

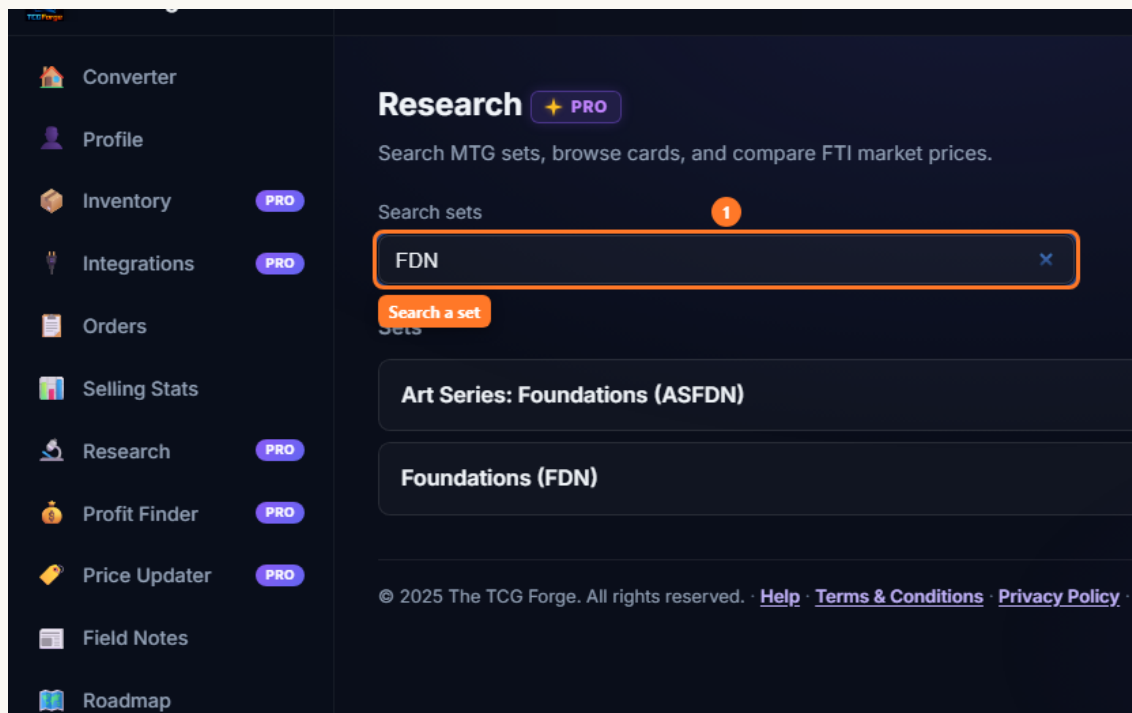
How I use AI to build The TCG Forge.

Kevin Granger

Solo founder and operator

The TCG Forge is the Seller OS I built for trading-card sellers: inventory and research, marketplace connections, orders, postage and labels, plus converter and listing tools.

I own the development process, architecture, release decisions, and overall direction of the product, while using AI to write and revise the application code. That's how I've built and operated it as a solo founder.



Research in Seller OS, with inventory, orders, integrations, and seller tools in the navigation. Feature access depends on plan and enabled capabilities.

PRODUCTION SNAPSHOT / AUGUST 30, 2026

113

registered users

61

paid accounts

75

order records

69

shipment records

Paid count includes lifetime access. Records include operator use; shipment states include failed and voided.

How sellers use it.

Sellers can work through an order, manage their stock, or prepare listings in the same application.

Orders and shipping

Import / enter order → Orders hub → Wallet + label → Pack list + tracking

Import support varies by channel. Not every order line is matched to inventory; tracking has ongoing edge-case maintenance.

Inventory to a connected channel / Pro

Import preview → Stock + locations → ManaPool / eBay → Quantity push

Native connection flows differ from file exports. Imports and quantity push are partial; updates require a supported active connection.

Listing preparation

Collection export → Converter → Channel file → Seller uploads

File templates aren't API publishing. Converter activity: 258,901 cards processed all-time as of August 30, 2026; not unique cards.

Channel	Order	Buyer	Tracking	Carrier / service	Postage	Ready	Actions
ManaPool				USPS · First-Class Mail Letter	\$0.78	7/26/2026, 2:55:15 PM	Open carrier label Open envelope Open pick list Mark
ManaPool				USPS · First-Class Mail Letter	\$0.78	7/26/2026, 2:55:14 PM	Open carrier label Open envelope Open pick list Mark

Seller OS fulfillment view. Identifying fields are redacted; orange annotations come from the in-app guide.

HOW I DELIVER IT

I define the work, AI implements, and tests and reviews support my release decisions. Review coverage varies; page 3 shows the checks for one change. I control commits, deployment, and production secrets.

909 live SDK runs / 409 automated test files / 8-tier deploy diagnostic suite

Portfolio snapshot: Aug 30, 2026. Runs exclude dry runs and aren't merges; files aren't pass counts. Default post-deploy subset: health, entitlements, production smoke.

A name wasn't enough to pack the right card.

Warehouse pack list fix / August 17, 2026

I printed a production pack list after a card-name fix. The name was right, but an order line not linked to inventory still showed Colorless and no set or collector details. A seller needs those details to pick the right card.

I set the rules: reuse the existing catalog service to resolve marketplace SKUs, show set and collector, include card type, and don't label unknown colors as Colorless. An AI backend agent implemented the change within the files I'd allowed.

I ran one unit test file, which passed, and reviewed a preview. A deploy diagnostic reported an environmental failure; the records don't attribute it to this change. No separate QA or Policy pass is recorded. I handled the review.

I committed it and deployed that night. I then downloaded the live pack list and confirmed the corrected attributes and subtitle. The seller could identify the card's printing and type even without an inventory link.

KLAUTH, UNRIVALED ANCIENT

BEFORE / FROM NOTES

Colorless / NM

Attributes: blank. Card type: missing.

AFTER / FROM NOTES

Nonfoil / Multicolor / NM / Dragon

Attributes: PLST #AFC-50

Illustrative comparison reconstructed from the supplied review notes. This is not a screenshot of the pack list PDF.

AUGUST 17, 2026 / SEVEN DOCUMENTED STEPS

1 Requirement

I found the gap

2 Implementation

AI backend change

3 Testing

One test file: PASS

4 Review

My manual verification

5 Approval

I committed the change

6 Deployment

File sync + restart

7 Verification

Live pack list checked

Two failures that changed my checks.

JUNE 17-18, 2026 / UTC DEPLOY WINDOW

131 files copied. The app still failed.

The sync copied 131 of 131 files, but code needed at startup was missing. The checker followed the deploy list's dependencies, not the application's entrypoint.

I directed the missing-file copy, fixed the list, and restarted the app. The default diagnostic subset and my smoke check passed; service returned in the same session. The checker now includes startup dependencies. I run it before most production syncs. That change came after the outage.

JULY 6, 2026 / DATABASE INTEGRITY

The server and backups disagreed.

An orphaned debug script kept replacing the SQLite database path. The server stayed attached to an unlinked file; backups copied a stale replacement. A restart exposed the old data.

I traced open files and filesystem writes, stopped the script, and cut over to a linked database. I reconciled roughly five paid accounts from payment-processor records. I added an unlinked-file alarm and a check that backups copy the file the server actually uses. The incident was closed the same day.

If your team needs someone who can build with AI and own the work through production, let's talk.

kevin.granger@kevingranger.com

thetcgforge.com

CompTIA Security+ and CySA+ active through July 2029.