



Ledgerly

Business command center

BILLING API

Partner Integration Guide

Everything your engineering team needs to run a complete billing system from your own software — authentication, plans, invoicing, payments & payment links, customers, products, quotations, statements, reports, webhooks, and error handling.

Audience

Partner developers

Base URL

<https://api.ledgerly.biz/api>

Format

JSON over HTTPS

Version

v1

FIELD NAMING All JSON request and response **bodies** use `snake_case` (e.g. `external_reference`, `unit_price`, `grand_total`). **Query-string parameters** use `camelCase` (e.g. `?externalReference=`). All requests must use HTTPS.

1 Base URL

All partner endpoints live under `/v1/partner`. For example, creating an invoice is:

```
POST https://api.ledgerly.biz/api/v1/partner/invoices
```

THE /API PREFIX The application is served under a context path of `/api`. If your deployment sits behind a proxy that strips `/api`, drop it from the base URL. Confirm the exact host with your Ledgerly contact — the `/v1/partner/...` suffix is always the same.

2 Authentication

Every request must include your API key as a Bearer token:

```
Authorization: Bearer lgy_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

- The key is issued once during onboarding and shown only that one time. Store it securely; Ledgerly cannot retrieve it again.
- Your key is mapped to your registered business. You never send a business ID — it is always resolved from the key.
- If a key is compromised, rotate it from the dashboard or contact Ledgerly for revocation.

Verify your key

```
GET /v1/partner/ping
Authorization: Bearer <your-api-key>
```

```
{
  "success": true,
  "partner_account_id": 1,
  "message": "Partner API authentication successful"
}
```

A missing or invalid key returns `401 Unauthorized`.

3 Plans & entitlements

Your account is on one of two subscription bundles, and every API call is gated by the bundle currently active on your account. Billing Suite is a strict superset of Billing Core.

Bundle	Price	Includes
Billing Core	₹999 / month	Invoice create / retrieve / list, PDF, cancel, payment recording, and webhooks
Billing Suite	₹1,999 / month	Everything in Core, plus customers, products, quotations, payment links, statements, payments listing, and reports

ENDPOINT LABELS Endpoints in this guide are labelled **Core** or **Suite**. The full Suite surface is now available: **Customers** (4.7), **Products** (4.8), **Quotations** (4.9), **Payment links** (4.10), **Payments listing** (4.11), **Statements** (4.12) and **Reports** (4.13). Calling a Suite endpoint on a Core plan returns 403 `PLAN_UPGRADE_REQUIRED`.

Trial & activation

- A new partner can start a **14-day free trial** of a chosen bundle from the Ledgerly dashboard (**Partner** → **Overview**).
- When the trial ends — or if you never started one — the API is **blocked** until an active paid subscription exists. Subscribe or upgrade from the same dashboard; billing is handled through Razorpay.
- Partners already integrated before plans were introduced keep Billing Core access automatically.

Plan-related errors

- `403 PLAN_REQUIRED` — no active subscription or trial. Start a trial or subscribe.
- `403 PLAN_UPGRADE_REQUIRED` — your bundle does not include the requested feature; upgrade to Billing Suite.

4 Endpoints

Method	Path	Purpose
GET	/v1/partner/ping	Verify authentication
POST	/v1/partner/invoices	Create and issue an invoice
GET	/v1/partner/invoices/{invoice_id}	Retrieve an invoice
GET	/v1/partner/invoices	List / search your invoices (paginated)
GET	/v1/partner/invoices?externalReference=...	Look up one invoice by your reference
GET	/v1/partner/invoices/{invoice_id}/pdf	Download the invoice PDF
POST	/v1/partner/invoices/{invoice_id}/cancel	Cancel an invoice
POST	/v1/partner/invoices/{invoice_id}/payments	Record a payment (if enabled)
GET	/v1/partner/invoices/{invoice_id}/payments	List payments on an invoice Suite
POST	/v1/partner/invoices/{invoice_id}/payment-link	Generate a payment link Suite
POST	/v1/partner/customers	Create a customer Suite
GET	/v1/partner/customers/{customer_id}	Retrieve a customer Suite
GET	/v1/partner/customers	List / search your customers Suite
PATCH	/v1/partner/customers/{customer_id}	Update a customer Suite
POST	/v1/partner/products	Create a product Suite
GET	/v1/partner/products/{product_id}	Retrieve a product Suite
GET	/v1/partner/products	List / search your products Suite
PATCH	/v1/partner/products/{product_id}	Update a product Suite
POST	/v1/partner/quotations	Create a quotation Suite
GET	/v1/partner/quotations/{quotation_id}	Retrieve a quotation Suite
GET	/v1/partner/quotations	List / search your quotations Suite
POST	/v1/partner/quotations/{quotation_id}/send	Mark a quotation as sent Suite
POST	/v1/partner/quotations/{quotation_id}/accept	Accept a quotation Suite
POST	/v1/partner/quotations/{quotation_id}/reject	Reject a quotation Suite

POST	<code>/v1/partner/quotations/{quotation_id}/cancel</code>	Cancel a quotation Suite
POST	<code>/v1/partner/quotations/{quotation_id}/convert</code>	Convert a quotation to an invoice Suite
GET	<code>/v1/partner/customers/{customer_id}/outstanding</code>	Customer outstanding balance Suite
GET	<code>/v1/partner/customers/{customer_id}/statement</code>	Customer ledger statement Suite
GET	<code>/v1/partner/reports/sales</code>	Sales report Suite
GET	<code>/v1/partner/reports/profit-loss</code>	Profit & loss report Suite
GET	<code>/v1/partner/reports/inventory-valuation</code>	Inventory valuation report Suite

`{invoice_id}` is the Ledgerly public invoice id returned on creation (e.g. `inv_a1b2c3...`); `{customer_id}` / `{product_id}` / `{quotation_id}` are the public customer/product/quotation ids (e.g. `cus_a1b2c3...`, `prd_a1b2c3...`, `qot_a1b2c3...`). Endpoints marked **Suite** require a Billing Suite subscription.

4.1 Create an invoice

```
POST /v1/partner/invoices
Authorization: Bearer <your-api-key>
Idempotency-Key: ORDER-123-INVOICE
Content-Type: application/json
```

The `Idempotency-Key` header is **required** (see section 5).

```
{
  "external_reference": "ORDER-123",
  "invoice_date": "2026-07-18",
  "customer": {
    "name": "Example Customer",
    "phone": "9876543210",
    "email": "customer@example.com",
    "gst_in": "23ABCDE1234F1Z5",
    "billing_address": {
      "line1": "Business Address",
      "city": "Bhopal",
      "state": "Madhya Pradesh",
      "postal_code": "462001"
    }
  },
  "place_of_supply": "Madhya Pradesh",
  "tax_inclusive": false,
  "items": [
    {
      "name": "Sample Product",
      "hsn_sac": "8471",
      "quantity": 2,
      "unit": "PCS",
      "unit_price": 1000,
      "gst_rate": 18,
      "discount": 0
    }
  ],
  "notes": "Thank you for your business."
}
```

Field reference

Field	Req.	Notes
<code>external_reference</code>	Yes	Your own order id. Must be unique for your account.
<code>invoice_date</code>	Yes	<code>YYYY-MM-DD</code> .
<code>customer.name</code>	Yes	
<code>customer.phone</code>	No	Used to match an existing customer; recommended.
<code>customer.email</code> / <code>gstn</code> / <code>billing_address</code>	No	Recorded on the invoice.
<code>place_of_supply</code>	No	State name or 2-digit GST code. Drives CGST/SGST vs IGST.
<code>tax_inclusive</code>	No	<code>true</code> if <code>unit_price</code> already includes GST. Default <code>false</code> .
<code>items[].name</code>	Yes	
<code>items[].hsn_sac</code>	No	HSN/SAC code; shown on the invoice/PDF.
<code>items[].quantity</code>	Yes	Must be positive.
<code>items[].unit_price</code>	Yes	
<code>items[].gst_rate</code>	No	Percentage (e.g. <code>18</code>). Omit for non-GST lines.
<code>items[].discount</code>	No	Flat amount for the line (not a percentage).
<code>additional_charges</code>	No	Untaxed extra (e.g. shipping); added as a separate line.
<code>notes</code>	No	Maximum 100 line items per invoice.

Response — 201 Created

```
{
  "success": true,
  "invoice_id": "inv_a1b2c3d4e5f6g7h8i9j0",
  "external_reference": "ORDER-123",
  "invoice_number": "INV-2026-0041",
  "status": "ISSUED",
  "subtotal": 2000,
  "taxable_amount": 2000,
  "tax_amount": 360,
  "cgst": 180, "sgst": 180, "igst": 0,
  "grand_total": 2360,
  "amount_paid": 0,
  "amount_due": 2360,
  "pdf_url": "/v1/partner/invoices/inv_a1b2c3.../pdf",
  "created_at": "2026-07-18T12:00:00"
}
```

The invoice is created and finalized (issued with a number) in a single call — there is no draft state to manage.

4.3 List & search invoices

```
GET /v1/partner/invoices?page=0&size=20
```

Returns a paginated list, newest first. Optional filters (query params are camelCase / lowercase):

Param	Notes
<code>q</code>	Case-insensitive substring match on <code>external_reference</code> .
<code>from</code> / <code>to</code>	<code>YYYY-MM-DD</code> bounds on creation date.
<code>page</code> / <code>size</code>	Zero-based page number and size (default <code>size=20</code>).

To fetch a **single** invoice by your reference, pass `?externalReference=ORDER-123` — this returns the invoice object, not a list.

4.4 Download the PDF

```
GET /v1/partner/invoices/{invoice_id}/pdf
```

Responds with `302 Found` and a `Location` header to a short-lived signed download URL. Configure your client to follow redirects.

4.5 Cancel an invoice

```
POST /v1/partner/invoices/{invoice_id}/cancel
```

Preserves accounting/audit history; the invoice is not deleted. Idempotent — cancelling an already-cancelled invoice returns its current state. Response has `status: "CANCELLED"`.

4.6 Record a payment optional

Available only if payment recording is enabled for your account; otherwise `403 PAYMENTS_NOT_ENABLED`.

```
{
  "amount": 2360,
  "payment_method": "UPI",
  "payment_date": "2026-07-18",
  "reference_number": "UPI-REF-9988",
  "notes": "Paid in full"
}
```

`amount` is required, positive, and cannot exceed the outstanding balance. `payment_method` defaults to `OTHER` (e.g. `CASH`, `CARD`, `UPI`, `BANK_TRANSFER`).

4.7 Customers Billing Suite

Manage the customers in your business directly. These endpoints require a **Billing Suite** subscription; on a Core plan they return `403 PLAN_UPGRADE_REQUIRED`. Customers are addressed by a public id (`cus_...`) — internal ids are never exposed. Only customers created through this API are returned by these endpoints.

Create a customer — only `name` is required. A duplicate `phone` for your business returns `409 DUPLICATE_CUSTOMER_PHONE`.

```
POST /v1/partner/customers
Content-Type: application/json
```

```
{
  "name": "Acme Corp",
  "phone": "9876543210",
  "email": "accounts@acme.example",
  "gstin": "23ABCDE1234F1Z5",
  "billing_address": {
    "line1": "12 Industrial Area",
    "city": "Bhopal",
    "state": "Madhya Pradesh",
    "postal_code": "462001"
  }
}
```

Response — 201 Created

```
{
  "customer_id": "cus_a1b2c3d4e5f6g7h8i9j0",
  "name": "Acme Corp",
  "phone": "9876543210",
  "email": "accounts@acme.example",
  "gstin": "23ABCDE1234F1Z5",
  "billing_address": { "line1": "12 Industrial Area", "city": "Bhopal", "state": "Madhya Pradesh", "postal_code": "462001" },
  "created_at": "2026-07-23T12:00:00",
  "updated_at": "2026-07-23T12:00:00"
}
```

Retrieve — `GET /v1/partner/customers/{customer_id}` returns the same shape, or `404 CUSTOMER_NOT_FOUND` if it does not belong to your account.

List & search — `GET /v1/partner/customers?page=0&size=20` . Paginated, newest first; optional `q` is a case-insensitive substring match on the customer name.

Update — `PATCH /v1/partner/customers/{customer_id}` . Patch semantics: only the fields you send are changed. Returns the updated customer.

4.8 Products Billing Suite

Manage the product catalogue in your business. These endpoints require a **Billing Suite** subscription; on a Core plan they return `403 PLAN_UPGRADE_REQUIRED` . Products are addressed by a public id (`prd_...`) — internal ids are never exposed. Only products created through this API are returned by these endpoints.

Create a product — `name` and `unit_price` are required. `sku` is auto-generated from name + price when omitted; a duplicate `sku` returns `409 DUPLICATE_SKU` and a duplicate `barcode` returns `409 DUPLICATE_BARCODE` . `unit` accepts an enum name (`PIECE` , `BOTTLE` , `KILOGRAM` , ...) or an abbreviation (`pcs` , `bt1` , `kg` , ...).

```
POST /v1/partner/products
Content-Type: application/json
```

```
{
  "name": "Aamla Tel",
  "description": "Herbal hair oil, 200 ml",
  "sku": "AAMLA-TEL-24",
  "barcode": "8901234567890",
  "hsn_sac": "3305",
  "unit_price": 24,
  "gst_rate": 18,
  "unit": "BOTTLE"
}
```

Response — 201 Created

```
{
  "product_id": "prd_a1b2c3d4e5f6g7h8i9j0",
  "name": "Aamla Tel",
  "description": "Herbal hair oil, 200 ml",
  "sku": "AAMLA-TEL-24",
  "barcode": "8901234567890",
  "hsn_sac": "3305",
  "unit_price": 24,
  "gst_rate": 18,
  "unit": "BOTTLE",
  "created_at": "2026-07-23T12:00:00",
  "updated_at": "2026-07-23T12:00:00"
}
```

Retrieve — `GET /v1/partner/products/{product_id}` returns the same shape, or `404 PRODUCT_NOT_FOUND` if it does not belong to your account.

List & search — `GET /v1/partner/products?page=0&size=20` . Paginated, newest first; optional `q` is a case-insensitive substring match on the product name or SKU.

Update — `PATCH /v1/partner/products/{product_id}` . Patch semantics: only the fields you send are changed. Returns the updated product.

4.9 Quotations **Billing Suite**

Create and manage quotations, then convert an accepted quote straight into an invoice. These endpoints require a **Billing Suite** subscription. Quotations are addressed by a public id (`qot_...`). Only quotations created through this API are returned.

Create a quotation — `customer_id` (`cus_...`) is optional; `valid_until` is required. Items may reference a catalog product via `product_id` (`prd_...`) or be ad-hoc lines with a `name` .

```
POST /v1/partner/quotations
Content-Type: application/json
```

```
{
  "customer_id": "cus_a1b2c3d4e5f6g7h8i9j0",
  "issue_date": "2026-08-01",
  "valid_until": "2026-08-15",
  "items": [
    {
      "product_id": "prd_a1b2c3d4e5f6g7h8i9j0",
      "name": "Aamla Tel",
      "quantity": 10,
      "unit_price": 24,
      "gst_rate": 18,
      "discount": 0
    }
  ],
  "notes": "Valid for 2 weeks."
}
```

Response — 201 Created

```
{
  "quotation_id": "qot_a1b2c3d4e5f6g7h8i9j0",
  "quotation_number": "QU0-2026-0001",
  "customer_name": "Acme Corp",
  "status": "DRAFT",
  "issue_date": "2026-08-01", "valid_until": "2026-08-15",
  "subtotal": 240, "tax_amount": 43.2, "discount_amount": 0, "total_amount": 283.2,
  "converted_invoice_id": null,
  "items": [ { "name": "Aamla Tel", "quantity": 10, "unit_price": 24, "gst_rate": 18,
  "tax_amount": 43.2, "line_total": 283.2 } ],
  "created_at": "2026-08-01T12:00:00"
}
```

Retrieve / list — `GET /v1/partner/quotations/{quotation_id}` and `GET /v1/partner/quotations?page=0&size=20` (newest first; optional `q` matches the number). A missing quotation returns `404 QUOTATION_NOT_FOUND`.

Lifecycle — `POST .../send` (DRAFT→SENT), `.../accept` (SENT→ACCEPTED), `.../reject` (SENT→REJECTED), `.../cancel`. Status values: `DRAFT`, `SENT`, `ACCEPTED`, `REJECTED`, `CANCELLED`, `CONVERTED`.

Convert to an invoice — `POST /v1/partner/quotations/{quotation_id}/convert` creates a draft invoice and returns its reference (retrievable via the invoices API):

```
{
  "quotation_id": "qot_a1b2c3...",
  "quotation_status": "CONVERTED",
  "invoice_id": "inv_z9y8x7...",
  "invoice_number": "INV-2026-0100",
  "invoice_status": "DRAFT"
}
```

4.10 Payment links Billing Suite

Generate a shareable payment link for an invoice. Requires **Billing Suite** and an active payment provider configured on your business; otherwise `403 PAYMENT_PROVIDER_NOT_CONFIGURED`. Both body fields are optional: `amount` defaults to the invoice's outstanding balance, and the provider is resolved automatically.

```
POST /v1/partner/invoices/{invoice_id}/payment-link
Content-Type: application/json
```

```
{
  "amount": 2360,
  "return_url": "https://yourapp.example/thank-you"
}
```

Response — 201 Created

```
{
  "invoice_id": "inv_a1b2c3d4e5f6g7h8i9j0",
  "payment_link": "https://pay.example/link/abc123",
  "amount": 2360,
  "expires_at": "2026-08-02T12:00:00"
}
```

4.11 Payments listing Billing Suite

List the payments (receipts) recorded against an invoice, newest first. Requires **Billing Suite**. `payment_id` is the business receipt number; a cancelled receipt appears with `status: "CANCELLED"` and does not count towards paid totals.

```
GET /v1/partner/invoices/{invoice_id}/payments
```

```
[
  {
    "payment_id": "RCP-2026-0007",
    "invoice_id": "inv_a1b2c3...",
    "amount": 2360,
    "payment_method": "UPI",
    "payment_date": "2026-07-18",
    "reference_number": "UPI-REF-9988",
    "status": "CLEARED",
    "created_at": "2026-07-18T12:00:00"
  }
]
```

4.12 Statements Billing Suite

Read a customer's outstanding balance and full ledger statement. Requires **Billing Suite**. Customers are addressed by their `cus_...` id; a positive balance is money the customer owes you.

```
GET /v1/partner/customers/{customer_id}/outstanding
```

```
{
  "customer_id": "cus_a1b2c3...",
  "name": "Acme Corp",
  "outstanding_balance": 1500
}
```

```
GET /v1/partner/customers/{customer_id}/statement
```

```
{
  "customer_id": "cus_a1b2c3...",
  "name": "Acme Corp",
  "opening_balance": 0, "total_debit": 2360, "total_credit": 860, "closing_balance": 1500,
  "entries": [
    { "date": "2026-07-18", "reference": "INV-2026-0041", "description": "Invoice",
      "debit": 2360, "credit": 0, "balance": 2360 }
  ]
}
```

4.13 Reports **Billing Suite**

Aggregate business reports. Require **Billing Suite**. Breakdowns are name/SKU based; internal ids are not exposed.

Sales report — `GET /v1/partner/reports/sales?from=2026-07-01&to=2026-07-31&reportType=DAILY`. `from` / `to` are required (`from` must not be after `to`); `reportType` groups `sales_by_date` (`DAILY` default, `MONTHLY` , `YEARLY`).

```
{
  "start_date": "2026-07-01", "end_date": "2026-07-31", "report_type": "DAILY",
  "total_sales": 10000, "total_tax": 1800, "total_discount": 0, "net_sales": 8200,
  "invoice_count": 4, "items_sold": 12,
  "sales_by_date": [ { "date": "2026-07-18", "sales": 2360, "net_sales": 2000,
    "invoice_count": 1 } ],
  "top_products": [ { "product_name": "Aamla Tel", "quantity_sold": 6, "total_sales": 6000
  } ],
  "top_customers": [ { "customer_name": "Acme Corp", "invoice_count": 2, "total_sales":
    7000 } ]
}
```

Profit & loss — `GET /v1/partner/reports/profit-loss?from=...&to=...` returns period revenue, purchases, expenses and computed gross/operating/net profit and margin.

Inventory valuation — `GET /v1/partner/reports/inventory-valuation` returns the current total inventory value (cost and selling), potential profit, stock counts, and a per-product breakdown (name, SKU, quantities, prices, stock status).

5 Idempotency & duplicate protection

Invoice creation requires an `Idempotency-Key` header (e.g. `ORDER-123-INVOICE`).

- Same key + **same** payload → returns the original invoice (no duplicate).
- Same key + **different** payload → `409 IDEMPOTENCY_CONFLICT` .
- Repeated `external_reference` (even new key) → `409 DUPLICATE_EXTERNAL_REFERENCE` .

TIP Always send the `Idempotency-Key` so network retries are safe.

6 Rate limits & statuses

60 requests per minute per account (default). Exceeding it returns `429 RATE_LIMITED` with a `Retry-After` header (seconds). Back off and retry after the indicated interval.

Status	Meaning
ISSUED	Created and finalized; awaiting payment.
PARTIALLY_PAID	Some payment recorded, balance remaining.
PAID	Fully paid.
CANCELLED	Cancelled; retained for audit.

7 Webhooks optional

Instead of polling, have Ledgerly notify your server when invoices change. Register HTTPS endpoints from the dashboard (**Partner** → **Webhooks**); a **signing secret** is shown once.

Event	Fires when
<code>invoice.created</code>	An invoice is created & issued
<code>invoice.paid</code>	A payment settles the invoice in full
<code>invoice.cancelled</code>	An invoice is cancelled

Each delivery is an HTTP `POST` with these headers:

Header	Meaning
<code>X-Ledgerly-Event</code>	The event type
<code>X-Ledgerly-Delivery-Id</code>	Unique id for this delivery attempt
<code>X-Ledgerly-Signature</code>	<code>sha256=<HMAC-SHA256 of the raw body, keyed with your secret></code>

```
{
  "event": "invoice.paid",
  "occurred_at": "2026-07-18T12:00:00Z",
  "data": {
    "invoice_id": "inv_a1b2c3...",
    "external_reference": "ORDER-123",
    "status": "PAID",
    "amount_paid": 2360,
    "amount_due": 0,
    "receipt_number": "RCP-2026-0007"
  }
}
```

Verify the signature (Node.js)

```
const crypto = require("crypto");

app.post("/webhooks/ledgerly", express.raw({ type: "application/json" })), (req, res) => {
  const expected = "sha256=" +
    crypto.createHmac("sha256", process.env.LEDGERLY_WEBHOOK_SECRET)
      .update(req.body).digest("hex");
  const sig = req.header("X-Ledgerly-Signature") || "";
  if (sig.length !== expected.length ||
    !crypto.timingSafeEqual(Buffer.from(expected), Buffer.from(sig)))
    return res.status(400).send("bad signature");

  const event = JSON.parse(req.body.toString("utf8"));
  // ... handle event.event / event.data ...
  res.sendStatus(200); // a 2xx marks the delivery delivered
});
```

RETRIES Return a 2xx quickly. Any non-2xx or timeout is retried with exponential backoff (up to 6 attempts) before the delivery is marked failed. Review recent deliveries and their status on the dashboard.

8 Errors

Every non-2xx response from the partner endpoints uses this shape:

```
{
  "success": false,
  "error": {
    "code": "DUPLICATE_EXTERNAL_REFERENCE",
    "message": "An invoice already exists for external_reference ORDER-123.",
    "field_errors": []
  }
}
```

Validation errors include per-field detail in `field_errors`: `[{ "field": "items", "message": "At least one item is required" }]`.

NOTE A 401 is the one exception to this shape — it returns `{ "success": false, "message": "...", "timestamp": "..." }` without an `error` object.

HTTP	Code	Meaning
400	<code>INVALID_REQUEST</code>	Malformed or invalid request.
400	<code>IDEMPOTENCY_KEY_MISSING</code>	Missing <code>Idempotency-Key</code> on create.
400	<code>LINE_ITEM_LIMIT_EXCEEDED</code>	More than 100 line items.
401	—	Missing or invalid API key.
403	<code>PLAN_REQUIRED</code>	No active subscription or trial. Start a trial or subscribe.
403	<code>PLAN_UPGRADE_REQUIRED</code>	Your bundle does not include this feature; upgrade to Billing Suite.
403	<code>PAYMENTS_NOT_ENABLED</code>	Payment recording not enabled for your account.
403	<code>PAYMENT_PROVIDER_NOT_CONFIGURED</code>	No active payment provider configured for payment links (Suite).
404	<code>INVOICE_NOT_FOUND</code>	No such invoice for your account.
404	<code>CUSTOMER_NOT_FOUND</code>	No such customer for your account (Suite).
404	<code>PRODUCT_NOT_FOUND</code>	No such product for your account (Suite).
404	<code>QUOTATION_NOT_FOUND</code>	No such quotation for your account (Suite).
409	<code>IDEMPOTENCY_CONFLICT</code>	Same key reused with a different payload.
409	<code>DUPLICATE_EXTERNAL_REFERENCE</code>	<code>external_reference</code> already used.
409	<code>DUPLICATE_CUSTOMER_PHONE</code>	A customer with that phone already exists (Suite).
409	<code>DUPLICATE_SKU</code>	A product with that SKU already exists (Suite).
409	<code>DUPLICATE_BARCODE</code>	A product with that barcode already exists (Suite).
429	<code>RATE_LIMITED</code>	Too many requests; honor <code>Retry-After</code> .
500	<code>INTERNAL_ERROR</code>	Unexpected server error.

9 Onboarding checklist

1. Provide your business details and a sample invoice to Ledgerly.

2. Receive your production API key (shown once — store it securely).
3. Start a free trial or subscribe to a plan (Core or Suite) from the dashboard (**Partner** → **Overview**) so the API is unblocked.
4. Call `GET /v1/partner/ping` to confirm authentication.
5. Create a test invoice and verify totals, GST split, numbering, and PDF layout.
6. Test cancellation and (if enabled) payment recording.
7. Optionally register a webhook endpoint and verify a test delivery's signature.
8. Go live.

Ledgerly Billing API — Partner Integration Guide · For integration support during business hours, contact your Ledgerly representative.