

Admin API - Dokumentation

- [Basket API - Dokumentation](#)
- [Transaktions API - Dokumentation](#)
- [Statistik API - Dokumentation](#)
- [Touchpoints / Customerjourney API - Dokumentation](#)
- [Order Webservice](#)
 - [Create / easy.affiliate REST-API \(Orders\)](#)
 - [Update / easy.affiliate REST-API \(Orders\)](#)
- [Publisher Webservice](#)
 - [Verwaltung von Projekten über die V6 REST-API](#)
 - [Publisher-Verwaltung über die V6 REST-API](#)
- [Externe API anbinden](#)
- [Werbemittel API](#)
 - [Admedia API \(V6\) - POST \(Create\)](#)
 - [Admedia API \(V6\) - GET \(Read\)](#)
- [Basket API](#)
 - [Basket Update API](#)

Basket API – Dokumentation

Die Basket API dient dem Export und der Analyse von Warenkorb-Daten, die über das Partnerprogramm-System erfasst wurden. Sie ermöglicht es, detaillierte Informationen zu einzelnen Basket-Items strukturiert als CSV-Datei abzurufen und weiterzuverarbeiten.

Die API kann im User-Interface unter dem Menüpunkt **Daten-API** verwendet werden.

API-Aufruf

https://www.domain.de/api//ACCESS-TOKEN/admin/ID/get-orders_baskets.csv

Wert	Beschreibung
www.domain.de	System-Partnerprogramm
ACCESS-TOKEN	Access Token des Users
ID	User-ID
.csv	Art des Downloads

Basket-API Filterung

Für die Basket API können verschiedene Parameter als Filter verwendet werden.

Nachfolgend eine Übersicht aller verfügbaren Parameter:

Parameter	Wert / Beispiel	Beschreibung
<code>condition[period][from]</code>	01.01.2025	Definiert das Startdatum des Zeitraums
<code>condition[period][to]</code>	31.01.2025	Definiert das Enddatum des Zeitraums
<code>condition[l:campaigns]</code>	1	Filtert alle Baskets, die einer bestimmten Kampagne zugeordnet sind. Mehrfachnennung möglich.

Parameter	Wert / Beispiel	Beschreibung
condition[datemode]	processing payout incoming	Bestimmt die Art des Zeitraums: <ul style="list-style-type: none"> processing: Baskets, die im Zeitraum validiert wurden payout: Baskets, die im Zeitraum ausgezahlt wurden incoming: Baskets, die im Zeitraum erfasst wurden (Transaktionsdatum)

Beispiel:

condition[period][from]=01.01.2025&condition[period][to]=31.01.2025&condition[datemode]=processing

Definition der Felder in der Basket-API

Die heruntergeladenen Transaktionen (z.B. als `.csv`) enthalten folgende Spalten:

Feldname	Beschreibung
id	Eindeutige ID des Basket-Items
orderid	Interne ID der zugehörigen Transaktion
timestamp	Zeitpunkt der Transaktion
ordertoken	Bestellnummer der Transaktion
trigger_id	ID des Triggers, der der Transaktion zugeordnet ist
is_basket	Gibt an, ob es sich um ein Basket-Item handelt (<code>true</code> / <code>false</code>)
basket_item_id	Eindeutige ID des Basket-Items
category	Übergebene Kategorie des Shops
productid	ID des Produkts im Shop
productname	Name des Produkts im Shop
amount	Anzahl der gekauften Produkte
price	Stückpreis des Produkts
original_price	Ursprünglicher Stückpreis vor Änderungen
status	Status des Basket-Items
total	Gesamtpreis (Anzahl × Preis)
commission	Provision, die der Publisher für das Basket-Item erhält
currency_rate	Umrechnungsrate bei Fremdwährung
currency	Währung der Transaktion

Feldname	Beschreibung
is_attributed	Gibt an, ob eine Attribution vorliegt (true / false)
attribution	Art der Attribution pro Basket-Item
attributed_price	Attribuierter Preis pro Basket-Item
attributed_total	Attribuierter Preis × Anzahl
trigger_value	Provisionshöhe pro Basket-Item
original_turnover	Getrackter Umsatz
price_brutto	Brutto-Preis des Basket-Items

Transaktions API - Dokumentation

Die Transaktions API ermöglicht den strukturierten Export und die Analyse von Transaktionen, die im Rahmen des Partnerprogramm-Systems erfasst wurden. Sie bietet detaillierte Einblicke in einzelne Transaktionsvorgänge und unterstützt bei der Abrechnung, Erfolgsmessung sowie der Integration in interne oder externe Systeme.

Die API kann im User-Interface unter dem Menüpunkt **Daten-API** verwendet werden.

API-Aufruf

<https://www.domain.de/api//ACCESS-TOKEN/admin/ID/get-orders.csv>

Wert	Beschreibung
www.domain.de	Systemdomain
ACCESS-TOKEN	Access Token des Users
ID	User-ID
.csv	Art des Downloads

Transaktions-API Filterung

Für die Transaktions API können verschiedene Parameter als Filter verwendet werden. Nachfolgend eine Übersicht aller verfügbaren Parameter:

Transaktions Status

Parameter	Wert	Beispiel	Beschreibung
<code>condition[l:status]</code>	<code>open</code>	<code>condition[l:status]=open</code>	Alle Transaktionen, welche als "Offen" im System hinterlegt sind
	<code>confirmed</code>	<code>condition[l:status]=confirmed</code>	Alle Transaktionen, welche als "Bestätigt" im System hinterlegt sind

Parameter	Wert	Beispiel	Beschreibung
	paid	condition[l:status]=paid	Alle Transaktionen, welche als "Ausgezahlt" im System hinterlegt sind
	anceled	condition[l:status]=anceled	Alle Transaktionen, welche als "Storniert" im System hinterlegt sind

Hinweis: Es ist eine Kombination aus mehreren Status möglich. Zum Beispiel:

condition[l:status]=confirmed,paid

Parameter	Wert / Beispiel	Beschreibung
condition[period][from]	01.01.2025	Definiert das Startdatum des Zeitraums
condition[period][to]	31.01.2025	Definiert das Enddatum des Zeitraums
condition[l:campaigns]	1	Filtert alle Baskets, die einer bestimmten Kampagne zugeordnet sind. Mehrfachnennung möglich.
condition[datemode]	processing payout incoming	Bestimmt die Art des Zeitraums: <ul style="list-style-type: none"> processing: Baskets, die im Zeitraum validiert wurden payout: Baskets, die im Zeitraum ausgezahlt wurden incoming: Baskets, die im Zeitraum erfasst wurden (Transaktionsdatum)

Beispiel:

condition[period][from]=01.01.2025&condition[period][to]=31.01.2025&condition[datemode]=processing

Definition der Felder in der Transaktions-API

Die heruntergeladenen Transaktionen (z.B. als `.csv`) enthalten folgende Spalten:

Feldname	Beschreibung
id	Eindeutige ID der Transaktion
status	Status der Transaktion
timestamp	Zeitpunkt der Transaktion

Feldname	Beschreibung
campaign_id	Kampagne, der die Transaktion zugeordnet wurde
attribution	Betrag der gezahlten Attribution
delivered_tagcode_count	Anzahl der ausgeführten Tags
delivered_tagcode_serversided_url	Übermittelter Server-Side-Trackingcall
ordertoken	Vom Advertiser übermittelte Bestellnummer
source	Quelle der Transaktion
project_id	ID des Projekts, dem die Transaktion zugeordnet wurde
admedia_id	Werbemittel-ID, über die die Transaktion erzeugt wurde
type	Typ der Transaktion (z.B. <i>Sale, Lead, View</i>)
commission	Provision, die dem Publisher zugeordnet wurde (wird beim Salesabgleich geupdated)
source_commission	Ursprüngliche Provision (wird nicht geupdated)
commission_group_id	Provisionsgruppe, in der sich der Publisher bei Transaktionserzeugung befand
trigger_id	Trigger-ID, die der Transaktion zugeordnet ist
description	Beschreibung, die bei der Transaktion übermittelt wurde
trigger_value	Vergütung, die in der entsprechenden Trigger-ID zum Zeitpunkt der Transaktion hinterlegt war
trigger_type	Art des Triggers (Fixvergütung oder prozentuale Vergütung)
turnover	Provisionsrelevanter Bestellwert
attributed_turnover	Attribuierter Bestellwert (falls Attributionsmodell aktiv)
original_turnover	Bestellwert in der Advertiser-Währung
action_id	Action-ID, die bei der Transaktion übermittelt wurde
salary_id	ID des Auszahlungsvorgangs, in dem die Transaktion abgerechnet wurde
session_id	Session-ID, die bei der Transaktion übermittelt wurde
order_currency	Währung der Bestellung
status_change_date	Datum der letzten Statusänderung
cancel_reason	Stornogrund, falls die Transaktion storniert wurde
source_turnover	Vom Advertiser übermittelter Bestellwert
last_change	Timestamp der letzten Änderung der Transaktion
user_agent	Clientinformationen des Nutzers, der die Transaktion ausgelöst hat

Feldname	Beschreibung
order_actions_id	Interner Verweis auf den provisionsrelevanten Klick oder Postview
bonus_id	ID eines Bonus-Laufs
customer_journey_status	Interne Kennziffer zur Customer Journey
ebestid	Gehashte Bestellnummer zur Anzeige für den Publisher
order_timestamp	Timestamp des Orderaufrufs
action_timestamp	Timestamp der Action, die zur Order führte
trigger_title	Name des Triggers
payoutdate	Auszahlungsdatum
campaign_group_title	Titel der Provisionsgruppe
project_title	Name des Projekts, dem die Transaktion zugeordnet wurde
campaign_title	Name der Kampagne, die der Transaktion zugeordnet wurde
advertiser_id	ID des Advertisers, dem die Transaktion zugeordnet wurde
advertiser_title	Name des Advertisers, dem die Transaktion zugeordnet wurde
publisher_id	ID des Publishers, dem die Transaktion zugeordnet wurde
referrer	Seite, auf der der klickauslösende Vorgang stattfand
country	Land, in dem die Transaktion erzeugt wurde
subid	Sub-ID, die bei der Transaktion übermittelt wurde
publisher_prenome	Vorname des Publishers (aus Systemdaten)
publisher_surname	Nachname des Publishers (aus Systemdaten)
publisher_searchtitle	Suchtitel des Publishers (aus Systemdaten)
basket_items	Anzahl der Artikel im Warenkorb
sub_status	Advertiser-interner Status zur Verwaltung der Transaktion
visibility	Sichtbarkeit für den Publisher
external_sources_entity_id	(Account) Id der externen Quelle
external_sources_currency_code	Währungscode aus der externen Quelle
external_sources_currency_rate	Umrechnungsrate anhand des external_sources_currency_code und der internen Währung
external_sources_turnover	Bestellwert aus der externen Quelle
source_project_id	Publisher/Project-ID aus der externen Quelle
source_project_title	Publisher/Project-Name aus der externen Quelle

Feldname	Beschreibung
turnover_brutto	Bestellwert inkl. Steuern

Statistik API - Dokumentation

Die Statistik API ermöglicht den Export und die Auswertung aggregierter Leistungsdaten im Rahmen des Partnerprogramm-Systems. Sie stellt unterschiedliche Statistik-Arten bereit, wie z. B. Auswertungen nach Tag, Kampagne, Projekt oder Werbemittel, und unterstützt so bei der Performance-Analyse und Optimierung.

Alle APIs können im User-Interface unter dem Menüpunkt **Daten-API** verwendet werden.

API-Aufruf

https://www.domain.de/api//ACCESS-TOKEN/admin/ID/get-statistic_daily.csv

Wert	Beschreibung
www.domain.de	System-Partnerprogramm
ACCESS-TOKEN	Access Token des Users
ID	User-ID
get-statistic_{Statistik-Art}	Art der Statistik (z.B. Tagesbasiert, etc.)
.csv	Art des Downloads

Statistik-API Filterung

Für die Statistik API können verschiedene Parameter als Filter verwendet werden.

Nachfolgend eine Übersicht aller verfügbaren Parameter:

Statistik-Arten

Statistik-Art	Parameter
Statistik nach Tag	<code>get-statistic_daily</code>
Statistik nach Advertiser	<code>get-statistic_advertiser</code>
Statistik nach Kampagnen	<code>get-statistic_campaign</code>
Statistik nach Werbemittel	<code>get-statistic_media</code>
Statistik nach Publisher	<code>get-statistic_publisher</code>
Statistik nach Projekt	<code>get-statistic_project</code>

Statistik-Art	Parameter
Statistik nach SubID	get-statistic_subid
Statistik nach Referrer	get-statistic_referrer
Statistik nach Channel	get-statistic_channel
Statistik nach Provisionsgruppe	get-statistic_provisionsgroup
Statistik nach Trigger	get-statistic_trigger
Statistik nach Journey	get-statistic_journey
Statistik nach A/B-Test	get-statistic_abtest
Statistik nach Gerätetyp	get-statistic_devicetype
Statistik nach Gerät	get-statistic_device
Statistik nach Browser	get-statistic_browser
Statistik nach Betriebssystem	get-statistic_os
Statistik nach Land	get-statistic_geocountry
Manuelle Statistik	get-statistic_manualstatistics

Parameter	Wert / Beispiel	Beschreibung
condition[period][from]	01.01.2025	Startdatum der Statistik
condition[period][to]	31.01.2025	Enddatum der Statistik
condition[paymentstatus]	all = alle 0 = nicht ausgezahlt 1 = ausgezahlt	Filtert nach Auszahlungsstatus
condition[dynamicdate]	currentmonth = aktueller Monat lastmonth = vorheriger Monat currentweek = aktuelle Woche last10d = letzte 10 Tage	Dynamische Zeiträume
condition[l:publisher]	1234,5678	Filtert nach Publisher-IDs
condition[project_id]	12345	Filtert nach Projekt-IDs
condition[bruttodata]	true	Gibt Bruttozahlen zurück (z. B. Klicks/Views)

Parameter	Wert / Beispiel	Beschreibung
<code>condition[l:mandant]</code>	1	Filtert nach Mandanten-IDs
<code>condition[l:advertiser]</code>	2,3	Filtert nach Advertiser-IDs
<code>condition[l:campaigns]</code>	1,3	Filtert nach Kampagnen-IDs

Aufschlüsselung nach weiteren Kriterien (Dynamische-Statistik)

Die Statistik kann zusätzlich nach weiteren Kriterien **aufgeschlüsselt (segmentiert)** werden. Hierzu wird der Parameter `condition[l:criteria]` verwendet.

Parameter	Wert	Beschreibung	Beispiel
<code>condition[l:criteria]</code>	<code>day</code>	Aufschlüsselung nach Tag	<code>condition[l:criteria]=day</code>
	<code>publisher</code>	Aufschlüsselung nach Publisher	<code>condition[l:criteria]=publisher</code>
	<code>media</code>	Aufschlüsselung nach Werbemittel	<code>condition[l:criteria]=media</code>
	<code>advertiser</code>	Aufschlüsselung nach Advertiser	<code>condition[l:criteria]=advertiser</code>
	<code>campaign</code>	Aufschlüsselung nach Kampagnenname	<code>condition[l:criteria]=campaign</code>
	<code>campaign_id</code>	Aufschlüsselung nach Kampagnen-ID	<code>condition[l:criteria]=campaign_id</code>
Kombination	z. B. <code>day,publisher,media</code>	Mehrdimensionale Aufschlüsselung	<code>condition[l:criteria]=day,publisher,media</code>

Hinzufügen von KPIs

Über den Parameter `condition[l:additional_parameters]` können zusätzliche **Key Performance Indicators (KPIs)** zur Statistik hinzugefügt werden.

Parameter	Wert	Beschreibung	Beispiel
-----------	------	--------------	----------

<code>condition[l:additional_parameters]</code>	<code>all</code>	Fügt alle verfügbaren KPIs hinzu	<code>condition[l:additional_parameters]=all</code>
---	------------------	----------------------------------	---

Touchpoints / Customerjourney API - Dokumentation

Die Customerjourney Orders API dient dem Export und der Analyse von Customer-Journey-Daten, die über das Partnerprogramm-System erfasst wurden. Sie ermöglicht es, detaillierte Informationen zu den einzelnen Kontaktpunkten (Actions) einer Bestellung strukturiert als CSV-Datei abzurufen und die Pfade der Nutzer bis zur Conversion präzise nachzuvollziehen.

Die API kann im User-Interface unter dem Menüpunkt **Daten-API** verwendet werden.

API-Aufruf

```
https://DOMAIN/api/ACCESS-TOKEN/admin/ID/get-customerjourney_orders.csv
```

Wert	Beschreibung
DOMAIN	System-Partnerprogramm Domain
ACCESS-TOKEN	Access Token des Users
ID	User-ID
.csv	Art des Downloads (Dateiformat)

Customerjourney-API Filterung

Für die Customerjourney API können verschiedene Parameter als Filter verwendet werden. Nachfolgend eine Übersicht der verfügbaren Parameter:

Parameter	Wert / Beispiel	Beschreibung
condition[period][from]	2026-02-04	Definiert das Startdatum des Zeitraums (Bestelldatum).
condition[period][to]	2026-02-04	Definiert das Enddatum des Zeitraums (Bestelldatum).
condition[dynamicdate]	currentmonth, lastmonth, last10d, currentweek	Definition eines dynamischen Zeitraums. Hinweis: Kann nicht zusammen mit <code>period[from]</code> oder <code>period[to]</code> verwendet werden.

Parameter	Wert / Beispiel	Beschreibung
<code>condition[l:campaigns]</code>	1	Filtert alle Daten, die einer bestimmten Kampagne zugeordnet sind. Mehrfachnennung möglich.
<code>condition[l:projects]</code>	1234	Alle Transaktionen, welche unter der entsprechenden Projekt-ID des Programms hinterlegt sind. Mehrfachnennung möglich.

Beispiele für gefilterte Aufrufe:

- **Fixer Zeitraum:** `condition[period][from]=2026-02-04&condition[period][to]=2026-02-04&condition[l:campaigns]=1`
- **Dynamisch:** `condition[dynamicdate]=currentweek&condition[l:projects]=1234`

Definition der Felder (Spalten)

Die heruntergeladenen Transaktionen enthalten folgende Spalten:

Feldname	Beschreibung
<code>order_id</code>	Interne ID der zugehörigen Transaktion / Bestellung
<code>ordertoken</code>	Bestellnummer der Transaktion aus dem Shopsystem
<code>campaign_id</code>	ID der Kampagne, welcher die Bestellung zugeordnet ist
<code>order_timestamp</code>	Zeitpunkt des Bestelleingangs (Format: YYYY-MM-DD HH:MM:SS+TZ)
<code>trigger_id</code>	ID des Triggers, der der Bestellung zugeordnet ist
<code>project_id</code>	ID des Projekts (Attribution Target)
<code>admedia_id</code>	ID des Werbemittels der Bestellung
<code>action_timestamp</code>	Zeitpunkt des Klicks oder der Einblendung (Touchpoint)
<code>type</code>	Art des Kontaktpunkts (z.B. click)
<code>action_campaign_id</code>	Kampagnen-ID des spezifischen Kontaktpunkts innerhalb der Journey
<code>action_project_id</code>	Projekt-ID des spezifischen Kontaktpunkts
<code>action_admedia_id</code>	Werbemittel-ID des spezifischen Kontaktpunkts
<code>referrer</code>	Die URL, von der der Nutzer zum Shop geleitet wurde (inkl. Tracking-Parametern)

Order Webservice

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 <code>application/json</code> gesetzt sein	String
X-Network-ID	Netzwerk-ID: meist <code>-1</code> 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,
    "commission": 20.00
  }' \
  https://SUBDOMAIN.de/ws/V6/admin/JSON/Orders

```

? Body-Parameter

Parameter	Beschreibung	Datentyp	Pflichtfeld
ordertoken	Eindeutiger Identifier für die Order. Muss systemweit eindeutig sein.	String	<input type="checkbox"/> Ja
campaign_id	ID der Kampagne, der die Order zugeordnet wird.	Integer	<input type="checkbox"/> Ja
trigger_id	ID des Triggers, der bei Nutzung von <code>ordertoken</code> zwingend angegeben werden muss.	Integer	<input type="checkbox"/> Ja
status	Status der Order: <code>0</code> = offen, <code>1</code> = bestätigt, <code>2</code> = storniert	Integer	<input type="checkbox"/> Ja
turnover	Bestellwert in Kampagnenwährung	Float	<input type="checkbox"/> Ja
original_turnover	Bestellwert in Fremdwährung	Float	Nein
commission	Provisionswert in Kampagnenwährung. Kann übergeben werden, falls selbst berechnet	Float	Nein
cancel_reason	Angabe eines Stornogrundes bei Status = 2 (storniert)	String	Nein

? Hinweise

Wichtige Implementierungsdetails:

- **ordertoken:** Muss **eindeutig** sein – idealerweise eine Kombination aus Zeitstempel, Shop-ID oder externer Ordnernummer.
- **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.

Update / easy.affiliate REST-API (Orders)

? Einleitung

Um Orders automatisiert im **easy.affiliate-System** bearbeiten zu können, stellt die easy Marketing GmbH eine Webservice-API zur Verfügung. Diese ermöglicht es, bestehende Orders im Nachgang zu bearbeiten, sobald sie den Validierungsprozess durchlaufen.

Jedem User stehen hierfür ein **Authentifizierungs-Token** und eine **Login_id** zur Verfügung, die über das Frontend abgerufen werden können.

- **User-ID:** Wo ist die User-ID hinterlegt? (Siehe Frontend-Profil)
- **Access-Token:** Wo ist der Access-Token hinterlegt? (Siehe API-Einstellungen)

Endpunkte

Zielgruppe	URL
Admin	<code>https://SUBDOMAIN.de/ws/V6/admin/JSON/Orders</code>
Advertiser	<code>https://SUBDOMAIN.de/ws/V6/advertiser/JSON/Orders</code>

Beispiele

Beispiel mit cURL

```
curl -X PUT \  
  -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" \  
  <URL>
```

```
-d '[{ "campaign_id": 1, "id": 12345, "status": 1, "turnover":100.00, "commission":12.34 },{
"campaign_id": 1, "ordertoken":"123a456b", "status": 2, "turnover": 0.00 }]' \
https://DOMAIN/ws/V6/admin/JSON/Orders
```

Codebeispiel

Mit der folgenden Methode können mehrere Orders geupdatet werden. Ein Beispielaufruf sieht folgendermaßen aus:

Headers

Variable	Wert / Bedeutung
Content-Type	application/json
X-Network-ID	-1
X-Auth-Token	ADMIN_APIUSER_TOKEN
X-Auth-ID	ADMIN_APIUSER_LOGIN_ID

Body

```
[
  {
    "id": "12345",
    "campaign_id": 1,
    "status": 1,
    "turnover": 10.00,
    "commission": 0.99
  },
  {
    "ordertoken": "123a456b",
    "campaign_id": 1,
    "trigger_id": 1,
    "status": 2,
    "turnover": 9.99,
    "cancel_reason": "Storno Grund"
  }
]
```

Variablenerläuterung

Headers

Variable	Bedeutung	Datentyp
Content-Type	Der Content-Type des Requests	String
X-NETWORKID	Hier wird die ID des Mandanten eingetragen. Wenn nur ein Mandant vorhanden ist oder mandantenübergreifend gearbeitet wird, muss der Wert "-1" eingetragen werden.	Integer
X-AUTH-TOKEN	Hier wird der API-Authentifizierungs-Token des Admin Nutzers hinterlegt.	String
X-AUTH-ID	Hier wird die ID des Admin Nutzers hinterlegt.	Integer

Body

“ Im nachfolgenden werden die Standard-Parameter aufgelistet. Es ist aber über die API möglich, alle zur Verfügung stehenden Parameter zu updaten.

Parameter	Erklärung	Datentyp	Pflichtfeld
id	Hier wird die Transaktionsid hinterlegt. Aufbau: <code>"id": "ID"</code>	String	Ja (oder <code>ordertoken</code>)
ordertoken	Hier wird der Ordertoken hinterlegt. Aufbau: <code>"ordertoken": "ORDERTOKEN"</code>	String	Ja (oder <code>id</code>).*
campaign_id	Hier wird die Campaign_id hinterlegt. Aufbau: <code>"campaign_id": CAMPAIGN_ID</code>	Integer	Ja
trigger_id	Hier wird die Trigger ID hinterlegt. Aufbau: <code>"trigger_id": TRIGGER_ID</code>	Integer	Nein, nur falls <code>ordertoken</code> genutzt wird.

Parameter	Erklärung	Datentyp	Pflichtfeld
status	Hier wird der Status hinterlegt. Werte: <code>-5</code> (ausstehende Nachbuchung), <code>0</code> (offen), <code>1</code> (bestätigt), <code>2</code> (storniert)	Integer	Nein
turnover	Hier wird der Bestellwert der Transaktion in Kampagnenwährung hinterlegt. Aufbau: <code>"turnover":TURNOVER</code>	Float	Nein
original_turnover	Hier wird der Bestellwert der Transaktion in Fremdwährung hinterlegt. Aufbau: <code>"original_turnover":TURNOVER_IN_FREMDWÄHRUNG</code>	Float	Nein
commission	Hier wird die Provisionshöhe in Kampagnenwährung hinterlegt, falls Sie die Commission selbst berechnen möchten.	Float	Nein
cancel_reason	Übergeben Sie den Stornogrund. Aufbau: <code>"cancel_reason":"Stornogrund"</code>	String	Nein

*) **Hinweis:** Wird der `ordertoken` genutzt, ist neben der `campaign_id` auch die `trigger_id` ein Pflichtfeld.

Publisher Webservice

Verwaltung von Projekten über die V6 REST-API

Einleitung

Die V6 REST-API bietet Admins die Möglichkeit, Projekte effizient zu verwalten. Mit dem `/Projects`-Endpoint können Projekte erstellt, ausgelesen, bearbeitet und gelöscht werden. Zusätzlich können Projekte anhand verschiedener Parameter gefiltert werden.

Endpoint

Für Admins: `https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects`

Parameter & Filter

Erforderliche Parameter (bei POST und PUT)

Parameter	Typ	Beschreibung
<code>title</code>	String	Titel des Projekts (Pflichtfeld)
<code>status</code>	Integer	Status des Projekts (Pflichtfeld)
<code>url</code>	String	URL des Projekts (Pflichtfeld)

Filteroptionen (bei GET)

Es kann nach allen Spalten der Tabelle `publisher.projects` gefiltert werden, z. B.:

- `id`
- `status`
- `publisher_id`
- `projecttype`
- `title`

- url
- channel_id
- hidden

Beispiel für GET mit Filtern:

```
curl -X GET -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" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?status=1&publisher_id=10"
```

CRUD-Operationen

GET: Abrufen von Projekten

Abrufen einer Liste von Projekten oder eines spezifischen Projekts anhand der `id`.

```
curl -X GET -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" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?id=5"
```

POST: Erstellen eines neuen Projekts

```
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 '{
  "title": "Neues Projekt",
  "status": 1,
  "url": "https://example.com"
}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects"
```

PUT: Aktualisieren eines bestehenden Projekts

```
curl -X PUT -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 '{
  "id": 5,
  "title": "Geändertes Projekt",
  "status": 2,
  "url": "https://updated-example.com"
}'
```

```
}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects"
```

DELETE: Löschen eines Projekts

Löschen eines Projekts anhand der `id`.

```
curl -X DELETE -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" "https://SUBDOMAIN.de/ws/V6/admin/JSON/Projects?id=5"
```

Antworten (Responses)

Erfolgreiche Antwort

```
{
  "id": 5,
  "status": 1,
  "publisher_id": 10,
  "projecttype": "page",
  "title": "Projektbeispiel",
  "description": "Beschreibung des Projekts",
  "url": "https://example.com",
  "channel_id": 2,
  "project_identifizier_string": "abc123",
  "reach": "1000",
  "statistic_type": "internal",
  "external_sources_entity_id": "-1",
  "trackingtemplate": "template",
  "kpi_whitelist": "kpi1,kpi2",
  "hidden": false,
  "login_type": "pub",
  "exclude_from_salary": 0,
  "default_project": false,
  "insert_timestamp": "2025-01-27T12:00:00"
}
```

Fehlerhafte Anfrage

```
{  
  "error": "Missing required field: title"  
}
```

Hinweise zur Fehlerbehandlung

- Bei fehlenden Pflichtfeldern (`title`, `status`, `url`) wird ein entsprechender Fehler zurückgegeben.
- Ungültige oder nicht vorhandene `id` bei GET, PUT oder DELETE resultieren in einer leeren Antwort oder einer Fehlermeldung.

Publisher-Verwaltung über die V6 REST-API

Einleitung

Die V6 REST-API ermöglicht es Admins, Publisher anzulegen und deren Details auszulesen. Die Funktionalität beschränkt sich aktuell auf **CREATE** und **READ**-Operationen.

Endpoint

Für Admins: `https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher`

CRUD-Operationen

POST: Publisher erstellen (CREATE)

Erstellt einen neuen Publisher und fügt die entsprechenden Daten in die Tabellen `global.logins` und `global.login_settings` ein.

Erforderliche Parameter:

Parameter	Typ	Beschreibung
<code>email</code>	String	Pflichtfeld, max. 255 Zeichen
<code>salutation</code>	String	Pflichtfeld, Werte: <code>mr</code> oder <code>mrs</code>
<code>prename</code>	String	Pflichtfeld, max. 255 Zeichen
<code>surname</code>	String	Pflichtfeld, max. 255 Zeichen

Optionale Parameter (Auszug): `company`, `street`, `zip`, `city`, `telephone`, `country` (ISO 3166 ALPHA-3), `billing_sepa_iban`, `companytype` (`un` oder `priv`), `language_interface` (Default: `DEU`), etc.

Beispiel-Request:

```
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 '{
  "email": "max.mustermann@example.com",
  "salutation": "mr",
  "prename": "Max",
  "surname": "Mustermann",
  "company": "Musterfirma",
  "street": "Musterstraße 1",
  "zip": "12345",
  "city": "Musterstadt",
  "country": "DEU",
  "billing_sepa_owner": "Max Mustermann",
  "billing_sepa_iban": "DE12345678901234567890",
  "billing_sepa_bic": "GENODEF1M01",
  "language_interface": "DEU",
  "billing_mode": 1,
  "billing_media": 1,
  "country_billing": "DEU",
  "country_publisher": "DEU",
  "billing_limit": 50
}' "https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher"
```

Erwartete Antwort:

```
{
  "id": 123,
  "name": "Max Mustermann",
  "email": "max.mustermann@example.com"
}
```

GET: Publisher auslesen (READ)

Gibt die Details eines spezifischen Publishers oder eine Liste von Publishern zurück.

Filteroptionen: Alle Felder der Tabelle können als Filter verwendet werden (z. B. `email`, `status`, `prename`, `surname`).

Beispiel-Request (spezifische ID):

```
curl -X GET -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"
"https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher?id=123"
```

Beispiel für GET mit Filtern:

```
curl -X GET -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"
"https://SUBDOMAIN.de/ws/V6/admin/JSON/Publisher?status=1&prename=Max"
```

Fehlerbehandlung

- **Fehlende Pflichtfelder (POST):**

```
{ "error": "Missing required field: email" }
```

- **Publisher nicht gefunden (GET):**

```
{ "error": "Publisher not found" }
```

Externe API anbinden

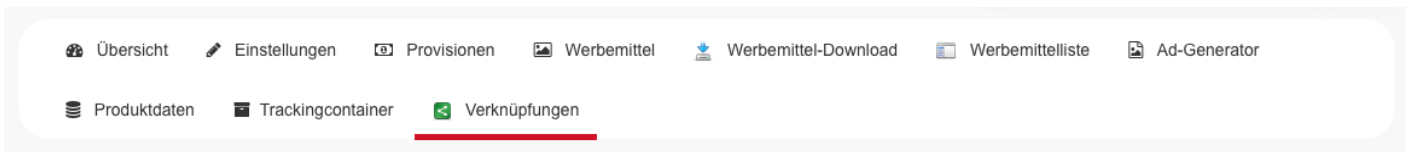
Es ist möglich für jedes externe Programm eine API für das System einzubinden. Die APIs werden durch den Support freigeschaltet. Hierfür muss bitte eine E-Mail an die Support@easy-m.de geschrieben werden. Anschließend wird überprüft, ob die API zur Verfügung steht oder angebunden werden kann.

Vorbereitung

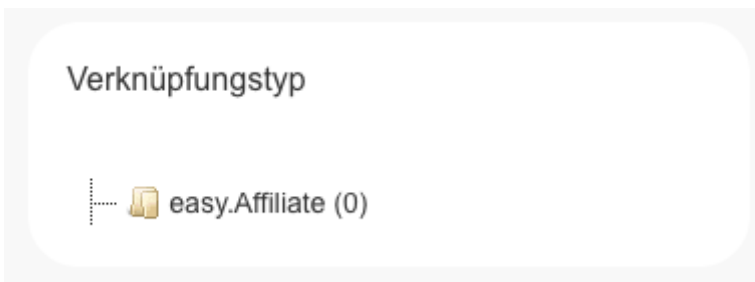
Um eine API für die Kampagne zu hinterlegen, muss zuerst die Kampagne angelegt werden.

Durchführung

Um eine API für eine Kampagne zu hinterlegen, muss auf das Kampagnen-Kontrollzentrum gewechselt werden. Anschließend muss zu dem Punkt "Verknüpfungen" navigiert werden:



Dort werden alle für die Kampagne verfügbaren APIs angezeigt:



Abschließend muss auf den Button "Neue Verknüpfung" geklickt werden, um die Daten, welche für die API benötigt werden, einzugeben.

Externe API Zugänge

Um die API des externen Dienstleisters anzusprechen werden entsprechende Zugangsdaten benötigt. Wo diese Daten in den einzelnen Systemen des Dienstleisters zu finden sind, wird in den einzelnen Anleitungen erläutert.

Werbemittel API

Admedia API (V6) - POST (Create)

Einleitung

Um Werbemittel nicht manuell, sondern durch einen automatisierten Prozess im easy.affiliate-System zu erstellen, stellt die easy Marketing GmbH eine Webservice-API bereit. Hierfür steht jedem User ein Authentifizierungs-Token und eine Login_id zur Verfügung. Der Authentifizierungs-Token sowie die Login_id werden dem User über das Frontend bereitgestellt.

User-ID: Wo ist die User-ID hinterlegt?

Access-Token: Wo ist der Access-Token hinterlegt?

Endpoint für Admin:

<https://SUBDOMAIN.de/ws/V6/admin/JSON/Admedia>

Endpoint für Advertiser:

<https://SUBDOMAIN.de/ws/V6/advertiser/JSON/Admedia>

Beispiel mit 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 "$EXAMPLE_JSON_DATA" <https://DOMAIN/ws/V6/admin/JSON/Admedia>
```

Codebeispiel

Mit der folgenden Methode können Werbemittel erstellt werden. Ein Beispielaufruf sieht folgendermaßen aus:

Headers:

```
Content-Type: application/json
X-Network-ID: -1
X-Auth-Token: ADMIN_APIUSER_TOKEN
X-Auth-ID: ADMIN_APIUSER_LOGIN_ID
```

Body:

```
{
  "campaign_id": 1,
  "project_ids": [50012],
  "title": "Werbemittel Titel",
  "deeplink": "https://example.de/link",
  "category_id": 5,
  "mediatype": "text",
  "linktext": "Klicken Sie hier",
  "status": 1
}
```

Variablenerläuterung

Headers

Variable	Bedeutung	Datentyp
Content-Type	Der Content-Type des Requests	String
X-NETWORKID	ID des Mandanten oder "-1"	Integer
X-AUTH-TOKEN	API-Authentifizierungs-Token des Admin Nutzers	String
X-AUTH-ID	ID des Admin Nutzers	Integer

Body

Parameter	Erklärung	Datentyp	Pflichtfeld
campaign_id	ID der Kampagne	Integer	Ja
project_ids	Whitelist von Projekt-IDs	JSON-ARRAY	Nein
title	Titel des Werbemittels	String	Ja

Parameter	Erklärung	Datentyp	Pflichtfeld
deeplink	Deeplink des Werbemittels	String	Ja
category_id	ID der Kategorie	Integer	Ja
click_tracking_getparameter	Click Tracking Parameter	String	Nein
valid_from	Gültig ab Datum	String	Nein
valid_until	Gültig bis Datum	String	Nein
mediatype	Typ des Werbemittels (z.B. text, banner)	String	Ja
linktext	Linktext (nur bei mediatype = text)	String	Ja, wenn Typ "text"
imageurl	Bild-URL (nur bei mediatype = banner)	String	Ja, wenn Typ "banner"
size_width	Breite des Banners	String	Ja, wenn Typ "banner"
size_height	Höhe des Banners	String	Ja, wenn Typ "banner"
status	Status des Werbemittels	Integer	Ja

Response

Nach erfolgreicher Erstellung gibt die API folgende Informationen als JSON zurück:

- Werbemittel-ID: Integer
- Werbemittel-Titel: String
- Werbemittel-Kategorie: String
- View-Link: String
- Click-Link: String
- View-Link (Shortener): String
- Click-Link (Shortener): String
- Adcode: String

Admedia API (V6) - GET (Read)

Abfrage generierter Werbemittel über die V6 REST-API

Um eine einfache und effiziente Verwaltung der generierten Werbemittel zu gewährleisten, stellt die V6 REST-API eine Abfragefunktion zur Verfügung. Mit dieser Funktion können Nutzer relevante Daten zu den Werbemitteln und den zugehörigen Trackinglinks abrufen.

Endpoint

Für Admins: `<https://SUBDOMAIN.de/ws/V6/admin/JSON/Admedia>`

Erforderliche Parameter:

- `campaign_id`: ID der betreffenden Kampagne (Pflichtfeld)
- `project_id`: ID des betreffenden Projekts (Optional)

Beispielaufruf mit cURL

```
curl -X GET -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" <https://DOMAIN/ws/V6/admin/JSON/Admedia?params[campaign_id]=1&params[project_id]=50001>
```

Erwartete Antwort

Nach erfolgreichem Aufruf gibt die API die folgenden Informationen zu den Werbemitteln im JSON-Format zurück:

- **id**: ID des Werbemittels (Integer)
- **title**: Titel des Werbemittels (String)
- **campaign_id**: ID der zugeordneten Kampagne (Integer)
- **project_id**: ID des zugeordneten Projekts (Integer, falls angegeben)
- **clicklink**: Clicklink des Werbemittels (String)

- **viewlink:** Viewlink des Werbemittels (String)
- **adcode:** Javascriptcode/HTML-Code des Trackinglinks (String)

Fehlerbehandlung

Bei fehlendem oder ungültigem `campaign_id` wird ein entsprechender Fehler zurückgegeben.

Bei nicht vorhandenen Daten für die angegebenen Parameter wird ein leerer Datensatz zurückgegeben.

Basket API

Basket Update API

? Einleitung

Zur automatischen Bearbeitung von Warenkörben, bieten wir eine Webservice-API an. Diese ermöglicht die nachträgliche Anpassung bereits existierender Warenkörbe. Jeder Nutzer erhält für den Zugriff einen Access-Token und eine User-ID, die über das Frontend bereitgestellt werden.

? Authentifizierung

Für den Zugriff sind folgende Header erforderlich:

Header	Beschreibung	Typ
Content-Type	Der Content-Type des Requests	String
X-Network-ID	Netzwerk-ID: meist <code>-1</code> für mandantenübergreifende API	Integer
X-Auth-Token	Dein Access-Token (im Frontend sichtbar)	String
X-Auth-ID	Deine User-ID (im Frontend sichtbar)	Integer

? Endpunkte

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

? Beispiel (cURL)

```
curl -X PUT -H "Content-Type: application/json" -H "X-Network-ID: -1" -H "X-Auth-Token: AUTH-TOKEN" -H "X-Auth-ID: USER-ID" -d '[{"basket_item_id": 76041, "amount": 2, "status": "0", "additional": {"variable1": "text", "variable2": 1}}]'
```

? Body-Parameter

Parameter	Beschreibung	Datentyp	Pflichtfeld
basket_item_id	Eindeutige ID des Warenkorbelements	Integer	<input type="checkbox"/> Ja
amount	Anzahl der Warenkorbelemente	Integer	Nein
price	Nettopreis des Warenkorbelements	Float	<input type="checkbox"/> Nein
status	Status der Order: <input type="checkbox"/> = offen, <input type="checkbox"/> = bestätigt, <input type="checkbox"/> = storniert	Integer	<input type="checkbox"/> Ja
turnover	Bestellwert in Kampagnenwährung	Float	<input type="checkbox"/> Ja
additional	Zusätzliche Informationen zum Basket-Item	JSON	Nein

Beispiel Request

Headers

```
Content-Type: application/json
X-Network-ID: -1
X-Auth-Token: ACCESS-TOKEN
X-Auth-ID: USER-ID
```

Body

```
[
  {
    "basket_item_id": 76041,
    "amount": 1,
    "status": "1",
    "additional": {
      "variable1": "text",
      "variable2": 1
    }
  }
]
```

