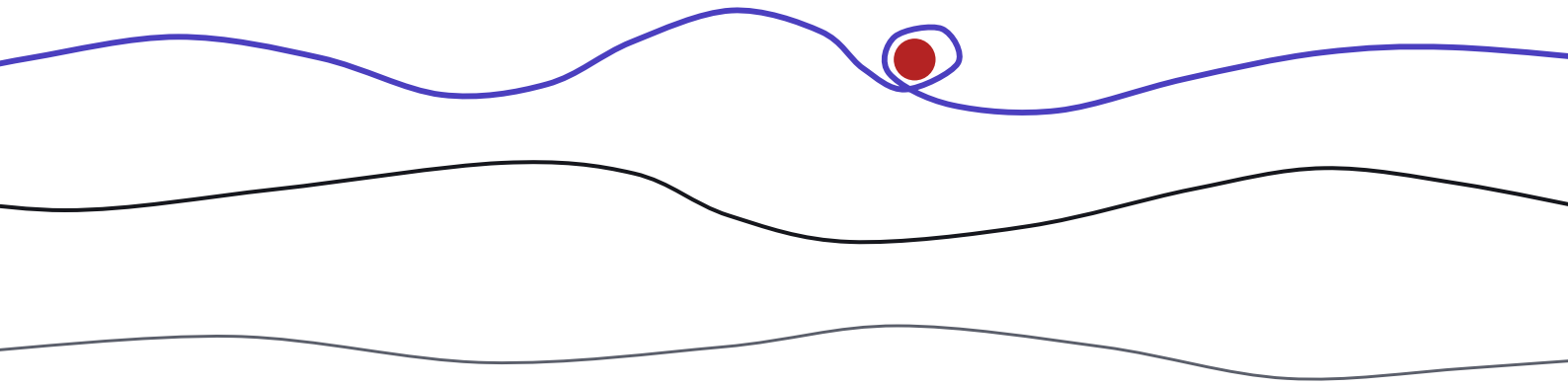
jupiter.au

Procurement MCP

An MCP server that searches Australian and Chinese marketplaces in one call and ranks offers by landed cost to a destination postcode, delivery time and seller trust.



VERSION

v0.1.0

DATE

25 November 2026

AUTHOR

belikh

SOURCE

github.com/belikh/procurement

Contents

Start

01 **Install** 3

01.1 Requirements 3

01.2 Install 3

01.3 Run 3

01.4 First search 4

01.5 Check the setup 4

Guides

02 **Connect an MCP client** 5

02.1 Stdio 5

02.2 Streamable HTTP 5

02.3 Tool call time limit 6

02.4 eBay notification endpoint 6

03 **Add marketplace keys** 6

03.1 Steps 6

03.2 Key notes 7

03.3 Errors 7

04 **Search for a product** 8

04.1 Run a search 8

04.2 Choose a sort 9

04.3 Search Chinese marketplaces 9

04.4 Change the destination 9

04.5 Search eBay auctions and overseas sellers 9

04.6 Refresh one offer 10

04.7 Timeouts 10

05 **Set forwarder rates** 10

05.1 Steps 11

05.2 Currency 11

06 **Keep search history** 11

06.1 Steps 11

06.2 Settings 12

06.3 What the cache key covers 12
06.4 Errors 12

Reference

07 Tools 14
07.1 search_offers 14
07.2 get_offer_detail 15
07.3 search_by_image 16
07.4 latest_deals 17
07.5 get_price_history 17
07.6 search_history 18
07.7 health 18
07.8 Prompt procurement_brief 19

08 Offer format 19
08.1 Fields 20
08.2 Shipping 21
08.3 Diagnostic codes 21
08.4 Tool failure 22

09 Marketplaces 23
09.1 Status 24
09.2 Delivery and ETA 25
09.3 Source notes 25

10 Settings 26
10.1 Destination 26
10.2 Keys 26
10.3 Apify actors 27
10.4 Deal feeds 27
10.5 Forwarder estimate and currency 28
10.6 Database 28
10.7 Behaviour 28
10.8 eBay notifications 29

11 Glossary 29

Concepts

12 How a search runs 31

12.1 Steps 31

12.2 One outcome per lane 31

12.3 Retries 32

12.4 Cache 32

13 Delivery and landed cost 32

13.1 Delivery status 32

13.2 Landed cost 33

13.3 Forwarder estimate 33

14 Ranking 34

14.1 cheapest 34

14.2 fastest 34

14.3 best_value 34

14.4 Currency 35

Develop

15 Build and test 36

15.1 Commands 36

15.2 Layout 37

15.3 Tests 37

15.4 Documentation 37

16 Add a marketplace 38

16.1 Steps 38

16.2 LaneSpec fields 39

16.3 SearchCtx fields 39

16.4 Apify actor lanes 40

17 Verify a lane 40

17.1 Run 40

17.2 Record fixtures 41

17.3 Mark a lane verified 41

Procurement MCP

Procurement MCP is a Model Context Protocol server with 7 tools. `search_offers` searches up to 16 marketplaces at once and returns one ranked list. Each *Offer* carries a delivery status and a *Landed cost* in AUD for the *Destination*, which is Brisbane 4000 by default.

Every source is a hosted API that the server calls with a key. With no keys set, the `ozbargain` and `cheapies` deal feeds work. Every other *Marketplace* returns a *Diagnostic* that names the variable to set.

RUN IT

```
git clone https://github.com/belikh/procurement
cd procurement
uv sync
uv run python server.py
```

Install

Search for a product

Tools

Delivery and landed cost



SEE ALSO

- Connect an MCP client
- Marketplaces
- Settings

SECTION

Start

Install the server and run your first search.

01

Install

Install the server with uv, start it and run a first search without any keys.

01.1

Requirements

REQUIREMENT	VERSION
Python	3.12, pinned in `.python-version`
uv	any current release
`mcp`	2.3 or later, installed by `uv sync`

01.2

Install

1. Clone the repository and install the dependencies.

   INSTALL

   ```
   git clone https://github.com/belikh/procurement
   cd procurement
   uv sync
   ```

2. Copy the settings template. Every key in it is optional, and an empty value counts as unset.

   CREATE .ENV

   ```
   cp .env.example .env
   ```

01.3

Run

The server speaks MCP over stdio by default. Set `MCP_HTTP_PORT` to serve streamable HTTP on `127.0.0.1`.

COMMAND	RESULT
`uv run python server.py`	MCP over stdio
`MCP_HTTP_PORT=8765 uv run python server.py`	Streamable HTTP at `http://127.0.0.1:8765/mcp`
`uv run mcp dev server.py`	The MCP Inspector, to call tools from a browser
`uv run procurement-mcp`	The same as `server.py`, through the installed script

01.4

First search

With no keys set, the `ozbargain` and `cheapies` deal feeds work. In the MCP Inspector, call `search_offers` with these arguments:

ARGUMENTS

```
{ "query": "ssd", "marketplaces": ["ozbargain", "ebay_au"], "sort": "cheapest",
"max_results": 2 }
```

The result lists the OzBargain deals whose title, brand or tags contain “ssd”, then one *Diagnostic* for `ebay_au`:

RESULT (SHORTENED)

```
[
  {
    "offer_id": "ozbargain:983075",
    "marketplace": "ozbargain",
    "title": "Lenovo Legion 5i ... 1TB SSD $2186 (RRP $2999) Delivered @ Amazon AU",
    "price_aud": 2186.0,
    "shipping": { "delivery": "direct", "delivery_basis": "the deal title says
    Delivered" },
    "kind": "offer",
    "landed_aud": 2186.0,
    "landed_basis": "item_only"
  },
  {
    "offer_id": "ebay_au:missing_key",
    "marketplace": "ebay_au",
    "title": "[ebay_au] not configured: set EBAY_APP_ID and EBAY_CERT_ID",
    "kind": "diagnostic",
    "diagnostic_code": "missing_key"
  }
]
```

The deal feeds hold about the latest 100 deals, so the titles you get differ. A query that matches no current deal returns a `no_results` diagnostic.

01.5

Check the setup

Call `health` to see which marketplaces can run. It lists each *Marketplace* with `ready`, the `missing` key, its `source` and its `status`. With no keys, `ready_count` is 2.

SEE ALSO

- Add marketplace keys
- Connect an MCP client
- Search for a product

SECTION

Guides

Common tasks, as numbered steps.

02

Connect an MCP client

Register the server with an MCP client over stdio or streamable HTTP.

An MCP client starts the server as a child process over stdio, or connects to a running server over HTTP. Replace `/path/to/procurement` with the path of your checkout.

02.1

Stdio

Most clients read a JSON entry with a command and its arguments. `uv run --directory` starts the server from the checkout, so the server finds its `.env` file.

MCP SERVER ENTRY

```
{
  "mcpServers": {
    "procurement": {
      "command": "uv",
      "args": ["run", "--directory", "/path/to/procurement", "python", "server.py"]
    }
  }
}
```

In Claude Code, one command adds the same entry:

CLAUDE CODE, STDIO

```
claude mcp add procurement -- uv run --directory /path/to/procurement python
server.py
```

02.2

Streamable HTTP

1. Start the server with a port. It listens on `127.0.0.1` only.

   SERVE HTTP

   ```
   MCP_HTTP_PORT=8765 uv run python server.py
   ```

2. Point the client at `http://127.0.0.1:8765/mcp`.

   CLAUDE CODE, HTTP

```
claude mcp add --transport http procurement http://127.0.0.1:8765/mcp
```

To reach the server from another machine, put a reverse proxy or a tunnel in front of the port. The server has no authentication of its own.

02.3

Tool call time limit

A search runs for up to `PROCUREMENT_DEADLINE_S` seconds, 55 by default. Many clients cut a tool call at 60 seconds. If your client allows longer calls, raise the deadline so that the Apify lanes, which take 30 to 90 seconds, can finish.

.ENV

```
PROCUREMENT_DEADLINE_S=110
```

02.4

eBay notification endpoint

Over HTTP the server also answers eBay's account-deletion notifications at `/ebay/notifications`. eBay keeps a new production keyset disabled until this endpoint answers its challenge.

SETTING	MEANING
`EBAY_DELETION_TOKEN`	The verification token you entered in the eBay developer portal.
`EBAY_DELETION_ENDPOINT`	The public URL registered in the portal. The challenge hash covers this string, so it must match exactly.

A `GET` with `challenge_code` returns `{"challengeResponse": "<sha256 hex>"}`. A `POST` is logged and answered with 204. The server stores no eBay user data.

SEE ALSO

- Search for a product
- Settings

03

Add marketplace keys

Set the API keys that turn each marketplace on, and confirm them with the health tool.

Each *Marketplace* runs when its key is set in `.env` or in the environment. The server reads `.env` from the repository root at start, so restart it after a change.

03.1

Steps

1. Open `.env` and set the keys you hold.

KEY	TURNS ON	WHERE TO GET IT
`APIFY_TOKEN`	`taobao`, `tmall`, `jd`, `1688`, `amazon_au`, `ebay_sold`, `woolworths`, `coles`, `aldi`, `gumtree`, and `search_by_image`	Apify console, Integrations
`EBAY_APP_ID` and `EBAY_CERT_ID`	`ebay_au`, and live `ebay_au` detail	eBay developer program
`TMAPI_TOKEN`	`1688` through TMAPI, and detail for `taobao`, `tmall`, `1688`, `jd`, `pdd`	TMAPI console
`SOCIAVAULT_API_KEY`	`facebook`	SociaVault
`KEEPA_API_KEY` and `KEEPA_DOMAIN`	`get_price_history`	Keepa
`GUMTREE_COOKIES`	`gumtree`, with `APIFY_TOKEN`	Cookies exported from a browser session on gumtree.com.au

2. Restart the server.
3. Call `health`. A marketplace with its key set reports `"missing": null`.

HEALTH, ONE MARKETPLACE

```
{
  "ebay_au": {
    "ready": true,
    "missing": null,
    "source": "eBay Browse API",
    "status": "unverified"
  }
}
```

4. Run one live search for the marketplaces you turned on. Each Apify lane starts one actor run, which spends Apify credits.

LIVE CHECK

```
uv run python scripts/live_smoke.py ebay_au jd
```

03.2

Key notes

- `EBAY_CLIENT_ID`, `EBAY_CLIENT_SECRET` and `TMAPI_API_TOKEN` are accepted as other names for the eBay and TMAPI keys.
- `EBAY_OAUTH_TOKEN` holds a bearer token you fetched yourself. With it set, the server skips the client-credentials call.
- `1688` uses TMAPI when `TMAPI_TOKEN` is set, and the Apify actor otherwise.
- `KEEPA_DOMAIN` has no default. Keepa's official clients list domain ids 1 to 12 and none is Australia. Set the id that your Keepa account documents for amazon.com.au.
- `gumtree` stays `degraded` in `health` and is never counted in `ready_count`. Without `GUMTREE_COOKIES` it returns a `degraded` diagnostic, because Gumtree blocks the actor.
- `health` shows key names only. It prints no key values.

03.3

Errors

DIAGNOSTIC	CAUSE	FIX
`[ebay_au] not configured: set EBAY_APP_ID and EBAY_CERT_ID`	The key is unset or empty.	Set the named variable and restart.
`TMAPI error 4013: API not subscribed; subscribe at console.tmapi.io, APIs List`	TMAPI answered 401.	Subscribe to the API in the TMAPI console.
`actor <name>: ...` with code `quota`	The Apify actor reported its free runs or credit as used up.	Add credit in Apify, or set another actor in the `*_APIFY_ACTOR` setting.

SEE ALSO

- Marketplaces
- Settings
- Verify a lane

04

Search for a product

Call search_offers, choose marketplaces and a sort, and read the three kinds of entry it returns.

`search_offers` takes a product keyword and returns one list: ranked offers, then comps, then diagnostics. Only `query` is required.

04.1

Run a search

1. Call `search_offers` with a query and a sort.

   CHEAPEST DELIVERED PRICE FOR 2 UNITS

   ```
   { "query": "usb-c hub", "sort": "cheapest", "qty": 2 }
   ```

2. To search fewer marketplaces, list them. The search then ends when the slowest listed lane ends.

   TWO MARKETPLACES

   ```
   { "query": "usb-c hub", "marketplaces": ["ebay_au", "amazon_au"] }
   ```

3. Read `kind` on each entry.

KIND	MEANING	ORDER
`offer`	A listing you can buy, ranked by `sort`.	First
`comp`	A past sale price from `ebay_sold`.	Second
`diagnostic`	A marketplace that returned no offers. `diagnostic_code` gives the reason.	Last

4. Compare offers on `landed_aud` and check `landed_basis` beside it. `exact` includes shipping, `item_only` leaves shipping out, and `estimated` holds a forwarder estimate.

04.2

Choose a sort

SORT	ORDERS BY
`best_value` (default)	A score of price, speed and trust.
`cheapest`	`landed_aud`. Grocery offers sort by `unit_price_aud`.
`fastest`	The later bound of `shipping.eta_days`, then price.

Ranking has the formulas.

04.3

Search Chinese marketplaces

`taobao`, `tmall`, `1688` and `jd` match Chinese text far better than English. The server does not translate. Pass the query in Chinese as `query_zh`, and give the item weight so that the forwarder estimate uses it.

CHINESE MARKETPLACES

```
{
  "query": "silicone kitchen utensils",
  "query_zh": "\u7845\u80f6\u53a8\u5177",
  "marketplaces": ["1688", "taobao", "jd"],
  "qty": 50,
  "est_weight_kg": 0.3
}
```

The `\u` escapes spell silicone kitchenware in Chinese characters. JSON reads an escape as the character itself, so the server gets the same text whether you type the escapes or the characters.

An offer whose `moq` is larger than `qty` goes to the end of the list in every sort.

04.4

Change the destination

The *Destination* is `SHIP_TO_COUNTRY` and `SHIP_TO_POSTCODE`, AU 4000 by default. To change it for one call, pass `postcode`, and `ship_to` for another country.

DELIVER TO MELBOURNE

```
{ "query": "usb-c hub", "postcode": "3000" }
```

`ebay_au` sends the postcode to eBay and gets the shipping options and costs for it. Pickup searches on `facebook` stay centred on `SHIP_TO_LAT` and `SHIP_TO_LNG`, which a call cannot change.

To drop listings that must be collected in person, pass `"include_pickup": false`.

04.5

Search eBay auctions and overseas sellers

ARGUMENT	VALUE	RESULT
`ebay_auctions`	`true`	Auction listings only. `price_aud` is the current bid and `auction` holds the bid count and end time.
`ebay_seller_scope`	`domestic` (default)	Items located in Australia.
`ebay_seller_scope`	`global`	Sellers worldwide on the AU marketplace.
`ebay_seller_scope`	`auto`	eBay's default ranking.

04.6

Refresh one offer

Call `get_offer_detail` with an `offer_id` from the search. `taobao`, `tmall`, `1688`, `jd` and `pdd` ids are fetched from TMAPI, and `ebay_au` ids from eBay. For every other marketplace the tool returns the offer as the most recent search held it.

ARGUMENTS

```
{ "offer_id": "ebay_au:v1|1234|0" }
```

04.7

Timeouts

All lanes run at once under one deadline of 55 seconds. A lane that has not replied by then is cancelled and returns a `timeout` diagnostic. Apify lanes take 30 to 90 seconds.

To give one search more time, pass `deadline_s` (5 to 300):

A LONGER SEARCH

```
{ "query": "usb-c hub", "marketplaces": ["taobao", "jd"], "deadline_s": 120 }
```

A search with a `timeout`, `upstream_error` or `quota` diagnostic is not cached, so the next call tries the source again.

SEE ALSO

- Tools
- Offer format
- Delivery and landed cost

05

Set forwarder rates

Replace the placeholder freight rates so that landed cost for Chinese offers matches your forwarder's rate card.

Offers from `taobao`, `tmall`, `1688`, `jd` and `pdd` ship within China only. Their `landed_aud` holds an estimate of a *Forwarder*'s charge. The defaults are placeholders: 12 AUD per kg, a 15 AUD minimum and a 0.5 kg item.

05.1

Steps

1. Read these four numbers from your forwarder's rate card.

SETTING	DEFAULT	MEANING
`FORWARDER_RATE_AUD_PER_KG`	`12.0`	Freight per kg, China to Australia.
`FORWARDER_MIN_CHARGE_AUD`	`15.0`	The smallest freight charge for one shipment.
`FORWARDER_FEE_PCT`	`0.0`	The forwarder's service fee as a fraction: `0.05` is 5 %.
`FORWARDER_DEFAULT_WEIGHT_KG`	`0.5`	The weight of one item when a search gives no `est_weight_kg`.

2. Set them in `.env`.

.ENV
```
FORWARDER_RATE_AUD_PER_KG=9.5
FORWARDER_MIN_CHARGE_AUD=20
FORWARDER_FEE_PCT=0.05
```

3. Restart the server and call `health`. `forwarder_estimate` shows the values in use.
4. Pass `est_weight_kg` on each search of Chinese marketplaces. Search results carry no weight, so the estimate otherwise uses the default.

05.2

Currency

`CNY_TO_AUD` (default `0.215`) and `NZD_TO_AUD` (default `0.92`) are fixed settings. The server fetches no live rates. Update them when the rates move.

SEE ALSO

- Delivery and landed cost
- Settings

06

Keep search history

Connect Postgres to cache search results for an hour and log every search.

With `DATABASE_URL` set, the server caches each search result and logs each search in Postgres. Without it the server keeps no state between calls, and `search_history` returns `db_disabled`.

06.1

Steps

1. Set the connection string in `.env`.

.ENV
```
DATABASE_URL=postgresql://procurement:__SET_ME__@db.example.org:5432/jupiter
```

2. Start the server. At start it creates the schema `procurement` and the tables `search_cache` and `search_log` if they are missing.
3. If the database role cannot create a schema, apply the migration once as a role that can, then grant the server's role access. The grants are at the end of the file, commented out.

APPLY THE SCHEMA BY HAND

```
psql "$DATABASE_URL" -f migrations/001_procurement_cache.sql
```

4. Call `health` and read `database`.

HEALTH, DATABASE

```
{
  "configured": true,
  "schema": "procurement",
  "enabled": true,
  "error": null,
  "cache_ttl_s": 3600
}
```

When `enabled` is `false`, `error` holds the Postgres error. The server still runs and searches without the cache.

5. Run a search, then call `search_history`.

SEARCHES OF THE LAST DAY THAT MENTION HUB

```
{ "source": "log", "query_contains": "hub", "since_hours": 24 }
```

06.2

Settings

SETTING	DEFAULT	MEANING
`DATABASE_URL`	unset	Postgres connection string. Unset turns the cache and the log off.
`PROCUREMENT_PG_SCHEMA`	`procurement`	Schema name. It must match `[a-z_][a-z0-9_]*`.
`PROCUREMENT_CACHE_TTL_S`	`3600`	Seconds a cached result is served.

The migration file creates the schema `procurement` by name. If you set another `PROCUREMENT_PG_SCHEMA`, let the server create the tables, or edit the file.

06.3

What the cache key covers

Two searches share a cached result when all of these match: `query`, `query_zh`, `est_weight_kg`, `include_pickup`, the marketplaces, `sort`, `qty`, the destination, `max_results`, `ebay_auctions` and `ebay_seller_scope`. `deadline_s` is left out of the key.

06.4

Errors

MESSAGE	CAUSE	FIX
`search history needs the database: DATABASE_URL is not set`	No database is configured.	Set `DATABASE_URL` and restart.
`search history needs the database: <Postgres error>`	The connection or the schema setup failed at start.	Fix the cause named in the message and restart.

SEE ALSO

- search_history
- How a search runs

SECTION

Reference

Every command, option and marketplace lane.

07

Tools

The 7 tools and the prompt, with every argument, its default and what each tool returns.

The server registers as `procurement-search` with 7 tools and 1 prompt.

TOOL	PURPOSE	NEEDS
`search_offers`	Search marketplaces and rank the offers.	A key per marketplace
`get_offer_detail`	Refresh one offer.	`TMAPI_TOKEN` or eBay keys for a live fetch
`search_by_image`	Find listings that match a photo.	`APIFY_TOKEN`
`latest_deals`	List the newest deals, most upvoted first.	Nothing
`get_price_history`	Amazon price history from Keepa.	`KEEPA_API_KEY`, `KEEPA_DOMAIN`
`search_history`	Read past searches.	`DATABASE_URL`
`health`	Report which marketplaces can run.	Nothing

07.1

search_offers

Searches the requested marketplaces at once under one deadline. Returns a list of entries: offers ranked by `sort`, then comps, then diagnostics. Offer format lists the fields.

ARGUMENT	TYPE	DEFAULT	MEANING
`query`	string	required	Product keyword. Sent as written to every source.
`marketplaces`	list of strings	all 16	The marketplaces to search. An unknown name returns an `unsupported` diagnostic.
`sort`	`cheapest`, `fastest`, `best_value`	`best_value`	How offers are ranked.
`qty`	integer, 1 or more	`1`	Quantity. Offers with a larger minimum order rank last.
`ship_to`	string	`SHIP_TO_COUNTRY`	Destination country code.
`postcode`	string	`SHIP_TO_POSTCODE`	Destination postcode.
`max_results`	integer, 1 to 50	`10`	Cap per marketplace.
`query_zh`	string	none	The query in Chinese, for `taobao`, `tmall`, `1688` and `jd`.
`est_weight_kg`	number above 0	`FORWARDER_DEFAULT_WEIGHT_KG`	Weight of one item, for the forwarder estimate.
`include_pickup`	boolean	`true`	Keep listings that must be collected in person.
`deadline_s`	number, 5 to 300	`PROCUREMENT_DEADLINE_S`	Seconds allowed for the whole search.
`ebay_auctions`	boolean	`false`	eBay: auction listings only.
`ebay_seller_scope`	`domestic`, `global`, `auto`	`domestic`	eBay: items located in AU, sellers worldwide, or eBay's default.

The result holds at most `max_results * 2` offers. Comps and diagnostics are added after that cut.

07.2

get_offer_detail

Returns one offer, or a tool failure.

ARGUMENT	TYPE	DEFAULT	MEANING
`offer_id`	string	required	An id from `search_offers`, in the form `<marketplace>:<item id>`.

MARKETPLACE OF THE ID	SOURCE OF THE ANSWER
`taobao`, `tmall`, `1688`, `jd`, `pdd`	TMAPI item detail, fetched live.
`ebay_au`	eBay Browse API, fetched live.
any other	The offer as the most recent search held it.

When a live fetch fails, the tool falls back to the stored offer, then to the failure. The server remembers the last 2000 offers in memory. With a database it also looks in unexpired cached searches.

FAILURE CODE	ERROR	CAUSE
`bad_id`	`offer id must look like <marketplace>:<item id>`	The id has no colon.
`diagnostic_id`	`this id names a diagnostic entry, which has no detail`	The id is a diagnostic's, such as `jd:timeout`.
`not_found`	`<marketplace> has no detail endpoint and no recent search holds this offer; run search_offers again`	The offer is in neither the memory nor the cache.
`missing_key`	`TMAPI_TOKEN is not set`	A live fetch needs a key that is unset.
`unsubscribed`	`TMAPI error 4013: API not subscribed; subscribe at console.tmapi.io, APIs List`	TMAPI answered 401.
`upstream_error`	The source's error	The source replied with an error.

07.3

search_by_image

Runs a reverse image search through an Apify actor and returns matching listings as offers. One search costs about 0.10 USD in Apify credit.

ARGUMENT	TYPE	DEFAULT	MEANING
`image_url`	string	required	Public URL of a product photo, JPG or PNG.
`destination`	`1688`, `alibaba`, `aliexpress`, `global`	`1688`	Where to search. `global` covers Amazon, Walmart, eBay and other retail sites.
`max_results`	integer, 1 to 50	`10`	Cap on matches.

The offers carry a price, a minimum order and a supplier name. They have no delivery status and no landed cost. A failed or empty search returns one diagnostic.

07.4

latest_deals

Reads the deal sites' RSS feeds, which hold about the latest 100 deals each. Returns deals sorted by net votes, highest first, then one diagnostic for each feed that failed.

ARGUMENT	TYPE	DEFAULT	MEANING
`sources`	list of `ozbargain`, `cheapies`	`["ozbargain"]`	The deal sites to read. `cheapies` is a New Zealand site priced in NZD.
`query`	string	none	Keep deals whose title, brand or tags contain every word.
`min_votes`	integer, 0 or more	`0`	Keep deals with at least this many net votes.
`max_results`	integer, 1 to 100	`20`	Cap on deals.

Each deal has a `community` block with votes, comments, clicks and expiry. `price_aud` is the first dollar amount in the deal's title.

07.5

get_price_history

Returns the Amazon price history of one product from Keepa, for the price sold by Amazon itself.

ARGUMENT	TYPE	DEFAULT	MEANING
`offer_id_or_asin`	string	required	`amazon_au:B0XXXXXXXX` or a bare 10-character ASIN.
`days`	integer, 1 to 365	`90`	Window for the price series.
`include_series`	boolean	`false`	Also return every price change in the window.

RESULT FIELD	MEANING
`asin`	The product id.
`current_aud`	The current price.
`avg_30d_aud`, `avg_90d_aud`	The 30-day and 90-day average.
`min_aud`, `max_aud`	The lowest and highest price Keepa has recorded.
`series`	With `include_series`: a list of `{ t, price_aud }`, oldest first. `t` is an ISO datetime in UTC.

FAILURE CODE	ERROR	CAUSE
`bad_id`	`expected a 10-character ASIN or amazon_au: <ASIN>`	The argument is no ASIN.
`missing_key`	`KEEPA_API_KEY is not set` or `KEEPA_DOMAIN is not set`	A Keepa setting is unset.
`not_found`	`Keepa has no price data for <ASIN>`	Keepa does not track the product.

07.6

search_history

Reads the Postgres search log or the cache, newest first.

ARGUMENT	TYPE	DEFAULT	MEANING
`source`	`log`, `cache`	`log`	`log`: every past search. `cache`: unexpired cached results.
`query_contains`	string	none	Keep searches whose query contains this text. Case is ignored.
`since_hours`	integer, 1 to 2160	`168`	How far back to look.
`limit`	integer, 1 to 200	`20`	Cap on rows.
`include_offers`	boolean	`false`	`cache` only: include the cached offers in each row.

SOURCE	ROW FIELDS
`log`	`id`, `created_at`, `query`, `marketplaces`, `sort`, `qty`, `ship_to`, `result_count`, `elapsed_ms`
`cache`	`cache_key`, `created_at`, `query`, `marketplaces`, `sort`, `expires_in_s`, `offer_count`, and `offers` on request

Without a database the tool returns a failure with code `db_disabled`.

07.7

health

Takes no arguments and shows no key values.

FIELD	MEANING
`status`	`ok` when the server answers.
`marketplaces`	For each marketplace: `ready`, `missing` (the unset key), `source` and `status`.
`ready_count`	The number of marketplaces with `ready: true`.
`image_search`	`ready` and `missing` for `search_by_image`.
`price_history`	`ready` and the list of `missing` Keepa settings.
`ship_to`	The default destination: `country`, `postcode`, `lat`, `lng`, `radius_km`.
`forwarder_estimate`	The freight rate, default weight, minimum charge, fee and GST rate in use.
`deadline_s`, `cny_to_aud`	The search deadline and the CNY rate in use.
`database`	`configured`, `schema`, `enabled`, `error`, `cache_ttl_s`.

`ready` is `true` when the key is set and the status is `verified` or `unverified`. Status lists the four values.

07.8

Prompt procurement_brief

Takes `query` and returns a system prompt. The prompt tells the model to call `search_offers` with `sort: best_value`, compare `landed_aud` with its `landed_basis`, and report diagnostics separately.

SEE ALSO

- Offer format
- Marketplaces
- Search for a product

08

Offer format

Every field of an entry in a search result, the diagnostic codes and the tool failure format.

`search_offers`, `search_by_image` and `latest_deals` return a list of entries with one shape. `kind` says what an entry is: `offer`, `comp` or `diagnostic`.

AN OFFER FROM EBAY_AU (EXAMPLE VALUES)

```
{
  "offer_id": "ebay_au:v1|1234|0",
  "marketplace": "ebay_au",
  "title": "USB-C Hub 7 in 1",
  "url": "https://www.ebay.com.au/itm/1234",
  "price_aud": 39.95,
  "currency": "AUD",
  "shipping": {
    "eta_days": [3, 7],
    "delivery": "direct",
    "delivery_basis": "eBay lists a shipping option for the buyer's postcode",
    "cost_aud": 8.5
```

```
  },
  "seller": { "shop_name": "example-seller" },
  "source_reliability": "authoritative",
  "kind": "offer",
  "landed_aud": 48.45,
  "landed_basis": "exact",
  "landed_breakdown": { "goods": 39.95, "shipping": 8.5 }
}
```

08.1

Fields

FIELD	TYPE	MEANING
`offer_id`	string	`<marketplace>:<item id>`. Pass it to `get_offer_detail`.
`marketplace`	string	The marketplace the entry belongs to.
`kind`	`offer`, `comp`, `diagnostic`	Default `offer`.
`title`, `title_en`	string	The listing title, and its English form when the source gives one.
`url`, `image`	string	Links to the listing and its picture.
`price_aud`	number	The price of one item in AUD. For an auction, the current bid.
`price_cny`	number	The price in CNY, on Chinese marketplaces.
`price_native`, `currency`	number, string	The price in `currency` when the source gave no AUD price. `price_aud` is then null.
`moq`	integer	Minimum order quantity.
`stock`	integer	Units in stock, when the source states it.
`unit_price_aud`, `unit`	number, string	Grocery: the price per unit, such as per 100 g or per litre.
`regular_price_aud`	number	Grocery: the price before a special. `price_aud` holds the special price.
`shipping`	object	See Shipping.
`seller`	object	`shop_name`, `shop_id`, `rating`, `verified`.
`seller_country`	string	Country code of the item or seller, from eBay.
`community`	object	Deal sites: `votes_pos`, `votes_neg`, `comment_count`, `click_count`, `expiry`, `brand`, `tags`.
`auction`	object	eBay auctions: `current_bid_aud`, `bid_count`, `end_time`, `time_left`, `reserve_met`.
`price_history`	object	Amazon offers: the Keepa summary, when Keepa is configured.

FIELD	TYPE	MEANING
`landed_aud`	number	Total in AUD for `qty` delivered to the destination.
`landed_basis`	`exact`, `estimated`, `item_only`	How firm `landed_aud` is.
`landed_breakdown`	object	The parts that sum to `landed_aud`.
`source_reliability`	`authoritative`, `aggregator`, `best_effort`	How direct the source is. Default `aggregator`.
`diagnostic_code`	string	Set when `kind` is `diagnostic`.
`raw`	object	The source's payload. Set when `PROCUREMENT_INCLUDE_RAW=1`.

A field with no value is null.

08.2 Shipping

FIELD	TYPE	MEANING
`delivery`	`direct`, `pickup`, `via_forwarder`, `no`, `unknown`	Whether and how the offer reaches the destination. Default `unknown`.
`delivery_basis`	string	What the delivery status rests on.
`cost_aud`	number	The shipping cost the source stated. Null means the source stated none. It does not mean free.
`eta_days`	`[min, max]`	Delivery window in days. Fixed per marketplace.
`method`	string	A label for the shipping method.
`domestic_cny`	number	The delivery leg inside China, in CNY.
`intl_estimate_aud`	number	The forwarder freight estimate in AUD.

Delivery and landed cost explains each `delivery` value and each `landed_basis`.

08.3 Diagnostic codes

A requested marketplace with no offers returns one diagnostic. Its `title` is `[<marketplace>] <detail>`.

DIAGNOSTIC_CODE	OFFER_ID	MEANING
`missing_key`	`<marketplace>:missing_key`	The key is not set. The title names the variable.
`timeout`	`<marketplace>:timeout`	No reply before the deadline. The upstream call was cancelled.
`upstream_error`	`<marketplace>:error`	The source replied with an error, or its reply could not be read.
`quota`	`<marketplace>:error`	An Apify actor reported its free runs or credit as used up.
`unsubscribed`	`<marketplace>:unsubscribed`	TMAPI answered 401: the API is not subscribed in the TMAPI console.
`degraded`	`<marketplace>:degraded`	The source needs extra setup. Gumtree needs `GUMTREE_COOKIES`.
`unsupported`	`<marketplace>:unsupported`	No source exists for this marketplace, or the name is unknown.
`no_results`	`<marketplace>:no_results`	The source ran and matched nothing.
`not_deliverable`	`<marketplace>:not_deliverable`	Offers were found and left out because they do not reach the destination.

A DIAGNOSTIC

```
{
  "offer_id": "ebay_au:missing_key",
  "marketplace": "ebay_au",
  "title": "[ebay_au] not configured: set EBAY_APP_ID and EBAY_CERT_ID",
  "kind": "diagnostic",
  "diagnostic_code": "missing_key"
}
```

08.4

Tool failure

`get_offer_detail`, `get_price_history` and `search_history` return this object when they cannot produce a result.

FIELD	TYPE	MEANING
`error`	string	What went wrong, in a sentence.
`code`	string	`bad_id`, `diagnostic_id`, `not_found`, `db_disabled`, or a diagnostic code.
`offer_id`	string	The id the call was about, when there is one.

A TOOL FAILURE

```
{ "error": "KEEPA_API_KEY is not set", "code": "missing_key", "offer_id": null }
```

SEE ALSO

- Tools
- Delivery and landed cost

09

Marketplaces

The 16 marketplace names, the hosted API behind each, the key it needs and its verification status.

Pass these names in `marketplaces` of `search_offers`. With no list, the server searches all 16.

MARKETPLACE	SOURCE	KEY	STATUS	BUDGET
`ebay_au`	eBay Browse API, `EBAY_AU` marketplace	`EBAY_APP_ID`, `EBAY_CERT_ID`	unverified	15 s
`ebay_sold`	Apify `caffein.dev/ebay-sold-listings`. Returns comps.	`APIFY_TOKEN`	unverified	65 s
`amazon_au`	Apify `junglee/free-amazon-product-scraper`	`APIFY_TOKEN`	unverified	75 s
`woolworths`	Apify `dromb/woolworths-au-product-search-catalog-unofficial`	`APIFY_TOKEN`	unverified	45 s
`coles`	Apify `dromb/coles-au-product-search-specials-stores-unofficial`	`APIFY_TOKEN`	unverified	45 s
`aldi`	Apify `dromb/aldi-au-product-search-catalog-unofficial`	`APIFY_TOKEN`	unverified	45 s
`facebook`	SociaVault Facebook Marketplace search	`SOCIAVAULT_API_KEY`	unverified	15 s

MARKETPLACE	SOURCE	KEY	STATUS	BUDGET
`gumtree`	Apify `crawlerbros/gumtree-scraper`	`APIFY_TOKEN`, `GUMTREE_COOKIES`	degraded	60 s
`ozbargain`	OzBargain RSS feed	none	verified	12 s
`cheapies`	Cheapies RSS feed (New Zealand)	none	verified	12 s
`taobao`	Apify `zen-studio/taobao-search-scraper`	`APIFY_TOKEN`	unverified	95 s
`tmall`	The same actor run as `taobao`	`APIFY_TOKEN`	unverified	95 s
`jd`	Apify `zen-studio/jd-com-search-scraper`	`APIFY_TOKEN`	unverified	95 s
`1688`	TMAPI `/1688/search/items`, else Apify `zen-studio/1688-wholesale-scraper`	`TMAPI_TOKEN` or `APIFY_TOKEN`	unverified	95 s
`pdd`	No search source. TMAPI item detail works through `get_offer_detail`.	`TMAPI_TOKEN`	unsupported	
`aliexpress`	No source yet		unsupported	

Budget is the longest one lane may run. The search deadline (55 seconds by default) also applies, and the shorter of the two wins.

09.1

Status

`health` reports one of four values for each marketplace.

STATUS	MEANING
`verified`	A live run returned offers and the parser matched them.
`unverified`	The lane is written from the source's documentation or from notes of an earlier run. No live run has confirmed it since the rewrite of 2026-11.
`degraded`	The lane returns nothing without extra setup.
`unsupported`	No source exists yet.

Verify a lane shows how to run a live check for the keys you hold.

09.2

Delivery and ETA

Each lane sets a fixed ETA window. The windows are constants in the code. No source supplies them.

MARKETPLACE	ETA_DAYS	DEFAULT DELIVERY
`ebay_au`	3 to 7	`direct` when eBay lists a shipping option to the postcode, else `unknown`
`amazon_au`	2 to 7	`direct`
`woolworths`, `coles`	1 to 5	`direct`
`aldi`	1 to 5	`pickup`
`gumtree`	2 to 5	`pickup`
`facebook`	1 to 4	`pickup`, or `direct` when the listing offers shipping
`ozbargain`	0 to 0	`direct` when the title says “Delivered”, else `unknown`
`cheapies`	0 to 0	`no`
`taobao`, `tmall`, `1688`, `jd`	10 to 25	`via_forwarder`

09.3

Source notes

Deal feeds

`ozbargain` and `cheapies` read the sites’ RSS feeds, which hold about the latest 100 deals. A search keeps deals whose title, brand or tags contain every query word. The price is the first dollar amount in the title. Cheapies prices are in NZD and convert with `NZD_TO_AUD`.

Grocery

`woolworths`, `coles` and `aldi` offers carry `unit_price_aud`, `unit` and `regular_price_aud`. `price_aud` holds the special price when there is one. Results pass a relevance filter: every query word must appear in the name, brand or category.

eBay

`price_aud` is set only when eBay reports the amount in AUD. Other currencies go to `price_native` with `currency`. `shipping.cost_aud` is set only when eBay states a cost for the destination postcode.

TMAPI

The 1688 endpoints are `/1688/search/items` and `/1688/item_detail`. The `/alibaba/...` paths belong to Alibaba.com, which this server does not use. The server sends the key as an `apikey` header and as an `apiToken` query parameter.

Tmall

Taobao actor items flagged `isTmall` are labelled `tmall`. If the actor's items carry no such flag, `tmall` returns a diagnostic and the listings stay under `taobao`.

Keepa

`get_price_history` needs `KEEPA_API_KEY` and `KEEPA_DOMAIN`. Prices are read as integer cents, and `-1` means no offer. The Keepa code has not been run against the live API.

Sources left out

- Amazon Product Advertising API 5.0 is deprecated in favour of the Creators API, which requires 10 qualifying sales in the past 30 days. The server uses the Apify actor.
- Every Costco actor on the Apify store targets costco.com (US, USD), so Costco has no lane.

SEE ALSO

- Add marketplace keys
- Settings

10

Settings

Every environment variable the server reads, with its default.

The server reads settings from the environment and from `.env` in the repository root. The environment wins over `.env`. Every setting is optional, and an empty value counts as unset. `.env.example` lists them all.

10.1

Destination

VARIABLE	DEFAULT	MEANING
`SHIP_TO_COUNTRY`	`AU`	Destination country code.
`SHIP_TO_POSTCODE`	`4000`	Destination postcode (Brisbane).
`SHIP_TO_LAT`	`-27.4698`	Latitude of the centre for pickup searches.
`SHIP_TO_LNG`	`153.0251`	Longitude of the centre for pickup searches.
`PICKUP_RADIUS_KM`	`50`	Radius for pickup searches.

10.2

Keys

VARIABLE	DEFAULT	MEANING
APIFY_TOKEN	unset	Apify token for every Apify actor lane and search_by_image.
TMAPI_TOKEN	unset	TMAPI key. TMAPI_API_TOKEN is accepted too.
EBAY_APP_ID	unset	eBay client id. EBAY_CLIENT_ID is accepted too.
EBAY_CERT_ID	unset	eBay client secret. EBAY_CLIENT_SECRET is accepted too.
EBAY_OAUTH_TOKEN	unset	A bearer token fetched by you. Skips the client-credentials call.
SOCIAVAULT_API_KEY	unset	SociaVault key for facebook.
KEEPA_API_KEY	unset	Keepa key for get_price_history.
KEEPA_DOMAIN	unset	Keepa's numeric domain id for amazon.com.au.
GUMTREE_COOKIES	unset	Cookies from a browser session on Gumtree.

10.3 Apify actors

Each lane runs the actor named here. Change a value to run another actor with the same output format.

VARIABLE	DEFAULT
TAOBAO_APIFY_ACTOR	zen-studio/taobao-search-scraper
JD_APIFY_ACTOR	zen-studio/jd-com-search-scraper
ALI1688_APIFY_ACTOR	zen-studio/1688-wholesale-scraper
AMAZON_APIFY_ACTOR	junglee/free-amazon-product-scraper
EBAY_SOLD_APIFY_ACTOR	caffein.dev/ebay-sold-listings
WOOLWORTHS_APIFY_ACTOR	dromb/woolworths-au-product-search-catalog-unofficial
COLES_APIFY_ACTOR	dromb/coles-au-product-search-specials-stores-unofficial
ALDI_APIFY_ACTOR	dromb/aldi-au-product-search-catalog-unofficial
GUMTREE_APIFY_ACTOR	crawlerbros/gumtree-scraper
IMAGE_SEARCH_APIFY_ACTOR	dev00/alibaba-1688-aliexpress-reverse-image-search-api

10.4 Deal feeds

VARIABLE	DEFAULT
OZBARGAIN_FEED_URL	https://www.ozbargain.com.au/deals/feed
CHEAPIES_FEED_URL	https://www.cheapies.nz/deals/feed

10.5 Forwarder estimate and currency

VARIABLE	DEFAULT	MEANING
FORWARDER_RATE_AUD_PER_KG	12.0	Freight per kg, China to Australia. A placeholder.
FORWARDER_DEFAULT_WEIGHT_KG	0.5	Weight of one item when a search gives none. A placeholder.
FORWARDER_MIN_CHARGE_AUD	15.0	Smallest freight charge. A placeholder.
FORWARDER_FEE_PCT	0.0	Forwarder service fee as a fraction.
GST_RATE	0.10	GST applied to the forwarded total.
CNY_TO_AUD	0.215	Fixed CNY to AUD rate.
NZD_TO_AUD	0.92	Fixed NZD to AUD rate, for Cheapies prices.

10.6 Database

VARIABLE	DEFAULT	MEANING
DATABASE_URL	unset	Postgres connection string. Unset: no cache, no log.
PROCUREMENT_PG_SCHEMA	procurement	Schema name. Must match `[a-z_][a-z0-9_]*`.
PROCUREMENT_CACHE_TTL_S	3600	Seconds a cached search is served.

10.7 Behaviour

VARIABLE	DEFAULT	MEANING
PROCUREMENT_DEADLINE_S	55	Seconds allowed for a whole search.
PROCUREMENT_TIMEOUT_S	8	Timeout of one direct API call (eBay, TMAPI, feeds).
PROCUREMENT_INCLUDE_RAW	0	1 adds each source's raw payload to offers as raw.
MCP_HTTP_PORT	unset	Serve streamable HTTP on 127.0.0.1:<port> at /mcp. Unset: stdio.

10.8 eBay notifications

VARIABLE	DEFAULT	MEANING
EBAY_DELETION_TOKEN	unset	Verification token for eBay account-deletion notifications.
EBAY_DELETION_ENDPOINT	https://procurement.jupiter.au/ebay/notifications	The URL registered in the eBay developer portal.

SEE ALSO

- Add marketplace keys
- Set forwarder rates
- Keep search history

11 Glossary

The terms these pages use for marketplaces, offers and costs.

Comp
A past sale price from ebay_sold. A comp shows what an item sold for and cannot be bought.

Destination
The country and postcode that offers must reach. The default is AU 4000, Brisbane.

Diagnostic
The entry a marketplace returns when it has no offers. Its diagnostic_code gives the reason.

Forwarder
A freight company that receives goods at a warehouse in China and ships them to Australia.

Landed cost
The total in AUD for the requested quantity delivered to the destination postcode.

Lane
The code that searches one marketplace through one hosted API. Each lane returns offers or one diagnostic.

Marketplace
A name you pass to search_offers, such as ebay_au or 1688. The server knows 16.

Offer
One listing you can buy, with a price, a delivery status and a landed cost.

SECTION

Concepts

How the server works and why.

12

How a search runs

The steps from a search_offers call to the ranked list, with the lane registry, the deadline and the cache.

The server contains no scrapers. Each *Marketplace* has a *Lane* that calls one hosted API: the eBay Browse API, TMAPI, SociaVault, an Apify actor or an RSS feed. A search runs every requested lane at once and merges what comes back.

12.1

Steps

1. **Resolve names.** The server lower-cases the requested names and drops repeats. A name that is not in the registry gets an `unsupported` diagnostic.
2. **Read the cache.** With a database, the server looks for an unexpired result with the same cache key and returns it.
3. **Check keys.** A lane whose key is unset is not started. It returns a `missing_key` diagnostic that names the variable.
4. **Run lanes.** The remaining lanes start together. Lanes in one group share one upstream run: `taobao` and `tmall` use one Taobao actor run.
5. **Apply the deadline.** Each lane may run for its own budget or until the search deadline, whichever is shorter. A lane that is still running is cancelled and returns a `timeout` diagnostic.
6. **Annotate.** The server sets a delivery status and a landed cost on every offer.
7. **Rank.** The server ranks the offers by `sort` and keeps the first `max_results * 2`.
8. **Assemble.** The result is the ranked offers, then the comps, then the diagnostics.
9. **Store.** The server caches the result and logs the search, when a database is configured.

12.2

One outcome per lane

Every requested marketplace returns offers or exactly one *Diagnostic*. A caller can tell an empty marketplace from a broken one, because the diagnostic's code says which it is: `no_results`, `timeout`, `missing_key` and so on.

Any exception inside a lane becomes an `upstream_error` diagnostic. `search_offers` itself does not fail when a source fails.

12.3

Retries

Direct API calls (eBay, TMAPI, feeds) retry up to 2 times on a connection error or on HTTP 429, 502, 503 and 504, with a wait of 0.5 then 1 second. An Apify actor call is sent once, because each call starts a paid actor run.

12.4

Cache

A failed search is not cached. If any diagnostic has the code `timeout`, `upstream_error` or `quota`, the result is returned and left out of the cache. The next call then tries the source again.

Cached results live for `PROCUREMENT_CACHE_TTL_S` seconds, 3600 by default. The server also remembers the last 2000 offers in memory, so that `get_offer_detail` can answer for marketplaces with no detail endpoint.

SEE ALSO

- Delivery and landed cost
- Ranking
- Add a marketplace

13

Delivery and landed cost

How the server decides whether an offer reaches the destination and what it costs delivered.

Every *Offer* gets two answers for the *Destination*: `shipping.delivery` says whether and how it arrives, and `landed_aud` says what `qty` units cost delivered. Comps and diagnostics get neither.

13.1

Delivery status

`shipping.delivery_basis` states in words what the status rests on.

DELIVERY	APPLIES TO
`direct`	`ebay_au` items for which eBay lists a shipping option to the postcode. `amazon_au`, `woolworths`, `coles`. `ozbargain` deals whose title says “Delivered”. `facebook` listings that offer shipping.
`pickup`	`gumtree`, `facebook`, `aldi`. Drop these with `include_pickup: false`.
`via_forwarder`	`taobao`, `tmall`, `1688`, `jd`, `pdd`. Sellers ship within China only.
`no`	`cheapies`, which lists New Zealand retailers.
`unknown`	The source does not say.

For `amazon_au`, `woolworths` and `coles` the status is a rule for the whole marketplace. The server does not check the delivery area of one item or one postcode, and `delivery_basis` says so.

Offers with `delivery: no` are left out of the result. The server counts them in one `not_deliverable` diagnostic for the marketplace.

13.2 Landed cost

`landed_aud` is the total for `qty` units. `landed_breakdown` lists its parts, and `landed_basis` says how firm the total is.

LANDED_BASIS	FORMULA	WHEN
`exact`	`price_aud * qty + shipping.cost_aud`	The source stated a shipping cost.
`exact`	`price_aud * qty`	A pickup listing.
`item_only`	`price_aud * qty`	The source stated no shipping cost, so the total leaves it out.
`estimated`	The forwarder estimate below	The delivery status is `via_forwarder`.

An offer with no AUD price has no `landed_aud`.

13.3 Forwarder estimate

A *Forwarder* receives goods in China and ships them to Australia. The server estimates that cost from a rate per kg:

FORWARDER ESTIMATE

```
freight  = max(FORWARDER_MIN_CHARGE_AUD, weight_kg * qty *
FORWARDER_RATE_AUD_PER_KG)
subtotal = goods + China domestic leg + freight
fee      = subtotal * FORWARDER_FEE_PCT
gst      = (subtotal + fee) * GST_RATE
landed   = subtotal + fee + gst
```

`weight_kg` is the `est_weight_kg` argument, or `FORWARDER_DEFAULT_WEIGHT_KG` when the search gives none. Search results carry no weight.

With the default settings, 10 units at 20 CNY each and 0.3 kg each cost:

PART	CALCULATION	AUD
`goods`	20 × 0.215 × 10	43.00
`china_domestic`	none stated	0.00
`forwarder_freight`	max(15, 0.3 × 10 × 12)	36.00
`forwarder_fee`	79.00 × 0.0	0.00
`gst`	79.00 × 0.10	7.90
`landed_aud`		86.90

PLACEHOLDER RATES

The default rate of 12 AUD per kg, the 15 AUD minimum and the 0.5 kg weight are placeholders. An `estimated` total can be far from what you pay. Set the rates from your forwarder's rate card before you compare Chinese offers with Australian ones.

SEE ALSO

- Set forwarder rates
- Ranking
- Offer format

14

Ranking

The three sorts of search_offers and the formula behind best_value.

`search_offers` ranks offers by the `sort` argument: `cheapest`, `fastest` or `best_value`. Comps, diagnostics and offers with `delivery: no` are not ranked.

In every sort, an offer whose `moq` is larger than `qty` goes to the end of the list.

14.1

cheapest

Sorts by `landed_aud`, lowest first. Between two offers with the same cost, one with a known delivery status comes before one with `unknown`.

A grocery offer sorts by `unit_price_aud`. A 2 L bottle at 4.50 AUD (2.25 per litre) then comes before a 1 L bottle at 3.10 AUD.

An offer with no price goes last.

14.2

fastest

Sorts by the later bound of `shipping.eta_days`, then by `price_aud`. An offer with no ETA counts as 99 days.

The ETA windows are fixed per marketplace. Delivery and ETA lists them.

14.3

best_value

Sorts by a score from 0 to 1, highest first:

BEST_VALUE SCORE

```
score = 0.5 * price + 0.25 * speed + 0.25 * trust
score = score - 0.05   when delivery is unknown
```

PART	VALUE
`price`	`1 - min(1, unit cost / 200)`. Unit cost is `landed_aud / qty`, or `price_aud` when there is no landed cost. An offer with neither counts as 50 AUD.
`speed`	`1 - min(1, latest ETA day / 30)`. An offer with no ETA counts as 14 days.
`trust`	The first of the four rules below that applies.

TRUST RULE	VALUE
The offer has deal-site votes	`0.5 + net votes / 20`, held between 0 and 1
The seller has a rating	`rating / 5`
`source_reliability` is `authoritative`	0.85
`source_reliability` is `aggregator`	0.7
None of these	0.5

An item that costs 200 AUD or more per unit scores 0 for price. Above that cost, only speed and trust separate offers.

14.4 Currency

`price_cny` converts with `CNY_TO_AUD` (default 0.215) and Cheapies prices with `NZD_TO_AUD` (default 0.92). Both are fixed settings. The server fetches no live rates.

SEE ALSO

- Delivery and landed cost
- Search for a product

SECTION

Develop

Build, test and release the server.

15

Build and test

Set up a checkout, run the linter, the type checker and the 65 tests, and find your way in the code.

15.1

Commands

CHECK A CHANGE

```
uv sync
uv run ruff check .
uv run pyright procurement server.py
uv run pytest
```

`uv run pytest` needs no network and no keys: HTTP is mocked with `respx`. It prints:

OUTPUT

```
65 passed in 4.59s
```

COMMAND	RESULT
`uv run ruff check .`	Lint. Line length 100. Prints `All checks passed!`.
`uv run pyright procurement server.py`	Type check in strict mode. Prints `0 errors, 0 warnings, 0 informations`.
`uv run pytest`	The test suite, without tests marked `live`.
`uv run pytest -m live`	Reserved for tests that call live sources and spend credits.
`uv run mcp dev server.py`	The MCP Inspector, to call tools by hand.

15.2

Layout

FILES

```
server.py              entry point; imports procurement.app
procurement/
  app.py               assembles the server, main()
  instance.py          the MCPServer instance and database lifespan
  tools.py             the 7 tools and the prompt
  config.py            Settings, read from the environment and .env
  models.py            Offer, Shipping, Location, diagnostics
  lanes/               one module per source; __init__.py is the registry
  lanes/base.py        runs lanes at once under one deadline
  landed.py            delivery status and landed cost
  ranking.py           the three sorts
  keepa.py             price history
  http.py              retry and Apify actor runs
  db.py                Postgres cache and search log
  routes.py            eBay account-deletion notification routes
scripts/live_smoke.py  one live search per marketplace; --record saves fixtures
tests/                 pytest suite, HTTP mocked with respx
migrations/            Postgres schema
docs-src/              the source of these pages
```

15.3

Tests

FILE	COVERS
`tests/test_search.py`	`search_offers` from call to result.
`tests/test_run_lanes.py`	The deadline, groups and one outcome per lane.
`tests/test_lanes.py`	The parser of each lane.
`tests/test_envelopes.py`	Apify error envelopes and the `quota` code.
`tests/test_landed.py`	Delivery status and landed cost.
`tests/test_rank.py`	The three sorts.
`tests/test_deals.py`	The deal feed parser and the query word filter.
`tests/test_detail.py`	`get_offer_detail`.
`tests/test_env_docs.py`	`.env.example` against the settings in `config.py`.

15.4

Documentation

The pages are Markdown in `docs-src/`. `jupiter-docs` generates `docs/*.md` and the site from them. Change a page in the same commit as the code it describes.

CHECK AND BUILD THE PAGES

```
npx jupiter-docs check
npx jupiter-docs build --md
```

SEE ALSO

- Add a marketplace
- Verify a lane

16

Add a marketplace

Write a lane for a new source and register it, so that search_offers and health pick it up.

A *Lane* is two functions and one registry entry. The orchestrator in `procurement/lanes/base.py` handles the deadline, the missing key and the diagnostics.

16.1

Steps

1. Create `procurement/lanes/example.py` with a parser. The parser maps the source's items to `Offer` objects and makes no HTTP call, so a test can feed it a recorded reply.

   PROCUREMENT/LANES/EXAMPLE.PY

   ```
   from typing import Any

   from procurement.common import float_or_none
   from procurement.models import Offer, Seller, Shipping

   def parse_example(items: list[dict[str, Any]], max_results: int) -> list[Offer]:
       return [
           Offer(
               offer_id=f"example:{it['id']}",
               marketplace="example",
               title=str(it["title"]),
               url=it.get("url"),
               price_aud=float_or_none(it.get("price")),
               seller=Seller(shop_name=it.get("seller")),
               shipping=Shipping(eta_days=[2, 6], method="Example post"),
           )
           for it in items[:max_results]
       ]
   ```

2. Add the search function. Raise `LaneError` with a diagnostic code for every failure you can name.

   PROCUREMENT/LANES/EXAMPLE.PY, CONTINUED

   ```
   import httpx

   from procurement.config import settings
   from procurement.http import LaneError, request_retry
   from procurement.lanes.base import SearchCtx

   async def search_example(ctx: SearchCtx) -> list[Offer]:
       async with httpx.AsyncClient(timeout=settings.default_timeout_s) as client:
           r = await request_retry(
               client, "GET", "https://api.example.org/search", params={"q":
               ctx.query}
           )
       if r.status_code != 200:
           raise LaneError("upstream_error", f"example returned HTTP
           {r.status_code}")
       return parse_example(r.json()["items"], ctx.max_results)
   ```

3. Register the lane in `_SPECS` of `procurement/lanes/__init__.py`.

   PROCUREMENT/LANES/__INIT__.PY

   ```
   LaneSpec(
       "example",
       example.search_example,
       "Example search API",
       budget_s=15,
   ),
   ```

4. If the lane needs a key, add the setting to `procurement/config.py` and to `.env.example`. Pass a `missing` function that returns the variable's name when it is unset. `tests/test_env_docs.py` fails when the two files disagree.

5. If the source does not state delivery for each offer, add a rule for the marketplace to `_RULES` in `procurement/landed.py`. Without a rule the offers get `delivery: unknown`.
6. Add a parser test to `tests/test_lanes.py`, then run the checks.

CHECK

```
uv run ruff check . && uv run pyright procurement server.py && uv run pytest
```

16.2

LaneSpec fields

FIELD	DEFAULT	MEANING
`name`	required	The marketplace name a caller passes.
`run`	required	`async (SearchCtx) -> list[Offer]`.
`source`	required	The API or actor, as `health` shows it.
`missing`	no requirement	Returns the name of the unset key, or `None` when the lane can run.
`budget_s`	`10`	The longest the lane may run, in seconds.
`group`	none	Lanes with the same group share one upstream run.
`status`	`unverified`	`verified`, `unverified`, `degraded` or `unsupported`.
`reliability`	`aggregator`	`authoritative`, `aggregator` or `best_effort`. Copied to diagnostics.

16.3

SearchCtx fields

FIELD	MEANING
`query`, `query_zh`	The query. `ctx.china_query` is `query_zh`, or `query` when there is none.
`max_results`, `sort`, `qty`	The arguments of the call.
`location`	The destination: `country`, `postcode`, `lat`, `lng`, `radius_km`.
`est_weight_kg`	The item weight, when given.
`ebay_auctions`, `ebay_seller_scope`	The eBay options.
`ctx.remaining()`	Seconds left before the search deadline.
`ctx.wait_s(cap)`	Seconds one upstream call may take: the cap, or the time left less 3 seconds. At least 5.

16.4

Apify actor lanes

Call `apify_run_sync(actor, run_input, timeout_s=ctx.wait_s(85))` from `procurement/http.py`. It sends the request once, because each call starts a paid run. It raises `LaneError` with `missing_key`, `upstream_error` or `quota`.

Set the lane to `verified` only after `scripts/live_smoke.py` returned offers for it.

SEE ALSO

- Verify a lane
- How a search runs

17

Verify a lane

Run one live search per marketplace with scripts/live_smoke.py and record the replies as test fixtures.

`scripts/live_smoke.py` runs one live `search_offers` call per *Marketplace* and prints one row for each. Use it after you add a key or change a lane.

> APIFY CREDITS
>
> Each Apify lane starts one actor run, which spends Apify credits. Name the marketplaces you want to check to avoid running all of them.

17.1

Run

LIVE CHECK

```
uv run python scripts/live_smoke.py                # every marketplace
uv run python scripts/live_smoke.py ebay_au jd     # a subset
uv run python scripts/live_smoke.py --record       # also save the replies
```

The script searches for “wireless mouse”, and for “full cream milk 2l” on `woolworths`, `coles` and `aldi`.

OUTPUT FOR OZBARGAIN AND CHEAPIES

```
marketplace     outcome      offers priced  diag    sec  note
ozbargain       diagnostic        0      0     1    0.1  [ozbargain] no match for
'wireless mouse'
cheapies        diagnostic        0      0     1    0.3  [cheapies] no match for
'wireless mouse'

0/2 marketplaces returned offers in 0.5s
no offers and no diagnostic: none
```

COLUMN	MEANING
`outcome`	`offers`, `diagnostic`, `empty` (neither) or `raised` (the tool raised an exception).

COLUMN	MEANING
`offers`, `priced`	The number of offers, and how many have a `price_aud`.
`diag`	The number of diagnostics.
`sec`	Seconds the search took.
`note`	The first diagnostic's title.

The last line names every marketplace that returned neither offers nor a diagnostic. It prints `none` when the server works as designed.

17.2 Record fixtures

`--record` writes each upstream reply to `tests/fixtures/<marketplace>/NN.json` and a summary to `tests/fixtures/MANIFEST.json`.

The recorder drops query strings and headers. It replaces the value of every configured secret found in a body with `REDACTED`. A secret is an environment variable of 8 or more characters whose name contains `TOKEN`, `KEY`, `SECRET`, `CERT`, `COOKIES` or `DATABASE_URL`.

Read a recorded file before you commit it. Listings can hold seller names and other personal data.

17.3 Mark a lane verified

1. Run the script for the marketplace and confirm the outcome is `offers`.
2. Record the reply and add a parser test in `tests/test_lanes.py` that reads the fixture.
3. Set `status="verified"` on the lane's `LaneSpec` in `procurement/lanes/__init__.py`.
4. Change the status in `docs-src/reference/marketplaces.md` and in `README.rst`.

SEE ALSO

- Marketplaces
- Add a marketplace