

eBay2Parcel

Nov 22, 2025 • spencerburleigh.com/ebay2parcel

I use [Parcel](#)^[a] to track all of my deliveries. The app has an Amazon integration but I purchase a remarkable number of clothes, tech, and vintage aviation memorabilia on eBay.

I built [eBay2Parcel](#)^[b] to automatically import eBay tracking numbers into Parcel. Once a day it asks eBay for my recent buyer orders, filters for the ones that have a tracking number and haven't been delivered, and sends them to Parcel.

Parcel is my favorite delivery tracker [1]: 300+ carriers supported, push notifications, nice widgets, Amazon login that stays on-device.

I've been building some eBay authentication tools [2] as a microservice for another project I've been working on which made this an easy natural extension.

Don't waste the 20 request limit

The add-delivery endpoint is for [premium](#)^[a] users, and is limited to [20 requests per day](#)^[c], counting failures. Making sure we don't send too many requests is most of the program:

- skip shipments eBay already marked delivered
- skip anything older than `MAX_SHIPMENT_AGE_DAYS` (45 by default), even if eBay still returns it in the 90-day window
- remember what you already posted in `tracking_history.json` (atomic write, file lock)
- treat “already added” as success so you don't retry
- stop processing the current eBay account on HTTP 429 (too many requests) and cap requests per account per run at 20 by default

The carrier names eBay returns are messy strings. The script translates some common names into Parcel's codes, but that mapping still has rough edges. Unrecognized names can pass through unchanged; unsupported-carrier errors get logged for a manual add. Those failed requests still count toward the daily limit.

Multiple eBay accounts work if you suffix the env vars (`EBAY_APP_ID_2` , and so on). Each account gets its own request counter, which resets on each run; if they share a Parcel key, budget for the combined requests across accounts and runs.

Install

The current version runs locally and requires Python 3.11+, your own [eBay developer app](#) ^[d], and a Parcel premium subscription with an API key from [web.parcelapp.net](#) ^[e]:

```
git clone https://github.com/sburl/eBay2Parcel.git
cd eBay2Parcel
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
cp .env.example .env
```

Fill in the eBay keys and `PARCEL_API_KEY`, then `python -m shared_ebay.generate_token` once, then `python main.py`. There's a `run_ebay2parcel.sh.example` for cron. On macOS, cron may need Full Disk Access, depending on where you keep the script and its files.

[1] Ivan if you're reading this let's get this natively integrated into Parcel!

[2] Also [open source!](#) ^[f]

Links

- a.** [parcelapp.net](#)
- b.** [github.com/sburl/eBay2Parcel](#)
- c.** [parcelapp.net/help/api-add-delivery.html](#)
- d.** [developer.ebay.com](#)
- e.** [web.parcelapp.net](#)
- f.** [github.com/sburl/eBayAPIHelpers](#)