



Pressable MCP

Guide for WooCommerce Stores

Setup instructions, tested workflows, and ready-to-use prompts for WooCommerce store owners on Pressable MCP.

Claude Desktop · Claude Code · Claude (Web) · ChatGPT · Codex
Gemini CLI · Cursor · Windsurf · VS Code + Copilot

 pressable.com/mcp ↗

What's inside

This guide covers six operational workflows for WooCommerce store owners, with setup instructions for every supported AI client and tested prompts you can use the same day.

SETUP

Connect your AI client

8 clients covered · OAuth + bearer token

WORKFLOW 01

Pre-launch readiness check

Before any sale or product launch

WORKFLOW 02

Backup and recovery

Before any significant change

WORKFLOW 03

Post-incident diagnosis

After something breaks

WORKFLOW 04

Security and plugin health

Ongoing store protection

WORKFLOW 05

Staging and deployment

Before pushing changes live

WORKFLOW 06

Monthly store health check

Ongoing visibility

APPENDIX

Operating rhythm

A suggested cadence for any store

APPENDIX

Agency case studies

OWUP · Inspry

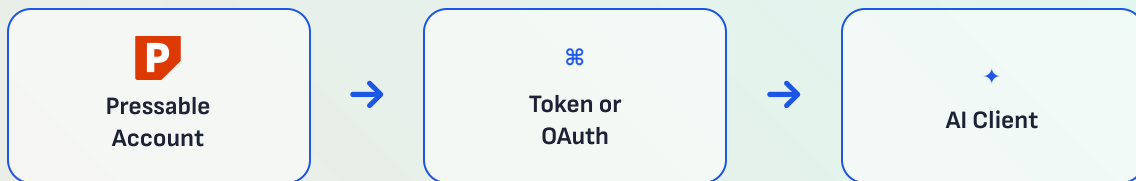
Prompts are tested and ready to use the same day.

Connect your AI client.

Two auth methods: bearer token for CLI and desktop clients, OAuth for browser-based clients. Start with Step 1 regardless of which client you use.

STEP 1 — GENERATE AN ACCESS TOKEN

Log in to my.pressable.com, go to Tools, then Pressable MCP, and generate a new access token. Copy it somewhere safe before leaving the page. Treat it like a password — anyone with the token can manage your sites. Claude Code and ChatGPT (browser) use OAuth instead and skip this step, but every other client needs this token.



Claude Desktop

Bearer token · Requires Node.js

1. Generate a token in MyPressable (Step 1).
2. Install Node.js from nodejs.org if needed.
3. Open `~/Library/Application Support/Claude/claude_desktop_config.json` (Mac) or `%APPDATA%\Claude\claude_desktop_config.json` (Windows).
4. Add the server config below, then fully quit and reopen Claude Desktop.

```
{ "mcpServers": { "pressable": { "command": "npx", "args": ["mcp-remote", "https://mcp.pressable.com", "--transport", "http-only", "--header", "Authorization:Bearer YOUR_TOKEN"] }}}}
```

Claude Code

OAuth · No token needed · Requires Node.js

1. Install Claude Code: `npm install -g @anthropic-ai/claude-code`
2. Add the server: `claude mcp add --transport http pressable https://mcp.pressable.com`
3. Run `claude` in your terminal, then type `/mcp`.
4. Click the Pressable entry, then Authenticate, and sign in via browser.

Claude (Web)

OAuth · No token needed · Browser only

1. Open Claude, go to Customize, then Connectors, and click the +. Choose Add custom connector.
2. (Team/Enterprise: an Owner or Primary Owner must first enable custom connectors under Organization Settings, then Connectors.)
3. Enter `https://mcp.pressable.com` and click Add. Leave the OAuth Client ID and Secret fields blank.
4. Sign in with your Pressable account and click Authorize to grant access.
5. To use it in a conversation, click the + in the message bar, choose Connectors, and toggle Pressable on.

ChatGPT (Browser)

OAuth · No token needed · Developer Mode required

1. Log in at `chatgpt.com`, go to Settings, then Apps, then Advanced, and enable Developer Mode.
2. Click Create App and enter `https://mcp.pressable.com` as the server URL.
3. Follow the OAuth prompt to authorize your Pressable account.
4. Ask ChatGPT to list your sites to confirm it is working.

MCP Servers is not available on all ChatGPT accounts yet.

Gemini CLI

Bearer token · Requires Node.js

1. Generate a token in MyPressable (Step 1).
2. Install: `npm install -g @google/gemini-cli`
3. Run: `gemini mcp add pressable https://mcp.pressable.com --transport sse`
4. Open `~/.gemini/settings.json` and add your token to the Authorization header. Reopen Gemini CLI to connect.

Gemini in the browser does not support MCP. Use the CLI.

Codex

Bearer token · Requires environment variable

1. Generate a token in MyPressable (Step 1).
2. Set an environment variable named `PRESSABLE_MCP_TOKEN` with your token (Mac: `launchctl setenv`, Windows: `setx`, Linux: add to `~/.config/environment.d/pressable-mcp.conf`).
3. In the Codex app, go to Settings, then Settings, then MCP servers, then click Add server.
4. Enter the Pressable MCP server URL and save.

Cursor / Windsurf / VS Code + Copilot

Bearer token · MCP settings or config file

1. Generate a token in MyPressable (Step 1).
2. In Cursor: Settings, then MCP, then add a new server with command `npx` and args pointing to `mcp-remote https://mcp.pressable.com` with your token as an Authorization header.
3. In Windsurf and VS Code + Copilot: add the same server block to your MCP config file.
4. Full config examples:
`pressable.com/knowledgebase/get-started-with-pressable-mcp/`

Your first 10 minutes

Run these right after you connect

1. List your Pressable sites and confirm your store appears.
2. Check the PHP version on [store] and flag if it needs to be updated.
3. Pull backup status for your store and confirm a backup ran in the last 24 hours.

WORKFLOW 01

01

Pre-launch readiness check

BEFORE ANY SALE OR LAUNCH

Sales events, product launches, and seasonal promotions send traffic spikes your store may not be ready for. This workflow runs a complete infrastructure check before any high-traffic event — cache, PHP, backups, plugin security, and change freeze — so you know exactly what to fix before customers arrive.

PRE-SALE READINESS — run 24–48 hours before any traffic event

Run a pre-launch readiness check on [store]. Confirm: edge cache is enabled, PHP version is 8.2 or higher, a backup completed in the last 24 hours, and no plugin or theme changes in the last 7 days. Also flag any plugins with known security vulnerabilities. Give me a prioritized list of anything to fix before go-live.

HIGH-TRAFFIC BASELINE — read-only snapshot before a sale

For [store], run a high-traffic readiness check. This is READ-ONLY. Report: site active status, PHP version, whether edge cache is enabled, and whether a scheduled backup completed in the last 24 hours. Flag anything that needs attention before traffic arrives.

CACHE CONFIRMATION — confirm edge cache is protecting the store

Check edge cache status on [store]. Is it enabled? Is defensive mode active? If a traffic spike is expected, enable defensive mode to aggressively cache and protect the store.

WHAT YOU GET

- ✓ Go/no-go readiness summary with prioritized fix list.
- ✓ PHP version, cache status, and backup confirmation.
- ✓ Plugin vulnerability flags surfaced before traffic arrives.
- ✓ Change freeze verification covering the last 7 days.

More prompts:

"Flush edge cache on [store] after I push my sale banner."

"Enable defensive cache mode on [store] to absorb a traffic spike."

"Create an on-demand backup of [store] right now before my sale goes live."

Backup and recovery

BEFORE ANY SIGNIFICANT CHANGE

WooCommerce stores can't afford to lose order data or product catalogs. These prompts let you create, verify, and restore backups without touching the Pressable dashboard — and give you a download link you can hold before making any risky change.

BACKUP BEFORE A MAJOR CHANGE — always run this first

Create an on-demand filesystem and database backup for [store] right now. Confirm both completed successfully and give me the download links.

BACKUP STATUS CHECK — confirm backup health

Show me all backups for [store] from the last 7 days. Was every scheduled backup successful? Give me the download link for the most recent filesystem and database backup.

RESTORE FROM BACKUP — after a bad deployment or data loss

Restore [store] from the most recent filesystem and database backups. Tell me which backup you're using before you start, and confirm when the restore is complete.

DATABASE-ONLY RESTORE — when only the database was affected

Restore only the database on [store] from the most recent database backup. Do not touch the filesystem. Confirm when done.

WHAT YOU GET

- ✓ On-demand backup created before any risky change, with download links.
- ✓ Backup history reviewed so gaps don't catch you off-guard.
- ✓ Full or database-only restore without dashboard access.
- ✓ Confirmation at every step before anything is overwritten.

More prompts:

"List all on-demand backups I've created for [store] and their expiry dates."

"Delete the oldest on-demand backup on [store] to free up a slot."

"Sync the database from [staging site] to [store] after I've verified changes on staging."

Post-incident diagnosis

AFTER SOMETHING BREAKS

When a store goes down or something stops working, the first question is always: what changed? These prompts pull the site activity log, backup history, and plugin state to help you reconstruct what happened and when — without needing SSH access or hours of manual dashboard review.

POST-INCIDENT AUDIT — after a store goes down or checkout breaks

Show me the site activity log for [store]. Summarize any plugin installs, updates, or deletions, theme changes, configuration changes, and any collaborator actions. I need to understand what changed before the incident started.

DATABASE AND BACKUP REVIEW — after data or sync issues

Show me all backups for [store] from the last 7 days. Which is the most recent successful backup? Was there any backup failure in that window? Give me the download link for the most recent filesystem and database backup.

PLUGIN CONFLICT INVESTIGATION — when a recent update caused a problem

List all plugins on [store] with their current versions and update availability. Flag any with known security vulnerabilities. Also show me the site activity log filtered to plugin-related actions only so I can see what changed recently.

WHAT YOU GET

- ✓ Chronological activity log showing every change made before the incident.
- ✓ Most recent 50 activity entries from the last 30 days, ready to review.
- ✓ Backup status and download links for rollback if needed.
- ✓ Plugin version list with update availability and vulnerability flags.

More prompts:

"Restore [store] from the most recent filesystem and database backups."
"Create an on-demand backup of [store] right now before I make changes."
"Show me the Pressable account activity log for the last 30 days."

Security and plugin health

ONGOING STORE PROTECTION

WooCommerce stores are high-value targets. Outdated plugins, unsupported PHP versions, and unresolved vulnerabilities are the most common entry points. These prompts give you a full security picture without needing a security plugin subscription or a developer on retainer.

FULL SECURITY SCAN — quarterly or after any security concern

Run a security check on [store]. List all plugins and themes with known vulnerabilities and their severity levels. Check PHP version, verify edge cache is enabled, and show me the last 10 site activity log entries. Give me a prioritized list of anything to fix.

PLUGIN VULNERABILITY ALERT — run after any plugin update notification

Show me all plugins on [store] with known security vulnerabilities. Include the plugin name, current version, severity level, and whether an update is available. Flag anything rated high or critical.

PHP VERSION AUDIT — to stay on supported versions

Check the PHP version on [store]. Flag if it is below 8.2. Recommend upgrading to the latest supported version if needed.

FIREWALL AND ACCESS AUDIT — for stores handling payment or sensitive data

List all egress firewall rules on [store] and flag any pointing to unexpected external IPs. Also list all SFTP accounts on the site with their usernames and last accessed times. Show me any collaborator-related entries from the site activity log.

WHAT YOU GET

- ✓ Prioritized vulnerability list with severity levels and available updates.
- ✓ PHP version status with an upgrade recommendation.
- ✓ Cache, firewall, and SFTP access summary.
- ✓ Recent site activity log for human review.

More prompts:

"Refresh firewall rules on [store] to re-sync with the infrastructure."
"List all SFTP accounts on [store] and show me when each was last accessed."
"Check Jetpack status on [store] and confirm the license is active."

Staging and deployment

BEFORE PUSHING CHANGES LIVE

WooCommerce stores cannot afford downtime during a deployment. A staging environment that mirrors production, a verified backup before any change, and an activity-log review before go-live are the difference between a smooth launch and a recovery situation.

STAGING SETUP — before any significant change to a live store

Create a new WordPress site on Pressable called [staging name]. Once it's active, sync the filesystem and database from [store] to the new staging site. Then disable edge cache on the staging site and give me the staging URL.

PRE-DEPLOYMENT CHECKLIST — before pushing staging changes to production

Run a pre-deployment check on [store]. Confirm: a backup completed in the last 24 hours, edge cache is enabled, PHP version is current, and no plugins have known security vulnerabilities. Also show me the site activity log to confirm no unexpected changes were made recently.

POST-DEPLOYMENT VERIFICATION — after pushing changes live

Check [store] after the deployment I just completed. Review the site activity log for any changes in the last hour. Confirm edge cache is still enabled and the site is active. Flag any plugins with known vulnerabilities that may have been introduced.

BACKUP BEFORE MAJOR CHANGE — always run before anything significant

Create an on-demand filesystem and database backup for [store] right now. Confirm both completed successfully and give me the download links.

WHAT YOU GET

- ✓ Staging site created and synced from production, with cache disabled.
- ✓ Pre-deployment checklist with backup, cache, PHP, and vulnerability confirmation.
- ✓ Post-deployment activity-log review showing exactly what changed.
- ✓ On-demand backup confirmation with download links.

More prompts:

"Create an on-demand backup of [store] before I start my deployment."

"Purge edge cache on [store] after my deployment."

"Sync the database from [staging site] to [store] once I've verified everything on staging."

Monthly store health check

ONGOING VISIBILITY

Most store owners only look at their hosting when something is wrong. A monthly check-in catches slow degradation before it becomes a problem: plugin drift, unpatched vulnerabilities, backup gaps, PHP version age, and access creep. These prompts give you a complete picture in one session.

FULL MONTHLY HEALTH REPORT — run at the end of every month

Generate a monthly health report for [store] covering: backup status and any failures in the last 30 days, all plugins with updates available this month, any plugins with known security vulnerabilities and their severity, PHP version status, and the site activity log for the last 30 days summarized by action type. Give me a recommended action list for next month.

MONTHLY ACCESS AUDIT — review who can reach the store

Audit access on [store] for this month. List all collaborators and their assigned roles. List all SFTP accounts and when each was last accessed. Flag any accounts that look unfamiliar or haven't been used in 90 days and recommend whether they should be removed.

PLUGIN AND THEME AUDIT — to catch update drift

List all plugins on [store]. Flag any with updates available, any with known security vulnerabilities, and any that are currently inactive and could be removed. Prioritize anything rated high or critical severity.

WHAT YOU GET

- ✓ Monthly report with backup health, plugin updates, and vulnerability summary.
- ✓ Access audit covering collaborators and SFTP accounts.
- ✓ Plugin and theme audit with update flags and vulnerability count.
- ✓ Recommended actions list ready to act on.

More prompts:

"List all plugin updates completed on [store] in the last 30 days."

"Check Jetpack status on [store] and confirm it's connected and licensed."

"Refresh firewall rules on [store] as part of my monthly security review."

How to run all six workflows together.

A suggested cadence for any WooCommerce store on Pressable. Adjust frequency based on your traffic and release schedule.

BEFORE ANY SALE OR LAUNCH

Run the pre-launch readiness check 24–48 hours ahead. Fix anything flagged before traffic arrives.

AFTER ANY DEPLOYMENT

Run the post-deployment verification prompt within 2 hours of pushing changes live.

WHEN SOMETHING BREAKS

Start with the post-incident audit to establish what changed. Review backup history and restore if you need to roll back.

MONTHLY

Run the full health report. Review backup health, plugin updates and vulnerabilities, PHP version, and the monthly access audit.

QUARTERLY

Run the full security scan. Update plugins with known vulnerabilities. Check PHP version against current support windows.

ONGOING

Keep an on-demand backup habit: create a backup before any significant change, and confirm the download link before proceeding.

Pressable MCP operates at the hosting layer: environments, backups, logs, caching, PHP, plugins, DNS, and infrastructure. WooCommerce order and product data lives in WordPress and is accessible via the WordPress MCP plugin, which you can enable per site under Site Configuration in MyPressable. Custom post types require `show_in_rest ⇒ true` in `register_post_type()` to appear via the WordPress REST API.

IN PRACTICE

How this is already working for WooCommerce stores.

Two agencies running Pressable MCP across their WooCommerce portfolios.

OWUP

Norway and Sweden · WooCommerce specialist · 60+ sites

Days to under an hour

resolving a legacy POS sync issue



OWUP builds WooCommerce stores for high-end watch and jewelry retailers, managing inventories of 20,000+ SKUs with constant price fluctuations and legacy POS systems. A metadata corruption issue had been causing inventory sync failures for years. Using Pressable MCP, the team accessed the database through their AI client, matched it against a POS data dump, and closed the issue in under an hour. Beyond that fix, OWUP tripled their speed optimization output and now generates automated monthly reports by connecting Pressable data to n8n and Slack.

"It's night and day. With Pressable MCP, we can move so much faster with information. It has truly made us realize we can double down on being experts in our field."

Ole Straume Andersen, Managing Director, OWUP

Full case study: pressable.com/case-studies/how-owup-uses-pressable-mcp-to-cut-debugging-time-from-days-to-minutes/

Inspiry

Atlanta, GA · WooCommerce agency · 60+ live sites · Since 2011

Full day to under 10 min

fleet-wide diagnostic checks



Inspiry has been building complex WooCommerce environments since 2011 and manages 60+ live client sites. A recurring malware infection that standard security tools kept missing was identified and resolved at the root cause using Pressable MCP. Post-update error scans that previously required a full day of manual site-by-site checking now finish in under 10 minutes. Client audit reports that used to require a separate application password per site now pull in a single step.

"AI is making it even more important to have that human touch. Having tools like the Pressable MCP allows us to communicate value more clearly, build better reports, and have more time to actually build the relationship."

Matt Schwartz, Founder, Inspiry

Full case study: pressable.com/case-studies/how-inspry-cut-a-full-day-of-site-diagnostics-down-to-minutes-with-pressable-mcp/



**Pressable MCP is
available now at no
extra cost for all
Pressable customers.**



pressable.com/mcp