

# Travolyo Partner Integration API

Developer Documentation · OpenAPI 3.1 · version 1.0.0 · partners@travolyo.com

## 1. Overview

The Partner Integration API lets you search and book Travolyo's hotels, activities, and rentals directly from your own platform, over plain REST/JSON. A typical integration covers four stages:

- **Discover** – search a destination for available hotels, activities, and rentals (there is no flat catalog list).
- **Price** – get live availability and rates for specific dates and guests.
- **Book** – lock a price (prebook), then create a confirmed booking.
- **Stay in sync** – receive webhooks for confirmations, cancellations, and price changes.

Base URL: <https://travolyob2b.com/api/partner/v1> · Support: partners@travolyo.com

## 2. Conventions

A few rules apply to every endpoint:

Topic	Rule
Transport	HTTPS only. Request and response bodies are JSON ( <code>Content-Type: application/json</code> on writes).
Dates	ISO-8601 calendar dates, <code>YYYY-MM-DD</code> (e.g. <code>2026-08-01</code> ). <code>check_out</code> must be after <code>check_in</code> .
Money	Each item returns a numeric <code>price</code> / <code>total_price</code> plus a 3-letter <code>currency</code> (e.g. <code>AED</code> ).
Pagination	List endpoints accept <code>?page</code> (default 1) and <code>?per_page</code> (default 25, max 100) and return a <code>meta</code> block.
References	Bookings are identified by a Travolyo <code>reference</code> (e.g. <code>TRV-VILRK1JM</code> ); you may also attach your own <code>partner_reference</code> .

Response envelope – every response has the same shape:

```
// success
{ "success": true, "message": null, "data": { ... }, "meta": { ... } }
// error
{ "success": false, "message": "Human readable message",
  "code": "machine_code", "errors": ["..."] }
```

List endpoints put the array in `data` and paging info in `meta` : `{ "page": 1, "per_page": 25, "total": 134, "total_pages": 6 }` .

## 3. Authentication

Every request must send your partner API key in the `auth-api-key` header. Keys look like `tv1_...` and are issued to you by Travolyo. Each key is either a `test` key or a `live` key (Stripe-style) – see section 13. The `mode` field on the response below tells you which one you're using. Start by confirming your key works:

```
curl https://travolyob2b.com/api/partner/v1/meta/ping \
-H "auth-api-key: tv1_your_key_here"

# 200
{ "success": true, "message": "pong",
  "data": { "partner": "acme-ota", "status": "active", "pricing_mode": "net",
            "mode": "live", "sandbox": false, "products": ["hotel", "activity", "rental"],
            "scopes": ["catalog:read", "availability:read", "booking:read", "booking:write"],
            "server_time": "2026-06-08T05:37:25Z" } }
```

Which mode am I in? `data.mode` is `"test"` or `"live"` depending on the key you sent ( `data.sandbox` is the boolean equivalent and stays for backward compatibility). Test keys never draw real credit.

**Enabled products.** `data.products` lists the product types you may use — a subset of `hotel`, `activity`, `rental`, set from what you requested. Calling `search`, `availability`, `catalog`, `prebook`, or `book` for a product you're not enabled for returns `403 product_not_enabled`.

Depending on your contract, requests may additionally require:

- **IP allowlist** — requests must originate from a registered source IP/CIDR, else `403 ip_not_allowed`.
- **mTLS** — a client certificate verified at the edge, else `403 mtls_required`.
- **HMAC request signing** — see below.

**Scopes.** Each key carries scopes ( `catalog:read`, `availability:read`, `booking:read`, `booking:write`, `webhook:manage` ). Calling an endpoint without its scope returns `403 insufficient_scope` — the required scope is listed per endpoint in section 12.

### Request signing (only if enabled for you)

Build the string "`<timestamp>.<METHOD>.<path>.<raw_body>`", HMAC-SHA256 it with your signing secret (hex output), and send both headers. Requests older than 5 minutes are rejected.

```
# pseudocode
ts = 1812345678          # unix seconds
msg = ts + "." + "POST" + "." + "/api/partner/v1/bookings" + "." + body
sig = hex(hmac_sha256(secret, msg))

# headers
X-Partner-Timestamp: 1812345678
X-Partner-Signature: 9f3a...e1
```

## 4. Pricing & currency

Every price reflects your contract's pricing mode. Only a single, final number is returned per item.

Mode	What you receive
<code>net</code>	A net rate that you mark up before selling to your customer.
<code>markup</code>	Travolyo's ready-to-sell price.

Your mode is shown in `GET /meta/ping` ( `pricing_mode` ) and echoed on availability responses. Prices are quoted in the item's `currency`; convert on your side if you sell in another currency.

## 5. Credit, settlement & idempotency

Bookings are **postpaid**: each confirmed booking draws down your **credit line**, and a cancellation releases it back. Check your balance any time:

```
GET /api/partner/v1/meta/credit
# returns { "data": { "currency": "AED", "credit_limit": 100000.0,
#                   "available": 98950.0, "utilized": 1050.0, "blocked": false } }
```

- If a booking would exceed your available credit you get `402 insufficient_credit`.
- If your account is on hold you get `402 credit_blocked`.

**Idempotency.** `POST /bookings` requires an **Idempotency-Key** header (any unique string per booking attempt — e.g. your order id). Replaying the same key returns the original booking with HTTP `200` instead of creating a duplicate or charging twice. Use it to safely retry on network errors.

## 6. Rate limits

Requests are throttled per partner (default 120/min, set by contract). Over the limit you get `429 rate_limited` with a `Retry-After` header (seconds) — wait that long and retry. Spread bulk catalog syncs out and cache static content on your side.

## 7. Errors

Errors use the standard envelope with a machine-readable `code` you can branch on:

```
{ "success": false, "message": "This API key is missing the required scope: booking:write",  
  "code": "insufficient_scope", "errors": ["This API key is missing the required scope: booking:write"] }
```

HTTP	code	Meaning & what to do
400	<code>bad_request</code> , <code>idempotency_key_required</code>	Fix the payload / add the <code>Idempotency-Key</code> header.
401	<code>unauthorized</code> , <code>signature_required</code> , <code>signature_expired</code> , <code>signature_invalid</code>	Check the API key / signature (and that the signature timestamp is within 5 minutes).
402	<code>insufficient_credit</code> , <code>credit_blocked</code>	Top up / contact us; do not retry blindly.
403	<code>ip_not_allowed</code> , <code>mtls_required</code> , <code>insufficient_scope</code> , <code>product_not_enabled</code>	Request blocked by policy — fix source IP, cert, scope, or request access to that product.
404	<code>not_found</code>	Resource not found or not available to you.
409	<code>duplicate_partner_reference</code>	Reuse the original booking; pick a new <code>partner_reference</code> .
422	<code>invalid_quote</code> , <code>prebook_expired</code> , <code>not_cancellable</code>	Re-run prebook / the booking can't be cancelled in its state.
429	<code>rate_limited</code>	Back off for <code>Retry-After</code> seconds.

## 8. Booking statuses

Status	Meaning
<code>confirmed</code>	Instantly confirmed and ticketed/voucherable. Credit has been drawn.
<code>pending</code> / <code>on_request</code>	Awaiting confirmation; you'll receive a <code>booking.confirmed</code> webhook when it clears.
<code>cancelled</code>	Cancelled; credit released. Fires a <code>booking.cancelled</code> webhook.

The `instant_confirmation` flag on catalog/prebook responses tells you up front whether a booking will be `confirmed` immediately or start as `pending` .

## 9. Step-by-step integration

The flow is search, rooms/rates, prebook, book, and cancel — the same shape as a typical bed-bank integration. Each step below shows the request and the response you can expect.

### Step 1 — Search a destination (dates + occupancy)

Start with a destination and dates; get back the available hotels and a "from" price for the stay.

```
curl -X POST https://travolyob2b.com/api/partner/v1/hotels/search \  
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \  
-d { "destination": "Dubai", "check_in": "2026-08-01", "check_out": "2026-08-04", "adults": 2, "rooms": 1 }  
  
# 200  
{ "success": true, "data": {  
  "destination": "Dubai", "check_in": "2026-08-01", "check_out": "2026-08-04", "nights": 3,  
  "pricing_mode": "net",  
  "hotels": [  
    { "hotel_id": 28, "name": "Sandbox Beach Hotel", "city": "Dubai", "star_rating": 4,  
      "image_url": "https://travolyob2b.com/rails/active_storage/blobs/.../cover.jpg",  
      "board_type": "Bed & Breakfast", "refundable": true, "instant_confirmation": true,  
      "from_price": 1200.0, "currency": "AED" }  
  ]  
},  
  "meta": { "total": 1 } }
```

Activities use `/activities/search` with `{destination, date, adults, children, category?}`; rentals use `/rentals/search` with `{destination, check_in, check_out, guests}` .

## Step 2 – Get rooms & rates for one hotel

Drill into a hotel from the search results to see each bookable room and its total price.

```
curl -X POST https://travolyob2b.com/api/partner/v1/hotels/availability \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-d '{ "hotel_id": 28, "check_in": "2026-08-01", "check_out": "2026-08-04", "adults": 2, "rooms": 1 }'

# 200
{ "success": true, "data": {
  "hotel_id": 28, "check_in": "2026-08-01", "check_out": "2026-08-04", "nights": 3, "pricing_mode": "net",
  "available_rooms": [
    { "room_id": 84, "name": "Standard Double", "board_type": "Bed & Breakfast", "refundable": true,
      "nights": 3, "rooms": 1, "total_price": 1200.0, "currency": "AED" }
  ]
}}
```

Activities use `/activities/availability` ( `{activity_id, date, adults, children, infants}` ); rentals use `/rentals/availability` ( `{rental_id, check_in, check_out, guests}` ).

## Step 3 – Prebook (lock the price for 15 minutes)

```
curl -X POST https://travolyob2b.com/api/partner/v1/bookings/prebook \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-d '{ "product_type": "hotel", "hotel_id": 27, "room_id": 83,
  "check_in": "2026-08-01", "check_out": "2026-08-04", "adults": 2 }'

# 200
{ "success": true, "data": {
  "prebook_token": "pbk_2a01...", "expires_at": "2026-08-01T06:08:29Z",
  "product_type": "hotel", "total_price": 1050.0, "currency": "AED",
  "instant_confirmation": true, "cancellation_policy": "refundable" } }
```

## Step 4 – Book (idempotent, draws credit)

```
curl -X POST https://travolyob2b.com/api/partner/v1/bookings \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-H "Idempotency-Key: ORD-2031-attempt-1" \
-d '{ "prebook_token": "pbk_2a01...", "partner_reference": "ORD-2031",
  "guest": { "name": "Jane Doe", "email": "jane@example.com", "phone": "971501112222" } }'

# 201
{ "success": true, "message": "Booking created", "data": {
  "reference": "TRV-VILRK1JM", "partner_reference": "ORD-2031", "product_type": "hotel",
  "status": "confirmed", "total_price": 1050.0, "currency": "AED",
  "pricing_mode": "net", "test": false,
  "details": { "hotel_id": 27, "check_in": "2026-08-01", "check_out": "2026-08-04",
    "rooms": 1, "guests": 2, "guest_name": "Jane Doe" },
  "created_at": "2026-06-08T05:53:29Z" } }
```

No prebook? You can also pass the full quote fields (`product_type`, `ids`, `dates`, `pax`) directly to `POST /bookings` – it re-validates and re-prices before charging.

## Step 5 – Retrieve or list bookings

```
GET /api/partner/v1/bookings/TRV-VILRK1JM # one booking
GET /api/partner/v1/bookings?page=1&per_page=25 # your bookings (paginated)
```

## Step 6 – Cancel (releases credit)

```
curl -X POST https://travolyob2b.com/api/partner/v1/bookings/TRV-VILRK1JM/cancel \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-d '{ "reason": "Customer changed plans" }'
# returns { "data": { "reference": "TRV-VILRK1JM", "status": "cancelled" } }
# 422 not_cancellable if the booking is already cancelled/completed.
```

## Step 7 – Activity & rental bookings (same flow, different fields)

Activities and rentals use the exact same prebook → book → cancel lifecycle as hotels; only the identifying fields change. The table shows the required `product_type` fields for each.

Product	Identify the item with	Dates / pax
hotel	hotel_id + room_id	check_in , check_out , adults , children , rooms
activity	activity_id	date (single day – no check_out), adults , children , infants
rental	rental_id + space_id	check_in , check_out , guests

**Activity – prebook then book:**

```
# prebook
curl -X POST https://travolyob2b.com/api/partner/v1/bookings/prebook \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-d '{ "product_type": "activity", "activity_id": 20, "date": "2026-08-01", "adults": 2, "children": 1 }'
# 200 → { "data": { "prebook_token": "pbk_7b42...", "product_type": "activity", "total_price": 390.0, "currency": "AED" } }

# book
curl -X POST https://travolyob2b.com/api/partner/v1/bookings \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" -H "Idempotency-Key: ORD-2032-1" \
-d '{ "prebook_token": "pbk_7b42...", "partner_reference": "ORD-2032",
  "guest": { "name": "Jane Doe", "email": "jane@example.com" } }'
# 201 → { "data": { "reference": "ACT-7K2M9XQP", "product_type": "activity", "status": "confirmed",
#   "total_price": 390.0, "currency": "AED", "details": { "activity_id": 20, "date": "2026-08-01" } } }
```

**Rental – prebook then book:**

```
# prebook
curl -X POST https://travolyob2b.com/api/partner/v1/bookings/prebook \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" \
-d '{ "product_type": "rental", "rental_id": 1, "space_id": 1,
  "check_in": "2026-09-01", "check_out": "2026-09-04", "guests": 2 }'
# 200 → { "data": { "prebook_token": "pbk_3921...", "product_type": "rental", "total_price": 1050.0, "currency": "AED" } }

# book
curl -X POST https://travolyob2b.com/api/partner/v1/bookings \
-H "auth-api-key: $KEY" -H "Content-Type: application/json" -H "Idempotency-Key: ORD-2033-1" \
-d '{ "prebook_token": "pbk_3921...", "partner_reference": "ORD-2033",
  "guest": { "name": "Jane Doe", "email": "jane@example.com" } }'
# 201 → { "data": { "reference": "RNT-J3JWAFQJ", "product_type": "rental", "status": "confirmed",
#   "total_price": 1050.0, "currency": "AED",
#   "details": { "rental_property_id": 1, "space_id": 1, "nights": 3, "guests": 2 } } }
```

**Booking without prebook (direct).** Skip the prebook call and pass the full quote fields straight to `POST /bookings` alongside `partner_reference` + `guest` – the API re-validates availability and re-prices before charging. e.g. for a rental: `{ "product_type": "rental", "rental_id": 1, "space_id": 1, "check_in": "2026-09-01", "check_out": "2026-09-04", "guests": 2, "guest": { "name": "Jane Doe" } } .`

## 10. Catalog detail & photos

There is no flat catalog listing – you discover inventory with **Search** (section 9), which returns the matching items for a destination and dates, each with one cover `image_url` . Take the `hotel_id` / `activity_id` / `rental_id` from a search result and fetch the full record – including the complete photo gallery and bookable units (rooms / spaces) – from the detail endpoint.

**Photos are links, never base64.** Every image is returned as an absolute URL in `photos[]` (and a single `image_url` cover). Hot-link them or cache them on your side.

```
GET /api/partner/v1/hotels/27
# 200
{ "success": true, "data": {
  "id": 27, "type": "hotel", "name": "Marina Bay Hotel", "star_rating": 5,
  "city": "Dubai", "country": "UAE", "address": "...", "board_type": "Room Only",
  "facilities": ["Pool", "Spa", "Free Wi-Fi"], "amenities": { ... },
  "guest_rating": 8.6, "reviews_count": 214,
  "image_url": "https://travolyob2b.com/rails/active_storage/blobs/.../cover.jpg",
  "photos": [
    "https://travolyob2b.com/rails/active_storage/blobs/.../1.jpg",
    "https://travolyob2b.com/rails/active_storage/blobs/.../2.jpg",
    "https://travolyob2b.com/rails/active_storage/blobs/.../3.jpg"
  ],
  "currency": "AED", "from_price": 350.0,
  "rooms": [ { "id": 83, "name": "Deluxe King", "board_type": "Bed & Breakfast",
    "max_adults": 2, "refundable": true, "price_per_night": 350.0, "currency": "AED" } ]
}}
```

Endpoint	Returns
GET /hotels/{id}	Full hotel + photos[] + rooms[]
GET /activities/{id}	Full activity + photos[] + pricing
GET /rentals/{id}	Full property + photos[] + spaces[] (each with its own photos[] )

## 11. Webhooks

Subscribe to push events so you don't have to poll. Manage subscriptions with the `webhook:manage` scope:

```
POST /api/partner/v1/webhooks
{ "url": "https://you.example.com/hooks", "event_types": ["booking.confirmed", "booking.cancelled"] }
# returns 201 { "data": { "id": 12, "url": "...", "event_types": [...], "secret": "whsec_..." } } # store the secret
```

Event	Sent when
booking.confirmed	Sent when a booking is confirmed
booking.cancelled	Sent when a booking is cancelled
availability.changed	Sent when an item's availability or price changes

### Delivery & retries

Each event is an HTTPS `POST` of the JSON envelope below. Respond `2xx` to acknowledge. Failed deliveries retry with exponential backoff (about 1m, 5m, 30m, 2h, then 6h – up to 5 attempts). Use `X-Partner-Event-Id` to de-duplicate retries.

Header	Purpose
X-Partner-Event	Event type, e.g. <code>booking.confirmed</code>
X-Partner-Event-Id	Stable id of the event (dedupe key)
X-Partner-Timestamp	Unix seconds when sent
X-Partner-Signature	HMAC-SHA256 signature (verify before trusting)

### Verifying a delivery

```
expected = hex(HMAC_SHA256(subscription_secret, "#{X-Partner-Timestamp}#{raw_body}"))
valid = constant_time_equals(expected, X-Partner-Signature)
# reject if invalid, or if the timestamp is older than a few minutes
```

```
// example payload
{ "id": "evt_9c54...", "type": "booking_confirmed",
  "created_at": "2026-06-08T06:02:59Z", "partner": "acme-ota",
  "data": { "reference": "TRV-VILRK1JM", "partner_reference": "ORD-2031",
    "status": "confirmed", "total_price": 1050.0, "currency": "AED", "test": false } }
```

## 12. Endpoint reference

All endpoints, grouped by category. Search is the entry point; drill into one item with Availability, then book.

### SEARCH

Method	Path	Description	Scope
POST	/api/partner/v1/activities/search	Search a destination for activities on a date	availability:read
POST	/api/partner/v1/hotels/search	Search a destination for available hotels (with a "from" price)	availability:read
POST	/api/partner/v1/rentals/search	Search a destination for available rentals	availability:read

### CATALOG

Method	Path	Description	Scope
GET	/api/partner/v1/activities/{id}	Activity detail (full content & photos)	catalog:read
GET	/api/partner/v1/hotels/{id}	Hotel detail (full content, photos & room types)	catalog:read
GET	/api/partner/v1/rentals/{id}	Rental detail (full content, photos & spaces)	catalog:read

### AVAILABILITY

Method	Path	Description	Scope
POST	/api/partner/v1/activities/availability	Live activity pricing for a date	availability:read
POST	/api/partner/v1/hotels/availability	Live rooms & rates for one hotel	availability:read
POST	/api/partner/v1/rentals/availability	Live rental space availability & rates	availability:read

### BOOKINGS

Method	Path	Description	Scope
GET	/api/partner/v1/bookings	List your bookings	booking:read
POST	/api/partner/v1/bookings	Create a booking (idempotent)	booking:write
POST	/api/partner/v1/bookings/prebook	Validate & price-lock, returns a prebook_token	availability:read
GET	/api/partner/v1/bookings/{reference}	Retrieve a booking	booking:read
POST	/api/partner/v1/bookings/{reference}/cancel	Cancel a booking (releases credit)	booking:write

### WEBHOOKS

Method	Path	Description	Scope
GET	/api/partner/v1/webhooks	List webhook subscriptions	webhook:manage
POST	/api/partner/v1/webhooks	Create a webhook subscription	webhook:manage
DELETE	/api/partner/v1/webhooks/{id}	Delete a webhook subscription	webhook:manage

### META

Method	Path	Description	Scope
--------	------	-------------	-------

GET	/api/partner/v1/meta/credit	Remaining settlement credit line	(none)
GET	/api/partner/v1/meta/ping	Auth / health check	—

Full field-level schemas and a live "try it out" console are on the [Full API reference](#) and [Try it out](#) pages.

### 13. Test & live keys (sandbox)

Test and live are decided per API key, not per account – the same partner holds both a test key and a live key. Which one you send on a request decides the mode of that request; there is no separate base URL or endpoint.

	Test key	Live key
meta/ping → mode	"test"	"live"
Credit	Never drawn	Drawn from your credit line
Booking record	"test": true	"test": false
Pricing	Identical – a test quote returns the same numbers as live, so you validate against real prices	
Webhooks	Fire for both, to your subscribed URL	

Use your test key to exercise the full search → prebook → book → cancel → webhook flow safely. When you're ready, switch the `auth-api-key` header to your live key – no other code changes. Keys are issued and rotated by Travolyo; you can view your keys (and their mode) read-only in the partner portal (section 15).

**Cancellations follow the booking, not the key.** A test booking always releases nothing on cancel; a live booking always releases its credit – regardless of which key cancels it.

### 14. Go-live checklist

- GET /meta/ping returns your partner and expected scopes .
- Source IPs registered (if IP allowlist / mTLS is on your contract).
- Catalog sync caches static content and respects the rate limit.
- Booking uses a unique Idempotency-Key and retries safely.
- You handle 402 (credit), 422 (re-prebook), and 429 ( Retry-After ).
- Webhook endpoint verifies X-Partner-Signature and de-dupes on X-Partner-Event-Id .
- Tested end-to-end with your test key, then switched the auth-api-key header to your live key.

### 15. Partner portal

Alongside the API there is a self-service web portal where your team can sign in and manage the integration without writing code. Get access in two steps:

1. Register at /partner/register with your company details.
2. Once Travolyo approves your account, sign in at /partner/sign\_in with your email and password.

Inside the portal you can:

Section	What you can do
Dashboard	At-a-glance booking counts and available credit for the selected environment.
Bookings	Browse, filter, view, and cancel your bookings.
Test Console	Run a guided search → details → book in test mode – no code, no credit charged.
Credit & Usage	Live credit balance, pricing mode, and rate limit.
Webhooks	View subscriptions and delivery attempts; update your endpoint URL.
Request Logs	Your recent API calls with status codes and errors.

**Test / Live toggle.** A switch in the sidebar flips the whole portal between your Test and Live data, so bookings and stats are never mixed across environments.

# Full endpoint reference

Every endpoint, grouped by category, with required scope, parameters, and request / response examples.

## Search

---

**POST** /api/partner/v1/activities/search

Search a destination for activities on a date

Required scope: `availability:read`

### REQUEST BODY

```
{
  "destination": "Dubai",
  "date": "2026-08-01",
  "adults": 2
}
```

### RESPONSES

200 — OK

400 — Invalid input

---

**POST** /api/partner/v1/hotels/search

Search a destination for available hotels (with a "from" price)

Required scope: `availability:read`

Returns hotels in the destination that have a room available for the dates/occupancy, each with the cheapest "from" price for the stay. Drill into one hotel with /hotels/availability.

### REQUEST BODY

```
{
  "destination": "Dubai",
  "check_in": "2026-08-01",
  "check_out": "2026-08-04",
  "adults": 2,
  "rooms": 1
}
```

### RESPONSES

200 — OK

```
{
  "success": true,
  "data": {
    "destination": "Dubai",
    "check_in": "2026-08-01",
    "check_out": "2026-08-04",
    "nights": 3,
    "pricing_mode": "net",
    "hotels": [
      {
        "hotel_id": 28,
        "name": "Sandbox Beach Hotel",
        "city": "Dubai",
        "country": "UAE",
        "star_rating": 4,
        "image_url": "https://travolyob2b.com/rails/active_storage/blobs/.../cover.jpg",
        "board_type": "Bed & Breakfast",
        "refundable": true,
        "instant_confirmation": true,
        "from_price": 1200.0,
        "currency": "AED"
      }
    ]
  },
  "meta": {
    "total": 1
  }
}
```

400 – Invalid dates

**POST** /api/partner/v1/rentals/search

Search a destination for available rentals

Required scope: [availability:read](#)

REQUEST BODY

```
{
  "destination": "Dubai",
  "check_in": "2026-08-01",
  "check_out": "2026-08-04",
  "guests": 2
}
```

RESPONSES

200 – OK

400 – Invalid dates

## Catalog

---

**GET** /api/partner/v1/activities/{id}

Activity detail (full content & photos)

Required scope: `catalog:read`

Full activity record incl. an array of photo URLs. Find activities with POST /activities/search.

**PARAMETERS**

Name	In	Required	Type	Description
id	path	yes	integer	

**RESPONSES**

200 – OK

404 – Resource not found or not available to you

**GET** /api/partner/v1/hotels/{id}

Hotel detail (full content, photos & room types)

Required scope: `catalog:read`

Full hotel record – description, facilities, amenities, ratings, an array of photo URLs, and bookable room types. There is no flat catalog listing; find hotels with POST /hotels/search.

**PARAMETERS**

Name	In	Required	Type	Description
id	path	yes	integer	

**RESPONSES**

200 – OK

404 – Resource not found or not available to you

**GET** /api/partner/v1/rentals/{id}

Rental detail (full content, photos & spaces)

Required scope: `catalog:read`

Full rental-property record incl. an array of photo URLs and bookable spaces. Find rentals with POST /rentals/search.

**PARAMETERS**

Name	In	Required	Type	Description
id	path	yes	integer	

**RESPONSES**

200 – OK

404 – Resource not found or not available to you

## Availability

---

**POST** /api/partner/v1/activities/availability

### Live activity pricing for a date

Required scope: `availability:read`

#### REQUEST BODY

```
{
  "activity_id": 20,
  "date": "2026-07-01",
  "adults": 2,
  "children": 1
}
```

#### RESPONSES

200 – OK

400 – Invalid input

---

**POST** /api/partner/v1/hotels/availability

### Live rooms & rates for one hotel

Required scope: `availability:read`

#### REQUEST BODY

```
{
  "hotel_id": 27,
  "check_in": "2026-07-01",
  "check_out": "2026-07-04",
  "adults": 2,
  "rooms": 1
}
```

#### RESPONSES

200 – OK

```
{
  "success": true,
  "data": {
    "hotel_id": 27,
    "nights": 3,
    "available_rooms": [
      {
        "room_id": 83,
        "name": "Deluxe King",
        "nights": 3,
        "rooms": 1,
        "total_price": 1050.0,
        "currency": "AED"
      }
    ],
    "pricing_mode": "net"
  }
}
```

400 – Invalid dates

---

**POST** /api/partner/v1/rentals/availability

### Live rental space availability & rates

Required scope: `availability:read`

#### REQUEST BODY

```
{
  "rental_id": 1,
  "check_in": "2026-07-01",
  "check_out": "2026-07-05",
  "guests": 2
}
```

#### RESPONSES

200 – OK

400 – Invalid dates

## Bookings

**GET** /api/partner/v1/bookings

### List your bookings

Required scope: `booking:read`

#### PARAMETERS

Name	In	Required	Type	Description
<code>page</code>	query	no	integer	
<code>per_page</code>	query	no	integer	

#### RESPONSES

200 – OK

**POST** /api/partner/v1/bookings

### Create a booking (idempotent)

Required scope: `booking:write`

Draws down your credit line and creates the booking. Requires an `Idempotency-Key` header; replaying it returns the original booking.

#### PARAMETERS

Name	In	Required	Type	Description
<code>Idempotency-Key</code>	header	yes	string	

#### RESPONSES

201 – Booking created

200 – Idempotent replay – returns the original booking (no second charge)

402 – Insufficient credit / credit blocked

409 – Duplicate partner\_reference

422 – Invalid quote / prebook expired

**POST** /api/partner/v1/bookings/prebook

Validate & price-lock, returns a prebook\_token

Required scope: `availability:read`

Re-checks availability and locks the price for 15 minutes. Pass the returned `prebook\_token` to `POST /bookings`.

#### RESPONSES

200 – Prebook token + locked price (15-minute TTL)

422 – Item unavailable / invalid quote

**GET** /api/partner/v1/bookings/{reference}

Retrieve a booking

Required scope: `booking:read`

#### PARAMETERS

Name	In	Required	Type	Description
<code>reference</code>	path	yes	string	

#### RESPONSES

200 – OK

404 – Resource not found or not available to you

**POST** /api/partner/v1/bookings/{reference}/cancel

Cancel a booking (releases credit)

Required scope: `booking:write`

#### PARAMETERS

Name	In	Required	Type	Description
<code>reference</code>	path	yes	string	

#### RESPONSES

200 – Cancelled

422 – Not cancellable in current state

## Webhooks

**GET** /api/partner/v1/webhooks

List webhook subscriptions

Required scope: `webhook:manage`

#### RESPONSES

200 – OK

**POST** /api/partner/v1/webhooks

**Create a webhook subscription**

Required scope: **webhook:manage**

Returns a signing `secret` once – store it to verify `X-Partner-Signature` on delivered events.

**REQUEST BODY**

```
{
  "url": "https://partner.example.com/webhooks",
  "event_types": [
    "booking.confirmed",
    "booking.cancelled"
  ]
}
```

**RESPONSES**

201 – Created (returns signing secret once)

422 – Invalid url / event types

**DELETE** /api/partner/v1/webhooks/{id}

**Delete a webhook subscription**

Required scope: **webhook:manage**

**PARAMETERS**

Name	In	Required	Type	Description
id	path	yes	integer	

**RESPONSES**

200 – Deleted

404 – Resource not found or not available to you

## Meta

**GET** /api/partner/v1/meta/credit

**Remaining settlement credit line**

Required scope: –

**RESPONSES**

200 – OK

```
{
  "success": true,
  "data": {
    "currency": "AED",
    "credit_limit": 100000.0,
    "available": 98950.0,
    "utilized": 1050.0,
    "blocked": false
  }
}
```

404 – Resource not found or not available to you

**GET** /api/partner/v1/meta/ping

### Auth / health check

Required scope: —

Verifies your credentials and returns your partner context. `mode` is `test` or `live` depending on the API key you sent (`sandbox` is the boolean equivalent, kept for backward compatibility). Test keys never draw real credit.

### RESPONSES

#### 200 — OK

```
{
  "success": true,
  "message": "pong",
  "data": {
    "partner": "acme-ota",
    "status": "active",
    "pricing_mode": "net",
    "mode": "live",
    "sandbox": false,
    "products": [
      "hotel",
      "activity",
      "rental"
    ],
    "scopes": [
      "catalog:read",
      "booking:write"
    ],
    "server_time": "2026-06-08T05:37:25Z"
  }
}
```

#### 401 — Missing/invalid/expired API key or signature

---