Google Workspace CLI Playbook

Control Gmail, Drive, Calendar, Docs & Sheets from your terminal

PREPARED BY C&C STRATEGIC CONSULTING — 2026

OVERVIEW

What Is the Google Workspace CLI?

A free, open-source command-line tool that gives you (and AI agents like Claude Code) full API access to your entire Google Workspace from the terminal.

The Problem

You're running your business across 10–15 browser tabs. Gmail, Drive, Calendar, Docs, Sheets — all fighting for screen space. Every tab switch breaks your focus. Every manual action is time you're not spending on revenue.

The Solution

One CLI tool. One terminal window. Claude Code sends emails, checks your calendar, browses your Drive, reads Docs, uploads files — all without opening Chrome. 20 minutes to set up. Works forever.

SUPPORTED SERVICES

Gmail

Send, read, search, and manage emails directly from the terminal

Google Drive

List, upload, download, and organize files and folders

Google Calendar

View events, create meetings, check availability

Google Docs

Read document content, create and edit docs

Google Sheets

Create spreadsheets, read/write cell data, manage ranges

40+ More APIs

Slides, Tasks, Chat, Meet, Admin — every Google Workspace API

Open Source & Free: github.com/googleworkspace/cli — no subscription, no API costs beyond Google's free tier. Install with one command.

SETUP

Installation & Authentication

Step 1: Install the CLI

```
# Install via npm (recommended)
npm install -g @googleworkspace/cli

# Verify installation
gws --version
```

Step 2: Quick Auth Setup

If you have gcloud CLI installed, this handles everything automatically:

```
# Creates GCP project, enables APIs, authenticates
gws auth setup
```

Step 3: Manual Setup (if needed)

1. **Google Cloud Console** — Create a new project at console.cloud.google.com
2. **Enable APIs** — Turn on Drive, Gmail, Calendar, Docs, Sheets (and any others you want)
3. **OAuth Consent Screen** — Set to External, add your email as a test user
4. **Create Credentials** — OAuth 2.0 Client ID, Desktop app type. Download the JSON.
5. **Save Client Secret** — Move to `~/.config/gws/client_secret.json`
6. **Login** — Run `gws auth login`, authorize in browser, done.

> **Pro Tip:** If you hit a keyring encryption error in Claude Code or subshells, export to plaintext credentials:
> `gws auth export --unmasked > ~/.config/gws/credentials.json`
> Then remove the encrypted file: `rm ~/.config/gws/credentials.enc`

COMMANDS

Command Cheat Sheet

Google Drive

```
# List files
gws drive files list --params '{"pageSize": 10}'

# List files in a specific folder
gws drive files list --params '{"q": "\"FOLDER_ID\" in parents"}'

# Upload a file
gws drive files create --json '{"name": "report.pdf", "parents": ["FOLDER_ID"]}' --upload
./report.pdf

# Delete a file
gws drive files delete --params '{"fileId": "FILE_ID"}'
```

Gmail

```
# List recent messages
gws gmail users messages list --params '{"userId": "me"}'

# Send an email (base64-encoded raw message)
RAW=$(echo -e "To: recipient@email.com\nSubject: Hello\n\nBody text" | base64)
gws gmail users messages send --params '{"userId": "me"}' --json "{\"raw\": \"$RAW\"}"
```

Google Calendar

```
# List today's events
gws calendar events list --params '{"calendarId": "primary", "timeMin": "2026-03-10T00:0
0:00Z", "timeMax": "2026-03-10T23:59:59Z", "singleEvents": true}'

# List this week's events
gws calendar events list --params '{"calendarId": "primary", "timeMin": "2026-03-10T00:0
0:00Z", "timeMax": "2026-03-15T23:59:59Z", "singleEvents": true, "orderBy": "startTime"}'
```

COMMANDS (CONTINUED)

More Services

Google Docs

```
# Read a document's content
gws docs documents get --params '{"documentId": "DOC_ID"}'
```

Google Sheets

```
# Create a new spreadsheet
gws sheets spreadsheets create --json '{"properties": {"title": "Q1 Budget"}}'

# Read a range of cells
gws sheets spreadsheets values get --params '{"spreadsheetId": "SHEET_ID", "range": "Shee
t1!A1:C10"}'
```

Useful Flags

FLAG	WHAT IT DOES
`--params '{...}'`	Query parameters (filters, IDs, pagination)
`--json '{...}'`	Request body for create/update operations
`--upload ./file`	Attach a local file for multipart upload
`--page-all`	Auto-paginate all results (NDJSON output)
`--page-limit N`	Max pages to fetch (default: 10)
`--dry-run`	Preview request without executing

Discover Any API Command

```
# See the schema for any endpoint
gws schema drive.files.list

# Browse all available services
gws schema
```

The CLI reads from Google's Discovery Service at runtime. Every Google Workspace API endpoint is available automatically. If Google adds a new API, the CLI picks it up without an update.

AI INTEGRATION

Using with Claude Code

Once authenticated, Claude Code can use `gws` commands directly.
No extra configuration needed. Just ask it to do things.

Real Examples (Things You Can Say)

"Check my calendar for today"

Claude Code runs the calendar events list command, parses the JSON, and gives you a clean summary of your meetings with times, locations, and links.

"Send Chris an email about the project update"

Claude Code composes the email, formats it, base64-encodes it, and sends it via the Gmail API. It shows up in your Sent folder like any other email.

"Show me what's in the Marketing folder on Drive"

Claude Code searches for the folder, gets its ID, then lists all files and subfolders inside it. You can drill down into any folder from there.

"Read the company logins doc"

Claude Code fetches the document content via the Docs API and displays it in readable format right in your terminal.

"Upload this report to the client's Drive folder"

Claude Code takes any local file and uploads it to a specific Google Drive folder using the multipart upload endpoint.

The Key Insight: Every Google Workspace action becomes something Claude Code can chain together. Calendar + Email + Drive + Docs = an AI-powered command center that runs inside your business.

REFERENCE

Environment Variables & Troubleshooting

Environment Variables

VARIABLE	PURPOSE
`GOOGLE_WORKSPACE_CLI_TOKEN`	Pre-obtained access token (highest priority)
`GOOGLE_WORKSPACE_CLI_CREDENTIALS_FILE`	Path to credentials JSON file
`GOOGLE_WORKSPACE_CLI_CONFIG_DIR`	Override config directory (default: ~/.config/gws)
`GOOGLE_WORKSPACE_PROJECT_ID`	GCP project ID override

Auth Priority Order

1. **GOOGLE_WORKSPACE_CLI_TOKEN** — env var access token
2. **credentials.enc** — encrypted credentials (OS keyring)
3. **credentials.json** — plaintext credentials file
4. **service-account.json** — service account key

Common Issues

Keyring Decryption Error

Subshells and AI tools often can't access the macOS Keychain. Fix: export to plaintext JSON and remove the .enc file.

401 Unauthorized

Your credentials.json may have the wrong format. It needs: type, client_id, client_secret, and refresh_token. Raw Google token responses won't work.

"Access Blocked" on Login

Your Google account isn't added as a test user. Go to Google Cloud Console, OAuth consent screen, and add your email under Test Users.

> **The refresh token never expires** (unless you revoke it, are inactive for 6+ months, or accumulate 50+ tokens). Once set up, this is truly set-and-forget.

Ready to Build Your AI Command Center?

Stop switching between 15 browser tabs.
Let Claude Code run your Google Workspace.

Join the community for tutorials, workflows, and support

start.ccstrategic.io/skool

YOUTUBE	WEBSITE	INSTAGRAM
@charlieautomates	ccstrategic.io	@charlieautomates