

[DocuCommit](#)

[Projects](#)

Login

Document Navigation

- [Phase 1 — Export from Shopify](#)
- [Phase 2 — Map and validate](#)
- [Phase 3 — Import](#)
- [What we do not migrate](#)

ZephyrCart

- **API reference**
 - **Products**
 - **Orders**
 - **Customers**
 - **Webhooks**
- **Architecture**
 - **How a request flows**
 - **Data model**
 - **Deployment topology**
- **Integrations**
 - **Stripe**
 - **Migrating from Shopify**
 - **ERP bridge**

- **Quickstart**
 - **Installation**
 - **Your first product**
 - **A test checkout, end to end**
- **Intro**
- **Changelog**

[Projects](#)

[ZephyrCart](#)

[Integrations](#)

Migrating from Shopify

Migrating from Shopify

A migration takes between an afternoon (10 SKUs, no historical orders) and several days (100k SKUs, two years of order history). Three phases.

Phase 1 — Export from Shopify

Use Shopify's GraphQL Admin API. The CLI has a helper that runs the exact queries we need:

```
zephyr migrate shopify pull \
  --shop my-shop.myshopify.com \
  --token shpat_... \
  --out ./shopify-export/
```

This writes one JSON file per resource type: `products.jsonl`, `customers.jsonl`, `orders.jsonl`, `collections.jsonl`. Each line is one Shopify record, untouched.

Phase 2 — Map and validate

Mapping is where most of the work lives. The CLI's default rules cover the 80 % case:

Shopify field	ZephyrCart field
<code>product.handle</code>	<code>product.sku</code> (if no other SKU set)
<code>product.product_type</code>	First collection assignment
<code>variant.sku</code>	<code>variant.sku</code>
<code>variant.price</code> (string)	<code>variant.default_price_cents</code> (× 100)
<code>customer.email</code>	<code>customer.email</code>

<code>customer.default_address</code>	<code>customer.addresses[0], is_default: true</code>
<code>order.line_items[]</code>	<code>order.line_items[]</code>

Override any rule with a `mapping.toml` next to the export:

```
[product]
sku_from = "variants[0].sku"

[customer]
exclude_if_marketing_opt_out = false
```

Validate before importing:

```
zephyr migrate shopify validate --in ./shopify-export/ --mapping ./mapping.toml
```

The validator reports field-by-field anything ZephyrCart can't accept (currencies it doesn't recognise, SKUs longer than its limit, etc.). Fix at the source, re-export, re-validate.

Phase 3 — Import

```
zephyr migrate shopify push --in ./shopify-export/ --mapping ./mapping.toml --concurrency 8
```

The CLI streams resources to the API with idempotency keys derived from the Shopify ID, so a re-run of an interrupted import is safe. Progress is checkpointed to `./shopify-export/.zephyr-state`.

What we do not migrate

- **Shopify scripts** — no equivalent in ZephyrCart; checkout rules express most of what scripts do.
- **Shopify Markets** — model the same shape with our [tax zones](#).
- **Custom apps' metafields** — these come through as `product.metadata` keys prefixed with the app handle; review and rename before going live.