



PaymentsAPI Platform Team | 2024-10-15



Introduction

PaymentsAPI v2 is a RESTful API that lets you accept payments, issue refunds, and manage customer payment methods programmatically. All requests and responses use JSON, and all endpoints are served exclusively over HTTPS.

This reference covers authentication, rate limits, endpoint specifications, error handling, webhook event delivery, and official SDKs to help you integrate quickly and reliably.

Authentication

Every request must be authenticated using a secret API key. Keys are issued per environment (sandbox and production) from the Developer Dashboard. Never expose secret keys in client-side code.

Header (recommended)	Authorization: Bearer sk_live_XXXXXXXXXXXX
Alternative header	X-API-Key: sk_live_XXXXXXXXXXXX
OAuth2 (server-to-server)	client_credentials grant via /v2/oauth/token

Security Best Practices

- Store secret keys in environment variables or a secrets manager, never in source control.
 - Use restricted/publishable keys on the client and secret keys only on your server.
 - Rotate keys immediately if you suspect a leak, via the Dashboard's Key Management page.
 - Always verify webhook signatures before processing event payloads.
-

Base URL

Production	https://api.paymentsapi.com/v2
Sandbox	https://sandbox.api.paymentsapi.com/v2

Rate Limits

Requests are rate-limited per API key to ensure platform stability. Exceeding the limit returns an HTTP 429 response.

Standard limit	100 requests/minute per API key
Burst allowance	20 requests
Response headers	X-RateLimit-Limit, X-RateLimit-Remaining, X-RateLimit-Reset

Endpoints Overview

Table 1: Core API Endpoints

Method	Path	Description	Auth Required
POST	/v2/payments	Create a new payment	Yes
GET	/v2/payments/{id}	Retrieve a payment by ID	Yes
POST	/v2/refunds	Create a full or partial refund	Yes
GET	/v2/customers/{id}/payment-methods	List a customer's saved payment methods	Yes

Create Payment

Method	POST
Path	/v2/payments
Description	Charges a payment source and creates a payment object.

Body Parameters

Table 2: Create Payment - Parameters

Name	Type	Required	Description
amount	integer	Yes	Amount in minor units (e.g. cents)
currency	string	Yes	ISO 4217 currency code, e.g. AUD
source	string	Yes	Tokenized payment method ID
description	string	No	Free-text description shown in dashboard
capture	boolean	No	Auto-capture funds; default true
metadata	object	No	Arbitrary key-value pairs for your records
idempotency_key	string (header)	Recommended	Prevents duplicate charges on retry

Create Payment

Sample request:

```
curl -X POST https://api.paymentsapi.com/v2/payments \
-H "Authorization: Bearer sk_live_XXXXXXXXXXXX" \
-H "Content-Type: application/json" \
-H "Idempotency-Key: 6c1f2e3a-9b7d-4e2a-8f10" \
-d '{
  "amount": 2500,
  "currency": "AUD",
  "source": "pm_1a2b3c4d",
  "description": "Order #4521",
  "capture": true,
  "metadata": { "order_id": "4521" }
}'
```

Sample Response

```
{
  "id": "pay_9f8e7d6c",
  "object": "payment",
  "amount": 2500,
  "currency": "AUD",
  "status": "succeeded",
  "source": "pm_1a2b3c4d",
  "created": "2024-11-02T04:15:32Z"
}
```

Error Codes

Table 3: Create Payment - Error Codes

HTTP Status	Error Code	Meaning
400	invalid_request	Missing or malformed parameter

HTTP Status	Error Code	Meaning
402	card_declined	Card issuer declined the transaction
409	idempotency_conflict	Idempotency key reused with different parameters
429	rate_limited	Too many requests

Retrieve Payment

Method	GET
Path	/v2/payments/{id}
Description	Fetches the current state of a previously created payment.

Parameters

Table 4: Retrieve Payment - Parameters

Name	Location	Type	Description
id	path	string	Unique payment identifier, e.g. pay_9f8e7d6c
expand	query	string	Comma-separated related objects to expand, e.g. customer

Retrieve Payment

Sample request:

```
curl https://api.paymentsapi.com/v2/payments/pay_9f8e7d6c \
-H "Authorization: Bearer sk_live_XXXXXXXXXXXX"
```

Sample Response

```
{
  "id": "pay_9f8e7d6c",
  "object": "payment",
  "amount": 2500,
  "currency": "AUD",
  "status": "succeeded",
  "source": "pm_1a2b3c4d",
  "created": "2024-11-02T04:15:32Z"
}
```

Error Codes

Table 5: Retrieve Payment - Error Codes

HTTP Status	Error Code	Meaning
401	authentication_error	Missing or invalid API key
404	payment_not_found	No payment exists with the given ID

Create Refund

Method	POST
Path	/v2/refunds
Description	Refunds all or part of a previously succeeded payment.

Body Parameters

Table 6: Create Refund - Parameters

Name	Type	Required	Description
payment_id	string	Yes	ID of the payment to refund
amount	integer	No	Partial refund amount in minor units; omit for full refund
reason	string	No	One of: duplicate, fraudulent, requested_by_customer
metadata	object	No	Arbitrary key-value pairs for your records

Create Refund

Sample request:

```
curl -X POST https://api.paymentsapi.com/v2/refunds \
-H "Authorization: Bearer sk_live_XXXXXXXXXXXX" \
-H "Content-Type: application/json" \
-d '{
  "payment_id": "pay_9f8e7d6c",
  "amount": 1000,
  "reason": "requested_by_customer"
}'
```

Sample Response

```
{
  "id": "rfnd_5c6d7e8f",
  "object": "refund",
  "payment_id": "pay_9f8e7d6c",
  "amount": 1000,
  "status": "succeeded",
  "created": "2024-11-03T09:22:10Z"
}
```

Error Codes

Table 7: Create Refund - Error Codes

HTTP Status	Error Code	Meaning
400	invalid_request	Missing or malformed parameter
404	payment_not_found	No payment exists with the given ID

HTTP Status	Error Code	Meaning
409	refund_amount_exceeds_balance	Requested refund exceeds remaining refundable balance

List Payment Methods

Method	GET
Path	/v2/customers/{id}/payment-methods
Description	Returns a paginated list of a customer's saved payment methods.

Parameters

Table 8: List Payment Methods - Parameters

Name	Location	Type	Description
id	path	string	Customer identifier
limit	query	integer	Number of items to return; default 10, max 100
starting_after	query	string	Pagination cursor, an existing payment method ID

List Payment Methods

Sample request:

```
curl "https://api.paymentsapi.com/v2/customers/cus_7a8b9c/payment-methods?limit=10" \
-H "Authorization: Bearer sk_live_XXXXXXXXXXXXX"
```

Sample Response

```
{
  "object": "list",
  "data": [
    {
      "id": "pm_1a2b3c4d",
      "type": "card",
      "brand": "visa",
      "last4": "4242",
      "exp_month": 12,
      "exp_year": 2027
    }
  ],
  "has_more": false
}
```

Error Codes

Table 9: List Payment Methods - Error Codes

HTTP Status	Error Code	Meaning
401	authentication_error	Missing or invalid API key
404	customer_not_found	No customer exists with the given ID

Error Handling Reference

PaymentsAPI uses conventional HTTP status codes to indicate success or failure. The table below summarizes all error codes returned across the API.

Table 10: Global Error Code Reference

HTTP Status	Error Code	Meaning
400	invalid_request	Malformed or missing parameters
401	authentication_error	Missing or invalid API key
402	card_declined	Card issuer declined the transaction
403	permission_error	API key lacks the required scope
404	resource_missing	Requested object does not exist
409	conflict	Duplicate operation or invalid resource state
429	rate_limited	Exceeded the rate limit
500	api_error	Unexpected server-side fault
503	service_unavailable	Temporary outage; retry with backoff

Webhooks

PaymentsAPI notifies your server of asynchronous events (such as `payment.succeeded`) via HTTP POST requests to a webhook URL you configure in the Dashboard. Your endpoint must respond with a 2xx status within 10 seconds to acknowledge receipt; otherwise the event is retried with exponential backoff for up to 24 hours.

Delivery Sequence

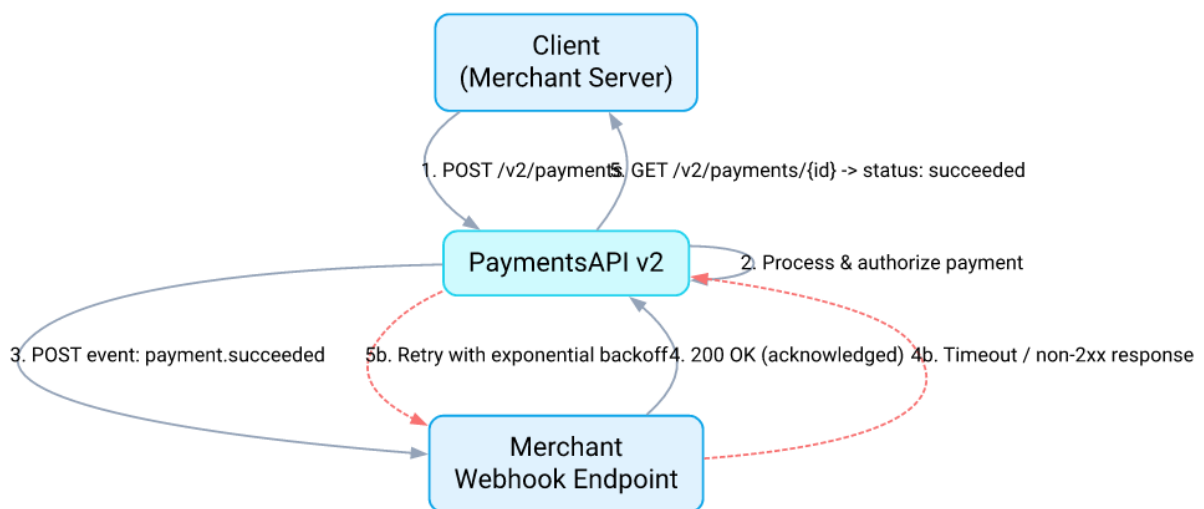


Figure 1: Webhook Delivery and Retry Sequence

The diagram below illustrates a successful payment webhook delivery, along with the retry path triggered by a timeout or non-2xx response.

Changelog

v2.3.0 — 2024-10-15	Added List Payment Methods endpoint; added metadata field to refunds
v2.2.0 — 2024-07-01	Introduced idempotency_key support; improved rate-limit headers
v2.1.0 — 2024-03-20	Added partial refunds; deprecated /v1/charges
v2.0.0 — 2023-11-05	Initial v2 release with OAuth2 client credentials support

SDKs & Resources

Table 11: Official SDKs

Language	Install	Repository
Node.js	npm install paymentsapi	github.com/paymentsapi/sdk-node
Python	pip install paymentsapi	github.com/paymentsapi/sdk-python
Java	Maven: com.paymentsapi:sdk	github.com/paymentsapi/sdk-java
Ruby	gem install paymentsapi	github.com/paymentsapi/sdk-ruby
Go	go get github.com/paymentsapi/sdk-go	github.com/paymentsapi/sdk-go
PHP	composer require paymentsapi/sdk-php	github.com/paymentsapi/sdk-php

Additional Resources

- Postman Collection - importable request library for all endpoints
- Status Page - real-time uptime and incident history
- Support - developer-support@paymentsapi.com