



ERGOWARE AIR

MCP Google Workspace Server

Technical Manual v1.0

Complete guide to installation, configuration and standalone use from Python scripts on Windows.

Contents

- 01 System overview
- 02 Requirements and installation
- 03 Google credentials configuration
- 04 Configuring the .env file
- 05 First run and OAuth2 authorization
- 06 Using tools from Python scripts
- 07 Complete tool reference
- 08 Common errors and solutions
- 09 Rate limits and security notes

1. System overview

- The MCP Google Workspace Server exposes a controlled set of tools to interact with Google Drive, Google Docs and Google Sheets from local Python scripts.
- The goal is not to replace the user interface, but to create a stable bridge between local automation, AI agents and Google Workspace resources.
- The server can be used by a standalone Python script, a dashboard, or an agent that needs to search files, read documents, create sheets, update cells or retrieve structured data.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



2. Requirements and installation

- Windows 10 or Windows 11 with Python 3.10 or later installed and available from the terminal.
- A project folder containing the MCP server files, requirements file and configuration templates.
- A Python virtual environment is recommended to keep dependencies isolated from the rest of the computer.
- Install packages with: `python -m pip install -r requirements.txt`

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



3. Google credentials configuration

- Create or open a Google Cloud project and enable the APIs needed for Drive, Docs and Sheets.
- Create OAuth client credentials for a desktop application and download the credentials JSON file.
- Place the credentials file inside the server configuration folder and keep it private. Do not publish it in a repository or send it by email.
- Use the Google account that has access to the Drive, Docs and Sheets resources you want the tools to manage.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



4. Configuring the .env file

- The .env file stores local configuration values such as paths, credentials file name, token location and default folders.
- Keep secrets outside public folders. If the project is deployed on a server, use a private path not reachable from the web.
- Typical values include `GOOGLE_CLIENT_SECRET_FILE`, `GOOGLE_TOKEN_FILE`, `MCP_HOST`, `MCP_PORT` and optional default Drive folder identifiers.

Operational note: this manual describes the English standalone package for `ergowareltd.com`. Keep credentials and tokens outside public directories.



5. First run and OAuth2 authorization

- Run the server or the authorization script from the configured virtual environment.
- At the first execution, Google opens an authorization flow in the browser. Confirm access with the correct Google account.
- After approval, a local token file is generated. Future executions reuse the token until it expires or is revoked.
- If authorization fails, delete the token file and repeat the OAuth process.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



6. Using tools from Python scripts

- A Python script can connect to the MCP server, list available tools and call them with structured JSON arguments.
- For Drive searches, use clear filename-based queries when possible. For Sheets operations, always specify spreadsheet ID, sheet name, cell range and values.
- For AI agents, keep the tool call explicit: the model should decide what to request, while the script validates arguments and handles errors.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



7. Complete tool reference

- Drive tools: search files, read metadata, list folders and retrieve links.
- Docs tools: read document text, create documents and append or update structured content.
- Sheets tools: read ranges, write values, create spreadsheets and update worksheets.
- Utility tools: health check, configuration inspection and basic diagnostic calls.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



8. Common errors and solutions

- Invalid credentials: verify the OAuth client JSON path and regenerate credentials if needed.
- Access denied: check that the Google account has permission on the target file or folder.
- Token expired or corrupted: delete the token file and repeat the authorization flow.
- Missing dependency: activate the virtual environment and reinstall requirements.
- Wrong file ID or range: copy the ID from the Google URL and verify the sheet name.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.



9. Rate limits and security notes

- Avoid unnecessary polling and batch operations where possible to reduce API usage.
- Use the minimum permissions required for the tools you actually need.
- Never expose credentials, tokens or private configuration files through public web folders.
- For production use, add logs, backup procedures, permission rules and clear responsibility for human review.

Operational note: this manual describes the English standalone package for ergowareltd.com. Keep credentials and tokens outside public directories.