS replacement), Federal Hierarchy, and both FFATA subaward APIs.

Part 5: Troubleshooting

spawn uvx ENOENT, or uvx not found. `uv` is missing, or your client started before it was on the PATH. Install with `brew install uv` (macOS) or `winget install astral-sh.uv` (Windows), confirm with `which uvx`, then fully quit and reopen the client. Closing the window is not enough on macOS.

403 or invalid key. SAM.gov keys rotate every 90 days. Regenerate in Account Details and update the config. Check for a trailing space.

A PSC lookup returns 404 for a code you know is real. The endpoint defaults to active codes only, so a retired PSC returns 404. Retry with `active_only='ALL'` to see the retired entry and its end date. D302 is the worked example: a genuine PSC, retired 2020-10-29.

A search returns far more records than expected. Check parameter spelling. The server sets `extra='forbid'` so a typo'd parameter name raises an error rather than being silently dropped, but a valid parameter with an unintended value still filters nothing.

Changes to config don't take effect. Every client caches config at launch. Fully quit and reopen; closing the window is not enough on macOS.

More info

- All eight MCP servers, tool counts, and key requirements: <https://1102tools.com/tools>
- Universal setup guide covering every server: <https://1102tools.com/downloads/1102tools-universal-setup.pdf>
- Source, changelog, and testing record: <https://github.com/1102tools/federal-contracting-mcps>

Updated August 2026 by James Jenrette / 1102tools. MIT licensed. Independently developed and not endorsed by any federal agency.