

Installationsanleitung für PayUnity Gambio

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

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

Version: 1.0.31

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

E-Mail: 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

1	Einleitung	5
1.1	Ablauf der Installation	5
1.2	System Anforderungen	6
2	Konfiguration	8
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
3	Modul Installation im Gambio Shop	13
3.1	Installation	13
3.2	Sprache	13
3.3	Updates und Upgrades	13
3.3.1	Update Checklist	13
3.3.2	Update Instruktionen	14
4	Modul Konfiguration im Gambio Shop	15
4.1	Konfiguration des Hauptmoduls	15
4.1.1	Ausstehende Bestellungen erstellen	15
4.1.2	Datenbank Codierung	16
4.2	Direktes Verbuchen von Bestellungen	16
4.3	Unsicherer Status	16
4.3.1	Setzen von Order-Status	16
4.4	Optional: Validierung	16
5	Einstellungen / Konfiguration von Zahlungsarten	18
5.1	Allgemeine Informationen zu den Zahlungsarten	18
5.2	Informationen zum Zahlungszustand	18
5.2.1	Bestellstatus "ausstehend" / bevorstehende Zahlung (oder ähnlich)	18
5.2.2	Abgebrochene / stornierte Bestellungen	19
5.3	Klarna	19
5.3.1	Unterstützte / Nicht unterstützte Funktionen	19
5.3.2	Set UP / Einrichten der Zahlart	19
5.3.3	Stornieren der Rechnung	20
5.3.4	Aktivierung von Rechnungen	20

5.3.5	Teilgutschriften und Teilaktivierungen	20
5.3.6	On Hold Bestellungen (Pending)	21
5.3.7	Testing	21
5.3.8	Zahlungsgebühren	21
5.3.9	Code Anpassungen für die Integration von MasterPass (Gambio GX 1.x)	21
5.3.10	Diverses	22
6	Das Modul in Aktion	23
6.1	Nützliche Transaktionsinformationen zur Bestellung	23
6.2	Zahlungen im Gambio Backend auslösen	24
6.2.1	Rückvergüten von Bestellungen	25
6.3	Verbuchen / Stornieren von Bestellungen	26
6.3.1	Verbuchen / stornieren von Bestellungen	26
6.4	Nutzung des Alias Managers / Token Lösung	26
6.5	Einrichten Cron Job	27
7	Testing	28
7.1	Testdaten	28
8	Fehler und deren Lösungen	31
8.1	Die Referrer URL erscheint in meinem Analytics Tool	31
9	Error Logging	32
10	Erweiterte Informationen	33
10.1	Transaction Object	33

1 Einleitung

Diese Anleitung beschreibt die Installation, die Konfiguration und die Nutzung der Zahlungsmoduls für Gambio 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
- Gambio Zahlungsmodul von 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 Gambio selber. Nachfolgend finden Sie die wichtigsten Anforderungen des Plugins:

- PHP Version: 5.4.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.

Gambio v3.2.x

Gambio Version 3.2.x verwendet einen anderen Checkout-Prozess wie neuere und ältere Versionen, und wird nicht unterstützt. Bitte installieren Sie eine neuere Version von Gambio wenn Sie diese verwenden.

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.

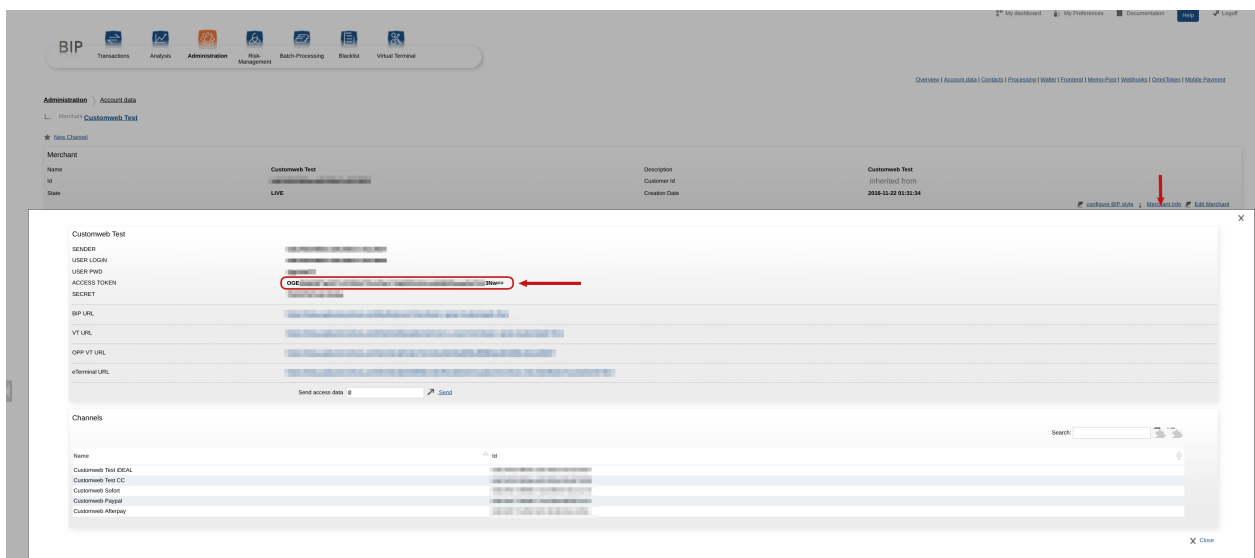


Abbildung 2.1: Die Authorization Bearer Token Information im PayUnity Backend.

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 Gambio Shop unter **Modul-Center > PayUnity > Open Main Configuration > 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 Gambio Shop unter **Modul-Center > PayUnity > Open Main Configuration > 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](#)

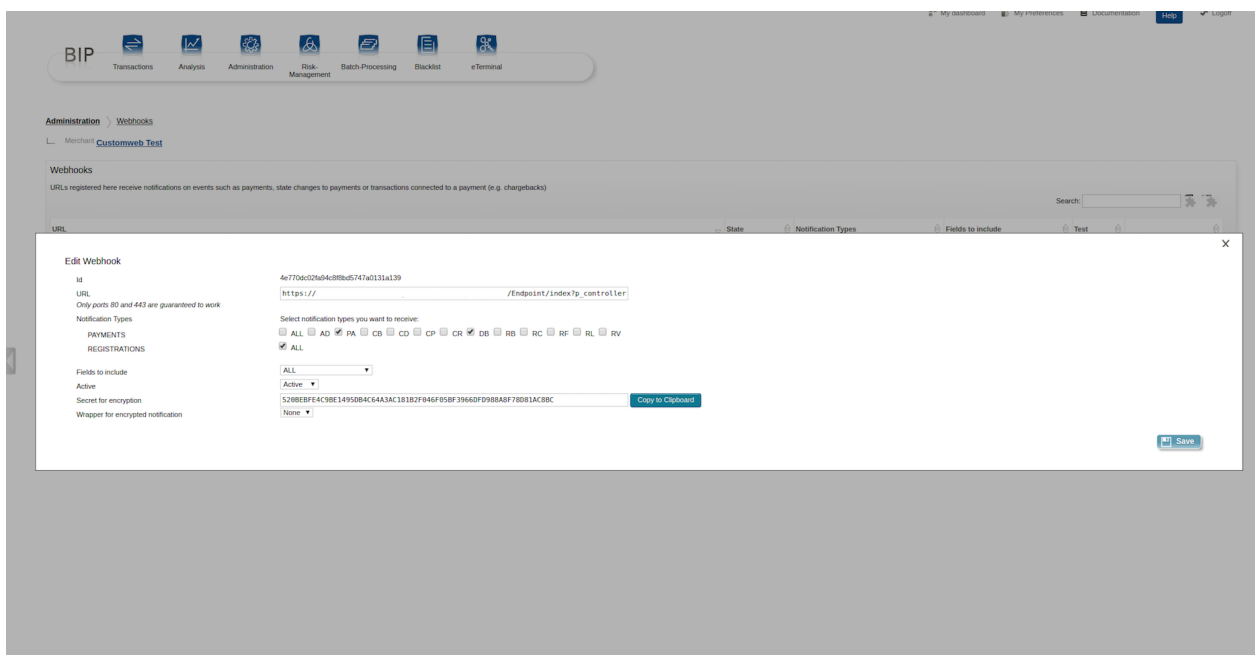


Abbildung 2.1: PayUnity Webhook Beispielkonfiguration in Ihrem Backend unter **Administration > Webhook**.

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 Gambio Shop unter **Modul-Center > PayUnity > Open Main Configuration > Setup** an PayUnity übermitteln. PayUnity wird Ihnen anschliessend ein sog. "secret for encryption" zustellen, welches Sie bitte in der Hauptmodulkofnfiguration 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 Gambio 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 Gambio 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.

3.2 Sprache

1. Navigieren Sie im entpackten Ordner zum Ordner "lang"
2. Laden Sie mit Ihrem bevorzugten FTP-Client den **gesamten Inhalt** dieses Ordners in das lang-Verzeichnis Ihres Shops hoch. Stellen Sie sicher, dass die Inhalte nicht überschrieben, sondern nur zusammengeführt werden.

3.3 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.

3.3.1 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, das 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.

3.3.2 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.

4 Modul Konfiguration im Gambio 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

Sie finden die Einstellungen des Hauptmoduls unter **"Module > Modul-Center > PayUnity Base Module"** . Installieren Sie das Modul und klicken Sie auf Tragen Sie auf "Hauptkonfiguration öffnen". Es sollte sich nun ein neues Fenster öffnen. Tragen Sie hier die einzelnen Optionen ein, welche Sie teilweise bereits oben in der Konfiguration in der Administrationsoberfläche von PayUnity festgelegt haben. Sollte Ihnen eine Option nicht klar sein, erhalten Sie durch klicken auf das Fragezeichen weitere Informationen zu den Einstellungen. Die zentralsten Einstellungen werden hier nochmals kurz erwähnt.

4.1.1 Ausstehende Bestellungen erstellen

Sie haben die Möglichkeit ausstehende Bestellungen zu erstellen. In diesem Fall wird in Ihrem Shop eine temporäre Bestellung angelegt mit dem Status "bevorstehende Zahlung". Dies ermöglicht, dass der Warenbestand in Ihrem Shop blockiert wird und an PayUnity jeweils die identische Bestellnummer wie im Shop übermittelt werden kann. Sollte der Kunde den Kauf jedoch abbrechen bleibt die Bestellung bestehen, ausser Sie aktivieren die Option abgebrochene Bestellungen entfernen. Dies führt dazu, dass die abgebrochenen Bestellungen regelmässig gelöscht werden. Wenn Sie keine ausstehenden Bestellungen erstellen möchten bleiben Sie beim Standardverhalten von xt:Commerce. In diesem Fall ist es jedoch auch nicht möglich, dass die Referenz (Zahlungsid) bei PayUnity der Bestellnummer im Shop entspricht. Es steht Ihnen jedoch frei beide Varianten zu benutzen.

4.1.2 Datenbank Codierung

Wenn Sie beim Anlegen Ihrer Datenbank nicht UTF-8 als Standard gewählt haben, kann dies zu Codierungsproblemen bei Parametern mit Umlauten führen. Wählen Sie in diesem Fall die Option, dass die Daten noch UTF-8 Codiert werden bevor diese zu PayUnity gesendet werden.

4.2 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.

4.3 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.

4.3.1 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.

4.4 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.

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 Gambio 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 Gambio möglich. Mehr Informationen dazu finden Sie im Kapitel [Transaktionsmanagement](#).

Bearbeiten von Klarna Transaktionen in Gambio

Bitte beachten Sie, dass Sie in **keinem** Fall eine Klarna Transaktion in Gambio 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 Gambio 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 Gambio 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](#).

5.3.9 Code Anpassungen für die Integration von MasterPass (Gambio GX 1.x)

Wenn Sie Gambio GX 2.x verwenden sind die folgenden Code Anpassungen für Sie nicht relevant.

Damit der Button "Zahlen mit MasterPass" beim Warenkorb für die Verwendung des External Checkouts mit Adressübergabe erscheint, müssen Sie bitte folgenden Anpassungen am Code durchführen. Wenn Sie MasterPass nur als normale Zahlungsart ohne Adressübergabe verwenden möchten, sind keine Änderungen nötig.

Sie finden im Zip in Ordner /changes/*.txt die Dateien mit den notwendigen Änderungen. Bitte kopieren Sie den Code dieser Dateien an die jeweilige Stelle wie unten beschrieben.

5.3.9.1 Anpassungen im File: shopping_cart.php

Bitte kopieren Sie den Codeschnipsel, den Sie in der Datei in /changes/shopping_cart.php.txt finden, vor folgenden Code:

```
$smarty->assign('language', $_SESSION['language']); (ca. auf Zeile 140)
```

5.3.9.2 Anpassungen im Files templates/[current-active-template]/module/shopping_cart.html

Bitte kopieren Sie den Code denn Sie in der Datei /changes/shopping_cart.html.txt finden vor die die Stelle vor dem **Tag-`{/if}`**:

5.3.10 Diverses

Die Zahlungsart MasterPass mit External Checkout kann nicht in Zusammenhang mit Gutscheinen verwendet werden. Wenn der Kunde einen Gutschein einlösen will, dann muss der Standard Checkout durchlaufen werden.

6 Das Modul in Aktion

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

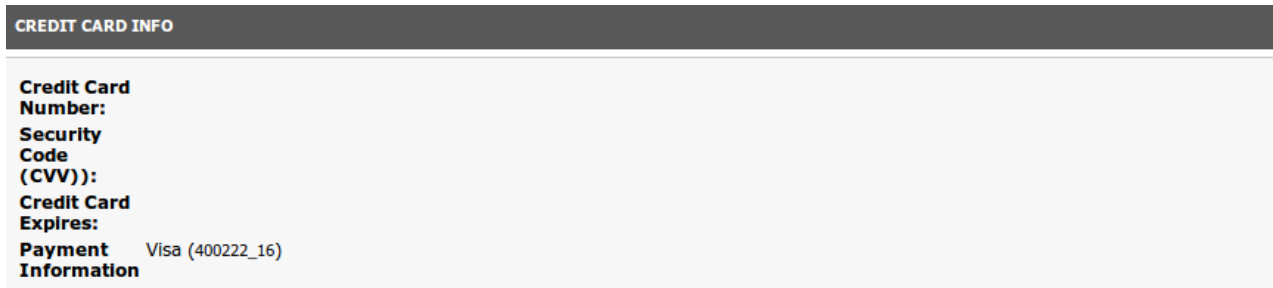


Abbildung 6.1: Transaktionsinformationen

6.1 Nützliche Transaktionsinformationen zur Bestellung

Sie finden in jeder Bestellung, die mit unserem Plugin abgewickelt wurde, eine Übersicht über die wichtigsten Informationen zur Transaktion. Unter anderem erlauben diese Informationen die eindeutige Zuordnung der Bestellungen zu den Transaktionen, welche Sie bei PayUnity im Backend sehen.

Transaktionsnummer


Transaktions ID	12
Transaktionsnummer	9_12
Bestell ID	9
Erstellt am	2013-11-26 18:03:28
Erneuert am	2013-11-26 18:03:59
Kunden ID	1
Zahlungsid	25691316
Zahlungs-Klasse	XXXXXXXXXX0003
Autorisierungsbetrag	125
Währung	EUR
Zahlungsmethode	Visa
Zahlungsid	25691316
Verbuchter Betrag	125
Transaktion autorisiert	Ja
Transaktion ist unsicher 	Nein
Transaktion verbucht	Ja
Transaktion bezahlt	Ja
Akzeptanz	test123
Kartennummer	XXXXXXXXXXXX0003
Kartenablaufdatum	06/16
Händler Referenz	9_12

Abbildung 6.1: Transaktionsinformationen.

6.2 Zahlungen im Gambio Backend auslösen

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

1. Gehen in die Kundenübersicht und aktivieren Sie einen Kunden. Durch Klick auf "Bestellung anlegen" öffnet sich ein neues Fenster zur Erfassung der Bestellung.
2. Klicken Sie auf Bearbeiten und wählen Sie Artikel und Artikeloptionen sowie Zahlungsweisen 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.

6.2.1 Rückvergüten von Bestellungen

Um Bestellungen rückzuvergüten, öffnen Sie die Transaktionsinformationen (vgl. oben). Sie erhalten Sie Informationen zur Transaktion dargestellt. Klicken Sie auf 'Rückerstattung' und wählen Sie den Artikel aus, den Sie zurückerstatten möchten. Mit einem Klick auf 'Rückerstattung' wird der Refund Request an PayUnity übermittelt.

Zurück

Teilgutschrift

Name	SKU	Typ	MwSt.	Menge	Total (exkl. MwSt.)	
Testartikel	ABC123	product	8 %	1	20.54	2
Selbstabholung (Selbstabholung der Ware in unserer Geschäftsstelle.)	shipping	shipping	0 %	1	0.00	0
					Total gutgeschriebener Betrag:	

Transaktion für weitere Rückvergütungen schliessen

Abbildung 6.1: Rückerstattung

Das Auslösen einer Gutschrift verändert den Zustand der Bestellung nicht.

6.3 Verbuchen / Stornieren von Bestellungen

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.

6.3.1 Verbuchen / stornieren von Bestellungen

Um Bestellungen zu verbuchen, öffnen Sie die Transaktionsinformationen (vgl. oben). Sie erhalten Sie Informationen zur Transaktion dargestellt. Weiter unten finden Sie ein Input Feld in welchem Sie den Betrag den Sie verbuchen möchten angeben können. Mit einem Klick auf "verbuchen" wird die Buchung an PayUnity übermittelt. Sie brauchen sich nicht mehr in das Backend von PayUnity einzuloggen.

Mit dem Klick auf "Transaktion abbuchen" erfolgt ein "Storno" der Transaktion und der reservierte Betrag wird auf der Karte Ihres Kunden sofort freigegeben.

Transaktion verbuchen

Betrag vergüten ?

Transaktion schliessen ?

 Ja, Transaktion schliessen

Transaktion abbuchen

Abbildung 6.1: capture oder cancel button.

6.4 Nutzung des Alias Managers / Token Lösung

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

ZAHLUNGSWEISE

Bitte wählen Sie die gewünschte Zahlungsweise aus.

Visa

Karten Nr: 9451 1231 0000 0111 Ablaufdatum: 12/2015 CVV: 123

Use previous stored credit
card:

xxxx xxxx xxxx 0111 ▼

Abbildung 6.1: Alias Manager

Bitte beachten Sie!

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 Einrichten Cron Job

Um die zeitgesteuerten Funktionen des Plugins zu aktivieren (bspw. Update Service, Löschen pendenteter Bestellungen, etc.), richten Sie in Ihrem Hosting einen zeitgesteuerten Request auf die Datei `payunitycw_cron.php` ein. Der regelmässige Aufruf löst die entsprechenden Aktionen aus. Wir raten zu einem generellen Intervall von ca. 15 Minuten

Wir empfehlen Ihnen hier eine Cron Engine wie beispielsweise [EasyCron](#). Damit können Sie die Datei (URL) mit einem externen Service aufrufen

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/faq>. 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 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 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 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.

10.1 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.

Initialisieren Sie Gambio in Ihrem Script, dies ist nötig um die Verbindung zur Datenbank herzustellen.

```
include ('includes/application_top.php');
```

Inkludieren Sie die Hauptdatei des Plugins sowie die Modulklassen.

```
require_once DIR_FS_DOCUMENT_ROOT.'admin/includes/PayUnityCw/init.php';
require_once DIR_FS_DOCUMENT_ROOT.'admin/includes/PayUnityCw/classes
/PayUnityCw/Entity/Util.php';
```

Nun können Sie die Transaktion laden und das Transaktionsobjekt extrahieren.

Laden der Transaktion per by Id:

```
$transactionById = PayUnityCw_Entity_Util::findTransactionByTransactionId
($transactionId);
$transactionObject = $transactionById->getTransactionObject();
```

Laden der Transaktion per external Id:

```
$transactionById = PayUnityCw_Entity_Util::
findTransactionEntityByTransactionExternalId($externalId);
$transactionObject = $transactionById->getTransactionObject();
```

Laden der Transaktion per Order ID:

```
$transactionsByOrderId = PayUnityCw_Entity_Util::
findTransactionsEntityByOrderId($orderId);
foreach($transactionsByOrderId as $transaction){
    $transactionObject = $transaction->getTransactionObject();
    //Do something with each object
}
```