

EVE PAY — ERPNext + Frappe Integration Guide (White-Label Edition)

Version 1.0 — January 2026

Overview

EVE Pay integrates seamlessly with **ERPNext / Frappe framework** via REST API calls, enabling:

- Multi-rail payment triggering
- Platform & partner fee computation
- Treasury confirmation
- Ledger reporting into ERPNext DocTypes
- Webhooks + token optionality

The integration is native to Frappe server scripts, client scripts, and background workers.

Base Endpoint

<https://api.evegrocer.app/api/v1>

Primary Use Cases

- Invoice & Sales Order payments
- Subscription billing
- Procurement payout automation
- B2B marketplace transactions
- E-commerce order settlement
- NGO / ESG contribution logging

Core Workflow

- 1 User opens an Invoice, Sales Order, or Payment Request
- 2 ERPNext POSTs data to `/payments/init`
- 3 EVE Pay returns available gateways + fees
- 4 ERPNext shows payment options to user
- 5 Payment processed (Stripe / PayPal / GCash / Coinbase / Bank)
- 6 Confirmation triggers ledger entry

API Endpoints

1 INIT Payment

POST /payments/init

Request Example

```
{  
    "external_ref": "INV-ERP-7721",  
    "amount": 980,  
    "currency": "USD",  
    "customer_id": "ERP-CUST-902",  
    "region": "global"  
}
```

Response

```
{  
    "status": "initialized",  
    "payment_reference": "EVE-170532102",  
    "amount": 980,  
    "currency": "USD",  
    "net_amount": 965.3,  
    "fees": {  
        "platform": 14.7,  
        "partner": 0  
    },  
    "next_action": "collect_payment"  
}  
]
```

ERPNext Integration Strategy

Server Script: REST Client

```
import requests  
import frappe  
  
payload = {  
    "external_ref": doc.name,  
    "amount": doc.grand_total,  
    "currency": doc.currency,  
    "region": "global",  
    "customer_id": doc.customer  
}  
  
res = requests.post(  
    "https://api.evegrocer.app/api/v1/payments/init",  
    json=payload,  
    timeout=10
```

```

)
data = res.json()

doc.payment_reference = data.get("payment_reference")
doc.platform_fee = data["fees"]["platform"]
doc.net_amount = data["net_amount"]
doc.save()

```

Payment Routing Options

Return gateways based on region:

Region	Fintech Rails
US	Stripe, PayPal
PH	GCash, PayMaya
EU	Klarna, SEPA
MENA	Tabby, Tamara
Global	Coinbase

Confirmation Webhook (Optional)

POST /eve-pay/confirm

Payload

```
{
  "orderGid": "ERP-7721",
  "amount": 960,
  "txHash": "0xHASH"
}
```

ERPNext can:

- mark invoice as PAID
- reconcile ledger accounts
- trigger token reward logic

Ledger Sync

Recommended fields inside ERPNext Custom DocType:

Field	Description
external_ref	ERPNext document ID
amount	Invoice total
platform_fee	EVE fee
partner_fee	optional
net_amount	settlement
currency	3-letter
payment_reference	returned ID
gateway	chosen fintech
timestamp	date / time

Requirements

- ERPNext v14+
- REST requests enabled
- Custom field permission setup
- Optional Supabase mirror

Optional Tokenization Layer

ERPNext may activate:

- carbon reward credits
- recycling proof upload
- governance staking
- BNPL token model

Blockchain is **not required** to use EVE Pay.

Why ERPNext Should Use EVE Pay

- ✓ No vendor lock-in
- ✓ No fintech contract required
- ✓ Multi-currency
- ✓ Future-proofed rail system
- ✓ Web3 optionality
- ✓ ESG scoring ready
- ✓ Perfect for India / PH / Asia markets

Ideal Industries

- Retail
- Manufacturing
- Logistics
- SaaS billing
- NGOs
- Marketplaces
- Franchise networks

\Developer Notes

ERPNext scripts should:

- run async
- store payment_reference
- avoid exposing API secrets
- route via server script, not browser

Support

Email: support@evegrocer.app

Hosting: Singapore + Manila

Version: 1.0

License: Commercial Use