

VBO API — Kurzüberblick

Die wichtigsten Informationen zur öffentlichen Schnittstelle des Ticketsystems (Vibe / VBO Eventmanager API v1)

Wichtig (Update 01.08.2026): Die API-Server wurden umgestellt. Bisher: www.vbotickets.com/... — Neu: eventmanager.vbotickets.com/.... Alle neuen Integrationen sollten die neue Domain verwenden.

1. Zweck der Schnittstelle

Die API dient dazu, Informationen aus dem Ticketsystem automatisiert abzurufen und mit externen Systemen zu verbinden. Typische Einsatzzwecke:

- SSO / Kundenkonten: Verbindung zwischen Kundenkonten im Ticketsystem und einer externen App.
- Events & Kalender: Eventdaten (Name, Tag, Ort, ...) für Website, Eventkalender oder individuelle Eventseiten.
- Bestellungen: Bestelldetails inkl. Tickets, Preise, Sitzplätze und Zahlungsinformationen.
- Tickets in eigener App: Ticketdaten (Event, Preis, Sitzplatz, Barcode/QR) im eigenen Design darstellen.
- Webhooks: Push-Benachrichtigungen bei Verkäufen, Stornos, Event-Änderungen und Ticket-Scans.

2. Architektur: Push/Pull

Die API ist als Push/Pull-Modell konzipiert: Ein Webhook liefert nur eine Ressourcen-ID (z. B. UniqueOrderID). Anschließend wird die vollständige Ressource per authentifiziertem REST-GET nachgeladen. Webhook-Signaturen sollten über den Header SecretHash mit dem API-Secret validiert werden.

3. Authentifizierung & API-Keys

API-Key und API-Secret werden unter Einstellungen → Erweiterungen → API Schlüssel erzeugt. Keys nach der Erzeugung sofort speichern — sie werden aus Sicherheitsgründen danach ausgeblendet und sind nur durch Neuerzeugung wieder sichtbar.

Erforderliche Request-Header

Header	Wert / Bedeutung
Authorization	Bearer <API-KEY>
SecretHash	HMAC-SHA256 über vollständige URL (+ Body bei POST/PATCH); Secret hex-dekodiert
Version	1.0 (Pflicht laut API-Doku)

Hash-Berechnung (Kurzfassung)

Input = vollständige, unkodierte API-URL (inkl. Query-String) + Request-Body (bei GET oft leer). Secret ist hex-kodiert und muss vor dem HMAC dekodiert werden. Ergebnis: Hex-String in Kleinbuchstaben. Beispiel-Base-URL (neu): <https://eventmanager.vbotickets.com/api2/...>

Hinweis: In der Help-Doku gibt es ein Golang-Beispiel, das das Secret als UTF-8-Bytes hasht. Die offizielle OpenAPI-Doku verlangt Hex-Decode des Secrets — für Produktionsintegrationen die OpenAPI-Variante priorisieren und ggf. mit dem Support abstimmen.

4. REST-Endpunkte (API v1)

Basis: <https://eventmanager.vbotickets.com> — Response-Format typischerweise { Status, Error, Data }.

Gruppe	Methode / Pfad	Beschreibung	Parameter
Account	GET /api2/account/token	SSO-Login-Token für Online-Sales-Plugin (gültig 2 Std.)	Query: email
Account	POST /api2/Account	Neues Kundenkonto anlegen	Body: AccountPatchModel
Account	PATCH /api2/Account/{id}	Kundenkonto aktualisieren	Path: id + Body
Event	GET /api2/event/allevnts	Eventliste für Zeitraum (Standard/Max. 30 Tage)	Query: startDate, endDate
Event	GET /api2/Event/{id}	Details zu einem Event	Path: Unique Event-ID
EventData	GET /api2/eventdate/eventdatesforevent	Eventtage eines Events (Max. 30 Tage)	Query: uniqueEventID, start/endDate
EventData	GET /api2/EventDate/{id}	Details zu einem Eventtag inkl. Venue	Path: Unique EventDate-ID
Order	GET /api2/Order/{id}	Bestellung inkl. Positionen, Zahlungen, Events	Path: UniqueOrderID; opt. sales/refundReceiptID
OrderDetail	GET /api2/orderdetail/ticketuser/{id}	Ticketnutzer (Name, Telefon, E-Mail, ...)	Path: Ticket-GUID
Tickets	GET /api2/tickets/byeventdate/{id}	Alle Tickets eines Eventtages	Path: EventDate-GUID

Zusätzlicher öffentlicher Eventlisten-Link (ohne HMAC)

Unter Einstellungen → Erweiterungen → VBO Api-Key gibt es einen einfachen JSON-Link für Veranstaltungen:

https://eventmanager.vbotickets.com/_modules/xml/get_events.json.asp?key=<API-KEY>&tix=1&adv=1

Bei freier Platzwahl: verfügbare Tickets. Bei Sitzplatz-Events: Platzhalterwert (nicht die echte Restmenge).

5. Webhooks

Konfiguration: Einstellungen → Erweiterungen → Webhooks. Trigger können auf alle oder ausgewählte Events eingeschränkt werden. Beliebig viele Webhooks/Trigger möglich.

Trigger	Payload-IDs	Bedeutung
order.sale / order.created	UniqueOrderID, SalesReceiptID	Neue Bestellung
order.refund	UniqueOrderID, RefundReceiptID	Stornierung / Erstattung
event.created / .modified / .deleted	UniqueEventID	Event angelegt, geändert, gelöscht
eventdate.created / .modified / .deleted	UniqueEventDateID	Eventtag angelegt, geändert, gelöscht
ticket.scanned	TicketID (GUID)	Ticket wurde gescannt
ticketuser.modified	TicketID	Ticketnutzer-Daten geändert (OpenAPI)

Ablauf Beispiel Verkauf: Webhook order.sale empfangen → HMAC-GET auf /api2/Order/{UniqueOrderID}?salesReceiptID=... → Daten aus Response verarbeiten (Account, OrderDetails, Barcodes, Venue, ...).

Ticket-Scan: Trigger ticket.scanned sendet Ticket-GUID → Nutzerinfos über GET /api2/orderdetail/ticketuser/{id} laden.

6. Wichtige Response-Felder (Auszug)

Event (GET /api2/Event/{id})

UniqueID, EventID, Name, Status, Date/Start/End, CategoryID/SubCategoryID, PriceRange, EventColor, EventIntro, EventTxt, EventVideo, PicBanner/PicSm/PicMd/PicLg, Sorting

EventData (GET /api2/EventDate/{id})

UniqueID, EventID, EventDateID, EventDateName, Date, DoorsOpen, EventDuration, IsActive/EventDateStatus, EventPicture, Venue (VenueID, Name, Phone, Email, Address)

Order (GET /api2/Order/{id})

UniqueOrderID, OrderDate, SalesMethod, ShippingType, Account (Name, Adresse, E-Mail), OrderPayments, OrderDetails (ItemName, Preis, Tax, TicketID, Barcode, OpenSeat/Seat, EventDate/Venue), Totals (Total, Tax, NetAmount, TicketCount)

Ticket / TicketUser

ApiTicketInfo: OrderID, Barcode, Section/Row/Seat, TicketName, TicketHolder, IsValid, TicketID —
APITicketUserInfo: UniqueOrderID, FirstName, LastName, Phone, Email, Company

7. SSO (Account-Token)

Mit GET /api2/account/token?email=... erhält man ein Anmelde-Token für das Online-Sales-Plugin (Gültigkeit: 2 Stunden). Zusammen mit Account-POST/PATCH lassen sich externe App-Konten und Ticketsystem-Konten koppeln bzw. synchronisieren.

8. Wichtige Regeln & Hinweise

- API kann unangekündigt um neue Objekte/Felder erweitert werden; bestehende Felder werden nicht unangekündigt geändert/gelöscht.
- Datumsbereiche bei Event-/EventDate-Listen: Standard und Maximum = 30 Tage.
- Fehlerstruktur: Status + Error (z. B. EntityNotFound) + Data.
- Webhook-Empfänger sollte SecretHash prüfen und idempotent arbeiten.
- API-Keys und Secrets wie Passwörter behandeln; nie in Frontend-Code committen.
- Zielgruppe laut Help: Admin (Ticketsystem und Stripe).

9. Quellen

- OpenAPI/Redoc: https://www.vbotickets.com/eventmanager/v3.0/api_docs.html
- Spec: https://vblobprod.blob.core.windows.net/awsvboticketscom/_images/master/v1.json
- Help: <https://help.get-vibe.de/595574-Grundlagen-der-Schnittstelle-API>
- Anlage Event-API-Felder: Event API Egocentric Systems (Help-Anhang)

Diese Zusammenfassung ist eine orientierende Kurzfassung und ersetzt nicht die offizielle API-Dokumentation. Bei Abweichungen gelten OpenAPI-Spec und Herstellerangaben.