

User guide

Start with your first workspace. Keep your terminals, Agents and project progress together.

For V0.8.1 · Updated 2026-09-07 · Windows 10/11 x64

01 / Install and get started

Termexo is a Windows workbench for real terminals and coding Agents. No Termexo account is required. To use an Agent, you still need its CLI and a signed-in account or a model provider's API key.

Choose an installation method

- **Download the installer:** open the [latest GitHub release](#), download the Windows x64 installer from Assets, and run it. The desktop app requires WebView2. Installer users do not need Rust or the source development toolchain.
- **Run through npm:** with Node.js 18.18 or later installed, run the following command in Windows PowerShell. The npm package includes the desktop executable; WebView2 is still required.

```
npx termexo@latest
```

For a global command, run `npm install --global termexo`, then launch `termexo`. Run the installation command again to update a global installation.

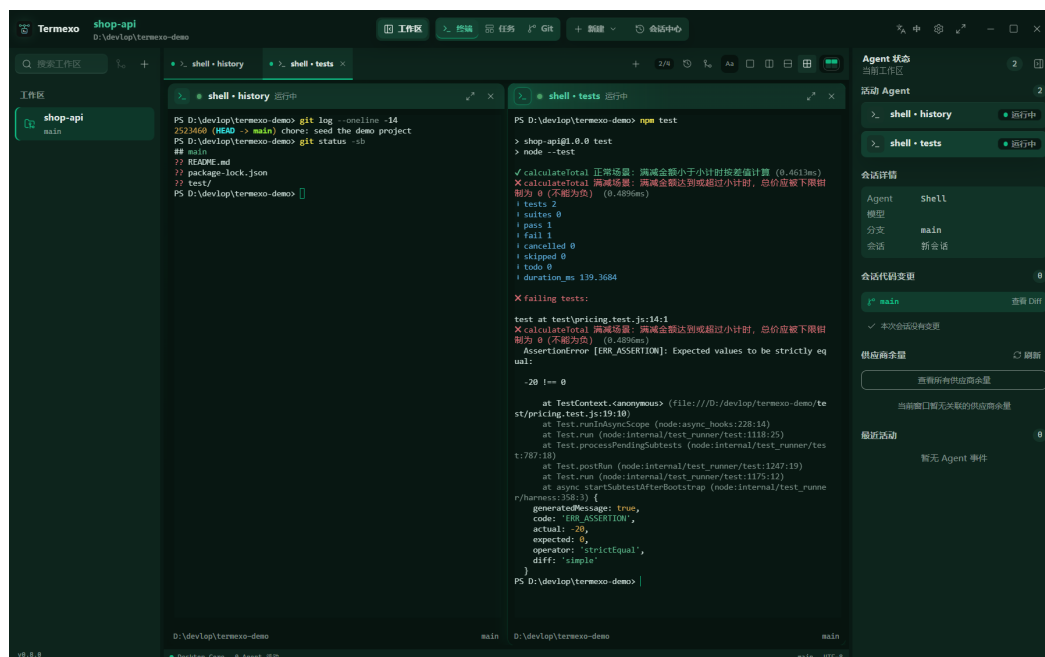
Your first session in three steps

1. **Create a workspace.** Use the first-run guide to create a workspace. Enter a project name and select a project folder that actually exists on this computer.
2. **Prepare an Agent.** Check the detection status of Claude Code, Codex or OpenCode in Settings. If needed, use the CLI install/update entry, review the installation preview and confirm. Complete the selected Agent's sign-in or model configuration.
3. **Open a terminal.** Click New on the tab bar, select an Agent, and confirm the working folder, account and model. Start it and enter a task. You can also open a regular terminal first to check the project path and command environment.

Start with a small check, such as: "Read this project and explain how to run its tests. Do not change any files yet." Confirm the folder, account and connection before assigning code changes.

02 / Workspaces and terminals

A workspace groups a project folder, terminal tabs and a layout. Create separate workspaces for different projects, then rename, color-code and reorder them. Switching projects does not require rearranging every terminal.



Workspaces organize projects; tabs organize terminals; the grid controls which terminals you see. Screenshot: Chinese interface.

Arrange your workbench

- Use New on the tab bar to start a regular shell, Claude Code, Codex or OpenCode. The working folder is a local directory, not a GitHub web address.
- Open several tabs, select the terminals to display, and adjust the grid. Choose 1 to 6 rows/columns or maximize a terminal. Drag tabs to reorder them.
- Each terminal runs a separate real process. Agents editing the same folder can affect each other's work. Use separate directories or Git worktrees when changes need isolation.

Understand status alerts

- **Running / thinking:** the Agent is processing a task or using tools.
- **Waiting for input / approval:** answer a question or authorize an operation. Click the persistent alert to reach that terminal, then read the full request.
- **Completed / failed / stopped:** inspect terminal output and file changes before accepting, troubleshooting or restarting. A status badge does not prove the code passed validation.

Automatic confirmation reduces approval prompts and expands what an Agent may execute without asking each time. Keep manual confirmation for unfamiliar or sensitive projects. An AUTO badge is not a security review.

03 / Resume sessions

1. Open the Session Center and search for local Claude Code, Codex or OpenCode sessions. Narrow the list by project, account or other available filters.
2. Check the folder, Agent, account and recent activity to identify the session you want.
3. Select Resume and confirm the launch options. Termexo uses the CLI's native resume capability, then lets you continue in a terminal.

Termexo discovers native session records read-only; it does not rewrite history to simulate recovery. Resuming context depends on the original session files, account, CLI version and project path still being available.

Restoring a workspace layout does not bring an exited process back to life. After closing the app or restarting Windows, unsaved work in a regular shell cannot be recovered from the layout. Use native session recovery for an Agent's saved context.

04 / Accounts, models and network

Sign-in accounts

Manage isolated sign-in accounts in Settings and follow the corresponding CLI's authorization flow. Select an account when launching a terminal; its header shows the current account. Switching accounts there restarts the terminal with a new session, so save or hand off current work first.

Model profiles

1. Open model settings and create a profile with a name, model identifier, endpoint and API key. Use the actual values provided by your supplier; documentation examples are not credentials.
2. Check that the protocol matches the selected Agent. A compatible Claude Code backend must support the relevant Anthropic API. An arbitrary model URL is not necessarily usable.
3. Save the profile, select it when starting or resuming a terminal, and send a short test request. Before switching an existing terminal, read the app's notice and make sure the current task can be interrupted.

API keys are stored in Windows Credential Manager. Not every provider supports balance or quota queries. Values marked as estimates or unavailable are not official billing data.

Proxies and npm networking

If you need a proxy, create a global or workspace network profile. Enter the real proxy and npm registry settings, test connectivity, then start terminals or install CLIs. Workspace settings take precedence over global settings. Restart affected terminals after changing the profile so they receive the new environment.

05 / Tasks, Git and handoffs

Assign a task to an Agent

1. Open the task board and create a task with a clear goal, project folder, priority and acceptance criteria. For example: "Add a search box; an empty query shows all items; tests pass."
2. Select an Agent and model, then start execution. The task is linked to a real terminal and the app returns to the terminal view, where you can check startup and handle approvals.
3. Track the task through stages such as To do, Executing, Completed and Verified. When the Agent finishes, check its output against your criteria. Give follow-up feedback if needed; verify the task only after it passes your review.

Review Git changes

Select the relevant terminal and open the Git view to inspect branches, commits and file diffs, in a single-column or side-by-side layout. The Git view follows the active terminal's working folder and uses the HEAD recorded when that terminal started as the session baseline.

Session changes are repository differences observed during that period, not an exclusive record of this Agent's actions. Edits from an editor, other terminals or other Agents may also appear. Review each change before committing.

Reuse prompts and hand off context

- Prompt assets let you save, search, favorite and reuse submitted prompts. Drafts are restored per terminal. Check paths, accounts and task context before reusing a prompt.
- Create a terminal- or workspace-scoped handoff that gathers tasks, prompts, terminal output and Git information within a chosen token budget.
- Preview the package and check its goal, completed work, validation results, risks and next steps. Then export Markdown/JSON or pass it to another Agent to continue.

Automatic redaction cannot guarantee removal of every secret. Before sharing, manually check for API keys, access tokens, internal addresses, customer data and source-code access restrictions. A handoff is a context document, not a lossless conversion of a native session file.

06 / Mobile and remote access

Remote access is off by default. When enabled, a phone, tablet or another computer can operate the desktop's existing workspaces and terminals in a browser. You do not need to install the Agents on your phone.

Enable it on the desktop

1. Open Remote Access in Settings and turn on the main switch. Follow the UI to configure and save the listening address and port. The default port is 7420.
2. For devices on your LAN, choose a reachable LAN listening address or listen on all interfaces. Listening only on 127.0.0.1 limits access to the host itself. Keep HTTPS enabled.
3. Confirm that the service is running. Select a reachable computer address in the address picker, then copy the generated link or display its QR code for your own device to scan.
4. Connect your phone to the same LAN or a VPN that can reach the computer. Open the link in a browser and use the access token as prompted. Any working folder you enter refers to a path on the computer running the desktop app.

HTTPS uses a self-signed certificate by default, so the browser may warn that it is not trusted on first access. Confirm that this is your own computer, the correct address and a trusted network before handling the warning according to your browser's rules. Do not ignore certificate warnings on unrelated websites.

If you cannot connect, check in this order

- Is the desktop app still running? Is the computer asleep? Does the remote service show an error?
- Does the link use the actual LAN IP? 0.0.0.0 is a listening setting, not the address to open on a phone. On a phone, localhost refers to the phone itself.
- Can the devices reach each other? Guest Wi-Fi may isolate clients. Check that Windows Firewall permits the selected port on the trusted network; do not disable the entire firewall.
- Has the port, IP address or token changed? Does the URL's protocol match the HTTPS setting?

Limits and disconnecting

The terminal size follows the client currently in use. A narrow phone view shows one terminal, and other clients may need to pan horizontally. Input from both devices affects the same process; avoid typing from both at once.

The remote page does not expose all desktop file import/export or app self-update features. When finished, turn off remote access in desktop Settings. Rotating the access token invalidates the old token and disconnects existing clients.

The token and QR code are remote-control credentials. Anyone holding them may operate terminals and affect files on the host. Never post them in public chats, Issues or screenshots. Do not expose the service directly to the public internet; use a trusted LAN or controlled VPN.

07 / Frequently asked questions

An Agent is not detected, or its command will not start.

Check its installation status in Settings, then confirm that the CLI can start in a system terminal. Run detection again after installation. Check PATH, Node/npm, the working folder and network settings. For sign-in failures, confirm that the selected isolated account has completed authorization.

History is missing, or resuming reports an invalid session.

Check that you are using the original account and project folder, that native session files still exist, and that the CLI can read them. Changing accounts, deleting CLI data or moving a project can affect recovery. Do not manually edit native session IDs. If necessary, start a new session and continue with a reviewed handoff package.

A model request fails with an authentication or API error.

Check the selected profile, model name, API key, and endpoint protocol and path. Confirm that the provider supports the interfaces required by the CLI. Then check balance and proxy settings. Keep error codes for troubleshooting, but do not publish full request headers or secrets.

I can see the terminal on my phone, but cannot copy or paste.

Browser clipboard access depends on a secure context and permissions. Use HTTPS and grant permission when prompted. If it remains unavailable, select text to copy or paste manually. Drag with a finger to scroll the terminal on a narrow screen.

How do I update Termexo?

Installer users can download from the [latest release page](#). npm users can run `npx termexo@latest` or update the global installation. Save files and finish or hand off running tasks before updating. Updating Termexo and updating an Agent CLI are separate operations; upgrade CLIs in their own management entry.

08 / Data and security

- Termexo works locally by default, with no Termexo account or mandatory cloud sync. Workspace state stays on the host, and API keys are stored in Windows Credential Manager.
- Agent CLIs still send requests to the model services you choose. Which code and context leave the computer depends on the CLI, provider configuration and privacy policy.
- The public website has separate third-party visit statistics. These do not track desktop or remote-workbench sessions and are not download counts.

Report problems through [GitHub Issues](#). Include your Termexo, Windows and CLI versions, reproduction steps, expected and actual behavior, and redacted error details. Hide accounts, keys, tokens, QR codes and sensitive paths before submitting.

Latest online guide: <https://www.termexo.com/guide.en.html>

Release history: [GitHub Releases](#)