

---

# EasyCouponExtension

Anwenderdokumentation

André Fischer

30.03.2026

## Inhaltsverzeichnis

<b>1 Was macht dieses Plugin?</b>	<b>2</b>
1.1 Diagramm: Ausloeser von aussen . . . . .	2
<b>2 Voraussetzungen kurz</b>	<b>2</b>
<b>3 Plugin-Konfiguration (technisch)</b>	<b>3</b>
<b>4 Webhook: Gutschein erzeugen und mailen</b>	<b>4</b>
4.1 URL . . . . .	4
4.2 Body / Datenformat . . . . .	4
4.3 Ablauf . . . . .	4
4.4 Antwort (Erfolg) . . . . .	5
<b>5 Webhook: Kundengruppe setzen</b>	<b>5</b>
5.1 URL . . . . .	5
5.2 Ergebnis . . . . .	5
<b>6 Test-Endpunkt</b>	<b>6</b>
<b>7 ActiveCampaign einrichten (grober Fahrplan)</b>	<b>6</b>
<b>8 Konsole (CLI)</b>	<b>6</b>
<b>9 Logging</b>	<b>7</b>
<b>10 Troubleshooting</b>	<b>7</b>

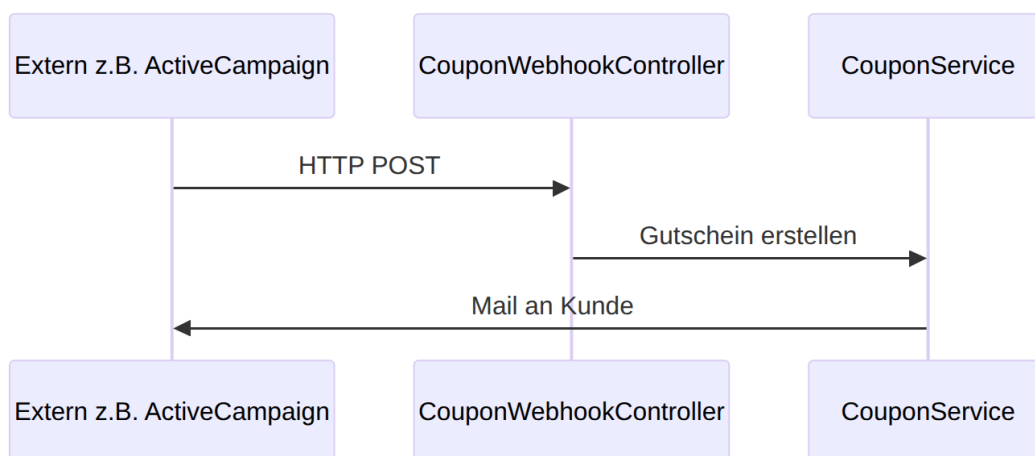
## 1 Was macht dieses Plugin?

Du nutzt **Net Inventors Easy Coupon** in Shopware und willst von **außen** (typischerweise **ActiveCampaign**) zwei Dinge anstoßen:

1. **Einen neuen Gutschein** aus einer **bestehenden Easy-Coupon-Vorlage** erzeugen und dem Kontakt per **E-Mail** mit dem Code zuschicken.
2. **Einem bestehenden Shopware-Kunden** anhand seiner **E-Mail-Adresse** eine **andere Kundengruppe** zuweisen.

Das Plugin selbst ist eine **dünne Schicht**: Es stellt **öffentliche API-Routen** bereit (ohne Shopware-Login), prüft optional einen **geheimen Schlüssel**, spricht mit dem **Easy-Coupon-Repository** und dem **Shopware-MailService**. Es gibt **keinen** eingebauten ActiveCampaign-Client – AC ruft nur per Webhook auf.

### 1.1 Diagramm: Ausloeser von aussen



**Pflicht:** Das Plugin **Neti EasyCoupon** muss installiert und aktiv sein, sonst fehlen die Repositories und Services.

**Shopware:** laut `composer.json 6.6.x (shopware/core ^6.6.0)`.

---

## 2 Voraussetzungen kurz

- **Easy Coupon** (Net Inventors) aktiv.

- Im Admin eine **Vorlage-Gutschein-ID** (UUID), von der Bedingungen, Rabatt, Wert usw. kopiert werden.
- **Mailversand** in Shopware funktioniert (Absender, SMTP/App, ...).
- Für Webhooks: **Security Key** gesetzt.

---

### 3 Plugin-Konfiguration (technisch)

Pfad: **Erweiterungen** → **Meine Erweiterungen** → **EasyCouponExtension** → **Konfigurieren**.

Feld	Interner Key	Rolle
Sicherheitsschlüssel	<b>securityKey</b>	Optional, aber für Produktion <b>dringend empfohlen</b> . Wenn gesetzt, muss jeder Webhook-Call den Query-Parameter <b>x-sec-key</b> mit <b>exakt diesem Wert</b> mitschicken.
ID des Verkaufskanals	<b>salesChannelId</b>	In der <code>config.xml</code> als <b>Pflichtfeld</b> geführt. Im <b>aktuellen PHP-Code</b> wird dieser Wert für den Mailversand nicht direkt aus der Config gelesen; die Sales-Channel-ID für E-Mails wird stattdessen aus dem <b>HTTP-Request</b> abgeleitet (Host/Domain, Request-Attribute) oder es greift ein fester Fallback. Du solltest die ID trotzdem sauber pflegen, damit die Konfiguration fachlich vollständig bleibt.

---

Feld	Interner Key	Rolle
Tage gültig	<b>couponDuration</b>	Wird genutzt, wenn die <b>Vorlage kein eigenes validUntil</b> hat: Ablaufdatum = heute + diese Tage.
Sendername der E-Mail	<b>emailSenderName</b>	Absendername für den Gutschein-Mailversand.
Sender-E-Mail	<b>senderEmail</b>	Absenderadresse für den Gutschein-Mailversand.

---

## 4 Webhook: Gutschein erzeugen und mailen

### 4.1 URL

```
1 POST https://<dein-shop>/api/v1/activecampaign/webhook/<couponId>?x-secret-key=<deinSecret>
```

- : **UUID der Easy-Coupon-Vorlage** in Shopware (der Datensatz, von dem kopiert wird – nicht der spätere Einmalcode).
- API-Version: üblich **v1**, muss zur Route `api/v{version}/...` passen.

### 4.2 Body / Datenformat

Der Controller liest die Daten mit `$request->request->all()` – das sind die **POST-Parameter** wie bei einem klassischen Formular. Struktur mit **E-Mail des Kontakts**:

- Erwarteter Aufbau: **contact** → **email** (also verschachtelte Keys wie bei ActiveCampaign-Webhooks, **wenn** dein Aufruf sie so in den Request-Parameter-Bag legt).

### 4.3 Ablauf

1. Security-Key prüfen (falls konfiguriert).
2. Vorlage per **couponId** laden – wenn nicht vorhanden → 404.

3. Neuen Gutschein anlegen: Das Plugin kopiert den konfigurierten Vorlage-Gutschein, erzeugt einen neuen Code und übernimmt dabei wichtige Eigenschaften wie die Zuordnung zu Verkaufskanälen.
4. **Deduplizierung:** Pro „Ereignis“ wird ein Hash im **Kommentar** des Gutscheins hinterlegt. Existiert bereits ein nicht gelöschter Gutschein mit demselben Hash, wird **kein zweiter** erzeugt; die API antwortet mit einer **Success-ähnlichen Meldung** und **HTTP 500**.
5. E-Mail mit Code versenden; Absender aus Config; Inhalt u. a. über Twig-Template oder Mail-Template (Mail tempated in den Kommentar des Kopier-Gutscheins).

#### 4.4 Antwort (Erfolg)

JSON u. a. mit **success: true** und einem Hinweis auf den Wert (**code-value** / Gutscheinwert – je nach Response im Code).

---

## 5 Webhook: Kundengruppe setzen

### 5.1 URL

```
1 POST https://<dein-shop>/api/v1/activecampaign/webhook/customergroup/<customerGroupId>?x-sec-key=<deinSecret>
```

- : UUID der **Kundengruppe** in Shopware.

### 5.2 Ergebnis

- 200: Zuweisung ok.
- 404: Kunde mit dieser E-Mail nicht gefunden.
- 403: falscher/fehlender Security Key.
- 400: E-Mail fehlt.
- 500: z. B. Kundengruppe existiert nicht oder Datenbankfehler.

**Hinweis:** Mehrfaches Zuweisen derselben Gruppe ist unkritisch; es wird einfach erneut **groupId** gesetzt.

## 6 Test-Endpunkt

```
1 /api/v1/test
```

(plus ggf. gleiche Host-/Prefix-Regeln wie oben)

Liefert u. a. die **ermittelte Sales-Channel-ID** nach derselben Logik wie beim Mailversand (Attribute, Context, Domain-Match, Fallback). Praktisch, um zu sehen, **welcher Kanal** für E-Mails aus einem Webhook-Request herausfällt – **ohne** Gutschein zu erzeugen.

---

## 7 ActiveCampaign einrichten (grober Fahrplan)

1. In AC eine **Automation** mit **Webhook** (o. ä.).
  2. URL wie oben inkl. **couponId** bzw. **customerGroupId** und **x-sec-key**.
  3. Testkontakt mit derselben E-Mail wie in Shopware auslösen.
- 

## 8 Konsole (CLI)

---

Befehl	Kurzbeschreibung
itk:coupon:create	Interaktiv: E-Mail, Vorlagen-ID, optional Mailversand. Beim aktuellen Stand lohnt sich nach Nutzung ein kurzer Test, ob der Mailversand wirklich im richtigen Sales Channel landet.
itk:coupon:list	Gutscheine auflisten.
itk:coupon:count	Anzahl.
itk:coupon:create-from-file	Codes aus Datei anlegen.

---

Hilfe: **php bin/console --help**

---

## 9 Logging

Datei:

```
var/log/itk_newsletter_coupons.log
```

Dort findest du Webhook-Eingänge, Sales-Channel-Auflösung, Fehler beim Anlegen des Gutscheins und beim Mailversand. Bei „es passiert nichts“ zuerst hier schauen.

---

## 10 Troubleshooting

---

Problem	Prüfen
403	<b>x-sec-key</b> exakt wie in der Config? Keine Leerzeichen?
400 / keine E-Mail	Kommen <b>contact</b> und <b>email</b> im Request an? Log + ggf. curl mit form-encoded testen.
404 beim Gutschein	<b>couponId</b> wirklich UUID der Vorlage in Easy Coupon?
404 Kundengruppe-Webhook	Kunde existiert in Shopware mit <b>gleicher</b> E-Mail?
Kein Mail	Mailer-Setup Shopware; Absender-Config Plugin; Log; Sales-Channel-Auflösung (/api/v1/test).
Doppelte Gutscheine	AC-Automation mehrfach ausgelöst? Dedup nur über Kommentar-Hash – bei geänderten Eingaben kann ein zweiter Code entstehen.

---