

Stealth Health — Partner API Integration Guide

Version: 0.1 (Draft) **Date:** March 2, 2026 **Base URL:** <https://api.stealth.health> (production) | <https://sandbox.stealth.health> (development) **Status:** Proposal / Scoping

See also: *Clinical Partner API Integration Guide* — a deeper integration where patient data, intake responses, appointment details, and transaction history are shared with the partner under a full BAA.

Table of Contents

1. Overview
2. HIPAA Compliance & PHI Boundaries
3. Authentication & Security
4. Partner Onboarding
5. Integration Flow
6. API Reference
 - 6.1 POST /partner/referrals
 - 6.2 GET /partner/referrals/:referral_id
 - 6.3 GET /partner/referrals
 - 6.4 GET /partner/referrals/summary
 - 6.5 GET /partner/products
 - 6.5a GET /partner/store/products
 - 6.5b POST /partner/store/orders
 - 6.6 GET /partner/referrals/:referral_id/tracking-link
 - 6.7 POST /partner/referrals/:referral_id/cancel
 - 6.8 GET /partner/events
7. Webhook Events
8. Data Models
9. Error Handling
10. Rate Limits
11. Environments
12. Appendix: Status Lifecycle

1. Overview

Stealth Health offers a **white-label telemedicine integration** that allows partner platforms to embed our enrollment, clinical review, and prescription fulfillment pipeline into their own product. The integration works as follows:

1. **Partner** directs their customer to a co-branded Stealth Health enrollment page.
2. **Patient** completes a medical intake questionnaire and (optionally) pays.
3. **Stealth Health doctors** review the intake and approve or deny a prescription.
4. **Stealth Health pharmacy partners** fulfill and ship approved prescriptions.
5. **Partner** receives real-time status updates via webhooks and can query the API for current state.

The partner never handles, stores, or receives protected health information (PHI). All clinical data stays within the Stealth Health platform. Partners interact exclusively with de-identified references and lifecycle status events.

2. HIPAA Compliance & PHI Boundaries

What Partners CAN Access

Data Point	Example
Partner-scoped reference ID	ref_abc123xyz
Enrollment status	enrolled , pending_review , approved , denied
Fulfillment status	awaiting_shipment , shipped , delivered
Timestamp of each status change	2026-03-02T14:30:00Z
Product category enrolled for	trt-cream , peptides
Payment status	payment_due , payment_complete
Tracking carrier (no tracking #)	USPS , UPS , Canada Post

What Partners CANNOT Access

Data Point	Reason
Patient name, DOB, gender	PHI — identity
Address, phone, email	PHI — contact info
Medical conditions, symptoms, allergies	PHI — clinical
Prescription details (medication, dosage, quantity)	PHI — treatment
Doctor notes, intake responses	PHI — clinical
Lab results, ID verification images	PHI — clinical / PII
Tracking numbers, shipping labels	Could be used to correlate PHI

Note: If a partner needs to display fulfillment tracking to their end user, they should direct the user to their Stealth Health patient portal, or we can discuss a tokenized tracking link approach (see [Section 6.6](#)).

Business Associate Agreement (BAA)

A signed BAA between Stealth Health and the partner is **required** before API credentials are issued. The BAA defines each party's obligations under HIPAA and limits the partner's role to a referral source — not a covered entity or business associate handling PHI.

3. Authentication & Security

3.1 API Keys

Each partner receives a **pair of API credentials**:

Credential	Purpose
<code>X-Partner-ID</code>	Identifies the partner (public, safe to log)
<code>X-API-Key</code>	Authenticates the request (secret, never log or expose client-side)

Both must be sent as HTTP headers on every request:

```
GET /partner/referrals HTTP/1.1
Host: api.stealth.health (or sandbox.stealth.health)
X-Partner-ID: ptr_acme_health
X-API-Key: sk_live_7f3a...redacted
Content-Type: application/json
```

3.2 Key Rotation

Partners can request key rotation at any time. When a new key is issued, the old key remains valid for **72 hours** to allow migration. Both keys are accepted during the overlap window.

3.3 Webhook Signature Verification

All webhook payloads include an **X-Stealth-Signature** header containing an HMAC-SHA256 signature of the request body, signed with the partner's **webhook secret** (provided at onboarding, separate from the API key).

```
X-Stealth-Signature: sha256=a1b2c3d4e5f6...
```

Partners **must** verify this signature before processing any webhook payload. See [Section 7](#) for implementation details.

3.4 Transport Security

- All API traffic **must** use TLS 1.2+.
- IP allowlisting is available on request.
- All requests are logged with partner ID, endpoint, timestamp, and response code (no PHI is logged).

4. Partner Onboarding

Step 1: BAA Execution

Sign the Business Associate Agreement.

Step 2: Configuration

Stealth Health provisions the following for each partner:

Item	Description
<code>partner_id</code>	Unique partner identifier (e.g. <code>ptr_acme_health</code>)
API key pair	Live and sandbox credentials
Webhook secret	For verifying webhook signatures
Enrollment slug(s)	Co-branded enrollment URLs returned per referral (e.g. <code>https://app.stealth.health/enroll/trt-cream?ref=ref_abc&wl=acme</code>)
Branding config	Logo, colors, and copy for co-branded enrollment pages
Webhook URL	Partner's HTTPS endpoint for receiving events

Step 3: Enrollment Page Setup

Stealth Health creates **partner-specific enrollment pages** based on the product categories the partner wants to offer. These pages:

- Use the partner's branding (logo, color scheme, custom copy).
- Include the appropriate medical intake questionnaire.
- Are accessible via partner-specific URLs or can be embedded via iframe.
- Automatically tag all submissions with the partner's `partner_id` .

Step 4: Sandbox Testing

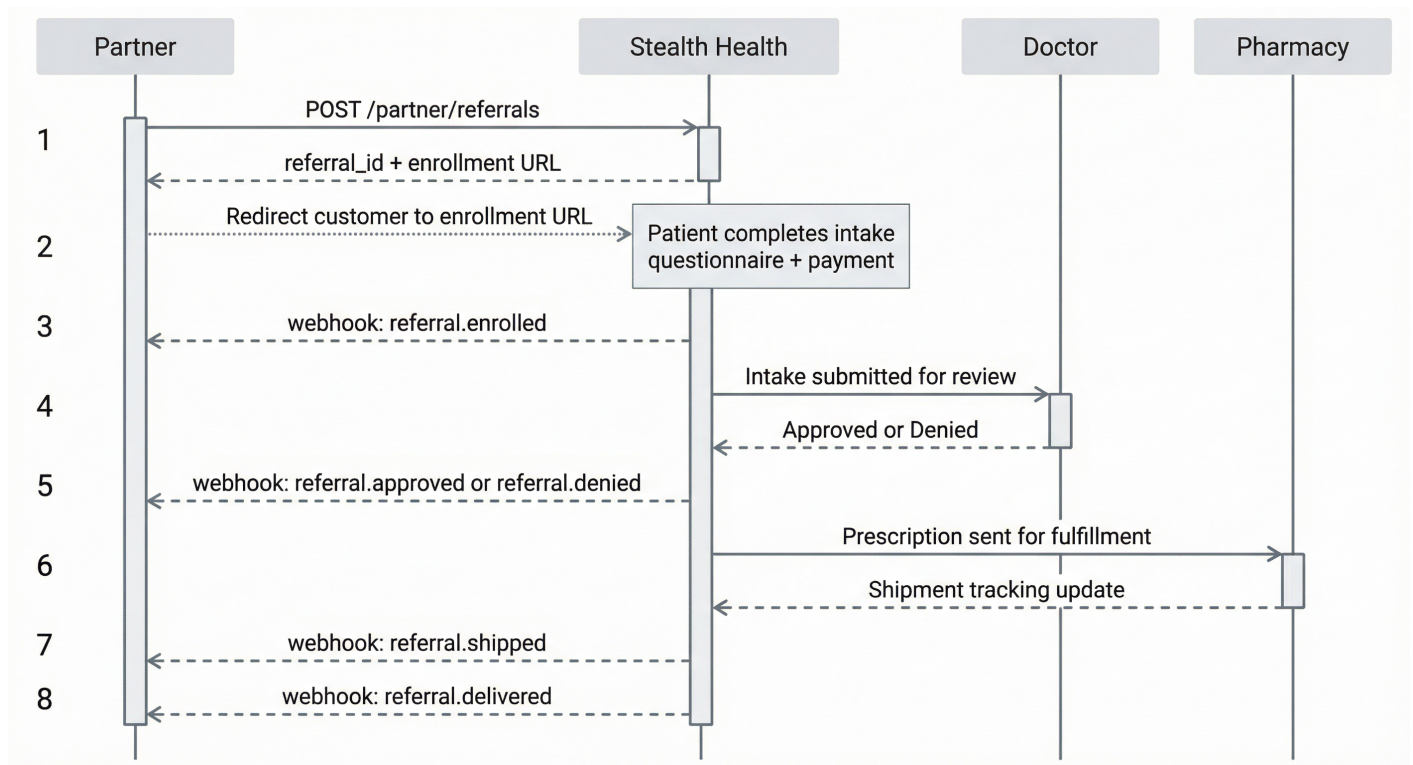
Partners test the full lifecycle in sandbox before going live.

Step 5: Go Live

Switch to production credentials and enrollment URLs.

5. Integration Flow

5.1 Sequence Diagram



5.2 Step-by-Step

Step 1 — Create Referral Partner calls `POST /partner/referrals` with a `partner_reference` (the partner's own customer ID) and the desired `product_category`. Stealth Health returns a `referral_id` and a unique enrollment URL.

Step 2 — Customer Enrollment Partner redirects or links their customer to the enrollment URL. The customer sees a co-branded page and completes the medical intake questionnaire.

Step 3 — Payment If payment is required for the product category, the customer pays during enrollment. Some categories are consult-only (payment after approval).

Step 4 — Clinical Review A licensed Stealth Health physician reviews the intake and either approves a prescription or denies the request (with a reason communicated to the patient, not the partner).

Step 5 — Fulfillment Approved prescriptions are sent to a licensed pharmacy partner for compounding and shipping. Partners receive status updates at each stage.

6. API Reference

All endpoints are prefixed with `/partner`.

Looking for the PHI-bearing endpoint families? Patients, appointments, intake responses, messages, transactions, **lab orders & requisitions** (`GET /partner/patients/:id/lab-orders` , `GET /partner/lab-orders/:id[/requisition|/results]`), patient documents, partner-submitted prescriptions/intake, and the payment modes (`hosted` / `headless` / `embedded_checkout` / `authorize_only`) are **clinical-tier** surfaces and are documented in the *Clinical Partner API Integration Guide* (lab orders: § 12; documents: § 12A; wearables data: § 12B). They are not available to referral-tier keys.

6.1 POST /partner/referrals

Create a new patient referral. Call this **before** sending a customer to enrollment.

Request:

```
{
  "partner_reference": "cust_12345",
  "product_category": "trt-cream",
  "metadata": {
    "campaign": "spring-2026",
    "source": "website"
  }
}
```

Field	Type	Required	Description
<code>partner_reference</code>	string	Yes	Partner's own identifier for this customer. Must be unique per partner. Max 128 chars.
<code>product_category</code>	string	Yes	One of the available categories (see 6.5).
<code>metadata</code>	object	No	Arbitrary key-value pairs for partner's own tracking (max 20 keys, 500 chars per value). Not used by Stealth Health.

Response (201 Created):

```
{
  "referral_id": "ref_abc123xyz",
  "partner_reference": "cust_12345",
  "product_category": "trt-cream",
  "enrollment_url": "https://app.stealth.health/enroll/trt-cream?ref=ref_abc123xyz&wl=acme",
  "status": "created",
  "created_at": "2026-03-02T14:00:00Z",
  "expires_at": "2026-03-09T14:00:00Z"
}
```

Note: Enrollment URLs expire after 7 days. A new referral must be created if the link expires.

6.2 GET /partner/referrals/:referral_id

Retrieve the current status of a referral.

Response (200 OK):

```

{
  "referral_id": "ref_abc123xyz",
  "partner_reference": "cust_12345",
  "product_category": "trt-cream",
  "status": "approved",
  "status_history": [
    { "status": "created",      "at": "2026-03-02T14:00:00Z" },
    { "status": "enrolled",    "at": "2026-03-02T14:15:00Z" },
    { "status": "pending_review", "at": "2026-03-02T14:15:00Z" },
    { "status": "approved",    "at": "2026-03-03T09:30:00Z" }
  ],
  "fulfillment": {
    "status": "shipped",
    "carrier": "USPS",
    "estimated_delivery": "2026-03-07",
    "updated_at": "2026-03-04T11:00:00Z"
  },
  "payment": {
    "status": "paid",
    "amount_cents": 14900,
    "currency": "USD",
    "paid_at": "2026-03-02T14:15:00Z"
  },
  "metadata": {
    "campaign": "spring-2026",
    "source": "website"
  },
  "created_at": "2026-03-02T14:00:00Z"
}

```

6.3 GET /partner/referrals

List all referrals for the authenticated partner.

Query Parameters:

Param	Type	Default	Description
<code>status</code>	string	(all)	Filter by status: <code>created</code> , <code>enrolled</code> , <code>pending_review</code> , <code>approved</code> , <code>denied</code> , <code>expired</code>
<code>fulfillment_status</code>	string	(all)	Filter by fulfillment: <code>awaiting_shipment</code> , <code>shipped</code> , <code>in_transit</code> , <code>delivered</code>
<code>partner_reference</code>	string	—	Look up by partner's own reference
<code>product_category</code>	string	(all)	Filter by category
<code>created_after</code>	ISO 8601	—	Only referrals created after this timestamp
<code>created_before</code>	ISO 8601	—	Only referrals created before this timestamp
<code>limit</code>	integer	50	Max results per page (1–200)
<code>cursor</code>	string	—	Pagination cursor from previous response

Response (200 OK):

```
{
  "referrals": [
    { "referral_id": "ref_abc123xyz", "partner_reference": "cust_12345", "product_category": "trt-cream", "status": "approved", "...": "..." },
    { "referral_id": "ref_def456uvw", "partner_reference": "cust_67890", "product_category": "peptides", "status": "enrolled", "...": "..." }
  ],
  "pagination": {
    "has_more": true,
    "next_cursor": "eyJjcmVhdGVkX2F0Ijo..."
  }
}
```

6.4 GET /partner/referrals/summary

Aggregate counts for dashboard / reporting.

Query Parameters:

Param	Type	Description
<code>period</code>	string	<code>day</code> , <code>week</code> , <code>month</code> , <code>all_time</code> (default: <code>month</code>)
<code>product_category</code>	string	Filter by category

Response (200 OK):

```
{
  "period": "month",
  "start": "2026-02-01T00:00:00Z",
  "end": "2026-02-28T23:59:59Z",
  "totals": {
    "created": 150,
    "enrolled": 120,
    "pending_review": 15,
    "approved": 90,
    "denied": 12,
    "expired": 3,
    "shipped": 75,
    "delivered": 60
  },
  "by_category": {
    "trt-cream": { "created": 80, "approved": 50, "denied": 5 },
    "peptides": { "created": 40, "approved": 30, "denied": 4 }
  },
  "revenue": {
    "total_cents": 756000,
    "currency": "USD"
  }
}
```

6.5 GET /partner/products

List product categories available to this partner.

Response (200 OK):

```

{
  "products": [
    {
      "category": "trt-cream",
      "display_name": "Testosterone Replacement Therapy (Cream)",
      "description": "Compounded testosterone cream for hormone optimization",
      "jurisdictions": ["US"],
      "enrollment_url_template": "https://app.stealth.health/enroll/trt-cream?ref={referral_id}&wl=acme",
      "requires_payment_at_enrollment": true,
      "price_range": {
        "min_cents": 9900,
        "max_cents": 19900,
        "currency": "USD"
      }
    },
    {
      "category": "peptides",
      "display_name": "BPC-157 Peptide Therapy",
      "description": "BPC-157 peptide for tissue repair and recovery support",
      "jurisdictions": ["US"],
      "enrollment_url_template": "https://app.stealth.health/enroll/peptides?ref={referral_id}&wl=acme",
      "requires_payment_at_enrollment": true,
      "price_range": {
        "min_cents": 14900,
        "max_cents": 24900,
        "currency": "USD"
      }
    }
  ]
}

```

6.5a GET /partner/store/products

Return the white label's **merged store catalog** — the global active **store_products** plus this white label's **store_product_overrides** (hidden products removed, re-ranked, and white-label-only products included). It mirrors exactly what the patient storefront would render, so a partner can drive their own checkout UI off the same SKUs. Wholesale fields (**costInCents** , **clinicUsePriceInCents**) are never exposed.

Gated by the **store** feature flag (production). Sandbox is open to every partner. Returns **403 STORE_ACCESS_REQUIRED** in production when the flag is off.

Response (200 OK):

```
{
  "products": [
    {
      "sku": "magnesium-glycinate",
      "name": "Magnesium Glycinate",
      "description": "Highly absorbable magnesium for sleep and recovery.",
      "category": "supplements",
      "categories": ["supplements", "sleep"],
      "image": "https://cdn.stealth.health/store/mag-glycinate.png",
      "price_cents": 2900,
      "currency": "USD",
      "requires_consult": false,
      "badge": "Best seller"
    }
  ]
}
```

***sku** is the stable Firestore document id — use it as the line-item key when pushing an order.*

6.5b POST /partner/store/orders

Push an OTC store order. The server **re-prices every line from the catalog** (**sku** + **quantity** only — a partner-supplied price is never trusted), applies the same free-shipping-over-\$50 / \$5.99-flat rule as the storefront, and routes payment per-order on **payment.mode** :

payment.mode	Behaviour
customer (default)	Mints a hosted payment link (a store_order paymentTransactions doc). The end customer pays; nothing is charged to the partner. Returns the link.
white_label_account	Charges the white label's card on file through the active OTC processor (Authorize.net or Stripe, per config/payments.otcProcessor) and fulfils the order synchronously (store_orders + ShipStation + emails). Requires a card vaulted via the reporting portal.

Request:

```
{
  "items": [
    { "sku": "magnesium-glycinate", "quantity": 2 },
    { "sku": "vitamin-d3-k2", "quantity": 1 }
  ],
  "customer": {
    "first_name": "Jane",
    "last_name": "Doe",
    "email": "jane@example.com",
    "phone": "+15555550123"
  },
  "shipping": {
    "address": "123 Main St",
    "address2": "Apt 4",
    "city": "Austin",
    "state": "TX",
    "zip": "78701",
    "country": "US"
  },
  "currency": "usd",
  "partner_reference": "po_2026_0042",
  "payment": { "mode": "customer" }
}
```

- `items[]` accepts `sku` (preferred), `product_id`, or `id`; `quantity` is clamped to 1–99.
- `customer.email` is required.
- `partner_reference` (≤120 chars) is echoed back and used as the idempotency seed for `white_label_account` charges — reuse the same value to safely retry without double-charging.

Response — `customer` mode (201 Created):

```
{
  "order": {
    "status": "payment_pending",
    "payment_mode": "customer",
    "transaction_id": "aBc123...",
    "payment_link": "https://app.acme.com/enroll/transaction/aBc123...",
    "partner_reference": "po_2026_0042",
    "currency": "usd",
    "subtotal_cents": 8700,
    "items": [
      { "sku": "magnesium-glycinate", "name": "Magnesium Glycinate", "quantity": 2,
"unit_price_cents": 2900 }
    ]
  }
}
```

Response — **white_label_account** mode (201 Created):

```
{
  "order": {
    "status": "confirmed",
    "payment_mode": "white_label_account",
    "order_id": "ord_...",
    "partner_reference": "po_2026_0042",
    "processor": "authorize_net",
    "transaction_ref": "60123456789",
    "currency": "usd",
    "subtotal_cents": 8700,
    "shipping_cents": 0,
    "total_cents": 8700,
    "items": [
      { "sku": "magnesium-glycinate", "name": "Magnesium Glycinate", "quantity": 2,
"unit_price_cents": 2900 }
    ]
  }
}
```

Errors:

Status	Code	Meaning
400	STORE_ORDER_INVALID	Missing <code>items</code> or <code>customer.email</code> .
400	STORE_PAYMENT_MODE_INVALID	<code>payment.mode</code> not <code>customer / white_label_account</code> .
400	STORE_PRODUCT_UNAVAILABLE	None of the requested SKUs are in this white label's store.
402	WHITE_LABEL_CARD_DECLINED	The card on file was declined.
409	WHITE_LABEL_BILLING_NOT_CONFIGURED	No card on file — add one in the reporting portal.
403	STORE_ACCESS_REQUIRED	Production access without the <code>store</code> feature flag.
502	STORE_CHARGE_FAILED	Upstream billing/charge error.

The white label's card on file is captured and managed in the **stealth-reporting** admin portal (White Labels → Card on file). It is tokenised in the browser (Accept.js or Stripe Elements) and vaulted in the active processor; only a masked summary is ever stored.

6.6 GET /partner/referrals/:referral_id/tracking-link

Generate a **tokenized, time-limited tracking link** that the partner can display to their end user. The link opens the Stealth Health patient portal's shipment tracking view without exposing the tracking number to the partner.

Response (200 OK):

```
{
  "referral_id": "ref_abc123xyz",
  "tracking_url": "https://portal.stealth.health/track/tkn_9f8e7d6c5b4a...",
  "expires_at": "2026-03-05T14:00:00Z"
}
```

Returns **404** if no shipment exists for the referral yet.

6.7 POST /partner/referrals/:referral_id/cancel

Request cancellation of a referral. Only possible if the referral has not yet reached **approved** status.

Request:

```
{
  "reason": "Customer requested cancellation"
}
```

Response (200 OK):

```
{
  "referral_id": "ref_abc123xyz",
  "status": "cancelled",
  "cancelled_at": "2026-03-02T16:00:00Z"
}
```

Returns **409 Conflict** if the referral is already approved or further in the lifecycle.

6.8 GET /partner/events

Retrieve webhook events that Stealth Health has emitted for the authenticated partner. Use this endpoint to **replay events** the receiver missed during an outage and to **reconcile** state after the dead-letter queue has fired.

Every event Stealth Health sends to a partner is also persisted in our internal event store with full delivery metadata (attempt count, last HTTP status, last error). This endpoint is the partner-facing read view over that store — scoped to the calling partner's events only.

Use it when:

- Your webhook receiver had a sustained outage and you need to backfill events that exhausted the 6-attempt retry budget.
- You want to verify a specific webhook was actually sent (e.g. a customer support inquiry: "we never got a **referral.shipped** for ref X").
- You need to reconcile downstream state after a deploy that introduced a handler bug.

Query Parameters:

Param	Type	Default	Description
<code>event_type</code>	string	(all)	Filter to one event type, e.g. <code>referral.approved</code> , <code>referral.shipped</code> . Repeat the param to OR multiple types.
<code>referral_id</code>	string	—	Only events for this referral.
<code>partner_reference</code>	string	—	Only events whose underlying referral matches this <code>partner_reference</code> .
<code>delivery_status</code>	string	(all)	<code>delivered</code> (received <code>2xx</code>), <code>pending_retry</code> (in the retry window), <code>failed</code> (exhausted retries — i.e. dead-letter), <code>suppressed</code> (no webhook URL configured).
<code>occurred_after</code>	ISO 8601	24 h ago	Lower bound on <code>created_at</code> . Default keeps the response small.
<code>occurred_before</code>	ISO 8601	now	Upper bound on <code>created_at</code> .
<code>limit</code>	integer	50	Max results per page (1–200).
<code>cursor</code>	string	—	Pagination cursor from a previous response.

Retention. Events remain queryable for **30 days** from `created_at` . Events older than 30 days are purged from the event store; the originating audit-log entries (which are PHI-redacted) are retained for the standard 6-year HIPAA window. If you need durable, long-term storage of every event your integration has ever received, persist them on receipt — do not rely on this endpoint as a system of record.

Response (200 OK):

```

{
  "events": [
    {
      "event_id": "evt_1a2b3c4d",
      "event_type": "referral.shipped",
      "referral_id": "ref_abc123xyz",
      "partner_reference": "cust_12345",
      "data": {
        "carrier": "USPS",
        "estimated_delivery": "2026-03-07",
        "occurred_at": "2026-03-04T11:00:00Z"
      },
      "metadata": {
        "campaign": "spring-2026"
      },
      "delivery": {
        "status": "failed",
        "attempt_count": 6,
        "first_attempted_at": "2026-03-04T11:00:01Z",
        "last_attempted_at": "2026-03-04T23:00:01Z",
        "last_http_status": 502,
        "last_error": "Bad Gateway"
      },
      "created_at": "2026-03-04T11:00:00Z"
    }
  ],
  "pagination": {
    "has_more": false,
    "next_cursor": null
  }
}

```

The top-level event shape is identical to what was originally **POST**-ed to your webhook URL (see § 8.2) with one addition: a **delivery** sub-object describing how that delivery attempt actually went.

Delivery sub-object

Field	Type	Description
<code>status</code>	string	<code>delivered</code> , <code>pending_retry</code> , <code>failed</code> , or <code>suppressed</code> .
<code>attempt_count</code>	integer	Number of HTTPS POST attempts made (0 if <code>suppressed</code>).
<code>first_attempted_at</code>	ISO 8601 null	Timestamp of the first delivery attempt. <code>null</code> if <code>suppressed</code> .
<code>last_attempted_at</code>	ISO 8601 null	Timestamp of the most recent delivery attempt.
<code>last_http_status</code>	integer null	HTTP status returned by the partner endpoint on the last attempt.
<code>last_error</code>	string null	Truncated error message captured on the last attempt (timeout, connection refused, TLS failure, etc.).

Replaying an event

This endpoint is read-only — it does not re-trigger delivery. To request a manual retry of a `failed` or `suppressed` event, contact `partners@stealth.health` with the `event_id` . We will re-queue it into the standard retry pipeline. Replay is rate-limited to 100 events per request and 1,000 events per partner per day.

Errors

Code	Description
<code>INVALID_QUERY_PARAM</code>	One of <code>event_type</code> , <code>delivery_status</code> , or a timestamp param failed validation.
<code>EVENTS_NOT_FOUND</code>	No events match the query. (Returned with HTTP <code>200</code> and an empty <code>events: []</code> ; the code appears in <code>pagination.note</code> for callers that prefer to branch on it.)

7. Webhook Events

7.1 Overview

Stealth Health sends HTTPS POST requests to the partner's registered webhook URL whenever a referral's status changes. Events are delivered **at least once** — partners must handle idempotency.

7.2 Payload Format

```
{
  "event_id": "evt_1a2b3c4d",
  "event_type": "referral.approved",
  "referral_id": "ref_abc123xyz",
  "partner_reference": "cust_12345",
  "data": {
    "status": "approved",
    "product_category": "trt-cream",
    "occurred_at": "2026-03-03T09:30:00Z"
  },
  "metadata": {
    "campaign": "spring-2026"
  },
  "created_at": "2026-03-03T09:30:01Z"
}
```

7.3 Event Types

Event	Trigger	data includes
<code>referral.enrolled</code>	Patient completes intake form	<code>product_category</code>
<code>referral.pending_review</code>	Intake submitted for doctor review	—
<code>referral.approved</code>	Doctor approves prescription	—
<code>referral.denied</code>	Doctor denies prescription	<code>denial_category</code> (e.g. <code>medical_contraindication</code> , <code>incomplete_information</code> , <code>not_a_candidate</code>)
<code>referral.payment_due</code>	Payment required from patient	<code>amount_cents</code> , <code>currency</code>
<code>referral.payment_complete</code>	Patient payment received	<code>amount_cents</code> , <code>currency</code>
<code>referral.awaiting_shipment</code>	Prescription sent to pharmacy	—
<code>referral.shipped</code>	Order shipped by pharmacy	<code>carrier</code> , <code>estimated_delivery</code>
<code>referral.in_transit</code>	Shipment in transit (carrier scan)	<code>carrier</code>
<code>referral.out_for_delivery</code>	Shipment out for delivery	<code>carrier</code>
<code>referral.delivered</code>	Shipment delivered	<code>carrier</code> , <code>delivered_at</code>
<code>referral.delivery_exception</code>	Delivery issue (failed attempt, etc.)	<code>carrier</code> , <code>exception_type</code>
<code>referral.cancelled</code>	Referral cancelled	<code>cancelled_by</code> (<code>partner</code> or <code>patient</code>)
<code>referral.expired</code>	Enrollment URL expired without completion	—

7.4 Signature Verification

```
import hmac
import hashlib

def verify_signature(payload_body: bytes, signature_header: str, webhook_secret: str) -> bool:
    expected = "sha256=" + hmac.new(
        webhook_secret.encode(),
        payload_body,
        hashlib.sha256
    ).hexdigest()
    return hmac.compare_digest(expected, signature_header)
```

7.5 Retry Policy

Attempt	Delay
1	Immediate
2	30 seconds
3	5 minutes
4	30 minutes
5	2 hours
6	12 hours

After 6 failed attempts (non-2xx response or timeout), the event is placed in a **dead letter queue**. Partners can retrieve missed events via **GET /partner/events** (see § 6.8).

Partners must respond with a **2xx** status code within **10 seconds** to acknowledge receipt.

8. Data Models

8.1 Referral Object

```
{
  "referral_id": "string – Stealth Health's unique identifier",
  "partner_reference": "string – Partner's customer identifier",
  "product_category": "string – e.g. trt-cream, peptides",
  "status": "string – See Appendix: Status Lifecycle",
  "status_history": [
    { "status": "string", "at": "ISO 8601 timestamp" }
  ],
  "fulfillment": {
    "status": "string | null",
    "carrier": "string | null",
    "estimated_delivery": "string (YYYY-MM-DD) | null",
    "updated_at": "ISO 8601 timestamp | null"
  },
  "payment": {
    "status": "string – due | paid | refunded | not_applicable",
    "amount_cents": "integer | null",
    "currency": "string – USD | CAD",
    "paid_at": "ISO 8601 timestamp | null"
  },
  "metadata": "object – Partner-provided key-value pairs",
  "created_at": "ISO 8601 timestamp",
  "expires_at": "ISO 8601 timestamp – Enrollment link expiry"
}
```

8.2 Event Object

```
{
  "event_id": "string – Unique event identifier (for idempotency)",
  "event_type": "string – e.g. referral.approved",
  "referral_id": "string",
  "partner_reference": "string",
  "data": "object – Event-specific payload (see Section 7.3)",
  "metadata": "object – Mirror of referral metadata",
  "created_at": "ISO 8601 timestamp"
}
```

9. Error Handling

9.1 HTTP Status Codes

Code	Meaning
200	Success
201	Created
400	Bad request — invalid parameters
401	Unauthorized — invalid or missing API key
403	Forbidden — valid key but insufficient permissions
404	Not found — referral doesn't exist or doesn't belong to this partner
409	Conflict — action not allowed in current state
422	Unprocessable — valid syntax but business rule violation
429	Rate limited
500	Internal server error

9.2 Error Response Format

```
{
  "error": {
    "code": "REFERRAL_NOT_CANCELLABLE",
    "message": "Referral ref_abc123xyz cannot be cancelled because it has already been approved.",
    "details": {
      "referral_id": "ref_abc123xyz",
      "current_status": "approved"
    }
  }
}
```

9.3 Common Error Codes

Code	Description
INVALID_PARTNER_REFERENCE	<code>partner_reference</code> already used or invalid format
INVALID_PRODUCT_CATEGORY	Category not available to this partner
REFERRAL_NOT_FOUND	Referral doesn't exist or unauthorized
REFERRAL_NOT_CANCELLABLE	Referral past the cancellable stage
REFERRAL_EXPIRED	Enrollment link has expired
STORE_ACCESS_REQUIRED	Production store access requires the <code>store</code> feature flag (addendum)
STORE_ORDER_INVALID	Missing/invalid <code>items</code> or <code>customer.email</code> on a store order
STORE_PAYMENT_MODE_INVALID	<code>payment.mode</code> must be <code>customer</code> or <code>white_label_account</code>
STORE_PRODUCT_UNAVAILABLE	Requested SKU(s) not in this white label's store
WHITE_LABEL_BILLING_NOT_CONFIGURED	No card on file for <code>white_label_account</code> charge
WHITE_LABEL_CARD_DECLINED	The white label's card on file was declined
STORE_CHARGE_FAILED	Upstream billing/charge error
RATE_LIMITED	Too many requests

10. Rate Limits

Tier	Requests / minute	Burst
Sandbox	60	10/sec
Production	300	50/sec
Enterprise	Custom	Custom

Rate limit headers are included in every response:

```
X-RateLimit-Limit: 300
X-RateLimit-Remaining: 298
X-RateLimit-Reset: 1709395200
```

11. Environments

Environment	Base URL	Purpose
Sandbox	<code>https://sandbox.stealth.health</code>	Testing with mock data, no real doctors or pharmacy
Production	<code>https://api.stealth.health</code>	Live environment

Sandbox behavior:

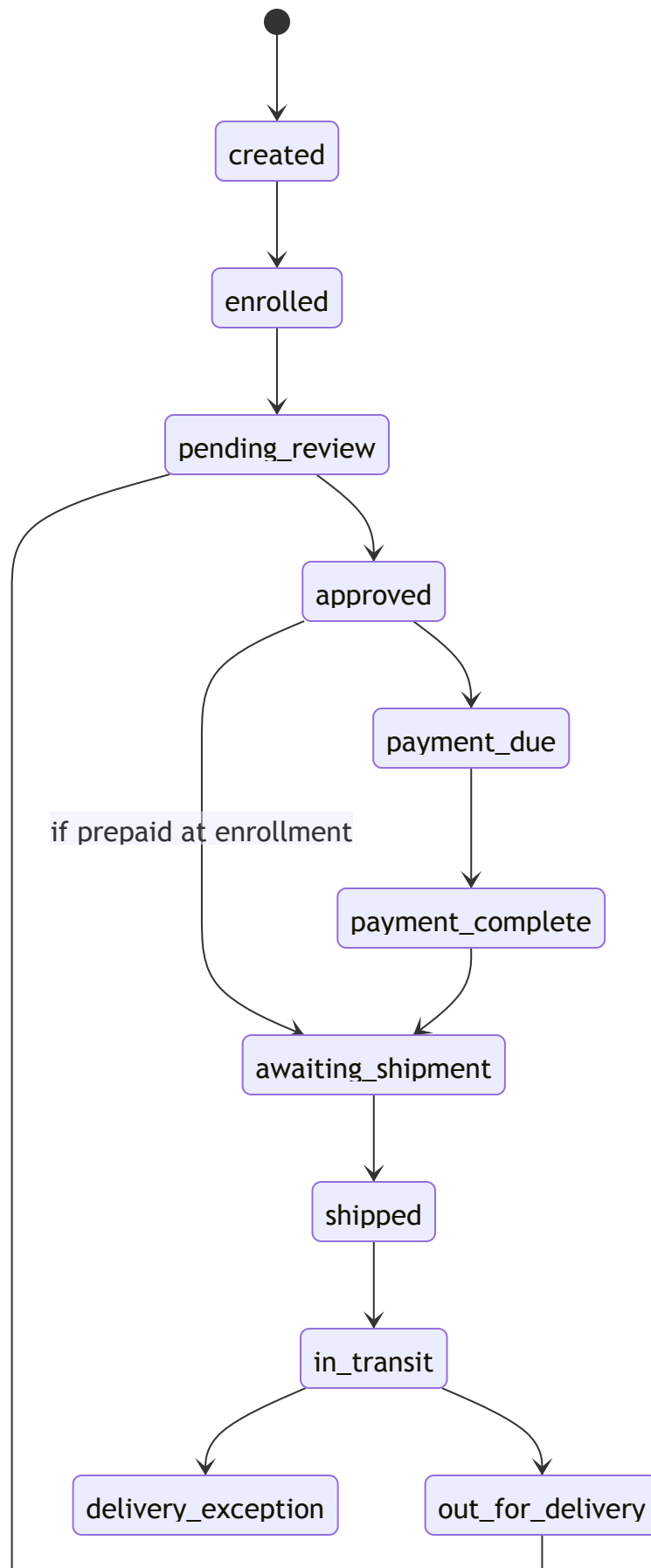
- Referrals auto-advance through the lifecycle every 60 seconds.
- Payment always succeeds with test card `4242 4242 4242 4242`.
- Webhooks are sent to the registered sandbox webhook URL.
- No real clinical review occurs.

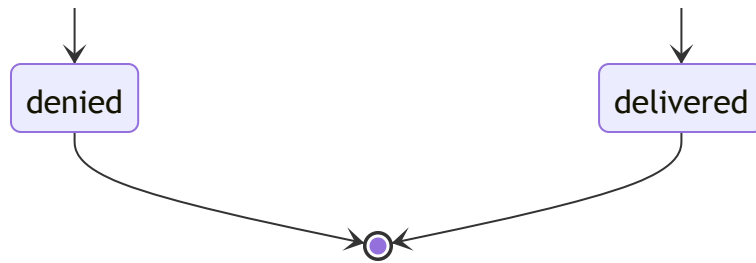
Disabling the lifecycle simulator (for testing real review / sign / reject). The auto-advancer races every referral to `delivered`, which prevents you from exercising a real prescriber review. Two opt-outs are available:

- **Tenant-wide** — set `disable_auto_advance: true` on your `partner_configs` record (ask your Stealth contact, or set it in the reporting portal). The simulator skips your tenant entirely; referrals stay where you (or a real doctor) put them.
- **Per-referral** — set `manual: true` on an individual `partner_referrals` doc to freeze just that case while the rest of your sandbox keeps simulating.

With the simulator off, a case submitted via `POST /partner/appointments` lands in the Stealth doctor review queue and only advances when a Stealth doctor signs (emits `appointment.prescription_signed` for clinical-tier partners) or rejects (emits `referral.denied`).

Appendix: Status Lifecycle





Terminal states: **delivered** , **denied** , **cancelled** , **expired**

Questions for Partner Discussion

1. **Revenue share model** — How will partner compensation be structured? Per-referral fee, revenue share, or flat monthly?
 2. **Branding depth** — Does the partner want full white-label (custom domain, emails from their domain) or co-branded?
 3. **Product categories** — Which categories does the partner want to offer?
 4. **Jurisdiction** — US only, Canada only, or both?
 5. **Patient support** — Will the partner handle first-line patient support, or will Stealth Health?
 6. **Reporting needs** — Does the partner need additional reporting endpoints beyond the summary endpoint?
 7. **Refund handling** — How should refunds be coordinated between partner and Stealth Health?
 8. **Re-enrollment** — Should the same customer be able to enroll for additional product categories, and how should that be linked?
-

This document is a proposal for discussion purposes. Endpoint paths, field names, and behaviors are subject to change during implementation.