Create / easy.affiliate REST-API (Orders)

□□ Einleitung

Um neue Orders automatisiert im easy.affiliate-System anzulegen, steht ein REST-konformer Webservice zur Verfügung. Über die HTTP-Methode **POST** können neue Transaktionen übermittelt werden. Jeder User erhält hierfür einen Authentifizierungs-Token und eine Login-ID, die im Frontend einsehbar sind.

□□ Authentifizierung

Für den Zugriff sind folgende Header erforderlich:

Header	Beschreibung	Typ
Content-Type	Muss auf `application/json` gesetzt sein	String
X-Network-ID	Netzwerk-ID: meist `-1` für mandantenübergreifende API	Integer
X-Auth-Token	Dein API-Token (im Frontend sichtbar)	String
X-Auth-ID	Deine Login-ID (im Frontend sichtbar)	Integer

□□ Endpunkte

- **Admin:** `https://SUBDOMAIN.de/ws/V6/admin/JSON/Orders`
- **Advertiser:** `https://SUBDOMAIN.de/ws/V6/advertiser/JSON/Orders`

□□ Beispiel (cURL)

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "X-Network-ID: -1" \
  -H "X-Auth-Token: ADMIN_APIUSER_TOKEN" \
  -H "X-Auth-ID: ADMIN_APIUSER_LOGIN_ID" \
  -d '[{
    "ordertoken": "testorder001",
    "campaign_id": 1,
    "trigger_id": 2,
    "status": 1,
    "turnover": 199.99,
    "trigger_tstamp": "1785315641"
  }]' \
  https://SUBDOMAIN.de/ws/V6/admin/JSON/Orders
```

□□ Body-Parameter

Parameter	Beschreibung	Datentyp	Pflichtfeld
ordertoken	Eindeutiger Identifier für die Order. Muss systemweit eindeutig sein. Wird kein Wert übergeben, generiert das System automatisch einen (`AUTOID_…`).	String	□ Ja
campaign_id	ID der Kampagne, der die Order zugeordnet wird.	Integer	□ Ja
trigger_id	ID des Triggers, der bei Nutzung von `ordertoken` zwingend angegeben werden muss. Alternativ per `trigger_title`.	Integer	□ Ja
status	Status der Order: `0` = offen, `1` = bestätigt, `2` = storniert. *(Bei der Anlage immer `0`.)*	Integer	□ Ja
turnover	Bestellwert in Kampagnenwährung. Komma und Punkt als Dezimaltrenner erlaubt.	Float	□ Ja

Parameter	Beschreibung	Datentyp	Pflichtfeld
commission	Provisionswert in Kampagnenwährung. Kann übergeben werden, falls selbst berechnet.	Float	Nein
cancel_reason	Angabe eines Stornogrundes bei Status = 2 (storniert).	String	Nein
currency	Bestellwährung als 3-stelliger ISO-Code (z. B. `EUR`, `CHF`). Default = Kampagnen-Einstellung, sonst `EUR`. Umrechnungskurs wird automatisch gesetzt.	String	Nein
descr	Freitext-Beschreibung der Order (max. 255 Zeichen).	String	Nein
trigger_tstamp	Bestellzeitpunkt als Unix-Timestamp. Fehlt der Wert, wird die aktuelle Zeit genutzt.	Integer	Nein
emid	Klick-/Action-ID aus dem Tracking (24-stellig). Dient der Klick-Zuordnung und kann die `campaign_id` auflösen.	String	Nein*
vc	Gutscheincode-Tracking: löst Kampagne/Werbemittel über die Admedia-Einstellungen auf.	String	Nein*
project_id	Direkte Zuordnung per Projekt (Publisher). Erzeugt eine Track-Action, sofern eine Connection zur Kampagne besteht.	Integer	Nein*
subid	Attributionswert (SubID), wird am Auftrag gespeichert (max. 500 Zeichen).	String	Nein
sub_status	Substatus der Order (frei belegbar).	String	Nein
get_parameters	Objekt `{ "key": "value", … }` mit beliebigen Zusatzfeldern (z. B. `voucher`, `coupon`).	Object	Nein

Parameter	Beschreibung	Datentyp	Pflichtfeld
(beliebiger Key)	Jeder weitere, nicht reservierte Parameter wird als Key/Value am Auftrag gespeichert (Key max. 25 Zeichen).	String	Nein

> * **Kampagnen-/Klick-Zuordnung:** `campaign_id` ist Pflicht, kann aber alternativ über `emid` oder `vc` aufgelöst werden. Für die Zuordnung zur ursprünglichen Klick-/View-Aktion sollte **eine** der Attributions-Methoden (`emid` oder `project_id`) mitgegeben werden.

> **Dublettenschutz:** Dieselbe Kombination aus `campaign_id` + `trigger_id` + `ordertoken` wird innerhalb abgelehnt (`ORDER_EXISTS` / `ORDER_REQUEST_FREEZED`).

□□ Hinweise

Wichtige Implementierungsdetails:

- **ordertoken:** Muss **eindeutig** sein – idealerweise eine Kombination aus Zeitstempel, Shop-ID oder externer Ordernummer.
- **trigger_id:** Wenn du diese nicht kennst, frage im Frontend deine Kampagnenkonfiguration ab.
- **Bulk-Aktionen:** Du kannst auch mehrere Orders in einem Array auf einmal anlegen.
- **Updates:** Wird eine Order mit gleichem `ordertoken` bereits im System gefunden, kann ein `PUT-Request` genutzt werden, um sie zu aktualisieren.

Revision #10
Created 2026-03-12 09:12:47 UTC by Daniel Hörenkamp
Updated 2026-07-29 09:00:51 UTC by Daniel Hörenkamp