Documentation Index

Fetch the complete documentation index at: /llms.txt

Use this file to discover all available pages before exploring further.

Skip to main content
Suby.fi home page



CRYP2NOVA

v3-beta
Search... ⌘K

- Dashboard v2
- Get API key
- Get API key

Search...
Navigation
Get started
Introduction

Documentation
API Reference

Get started

- Introduction
- Authentication
- Quickstart
- Migrating from v2

Archived copy · cryp2nova.com

Core concepts

- Idempotency
- Webhooks
- Errors & responses
- Changelog

On this page

- What is Suby.fi?
- What's new in v3
- Payment methods
- Supported currencies
- Supported crypto networks
- Base URL & environments
- Getting started

Get started

Introduction

The Suby.fi v3 merchant API — accept card and crypto payments, subscriptions, and saved payment methods.

v3-beta. This is the current Suby.fi API. It is RESTful and resource-oriented (plural resources, `POST /v3/payments` rather than `POST /api/payment/create`). The surface is stabilising — breaking changes, if any, are announced in the changelog. Migrating from v2? See the migration guide.

What is Suby.fi?

Suby.fi is a payment platform for one-time and subscription payments. Merchants accept **credit/debit cards + alternative payment methods (APMs)** and **cryptocurrency** payments, with pricing in **USD or EUR**, and can automate access management (Discord, Telegram, and more) post-payment. Whether you run a SaaS product, an online community, or a digital storefront, Suby provides the infrastructure to take payments, manage subscriptions, save customer payment methods, and grant access automatically.

What's new in v3

Area	v2 (legacy)	v3
URL style	`POST /api/payment/create`	`POST /v3/payments` (RESTful, plural)
Version prefix	`/api/*` (unversioned)	`/v3/*`
Saved cards	—	**Setup Intents + saved payment methods**
Hosted checkout	`paymentUrl` only	**Checkout Sessions** (`cs_…`)
Off-session charges	—	Charge a saved payment method directly (`offSession`)
Customers	implicit	first-class **Customer** resource with billing address
Webhooks	dashboard-config only	**Webhook Endpoints** managed via API, with secret rotation
Analytics	—	org / product / customer **analytics** endpoints

Payment methods

You don't configure a fixed list of methods on the product. Instead, the payment method is chosen **when the payment is created**, via the `method` field on `POST /v3/payments` (or on the hosted Checkout Session):

Archived copy · cryp2nova.com

- **Cards & APMs** — `CARD`, plus wallet/APM rails such as `APPLE_PAY`, `GOOGLE_PAY`, `KLARNA`, `IDEAL`, `BANCONTACT`, `TWINT`, `BLIK`, `AFFIRM`, `MULTIBANCO`, `PAYPAL`, `SEPA_DIRECT_DEBIT`, `ACH_DIRECT_DEBIT`. Card/APM instrument details go in the `card` group (e.g. `card.tokenizedInstrument`). A card/APM payment returns a `checkout` handle (redirect / 3DS / client secret) to complete the flow.
- **Crypto** — `method=CRYPTO` (plus the `crypto` group: `crypto.mode`, `crypto.chainId`, `crypto.assetId`). A crypto payment returns an `instruction` (a QR/deposit address, or wallet-connect calldata).

Subscriptions support `CARD`, `APPLE_PAY`, `GOOGLE_PAY`, and `CRYPTO`. Card/wallet subscriptions renew off-session (with automatic smart-retry on failed renewals); crypto subscriptions renew via a per-cycle renew email. Which crypto networks and assets a customer can pay with is configured at the **account level** (per environment), not per product.

Supported currencies

Products are priced in **USD** or **EUR** (the product's `currency`). Card payments are charged in the product's currency; crypto payments are sized to the fiat price at pay time and settled in stablecoins.

Supported crypto networks

Crypto payments settle **same-chain** into stablecoins. Supported mainnet networks:

Network	`chainId`	Assets
Bitcoin	`21`	BTC
Ethereum	`1`	ETH, USDC, USDT
Base	`8453`	USDC, EURC, ETH
Arbitrum	`42161`	USDC, ETH
BNB Chain	`56`	BNB, USDC, USDT
Polygon	`137`	POL, USDC, USDT0
Solana	`101`	SOL, USDC, USDT
Monad	`143`	MON, USDC, USDT0

The exact assets a given account accepts are set in your account configuration. In **sandbox**, crypto is limited to **Base Sepolia** (`chainId` `84532`, USDC/ETH).

Base URL & environments

Environment	Base URL	API key prefix
Production (LIVE)	`https://api.beta.suby.fi`	`sk_live_…`
Sandbox	`https://api.beta.suby.fi`	`sk_sandbox_…`

The environment is derived from your **API key prefix** — a `sk_sandbox_…` key operates entirely in the sandbox (simulated card + crypto, no real funds); a `sk_live_…` key operates in production. All v3 routes live under the `/v3` prefix, e.g. `https://api.beta.suby.fi/v3/payments`. Authenticate every request with the `X-Suby-Api-Key` header.

During the beta, the v3 API is served at **`api.beta.suby.fi`** — `api.suby.fi` keeps serving the legacy v2 API for existing merchants. When v3 reaches GA it will move to `api.suby.fi`; update your base URL then.

Getting started

1. **Create a Suby.fi account** at dashboard.beta.suby.fi.
2. **Generate your API key** from dashboard settings. Store it securely — it is shown once. Use it in the `X-Suby-Api-Key` header.
3. **Create a product** (`POST /v3/products`) — one-time or recurring.
4. **Take a payment** (`POST /v3/payments`) or mint a **hosted checkout session** (`POST /v3/checkout/sessions`).
5. **Register a webhook endpoint** (`POST /v3/webhook-endpoints`) and verify signatures.
6. **Get approved for card payments** if you plan to accept cards (business verification via the dashboard onboarding).

CRYP2NOVA

Archived copy · cryp2nova.com

Authentication

How to authenticate requests and manage LIVE vs sandbox.
↗

Quickstart

Create a product and take your first payment in minutes.
↗

Webhooks

Event types, payloads, and HMAC signature verification.
↗

API Reference

Every endpoint, parameter, and response.

Was this page helpful?

Yes No

Archived copy · cryp2nova.com

[Authentication]

Archived copy · cryp2nova.com

Ask a question...
⌘I

website
discord
Powered by



This documentation is built and hosted on Mintlify, a developer documentation platform

Archived copy · cryp2nova.com