API REFERENCE

THE AVEO EOS API.

Every endpoint, object, permission, error and webhook event, as at version 2026-11-15. The same reference is at aveoevents.com/eos/api, and the API serves it as OpenAPI.

37	10	49	20
ENDPOINTS	OBJECTS	ERROR CODES	WEBHOOK EVENTS

CONTENTS

IN THIS DOCUMENT.

01 **THE BASICS**
Address, keys, permissions, plans and errors.

02 **THE API KEY**
Who you are, what your key may do, and the company it belongs to.

03 **EVENTS**
Your events with their dates, venue and ticket types, and live availability.

04 **CHECKOUT**
Sell tickets from your own site: hold them, take the buyer's details, then payment.

05 **ORDERS**
Every order, paid or not: read them, cancel, refund, mark paid and resend tickets.

06 **GUESTS**
Ticket holders for an event: names, emails, ticket types and whether they are inside.

07 **CHECK-INS**
Scan tickets in and out from your own door app, with gates and re-entry.

08 **DISCOUNT CODES**
Codes buyers type at checkout, on one event or many.

09 **ENQUIRIES**
Messages from your own contact form, straight into Customer cases.

10 **REPORTS**
Sales by day for one event or all of them.

11 **WEBHOOK ENDPOINTS**
Addresses EOS calls when something happens, signed so you can trust them.

12 **OBJECTS**
Every object the API returns, field by field.

13 **WEBHOOK EVENTS**
What each event means, and the object it carries.

01 THE BASICS.

BASE ADDRESS	https://eos.aveoevents.com/api/v1
AUTHENTICATION	Authorization: Bearer <key>. Secret keys (eos_sk_) belong on your server; publishable keys (eos_pk_) may be used in a browser for the company, events and checkout only.
SANDBOX	Keys starting eos_sk_test_ or eos_pk_test_ reach your sandbox, with Stripe in test mode.
FORMAT	Requests and answers are JSON. Lists take limit (1 to 100) and cursor, and give back has_more and next_cursor.
IDEMPOTENCY	Writes that create something take an Idempotency-Key header; the first answer is kept for 24 hours.
RATE LIMITS	Each key has a request limit a minute (X-RateLimit-Limit); your plan sets the most a key may have.
VERSION	AVEO-Version: 2026-11-15, on every answer.
OPENAPI	https://eos.aveoevents.com/api/v1/openapi.json

PERMISSIONS.

PERMISSION	LETS A KEY
company:read	Read the company profile: Name, shop address, currency and contact details. Publishable keys too.
events:read	Read events: Events, dates, venues and ticket types with live availability. Publishable keys too.
events:write	Publish events: Put events on sale or take them off sale.
checkout:write	Create checkouts: Hold tickets for a buyer and take them to payment. Publishable keys too.
orders:read	Read orders: Orders, items, buyers, payments and totals.
orders:write	Manage orders: Cancel, refund, mark paid and resend tickets.
attendees:read	Read guests: Ticket holders and whether they are inside.
attendees:write	Update guests: Change a ticket holder's name or email and resend their ticket.

PERMISSION	LETS A KEY
checkins:read	Read check-ins: Every scan at the door, with the gate and the result.
checkins:write	Check guests in: Scan tickets in and out at the door.
discounts:read	Read discount codes: Codes, their discounts, limits and uses.
discounts:write	Manage discount codes: Create codes on events and remove them.
enquiries:write	Send enquiries: Pass messages from your own contact form into Customer cases.
reports:read	Read sales reports: Orders, tickets and revenue by day.
webhooks:read	Read webhook endpoints: Endpoints and their delivery log.
webhooks:write	Manage webhook endpoints: Add, change, test and remove endpoints, and roll their secrets.

BY PLAN.

PLAN	API	WEBHOOK ENDPOINTS	REQUESTS A MINUTE, EACH KEY
Starter	Website checkout	None	60
Professional	Standard	2	120
Business	Full	10	300
Enterprise	Full	25	600
Corporate	Custom	As agreed	1200

Website checkout: read the company and events, and start checkouts. Standard: also read orders, guests, check-ins, discounts and reports, send enquiries and manage webhooks. Full: every permission. Corporate: set for you.

THE BASICS

ERRORS.

TYPE	STATUS	WHEN
invalid_request_error	400, 404, 409, 422	Something in the request is wrong: a missing or bad parameter, an unknown id, or a state the action does not allow.
authentication_error	401	No key, a key that does not exist, or one that has expired or been revoked.
permission_error	403	The key is fine but lacks the permission, or the address or web page it was used from is not allowed.
idempotency_error	409	An Idempotency-Key was reused with a different request.
rate_limit_error	429	Too many requests for this key in the last minute. Wait for the time in Retry-After.
api_error	500, 502	Something went wrong on our side or in the ticket engine. Safe to retry with the same Idempotency-Key.

CODE	STATUS	WHAT IT MEANS
missing_key	401	No key was sent. Send it as Authorization: Bearer followed by the key.
key_malformed	401	What was sent is not an AVEO EOS API key.
key_unknown	401	There is no such key. Check that all of it was copied.
key_revoked	401	The key was revoked. Make a new one on the API keys page.
key_expired	401	The key has expired. Make a new one on the API keys page.
key_wrong_company	401	The key belongs to another company.
company_not_active	403	The company's AVEO EOS account is not active.
missing_scope	403	The key does not have the permission this needs. param names the permission.
plan_scope	403	The company's AVEO EOS plan does not include this permission, whatever the key holds. param names it; a bigger plan opens it.
publishable_key	403	A publishable key cannot do this. Call it from your server with a secret key.
secret_key_in_browser	403	A secret key was sent from a web page (the request had an Origin header). Use a publishable key in web pages.
origin_not_allowed	403	The web page's address is not one of this publishable key's web addresses.
ip_not_allowed	403	The request came from an IP address the key is not allowed from.
parameter_missing	400	A required parameter is missing. param names it.
parameter_invalid	400	A parameter has the wrong type or form. param names it and message says what is expected.
invalid_json	400	The body is not a JSON object.
body_too_large	400	The body is over 64 KB.

CODE	STATUS	WHAT IT MEANS
invalid_limit	400	limit must be a whole number from 1 to 100.
invalid_cursor	400	The cursor is not one this API gave you.
invalid_idempotency_key	400	Idempotency-Key must be 1 to 255 characters.
not_found	404	There is no such object in this company.
unknown_endpoint	404	There is no such endpoint. Check the method and the path against the reference.
session_expired	409	The checkout session's hold has ended. Create a new session.
session_complete	409	The session's order is already placed. To undo it, cancel the order.
already_paid	409	The order is already paid.
order_cancelled	409	The order was cancelled.
event_cancelled	409	A cancelled event cannot be put on sale.
code_exists	409	The discount code already exists on one of the events.
not_scannable	409	The event has no door list yet. Set one up in EOS.
not_on_sale	422	The event's tickets are not on sale, or the company is not taking orders.
unavailable	422	A ticket type is sold out or off sale, or the quantity is outside its limits.
discount_invalid	422	The discount code does not work on this event.
payments_off	422	The company cannot take that kind of payment right now.
date_of_birth_required	422	A ticket type has an age limit, so that guest's date_of_birth is needed.
age_restricted	422	A guest is outside the ticket type's age limit on the day of the event.
not_refundable	422	There is no card payment to refund, or the amount is more than was paid.
unknown_event	422	One of event_ids is not an upcoming event with tickets.
no_events	422	There are no upcoming events with tickets to put the code on.
not_created	422	The discount code could not be created on any of the events.
plan_limit	422	The company's plan has no room for this: another webhook endpoint (Starter has none), or another device scanning at the same time (a door app counts as one for each key). message says the limit: move up a plan, or ask AVEO.
too_many_endpoints	422	A company can have up to 100 API webhook endpoints.
rejected	422	The ticket engine refused the request. message says why.
idempotency_key_reused	409	The Idempotency-Key was used before with a different request.
request_in_progress	409	A request with the same Idempotency-Key is still running. Retry shortly.
rate_limited	429	Too many requests for this key in the last minute. Wait for the seconds in Retry-After.
too_many_failures	429	Too many requests from this address failed the key check in the last minute. Wait a minute and check the key.
too_many_holds	429	This visitor already holds as many checkout sessions as a publishable key allows. Complete or cancel one first.
internal	500	Something went wrong on our side. It is logged, and retrying with the same Idempotency-Key is safe.
engine_unavailable	502	The ticket engine did not answer. Retry in a moment.

02 THE API KEY.

Who you are, what your key may do, and the company it belongs to.

GET /api/v1/

Check your key

Who the key belongs to, its permissions and when it expires. A good first call.

Permission: any key · publishable keys too

ANSWER, 200


```
{
  "object": "api_key",
  "livemode": true,
  "kind": "secret",
  "name": "Website",
  "scopes": [
    "events:read",
    "checkout:write"
  ],
  "expires_at": null,
  "rate_per_minute": 120,
  "company": {
    "id": 12,
    "slug": "acme-events",
    "name": "Acme Events"
  }
}
```


GET /api/v1/company

Get the company

The company's name, shop address, currency and contact details.

Permission: company:read · publishable keys too

ANSWER, 200


```
{
  "object": "company",
  "id": 12,
  "slug": "acme-events",
  "name": "Acme Events",
  "shop_url": "https://eos.aveoevents.com/acme-events",
  "currency": "GBP",
  "timezone": "Europe/London",
  "contact": {
    "email": "hello@example.com",
    "phone": null,
    "website": "https://example.com"
  },
  "card_payments": true,
  "livemode": true
}
```

03 EVENTS.

Your events with their dates, venue and ticket types, and live availability.

GET /api/v1/events

List events

Published events, soonest first.

Permission: events:read · publishable keys too · paged

FIELD	TYPE	WHAT IT IS
when	string: upcoming, past, all	upcoming (default), past or all.
expand	string	ticket_types to include each event's ticket types.
limit	integer	How many to return, 1 to 100. Default 25.
cursor	string	The next_cursor from the previous page.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/events",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "event",
      "id": 41,
      "slug": "example-night",
      "title": "Example Night",
      "status": "on_sale",
      "published": true,
      "category": "music",
      "starts_at": "2026-12-05T19:00:00.000Z",
      "ends_at": "2026-12-05T23:30:00.000Z",
      "timezone": "Europe/London",
      "currency": "GBP",
      "price_from": 26,
      "summary": "An evening of live music.",
      "description": "Doors at 7pm.\n\nOver 18s only.",
      "venue": {
        "name": "Example Hall",
        "address": "1 Example Street, London"
      },
      "image_url": "https://eos.aveoevents.com/s/acme-events/uploads/hero.jpg",
  ...
```

GET /api/v1/events/{id}

Get an event

One event with its ticket types and live availability.

Permission: events:read · publishable keys too

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.

ANSWER, 200


```
{
  "object": "event",
  "id": 41,
  "slug": "example-night",
  "title": "Example Night",
  "status": "on_sale",
  "published": true,
  "category": "music",
  "starts_at": "2026-12-05T19:00:00.000Z",
  "ends_at": "2026-12-05T23:30:00.000Z",
  "timezone": "Europe/London",
  "currency": "GBP",
  "price_from": 26,
  "summary": "An evening of live music.",
  "description": "Doors at 7pm.\n\nOver 18s only.",
  "venue": {
    "name": "Example Hall",
    "address": "1 Example Street, London"
  },
  "image_url": "https://eos.aveoevents.com/s/acme-events/uploads/hero.jpg",
  "url": "https://eos.aveoevents.com/acme-events/events/example-night",
  "gates": [
    {
      "id": "main",
      "name": "Main entrance",
      "allow": []
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

GET /api/v1/events/{id}/ticket-types

List ticket types

The event's ticket types on sale or coming up, with what is left.

Permission: events:read · publishable keys too

FIELD	TYPE	WHAT IT IS

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/events/41/ticket-types",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "ticket_type",
      "id": 9001,
      "product_id": 812,
      "event_id": 41,
      "name": "General admission",
      "description": "Entry from 7pm.",
      "price": 25,
      "booking_fee": 1,
      "price_with_booking_fee": 26,
      "booking_fee_payer": "buyer",
      "paid_at_door": false,
      "door_price": null,
      "currency": "GBP",
      "on_sale": true,
      "sold_out": false,
      "min_per_order": 1,
      "max_per_order": 10,
      "remaining": 180,
      "note": null
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/events/{id}/publish`

Publish an event

Shows the event in the shop and puts its tickets on sale.

Permission: events:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.

ANSWER, 200


```
{
  "object": "event",
  "id": 41,
  "slug": "example-night",
  "title": "Example Night",
  "status": "on_sale",
  "published": true,
  "category": "music",
  "starts_at": "2026-12-05T19:00:00.000Z",
  "ends_at": "2026-12-05T23:30:00.000Z",
  "timezone": "Europe/London",
  "currency": "GBP",
  "price_from": 26,
  "summary": "An evening of live music.",
  "description": "Doors at 7pm.\n\nOver 18s only.",
  "venue": {
    "name": "Example Hall",
    "address": "1 Example Street, London"
  },
  "image_url": "https://eos.aveoevents.com/s/acme-events/uploads/hero.jpg",
  "url": "https://eos.aveoevents.com/acme-events/events/example-night",
  "gates": [
    {
      "id": "main",
      "name": "Main entrance",
      "allow": []
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.
409	event_cancelled	A cancelled event cannot be put on sale.

POST `/api/v1/events/{id}/unpublish`

Unpublish an event

Takes the event out of the shop. Orders already placed stay valid.

Permission: events:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.

ANSWER, 200


```
{
  "object": "event",
  "id": 41,
  "slug": "example-night",
  "title": "Example Night",
  "status": "on_sale",
  "published": false,
  "category": "music",
  "starts_at": "2026-12-05T19:00:00.000Z",
  "ends_at": "2026-12-05T23:30:00.000Z",
  "timezone": "Europe/London",
  "currency": "GBP",
  "price_from": 26,
  "summary": "An evening of live music.",
  "description": "Doors at 7pm.\n\nOver 18s only.",
  "venue": {
    "name": "Example Hall",
    "address": "1 Example Street, London"
  },
  "image_url": "https://eos.aveoevents.com/s/acme-events/uploads/hero.jpg",
  "url": "https://eos.aveoevents.com/acme-events/events/example-night",
  "gates": [
    {
      "id": "main",
      "name": "Main entrance",
      "allow": []
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

04 CHECKOUT.

Sell tickets from your own site: hold them, take the buyer's details, then payment.

POST `/api/v1/checkout/sessions`

Create a checkout session

Holds the tickets, usually for 15 minutes (until expires_at), and gives back a session. Then either send the buyer to its url (hosted checkout on the company's EOS shop) or call complete with their details (your own checkout).

Permission: checkout:write · publishable keys too · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
event_id *	integer	The event.
items *	array	What to hold: ticket_type_id and quantity for each line.
items.ticket_type_id *	integer	A ticket type of the event.
items.quantity *	integer	How many, within the ticket type's limits.
discount_code	string	A code to apply.
metadata	object	Up to 20 string keys and values of your own.

REQUEST BODY

```
{
  "event_id": 41,
  "items": [
    {
      "ticket_type_id": 9001,
      "quantity": 2
    }
  ],
  "metadata": {
    "basket": "8812"
  }
}
```

ANSWER, 201


```
{
  "object": "checkout_session",
  "id": "cs_9f2c4e1a7b3d5f8e0a6c2b4d",
  "status": "open",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "url": "https://eos.aveoevents.com/acme-events/checkout-link/eyJhbGciOi...",
  "expires_at": "2026-11-20T10:30:00.000Z",
  "currency": "GBP",
  "amount_items": 50,
  "booking_fee": 1,
  "booking_fee_payer": "buyer",
  "amount_total": 51,
  "discount_code": null,
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
    }
  ],
  "payment": null,
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.
422	not_on_sale	The event's tickets are not on sale (announced or cancelled), or the company is not taking orders.
422	unavailable	A ticket type is sold out, off sale, or the quantity is outside its limits.
422	discount_invalid	The discount code does not work on this event.

GET `/api/v1/checkout/sessions/{id}`

Get a checkout session

The session as it stands: still open, complete (the order is placed) or expired.

Permission: checkout:write · publishable keys too

FIELD	TYPE	WHAT IT IS
id *	string	The session id.

ANSWER, 200


```
{
  "object": "checkout_session",
  "id": "cs_9f2c4e1a7b3d5f8e0a6c2b4d",
  "status": "open",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "url": "https://eos.aveoevents.com/acme-events/checkout-link/eyJhbGciOi...",
  "expires_at": "2026-11-20T10:30:00.000Z",
  "currency": "GBP",
  "amount_items": 50,
  "booking_fee": 1,
  "booking_fee_payer": "buyer",
  "amount_total": 51,
  "discount_code": null,
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
    }
  ],
  "payment": null,
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/checkout/sessions/{id}/complete`

Complete a checkout session

Places the order with the buyer's details. Free orders are done and tickets are emailed. For card, the response's payment block has what Stripe's Payment Element needs (the amount is the order plus AVEO's booking fee); the order is paid and the tickets emailed as soon as Stripe confirms. Tickets paid at the door are free online, so an order of only those is done at once.

Permission: checkout:write · publishable keys too · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	string	The session id.
buyer *	object	Who is buying.
buyer.first_name *	string	First name.
buyer.last_name *	string	Last name.

FIELD	TYPE	WHAT IT IS
buyer.email *	string	Email.
attendees	array	One per ticket, in item order, when guests are named. Defaults to the buyer.
attendees.first_name *	string	First name.
attendees.last_name *	string	Last name.
attendees.email	string	Email.
attendees.date_of_birth	date	YYYY-MM-DD. Required for a ticket type with an age limit; checked against the event's date and shown at the door.
payment_method	string: card	card (the default and only method; pay_later is no longer offered). Ignored for free orders.
marketing	boolean	The buyer agreed to hear about future events.

REQUEST BODY

```
{
  "buyer": {
    "first_name": "Sam",
    "last_name": "Taylor",
    "email": "sam@example.com"
  },
  "payment_method": "card"
}
```

ANSWER, 200


```
{
  "object": "checkout_session",
  "id": "cs_9f2c4e1a7b3d5f8e0a6c2b4d",
  "status": "complete",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "url": "https://eos.aveoevents.com/acme-events/checkout-link/eyJhbGciOi...",
  "expires_at": "2026-11-20T10:30:00.000Z",
  "currency": "GBP",
  "amount_items": 50,
  "booking_fee": 1,
  "booking_fee_payer": "buyer",
  "amount_total": 51,
  "discount_code": null,
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
    }
  ],
  "payment": {
  ...
```


STATUS	CODE	WHEN
409	session_expired	The hold ended. Create a new session.
409	session_complete	This session's order is already placed.
422	payments_off	The company cannot take that kind of payment right now.
422	date_of_birth_required	A ticket type has an age limit: send that guest's date_of_birth.
422	age_restricted	A guest is outside the ticket type's age limit on the day of the event.

POST `/api/v1/checkout/sessions/{id}/cancel`

Cancel a checkout session

Lets the held tickets go straight back on sale.

Permission: checkout:write · publishable keys too

FIELD	TYPE	WHAT IT IS
id *	string	The session id.

ANSWER, 200


```
{
  "object": "checkout_session",
  "id": "cs_9f2c4e1a7b3d5f8e0a6c2b4d",
  "status": "cancelled",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "url": "https://eos.aveoevents.com/acme-events/checkout-link/eyJhbGciOi...",
  "expires_at": "2026-11-20T10:30:00.000Z",
  "currency": "GBP",
  "amount_items": 50,
  "booking_fee": 1,
  "booking_fee_payer": "buyer",
  "amount_total": 51,
  "discount_code": null,
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
    }
  ],
  "payment": null,
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.
409	session_complete	The order was already placed: cancel the order instead.

05 ORDERS.

Every order, paid or not: read them, cancel, refund, mark paid and resend tickets.

GET /api/v1/orders

List orders

Orders across every event, newest first.

Permission: orders:read · paged

FIELD	TYPE	WHAT IT IS
event_id	integer	Only this event's orders.
q	string	Search by order reference, name or email.
limit	integer	How many to return, 1 to 100. Default 25.
cursor	string	The next_cursor from the previous page.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/orders",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "order",
      "id": "O-7KQ2M4X",
      "number": 5501,
      "event_id": 41,
      "status": "completed",
      "payment_status": "paid",
      "currency": "GBP",
      "total": 50,
      "fees": 0,
      "tax": 0,
      "discount_code": null,
      "buyer": {
        "first_name": "Sam",
        "last_name": "Taylor",
        "email": "sam@example.com"
      },
      "items": [
        {
          "ticket_type_id": 9001,
  ...
```

GET /api/v1/orders/{id}

Get an order

One order with its items and tickets.

Permission: orders:read

FIELD	TYPE	WHAT IT IS
id *	string	The order reference (O-XXXXXXX).

ANSWER, 200


```
{
  "object": "order",
  "id": "O-7KQ2M4X",
  "number": 5501,
  "event_id": 41,
  "status": "completed",
  "payment_status": "paid",
  "currency": "GBP",
  "total": 50,
  "fees": 0,
  "tax": 0,
  "discount_code": null,
  "buyer": {
    "first_name": "Sam",
    "last_name": "Taylor",
    "email": "sam@example.com"
  },
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST /api/v1/orders/{id}/cancel

Cancel an order

Cancels the order: its tickets stop working and go back on sale. It does not refund a card payment: call refund for that, which also cancels a fully refunded order.

Permission: orders:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS

FIELD	TYPE	WHAT IT IS
id *	string	The order reference (O-XXXXXXX).

ANSWER, 200

```
{
  "object": "order",
  "id": "O-7KQ2M4X",
  "number": 5501,
  "event_id": 41,
  "status": "cancelled",
  "payment_status": "paid",
  "currency": "GBP",
  "total": 50,
  "fees": 0,
  "tax": 0,
  "discount_code": null,
  "buyer": {
    "first_name": "Sam",
    "last_name": "Taylor",
    "email": "sam@example.com"
  },
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
  ...
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/orders/{id}/refund`

Refund an order

Refunds the card payment through the company's Stripe account, in full or in part. A full refund cancels the tickets.

Permission: orders:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	string	The order reference (O-XXXXXXX).
amount	number	How much to refund. Leave out for everything.

REQUEST BODY

```
{
  "amount": 25
}
```

ANSWER, 200

```
{
  "object": "order",
  "id": "O-7KQ2M4X",
  "number": 5501,
  "event_id": 41,
  "status": "completed",
  "payment_status": "partially_refunded",
  "currency": "GBP",
  "total": 50,
  "fees": 0,
  "tax": 0,
  "discount_code": null,
  "buyer": {
    "first_name": "Sam",
    "last_name": "Taylor",
    "email": "sam@example.com"
  },
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
  ...
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.
422	not_refundable	There is no card payment to refund, or the amount is more than was paid.

POST `/api/v1/orders/{id}/mark-paid`

Mark an order paid

For an order awaiting payment when the money has arrived another way. The tickets are emailed.

Permission: orders:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	string	The order reference (O-XXXXXXX).

ANSWER, 200


```
{
  "object": "order",
  "id": "O-7KQ2M4X",
  "number": 5501,
  "event_id": 41,
  "status": "completed",
  "payment_status": "paid",
  "currency": "GBP",
  "total": 50,
  "fees": 0,
  "tax": 0,
  "discount_code": null,
  "buyer": {
    "first_name": "Sam",
    "last_name": "Taylor",
    "email": "sam@example.com"
  },
  "items": [
    {
      "ticket_type_id": 9001,
      "product_id": 812,
      "name": "General admission",
      "quantity": 2,
      "unit_price": 25,
      "total": 50,
      "price_before_discount": null
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.
409	already_paid	The order is already paid.
409	order_cancelled	The order was cancelled.

POST `/api/v1/orders/{id}/resend`

Resend tickets

Emails the order's tickets again to the buyer and guests.

Permission: orders:write

FIELD	TYPE	WHAT IT IS
id *	string	The order reference (O-XXXXXXX).

ANSWER, 200

```
{
  "object": "resend",
  "order_id": "O-7KQ2M4X",
  "emails": 2
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

06 GUESTS.

Ticket holders for an event: names, emails, ticket types and whether they are inside.

`GET` `/api/v1/events/{id}/attendees`

List guests

Everyone holding a ticket to the event.

Permission: attendees:read · paged

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
q	string	Search by name, email or ticket code.
limit	integer	How many to return, 1 to 100. Default 25.
cursor	string	The next_cursor from the previous page.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/events/41/attendees",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "attendee",
      "id": "A-7Q2KX9",
      "event_id": 41,
      "order_id": "O-7KQ2M4X",
      "order_number": 5501,
      "ticket_type_id": 9001,
      "ticket_type": "General admission",
      "first_name": "Sam",
      "last_name": "Taylor",
      "email": "sam@example.com",
      "status": "active",
      "checked_in": false,
      "checked_in_at": null
    }
  ]
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

GET /api/v1/events/{id}/attendees/{attendee_id}

Get a guest

One ticket and its holder.

Permission: attendees:read

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
attendee_id *	string	The ticket code.

ANSWER, 200

```
{
  "object": "attendee",
  "id": "A-7Q2KX9",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "order_number": 5501,
  "ticket_type_id": 9001,
  "ticket_type": "General admission",
  "first_name": "Sam",
  "last_name": "Taylor",
  "email": "sam@example.com",
  "status": "active",
  "checked_in": false,
  "checked_in_at": null
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

PATCH /api/v1/events/{id}/attendees/{attendee_id}

Update a guest

Changes the name or email on a ticket.

Permission: attendees:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
attendee_id *	string	The ticket code.
first_name	string	First name.
last_name	string	Last name.

FIELD	TYPE	WHAT IT IS
email	string	Email.

REQUEST BODY

```
{
  "first_name": "Jordan",
  "last_name": "Lee",
  "email": "jordan@example.com"
}
```

ANSWER, 200

```
{
  "object": "attendee",
  "id": "A-7Q2KX9",
  "event_id": 41,
  "order_id": "O-7KQ2M4X",
  "order_number": 5501,
  "ticket_type_id": 9001,
  "ticket_type": "General admission",
  "first_name": "Jordan",
  "last_name": "Lee",
  "email": "jordan@example.com",
  "status": "active",
  "checked_in": false,
  "checked_in_at": null
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/events/{id}/attendees/{attendee_id}/resend`

Resend a ticket

Emails this one ticket to its holder again.

Permission: attendees:write

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
attendee_id *	string	The ticket code.

ANSWER, 200

```
{
  "object": "resend",
  "attendee_id": "A-7Q2KX9",
  "emails": 1
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

07 CHECK-INS.

Scan tickets in and out from your own door app, with gates and re-entry.

POST `/api/v1/events/{id}/check-ins`

Scan a ticket

The same check-in as the EOS scanner: each ticket once, gates, exits and re-entry, age and staff passes. Send what the QR says, the ticket code, or a link to the ticket.

Permission: checkins:write

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
code *	string	What was scanned or typed.
gate	string	The gate's id, from the event's gates. Defaults to the first gate.
mode	string: entry, exit	entry (default) or exit.
paid_at_door	boolean	For a ticket paid at the door: the door has taken its price. Send it with the same code after a payment_due answer; the payment is recorded and the guest let in.

REQUEST BODY

```
{
  "code": "A-7Q2KX9",
  "gate": "main"
}
```

ANSWER, 200


```
{
  "object": "check_in",
  "event_id": 41,
  "attendee_id": "A-7Q2KX9",
  "result": "admitted",
  "ok": true,
  "message": "Admit",
  "name": "Sam Taylor",
  "ticket_type": "General admission",
  "gate": "Main entrance",
  "mode": "entry",
  "age": 27,
  "note": null,
  "addons": [
    "Cloakroom"
  ],
  "first_scan": null,
  "door_payment": null,
  "created_at": "2026-12-05T19:02:11.000Z"
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.
409	not_scannable	The event has no door list yet. Set one up in EOS.
422	plan_limit	The plan's devices are all scanning: each API key that checks guests in counts as one device, as a phone does.

GET `/api/v1/events/{id}/check-ins`

List check-ins

Every scan at the event's doors, newest first.

Permission: checkins:read · paged

FIELD	TYPE	WHAT IT IS
id *	integer	The event's id.
limit	integer	How many to return, 1 to 100. Default 25.
cursor	string	The next_cursor from the previous page.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/events/41/check-ins",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "check_in",
      "event_id": 41,
      "attendee_id": "A-7Q2KX9",
      "result": "admitted",
      "ok": true,
      "message": "Admit",
      "name": "Sam Taylor",
      "ticket_type": "General admission",
      "gate": "Main entrance",
      "mode": "entry",
      "age": 27,
      "note": null,
      "addons": [
        "Cloakroom"
      ],
      "first_scan": null,
      "door_payment": null,
      "created_at": "2026-12-05T19:02:11.000Z"
    }
  ...
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

08 DISCOUNT CODES.

Codes buyers type at checkout, on one event or many.

GET /api/v1/discount-codes

List discount codes

Every code on upcoming events, one row per code.

Permission: discounts:read

FIELD	TYPE	WHAT IT IS
event_id	integer	Only codes on this event.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/discount-codes",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "discount_code",
      "code": "EARLYBIRD",
      "discount_type": "percentage",
      "amount": 10,
      "max_uses": 100,
      "uses": 3,
      "tickets": 5,
      "expires_at": null,
      "events": [
        {
          "event_id": 41,
          "title": "Example Night",
          "uses": 3
        }
      ]
    }
  ]
}
```


POST /api/v1/discount-codes

Create a discount code

Makes the code on the events you name, or on every upcoming event.

Permission: discounts:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS

FIELD	TYPE	WHAT IT IS
code *	string	3 to 30 letters, numbers, - or _. Not case-sensitive.
discount_type *	string: percentage, fixed, none	percentage, fixed or none.
amount	number	Percent (1 to 100) or amount off each ticket. Not for none.
max_uses	integer	Most orders per event.
expires_at	timestamp	When it stops working.
event_ids	array	The events. Leave out for every upcoming event.

REQUEST BODY

```
{
  "code": "EARLYBIRD",
  "discount_type": "percentage",
  "amount": 10,
  "max_uses": 100,
  "event_ids": [
    41
  ]
}
```

ANSWER, 201

```
{
  "object": "discount_code",
  "code": "EARLYBIRD",
  "discount_type": "percentage",
  "amount": 10,
  "max_uses": 100,
  "uses": 0,
  "tickets": 0,
  "expires_at": null,
  "events": [
    {
      "event_id": 41,
      "title": "Example Night",
      "uses": 0
    }
  ]
}
```

STATUS	CODE	WHEN
409	code_exists	The code already exists on one of those events.
422	unknown_event	One of event_ids is not an upcoming event with tickets.
422	no_events	There are no upcoming events with tickets to put the code on.

STATUS	CODE	WHEN
422	not_created	The ticket engine refused the code on every event named.

DELETE `/api/v1/discount-codes/{code}`

Delete a discount code

Removes the code from every event it is on.

Permission: discounts:write

FIELD	TYPE	WHAT IT IS
code *	string	The code.

ANSWER, 200

```
{
  "object": "discount_code",
  "code": "EARLYBIRD",
  "deleted": true
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

09 ENQUIRIES.

Messages from your own contact form, straight into Customer cases.

POST `/api/v1/enquiries`

Send an enquiry

Opens a customer case with the message, emails the customer its reference and alerts the team, as the shop's contact form does.

Permission: enquiries:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
name *	string	Their name.
email *	string	Their email.
phone	string	Their phone.
subject *	string	A short subject.
message *	string	The message.
topic	string: tickets, refunds, event-day, accessibility, safety, privacy, complaints, other	What it is about; sets the case's priority. Default other.
order_id	string	An order reference it is about.

REQUEST BODY

```
{
  "name": "Sam Taylor",
  "email": "sam@example.com",
  "subject": "Wheelchair access",
  "message": "Is there step-free access to the hall?",
  "topic": "accessibility"
}
```

ANSWER, 201


```
{
  "object": "enquiry",
  "id": "CS-4KQ8ZP",
  "status": "open",
  "subject": "Wheelchair access",
  "name": "Sam Taylor",
  "email": "sam@example.com",
  "created_at": "2026-11-20T10:15:00.000Z"
}
```

10 REPORTS.

Sales by day for one event or all of them.

`GET` `/api/v1/reports/sales`

Sales by day

Orders, tickets, sales and refunds per day, for one event or every event in the period.

Permission: reports:read

FIELD	TYPE	WHAT IT IS
event_id	integer	Only this event.
currency	string	Without event_id: only events in this currency. Default the company's currency.
from	date	First day (YYYY-MM-DD). Default 30 days ago.
to	date	Last day. Default today.

ANSWER, 200


```
{
  "object": "sales_report",
  "event_id": 41,
  "from": "2026-11-01",
  "to": "2026-11-30",
  "currencies": [
    "GBP"
  ],
  "totals": {
    "orders": 120,
    "tickets": 214,
    "sales": 5350,
    "refunds": 50
  },
  "days": [
    {
      "date": "2026-11-20",
      "orders": 9,
      "tickets": 16,
      "sales": 400,
      "refunds": 0
    }
  ]
}
```

11 WEBHOOK ENDPOINTS.

Addresses EOS calls when something happens, signed so you can trust them.

GET /api/v1/webhook-endpoints

List endpoints

Your webhook endpoints.

Permission: webhooks:read

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/webhook-endpoints",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "webhook_endpoint",
      "id": 7,
      "url": "https://example.com/hooks/eos",
      "description": "Orders into our CRM",
      "events": [
        "order.paid",
        "order.refunded"
      ],
      "active": true,
      "created_at": "2026-11-20T09:00:00.000Z",
      "last_status": 200,
      "last_attempt_at": "2026-11-20T10:15:02.000Z"
    }
  ]
}
```


POST /api/v1/webhook-endpoints

Add an endpoint

Adds an https address. The signing secret is in the response once: keep it.

Permission: webhooks:write · takes an Idempotency-Key

FIELD	TYPE	WHAT IT IS
url *	string	Your https address. Private network addresses are refused.
description	string	What it is for.
events	array	Event types (20 kinds); leave out or [] for all.

REQUEST BODY

```
{
  "url": "https://example.com/hooks/eos",
  "description": "Orders into our CRM",
  "events": [
    "order.paid",
    "order.refunded"
  ]
}
```

ANSWER, 201

```
{
  "object": "webhook_endpoint",
  "id": 7,
  "url": "https://example.com/hooks/eos",
  "description": "Orders into our CRM",
  "events": [
    "order.paid",
    "order.refunded"
  ],
  "active": true,
  "created_at": "2026-11-20T09:00:00.000Z",
  "last_status": null,
  "last_attempt_at": null,
  "secret": "whsec_MfKQ9r8GKYqrTwJg..."
}
```

STATUS	CODE	WHEN
422	plan_limit	The company's plan has no room for another endpoint (Starter has none).
422	too_many_endpoints	A company can have up to 100 API webhook endpoints.

GET `/api/v1/webhook-endpoints/{id}`

Get an endpoint

One endpoint.

Permission: webhooks:read

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.

ANSWER, 200

```
{
  "object": "webhook_endpoint",
  "id": 7,
  "url": "https://example.com/hooks/eos",
  "description": "Orders into our CRM",
  "events": [
    "order.paid",
    "order.refunded"
  ],
  "active": true,
  "created_at": "2026-11-20T09:00:00.000Z",
  "last_status": 200,
  "last_attempt_at": "2026-11-20T10:15:02.000Z"
}
```

STATUS	CODE	WHEN
404	not_found	No such object in this company.

PATCH `/api/v1/webhook-endpoints/{id}`

Change an endpoint

Changes its address, description, events, or turns it off and on.

Permission: webhooks:write

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.
url	string	New address.
description	string	New description.
events	array	New event types.
active	boolean	On or off.

REQUEST BODY

```
{
  "active": false
}
```

ANSWER, 200


```
{
  "object": "webhook_endpoint",
  "id": 7,
  "url": "https://example.com/hooks/eos",
  "description": "Orders into our CRM",
  "events": [
    "order.paid",
    "order.refunded"
  ],
  "active": false,
  "created_at": "2026-11-20T09:00:00.000Z",
  "last_status": 200,
  "last_attempt_at": "2026-11-20T10:15:02.000Z"
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

DELETE `/api/v1/webhook-endpoints/{id}`

Remove an endpoint

Removes it; nothing more is sent there.

Permission: webhooks:write

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.

ANSWER, 200


```
{
  "object": "webhook_endpoint",
  "id": 7,
  "deleted": true
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/webhook-endpoints/{id}/roll-secret`

Roll the signing secret

Makes a new signing secret at once; the old one stops working.

Permission: webhooks:write

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.

ANSWER, 200


```
{
  "object": "webhook_endpoint",
  "id": 7,
  "url": "https://example.com/hooks/eos",
  "description": "Orders into our CRM",
  "events": [
    "order.paid",
    "order.refunded"
  ],
  "active": true,
  "created_at": "2026-11-20T09:00:00.000Z",
  "last_status": 200,
  "last_attempt_at": "2026-11-20T10:15:02.000Z",
  "secret": "whsec_Q2Vz9ZpL0..."
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/webhook-endpoints/{id}/test`

Send a test event

Sends a ping event to the endpoint now and returns the delivery.

Permission: webhooks:write

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.

ANSWER, 200


```
{
  "object": "webhook_delivery",
  "id": "msg_2c8d4e6f8a0b4c6d8e0f2a4b6c8d0e2f",
  "type": "ping",
  "created_at": "2026-11-20T10:15:01.000Z",
  "attempts": 1,
  "delivered": true,
  "delivered_at": "2026-11-20T10:15:02.000Z",
  "next_attempt_at": null,
  "last_error": null
}
```


STATUS	CODE	WHEN

STATUS	CODE	WHEN
404	not_found	No such object in this company.

GET `/api/v1/webhook-endpoints/{id}/deliveries`

List deliveries

What was sent to the endpoint, newest first, with retries and errors.

Permission: webhooks:read · paged

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.
limit	integer	How many to return, 1 to 100. Default 25.
cursor	string	The next_cursor from the previous page.

ANSWER, 200


```
{
  "object": "list",
  "url": "/api/v1/webhook-endpoints/7/deliveries",
  "has_more": false,
  "next_cursor": null,
  "data": [
    {
      "object": "webhook_delivery",
      "id": "msg_2c8d4e6f8a0b4c6d8e0f2a4b6c8d0e2f",
      "type": "order.paid",
      "created_at": "2026-11-20T10:15:01.000Z",
      "attempts": 1,
      "delivered": true,
      "delivered_at": "2026-11-20T10:15:02.000Z",
      "next_attempt_at": null,
      "last_error": null
    }
  ]
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

POST `/api/v1/webhook-endpoints/{id}/deliveries/{delivery_id}/retry`

Retry a delivery

Sends a delivery again now, even one that already succeeded.

Permission: webhooks:write

FIELD	TYPE	WHAT IT IS

FIELD	TYPE	WHAT IT IS
id *	integer	The endpoint's id.
delivery_id *	string	The delivery's id.

ANSWER, 200


```
{
  "object": "webhook_delivery",
  "id": "msg_2c8d4e6f8a0b4c6d8e0f2a4b6c8d0e2f",
  "type": "order.paid",
  "created_at": "2026-11-20T10:15:01.000Z",
  "attempts": 1,
  "delivered": true,
  "delivered_at": "2026-11-20T10:15:02.000Z",
  "next_attempt_at": null,
  "last_error": null
}
```


STATUS	CODE	WHEN
404	not_found	No such object in this company.

12 OBJECTS.

Company.

The company the key belongs to.

FIELD	TYPE	WHAT IT IS
id	integer	The company's number on AVEO EOS.
slug	string	Its shop address on EOS.
name	string	Its name.
shop_url	string	The company's ticket shop on EOS.
currency	string	Its usual currency (ISO 4217).
timezone	string	Its time zone.
contact	object	Email, phone and website, as set on the company.
contact.email	string	Contact email.
contact.phone	string, or null	Contact phone.
contact.website	string, or null	Website.
card_payments	boolean	Whether the company can take card payments now (its Stripe account is connected and active).
livemode	boolean	False in the company's sandbox, where everything is test data and payments use Stripe's test mode.

Event.

An event with its dates, venue and, when asked for, its ticket types.

FIELD	TYPE	WHAT IT IS
id	integer	The event's id.
slug	string	Its address in the shop.
title	string	Its name.
status	string	on_sale, announced, sold_out, postponed or cancelled.

FIELD	TYPE	WHAT IT IS
published	boolean	Whether it shows in the shop.
category	string	Its category.
starts_at	timestamp	Start (UTC).
ends_at	timestamp	End (UTC).
timezone	string	The event's local time zone.
currency	string	Ticket currency (ISO 4217).
price_from	number, or null	The lowest price of one ticket as a buyer pays it: AVEO's booking fee included, or the door price for a ticket paid at the door.
summary	string	One line about it.
description	string	The full description, paragraphs separated by blank lines.
venue	object	Where it is.
venue.name	string	Venue name.
venue.address	string	Venue address.
image_url	string, or null	The main picture.
url	string	The event's page in the company's shop.
gates	array	The event's doors. Send a gate's id as gate when scanning.
gates.id	string	The gate's id.
gates.name	string	Its name, as stewards see it.
gates.allow	array	The kinds of pass it lets in (such as vip or staff); empty means everyone.
reentry	boolean	Whether guests may leave and come back in.
ticket_types	array	Ticket types, on GET /events/{id} or with expand=ticket_types.

Ticket type.

One price of a ticket for an event, with live availability.

FIELD	TYPE	WHAT IT IS
id	integer	The ticket type's id. Use it in checkout items.

FIELD	TYPE	WHAT IT IS
product_id	integer	The ticket it is a price of.
event_id	integer	The event.
name	string	Its name.
description	string	What it includes.
price	number	Price per ticket online, including taxes; 0 for a ticket paid at the door.
booking_fee	number	AVEO's booking fee on an order of this ticket type alone. Charged once per order on card orders, at the band of the order's most expensive ticket: £10 to £19.99 £0.75, £20 to £49.99 £1.00, £50 to £99.99 £1.50, £100 to £199.99 £2.00, £200 and over £2.50; none under £10, on free tickets or at the door.
price_with_booking_fee	number	price plus booking_fee: what one ticket costs a buyer. Show this from the first price on your pages (the headline price must include mandatory fees).
booking_fee_payer	string: buyer, company	Who pays AVEO's booking fee, as agreed with AVEO: buyer (on top of the price) or company (absorbed in the company's prices; booking_fee is then 0 and price_with_booking_fee equals price).
paid_at_door	boolean	Free to book online; its door_price is taken at the door before the guest goes in.
door_price	number, or null	What is paid at the door.
currency	string	Currency.
on_sale	boolean	Inside its sale window.
sold_out	boolean	None left.
min_per_order	integer	Fewest per order.
max_per_order	integer	Most per order.
remaining	integer, or null	Tickets left, when limited.
note	string, or null	Why it is off sale, when it is.

Checkout session.

Tickets held for a buyer, from the moment you choose them until the order is placed or the hold ends.

FIELD	TYPE	WHAT IT IS

FIELD	TYPE	WHAT IT IS
id	string	The session's id.
status	string	open, complete, expired or cancelled.
event_id	integer	The event.
order_id	string	The order that will be (or was) placed.
url	string	Send the buyer here to finish on the company's own EOS checkout (hosted checkout).
expires_at	timestamp	When the hold ends and the tickets go back on sale.
currency	string	Currency.
amount_items	number	The tickets and extras after any discount.
booking_fee	number	The booking fee the buyer pays on top: once per order, at the band of its most expensive ticket; 0 when nothing is paid online or when the company absorbs the fee.
booking_fee_payer	string: buyer, company	buyer or company: who pays AVEO's booking fee.
amount_total	number	What the buyer pays online: amount_items plus booking_fee.
discount_code	string, or null	The code applied.
items	array	What is held.
payment	object, or null	After complete with card: what Stripe's Payment Element needs.
payment.method	string	card or free (free orders and tickets paid at the door).
payment.amount	number	What the card is charged: the order and the booking fee.
payment.booking_fee	number	The booking fee in it.
payment.client_secret	string, or null	The PaymentIntent's client secret.
payment.publishable_key	string, or null	Stripe publishable key.
payment.stripe_account	string, or null	The company's connected Stripe account, for Stripe.js.
metadata	object	Up to 20 keys of your own, returned as sent.
created_at	timestamp	When it was made.

Order.

An order and its tickets.

FIELD	TYPE	WHAT IT IS
id	string	The order reference, as in the buyer's email (O-XXXXXXX).
number	integer	The order's number.
event_id	integer	The event.
status	string	reserved, awaiting_payment, completed, cancelled or abandoned.
payment_status	string	paid, unpaid, no_payment_required, refunded or partially_refunded.
currency	string	Currency.
total	number	The order's tickets and extras.
fees	number	Fees in the total.
tax	number	Tax in the total.
discount_code	string, or null	The code used.
buyer	object	Who bought it.
buyer.first_name	string	First name.
buyer.last_name	string	Last name.
buyer.email	string	Email.
items	array	Lines of the order.
attendees	array	One per ticket.
created_at	timestamp	When it was placed.
channel	string: shop, api, issued, promoter	Where it came from: shop (the EOS shop), api (your own site or app), issued (tickets issued in EOS) or promoter.
checkout_session_id	string, or null	The API checkout session it came from.
metadata	object	What your system sent with the checkout session, returned as sent. Empty for orders from elsewhere.
booking_fee	number	AVEO's booking fee the buyer paid by card on top of total, once per order; 0 when no card payment was taken.

Guest.

One ticket and the person holding it.

FIELD	TYPE	WHAT IT IS
id	string	The ticket code (also in the QR).
event_id	integer	The event.
order_id	string, or null	Its order's reference, when known.
order_number	integer, or null	Its order's number.
ticket_type_id	integer	Its ticket type.
ticket_type	string	The ticket type's name.
first_name	string	First name.
last_name	string	Last name.
email	string	Email.
status	string	active, cancelled or awaiting_payment.
checked_in	boolean	Inside now.
checked_in_at	timestamp, or null	Last admitted.

Check-in.

A scan at the door and what happened.

FIELD	TYPE	WHAT IT IS
event_id	integer	The event.
attendee_id	string, or null	The ticket scanned, when it was one.
result	string	admitted, readmitted, exited, duplicate, wrong_gate, not_inside, invalid, staff or payment_due (a ticket paid at the door that has not been paid).
ok	boolean	Let them through.
message	string	What to show the steward.
name	string, or null	The guest's name.

FIELD	TYPE	WHAT IT IS
ticket_type	string, or null	Their ticket.
gate	string, or null	The gate's name.
mode	string	entry or exit.
age	integer, or null	The guest's age on the day of the event, when the ticket has a date of birth.
note	string, or null	Something the steward should check, such as ID when the ticket type has an age limit and no date of birth.
addons	array	Extras the guest bought, to hand over or check at the door.
first_scan	object, or null	For a duplicate: when and at which gate the ticket first got in. On scans read back from the list, always null.
first_scan.at	timestamp	When.
first_scan.gate	string, or null	The gate's name.
door_payment	object, or null	A ticket paid at the door: its price, and whether the door has taken it. On scans read back from the list, always null.
door_payment.amount	number	The door price.
door_payment.currency	string	Currency.
door_payment.paid	boolean	Taken: recorded when a scan is sent with paid_at_door.
created_at	timestamp	When.

Discount code.

A code buyers type at checkout. One code can work on many events.

FIELD	TYPE	WHAT IT IS
code	string	The code, in capitals.
discount_type	string	percentage, fixed (amount off each ticket) or none (unlocks hidden tickets).
amount	number	Percent or amount off.
max_uses	integer, or null	Most orders per event.
uses	integer	Orders that used it.

FIELD	TYPE	WHAT IT IS
tickets	integer	Tickets bought with it.
expires_at	timestamp, or null	When it stops working.
events	array	The events it works on, with each one's uses.

Webhook endpoint.

An address EOS sends signed events to.

FIELD	TYPE	WHAT IT IS
id	integer	Its id.
url	string	Your https address.
description	string	What it is for.
events	array	Event types it receives; empty means all.
active	boolean	Receiving events.
created_at	timestamp	Added.
last_status	integer, or null	HTTP status of the last attempt.
last_attempt_at	timestamp, or null	Last attempt.
secret	string	The signing secret: only when the endpoint is made or its secret rolled.

Webhook delivery.

One event sent (or being sent) to one endpoint.

FIELD	TYPE	WHAT IT IS
id	string	The webhook-id header value.
type	string	The event type.
created_at	timestamp	When it happened.
attempts	integer	Tries so far.
delivered	boolean	Your endpoint answered 2xx.

FIELD	TYPE	WHAT IT IS
delivered_at	timestamp, or null	When.
next_attempt_at	timestamp, or null	The next retry.
last_error	string, or null	Why the last try failed.

13 WEBHOOK EVENTS.

Each call carries the event's type and the same object the API returns for it, signed to the Standard Webhooks scheme: check webhook-signature (v1, HMAC-SHA256 over id.timestamp.body with your endpoint's secret) and that webhook-timestamp is under five minutes old. Answer within ten seconds; a failed call is tried again for about four hours.

THE BODY

```
{
  "id": "evt_5c1d8e2f9a3b7c4d6e0f1a2b",
  "type": "order.paid",
  "api_version": "2026-11-15",
  "created_at": "2026-11-20T10:15:01.000Z",
  "company_id": 12,
  "livemode": true,
  "data": { "object": { "object": "order", "id": "O-7KQ2M4X", ... } }
}
```

EVENT	OBJECT	WHEN
order.created	order	An order was placed (in the shop, through the API or issued in EOS): paid, free, or waiting for payment.
order.paid	order	An order's payment arrived: by card through Stripe, marked paid, or a free order.
order.payment_failed	order	A card payment for an order failed. The order still waits; the buyer may try again. Carries payment_error.
order.cancelled	order	An order was cancelled (by staff, through the API, or left unpaid) and its tickets went back on sale.
order.refunded	order	Money was refunded on an order through Stripe, in full or in part.
checkout.session.completed	checkout_session	A checkout session's order was placed (API checkout).
checkout.session.expired	checkout_session	A checkout session's hold ended before the order was placed; its tickets went back on sale.
checkout.session.cancelled	checkout_session	A checkout session was cancelled and its tickets went back on sale.
attendee.checked_in	check_in	A ticket was admitted at the door, by any scanner: the EOS scanner, the API, or a check-in by hand.
attendee.checked_out	check_in	A guest was scanned out.

EVENT	OBJECT	WHEN
attendee.updated	attendee	A ticket holder's name or email changed, in EOS or through the API.
event.created	event	An event was made in EOS (it is not on sale until published).
event.updated	event	An event's details or ticket types changed.
event.published	event	An event went on sale.
event.unpublished	event	An event was taken off sale.
event.cancelled	event	An event was cancelled.
discount_code.created	discount_code	A discount code was created, in EOS or through the API.
discount_code.deleted	discount_code	A discount code was removed from every event it was on.
enquiry.created	enquiry	A message came in: through the shop's contact form or the API. It is a customer case in EOS.
ping	ping	A test, sent when you test an endpoint.

AVEO

BUILD WITH AVEO.

The reference at aveoevents.com/eos/api is always the latest, with the developer guide and an OpenAPI document for your tools.

ASK FOR A TRIAL	aveoevents.com/eos
DOCUMENTS AND GUIDES	documents@aveoevents.com
SUPPORT	support@aveoevents.com
API	aveoevents.com/eos/api

AVEO EOS is made and run by AVEO. 22 November 2026.