UPLOAD TO AI · GET INSTALL HELP

1102tools MCP Setup

This document is meant to be uploaded to the AI of your choice for technical support. Drop the PDF into Claude, ChatGPT, Gemini, or Copilot. Tell it which 1102tools MCP you want and which client you use. The AI will walk you through every step. About 10 minutes from API key to first query.

Claude	ChatGPT	Gemini	Copilot
via Desktop	via Codex	via CLI	via VS Code

Tool
8 MCP servers
github.com/1102tools/federal-contracting-mcps

Date
April 2026
1102tools.com

How to use this guide

This document is meant to be uploaded to the AI of your choice for technical support installing any 1102tools MCP server. Drop this PDF into Claude, ChatGPT, Gemini, or Copilot. Then tell the AI which MCP you want and which client you use. Example prompts:

- "Help me install the SAM.gov MCP on Claude Desktop. I'm on macOS."
- "Install all 8 MCPs in Codex CLI. I have a SAM.gov key but not the others yet."
- "Walk me through setting up GSA Per Diem in Gemini CLI on Windows."

The AI will read the sections it needs (API keys, your platform's config file, the per-MCP package name and env var) and produce step-by-step instructions tailored to your setup. Most users go from zero to first working query in about 10 minutes.

If you prefer to install yourself, every snippet you need is below. The structure is: Part 1 covers free API keys, Part 2 covers the five client platforms, Part 3 is the MCP catalog (package names and env vars), Part 4 verifies, Part 5 is troubleshooting.

What 1102tools is

Eight MCP (Model Context Protocol) servers that turn federal data APIs into something you can ask in plain English from your existing AI client. No paid research platforms, no CSV exports, no learning a new API.

MCP	API key	What it covers
SAM.gov	Required	Entity registration, exclusions, opportunities, contract awards (FPDS), federal hierarchy, FFATA subawards. 19 tools.
USASpending	None	Federal contracts, grants, FFATA subawards, recipient profiles, agency depth, geographic search. 55 tools.
GSA CALC+	None	Awarded NTE hourly rates from GSA MAS contracts, 230K+ records. 8 tools.
BLS OEWS	Required	Market wage data, ~830 occupations across 530+ metro areas. 7 tools.
GSA Per Diem	Required	Federal travel per diem for all CONUS, lodging and M&IE. 6 tools.
eCFR	None	Current CFR text updated daily, FAR/DFARS clause lookup, version comparison. 13 tools.
Federal Register	None	Federal Register documents since 1994, proposed/final rules, executive orders. 8 tools.
Regulations.gov	Required	Federal rulemaking dockets, public comments, docket histories. 8 tools.

All ship as PyPI packages installable via `uvx`. Same code, every platform.

Part 1: Get your free API keys

Skip this section for any MCP you do not plan to use. All four keys are free.

SAM.gov key

Required for the SAM.gov MCP.

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.

BLS key

Required for the BLS OEWS MCP.

1. Go to https://data.bls.gov/registrationEngine and register.
2. The key arrives by email. Activate via the link.
3. Copy the key.

No rotation. The key is permanent unless you regenerate it.

GSA Per Diem key (api.data.gov)

Required for the GSA Per Diem MCP. This is a free api.data.gov key that also works with several other federal APIs.

1. Go to https://api.data.gov/signup.
2. Fill out the short form. The key appears on screen and is emailed.
3. Copy the key.

No rotation.

Regulations.gov key (api.data.gov)

Required for the Regulations.gov MCP. Same kind of key as GSA Per Diem (an api.data.gov key), but you can reuse the one you already got, or sign up again.

1. Same flow as above: https://api.data.gov/signup.
2. Reuse the key you already received.

Part 2: Pick your platform

Same MCP packages. Multiple ways to plug them in. Pick the path that matches your AI client.

Platform	Best for	Install path
Claude Desktop	GUI users on Mac or Windows	JSON config in `claude_desktop_config.json`
Claude Code	Claude users in the terminal	`claude mcp add` one-liner
Codex Desktop	OpenAI users who want a GUI	Settings, Integrations and MCP, Add server
Codex CLI	OpenAI users in the terminal	`codex mcp add` or TOML config
Gemini CLI	Google stack users	`gemini mcp add` or JSON in `~/.gemini/settings.json`
Copilot	VS Code, Visual Studio, JetBrains, or Copilot CLI	One-click MCP gallery, `mcp.json`, or `/mcp add`

All paths use the same package and the same keys from Part 1.

Option A: Claude Desktop

For Mac and Windows users who want a GUI.

Install Claude Desktop. Download from https://claude.ai/download. Install, open at least once, and sign in.

Edit the config file. Open the config file for your OS:

- macOS: `~/Library/Application Support/Claude/claude_desktop_config.json`
- Windows: `%APPDATA%\Claude\claude_desktop_config.json`

Paste this block (omit any MCP you do not want, omit the `env` block for keyless MCPs). If the file already has content, merge into the existing `mcpServers` object.

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

```
    "usaspending": {
      "command": "uvx",
      "args": ["usaspending-mcp"]
    },
    "gsa-calc": {
      "command": "uvx",
      "args": ["gsa-calc-mcp"]
    },
    "bls-oews": {
      "command": "uvx",
      "args": ["bls-oews-mcp"],
      "env": { "BLS_API_KEY": "your-key-here" }
    },
    "gsa-perdiem": {
      "command": "uvx",
      "args": ["gsa-perdiem-mcp"],
      "env": { "PERDIEM_API_KEY": "your-key-here" }
    },
    "ecfr": {
      "command": "uvx",
      "args": ["ecfr-mcp"]
    },
    "federal-register": {
      "command": "uvx",
      "args": ["federal-register-mcp"]
    },
    "regulations-gov": {
      "command": "uvx",
      "args": ["regulationsgov-mcp"],
      "env": { "REGULATIONS_GOV_API_KEY": "your-key-here" }
    }
  }
}
```

Replace each `your-key-here` with the corresponding key from Part 1. Save. Fully quit Claude Desktop (Cmd+Q on macOS; right-click the Claude icon in the system tray, then Quit on Windows) and reopen.

Option B: Claude Code

For Anthropic's official terminal CLI. Works on macOS, Windows (via WSL), and Linux.

Install Claude Code. Follow the install at https://claude.com/claude-code. Sign in.

Add the MCPs. One command per MCP. Skip any you do not want.

```
claude mcp add sam-gov --scope user --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
claude mcp add usaspending --scope user -- uvx usaspending-mcp
claude mcp add gsa-calc --scope user -- uvx gsa-calc-mcp
claude mcp add bls-oews --scope user --env BLS_API_KEY=your-key-here -- uvx bls-oews-mcp
claude mcp add gsa-perdiem --scope user --env PERDIEM_API_KEY=your-key-here -- uvx gsa-perdiem-mcp
claude mcp add ecfr --scope user -- uvx ecfr-mcp
claude mcp add federal-register --scope user -- uvx federal-register-mcp
claude mcp add regulations-gov --scope user --env REGULATIONS_GOV_API_KEY=your-key-here -- uvx regulationsgov-mcp
```

Verify with `claude mcp list`. Inside a session, run `/mcp` to see live status.

Scopes. `--scope user` makes it available across all your projects. `--scope project` shares it via a committed `.mcp.json` (good for teams). Default is `local` (this project, you only).

Option C: Codex Desktop

For ChatGPT subscribers who want a GUI app on macOS or Windows. Codex is included with eligible ChatGPT plans, subject to plan limits. Sign in with your existing ChatGPT account. Same login works across the Codex app, Codex CLI, and the Codex IDE extension.

Install Codex Desktop.

- macOS (Apple Silicon): https://persistent.oaistatic.com/codex-app-prod/Codex.dmg
- macOS (Intel): https://persistent.oaistatic.com/codex-app-prod/Codex-latest-x64.dmg
- Windows (Microsoft Store): https://get.microsoft.com/installer/download/9PLM9XGG6VKS

Or visit https://developers.openai.com/codex/app. Sign in.

Add MCPs via the GUI. Open **Settings**, go to **Integrations and MCP, MCP**, click **Add server**, choose **STDIO / Local command**, then for each MCP fill in:

- SAM.gov: Name `sam-gov`, Command `uvx`, Args `sam-gov-mcp`, Env `SAM_API_KEY=your-key-here`
- USASpending: Name `usaspending`, Command `uvx`, Args `usaspending-mcp`
- GSA CALC+: Name `gsa-calc`, Command `uvx`, Args `gsa-calc-mcp`
- BLS OEWS: Name `bls-oews`, Command `uvx`, Args `bls-oews-mcp`, Env `BLS_API_KEY=your-key-here`

- GSA Per Diem: Name `gsa-perdiem`, Command `uvx`, Args `gsa-perdiem-mcp`, Env `PERDIEM_API_KEY=your-key-here`
- eCFR: Name `ecfr`, Command `uvx`, Args `ecfr-mcp`
- Federal Register: Name `federal-register`, Command `uvx`, Args `federal-register-mcp`
- Regulations.gov: Name `regulations-gov`, Command `uvx`, Args `regulationsgov-mcp`, Env `REGULATIONS_GOV_API_KEY=your-key-here`

Save after each. Servers start on next session.

Or edit the TOML directly. Codex Desktop reads the same config file as the CLI. Open `~/.codex/config.toml` (macOS/Linux) or `%USERPROFILE%\.codex\config.toml` (Windows) and append:

```
[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"

[mcp_servers.usaspending]
command = "uvx"
args = ["usaspending-mcp"]

[mcp_servers.gsa-calc]
command = "uvx"
args = ["gsa-calc-mcp"]

[mcp_servers.bls-oews]
command = "uvx"
args = ["bls-oews-mcp"]
[mcp_servers.bls-oews.env]
BLS_API_KEY = "your-key-here"

[mcp_servers.gsa-perdiem]
command = "uvx"
args = ["gsa-perdiem-mcp"]
[mcp_servers.gsa-perdiem.env]
PERDIEM_API_KEY = "your-key-here"

[mcp_servers.ecfr]
command = "uvx"
args = ["ecfr-mcp"]
```

```
[mcp_servers.federal-register]
command = "uvx"
args = ["federal-register-mcp"]

[mcp_servers.regulations-gov]
command = "uvx"
args = ["regulationsgov-mcp"]
[mcp_servers.regulations-gov.env]
REGULATIONS_GOV_API_KEY = "your-key-here"
```

`startup_timeout_sec = 60` covers uvx's cold-start cost on first launch. Save and restart Codex Desktop.

Option D: Codex CLI

For ChatGPT subscribers who live in the terminal. Same package, same auth, same `~/.codex/config.toml` as Codex Desktop. Configuring one configures both.

Install Codex CLI. `npm install -g @openai/codex` or `brew install --cask codex` on macOS. Then run `codex` and sign in.

Add MCPs via `codex mcp add`. Or paste the TOML block from Option C; both clients read the same file.

```
codex mcp add sam-gov --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
codex mcp add usaspending -- uvx usaspending-mcp
codex mcp add gsa-calc -- uvx gsa-calc-mcp
codex mcp add bls-oews --env BLS_API_KEY=your-key-here -- uvx bls-oews-mcp
codex mcp add gsa-perdiem --env PERDIEM_API_KEY=your-key-here -- uvx gsa-
perdiem-mcp
codex mcp add ecfr -- uvx ecfr-mcp
codex mcp add federal-register -- uvx federal-register-mcp
codex mcp add regulations-gov --env REGULATIONS_GOV_API_KEY=your-key-here -- uvx
regulationsgov-mcp
```

The `--env` flag must come before the `--` separator. Verify with `codex mcp list`.

Option E: Gemini CLI

For Google's Gemini CLI. Works on macOS, Windows, and Linux. Requires Node.js 20+. Note: the consumer gemini.google.com chat does not support custom MCP servers; the CLI does. Gemini Code

Assist (VS Code) and Android Studio's Gemini agent also read MCP config from `~/.gemini/settings.json`.

Install Gemini CLI. `npm install -g @google/gemini-cli` or `brew install gemini-cli` on macOS/Linux. Run `gemini` and sign in with your Google account.

Add MCPs via `gemini mcp add`.

```
gemini mcp add --scope user sam-gov --env SAM_API_KEY=your-key-here -- uvx sam-gov-mcp
gemini mcp add --scope user usaspending -- uvx usaspending-mcp
gemini mcp add --scope user gsa-calc -- uvx gsa-calc-mcp
gemini mcp add --scope user bls-oews --env BLS_API_KEY=your-key-here -- uvx bls-oews-mcp
gemini mcp add --scope user gsa-perdiem --env PERDIEM_API_KEY=your-key-here -- uvx gsa-perdiem-mcp
gemini mcp add --scope user ecfr -- uvx ecfr-mcp
gemini mcp add --scope user federal-register -- uvx federal-register-mcp
gemini mcp add --scope user regulations-gov --env REGULATIONS_GOV_API_KEY=your-key-here -- uvx regulationsgov-mcp
```

`--scope user` makes it global (`~/.gemini/settings.json`). Default is project scope. Verify with `gemini mcp list`.

Or paste JSON. Open `~/.gemini/settings.json`. Use the same JSON block from Option A (Claude Desktop). Both clients read the same shape.

Option F: Copilot

Copilot supports MCP servers across multiple IDEs and the CLI. Pick the one you use.

VS Code. Open the **Extensions** view, search `@mcp` to browse the MCP gallery, OR open `.vscode/mcp.json` (workspace) or your user-level `mcp.json` via Command Palette, then **MCP: Open User Configuration**.

VS Code uses the top-level key `servers` (not `mcpServers`) and a `type: "stdio"` field. Different from the other clients. Block:

```
{
  "servers": {
    "sam-gov": {
      "type": "stdio",
      "command": "uvx",
```

```
      "args": ["sam-gov-mcp"],
      "env": { "SAM_API_KEY": "${input:sam_api_key}" }
    },
    "usaspending": { "type": "stdio", "command": "uvx", "args": ["usaspending-
mcp"] },
    "gsa-calc": { "type": "stdio", "command": "uvx", "args": ["gsa-calc-mcp"] },
    "bls-oews": {
      "type": "stdio",
      "command": "uvx",
      "args": ["bls-oews-mcp"],
      "env": { "BLS_API_KEY": "${input:bls_api_key}" }
    },
    "gsa-perdiem": {
      "type": "stdio",
      "command": "uvx",
      "args": ["gsa-perdiem-mcp"],
      "env": { "PERDIEM_API_KEY": "${input:perdiem_api_key}" }
    },
    "ecfr": { "type": "stdio", "command": "uvx", "args": ["ecfr-mcp"] },
    "federal-register": { "type": "stdio", "command": "uvx", "args": ["federal-
register-mcp"] },
    "regulations-gov": {
      "type": "stdio",
      "command": "uvx",
      "args": ["regulationsgov-mcp"],
      "env": { "REGULATIONS_GOV_API_KEY": "${input:regulations_gov_api_key}" }
    }
  },
  "inputs": [
    { "id": "sam_api_key", "type": "promptString", "description": "SAM.gov API
key", "password": true },
    { "id": "bls_api_key", "type": "promptString", "description": "BLS API key",
"password": true },
    { "id": "perdiem_api_key", "type": "promptString", "description": "GSA Per
Diem API key", "password": true },
    { "id": "regulations_gov_api_key", "type": "promptString", "description":
"Regulations.gov API key", "password": true }
  ]
}
```

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

Visual Studio 2026 / 2022 17.14+. Use `%USERPROFILE%\.mcp.json` (global) or `<SOLUTIONDIR>\.mcp.json` (per-solution, source-controllable). Same `servers` schema as VS

Code. 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 statusbar icon, then **Edit Settings, Model Context Protocol, Configure**. Paste the same `mcpServers` JSON shape used for Claude Desktop / Gemini. 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` (or any MCP package), Env: `{"SAM_API_KEY": "your-key-here"}`). Or edit `~/.copilot/mcp-config.json` directly. Note this config uses `mcpServers` (not `servers`) and `type: "local"` (not `stdio`). Manage with `/mcp show`, `/mcp edit`, `/mcp disable`.

Part 3: MCP catalog

Every MCP, every package name, every env var, in one place. The AI reading this doc uses these to compose the right snippet for any platform from Part 2.

SAM.gov

- MCP name: `sam-gov`
- PyPI package: `sam-gov-mcp`
- Required env var: `SAM_API_KEY`
- Get the key: https://sam.gov, Account Details, API Key
- 19 tools. Try: "Show me all sources sought notices for small business set-asides posted this week."

USASpending

- MCP name: `usaspending`
- PyPI package: `usaspending-mcp`
- No API key.
- 55 tools. Try: "Pull the 12-month award history for Leidos. Break down by agency, NAICS, and ceiling vs. obligated amounts."

GSA CALC+

- MCP name: `gsa-calc`

- PyPI package: `gsa-calc-mcp`
- No API key.
- 8 tools. Try: "What's the median ceiling rate for a Senior Software Engineer with a Bachelor's and 10 years experience?"

BLS OEWS

- MCP name: `bls-oews`
- PyPI package: `bls-oews-mcp`
- Required env var: `BLS_API_KEY`
- Get the key: https://data.bls.gov/registrationEngine
- 7 tools. Try: "What's the 75th-percentile wage for Computer and Information Research Scientists in the DC metro area?"

GSA Per Diem

- MCP name: `gsa-perdiem`
- PyPI package: `gsa-perdiem-mcp`
- Required env var: `PERDIEM_API_KEY` (an api.data.gov key)
- Get the key: https://api.data.gov/signup
- 6 tools. Try: "What are the per diem rates for Austin, TX in October?"

eCFR

- MCP name: `ecfr`
- PyPI package: `ecfr-mcp`
- No API key.
- 13 tools. Try: "Pull the current text of FAR 52.212-4 and compare to 2020."

Federal Register

- MCP name: `federal-register`
- PyPI package: `federal-register-mcp`
- No API key.
- 8 tools. Try: "Show me all proposed FAR rules with open comment periods."

Regulations.gov

- MCP name: `regulations-gov`
- PyPI package: `regulationsgov-mcp`
- Required env var: `REGULATIONS_GOV_API_KEY` (an api.data.gov key)
- Get the key: https://api.data.gov/signup
- 8 tools. Try: "Show me the latest comments on the FAR 2025-XX docket."

Part 4: Verify it works

Open a new chat or session in your client and try a query that matches one of the MCPs you installed. For example:

> *Show me all sources sought notices for small business set-asides posted this week.*

You should get a short list of opportunities. The first call takes 30 to 60 seconds while uvx downloads and caches the package. After that, it's instant.

If you get an error, see Part 5.

Part 5: Troubleshooting

General

The MCP didn't load. Restart your client fully. Most clients only read MCP config at launch.

The key was rejected. Recheck the key: no extra quotes, no trailing spaces, no line breaks. SAM.gov keys rotate every 90 days. Regenerate at sam.gov.

The first call hangs. First run downloads the Python package. Allow 30 to 60 seconds, then retry.

Verify the install per platform with the matching list command: `claude mcp list`, `codex mcp list`, `gemini mcp list`, or VS Code's **MCP: List Servers** Command Palette action.

uvx not found. Install `uv`: `brew install uv` (macOS) or `winget install astral-sh.uv` (Windows). Restart your shell so the new PATH is picked up. Verify with `which uvx`.

Claude Desktop

The config file doesn't exist. Create it. Paste the block, save, restart Claude Desktop.

The MCP isn't available. Open Claude Desktop **Settings, Extensions**. Confirm the MCP is listed and toggled on. Click the entry, then **Configure** to recheck the API key. A red error badge usually means a missing or incorrect key.

Server logs:

- macOS: `tail -n 50 -f ~/Library/Logs/Claude/mcp-server-{name}.log`
- Windows: `%APPDATA%\Claude\logs\mcp-server-{name}.log`

Claude Code

`claude mcp list` returns nothing. Re-run `claude mcp add` with `--scope user` so it persists across projects. Check that uvx is on your PATH (`which uvx`). Inside a session, `/mcp` shows live connection status.

Codex Desktop and Codex CLI

Both share `~/.codex/config.toml`, so a fix in one applies to the other.

The server doesn't appear in Codex Desktop's Settings UI. Edit `~/.codex/config.toml` directly and restart Codex; the TOML block is authoritative.

`codex mcp list` returns nothing. Verify `~/.codex/config.toml` exists and has the `[mcp_servers.{name}]` block. TOML is whitespace-sensitive; make sure block headers are on their own lines.

First call times out. Add `startup_timeout_sec = 60` to the `[mcp_servers.{name}]` block to cover uvx's cold-start cost.

Gemini CLI

Gemini doesn't see the tool. Run `gemini mcp list` to confirm registration. If it shows disabled, run `gemini mcp enable {name}`. Verify the JSON parses cleanly with `python3 -m json.tool ~/.gemini/settings.json`. Restart Gemini CLI.

Copilot

Copilot Chat doesn't show the tools. Make sure Copilot Chat is in **Agent** mode, not Ask or Edit mode. In VS Code: Command Palette, **MCP: List Servers** to confirm the MCP is registered and

started. If status shows an error, click the entry to see stderr; missing API key is the most common cause. JetBrains: click the Copilot status-bar icon, then **MCP Servers** panel. Copilot CLI: `/mcp show`.

VS Code can't find `uvx`. Install uv first: `curl -LsSf https://astral.sh/uv/install.sh | sh` (macOS or Linux) or `powershell -c "irm https://astral.sh/uv/install.ps1 | iex"` (Windows). Reload VS Code so it picks up the new PATH.

Part 6: More info

- Website: https://1102tools.com
- Tools page (every config snippet): https://1102tools.com/tools
- Install page (with video walkthrough): https://1102tools.com/install
- GitHub: https://github.com/1102tools/federal-contracting-mcps
- Issues: https://github.com/1102tools/federal-contracting-mcps/issues

> Setup guide prepared April 2026 by James Jenrette / 1102tools. MIT licensed. Source: github.com/1102tools/federal-contracting-mcps.