

● WARMGTM · MCP SERVER

Integration Guide

Connect your AI assistant to WarmGTM and manage LinkedIn campaigns, leads, conversations, posts, and analytics through natural language.

Claude.ai

Claude Desktop

Claude Code

ChatGPT

Cursor

Windsurf

REST API

SERVER URL

<https://mcp.warmgtm.com>

TOOLS AVAILABLE

43 MCP Tools

The WarmGTM MCP Server lets you drive your entire LinkedIn outreach workflow from any AI assistant that speaks the Model Context Protocol, or via a clean REST API if you prefer to build your own integration.

43

MCP TOOLS

7+

PLATFORMS

| 120

REQ/MIN API

OAuth

2.1 READY

Prerequisites

- A **WarmGTM account** with an active subscription
- Your **WarmGTM API key**, grab it from the WarmGTM Dashboard → Settings → API Keys

 **TIP** Always start by calling `get_encrypted_identifiers`, it returns the encrypted IDs required by most other tools.

≡ In this guide

[01 Overview & Prerequisites](#)[02 Available Tools \(43\)](#)[03 Claude.ai \(Web\) setup](#)[04 Claude Desktop setup](#)[05 Claude Code \(CLI\) setup](#)[06 ChatGPT Custom GPT Actions](#)[07 Cursor setup](#)[08 Windsurf setup](#)[09 Any MCP-compatible client](#)[10 REST API \(cURL, Postman\)](#)[11 REST endpoints reference](#)[12 Social Signals \(Keyword Posts\)](#)[13 Troubleshooting](#)[14 Support & Resources](#)

43 tools organised by category. Each one is available over MCP and as a REST endpoint.



Read & Analytics

19

`get_encrypted_identifiers` , senders, campaigns & org IDs

`get_dashboard_info` , connections, messages, campaign stats

`get_campaign_info` , list campaigns with analytics

`get_campaign_routines_v2` , smart & shared routines

`get_campaign_detail` , single-campaign deep dive

`get_campaign_members` , members of a routine

`get_campaign_lead_conversations` , post-reply threads

`get_posts` / `get_post` , LinkedIn posts

`get_crm_integrations` , configured CRMs

`get_lead_lists` , filter by sender, type, name

`get_lead_list_members` , paginated list members

`get_linkedin_senders` , sender accounts + stats

`get_chat_members` , conversation members

`get_member_conversations` , full message thread

`get_comments_list` , Social Signal approvals

`get_webhooks` / `get_webhooks_list`

`get_connections` , LinkedIn 1st-degree connections



Posts

7

`create_post` , compose or schedule

`update_post` , edit content or reschedule

`delete_post` , remove post or single schedule

`fetch_linkedin_posts` , bulk import from LinkedIn

`sync_post` , pull latest likes & comments

`import_post_leads` , likers/commenters → leads

`generate_content` , AI-generated post copy



Leads & Campaigns

3

`create_leads` , from search URL, CSV, Sales Nav, post engagement

`create_lead_list` , from specific profile URLs

`add_leads_to_campaign` , push leads into a campaign



Messaging & Comments

2

`send_message` , DM a prospect via a sender

`set_comment` , approve / reject Social Signal comments

Webhooks

4

`create_webhook` , real-time event notifications

`delete_webhook` , remove permanently

`update_webhook` , change URL / events / status

`send_chat_webhook_data` , push a full thread on demand

CRM & API

2

`sync_contacts_to_crm` , HubSpot, Salesforce & more

`create_api_key` , programmatic access

Social Signals (Keyword Posts)

6

Track LinkedIn posts matching keywords, surfaces prospects actively posting around your topics. Each fetch counts against the sender's daily quota (default 3/day).

`fetch_linkedin_keyword_posts` ,
create/reactivate + queue scrape (supports recurring
schedules)

`update_keyword_post_schedule` , reschedule
recurring or one-off

`retrieve_keyword_posts` , read scraped results
with filters

`change_keyword_post_status` , pause / resume

`get_linkedin_keyword_post_status` , signals +
sync times + quota

`delete_keyword_post` , soft-delete, stop future
scrapes

1 Claude.ai (Web)

Connects via MCP Connectors with OAuth 2.1, no API key pasting in config files.

- 1 Open `claude.ai`
- 2 Go to **Settings** → **Connectors**
- 3 Click **Add custom MCP connector**
- 4 Enter the URL: `https://mcp.warmgtm.com/mcp`
- 5 Click **Connect**, you'll be redirected to the WarmGTM authorization page
- 6 Enter your WarmGTM API key and click **Authorize**
- 7 Start a new chat and ask about your campaigns

Example prompts

// Try any of these

Show me my dashboard stats for the last 30 days

List all active campaigns

How many leads are in my "Tech Founders" lead list?

Show me recent conversations from my LinkedIn sender

2 Claude Desktop

Edit the Claude Desktop configuration file on your machine.

macOS: `~/Library/Application Support/Claude/claude_desktop_config.json`

Windows: `%APPDATA%\Claude\claude_desktop_config.json`

```
{
  "mcpServers": {
    "warmgtm": {
      "type": "streamable-http",
      "url": "https://mcp.warmgtm.com/mcp",
      "headers": {
        "X-API-KEY": "YOUR_WARMGTM_API_KEY"
      }
    }
  }
}
```

i NOTE Replace `YOUR_WARMGTM_API_KEY` with your actual key and restart Claude Desktop after saving.

3 Claude Code (CLI)

One command, installed globally for every future session.

```
claude mcp add warmgtm \
  --transport http \
  https://mcp.warmgtm.com/mcp \
  --header "X-API-KEY: YOUR_WARMGTM_API_KEY"
```

4 ChatGPT (Custom GPT Actions)

Uses the OpenAPI spec to import WarmGTM as GPT Actions.

✓ **NOTE** ChatGPT uses REST endpoints (/api/*), not the MCP protocol. All the same data is available.

- 1 Go to **ChatGPT** → **Explore GPTs** → **Create a GPT**
- 2 Click the **Configure** tab
- 3 Scroll down to **Actions** → click **Create new action**
- 4 Click **Import from URL** and enter: `https://mcp.warmgtm.com/openapi.json`
- 5 All endpoints will be imported automatically
- 6 Under **Authentication**: API Key → Custom → Header name `X-API-KEY` → your key
- 7 Save, then try "Get my encrypted identifiers" in the preview panel

5 Cursor & Windsurf

Both editors use the same streamable-HTTP MCP config.

Cursor: add to `.cursor/mcp.json` (project root or global). Windsurf: add to the MCP configuration.

```
{
  "mcpServers": {
    "warmgtm": {
      "type": "streamable-http",
      "url": "https://mcp.warmgtm.com/mcp",
      "headers": {
        "X-API-KEY": "YOUR_WARMGTM_API_KEY"
      }
    }
  }
}
```

Restart the editor after saving, WarmGTM tools will appear in the MCP tool list.

6 Any MCP-Compatible Client

Works with any client supporting the Streamable HTTP transport.

SETTING	VALUE
URL	<code>https://mcp.warmgtm.com/mcp</code>
Auth Header	<code>X-API-KEY: YOUR_WARMGTM_API_KEY</code>
Transport	Streamable HTTP (POST/GET/DELETE on <code>/mcp</code>)

Every tool is also a standard REST endpoint, perfect for scripts, dashboards, or any HTTP client.

BASE URL

mcp.warmgtm.com/api

AUTH

X-API-KEY header

DOCS

</docs> (Swagger UI)

Quick examples

```
# Get encrypted identifiers
curl -X POST -H "X-API-KEY: your_key" \
  https://mcp.warmgtm.com/api/encrypted-identifiers

# Dashboard analytics for a date range
curl -H "X-API-KEY: your_key" \
  "https://mcp.warmgtm.com/api/dashboard-info?start_date=2025-01-01&end_date=2025-01-31"

# List active campaigns
curl -H "X-API-KEY: your_key" \
  "https://mcp.warmgtm.com/api/campaign-info?campaign_status=active&limit=10"

# Send a LinkedIn message
curl -X POST -H "X-API-KEY: your_key" -H "Content-Type: application/json" \
  -d '{"linkedin_user_information_id":"abc123","profile_id":"xyz","message":"Hi!"}' \
  https://mcp.warmgtm.com/api/send-message

# Sync contacts to CRM
curl -X POST -H "X-API-KEY: your_key" -H "Content-Type: application/json" \
  -d '{"member_ids":[1,2,3],"member_ids_type":"lead","crm_integration_ids":[10]}' \
  https://mcp.warmgtm.com/api/sync-contacts
```

 **POSTMAN** Import → paste <https://mcp.warmgtm.com/openapi.json> . All endpoints arrive with descriptions and parameters pre-filled.

Endpoints Reference

ANALYTICS & READ

METHOD	ENDPOINT	REQUIRED	OPTIONAL
POST	/api/encrypted-identifiers	—	—
GET	/api/dashboard-info	—	start_date, end_date, sender IDs, routine IDs
POST	/api/dashboard-info	—	same as GET, JSON body
GET	/api/campaign-info	—	limit, offset, term, status, type, sort
GET	/api/campaign-routines-v2	—	limit, offset, term, status, unassigned
GET	/api/campaign-detail	campaign_id	start_date, end_date
POST	/api/campaign-members	routine_id or name	sender_id, stage, limit, offset
POST	/api/campaign-lead-conversations	—	routine_id, name, stage, status, paging
GET	/api/lead-lists	—	user_ids, import_type, term, paging
GET	/api/lead-list-members	lead_list_id	term, limit, offset
GET	/api/linkedin-senders	—	linkedin_user_id
GET	/api/connections	sender_id	term, limit, offset
GET	/api/chat-members	sender IDs	routine IDs, term, tags, paging
GET	/api/member-conversations	sender_id, member_id	limit, offset
GET	/api/comments-list	org_member_id	status, sender IDs, paging
GET	/api/crm-integrations	—	crm_name

POSTS

METHOD	ENDPOINT	REQUIRED	OPTIONAL
GET	/api/posts	sender_id	status
GET	/api/posts/:post_id	sender_id	scheduled_id

METHOD	ENDPOINT	REQUIRED	OPTIONAL
POST	/api/posts	sender_id, post	schedule_now, scheduled_at, org_member_id
PUT	/api/posts/:post_id	sender_id	post, schedule, scheduled_id
DEL	/api/posts/:post_id	sender_id	scheduled_id
POST	/api/posts/:post_id/sync	sender_id	—
POST	/api/posts/:post_id/import_leads	sender_id, import_type	—
POST	/api/posts/generate_content	sender_id	topic, tone, length
POST	/api/posts/fetch_linkedin	sender_id	—

LEADS, MESSAGING & WEBHOOKS

METHOD	ENDPOINT	REQUIRED	OPTIONAL
POST	/api/create-leads	sender_id, campaign_name, list_type	url, lead_count, post_url, category
POST	/api/add-leads-to-campaign	campaign_id, list_id, lead_ids	—
POST	/api/lead-list	sender_id, campaign_id, profiles, count	lead_list_id
POST	/api/send-message	sender_id, profile_id, message	—
POST	/api/set-comment	comment, uid, id, action_type	—
GET	/api/get-webhook	—	id, level
GET	/api/get-webhooks-list	—	—
POST	/api/create-webhook	level, url, events, name, org_member_id	status, all_campaigns
POST	/api/update-webhook	id	activate, url, events, status, name
DEL	/api/delete-webhook	id (query)	—
POST	/api/send-chat-webhook-data	webhook_id, member_id	—
POST	/api/sync-contacts	member_ids, type, integration_ids	—
POST	/api/create-api-key	—	description

SOCIAL SIGNALS (KEYWORD POSTS)

METHOD	ENDPOINT	REQUIRED	OPTIONAL
POST	/api/keyword-posts/fetch	sender_id, keyword	schedule, date, content_type, match, tz
GET	/api/keyword-posts/retrieve	sender_id	keyword, author_name, limit, offset
GET	/api/keyword-posts/status	sender_id	limit, offset
POST	/api/keyword-posts/update-schedule	sender_id, kp_id, schedule_time	—
POST	/api/keyword-posts/status-change	sender_id, kp_id, status	—
DEL	/api/keyword-posts/delete	sender_id, kp_id (query)	—

Social Signal examples

```
# Create a Social Signal and kick off a scrape
curl -X POST -H "X-API-KEY: your_key" -H "Content-Type: application/json" \
  -d '{"linkedin_sender_id":"abc123","keyword":"AI sales","schedule_time":"daily"}' \
  https://mcp.warmgtm.com/api/keyword-posts/fetch

# Retrieve scraped posts
curl -H "X-API-KEY: your_key" \
  "https://mcp.warmgtm.com/api/keyword-posts/retrieve?linkedin_sender_id=abc123&limit=20"

# Pause a Social Signal
curl -X POST -H "X-API-KEY: your_key" -H "Content-Type: application/json" \
  -d '{"linkedin_sender_id":"abc123","linkedin_keyword_post_id":"kp_42","status":"paused"}' \
  https://mcp.warmgtm.com/api/keyword-posts/status-change
```

ISSUE	SOLUTION
"Missing API key" error	Ensure <code>X-API-KEY</code> header is set correctly, or the OAuth flow is fully completed for Claude.ai.
Tools not appearing	Restart the client after updating config. Verify the URL is <code>https://mcp.warmgtm.com/mcp</code> .
ChatGPT actions fail	Re-import the OpenAPI spec and confirm API key auth is set to Custom header <code>X-API-KEY</code> .
Rate limit on <code>/api</code>	120 req/min per IP, wait a moment and retry.
Rate limit on <code>/mcp</code>	60 req/min per IP, wait a moment and retry.
OAuth redirect fails (Claude.ai)	Make sure you're using <code>https://mcp.warmgtm.com/mcp</code> (not <code>/api</code>).

Need help?

 API docs, mcp.warmgtm.com/docs  OpenAPI spec, mcp.warmgtm.com/openapi.json

 Dashboard, app.warmgtm.com