

Nachbuchungsanfragen per API abrufen und bearbeiten (V6-REST-API)

Über die V6-REST-API können Sie Nachbuchungsanfragen Ihrer Publisher automatisiert abrufen, prüfen, annehmen, ablehnen und bearbeiten. Die API bietet dieselben Funktionen wie das **Nachbuchungstool** in der Oberfläche. Jede Entscheidung, die Sie per API treffen, erscheint dort ebenfalls und wird in der Historie der Anfrage mit Zeitpunkt und Benutzer protokolliert.

1. Zugang

Basis-URL

Rolle	Basis-URL
Advertiser-Login	<code>https://<Ihre-Domain>/ws/V6/advertiser/REST/</code>
Admin-Login (Netzwerkbetreiber)	<code>https://<Ihre-Domain>/ws/V6/admin/REST/</code>

`<Ihre-Domain>` ist die Domain Ihrer Installation, also dieselbe Adresse, unter der Sie sich in der Oberfläche anmelden.

Authentifizierung

Jeder Aufruf braucht drei HTTP-Header:

Header	Wert
<code>X-Auth-ID</code>	ID Ihres Benutzers bzw. Advertiser-Logins
<code>X-Auth-Token</code>	API-Token dieses Benutzers
<code>X-Network-ID</code>	<code>1</code>

Wo Sie den Token finden: Im Menü unter *API* → *Daten-API*. Dort wird Ihr API-Token angezeigt.

Berechtigung

Der Benutzer braucht das Recht **Nachbuchungstool** (`bookingrequests_tool`). Zum Abrufen genügt Leserecht, für Entscheidungen und Änderungen ist Lese- und Schreibrecht nötig. Die API arbeitet immer mit genau den Rechten des Benutzers, dessen Token Sie verwenden. Ein Advertiser-Login sieht nur die Nachbuchungen seiner eigenen Kampagnen.

2. Endpunkte im Überblick

Methode	Pfad	Zweck
GET	Bookingrequests	Liste der Nachbuchungsanfragen (mit Filtern), optional als CSV
GET	Bookingrequests/{id}	eine Anfrage mit Dubletten, Gutschein-Prüfung, Trigger-Liste und Historie
POST	Bookingrequests	Aktionen: annehmen, ablehnen, Trigger zuweisen, Provisionsvorschau, CSV-Import
PUT	Bookingrequests/{id}	eine Anfrage bearbeiten (Trigger, Warenkorbwert, Sale/Lead, Provision)
GET / PUT / POST / DELETE	Bookingrequesthints	Hinweise für Publisher (Text und Screenshots) pflegen

Die vollständige technische Beschreibung mit allen Feldern finden Sie in der OpenAPI-/Swagger-Dokumentation unter `https://<Ihre-Domain>/ws/docs/v6`. Dort melden Sie sich mit Login-ID und Token an.

3. Nachbuchungsanfragen abrufen

```
curl "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests?params[state]=open&params[limit]=200" \
-H "X-Auth-ID: <ID>" \
-H "X-Auth-Token: <API-TOKEN>" \
-H "X-Network-ID: 1"
```

Filter

Parameter	Werte	Bedeutung
-----------	-------	-----------

<code>params[state]</code>	<code>open</code> , <code>accepted</code> , <code>rejected</code> , <code>all</code> (auch als Liste, z. B. <code>open,rejected</code>)	offen (wartet auf Entscheidung), angenommen, abgelehnt, alle
<code>params[date_field]</code>	<code>request</code> (Standard), <code>order</code>	worauf sich <code>from</code> / <code>to</code> beziehen: Eingang der Anfrage oder Kaufdatum
<code>params[from]</code> , <code>params[to]</code>	YYYY-MM-DD	Zeitraum
<code>params[search]</code>	Text	Order-ID, Bestellnummer (auch Teil), Publisher-ID, Projekt-ID oder Anfrage- Nr.
<code>params[campaign_id]</code>	ID oder Liste	nur bestimmte Kampagnen (Programme)
<code>params[publisher_id]</code>	ID oder Liste	nur bestimmte Publisher
<code>params[project_id]</code>	ID oder Liste	nur bestimmte Publisher-Projekte
<code>params[request_id]</code>	ID oder Liste	nur bestimmte Anfragen (hochgeladene Dateien)
<code>params[duplicates]</code>	<code>only</code> , <code>none</code>	nur Dubletten / keine Dubletten
<code>params[voucher]</code>	<code>with</code>	nur Anfragen mit Gutscheincode
<code>params[sort]</code>	<code>request_timestamp</code> (Standard), <code>order_timestamp</code> , <code>turnover</code> , <code>commission</code> , <code>id</code>	Sortierfeld
<code>params[dir]</code>	<code>desc</code> (Standard), <code>asc</code>	Sortierrichtung
<code>params[limit]</code>	Standard <code>50</code> , höchstens <code>1000</code>	Anzahl je Seite
<code>params[offset]</code>	Standard <code>0</code>	Blättern
<code>params[format]</code>	<code>json</code> (Standard), <code>csv</code>	<code>csv</code> liefert alle passenden Zeilen als CSV-Text im Feld <code>csv</code>

Jede Zeile enthält Hinweise (`warnings`), die Ihnen bei der Entscheidung helfen:

Hinweis	Bedeutung
<code>duplicate</code>	Es gibt bereits eine Transaktion mit derselben Bestellnummer, demselben Trigger und derselben Kampagne
<code>voucher_unknown</code>	Gutscheincode ist beim Programm nicht hinterlegt
<code>voucher_other_publisher</code>	Gutscheincode ist einem anderen Publisher zugeordnet
<code>voucher_expired</code>	Gutscheincode war zum Kaufzeitpunkt nicht gültig
<code>voucher_used</code>	Gutscheincode wurde bereits für eine andere Bestellung genutzt
<code>commission_changed</code>	Die neu berechnete Provision weicht von der angefragten ab
<code>commission_rule_missing</code>	Für diesen Trigger ist keine Provision hinterlegt

Trigger einer Kampagne abfragen

```
curl "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests?params[view]=triggers&params[campaign_id]=1" \
-H "X-Auth-ID: <ID>" -H "X-Auth-Token: <API-TOKEN>" -H "X-Network-ID: 1"
```

Eine einzelne Anfrage

```
curl "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests/123456" \
-H "X-Auth-ID: <ID>" -H "X-Auth-Token: <API-TOKEN>" -H "X-Network-ID: 1"
```

Neben den Grunddaten liefert die Detailansicht:

- **Dubletten:** bestehende Transaktionen mit derselben Bestellnummer, demselben Trigger und derselben Kampagne, unabhängig vom Publisher,
- **Gutschein-Prüfung:** ob der Code unbekannt ist, einem anderen Publisher gehört, zum Kaufzeitpunkt ungültig war oder schon eingelöst wurde,
- die **verfügbaren Trigger** der Kampagne,
- die **Historie** aller Änderungen.

“ Diese Prüfungen sind Hinweise für Ihre Entscheidung. Sie blockieren nichts.

4. Annehmen, ablehnen, Trigger zuweisen

Alle Aktionen laufen über `POST Bookingrequests` mit einem JSON-Body. In `order_ids` können Sie mehrere Anfragen auf einmal übergeben.

Annehmen

```
curl -X POST "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests" \
-H "X-Auth-ID: <ID>" -H "X-Auth-Token: <API-TOKEN>" -H "X-Network-ID: 1" \
-H "Content-Type: application/json" \
-d '{ "action": "accept", "order_ids": [123456, 123457], "status": 1 }'
```

status	Ergebnis
0	übernommen als offene Transaktion
1	übernommen als bestätigte Transaktion

Beim Annehmen wird die Provision nach den Kampagneneinstellungen zum Kaufzeitpunkt neu berechnet. Das gilt nicht, wenn sie vorher manuell überschrieben wurde (siehe Abschnitt 5).

Ablehnen

```
curl -X POST "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests" \
  -H "X-Auth-ID: <ID>" -H "X-Auth-Token: <API-TOKEN>" -H "X-Network-ID: 1" \
  -H "Content-Type: application/json" \
  -d '{ "action": "reject", "order_ids": [123458], "reject_code": "already_booked", "reason":
  "Bestellung bereits über Partner X getrackt" }'
```

Ein Ablehnungsgrund (`reject_code`) ist Pflicht:

reject_code	Bedeutung
already_booked	bereits vorhanden / schon gebucht
cancelled	Bestellung storniert
other_channel	anderer Kanal
no_valid_sale	kein valider Sale
other	Sonstiges (dann ist <code>reason</code> als Freitext Pflicht)

Der Publisher sieht Grund und Text.

Trigger zuweisen (mehrere Anfragen derselben Kampagne)

```
{ "action": "trigger", "order_ids": [123456, 123457], "trigger_id": 2 }
```

Provisionsvorschau (ohne zu speichern)

```
{ "action": "preview", "order_id": 123456, "trigger_id": 2, "turnover": 89.90 }
```

Die Vorschau liefert die Provision, die sich mit diesem Trigger und Warenkorbwert ergeben würde. Mit `"lead": true` wird sie für einen Lead ohne Warenkorbwert berechnet. Für die Vorschau genügt Leserecht.

Die Aktionen `accept`, `reject` und `trigger` liefern je Order ein eigenes Ergebnis. So sehen Sie, welche Orders verarbeitet wurden und welche nicht.

5. Eine Anfrage bearbeiten

```
curl -X PUT "https://<Ihre-Domain>/ws/V6/advertiser/REST/Bookingrequests/123456" \
  -H "X-Auth-ID: <ID>" -H "X-Auth-Token: <API-TOKEN>" -H "X-Network-ID: 1" \
  -H "Content-Type: application/json" \
  -d '{ "trigger_id": 1, "turnover": 79.90 }'
```

Feld	Bedeutung
<code>trigger_id</code>	Trigger/Aktion ändern
<code>turnover</code>	Warenkorbwert ändern
<code>lead</code>	als Lead statt Sale behandeln
<code>commission_manual</code> + <code>commission</code>	Provision manuell überschreiben (für Sonderfälle)

“ Eine manuell überschriebene Provision wird markiert und beim Annehmen **nicht** neu berechnet.

6. Entscheidungen gesammelt per CSV übergeben

Viele Entscheidungen auf einmal können Sie auch als CSV übergeben. Das Format ist dasselbe wie beim Import im Nachbuchungstool:

Spalte	Inhalt
<code>order_id</code>	ID der Nachbuchungsanfrage
<code>action</code>	<code>accept</code> , <code>reject</code> oder <code>trigger</code>
<code>status</code>	bei <code>accept</code> : <code>0</code> oder <code>1</code>
<code>reject_code</code> , <code>reason</code>	bei <code>reject</code>
<code>trigger_id</code>	bei <code>trigger</code>

```
{ "action": "import", "content":  
  "order_id;action;status;reject_code;reason;trigger_id\n123456;accept;1;;; \n123458;reject;;;cancelled;;", "apply": true }
```

Ohne `apply` (Standard `false`) wird die Datei nur geprüft. Übernommen wird erst ein Aufruf mit `"apply": true`. Wir empfehlen, immer zuerst einen Prüflauf zu machen.

7. Hinweise für Publisher pflegen

Über `Bookingrequesthints` (`GET` / `PUT` / `POST` / `DELETE`) pflegen Sie den Hinweistext mit bis zu sechs Screenshots. Publisher sehen ihn beim Einreichen einer Nachbuchung, zum Beispiel mit der Angabe, wo die Bestellnummer steht oder welche Daten nicht mitgeschickt werden sollen. Es gibt einen allgemeinen Hinweis für alle Kampagnen und optional einen eigenen je Kampagne, der dann Vorrang hat.

8. Benachrichtigung bei neuen Anfragen

Unabhängig von der API erhalten Sie bei jeder neuen Nachbuchungsanfrage eine E-Mail an Ihre Kontaktadresse. Sie enthält Kampagne, Publisher, Anzahl, Warenkorbwert, mögliche Dubletten, Gutschein-Hinweise und die Zahl aller offenen Anfragen. Die Mail eignet sich gut als Auslöser für einen API-Abruf. Alternativ fragen Sie die API regelmäßig ab, zum Beispiel einmal pro Stunde mit `params[state]=open`.

9. Typischer Ablauf einer Integration

1. **Abrufen:** `GET Bookingrequests?params[state]=open`
 2. **Abgleichen:** Bestellnummer und Warenkorbwert mit Ihrem Shop- oder ERP-System vergleichen. Optional die Detailansicht `GET Bookingrequests/{id}` für Dubletten und Gutscheine nutzen.
 3. **Korrigieren (optional):** `PUT Bookingrequests/{id}` bei abweichendem Warenkorbwert oder Trigger.
 4. **Entscheiden:** `POST Bookingrequests` mit `accept` (Status 0 oder 1) bzw. `reject` und Grund.
-

10. Fehlercodes

HTTP	Bedeutung
400	Eingabe ungültig; die Antwort nennt den Grund
403	Zugangsdaten falsch, fehlendes Recht „Nachbuchungstool“ oder Funktion für die Installation nicht freigeschaltet
404	Anfrage nicht gefunden bzw. nicht für diesen Zugang sichtbar
405	Methode nicht unterstützt: Nachbuchungen werden abgelehnt (<code>action: reject</code>), nicht gelöscht

Fehlerantworten haben immer einen JSON-Body mit `error: true` und einer Meldung in `msg`. Bei ungültigen Zeilen steht zusätzlich die Liste `errors` dabei.

Bei Fragen zur Einrichtung wenden Sie sich gern an unseren Support.

Revision #1

Created 14 October 2026 13:40:36 by Daniel Hörenkamp

Updated 14 October 2026 13:45:30 by Daniel Hörenkamp