Baskettracking (Clientsided)

- Einführung
  - Überblick
- Technische Grundlage
  - Funktionsweise
- Implementierung
  - Einführung
  - Script einbinden
  - Umgang mit Additional Parameters auf Transaktionsebene

Einführung

Überblick

Der Trackingcode dient der Erfassung des Warenkorbs und muss im `<body>`-Tag der Bestellabschlussseite des Shops eingebaut werden.

Die Variablen der Funktion `eamTrckAddBasketItem()` müssen mit den entsprechenden Werten befüllt werden.

Warum Baskettracking?

- **Präzisere Attribution:** Jeder Artikel im Warenkorb kann einzeln übergeben werden.
- **Flexibilität:** Neben den Pflichtfeldern können zusätzliche Parameter (z. B. Gutscheincode) übergeben werden.

Die Rolle der EMID

Für die **Attribution** ist die sogenannte **EMID** (easy.MARKETING-ID) entscheidend.

- Sie wird beim Klick über den Click-Link an die Landingpage übergeben.
- Der Advertiser-Server muss diese ID beim serverside-Call wieder an das System zurückspielen.
- Ohne EMID ist keine vollständige Customer Journey nachvollziehbar.

Technische Grundlage

Funktionsweise

Das Baskettracking basiert auf einem **JavaScript-Aufruf** im Shop.

- Beim Aufruf der Bestellabschlussseite wird das Tracking-Skript geladen.
- Für jedes Produkt im Warenkorb wird die Funktion `eamTrckAddBasketItem()` mit den entsprechenden Parametern aufgerufen (z. B. Preis, Anzahl, Produktname).
- Anschließend wird mit `eamTrckSubmitBasket()` der gesamte Warenkorb an das Tracking-System von *easy.MARKETING* übermittelt.

Rolle der EMID

- Die **EMID** wird beim Klick auf ein Werbemittel erzeugt und an den Shop übergeben.
- Sie muss im Tracking-Code eingebunden werden, um die Transaktion korrekt zu attribuieren.
- Ohne EMID kann die **Customer Journey nicht korrekt zugeordnet** werden.

Implementierung

Einführung

Client-seitiges Baskettracking

Das client-seitige Baskettracking ermöglicht die Übergabe von Warenkorbdaten direkt aus dem Shop-Frontend an das Tracking-System von *easy.MARKETING*. Dabei werden die einzelnen Artikel per JavaScript an das Skript übergeben und anschließend gesammelt übertragen.

Funktionsweise (Client)

- **Skript-Einbindung:** Beim Aufruf der Bestellabschlussseite wird das Tracking-Skript eingebunden.
- **Artikel-Übergabe:** Für jedes Produkt im Warenkorb wird `eamTrckAddBasketItem(...)` aufgerufen.
- **Abschluss:** Mit `eamTrckSubmitBasket()` werden alle gesammelten Artikel an das Tracking-System übermittelt.
- **Transaktionserstellung:** Jeder Basketaufruf erzeugt automatisch auch eine Transaktion im System.
  - Zu jedem Basket gehört die entsprechende Transaktion
- **Attribuierung:** Die **EMID** wird beim Klick auf ein Werbemittel erzeugt und muss unbedingt mit übergeben werden, um die Customer Journey korrekt zu attribuieren.

Verarbeitung von Zusatzdaten (Additional Parameters)

Zusätzlich zu den Standardparametern können pro Artikel optionale Informationen in einem Objekt (Key-Value-Pairs) übergeben werden. Hierbei gilt folgende Logik für die Speicherung:

1. **Artikelebene:** Die Parameter werden individuell für jeden einzelnen Artikel gespeichert.
2. **Transaktionsebene:** Das System übernimmt die Zusatzdaten des **zuletzt aufgerufenen Artikels** (`eamTrckAddBasketItem`) und speichert diese zusätzlich auf Transaktionsebene ab.

> **Wichtiger Hinweis:** Wenn bestimmte Informationen (z. B. Versandart oder Gutscheincode) für die Auswertung auf Transaktionsebene relevant sind, müssen diese zwingend bei **jedem** Artikelaufruf identisch mitgegeben werden. Nur so ist sichergestellt, dass sie unabhängig von der Artikelreihenfolge korrekt auf Transaktionsebene erfasst werden.

Script einbinden

Das Skript für die Einbindung der Bibliothek, aus denen die Functions für die Erfassung des Warenkorbs geladen werden, muss auf der **Bestellabschlussseite** im `<body>` eingebunden werden:

```
<script src="https://SUB-DOMAIN/trck/ebasket/ebasket.js"></script>
```

> **Hinweis:** `SUB-DOMAIN` durch die eigene Subdomain ersetzen.

Basket-Items übergeben & absenden

Für **jeden** Artikel im Warenkorb wird `eamTrckAddBasketItem()` aufgerufen.
Am Ende wird **einmal** `eamTrckSubmitBasket()` ausgeführt.

```
<script>
  /* Artikel ins Tracking übergeben */
  eamTrckAddBasketItem(
    'CAMPAIGN_ID',
    'TOKEN',
    'TRIGGER_ID',
    'ARTICLE_NUMBER',
    'AMOUNT',
    'PRICE',
    'EMID',
    'PRODUCTNAME',
    'CATEGORY',
    {
      'variable'   : 'VARIABLENINHALT',
      'variable_2' : 'VARIABLENINHALT_2'
    }
  );
```

```
  /* Gesamten Warenkorb absenden */
  eamTrckSubmitBasket();
</script>
```

Vollständiges Beispiel (2 Artikel)

```
<script src="https://pvn.beispiel.de/trck/ebasket/ebasket.js"></script>

<script>
  eamTrckAddBasketItem(
    '1',
    '12345',
    '1',
    '002973000902',
    '1',
    '335.29',
    '5eeb0e0dabf7ad282b28c4a2',
    'Samsung Fernseher',
    'Unterhaltungselektronik',
    { 'Farbe': 'Hellgrau' }
  );

  eamTrckAddBasketItem(
    '1',
    '12345',
    '1',
    '000657001803',
    '1',
    '25.13',
    '5eeb0e0dabf7ad282b28c4a2',
    'TV-WANDHALTER',
    'Wohnzimmer',
    { 'Farbe': 'Schwarz', 'Unterkategorie': 'TV Halterung' }
  );

  eamTrckSubmitBasket();
</script>
```

Vollständiges Beispiel mit Abzug von Gutschein (3 Artikel)

```
<script src="https://pvn.beispiel.de/trck/ebasket/ebasket.js"></script>

<script>
  eamTrckAddBasketItem(
    '1',
    '12345',
    '1',
    '002973000902',
    '1',
    '335.29',
    '5eeb0e0dabf7ad282b28c4a2',
    'Samsung Fernseher',
    'Unterhaltungselektronik',
    { 'vc': 'Gutscheincode12345', 'Farbe': 'Hellgrau' }
  );

  eamTrckAddBasketItem(
    '1',
    '12345',
    '1',
    '000657001803',
    '1',
    '25.13',
    '5eeb0e0dabf7ad282b28c4a2',
    'TV-WANDHALTER',
    'Wohnzimmer',
    { 'vc': 'Gutscheincode12345', 'Farbe': 'Schwarz', 'Unterkategorie': 'TV Halterung' }
  );

  eamTrckAddBasketItem(
    '1',
    '12345',
    '1',
    '000657001809',
    '1',
```

```
    '-10.00',
    '5eeb0e0dabf7ad282b28c4a2',
    'Gutscheincode12345',
    'Voucher',
    { 'vc': 'Gutscheincode12345' }
  );

  eamTrckSubmitBasket();
</script>
```

Variablenbeschreibung

Parameter	Beschreibung
CAMPAIGN_ID	ID der Kampagne, z. B. `'1'`.
TOKEN	Eindeutige Bestellnummer / Order ID.
TRIGGER_ID	ID des Triggers, z. B. `'1'`.
ARTICLE_NUMBER	Artikelnummer / SKU.
AMOUNT	Menge als String, z. B. `'1'`.
PRICE	Netto-Preis pro Stück, mit **Punkt** als Dezimaltrennzeichen, z. B. `'25.13'`.
EMID	Eindeutige Tracking-ID aus dem Werbemittelklick.
PRODUCTNAME	Produktname als String.
CATEGORY	Produktkategorie, z. B. `'Unterhaltungselektronik'`.
Additional	Optionale Parameter als Key-Value-Paare, z. B. `{ 'vc': 'Gutschein12345' }`.

Additional Parameter

Weitere Parameter können flexibel als **Key-Value-Strings** übergeben werden:

```
{ 'vc': 'Gutschein12345', 'Brand': 'Samsung', 'Farbe': 'Hellgrau' }
```

Hinweise zur Platzierung & Validierung

- **Seite:** Bestellabschlussseite / Thank-You-Page.
- **Position:** Innerhalb des `<body>`.
- **Reihenfolge:** Erst alle `eamTrckAddBasketItem(...)` Aufrufe, dann **einmal** `eamTrckSubmitBasket()`.
- **Preise:** Netto, mit Punkt als Dezimaltrennzeichen.
- **EMID:** Muss zwingend übergeben werden, sonst keine Attribution.
- **Mehrere Artikel:** Pro Artikel ein eigener `eamTrckAddBasketItem()` Aufruf.
- **Debug-Tipp:** In der Browser-Konsole prüfen, ob Fehler beim Laden des Skripts oder beim Funktionsaufruf auftreten.

Umgang mit Additional Parameters auf Transaktionsebene

Diese Dokumentation erläutert die Logik der Datenverarbeitung von Zusatzparametern (`Additional Parameters`) und deren Auswirkungen auf das Tracking und externe Systeme (z. B. Tagmanager).

Verarbeitungslogik

Obwohl die Funktion `eamTrckAddBasketItem()` Parameter für das einzelne Produkt entgegennimmt, werden diese Werte in unserem System auch an den **Gesamtdatensatz der Transaktion** (Order-Ebene) geschrieben.

Da eine Transaktion aus mehreren Artikeln bestehen kann, gilt bei der Verarbeitung folgende Regel:

> **Der letzte Wert gewinnt.** Wenn unterschiedliche Artikel denselben Parameter-Key (z. B. `vc` oder `shipping_type`) mit unterschiedlichen Werten übergeben, wird der Wert des **letzten** `eamTrckAddBasketItem()`-Aufrufs als finaler Wert für die gesamte Transaktion gespeichert.

Wichtige Richtlinie für die Implementierung

Damit Daten auf Transaktionsebene (z. B. für die Nutzung als Trigger-Bedingung im Tag Manager) zuverlässig zur Verfügung stehen, muss folgendes beachtet werden:

Parameter, die für die gesamte Bestellung gelten, müssen in jedem Basket-Item mit identischem Wert übergeben werden.

Warum ist das wichtig?

1. **Tag Manager Bedingungen:** Viele Tag Manager nutzen Variablen der Transaktionsebene, um zu entscheiden, ob ein Pixel gefeuert werden soll. Ist der Wert inkonsistent oder fehlt er im letzten Artikel, schlägt die Bedingung fehl.
2. **Datenkonsistenz:** In Exporten und Reports wird der Parameter der Transaktion zugeordnet. Unterschiedliche Werte innerhalb einer Bestellung führen hier zu Zufallsergebnissen (basierend auf der Sortierung der Artikel im Code).

Beispiel: Richtige vs. Falsche Implementierung

❌ Falsch: Inkonsistente Werte

Hier würde der Wert für `customer_group` auf Transaktionsebene mit "Bestandskunde" überschrieben werden, obwohl der erste Artikel "Neukunde" meldet.

```
// 1. Artikel
eamTrckAddBasketItem(..., { 'customer_group': 'Neukunde' });

// 2. Artikel
eamTrckAddBasketItem(..., { 'customer_group': 'Bestandskunde' });

// Ergebnis Transaktionsebene: 'Bestandskunde'
```

✅ Richtig: Konsistente Werte

Soll ein Parameter sicher für die gesamte Transaktion gewertet werden, muss er bei jedem Aufruf identisch sein.

```
// 1. Artikel
eamTrckAddBasketItem(..., {
```

```
  'vc': 'SOMMER24',
  'shipping_method': 'Express',
  'color': 'Rot'
});

// 2. Artikel
eamTrckAddBasketItem(..., {
  'vc': 'SOMMER24',
  'shipping_method': 'Express',
  'color': 'Blau'
});

// Ergebnis Transaktionsebene: 'SOMMER24' und 'Express' sind korrekt gesetzt.
```

Zusammenfassung für Entwickler

1. **Globale Parameter** (Gutscheincodes, Kundengruppen, Versandarten) immer an **alle** Artikel der Bestellung hängen.
2. Sicherstellen, dass die Werte innerhalb eines Warenkorbs für diese globalen Keys **identisch** sind.
3. Nur **produktspezifische Parameter** (Größe, Farbe) dürfen variieren, wobei zu beachten ist, dass im Transaktions-Header nur der Wert des letzten Artikels sichtbar sein wird.