

[DocuCommit](#)

[Projects](#)

Login

Document Navigation

- [Register an endpoint](#)
- [Verify signatures](#)
- [Retries](#)
- [Event catalogue](#)

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](#)

[API reference](#)

Webhooks

Webhooks

ZephyrCart pushes events to endpoints you register. Endpoints must respond 2xx within 10 seconds to count as delivered.

Register an endpoint

```
curl -sS https://api.zephyrcart.io/v1/webhook_endpoints \  
-H "Authorization: Bearer $ZEPHYR_API_KEY" \  
-d '{  
  "url": "https://hooks.example.com/zephyr",  
  "events": ["order.created", "order.refunded"],  
  "description": "ERP bridge - production"  
}'
```

The response includes a `signing_secret` starting `whsec_....`. Store it; you'll need it to verify signatures.

Verify signatures

Every delivery carries:

- `X-ZephyrCart-Delivery-Id` — unique ULID per delivery
- `X-ZephyrCart-Delivery-Attempt` — 1 on first send, increments on retry
- `X-ZephyrCart-Timestamp` — Unix seconds, used in the signature
- `X-ZephyrCart-Signature` — `t=<ts>,v1=<hex-hmac-sha256>`

The signed string is `<timestamp>.<raw-request-body>`. Compute HMAC-SHA256 with the signing secret and compare in constant time:

```

import hmac, hashlib, time

def verify(payload: bytes, header: str, secret: str, tolerance_seconds: int =
300) -> bool:
    parts = dict(p.split("=", 1) for p in header.split(","))
    ts, sig = int(parts["t"]), parts["v1"]
    if abs(time.time() - ts) > tolerance_seconds:
        return False
    expected = hmac.new(
        secret.encode(),
        f"{ts}.".encode() + payload,
        hashlib.sha256,
    ).hexdigest()
    return hmac.compare_digest(expected, sig)

```

A request that fails verification must be rejected with HTTP 401. Do not log the body.

Retries

Deliveries retry on any non-2xx or timeout, with exponential backoff capped at 24 hours total — roughly 1m, 5m, 15m, 1h, 3h, 6h, 12h, then we give up and mark the delivery failed. Persistent failures put the endpoint in `status=failing` after 24 hours; the dashboard shows the last 100 delivery attempts.

Event catalogue

Event	When fired
<code>cart.created</code>	New cart
<code>cart.updated</code>	Line items, addresses, or coupons changed
<code>cart.abandoned</code>	No activity for 60 minutes
<code>checkout.session.created</code>	Hosted checkout URL minted
<code>checkout.session.completed</code>	Payment authorised
<code>order.created</code>	Placed; first webhook after checkout . completed
<code>order.updated</code>	Mutable-field change
<code>order.refunded</code>	Each refund, full or partial
<code>order.cancelled</code>	Before fulfilment
<code>customer.created</code>	
<code>customer.merged</code>	target is folded into source
<code>product.created</code>	
<code>product.archived</code>	Soft-delete
<code>inventory.low</code>	Configurable threshold per product

