

[DocuCommit](#)

[Projects](#)

Login

Document Navigation

- [Set up your Stripe account](#)
- [Stripe Tax handoff](#)
- [Code sample — TypeScript SDK](#)
- [Refund flow](#)
- [Disputes](#)

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](#)

Stripe

Stripe

Stripe is the default payment provider for new tenants. The integration covers:

- Card payments via Stripe Payment Intents
- Stored payment methods on the customer record
- Stripe Tax handoff for jurisdictions where we don't run our own engine
- Refunds, partial refunds, and disputes

Set up your Stripe account

Two keys, in your tenant's **Settings # Payments # Stripe** page:

Field	Where to find it
Publishable key	Stripe Dashboard # Developers # API keys # "Publishable key"
Secret key	Stripe Dashboard # Developers # API keys # "Secret key" (test or live)
Webhook signing secret	We register the webhook with Stripe on save and store the secret

When you save, ZephyrCart calls Stripe to register a webhook endpoint at `https://api.zephyrcart.io/v1/internal/stripe/webhook` scoped to your account.

Stripe Tax handoff

Inside the EU, ZephyrCart's tax engine handles VAT. In the US sales-tax jurisdictions and a few others, Stripe Tax is the smoother experience. Flip per-tenant:

```
zephyr settings set payments.stripe.use_stripe_tax true
```

Effect:

- Tax is computed on the Stripe side during Payment Intent creation
- The order's `tax_calculation_source` is `stripe_tax` (was `zephyr`)
- The line-item-level breakdown is preserved verbatim from Stripe

Code sample — TypeScript SDK

```
import { Zephyr } from "@zephyrcart/sdk";

const client = new Zephyr({ apiKey: process.env.ZEPHYR_API_KEY! });

const checkout = await client.checkouts.create({
  cart: cartId,
  payment: {
    method: "stripe",
    return_url: "https://shop.example.com/return",
  },
});

// checkout.checkout_url is the hosted page; redirect the customer there
```

Refund flow

A refund triggered through ZephyrCart's API also refunds through Stripe automatically. There is no "refund only in Stripe" mode — refunds must always originate in ZephyrCart so the order record stays the source of truth.

```
curl -sS https://api.zephyrcart.io/v1/orders/order_.../refunds \
  -H "Authorization: Bearer $ZEPHYR_API_KEY" \
  -d '{"amount_cents": 1490, "reason": "requested_by_customer"}'
```

Disputes

When a chargeback is opened in Stripe, we emit `order.disputed`. The order's status is unchanged; the dispute is a sidecar object you query via `GET /v1/orders/{id}/disputes`.