

# Installationsanleitung für PayUnity Magento

Diese Anleitung erklärt die Installation und Benutzung der PayUnity Extension für Magento.

**Release Datum:** Mon, 07 Aug 2023 14:01:06 +0200

**Version:** 6.0.55

wallee AG  
General-Guisan-Strasse 47  
CH-8400 Winterthur

E-Mail: [info@sellxed.com](mailto:info@sellxed.com)

Telefon:

CH: +41 (0)44 505 13 60

DE: +49 (0)40 2999 60117

UK: +44 (0)161 850 6890

US: +1 (0)205 557 5869

© copyright by wallee AG Mon, 07 Aug 2023 14:01:06 +0200

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung .....</b>	<b>6</b>
1.1	Ablauf der Installation .....	6
1.2	System Anforderungen .....	7
<b>2</b>	<b>Konfiguration .....</b>	<b>8</b>
2.1	Grundkonfiguration des Hauptmoduls .....	8
2.2	Konfiguration Zahlungsarten .....	8
2.3	Konfiguration Webhook .....	10
2.3.1	Konfiguration Webhook durch Händler .....	11
2.3.2	Konfiguration Webhook durch PayUnity .....	11
2.4	Aktivierung und Testing .....	12
<b>3</b>	<b>Modul Installation im Magento Shop .....</b>	<b>13</b>
3.1	Installation .....	13
3.1.1	Installation Magento 1 .....	13
3.1.2	Magento 2 .....	13
3.2	Updates und Upgrades .....	14
3.2.1	Update Checklist .....	14
3.2.2	Update Instruktionen .....	15
<b>4</b>	<b>Modul Konfiguration im Magento Shop .....</b>	<b>16</b>
4.1	Konfiguration des Hauptmoduls .....	16
4.1.1	Magento 1 .....	16
4.1.2	Magento 2 .....	16
4.2	Konfiguration der Zahlungsart .....	17
4.2.1	Magento 1 .....	17
4.2.2	Magento 2 .....	18
4.3	Direktes Verbuchen von Bestellungen .....	19
4.4	Unsicherer Status .....	19
4.4.1	Setzen von Order-Status .....	19
4.5	Optional: Validierung .....	19
4.5.1	Nutzung der integrierten Multi-Shop Funktionalität von Magento .....	20
<b>5</b>	<b>Einstellungen / Konfiguration von Zahlungsarten .....</b>	<b>21</b>
5.1	Allgemeine Informationen zu den Zahlungsarten .....	21
5.2	Informationen zum Zahlungszustand .....	21
5.2.1	Bestellstatus "ausstehend" / bevorstehende Zahlung (oder ähnlich) .....	21
5.2.2	Abgebrochene / stornierte Bestellungen .....	22

5.3	Klarna .....	22
5.3.1	Unterstützte / Nicht unterstützte Funktionen .....	22
5.3.2	Set UP / Einrichten der Zahlart .....	22
5.3.3	Stornieren der Rechnung .....	23
5.3.4	Aktivierung von Rechnungen .....	23
5.3.5	Teilgutschriften und Teilaktivierungen .....	23
5.3.6	On Hold Bestellungen (Pending) .....	24
5.3.7	Testing .....	24
5.3.8	Zahlungsgebühren .....	24
<b>6</b>	<b>Das Modul in Aktion .....</b>	<b>25</b>
6.1	Verbuchen / Stornieren von Bestellungen .....	25
6.1.1	Magento 1 .....	25
6.1.2	Magento 2 .....	25
6.2	Nützliche Transaktionsinformationen zur Bestellung .....	27
6.2.1	Magento 1 .....	27
6.2.2	Magento 2 .....	28
6.3	Zahlungen im Magento Backend auslösen .....	29
6.3.1	Magento 1 .....	29
6.3.2	Magento 2 .....	30
6.4	Gutschriften .....	30
6.4.1	Magento 1 .....	30
6.4.2	Magento 2 .....	32
6.5	Nutzung des Alias Managers / Token Lösung .....	35
6.5.1	Magento 1 .....	35
6.5.2	Magento 2 .....	36
6.6	Steuereinstellungen .....	38
6.6.1	Magento 1 .....	38
6.7	OneStepCheckout .....	38
6.7.1	Magento 1 .....	39
6.7.2	Magento 2 .....	39
6.8	Einrichten Cron Job .....	39
6.8.1	Magento 1 .....	39
6.8.2	Magento 2 .....	39
6.9	Benutzung der Magento 1 API .....	39
6.9.1	Generelle Informationen zur Nutzung der API .....	40
6.9.2	API Calls .....	40
<b>7</b>	<b>Testing .....</b>	<b>41</b>
7.1	Testdaten .....	41
<b>8</b>	<b>Fehler und deren Lösungen .....</b>	<b>44</b>
8.1	Modul wird nicht angezeigt .....	44

8.2	Die Referrer URL erscheint in meinem Analytics Tool .....	44
<b>9</b>	<b>Kompatibilität zu Third-Party Plugins .....</b>	<b>45</b>
9.1	Magento 1 .....	45
9.1.1	Gebühren und Rabatte in Magento .....	45
9.2	Abfrage Geschlecht und Geburtstag für Magento 1 .....	45
9.3	Abfrage Geschlecht und Geburtstag für Magento 2 .....	45
<b>10</b>	<b>Error Logging .....</b>	<b>46</b>
10.1	Log Levels .....	46
10.2	Log Location .....	46
<b>11</b>	<b>Erweiterte Informationen .....</b>	<b>47</b>
11.1	Transaction Object .....	47
11.1.1	Magento 1 .....	47
11.1.2	Magento 2 .....	47

# 1 Einleitung

Diese Anleitung beschreibt die Installation, die Konfiguration und die Nutzung der Zahlungsmoduls für Magento und PayUnity.

Bevor Sie mit der Installation beginnen können, stellen Sie sicher, dass Sie im Besitz aller notwendigen Daten sind:

- Benutzername und Passwort zum Login des Backends von PayUnity
- Magento Zahlungsmodul von [sellxed.com/shop](https://sellxed.com/shop)
- Zugangsdaten zu Ihrem Server und Shop

Falls Sie noch nicht im Besitze eines Vertrags mit PayUnity sind, können Sie diesen gerne auch direkt durch uns beziehen.

**Beachten Sie, dass Sie für unsere Plugins mindestens die PHP Version 5.6 verwenden müssen. PHP 8 oder höher wird derzeit nicht unterstützt.**

## 1.1 Ablauf der Installation

In diesem Dokument erhalten Sie alle Informationen, welche für die Installation des Moduls notwendig sind. Für die Inbetriebnahme des Moduls sollten Sie die einzelnen Punkte der Checkliste strikt befolgen. Nur so kann ein sicherer Einsatz in Übereinstimmung mit allen Sicherheitsregularien gewährleistet werden.

1. Konfiguration der Testoberfläche anhand der Integrationsdaten von PayUnity. Sie finden die Testplattform unter <https://test.payunity.com/bip/login>
2. Konfiguration der Grundeinstellungen des Zahlungsmoduls
3. Konfiguration der Zahlungsarten
4. Durchführen einer Testbestellung anhand der beigelegten **Testdaten** am Ende dieses Dokuments
5. Sofern der Test erfolgreich verlaufen ist, erfolgt die Konfiguration der Live Daten in Ihrem Shop. Mit den erhaltenen Zugangsdaten können Sie sich in der Live Umgebung einloggen: <https://payunity.com/bip/login>

## Installationservice

Unsere Zahlungsmodule sind standardmässig so vorkonfiguriert, dass die Installation gleich nach dem Eintragen der benötigten Keys funktionieren sollte. Sollten Sie einen Setting aus der Konfiguration des Moduls nicht kennen, können Sie sich gerne mit unserem Support in Verbindung setzen.

Unser Support Team steht Ihnen während der Geschäftszeiten gerne zur Verfügung: <http://www.sellxed.com/support>. Sie haben zudem jederzeit die Möglichkeit unseren Installationservice zu bestellen. Wir sorgen für eine reibungslose Installation in Ihrem Shop: <http://www.sellxed.com/shop/de/integration-und-installation.html>

## .htaccess Verzeichnisschutz

Für den Test des Zahlungsmoduls muss jeglicher Verzeichnisschutz oder IP Sperre auf Ihrem Server zwingend ausgeschaltet sein, sonst kann es sein, dass das Zahlungsfeedback von PayUnity nicht bis zu Ihrem Shop durchdringt.

## 1.2 System Anforderungen

Generell hat das Plugin die gleichen Anforderungen an das System wie Magento selber. Nachfolgend finden Sie die wichtigsten Anforderungen des Plugins:

- PHP Version: 5.4.x oder höher
- Magento: 1.7.x oder höher
- OpenSSL: Aktuelle Version mit Unterstützung von TLS 1.2 oder höher.
- fsockopen: Die PHP Funktion fsockopen muss zur Verfügung stehen. Das Plugin muss Verbindungen zu anderen Servern öffnen können.
- PHP Funktionen: Im weiteren müssen alle gängigen PHP Funktionen zur Verfügung stehen.

## 2 Konfiguration

### 2.1 Grundkonfiguration des Hauptmoduls

Die Zugangsdaten für die Testumgebung erhalten Sie von PayUnity.

Navigieren Sie zum [Hauptmodul](#) in Ihrem Shop und tragen dort die folgenden Zugangsdaten ein:

- Authorization Bearer Token (Access Token)
- Entity ID (CHANNEL ID)
- Die Parameter User ID und User Password sind veraltet und werden nicht mehr benötigt, verwenden Sie stattdessen das Authorization Bearer Token.
- Den Parameter **Security Sender und Security Hash Secret** benötigen Sie für die vorliegende Integration nicht mehr. Sie können diese Eingaben einfach ignorieren.

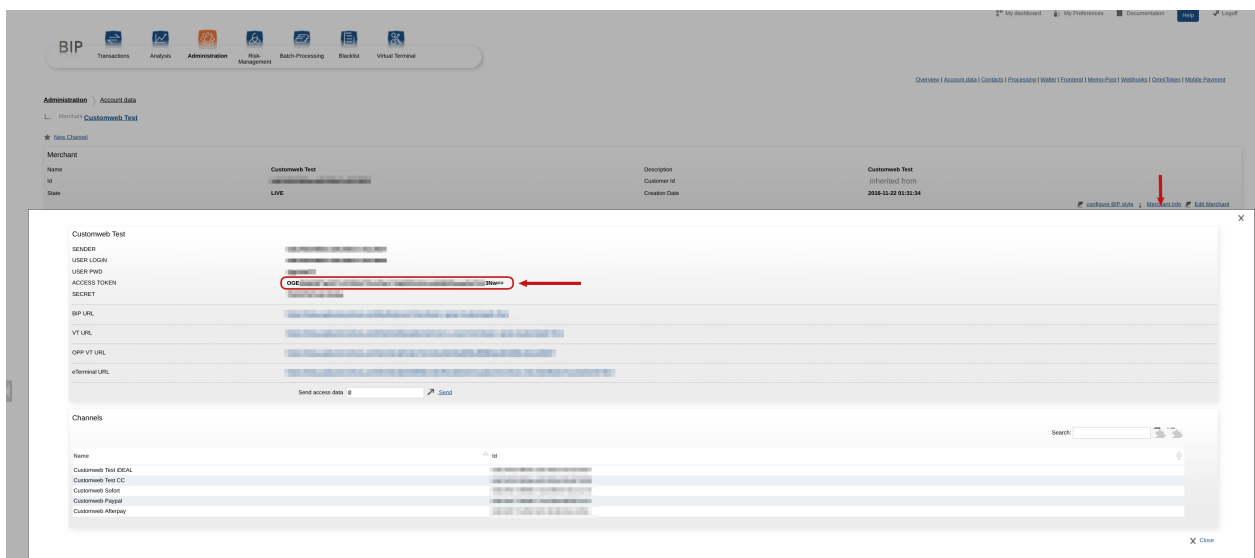


Abbildung 2.1: Die Authorization Bearer Token Information im PayUnity Backend.

Daneben haben Sie noch weitere Eingabemöglichkeiten wie zum Beispiel die **Custom Parameters**. Diese lassen Sie grundsätzlich leer ausser Sie werden von PayUnity zu einer Eingabe aufgefordert.

### 2.2 Konfiguration Zahlungsarten

Um die Zahlungsarten zu aktivieren und weitere zahlartenspezifische Einstellungen vorzunehmen, navigieren Sie zu deren Einstellungen. (Wo Sie diese finden wird sonst [hier](#) erklärt.

Sie können in der Konfiguration der Zahlungsmethoden anhand der Regeln definieren, welche Entity ID (Channel ID) für welche Beträge verwendet werden soll. Dies ermöglicht Ihnen beispielsweise für kleinere Beträge einen non-3D Secure Channel zu verwenden. Für das Set Up und die entsprechenden Verträge setzen Sie sich mit PayUnity in Verbindung.



Channel Conditions

the unit which comes transactions into the system.

[STORE VIEW]

▲ Beside the default channel ID a set of channel IDs depending on the order total amount can be defined. Each line must contain a lower amount, an upper amount and the applicable Channel ID. The format is as follow 'lower amount;upper amount;Channel ID' as shown in the following example: '10.00;200.00;23413113213131231353'. The upper boundaries are not included. If you leave this field empty or if an amount is outside any range the default channel id is applied. If multiple conditions match the last one is taken.

Entity ID (MoTo)

[STORE VIEW]

▲ If this payment method should support recurring and moto transactions, you have to specify a second entity ID without 3D secure.

COPYandPAY Style

[STORE VIEW]

▲ Define the style to be used for the COPYandPAY payment form.

Authorised status

[STORE VIEW]

▲ This status is set when the payment was successful and it is authorised.

Uncertain status

[STORE VIEW]

▲ You can specify the order status for new orders that have an uncertain authorisation status.

Captured status

[STORE VIEW]

▲ You can specify the order status for orders that are captured either directly after the order or manually in the back-end.

Approved payment Status

[STORE VIEW]

▲ You can specify the order status for orders that are approved after being in an uncertain state.

Denied Payment Status

[STORE VIEW]

▲ You can specify the order status for orders that are denied after being in an uncertain state.

Authorisation Method

[STORE VIEW]

▲ Select the authorisation method to use in order to process this payment method.

**Abbildung 2.1:** Konfiguration der Zahlungsart am Beispiel Magento (Symbolbild, die Darstellung kann bei Ihnen abweichen).

### PCI: Neue Autorisierungsmethode **Widget**

Beachten Sie, dass die Verwendung des Hidden Mode an zusätzliche Zertifizierungsanforderungen geknüpft ist (vgl unser [Blogeintrag](#) zu diesem Thema). Aus diesem Grund wird dieser in den neuen Versionen nicht mehr ausgeliefert. Sollten Sie in Ihren Zahlmethoden noch den Hidden Mode verwenden. Nutzen Sie zukünftig bitte nur noch **Widget**. Der Hidden Mode Support wird in den kommenden Monaten eingestellt.

## 2.3 Konfiguration Webhook

In Ausnahmefällen kann es passieren, dass die Zahlungsbenachrichtigung an Ihren Shop fehlschlägt, weil diese nicht bis zu Ihrem Shop durchdringt oder der Käufer den Browser unmittelbar nach dem Kauf schliesst. Hierzu empfehlen wir ein sogenanntes Webhook im PayUnity Backend unter **Administration > Webhook** einzurichten. Bitte achten Sie darauf, dass Sie die von unserem Modul dynamisch konfigurierte URL verwenden. Diese URL entnehmen Sie dem installierten Modul in Ihrem Magento Shop unter **System > PayUnity for Magento 1 and under Stores > PayUnity for Magento 2 > Setup**.

### 2.3.1 Konfiguration Webhook durch Händler

Wenn Sie das Webhook nicht durch PayUnity konfigurieren lassen können, müssten Sie bitte wie wie oben beschrieben, die URL aus Ihrem Magento Shop unter **System > PayUnity for Magento 1 and under Stores > PayUnity for Magento 2 > Setup** kopieren und das PayUnity Backend unter **Administration > Webhook** einfügen. Des weiteren müssten Sie bitte ein selbst festgelegtes "secret for encryption" generieren und die Werte in die darauf folgenden Felder, wie auf dem Bild unten zu Sehen, speichern.

#### Customweb "secret for encryption" Generator

Bitte beachten Sie, dass für den Schlüssel nur ASCII Zeichen verwendet werden dürfen. Damit Sie keine unerlaubten Zeichen benutzen, verwenden Sie bitte unseren [Schlüssel Generator](#)

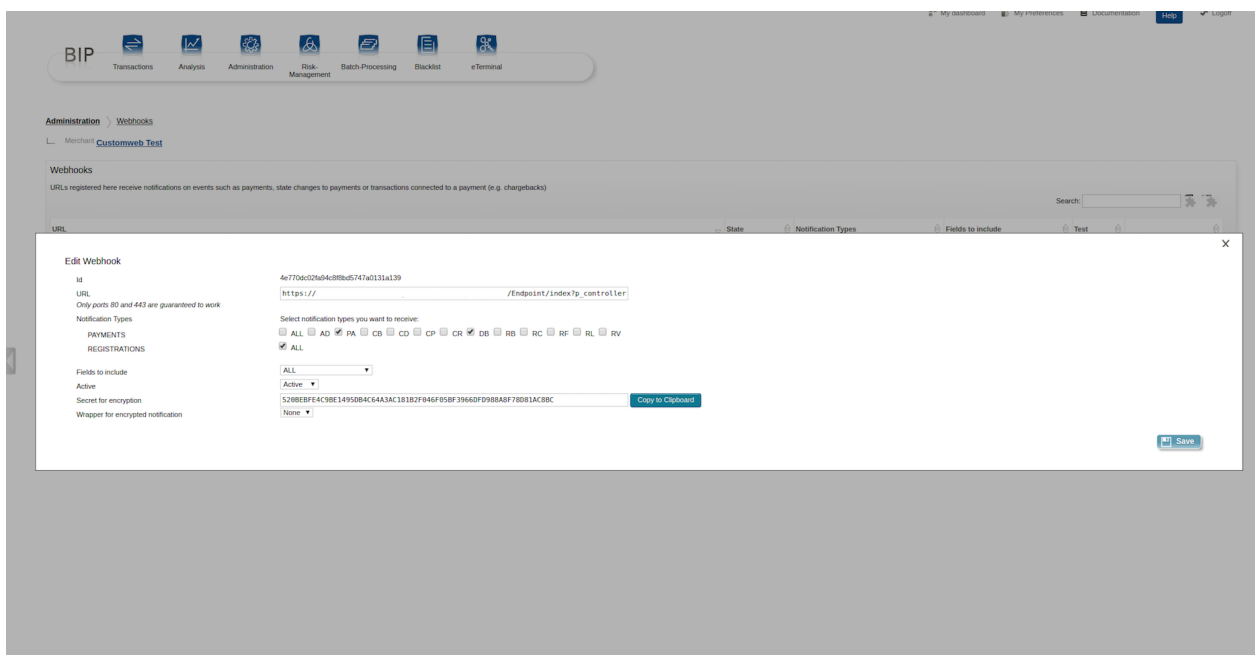


Abbildung 2.1: PayUnity Webhook Beispielkonfiguration in Ihrem Backend unter **Administration > Webhook**.

### 2.3.2 Konfiguration Webhook durch PayUnity

Wenn Sie das Webhook durch PayUnity konfigurieren lassen können, müssten Sie bitte wie wie oben beschrieben, die URL aus Ihrem Magento Shop unter **System > PayUnity for Magento 1 and under Stores > PayUnity for Magento 2 > Setup** an PayUnity übermitteln. PayUnity wird Ihnen anschliessend ein sog. "secret for encryption" zustellen, welches Sie bitte in der Hauptmodulkonfiguration des Moduls speichern müssten.

## 2.4 Aktivierung und Testing

Nach der Aktivierung und der Konfiguration der Zahlungsarten führen Sie einen Test mittels der in der Anleitung publizierten [Testdaten](#) durch.

Sofern diese Tests erfolgreich verlaufen sind, schalten Sie den Operation Mode vom Testmodus in den "Live Mode" und ersetzen die obigen Sicherheitsmerkmale durch Ihre persönlichen Zugangsdaten für die Live Plattform. Diese Daten erhalten Sie direkt von Ihrem Payment Service Provider. In der Regel sollten USER ID etc für den Live Mode identisch sein.

## 3 Modul Installation im Magento Shop

### 3.1 Installation

Sie sollten zu diesem Zeitpunkt bereits im Besitz des Moduls sein. Falls nicht erhalten Sie die Moduldateien in Ihrem Benutzerkonto im [sellxed Shop](#) (Menüpunkt "Meine Downloadartikel"). Um das Modul in Ihrem Magento Shop zu installieren, führen Sie bitte folgende Schritte durch:

1. Laden Sie das Plugin herunter. Den Download finden Sie in Ihrem sellxed.com Account unter "Meine Downloadartikel".
2. Entpacken Sie das Archive, welches Sie heruntergeladen haben.
3. Im entpackten Ordner finden Sie einen Ordner "files"
4. Für einige Shops werden verschiedene Versionen des Plugins zur Verfügung gestellt.
5. Laden Sie nun mit Ihrem preferierten FTP Programm **den ganzen Inhalt** dieses Ordners ins Stammverzeichnis Ihres Shops. Bei gewissen Shops gibt es einen spezifischen Ordner, welcher die Plugins enthält. Sollte dies der Fall sein laden Sie das Plugin in diesen Ordner hoch. Stellen Sie sicher, dass die Ordner nur zusammengeführt werden und nicht ersetzt.
6. Falls nicht bereits geschehen, loggen Sie sich bitte in Ihrem Shop ein.

Hier finden Sie die Installationsanleitung für Magento 1 sowie Magento 2.

#### 3.1.1 Installation Magento 1

##### Compiler Cache / Caching

Bevor Sie mit der Installation des Moduls beginnen stellen Sie bitte sicher, dass der Compiler Cache deaktiviert ist. Vergewissern Sie sich zudem, dass der Cache ihrer Magento 1 Installation komplett geleert wurde.

1. Erstellen Sie ein komplettes Backup ihres Shops.
2. Entpacken Sie die ZIP Datei des Moduls.
3. Kopieren Sie den **gesamten Inhalt** des Ordners "files" via FTP ins Root Verzeichnis ihrer Shopinstallation bzw. in den Ordner in welchem die Plugins normalerweise abgelegt werden.

#### 3.1.2 Magento 2

Im Folgenden werden die Schritte für die Installation und das Update der Extensions erklärt.

##### 3.1.2.1 Installations Requirements

Bitte beachten Sie, folgende Voraussetzungen sind für den Betrieb von Magento 2 zwingend zu erfüllen:

1. System requirements von Magento (<http://devdocs.magento.com/guides/v2.0/install-gde/system-requirements.html>)
2. Der Cron Job muss korrekt eingerichtet sein, sonst können keine Extensions installiert werden. Weitere Informationen finden Sie hier: <http://devdocs.magento.com/guides/v2.0/config-guide/cli/config-cli-subcommands-cron.html>

### 3.1.2.2 Installation

Für die Installation der Extension führen Sie bitte folgende Schritte durch:

#### 1. Readiness Check

Vor dem Upload des Moduls ist zwingend der Readiness Check für Magento 2 durchzuführen. Dazu gehen Sie wie folgt vor:

1. 1. Speichern Sie das Skript welches Sie auf [dieser Seite](#) herunterladen können als cw-setup.php auf ihrem Rechner.
2. 2. Laden Sie dieses cw-setup.php Skript dann in das Root Verzeichnis ihres Magento 2 Shops.
3. 3. Verbinden Sie sich über SSH mit ihrem Server und führen Sie dieses Kommando aus: `php cw-setup.php`
4. 4. Das ausgeführte Skript wird Ihnen dann weitere Befehle anzeigen welche Sie ausführen müssen.

#### Hinweis zum cw-setup.php Skript

Das zur Verfügung gestellte Skript generiert eine Liste von Kommandos welche Sie ausführen müssen um ihre Magento 2 Installation aufzuräumen. Das Skript selber führt nichts aus, es generiert lediglich die nötigen Kommandos.

#### 2. Upload

Laden Sie den Inhalt des Ordners **files\_2** nach dem Kauf in das Hauptverzeichnis Ihres Servers. Führen Sie danach die vom cw-setup.php generierten Kommandos erneut aus.

## 3.2 Updates und Upgrades

Während der Supportdauer, erhalten Sie unbeschränkten Zugriff auf Updates und Upgrades in Ihrem Kundenkonto. Um über solche Updates informiert zu werden abonnieren Sie bitte den RSS-Feed. Mehr Informationen hierzu finden Sie hier: [http://www.sellxed.com/de/updates\\_upgrades](http://www.sellxed.com/de/updates_upgrades).

### 3.2.1 Update Checklist

Bitte führen Sie vor jedem Update folgende Schritte durch:

1. Erstellen Sie immer ein Backup der Files und der Datenbank, bevor Sie mit dem Update beginnen
2. Testen Sie das Update auf einem **Testsystem**. Wir testen die Module intern, es kann aber sein, dass im Zusammenspiel mit anderen Plugins weitere Fehler auftreten.

3. Warten Sie bis die Files vollständig auf den Server gespielt sind und besuchen Sie dann die Konfigurationsseite des Hauptmoduls. Falls vorhanden leeren Sie bitte auch noch alle Caches des Shopssystems

Je nach Update kann es sein, dass eine Datenbankmigration durchgeführt werden muss. Wir empfehlen Ihnen daher, dass Update zu einem Zeitpunkt durchzuführen, an dem nicht zu viele Kunden auf Ihrem Shop zugreifen.

### Vorsicht bei Updates im Live Shop

Besondere Vorsicht ist bei Updates im Live Shop geboten. Testen Sie ein Update immer zuerst in einem Testsystem. Unser Support Team hilft Ihnen gerne bei Problemen. Bei Updates direkt im Live-System kann es jedoch zu Unterbrüchen kommen, welche je nach Support Ressourcen bis zu 2 Tage dauern können.

## 3.2.2 Update Instruktionen

Bevor Sie das Update aufspielen, lesen Sie bitte die Update Instruktionen. Diese finden sich am Ende der Pluginseite im Changelog. Sofern nichts weiteres in den Update Instruktionen steht, können die Files einfach überschrieben werden.

### 3.2.2.1 Magento 1

1. Download the content of the folder **files\_1** into the root directory of your server.
2. Then clear the Magento cache

### 3.2.2.2 Magento 2

1. Download the content of the folder **files\_2** into the root directory of your server.
2. Then click on **System > Web Setup Wizard > Component Manager**. Two modules will appear (customweb/module-base and customweb/module-PayUnity or similar). For the installation of the components click on Actions > Update.

## 4 Modul Konfiguration im Magento Shop

Die Konfiguration teilt sich in zwei Stufen. Der erste Schritt umfasst die Konfiguration des Hauptmoduls mit den Grundeinstellungen (vgl. [Konfiguration des Hauptmoduls](#)). In einem zweiten Schritt können für jede [Zahlungsart](#) individuelle Konfigurationen vorgenommen werden. Dies bringt Ihnen volle Flexibilität und grösste Anpassung an Ihre Prozesse.

### Erstellen Sie Backups!

Bitte erstellen Sie auf jeden Fall ein Backup des Hauptverzeichnisses Ihres Shops. Bei Problemen können Sie so jederzeit wieder zum Ursprungszustand zurückkehren.

Im Weiteren empfehlen wir Ihnen die Integration zuerst auf einem Testsystem durchzuführen. Es kann immer wieder zu Komplikationen mit von Ihnen installierten Drittmodulen kommen. Bei Fragen steht Ihnen unser kompetenter Support gerne zur Verfügung.

### 4.1 Konfiguration des Hauptmoduls

#### 4.1.1 Magento 1

Die Einstellungen des Hauptmoduls finden Sie unter **System > Configuration > PayUnity**. Tragen Sie hier nun die Einstellung ein welche Sie bereits in ihrem PayUnity Account getätigt oder von PayUnity erhalten haben. Informationen zu den spezifischen Einstellungen finden Sie direkt unter den jeweiligen Parameter.

Der Parameter "Time-Out for pending Payments" definiert nach wievielen Minuten der Bestellungsstatus auf "Fehlgeschlagen" wechselt (0 bedeutet keine Änderung) sollte kein Feedback zur Transaktion von PayUnity kommen. Dies tritt zum Beispiel auf wenn ein Kunde seinen Browser schliesst. Weitere Informationen diesbezüglich können unter [Order Status](#) gefunden werden.

#### 4.1.2 Magento 2

Die Einstellung des Hauptmoduls finden Sie unter **Stores > Configuration > Sales > PayUnity > General**.

Tragen Sie hier nun die Einstellung ein welche Sie bereits in ihrem PayUnity Account getätigt oder von PayUnity erhalten haben. Informationen zu den einzelnen Parameter finden Sie in den Help Boxen.



## Shop



<b>Kunden Alias Management</b>	<input type="text" value="Nein"/>	[STORE VIEW]
	Sie können Ihren Kunden gestatten ihre Aliasse auf ihrem Konto zu verwalten/löschen.	
<b>Benachrichtigung abwarten</b>	<input type="text" value="Nein"/>	[WEBSITE]
	Es kann in einigen Fällen ein paar Sekunden dauern bis die Transaktion erfolgreich genehmigt wird. Wenn diese Option aktiviert ist, wird die Benachrichtigung erwartet.	

## General



<b>Betriebsmodus</b>	<input type="text" value="Test Modus"/>	[STORE VIEW]
	If the test mode is selected the test account id is used. Otherwise the configured live account id is used.	
<b>Live account ID</b>	<input type="text"/>	[STORE VIEW]
	The live account ID as given by the <a href="#">Salesway</a> .	
<b>Live API password</b>	<input type="text"/>	[STORE VIEW]
	The live API password as given by <a href="#">Salesway</a> .	
<b>Test Account ID</b>	<input type="text"/>	[STORE VIEW]
	The test account ID as given by the <a href="#">Salesway</a> . Generic test account: 401860-17795278	
<b>Test API password</b>	<input type="text"/>	[STORE VIEW]
	The test API password as given by <a href="#">Salesway</a> . Password for generic test account: 8e7Yn5yk	
<b>Liability Shift Behavior</b>	<input type="text" value="Mark transactions without liability shift as uncertain"/>	[STORE VIEW]
	If no liability shift is given by a transaction, should the transaction be marked as uncertain?	
<b>Bestell-Präfix</b>	<input type="text"/>	[STORE VIEW]
	Here you can insert an order prefix. The prefix allows you to change the order number that is transmitted to <a href="#">Salesway</a> . The prefix must contain the tag {id}. It will then be replaced by the order number (e.g. name_{id}).	

**Abbildung 4.1:** Magento 2: Konfiguration des Basis Moduls. Bitte beachten Sie, dass dies ein Symbolbild ist und die Einstellungen bei Ihnen anders aussehen können.

## 4.2 Konfiguration der Zahlungsart

### 4.2.1 Magento 1

Nachdem das Hauptmodul erfolgreich konfiguriert wurde, finden Sie unter System > Konfiguration > Zahlungsarten die Einstellungen zu den einzelnen Zahlungsarten in Ihrem Shop. Sie können für jede Zahlungsart individuelle Einstellungen vornehmen und so die Zahlung optimal an Ihre Prozesse anpassen. Die zentralen Optionen werden in der Folge genauer erläutert.

#### 4.2.1.1 Direktes Anlegen von Rechnungen

Die "Rechnungserstellung" legt fest, ob die Rechnung direkt angelegt werden soll. Angelegte Rechnungen können nicht mehr verändert werden. Wenn Sie also die bestellten Artikel für die Verbuchung nachträglich noch anpassen möchten, stellen Sie hier bitte auf "verzögerte Erstellung".

### 4.2.2 Magento 2

Nachdem das Hauptmodul erfolgreich konfiguriert wurde, finden Sie unter Stores > Verkäufe > Zahlungsarten die Einstellungen zu den einzelnen Zahlungsarten in Ihrem Shop. Sie können für jede Zahlungsart individuelle Einstellungen vornehmen und so die Zahlung optimal an Ihre Prozesse anpassen.

#### 4.2.2.1 Verbuchung

Wählen Sie für jede Zahlungsart, ob Sie direkt oder verzögert verbuchen möchten. Bei einer verzögerten Buchung wird der Betrag nur reserviert. Es ist daher wichtig, dass Sie untenstehende Ausführungen zur Verbuchung solcher Transaktionen befolgen. Reservationen sind in der Regel nur einige Tage gültig. Aus diesem Grund empfehlen wir Ihnen die direkte Verbuchung. Die verzögerte Verbuchung ist nicht bei allen Zahlungsarten verfügbar.

#### 4.2.2.2 Autorisierungsart

Für einige Zahlungsarten können Sie zwischen verschiedenen Autorisierungsarten wählen. Die Autorisierungsart bestimmt die Art und Weise wie die Zahlformulare in Ihrem Shop dargestellt werden. Beachten Sie, dass für gewisse Autorisierungsarten bei Kreditkarten erweiterte PCI Anforderungen existieren (<https://www.sellxed.com/de/blog/pci-300-nderungen>).

## 4.3 Direktes Verbuchen von Bestellungen

Mit der Option "Buchung" legen Sie fest, ob Zahlungen direkt verbucht, oder zuerst autorisiert werden. Sofern Sie in einem ersten Schritt nur autorisieren möchten, stellen Sie sicher, dass die Zahlungen später verbucht werden.

Eine Reservation ist, abhängig von Ihrem Acquiring Vertrag, nur eine gewisse Zeit garantiert. Sollten Sie in dieser Zeit nicht verbuchen, kann es sein, dass die Autorisierung nicht mehr garantiert wird. Weitere Informationen zum Prozess der Verbuchung finden Sie weiter unten.

### Unterschiedliche Einstellungen zwischen PayUnity - Backend und Modul

Es kann sein, dass die Einstellungen im Modul für die Zahlungsweise Ihre Einstellungen im Backend von PayUnity einfach überschreiben.

## 4.4 Unsicherer Status

Sie können Bestellungen, für welche die Zahlung noch nicht garantiert werden kann, speziell markieren. Dies erlaubt Ihnen diese Bestellung vor dem Versand noch einmal manuell zu prüfen.

### 4.4.1 Setzen von Order-Status

Sie können für jede Zahlungsart festlegen in welchen Status die Bestellung je nach Buchungsstatus verschoben werden soll. Es handelt sich hier um den initialen Status der Bestellung.

## 4.5 Optional: Validierung

Mit der Validierung steuern Sie den Zeitpunkt wann eine Zahlungsart im Checkout angezeigt werden soll. Diese Einstellung ist für Module relevant bei denen die Benutzung von Voraussetzungen Ihres Kunden abhängig ist. Beispielsweise wenn noch eine Bonitätsprüfung erfolgen soll, oder die Zahlungsart nur in gewissen Ländern verfügbar ist. Es kann sein, dass diese Einstellung bei Ihnen nicht ersichtlich ist. Dann überspringen Sie diesen Punkt. Damit die Bonitätsprüfung beziehungsweise Adressvalidierung auch mit europäischen Zeichen funktioniert, muss bei gewissen PSP-Einstellungen das Charset des "Blowfish mode" auf "UTF-8" gesetzt werden.

Sollte die Einstellung ersichtlich sein, haben Sie die Wahl zwischen folgenden Optionen:

- **Validierung vor Auswahl der Zahlungsmethode:** Es erfolgt vor der Auswahl der Methode eine Validierungsprüfung. Sofern die Voraussetzungen nicht erfüllt sind, wird die Zahlungsart nicht angezeigt.
- **Validierung nach der Auswahl der Zahlungsmethode:** Die Prüfung der Voraussetzung erfolgt nach der Wahl und vor der Bestätigung der Bestellung.

- **Während der Autorisierung:** Die Validierungsprüfung erfolgt durch PayUnity während dem Autorisierungsprozess. Die Zahlungsmethode wird auf jeden Fall angezeigt.

#### **4.5.1 Nutzung der integrierten Multi-Shop Funktionalität von Magento**

Das Zahlungsmodul unterstützt das Multishop Feature von Magento. Es sind keine weiteren Anpassungen notwendig. Das Modul erkennt automatisch, welchem Shop die Bestellung zugeordnet werden muss. Voraussetzung ist jedoch, dass die einzelnen Subshops innerhalb von Magento richtig konfiguriert wurden.

## 5 Einstellungen / Konfiguration von Zahlungsarten

### 5.1 Allgemeine Informationen zu den Zahlungsarten

Das Plugin wird mit einer grossen Auswahl von Zahlungsarten ausgeliefert. Sollte eine Zahlungsart nicht verfügbar sein, bitten wir Sie sich mit uns in Verbindung zu setzen.

Für die Nutzung einer Zahlungsmethode muss die Zahlungsart in Ihrem Konto bei PayUnity und im Shop aktiviert sein. Informationen zur Konfiguration des Zahlungsmoduls finden Sie weiter oben.

In den folgenden Punkten finden Sie wichtige Informationen zu bestimmten Zahlungsarten, die sich vom Standard unterscheiden.

### 5.2 Informationen zum Zahlungszustand

Sie können für jede Zahlungsart den initialen Zahlungsstatus individuell definieren (Status für autorisierte Zahlungen etc). Dabei definieren Sie den Zahlungsstatus für die einzelnen Zustände abhängig von der Verarbeitungsart der Bestellung (verbucht, autorisiert, etc.). Es handelt sich dabei um den initialen Status, welcher die Bestellung annimmt. Abhängig von der durch Sie durchgeführten Mutationen kann sich der Status ändern.

#### Wichtige Info bezüglich dem Bestellstatus

Setzen Sie den Status nie auf **Pending PayUnity** oder einen ähnlich lautenden pending status, der durch das Modul eingeführt wird.

#### 5.2.1 Bestellstatus "ausstehend" / bevorstehende Zahlung (oder ähnlich)

Bestellungen mit dem Status 'pending PayUnity' sind pendente Bestellungen. Der Status wird gesetzt, wenn der Kunde für die Zahlung umgeleitet wird aber nicht erfolgreich zurückkehrt oder das Feedback nicht bis zu Ihrem Shop gekommen ist (Kunde hat das Fenster auf der Payment Page geschlossen und die Zahlung nicht abgeschlossen). Abhängig von der Zahlungsart werden diese Bestellungen automatisch in abgebrochene Bestellungen umgewandelt und der Lagerbestand wieder freigegeben (sofern der Cronjob Aktiviert ist). Diese Zeitspanne hängt von den Eigenheiten der Zahlungsart ab, und kann nicht konfiguriert werden.

Wenn Sie eine Vielzahl ausstehender Bestellungen haben, deutet dies in der Regel darauf hin, dass die Notifikationen von Ihrem Webserver zu PayUnity blockiert werden. In diesem Fall überprüfen Sie die Einstellungen Ihrer Firewall und bitten den Hoster die IPs und User Agents von PayUnity zu aktivieren.

## 5.2.2 Abgebrochene / stornierte Bestellungen

Bestellungen, die als abgebrochen markiert werden, wurden wie oben beschrieben automatisch in diesen Status aufgrund des Timeouts versetzt. Abgebrochen werden aber auch alle Bestellungen, welche von den Kunden aktiv abgebrochen wurden.

## 5.3 Klarna

Sie können Klarna direkt über PayUnity abwickeln. Wir gehen in der Folge auf die Besonderheiten des Klarna Setups ein.

### 5.3.1 Unterstützte / Nicht unterstützte Funktionen

Das Modul unterstützt folgende Funktionen nicht:

- Es können keine Bestellungen im Backend von Magento angelegt oder verändert werden.
- Teilaktivierungen und Teilgutschriften können nur direkt aus dem Backend von PayUnity <https://payunity.com/bip/login> erstellt werden. Aktivierungen oder Gutschriften der gesamten Rechnung sind innerhalb von Magento möglich. Mehr Informationen dazu finden Sie im Kapitel [Transaktionsmanagement](#).

### Bearbeiten von Klarna Transaktionen in Magento

Bitte beachten Sie, dass Sie in **keinem** Fall eine Klarna Transaktion in Magento bearbeiten dürfen.

Die restlichen Funktionen können durch das Modul durchgeführt werden. Im Folgenden finden Sie eine Beschreibung der wichtigsten Funktionen.

### 5.3.2 Set UP / Einrichten der Zahlart

Aktivieren Sie die Zahlart an gewohnter Stelle. Beachten Sie bitte folgendes bei der Installation:

#### 5.3.2.1 Anpassen des Anzeigenamens der Zahlart

Den angezeigten Namen im Frontend ändern Sie unter **Titel**. Bitte verwenden Sie den Titel **Rechnung** für die Zahlart. Im Weiteren definieren Sie unter **Anzeigen des Logos**, ob das Klarna Logo dem Kunden im Frontend entsprechend angezeigt wird.

#### 5.3.2.2 Anpassen der Zahlartbeschreibung

Die Beschreibung der Zahlart können Sie unter **Description** ändern. Bitte hinterlegen Sie hier folgenden Text: "**In 14 Tagen bezahlen**".

### 5.3.2.3 Wahl der Autorisierungsmethode

Wir empfehlen Ihnen die Autorisierungsmethode in der Zahlart auf **Server** einzustellen. In diesem Fall findet keine Weiterleitung auf die Payment Page von PayUnity statt und Sie können im Shop den gesamten Checkout Flow kontrollieren.

### 5.3.2.4 Klarna Merchant ID

Tragen Sie in den Einstellungen der Zahlart die Klarna Merchant ID ein. Diese erhalten Sie entweder direkt von PayUnity oder finden Sie in Ihrem Klarna Vertrag.

## 5.3.3 Stornieren der Rechnung

Wie Sie Rechnungen stornieren können, wird weiter unten im Kapitel Verbuchen / Stornieren von Bestellungen beschrieben. Mehr Informationen dazu finden Sie im Kapitel [Transaktionsmanagement](#).

### Verzögerte Rechnungsaktivierung

Beachten Sie, dass Sie in der Einstellung der Zahlart sicherstellen müssen, dass die Rechnungserstellung auf verzögert eingestellt werden muss, damit Sie die Rechnungen in Magento aktivieren können.

## 5.3.4 Aktivierung von Rechnungen

Wie Sie Rechnungen aktivieren (verbuchen) können, wird weiter unten beschrieben. Mehr Informationen dazu finden Sie im Kapitel [Transaktionsmanagement](#). Teilaktivierungen sind nur im [BIP](#) von PayUnity möglich.

### Verzögerte Rechnungsaktivierung

Beachten Sie, dass Sie in der Einstellung der Zahlart sicherstellen müssen, dass die Rechnungserstellung auf verzögert eingestellt werden muss, damit Sie die Rechnungen in Magento aktivieren können.

## 5.3.5 Teilgutschriften und Teilaktivierungen

Teilgutschriften und Teilaktivierungen können nur direkt im [BIP](#) von PayUnity durchgeführt werden.

### 5.3.6 On Hold Bestellungen (Pending)

Pending Bestellungen werden vom Modul nicht unterstützt. Damit diese vom System richtig gehandelt werden, müssen Sie im BIP für die Zahlungsart Klarna das "Status Check interval" Feld leerlassen. Bitte wenden Sie sich an PayUnity um diese Einstellung vorzunehmen.

Sofern Sie dieses Szenario testen möchten, verweisen wir hier auf die Testinformationen von Klarna (vgl. unten)

### 5.3.7 Testing

Um die Zahlart Klarna zu testen, sollten spezielle Testdaten verwendet werden. Stellen Sie aber zuerst sicher, dass der **Operation Mode auf Test ist und der Testmode auf External**.

Die offiziellen Testdaten von Klarna finden Sie direkt im Developer Portal von Klarna unter diesem Link: <https://developers.klarna.com/en/de+php/kpm/test-credentials>.

### 5.3.8 Zahlungsgebühren

Sofern Sie Zahlungsgebühren für eine Zahlart erheben möchten, nutzen Sie bitte die kompatiblen [Zahlungsgebühren Module](#). Diese Gebühren werden dann direkt an Klarna übertragen.

#### Zahlungsgebühren

Bitte beachten Sie, dass Sie keine Zahlungsgebühren im BIP von PayUnity setzen dürfen. Dafür nutzen Sie bitte die kompatiblen [Zahlungsgebühren Module](#).



## 6 Das Modul in Aktion

In der Folge finden Sie eine Übersicht über die wichtigsten Funktionen im täglichen Gebrauch des PayUnity Moduls.

### 6.1 Verbuchen / Stornieren von Bestellungen

#### 6.1.1 Magento 1

Bitte beachten Sie:

Das Transaktionsmanagement zwischen PayUnity und Ihrem Shop ist nicht synchronisiert. Wenn Sie Zahlungen bei PayUnity verbuchen, wird der Status im Shop nicht angepasst und ein zweites Verbuchen im Shop ist nicht möglich.

##### 6.1.1.1 Fall 1: Rechnungserstellung direkt nach der Bestellung

Vorliegend wird das Vorgehen beschrieben, sofern Sie oben in der Konfiguration der Zahlungsart das direkte Anlegen der Rechnung nach der Bestellung angelegt haben und Zahlungen nur autorisieren, nicht direkt verbuchen (zusätzliche Informationen zur Konfiguration vgl. [Anlegen der Rechnung](#)).

Für die Verbuchung öffnen Sie die Rechnung und klicken auf "erfassen". Sollten Sie "Abbrechen" klicken, wird die Autorisation storniert.

##### 6.1.1.2 Fall 2: Verzögerte Rechnungserstellung

Sollten Sie die Rechnungserstellung auf Verzögert eingestellt haben, müssen Sie die Rechnung zuerst anlegen indem Sie auf den Button Rechnung klicken. Wählen Sie unten bei Betrag "Capture online" und klicken Sie auf Rechnung übermitteln. Die Rechnung ist nun verbucht.

Bitte beachten Sie, wenn Sie nur Teile der Bestellung verbuchen möchten, funktioniert dies nur im Falle der verzögerten Rechnungserstellung.

##### 6.1.1.3 Stornieren einer Bestellung

Um Transaktionen zu stornieren, öffnen Sie zuerst die zugehörige Rechnung und klicken auf "Abbrechen". Danach gehen Sie zurück in die entsprechende Bestellung und stornieren auch diese Bestellung indem Sie auf "Abbrechen" klicken.

#### 6.1.2 Magento 2

### 6.1.2.1 Verbuchen Bug in Magento (v. 2.0.0)

In der Magento Version 2.0.0 existiert ein Bug. Es ist möglich, dass Rechnungen nicht korrekt verbucht werden. Bitte tätigen Sie folgende Änderungen in der Datei: **/vendor/magento/module-sales/Controller/Adminhtml/Order/Invoice/Capture.php**:

1. Löschen Sie die auf dem folgenden Bild ersichtlichen Zeilen, welche mit einem "-" beginnen.
2. Fügen Sie die mit einem "+" markierten Zeilen hinzu .

Hier finden Sie den hinzuzufügenden Code:

```
+ )->addComment( __('The credit memo has been created automatically.'));
```

**Method:** execute()

**Diff:**

```
-     try {
-         $invoiceManagement = $this->_objectManager->get('Magento\Sales\Api\InvoiceManagementInterface');
-         $invoiceManagement->setCapture($invoice->getEntityId());
+         $invoice->capture();
+         $invoice->getOrder()->setIsInProgress(true);
```

**Abbildung 6.1:** Opening the Invoice

### 6.1.2.2 Rechnungserfassung auslösen

#### Bitte beachten Sie

Das Transaktionsmanagement zwischen Ihrem Shop und PayUnity ist nicht synchronisiert. Wenn Sie die Zahlung auf Seiten PayUnity erfassen, wird der Status nicht angepasst und eine zweite Erfassung aus dem Shop ist nicht möglich.

In den Einstellungen der Zahlungsart können Sie einstellen ob die Rechnung direkt nach ihrer Ersellung erfasst werden soll ode ob dies erst geschehen soll wenn die Zahlung autorisiert wurde. sie können zudem einstellen ob die Rechnung bereits ertelt werden sollte.

Um eine Rechnung zu erfassen gehen Sie wie folgt vor:

1. Öffnen Sie die Bestellung und erstellen Sie eine Rechnung
2. Wenn das Settlement verzögert ist können Sie die Rechnung noch ändern.
3. Klicken Sie auf "Online erfassen"

<b>Invoice Settlement</b>	<input type="text" value="Deferred settlement"/>	[WEBSITE]
	<p>Choose settlement after order in case you want an invoice to be created with your order. The state of the invoice will set according to your capture status.</p>	
<b>Customer Address</b>	<input type="text" value="No address is sent"/>	[STORE VIEW]
	<p>Should the customer address be sent?</p>	
<b>Capturing</b>	<input type="text" value="Deferred"/>	[STORE VIEW]
	<p>Should the amount be captured automatically after the order (direct) or should the amount only be reserved (deferred)?</p>	
<b>Authorisation Method</b>	<input type="text" value="Payment Page"/>	[STORE VIEW]
	<p>Select the authorisation method to use in order to process this payment method.</p>	
<b>Alias Manager</b>	<input type="text" value="Inactive"/>	[STORE VIEW]
	<p>The alias manager allows the customer to select from a credit card previously stored.</p>	

Abbildung 6.1: Capture options

### 6.1.2.3 Stornieren von Bestellungen

Um eine Stornierung auszulösen müssen Sie den "Void" button benutzen, und dann "OK" betätigen. Wenn Sie den "Cancel" Button nutzen möchten müssen Sie die Zahlung im PayUnity backend stornieren.

## 6.2 Nützliche Transaktionsinformationen zur Bestellung

### 6.2.1 Magento 1

Sie finden in jeder Bestellung, die mit unserem Modul abgewickelt wurde, eine Übersicht über die wichtigsten Informationen zur Transaktion, sowie eine Transaktionshistory.


Payment Information		
<b>Authorisation amount</b>	161.0000	
<b>Currency</b>	CHF	
<b>Payment method</b>	MasterCard	
<b>Payment ID</b>	23679571	
<b>Captured amount</b>	161	
<b>Transaction authorised</b>	Yes	
<b>Transaction uncertain</b> 	No	
<b>Transaction captured</b>	Yes	
<b>Transaction paid</b>	Yes	
<b>Alias</b>	XXXXXXXXXXXX0007 (12/13)	
<b>Alias Token</b>	8B1974E9-59EB-442C-A1D4-34BEC568EE23	
<b>Acceptance</b>	test123	
<b>Card number</b>	XXXXXXXXXXXX0007	
<b>Card expiry date</b>	12/13	
<b>Merchant reference</b>	sellxed_og_100000087	
Transaction history		
Date	Action	Message
2013-09-12 12:53:50	authorization	The amount of 161.00 is authorized.
2013-09-12 12:53:50	capturing	The amount of 161.00 is captured.
Order was placed using CHF		

Abbildung 6.1: Transaction Information.

## 6.2.2 Magento 2




Sie finden in jeder Bestellung, die mit unserem Modul abgewickelt wurde, eine Übersicht über die wichtigsten Informationen zur Transaktion, sowie eine Transaktionshistory.

## Payment & Shipping Method

### Payment information



Visa

Transaction ID	14
Authorisation Amount	107
Currency	CHF
Payment period	Visa
Payment ID	h1ElhGAKOSCitASfE0IbOSHndA
Test Transaction 	Yes
Transaction authorised	Yes
Transaction uncertain 	No
Transaction paid	Yes
Card Type	Saferpay Test Card
Effective Payment Method Name 	SaferpayTestCard
Authorisation Method	PaymentPage
3D Secure	Successful

Dec 14, 2015 11:08:14 AM | Authorization  
 The amount of 107.00 is authorized. (Customer successfully returned from the Saferpay payment page.)

The order was placed using CHF.

### Shipping & Handling Information

Flat Rate - Fixed CHF10.00

Abbildung 6.1: Transaction Information.

## 6.3 Zahlungen im Magento Backend auslösen

### 6.3.1 Magento 1

Mit dem Magento Zahlungsmodul können Sie Bestellungen auch direkt aus dem Shop auslösen, sogenannte Mail Order / Telephone Order (MOTO).

1. Gehen Sie in die Übersicht der Bestellung und klicken Sie oben rechts auf "Bestellung anlegen".
2. Es öffnet sich das gewohnte Fenster für die Erstellung von Bestellungen. Wählen Sie wie den gewünschten Kunden, Artikel und Adresse aus

3. Im Feld Zahlungsart, wählen Sie die gewünschte Zahlungsart und klicken auf "Bestellung abschliessen".
4. Abhängig von der Autorisierungsart, welche Sie bei der ausgewählten Zahlungsart hinterlegt haben, öffnet sich die Payment Page von PayUnity oder eine Eingabe für die Kreditkarte wie im Bild unten. Erfassen Sie die Kreditkarte Ihres Kunden.

#### Mail order/telephone order authorization

---

Card holder name\*   
Please enter the card holder's name.

Card number\*   
Please enter your card number

Card expiration date\*    
Select the expiry date of your card

CVC Code\*   
Please enter the CVC code from your card. You can find the code on the back of the card

 **Continue**

Abbildung 6.1: MoTo Transaktionen in Magento

## 6.3.2 Magento 2

Mit dem Magento Zahlungsmodul können Sie Bestellungen direkt aus dem Shopbackend heraus erstellen. Dies Bestellungen werden Mail Order / Telephone Order (MOTO) genannt.

1. Navigieren Sie zur Bestellungsübersicht und klicken Sie auf "Neue Bestellung".
2. Wählen Sie den gewünschten Kunden, Artikel sowie die Adresse
3. Wählen Sie die gewünschte Zahlungsart und klicken Sie dann auf "Bestellung abschliessen".
4. Danach werden Sie, abhängig von Ihren Einstellungen, entweder auf die Payment Page von PayUnity weiter geleitet oder aufgefordert die Ihre Zahlungs Details direkt im shop einzugeben.

Falls ein Alias / Token des Kunden existiert kann alternativ auch dieses verwendet werden.

## 6.4 Gutschriften

### 6.4.1 Magento 1

Für verbuchte Transaktionen können Sie auch Gutschriften erstellen und automatisch an PayUnity übermitteln. Zu diesem Zweck öffnen Sie die Rechnung einer bereits verbuchten

Bestellung. Durch den Klick auf "Credit Memo" öffnet sich der Dialog für die Gutschriften. Passen Sie den Betrag an, welchen Sie zurückerstatten wollen. Mit dem Klick auf Refund wird die Gutschrift an PayUnity übermittelt.

Hier finden Sie die einzelnen Schritte nochmals mit Screenshots dargestellt:

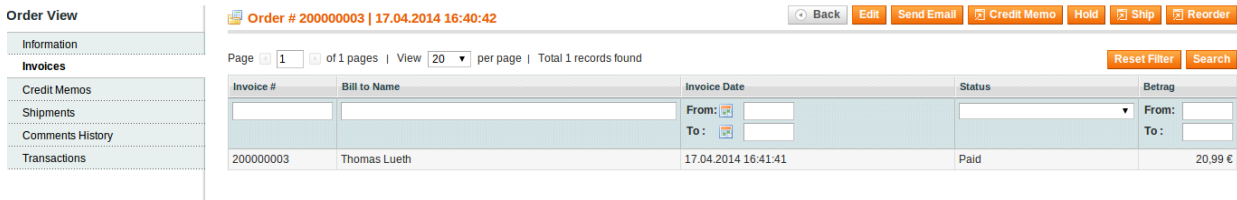


Abbildung 6.1: 1. Öffnen der Rechnung

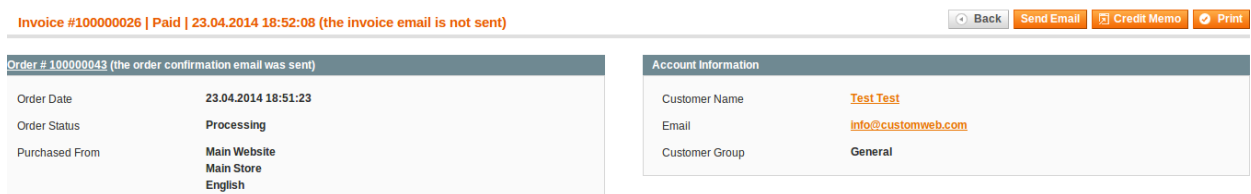


Abbildung 6.1: 2. Klicken Sie auf Credit Memo - wichtig die Rechnung muss zuerst geöffnet werden.



Abbildung 6.1: 3. Anpassen der Beträge (vgl. unten Gutschrift auf Basis der Menge).

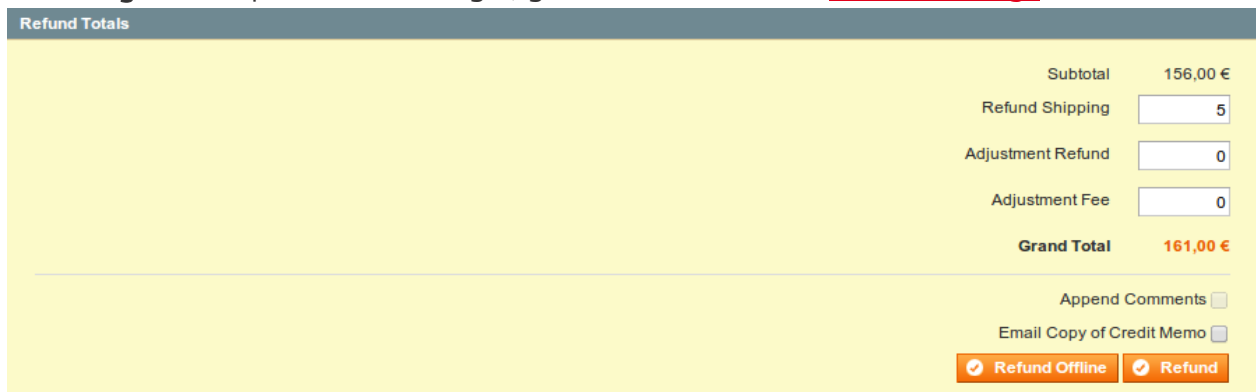


Abbildung 6.1: 4. Für die Übermittlung klicken Sie auf Refund.

✔ The credit memo has been created.

Order # 10000041 | 23.04.2014 18:26:16 Back Send Email Reorder

Order # 10000041 (the order confirmation email was sent)		Account Information	
Order Date	23.04.2014 18:26:16	Customer Name	<a href="#">Test Test</a>
Order Status	Closed	Email	<a href="mailto:info@customweb.com">info@customweb.com</a>
Purchased From	Main Website Main Store English	Customer Group	General
Placed from IP	160.85.155.47		

Abbildung 6.1: Warten Sie auf die Bestätigung der Gutschrift.

### 6.4.1.1 Gutschrift basierend auf Produktmengen

Sie können Gutschriften auf Basis der Produktmengen durchführen. Dafür passen Sie nach dem Öffnen des Credit Memo Dialogs die Anzahl der Produkte an, welche Sie der Kundenkarte gutschreiben möchten.

Sie können so viele einzelne Gutschriften durchführen, bis Sie die gesamte Menge der ursprünglich bestellten Artikel gutgeschrieben haben. Der Status der Bestellung wechselt danach automatisch in den State **complete**.

Order # 20000055 | 02.12.2013 16:37:59 Back Edit Send Email Credit Memo Hold Ship Reorder

Page 1 of 1 pages | View 20 per page | Total 2 records found Reset Filter Search

Credit Memo #	Bill to Name	Created At	Status	Refunded
200000001	Leandra Similovic	02.12.2013 16:42:10	Refunded	35,00 €
200000002	Leandra Similovic	02.12.2013 16:47:03	Refunded	146,00 €

Abbildung 6.1: Übersicht aller getätigten Gutschriften

### 6.4.1.2 Gutschreiben eines beliebigen Betrags

Da die einzelnen Positionen der Bestellung an PayUnity übermittelt werden müssen, können wir die Funktion **Adjustment Refund** - also die Gutschrift eines beliebigen Betrags nicht unterstützen. Es können nur jeweils ganze Artikel durch Anpassen der Gutschriftsmengen gutgeschrieben werden.

## Anpassung der Beträge bei mehreren Refunds

Es ist möglich, mehrere Gutschriften auf einer Bestellung durchzuführen. Bitte beachten Sie, dass Sie maximal 100 % des ursprünglichen Betrages gutschreiben können. Wenn Sie bereits eine Gutschrift durchgeführt haben, müssen Sie durch Anpassen der Menge und der Adjustment Fee darauf achten, dass nicht mehr als der initiale Betrag gutgeschrieben wird.

## 6.4.2 Magento 2



### 6.4.2.1 Bug in der Version 2.0.0

In der Magento Version 2.0.0 existiert ein Bug. Es ist möglich, dass Rechnungen nicht korrekt verbucht werden. Bitte tätigen Sie folgende Änderungen in der Datei: `/vendor/magento/module-sales/Model/Order/Payment.php` :

1. Löschen Sie die auf dem folgenden Bild ersichtlichen Zeilen, welche mit einem "-" beginnen.
2. Fügen Sie die mit einem "+" markierten Zeilen hinzu .

**Method:** `registerRefundNotification($amount)`

**Diff:**

```

        $creditmemo->setPaymentRefundDisallowed(
            true
        )->setAutomaticallyCreated(
            true
-       )->register()->addComment(
+       )->addComment(
            __('The credit memo has been created automatically.')
        );
    
```

**Abbildung 6.1:** Adjustments so you are able to enter a credit memo.

Hier finden Sie den hinzuzufügenden Code:

```
+ )->addComment( __('The credit memo has been created automatically.'));
```

### 6.4.2.2 Rückerstattungen ausführen

Für bereits getätigte Transaktionen können Sie eine Rückerstattung auslösen und diese automatisch an PayUnity übergeben. Um dies zu tun öffnen Sie eine verbuchten Bestellung. Klicken Sie auf "Credit Memo". Ein Fenster für die Rückerstattung wird sich öffnen. Hier können Sie die Anzahl Produkte welche Sie rückerstatten möchten anpassen.

Hier finden Sie die verschiedenen Schritte als Bilder:

#000000003

🔍 🔔 👤 admin ▾

[← Back](#)   [Send Email](#)   [Credit Memo](#)   [Hold](#)   [Ship](#)

- ORDER VIEW**
- Information
  - Invoices
  - Credit Memos
  - Shipments
  - Comments History
  - Transactions

🔍   [Filters](#)   [Default View](#) ▾   [Columns](#) ▾   [Export](#) ▾

Select it... ▾   1 records found   20 ▾ per page   <   1   of 1   >

<input type="checkbox"/>	Invoice ↓	Invoice Date	Order Number	Order Date	Bill-to Name	State	Amount	Action
<input type="checkbox"/>	000000003	Dec 14, 2015 7:55:37 AM	000000003	Dec 14, 2015 7:54:41 AM	Test Customweb	Paid	CHF39.00	<a href="#">View</a>

Abbildung 6.1: 1. Opening the Invoice

New Memo for #000000003

🔍 🔔 👤 admin ▾

[← Back](#)   [Reset](#)

Order & Account Information

**Order # 000000003** (The order confirmation email was sent)

Order Date	Dec 14, 2015, 7:54:41 AM
Order Status	Processing
Purchased From	Main Website Main Website Store Default Store View
Placed from IP	212.90.201.186

Account Information

Customer Name	Guest
Email	<a href="mailto:info@customweb.com">info@customweb.com</a>
Customer Group	NOT LOGGED IN

Address Information

**Billing Address** [Edit](#)

Test Customweb  
Teststrasse 2  
Zürich, Zug, 8032  
Switzerland  
T: 555466684

**Delivery address** [Edit](#)

Test Customweb  
Teststrasse 2  
Zürich, Zug, 8032  
Switzerland  
T: 555466684

Abbildung 6.1: 2. Click on Credit Memo - Important the invoice needs to be opened first.

**Refund Totals**

Subtotal	156,00 €
Refund Shipping	<input type="text" value="5"/>
Adjustment Refund	<input type="text" value="0"/>
Adjustment Fee	<input type="text" value="0"/>
<b>Grand Total</b>	<b>161,00 €</b>

Append Comments  
 Email Copy of Credit Memo

Abbildung 6.1: 3. Choose which products you would like to credit and click on Refund Online.

#000000003

🔍 🔔 👤 admin ▾

[← Back](#)   [Send Email](#)   [Credit Memo](#)   [Hold](#)   [Ship](#)

---

**ORDER VIEW**

Information

Invoices

**Credit Memos**

Shipments

Comments History

Order History

Transactions

Search by keyword

Select It... ▾   0 records found   20 ▾ per page   < 1 of 1 >

	Credit Memo	Created	Order	Order Date	Bill-to Name	State	Refunded	Action
We couldn't find any records.								

Abbildung 6.1: Overview of all the credits made

## Anpassungen für mehrere Rückerstaungen

Es ist möglich für eine Bestellug mehrere Rückerstattungen auszulösen. Es ist nicht möglich mehr als 100% des ursprünglichen Betrags rückzuerstatten.

## 6.5 Nutzung des Alias Managers / Token Lösung

### 6.5.1 Magento 1

Mit dem Alias Manager können Kreditkarten Ihrer Kunden für spätere Einkäufe bei PayUnity sicher gespeichert werden. Die Nutzung wird aktiviert indem Sie die Option "Alias Manager" in der jeweiligen Zahlungsart aktivieren. Für spätere Einkäufe kann der Kunde zwischen den bereits gespeicherten Karten wählen und muss so nicht die gesamte Karte nochmals erfassen.

**Visa**

Card N°: 9451 1231 0000 0111 Date of Expiry: 12/2015 CVV: 123

**Saved cards:**

xxxx xxxx xxxx 0111

You may choose from one of the cards previously used on this site.

**Card holder name \***

Test Alias

Please enter the card holder's name.

**Card number \***

xxxx xxxx xxxx 0111

Please enter your card number

**Card expiration date \***

12 2015

Select the expiry date of your card

**CVC Code \***

Please enter the CVC code from your card. You can find the code on the back of the card

Abbildung 6.1: Alias Manager Transaktion

**Achtung!**

Die Nutzung des Alias Managers benötigt die Freischaltung der entsprechenden Option bei PayUnity. Nehmen Sie dafür bitte direkt mit dem Support Kontakt auf.

### 6.5.1.1 Entfernen von Kunden Aliasen

Sie können gespeicherte Aliase von Kunden auch wieder entfernen. Öffnen Sie hierfür den Kunden unter Customers und gehen in den Tab PayUnity Aliases. Hier können Sie den Alias wieder entfernen.

Admin Customweb Back Reset Create Order Delete Customer Save Customer Save and Continue Edit

Page 1 of 1 pages | View 20 per page | Total 1 records found Reset Filter Search

Beispiel-Alias	Zahlungsmethode	Aktion
XXXXXXXXXXXX0002 (02/16)	Visa	Löschen

Abbildung 6.1: Entfernen von gespeicherten Aliasen.

### 6.5.2 Magento 2

Dank dem Alias Manager / Token können Kreditkartendaten des Kunden für spätere Einkäufe gespeichert werden. Somit muss der Kunde seine Kreditkartendaten nicht erneut eingeben.

### 6.5.2.1 Aktivierung des Alias Manager / Token

Der Alias Manager / Token kann in den Einstellungen der zahlungsarten aktiviert werden.

#### Payment

Check / Money order

Test Payment Method

VISA

Hans Muster  
 Beispielstrasse 1  
 Zürich, Zürich 8000  
 Switzerland  
 044 123 45 67  
[Edit](#)



#### Stored Cards

xxxxxxxxxxxx0002

You may choose one of your stored cards.

#### Card holder name \*

Hans Muster

Please enter the card holder's name.

#### Card number \*

xxxxxxxxxxxx0002

Please enter your card number.

#### Expiry Date \*

05 2018

Select the expiry date of your card.

#### CVC Code \*

Please enter the CVC code from your card. You can find the code on the back of the card.

**Place Order**

#### Order Summary

Cart Subtotal CHF28.00

**Order total CHF28.00**

2 Items in Cart



LifeLong Fitness IV CHF28.00

Qty: 2

Abbildung 6.1: Alias Manager Transaction

### 6.5.2.2 Löschen des Alias / Token durch den Kunden

Sie können Ihren Kunden erlauben die Kreditkaren Alias / Token in deren Kundenaccount unter "Meine PayUnity Alias / Token" zu löschen. Diese Funktion kann in den Einstellungen der Zahlungsart aktiviert werden.

<b>Customer Alias Management</b>	Yes	[STORE VIEW]
	<small>You can allow the customers to manage/delete their aliases in their account.</small>	
<b>Await Notification</b>	Yes	[WEBSITE]
	<small>In some cases it may take a few seconds until the transaction is successfully authorized. When this option is enabled, the notification is awaited.</small>	
<b>External Checkout: Guest Checkout</b>	Show Account Selection	[WEBSITE]
	<small>When using an external checkout, the customer can either be asked to choose an option to authenticate (as guest, register or login) or he can always be checked out as guest. For the second option to work, guest checkout has to be enabled in Magento.</small>	

Abbildung 6.1: Activating the Alias Manager for Customers.

## 6.6 Steuereinstellungen

### 6.6.1 Magento 1

Damit die Steuern der Artikel und Gutscheinen korrekt an Rechnungsanbieter wie beispielsweise Klarna übermittelt werden, stellen Sie bitte sicher, dass Sie in Ihrem Magento Shop die Steuer Einstellungen entsprechend den Screenshots weiter unten eingestellt haben. Die Einstellungen finden Sie unter **System > Configuration > Tax**.

Calculation Settings		
Tax Calculation Method Based On	Total	[WEBSITE]
Tax Calculation Based On	Shipping Address	[WEBSITE]
Catalog Prices	Excluding Tax	[WEBSITE]
	<small>▲ Whether catalog prices entered by admin include tax.</small>	
Shipping Prices	Excluding Tax	[WEBSITE]
	<small>▲ Whether shipping amounts entered by admin or obtained from gateways include tax.</small>	
Apply Customer Tax	Before Discount	[WEBSITE]
Apply Discount On Prices	Including Tax	[WEBSITE]
	<small>▲ Apply discount on price including tax is calculated based on store tax, if "Apply Tax after Discount" is selected.</small>	
Apply Tax On	Custom price if available	[WEBSITE]
Enable Cross Border Trade	No	[WEBSITE]
	<small>▲ When catalog price includes tax, enable this setting will fix the price no matter what the customer's tax rate is.</small>	

Abbildung 6.1: Empfohlene Steuerberechnungseinstellungen für Magento.

Bitte stellen Sie zudem sicher, dass in den Einstellungen zu Ihren Vouchern im Tab Actions die Einstellung "Apply to Shipping Amount" auf Nein gestellt ist. Magento weist die Steuer auf den Rabatten für die Lieferung nicht korrekt aus.

## 6.7 OneStepCheckout

### 6.7.1 Magento 1

Der Checkout von Magento kann angepasst und auf einen Schritt reduziert werden. Sogenannte OnePageCheckout. Unser Modul unterstützt die wichtigsten OnePageCheckout Module. Wenn Sie sich überlegen ein OneStepCheckout Modul mit unserem PayUnity Magento Modul einzusetzen, stellen Sie sicher, dass Sie eine der folgenden Extensions verwenden:

- OneStepCheckout von [www.onestepcheckout.com](http://www.onestepcheckout.com)
- OneStepCheckout von [Ahead Works](#)
- LightCheckout von [GoMage](#)
- Firecheckout von [Templates Master](#)

### 6.7.2 Magento 2

Es werden keine OneStep Checkout Module für Magento 2 unterstützt.

## 6.8 Einrichten Cron Job

### 6.8.1 Magento 1

Um die zeitgesteuerten Funktionen des Plugins zu aktivieren (bspw. Update Service, Löschen pendenteter Bestellungen, etc.) müssen Sie in Magento Cron Jobs einrichten. Insbesondere die Update funktion ermöglicht Ihnen nachträgliche Änderungen der Transaktion bei PayUnity über die API abzuholen (Bitte beachten Sie es kann sein, dass hier zusätzliche Optionen freigeschaltet werden müssen.

Das Modul verwendet den Standard Magento Cron. Informationen zur Einrichtung finden Sie [hier](#).

Wir raten Ihnen ein Corn Engine zu verwenden wie z.B. [EasyCron](#). Dies erlaub es Ihnen die Datei (URL) mit einbem externen Service zu öffnen.

### 6.8.2 Magento 2

Das Modul benutzt das standart Cron Engine von Magento 2. Mehr Informationen dazu findn Sie [hier](#).

## 6.9 Benutzung der Magento 1 API

Unser Modul gibt Ihnen über die API zahlreiche hilfreiche Informationen zur Transaktion. Wir beschreiben Ihnen hier kurz, welche calls möglich sind und welche Informationen zur Verfügung gestellt werden

## 6.9.1 Generelle Informationen zur Nutzung der API

Allgemeine Informationen zur Verwendung der Magento SOAP API finden Sie hier: <http://www.magentocommerce.com/api/soap/introduction.html>

## 6.9.2 API Calls

Wir haben hier in der Übersicht dargestellt, welche API Calls für welche Version der API unterstützt werden können.

### 6.9.2.1 API V1

- `$client->call($session, 'payunitycw_transaction.list');` //Liste über alle Transaktionen
- `$client->call($session, 'payunitycw_transaction.info', {transactionsId});` // Einzelne Transaktion

### 6.9.2.2 API V2

- `$client->payunitycwTransactionList($session);` //Liste über alle Transaktionen
- `$client->payunitycwTransactionInfo($session, {transactionsId});` //Einzelne Transaktion

### 6.9.2.3 XML-RPC

- `$client->call('call', array($session, 'payunitycw_transaction.list');` //Liste über alle Transaktionen
- `$client->call('call', array($session, 'payunitycw_transaction.info', {transactionsId}));`  
//Einzelne Transaktion

### 6.9.2.4 REST

- `$this->callApi('payunitycw/transactions');` //Liste über alle Transaktionen
- `$this->callApi('payunitycw/transactions/{transactionsId}');` //Einzelne Transaktion



## 7 Testing

Bevor Sie vom Test Modus in den Live Modus wechseln, sollten Sie das Modul ausführlich testen.

### Testen

Vergessen Sie nicht, nach erfolgreichem Testing den Betriebsmodus von Test auf Live umzustellen.

### 7.1 Testdaten

Nachfolgend finden Sie die Testdaten für die Zahlungsmethoden:

#### American Express

Kartennummer	377777777777770	No 3D Secure
Verfallsdatum	12/2020	
CVC:	123	
Kartennummer	375987000000005	3D Secure
Verfallsdatum	12/2020	
CVC:	123	

#### Carte Bleue

Kartennummer	5555555555554444
Verfallsdatum	12/2020
CVC:	123

#### Dankort

Kartennummer	5019717010103742
Verfallsdatum	12/2020
CVC:	123

#### Diners Club

Kartennummer	36961903000009
Verfallsdatum	12/2020
CVC:	123

#### Sepa Direct Debits

IBAN	AT152011128161647502	Austria (AT)
BIC	GIBAATWWXXX	
IBAN	DE23100000001234567890	Germany (DE)
BIC	MARKDEF1100	
IBAN	ES9121000418450200051332	Spain (ES)
BIC	CAIXESBBXXX	

#### Discover Card

Kartenummer 6011587918359498  
 Verfallsdatum 12/2020  
 CVC: 123

**giropay**

IBAN	AT152011128161647502	Austria (AT)
BIC	GIBAATWWXXX	
IBAN	DE23100000001234567890	Germany (DE)
BIC	MARKDEF1100	
IBAN	ES9121000418450200051332	Spain (ES)
BIC	CAIXESBBXXX	

**JCB**

Kartenummer 3541599999092431  
 Verfallsdatum 12/2020  
 CVC: 123

**Klarna Invoice**

Klarna (Approved)  
 Klarna (Denied)  
 Klarna (Pending -> Approved)  
 Klarna (Pending -> Denied)

**Maestro**

Kartenummer 6799851000000032  
 Verfallsdatum 12/2021  
 CVC: 123

**MasterCard**

Kartenummer	5454545454545454	No 3D Secure
Verfallsdatum	12/2021	
CVC:	123	
Kartenummer	5212345678901234	3D Secure
Verfallsdatum	12/2021	
CVC:	123	

**Visa**

Kartenummer	4200000000000000	No 3D Secure
Verfallsdatum	12/2020	
CVC:	123	
Kartenummer	4012001037461114	3D Secure
Verfallsdatum	12/2020	
CVC:	123	
Kartenummer	4000000000000010	3D Secure 2.0
Verfallsdatum	12/2021	

CVC: 123

**V PAY**

Kartenummer 48220000000000000003

Verfallsdatum 12/2020

CVC: 123

## 8 Fehler und deren Lösungen

Sie finden ausführliche Informationen unter <http://www.sellxed.com/de/fag>. Sollte sich Ihr Problem mit den zur Verfügung gestellten Informationen nicht lösen lassen, melden Sie sich bitte direkt bei uns: <http://www.sellxed.com/de/support>

### 8.1 Modul wird nicht angezeigt

Wenn das Modul nicht angezeigt wird, obwohl die Files korrekt auf den Server geladen wurden, dann prüfen Sie bitte Folgendes:

- Im Verzeichnis /app/etc befindet sich die local.xml
- Dort muss sichergestellt sein, dass **disable\_local\_modules** auf false gestellt ist.

### 8.2 Die Referrer URL erscheint in meinem Analytics Tool

Unter Umständen kann die Referrer URL von PayUnity fälschlicherweise im Analytics Tool erscheinen. Dies passiert, wenn der Kunde mit einer HTTP Header Weiterleitung zurück in den Shop geleitet wird. Die meisten Analytic Tools können dieses Problem jedoch minimieren, indem die entsprechende URL aus dem Report ausgeschlossen wird.

Für Google Analytics hilft dieses Step-by-Step Guide am Beispiel von PayPal: [Weitere Infos sind unter Punkt 4 beschrieben.](#)

## 9 Kompatibilität zu Third-Party Plugins

Unten aufgeführte Plugins sind kompatibel mit unseren Zahlungsmodulen und erleichtern Ihnen in gewissen Bereichen die Arbeit.

### 9.1 Magento 1

#### 9.1.1 Gebühren und Rabatte in Magento

Um Gebühren und Rabatte basierend auf den Zahlungsarten der PayUnity Zahlungsschnittstelle zu konfigurieren, ist es Ihnen möglich folgendes 3rd-Party Plugin zu verwenden.

- [Fooman Surcharge](#)

### 9.2 Abfrage Geschlecht und Geburtstag für Magento 1

Für gewisse Zahlungsanbieter ist es nötig den Geburtstag sowie das Geschlecht des Kunden zu prüfen. Magento 1 tut dies standartmässig nicht. Sie können diese Abfrage für Magento 1 unter "System > Configuration | Customers > Customer Configuration > Name and Address Options" aktivieren.

### 9.3 Abfrage Geschlecht und Geburtstag für Magento 2

Für gewisse Zahlungsanbieter ist es nötig den Geburtstag sowie das Geschlecht des Kunden zu prüfen. Magento 2 tut dies standartmässig nicht. Sie können diese Abfrage für Magento 2 unter "Stores > Configuration | Customers > Customer Configuration > Name and Address Options" aktivieren.

## 10 Error Logging

Das Modul logged verschiedene unerwartete Fehler und sonstige Fehler abhängig vom konfigurierten Logging Level. Sollte es ein Problem mit dem Modul geben kann das Log Hinweise auf dessen Ursache liefern.

### 10.1 Log Levels

Sie können das Log Level in den PayUnity Einstellungen konfigurieren. Die Log Nachrichten werden dadurch nicht beeinflusst.

- Error: Logt nur unerwartete Fehler. (Standart)
- Info: Logt generelle Laufzeit Informationen.
- Debug: Logt fürs Debugging nützliche Informationen.

### 10.2 Log Location

Die Log Datei wird im Magento Log Verzeichnis gespeichert. (Default Path: {shopRootDirectory}/var/log)

# 11 Erweiterte Informationen

Dieser Abschnitt des Manuals beinhaltet Informationen für Anwender mit speziellen Anforderungen. Sämtliche hier aufgeführte Informationen und Konfigurationen sind optional und für die normale Verwendung des Plugins nicht erforderlich.

## 11.1 Transaction Object

In diesem Abschnitt finden Sie Informationen wie Sie Daten zu einer Transaktion aus der Datenbank auslesen können um diese, z.B. in einem ERP System, weiter zu verarbeiten.

Für die Code Snippets in diesem Abschnitt wird angenommen, dass sich Ihr Script im Stammordner des Shops, in einer Standard Ordnerstruktur befindet.

### 11.1.1 Magento 1

Initialisieren der Magento Grundfunktionen.

```
define('MAGENTO_ROOT', getcwd());
$mageFilename = MAGENTO_ROOT . '/app/Mage.php';
require_once $mageFilename;
Mage::setIsDeveloperMode(true);
ini_set('display_errors', 1);
umask(0);
Mage::app();
```

Nun können Sie die Transaktion laden und das TransaktionsObjekt extrahieren.

Laden des Transaktionsobjekt by Id:

```
$dbtransaction = Mage::getModel('payunitycw/transaction')->load($transactionId);
$transactionObject = $dbtransaction->getTransactionObject();
```

Laden des Transaktionsobjekt by Order Number:

```
$dbtransaction = Mage::getModel('payunitycw/transaction')->load($orderId, 'transaction_external_id');
$transactionObject = $dbtransaction->getTransactionObject();
```

### 11.1.2 Magento 2

Initialisieren der Magento Grundfunktionen.

```
use Magento\Framework\App\Bootstrap;
require __DIR__ . '/app/bootstrap.php';
$bootstrap = Bootstrap::create(BP, $_SERVER);
```

```
$obj = $bootstrap->getObjectManager();  
$state = $obj->get('Magento\Framework\App\State');  
$state->setAreaCode('frontend');
```

Laden der Transaktions Factory.

```
$factory = $obj->get  
( 'Customweb\PayUnityCw\Model\Authorization\TransactionFactory' )->create();
```

Nun können Sie die Transaktion laden und das TransaktionsObjekt extrahieren.

Laden des Transaktionsobjekt by Id:

```
$dbTransaction = $factory->load($transactionId, 'entity_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```

Laden des Transaktionsobjekt by Order Number:

```
$dbTransaction = $factory->load($orderId, 'order_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```

Laden des Transaktionsobjekt by Payment Id:

```
$dbTransaction = $factory->load($paymentId, 'payment_id');  
$transactionObject = $dbTransaction->getTransactionObject();
```