

[DocuCommit](#)

[Projects](#)

Login

Document Navigation

- [1. Create a cart](#)
- [2. Attach a customer](#)
- [3. Create a checkout session](#)
- [4. Complete with a test card](#)
- [What just happened](#)

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](#)

[Quickstart](#)

A test checkout, end to end

A test checkout, end to end

In test mode, no money moves and no order email is sent. The flow walks the same code paths as production, including tax calculation and webhook delivery.

1. Create a cart

```
zephyr carts create --line-item prod_01HXY4M9C0AVE0M1QG9S9XK7T2:2
# # cart_01HXY4QH6QMEN0SEN1GHX7YK9C
```

2. Attach a customer

```
zephyr carts attach-customer cart_01HXY4QH6QMEN0SEN1GHX7YK9C \
--email "ali@example.com" \
--shipping-address-de
```

The `--shipping-address-de` flag uses a built-in test address for Germany — good enough to make the tax engine happy.

3. Create a checkout session

```
zephyr checkouts create --cart cart_01HXY4QH6QMEN0SEN1GHX7YK9C
```

The response includes a `checkout_url`. In test mode that URL renders a sandbox payment form.

4. Complete with a test card

Open the `checkout_url`, enter card 4242 4242 4242 4242, any future expiry, any CVC, any postal code. The session completes within ~2 seconds and emits two webhooks:

- `checkout.session.completed`
- `order.created`

See those land:

```
zephyr webhooks tail --event 'order.created'
```

What just happened

A new order resource exists at `GET /v1/orders/{order_id}`. In test mode it's flagged `livemode=false` and is invisible to the production dashboard. Test orders are pruned after 30 days; keep the order id somewhere if you want to refer to it later.

You're done. Read [How a request flows](#) next for the mental model behind the API.