

Create / easy.affiliate REST-API (Baskets)

? Einleitung

Um neue Orders **inklusive Warenkorb** automatisiert im easy.affiliate-System anzulegen, steht ein REST-konformer Webservice zur Verfügung. Über die HTTP-Methode `POST` wird pro Aufruf eine Bestellung mit ihren Warenkorb-Positionen übermittelt.

Die Bestellung durchläuft dabei dieselbe Logik wie ein getrackter Verkauf mit Warenkorb (`/trck/ebasket/`): Publisher-Zuordnung, Provision je Position (Regeln nach Artikel, Marke, Kategorie oder Default des Triggers), Bestellwert sowie serverseitige Tags und Postbacks der Kampagne. **Bestellwert, Provision und Status werden immer vom System berechnet** und können nicht übergeben 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
<code>Content-Type</code>	Muss auf <code>application/json</code> gesetzt sein	String
<code>X-Network-ID</code>	Netzwerk-ID des Mandanten (muss > 0 sein, nicht <code>-1</code>). Die Kampagne muss zu diesem Mandanten gehören.	Integer
<code>X-Auth-Token</code>	Dein API-Token (im Frontend sichtbar)	String
<code>X-Auth-ID</code>	Deine Login-ID (im Frontend sichtbar)	Integer

Berechtigung: Nur Admin-Logins mit Lese- und Schreibrecht auf die Bestellungen (`administration_orders`). Der Login muss zum Mandanten aus `X-Network-ID` gehören.

? Endpunkte

- **Admin:** `https://SUBDOMAIN.de/ws/V6/admin/REST/Baskets`

? 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 '{
    "campaign_id": 123,
    "trigger_id": 5,
    "ordertoken": "E-10045",
    "emid": "5f0123456789abcdef012345",
    "basket": [
      {"article_number": "A-1", "productname": "Hemd", "category": "Herren", "amount": 2,
"price": 19.90, "additional": {"brand": "X"}},
      {"article_number": "B-7", "amount": 1, "price": 5}
    ]
  }' \
  https://SUBDOMAIN.de/ws/V6/admin/REST/Baskets
```

? Body-Parameter (Order)

Parameter	Beschreibung	Datentyp	Pflichtfeld
<code>campaign_id</code>	ID der Kampagne, der die Order zugeordnet wird.	Integer	☐ Ja
<code>trigger_id</code>	ID eines Triggers dieser Kampagne (z. B. Sale-Trigger).	Integer	☐ Ja
<code>ordertoken</code>	Eindeutiger Identifier für die Order (eigene Bestellnummer). 1-255 druckbare ASCII-Zeichen, ohne Leerzeichen und ohne <code>() ;</code> .	String	☐ Ja

Parameter	Beschreibung	Datentyp	Pflichtfeld
<code>basket</code>	Liste der Warenkorb-Positionen (min. 1, max. 200), siehe unten.	Array	☐ Ja
<code>emid</code>	Klick-/Action-ID aus dem Tracking (24-stellig). Muss zu einem Klick dieser Kampagne gehören.	String	Nein*
<code>project_id</code>	Direkte Zuordnung per Projekt (Publisher). Das Projekt muss mit der Kampagne verbunden sein.	Integer	Nein*
<code>currency</code>	Bestellwährung als 3-stelliger ISO-Code (z. B. EUR, CHF). Default = Kampagnen-Einstellung, sonst EUR. Unbekannte Währungen werden abgelehnt.	String	Nein
<code>trigger_tstamp</code>	Bestellzeitpunkt als Unix-Timestamp (Sekunden). Fehlt der Wert, wird die aktuelle Zeit genutzt.	Integer	Nein
<code>descr</code>	Freitext-Beschreibung der Order (max. 255 Zeichen).	String	Nein

* **Publisher-Zuordnung:** Es darf **entweder** `emid` **oder** `project_id` übergeben werden, nicht beides. Beide werden vorab geprüft: Eine unbekannte/abgelaufene `emid`, eine `emid` einer anderen Kampagne oder ein nicht verbundenes Projekt wird abgelehnt, statt die Order still dem Systempartner zuzuordnen. Ohne beide Angaben entscheidet die Kampagne, ob die Order an den Systempartner geht (Antwort enthält dann `notice: NO_SESSION_FOUND`).

Nicht erlaubt: `turnover`, `commission`, `status` und alle weiteren, nicht aufgeführten Felder werden mit einem Fehler abgelehnt.

? Body-Parameter (Warenkorb-Position in `basket`)

Parameter	Beschreibung	Datentyp	Pflichtfeld
<code>article_number</code>	Artikelnummer / SKU (1-255 Zeichen). Provisionsregeln nach Artikel greifen darauf.	String	☐ Ja

Parameter	Beschreibung	Datentyp	Pflichtfeld
amount	Menge, ganze Zahl ≥ 1 .	Integer	☐ Ja
price	Preis einer Einheit, ≥ 0 , nur Punkt als Dezimaltrenner (z. B. 19.90), max. 4 Nachkommastellen. Bei Brutto-Tracking der Kampagne wird die MwSt. wie beim Tracking abgezogen.	Float	☐ Ja
productname	Produktname (max. 255 Zeichen).	String	Nein
category	Kategorie für Provisionsregeln nach Kategorie. Leer = Artikelnummer.	String	Nein
additional	Objekt { "key": "value", ... } mit max. 50 flachen Zusatzfeldern, wird an der Position gespeichert. additional.brand wird für Provisionsregeln nach Marke genutzt. Keys: 1-50 Zeichen aus A-Z a-z 0-9 _ - ; reservierte Keys (z. B. currency, emid, campaign_id, commission, turnover) sind nicht erlaubt.	Object	Nein

Der Bestellwert der Order ergibt sich aus der Summe `price × amount` aller Positionen, die Provision aus der Summe der Positions-Provisionen.

Dublettenschutz: Dieselbe Kombination aus `campaign_id` + `trigger_id` + `ordertoken` wird abgelehnt (`ORDER_EXISTS`), die Antwort enthält die `order_id` der vorhandenen Order. Ein erneuter Aufruf, z. B. nach einem Timeout, ist dadurch gefahrlos möglich.

? Antwort

Erfolg:

```
{
  "order_inserted": true,
  "order_id": 4711815,
  "emid": "5f0123456789abcdef012345",
```

```
"sessionId": "5f0123456789abcdef000001",
"basket": [
  {"id": 981234, "productid": "A-1", "productname": "Hemd"},
  {"id": 981235, "productid": "B-7", "productname": ""}
]
```

Fehler: `order_inserted: false`, `order_id: -1`, dazu `error` und `error_description`. Alle Eingabefehler werden gesammelt in einer Antwort geliefert.

Error	Bedeutung
<code>INVALID_REQUEST</code>	Eingabe ungültig (Pflichtfeld fehlt, Format falsch, unbekanntes Feld, Trigger gehört nicht zur Kampagne, <code>emid</code> / <code>project_id</code> passt nicht, Währung unbekannt). Details in <code>error_description</code> .
<code>ORDER_EXISTS</code>	Order mit diesem <code>ordertoken</code> existiert bereits für Kampagne und Trigger; <code>order_id</code> enthält die vorhandene Order.
<code>TRACKING_UNAVAILABLE</code>	Das Tracking hat nicht geantwortet, die Order wurde vermutlich nicht angelegt. Erneut senden (Dublettenschutz greift).
sonstige	Ablehnung durch die Tracking-Logik bzw. Kampagnen-Einstellungen.

? Hinweise

Wichtige Implementierungsdetails:

- **Ein Aufruf = eine Order:** Der Body ist ein einzelnes JSON-Objekt, kein Array. Mehrere Orders werden mit mehreren Aufrufen angelegt.
- **ordertoken:** Muss je Kampagne und Trigger eindeutig sein – idealerweise die externe Bestellnummer des Shops.
- **trigger_id:** Wenn du diese nicht kennst, frage im Frontend deine Kampagnenkonfiguration ab.
- **Preise:** Immer mit Punkt als Dezimaltrenner übergeben (`12.50`, nicht `12,50`).

Revision #1

Created 15 October 2026 14:53:33 by Daniel Hörenkamp

Updated 15 October 2026 14:54:26 by Daniel Hörenkamp