

**Cheatsheet**

**Transaktionen**

**v1.3**

© 2017 XQueue GmbH. Alle Rechte vorbehalten.

Dokumentation für das XQueue System

Diese Dokumentation darf ohne vorherige schriftliche Genehmigung durch die XQueue GmbH weder teilweise noch ganz reproduziert, in Datenbanken gespeichert oder in irgendeiner Form übertragen werden. Der Inhalt dieser Dokumentation dient ausschließlich zu Informationszwecken, kann jederzeit geändert werden und stellt keine Verpflichtung seitens der XQueue GmbH dar. Für Fehler der in dieser Dokumentation enthaltenen Informationen wird keine Haftung übernommen.

XQueue GmbH, Christian-Pleß-Str. 11-13, 63069 Offenbach am Main

<b>Inhalt und Ziel</b>	Dieses Dokument ist als Crashkurs und Schnellnachschlagewerk für die Verwendung von Transaktionen zu verstehen.
<b>Typ</b>	Cheatsheet
<b>Version</b>	1.3
<b>Autor</b>	Marcus Beckerle
<b>Erstellt</b>	24.01.2017
<b>Letzte Änderung</b>	28.11.2017

# Was ist für eine Transaktion notwendig?

## Einführung

---

Als Transaktionsmail wird eine E-Mail bezeichnet, die auf Grund eines Ereignisses, das von einem externen System erzeugt wird, ausgelöst wird. Als Beispiele sind etwa Bestellbestätigungsmails, Passwortänderungsmails und andere Mails zu sehen, die möglichst in Echtzeit an einzelne Kontakte zugestellt werden müssen.

## Elemente

---

Damit die Eindeutigkeit der Begriffe für verschiedene Elemente gegeben ist, wird im Folgenden auf die Definitionen eingegangen welche in diesem Dokument verwendet werden.

- REST API<sup>1</sup>: REST steht für REpresentational State Transfer und stellt eine Möglichkeit dar, programmatisch Daten mit Maileon auszutauschen.
- Transaktion: Eine Transaktion wird im UI auch als Kontaktereignis bezeichnet und stellt einen Datensatz dar, der von einem externen System in Form eines (JSON) Datensatzes über die REST API<sup>2</sup> an Maileon gemeldet wird.
- Transaktionstyp: Zur Strukturierung von Transaktionen muss für jede Art einer Transaktion (etwa einer Bestellbestätigung) ein Transaktionstyp hinterlegt werden. Dieser enthält die Namen aller verfügbaren Attribute und deren Typ (Zeichenkette, ganze Zahl, Datum, ...). Somit ist es möglich, im UI auf die Attribute zuzugreifen und diese in ein Newsletter-Template einzufügen. Weiterhin kann man hier vorgeben, ob ein Attribut ein Pflichtfeld ist oder ob die Transaktion auch ohne das Attribut akzeptiert wird.
- Transaktionsmail: Die Transaktionsmail ist eine besondere Art der Triggermails. Für eine Transaktionsmail muss als Auslöser „Versand bei Kontakteinzelereignis“ angegeben werden. Hier wird nun spezifiziert bei welchem Transaktionstypen die Mail versendet werden soll. Wird nun eine Transaktion vom entsprechenden Typen an Maileon gesendet, so wird die Transaktionsmail versendet.

## Skizzierung des Ablaufs

---

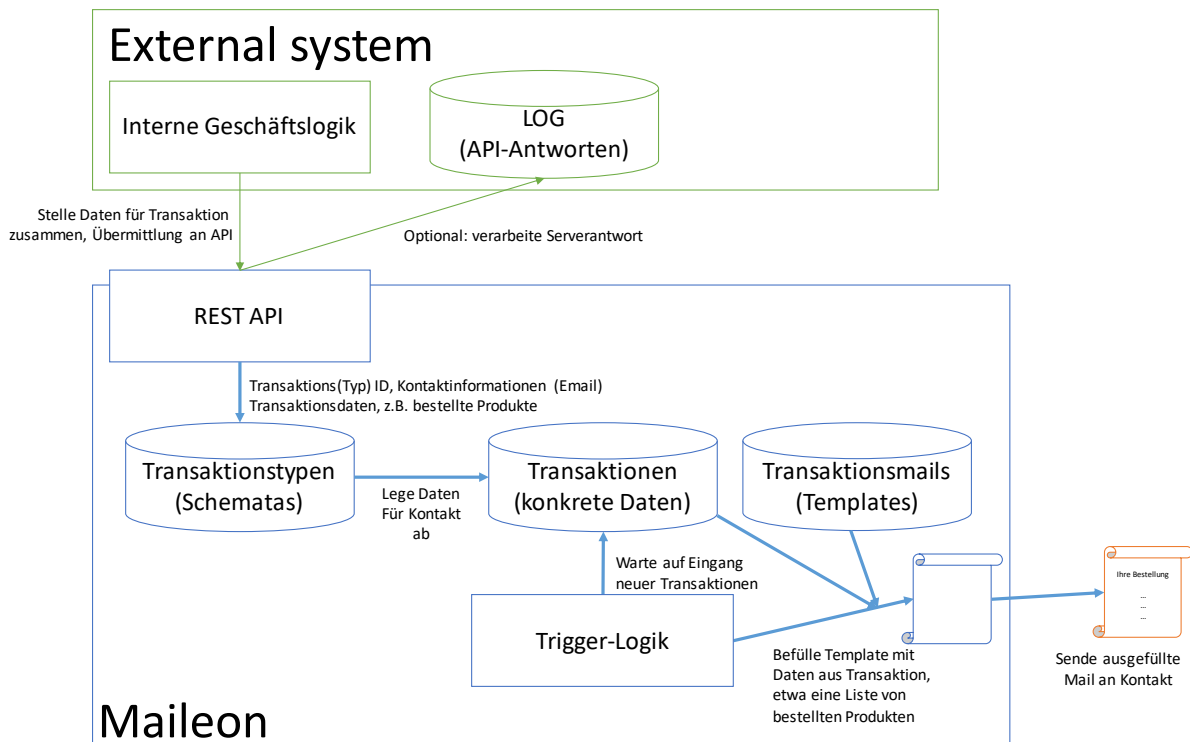
1. Zunächst sammelt das externe System alle notwendigen Daten und sendet dann eine Anfrage an die REST-API von Maileon die folgende Informationen enthält:
  - die ID des Transaktionstypen
  - ein Identifikator für den Kontakt zu dem die Transaktion gehört, also die Emailadresse, die externe ID oder die interne Maileon ID
  - die Daten, die mit der Transaktion übermittelt werden sollen

---

<sup>1</sup> <http://dev.maileon.com/>

<sup>2</sup> <http://dev.maileon.com/api/rest-api-1-0/transactions/create-transactions/>

2. Maileon vergleicht die Daten mit dem angegebenen Transaktionstypen und ermittelt, ob der Kontakt überhaupt beschickt werden darf (Kontakt existiert und hat eine valide Permission). Wenn alles OK ist, wird die Transaktion gespeichert.
3. Eine Komponente überwacht nun eingehende Transaktionen und wenn es eine Transaktionsmail gibt, das auf die Transaktion wartet, wird das Template mit den Daten aus der Transaktion befüllt und an den Kontakt versendet.



## Beispiele

Auf der Seite zum Erstellen von Transaktionen<sup>3</sup> (ganz unten) befinden sich zwei Beispiele, die zeigen, wie man Transaktionen in Maileon einsetzt.

### Simple Transaction<sup>4</sup>:

In diesem Beispiel kann eine leere Transaktion erzeugt werden, das heißt eine Transaktion OHNE irgendwelche Daten. Dazu muss im jeweiligen Maileon Account unter „Listen und Kontakte“ ein Ereignistyp erzeugt werden. Der Name ist frei wählbar. Nach dem Speichern kann die ID des Transaktionstypen angezeigt werden, indem auf den neu erstellten Typen geklickt wird. Die ID steht hinter der Markierung „API“ (Abbildung 1).

<sup>3</sup> <http://dev.maileon.com/api/rest-api-1-0/transactions/create-transactions/>

<sup>4</sup> [https://hosting.maileon.com/service/examples/en/transaction\\_example\\_simple/index.php](https://hosting.maileon.com/service/examples/en/transaction_example_simple/index.php)

**Kontakt ereignis-Erstellung**  
Erstellen Sie mittels dieses Formulars ein neues Kontakt ereignis.

Kontakt ereignis: **orderExample**  
[+ Kontakt ereignis-Spezifikation exportieren](#)

API-ID: **API 249**

Vorhaltefrist: *nicht definiert*

**Eigenschaften**  
*Folgend sind die verfügbaren Eigenschaften des Kontakt ereignisses definiert.*

Bezeichnung	Beschreibung	Pflicht	Format
gender			Text
title			Text
first_name		✓	Text
last_name		✓	Text
phone			Text
campaign			Text
webshop_id		✓	Text
language		✓	Text
customer_no		✓	Text
order_no		✓	Text
currency		✓	Text
billing_address.name		✓	Text

**Abbildung 1: ID des Transaktionstypen**

In dem Formular muss nun der API-Key des Accounts, sowie der Transaktionstyp und ein valider Kontakt eingegeben werden. Nach Abschicken sollten die Daten auf der Transaktionstyp-Übersichtsseite zu sehen sein (Abbildung 2). Wenn es ein Mailing gibt, das den Transaktionstypen als Auslöser hat, so wird dieses umgehend versendet.

**Kontakt ereignisse**  
Administrieren Sie in diesem Bereich die verfügbaren Kontakt ereignisse. Änderungen an gespeicherten Kontakt ereignissen sind aufgrund der Vermeidung späterer Inkonsistenzen nicht möglich.

5 Kontakt ereignisse verfügbar

Kontakt ereignis	Beschreibung	Ereignisdaten	Eigenschaften
<input type="checkbox"/> cart_reminder		10	115
<input type="checkbox"/> order		0	107
<input type="checkbox"/> orderExample		0	26
<input type="checkbox"/> TextHTMLContent		0	1
<input type="checkbox"/> UrlAndTextTest		0	2

**Kontakt ereignis-Historie von Kontakt ereignis "orderExample"**  
Die folgende Auflistung beinhaltet die folgenden Ereignisse des selektierten Kontakt ereignisses.

Datum	Kontakt	Datum	Datum
24.01.2017 14:01:28	maxschmidt@max.de	first_name	Max
		last_name	Mustermann
		phone	0621
		email	max@max.de
		webshop_id	My Webshop 0
		language	de
		customer_no	123456789
		order_no	000000000
		currency	€
		billing_address.name	Max Mustermann
		billing_address.street	Maximilianstr. 1
		billing_address.zip	80539 München
		billing_address.city	München
		billing_address.country	Deutschland
		billing_address.street2	Maximilianstr. 2
		billing_address.zip2	80539 München
		billing_address.city2	München
		billing_address.country2	Deutschland
		order_date	2017-01-24
		total	123.45
		payment_type	Credit Card

**Abbildung 2: Transaktionsdaten**

## Complex Transaction (Order Confirmation)<sup>5</sup>

In diesem Beispiel wird eine Transaktion an Maileon geschickt, die einer Bestellbestätigung nachempfunden wurde. Da dies ein komplexerer Typ ist, sieht der Code vor, dass der Transaktionstyp beim ersten Gebrauch via API angelegt wird. Aktiviert wird dies durch Anklicken der Checkbox aus Abbildung 3. Die ID des erstellten Typen wird angezeigt (Abbildung 4) und sollte notiert werden. Da der Name des Typen eindeutig sein muss, würde sonst eine Fehlermeldung angezeigt werden, wenn der Typ erneut angelegt werden soll. Keine Sorge jedoch: die ID kann man, wie beschrieben, auch in der Maileon-ID anzeigen lassen (Abbildung 5).

### Maileon - Transactions for Order-Confirmation-Mails

The form contains the following elements:

- Api-Key:** Input field with placeholder text: "- Bitte geben Sie hier Ihren API-Key ein -"
- Transaction-ID:** Input field with placeholder text: "- Please enter your Transaction-ID -"
- Checkbox:** "I do not have such a transaction type in my account, create a new one:
- Text:** "If you want a sample mail to be triggered, download the template [here](#). Upload the template to your account, create a mailing based on it and activate it to be sent on occurrence of the above configured event. Then a mailing like [this one](#) will be sent to you."
- Email:** Input field with placeholder text: "- Please enter the Email-Address the transaction should be sent to (must exist in Maileon)-". Below it, a note: "Please make sure the email address exists in your Maileon account and has a valid permission (= not NONE)."
- Show debug output:**
- Button:** "Send transaction..."

**Abbildung 3: Formular des erweiterten Beispiels**

## Maileon - Transactions - A more complex example

- Creating new transaction type named 'orderExample':

### cURL session log

#### Result

```
status code: 200 OK
is success: true
is client error: false

body data:
<transaction_type_id>250</transaction_type_id>

Body contains XML.
result type: integer
```

**success** (Status code: 200 - OK)

**The transaction-ID of the created transaction type is: 250, please note it for later use.**

**Abbildung 4: ID des neu erzeugten Transaktionstypen**

<sup>5</sup> [https://hosting.maileon.com/service/examples/en/transaction\\_example\\_complex/index.php](https://hosting.maileon.com/service/examples/en/transaction_example_complex/index.php)

Anschließend wird ein konkreter Datensatz erzeugt und an Maileon gesendet. Die Nutzdaten in diesem Fall bestehen aus einer Reihe von Feldern, die für eine Bestellung notwendig sind, z.B. Vorname, Nachname, Zahlungsmethode, Gesamtsumme etc. Die genauen Attributnamen können natürlich jederzeit im Transaktionstypen im Maileon UI eingesehen werden (Abbildung 5). Die konkreten Daten werden schließlich als JSON serialisiert und im Testclient vor dem Versand angezeigt.

### Kontaktereignis-Erstellung

Erstellen Sie mittels dieses Formulars ein neues Kontaktereignis.

Kontaktereignis: **orderExample**  
+ Kontaktereignis-Spezifikation exportieren

API-ID: API 250

Vorhaltefrist: *nicht definiert*

**Eigenschaften**  
*Folgend sind die verfügbaren Eigenschaften des Kontaktereignisses definiert.*

Bezeichnung	Beschreibung	Pflicht	Format
gender			Text
title			Text
<span style="border: 1px solid red; padding: 2px;">first_name</span>		✓	Text
<span style="border: 1px solid red; padding: 2px;">last_name</span>		✓	Text
phone			Text
campaign			Text
webshop_id		✓	Text
language		✓	Text
customer_no		✓	Text
order_no		✓	Text
currency		✓	Text
billing_address.name		✓	Text
billing_address.street1		✓	Text
billing_address.street2		✓	Text
billing_address.zip		✓	Text
billing_address.city		✓	Text
shipping_address.name		✓	Text
shipping_address.street1		✓	Text
shipping_address.street2		✓	Text
shipping_address.zip		✓	Text
shipping_address.city		✓	Text
items		✓	JSON
order_date		✓	Text
<span style="border: 1px solid red; padding: 2px;">total</span>		✓	Text
<span style="border: 1px solid red; padding: 2px;">payment_type</span>		✓	Text
voucher_code			Text

Fenster schließen

**Abbildung 5: Transaktionstyp für eine Bestellbestätigung**

Wichtig: nach Absenden der Transaktion erzeugt Maileon einen Report, in dem für jeden Kontakt einzeln aufgelistet ist, ob der Versand angenommen wurde oder nicht:

```
{"reports":[{"contact":{"id":365045,"email":"marcus.beckerle@xqueue.com"},"queued":true]}
```

Wenn eine Transaktion nicht erfolgreich war, wird eine entsprechende Meldung ausgegeben.

Beispiele:

- Fehlendes Pflichtfeld  

```
{"reports":[{"contact":{"id":365045,"email":"marcus.beckerle@xqueue.com"}, "queued":false, "message":"The required attributes [first_name, last_name] are missing."}]}
```
- Falsches Format (Date ungültig):  

```
{"reports":[{"contact":{"id":365045,"email":"marcus.beckerle@xqueue.com"}, "queued":false, "message":"The provided attribute 'order_date' cannot be transformed: timestamp expected."}]}
```

Im PHP Code werden die Daten einfach dem entsprechenden Objekt übergeben:

```
$transaction->content['first_name'] = "Max";  
$transaction->content['last_name'] = "Mustermann";  
$transaction->content['order_date'] = "11.09.2016";  
$transaction->content['total'] = "227,90";  
$transaction->content['payment_type'] = "Credit Card";
```

Den Code (PHP) beider Beispiele kann man auf dem jeweiligen Formular [simple<sup>6</sup>, complex<sup>7</sup>] unten rechts herunterladen. Die Beispiele sind mit weiteren Informationen auf der entsprechenden Seite in der Dokumentation<sup>8</sup> verlinkt.

Hinweis: Es können mehrere Transaktionen an verschiedene Kontakte auf einmal übergeben werden, es darf jedoch keine Transaktionen an denselben Kontakt im selben Aufruf enthalten sein.

Hinweis: Transaktionen werden immer als Array an die Schnittstelle übergeben, also in eckigen Klammern [{...Transaktionsdaten 1...},{... Transaktionsdaten 2...},{... Transaktionsdaten 2...}]. Wenn nur eine einzelne Transaktion gesendet werden soll muss diese dennoch als Array übergeben werden: [{...Transaktionsdaten...}]. Im PHP Code kann hierzu die Transaktion einfach als Array an den Transaktionsservice übergeben werden: array(\$transaction)

---

<sup>6</sup> [https://hosting.maileon.com/service/examples/en/transaction\\_example\\_simple/index.php](https://hosting.maileon.com/service/examples/en/transaction_example_simple/index.php)

<sup>7</sup> [https://hosting.maileon.com/service/examples/en/transaction\\_example\\_complex/index.php](https://hosting.maileon.com/service/examples/en/transaction_example_complex/index.php)

<sup>8</sup> [http://dev.maileon.com/api/rest-api-1-0/transactions/create-transactions/#Running\\_examples\\_with\\_code\\_download](http://dev.maileon.com/api/rest-api-1-0/transactions/create-transactions/#Running_examples_with_code_download)