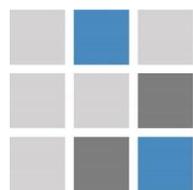


Technische Dokumentation

paydirekt



Logata
Digital Solutions

Änderungshistorie

Version	Datum	Autor	Beschreibung
0.1	21.02.2018	Dennis Mandel	Initiale Version
0.2	22.02.2018	Andreas Kemper	Layout-Anpassungen
0.3	22.02.2018	Christian Orthmann	Review
1.0	28.02.2018	Andreas Kemper	Initiale Auslieferung
1.1	13.03.2018	Dennis Mandel	Änderungen laut PAYD-416 (Statusmodell, Sicherheit, Datenhaltung)
1.2	11.04.2019	Fabian Wiltink	Anpassen des Layouts

Inhaltsverzeichnis

1. Einleitung	4
2. Shopsystem	5
2.1. Frontend	5
2.2. Backend	5
2.3. Datenhaltung	5
2.4. Sicherheit	5
3. Plug-In	6
4. Core Modul	6
5. Status Modell	6
6. Build-System	8
7. Tests	8
8. Entwicklungs-Standards	8
9. Hilfe und Support	9
9.1. Fragen/Support zum Plugin (z.B. Installation, Konfiguration, Betrieb...)	9
9.1.1. Support-Zeiten	9
9.1.2. Support Mail	9
9.1.3. Support Hotline	9
9.2. Fragen/Support zu paydirekt (z.B. Abrechnung, Händlerportal, Transaktionsanalyse, Betrugsfälle...)	9
9.2.1. Support-Zeiten	9
9.2.2. Support Mail	9
9.2.3. Support Hotline	9

1. Einleitung

Diese Technische Dokumentation richtet sich an Plug-In-Entwickler, die mit dem Core-Framework ein Zahlungs-Plug-In entwickeln oder anpassen möchten. Es gibt einen Überblick über die Entwicklungsumgebung. Beschrieben werden die Frontend/Backend Klassen, das Build-System und die eingesetzten Entwicklungsstandards.

Diese Dokumentation gilt für das Core Modul und die folgende Plug-Ins:

- Gambio GX2 – GX3
- JTL 4.0
- Magento Version 1.x.
- Magento Version 2.x.
- OpenCart 2.1
- OpenCart 2.3
- osCommerce 2.3
- osCommerce 2.4
- OXID CE & PE 4.7 – 6.x
- OXID EE 5.0 – 6.x
- PrestaShop 1.6
- PrestaShop 1.7
- shopware 5.0 – 5.2.1
- shopware 5.2.2. – 5.x
- Virtuemart 2.6 – 3.x
- WooCommerce 2.6 – 3.x
- xt:commerce Veyton 4.2
- xt:commerce Veyton 5.x

2. Shopsystem

2.1. Frontend

Die Frontend Klassen bilden die reine Logik des Bezahlprozesses ab. Hierbei gibt es, nur eine minimale Frontend-Ausgabe (z.B. zur Ausgabe eines Fehlers während des Bezahlprozesses).

Die Klassen decken folgendes ab:

- Checkout (Shop > paydirekt-API)
- Redirect Handling (Shop > paydirekt)
- Return Handling (paydirekt > Shop)
- Notify Handling (paydirekt-API > Shop)

2.2. Backend

Die Backend Klassen beinhalten neben der Logik Oberflächen Komponenten.

Die Klassen decken folgendes ab:

- Installations-/ Konfigurations-Handling
- Capture Handling
- Refund Handling
- Order Handling
- Report Handling

2.3. Datenhaltung

Falls es notwendig ist, wird eine Datenbank-Tabelle zur Speicherung von Daten angelegt werden (solange es die „best practice“ Regeln erlauben). In der Regel beinhaltet eine Order ein extra Feld zur Speicherung der Transaktions-ID. Dieses wird zur Speicherung der Checkout-ID verwendet. In einigen Fällen (z.B. Shopware) ist es notwendig, dass Temporäre Daten in einer separaten Datenbank-Tabelle gespeichert werden. Hier wird darauf geachtet werden, dass diese das Shop-System nicht beeinträchtigen.

2.4. Sicherheit

Die Plug-Ins sind Bestandteil des jeweiligen Shop-Systems und halten sich an deren „best practices“ Regeln zum Entwickeln von Plug-Ins. Für eingehende Meldungen (z.B. für die Notify-Funktion) muss min. eine IP-Adress-Validierung stattfinden, die das paydirekt System identifizieren kann.

3. Plug-In

Jedes Plug-In hat in der ausgelieferten (gebauten) Version das Core-Modul in der jeweils passenden Version integriert. Die Kommunikation zur paydirekt-API findet ausschließlich über die Core-Funktionen statt.

4. Core Modul

Das Core-Modul beinhaltet grundlegende Funktionen zur Kommunikation mit der paydirekt-API.

Bei der Entwicklung des „Core“ Systems wurde darauf geachtet, statische Funktionen zu erstellen, um den Zugriff so einfach wie möglich zu halten. Die Ausnahme bildet die „Logger“ Klasse. Diese wird mit der aufrufenden Klasse und dem eingestellten Log-Level gefüllt und bietet über ein Logger-Objekt Möglichkeiten, ein Ereignis zu loggen bzw. wieder auszulesen.

Die Grundfunktionen bilden die paydirekt-API ab und besitzen wenig bis gar keine Logik.

Zudem bietet das Core-Modul noch weitere Werkzeuge, die das Verarbeiten der Daten aus den Shop-System erleichtern sollen. Hierzu gehört neben diversen Überprüfungs- und Konvertierungsfunktionen eine Übersetzung von Meldungen der paydirekt-API in ein leserliches Format.

5. Status Modell

	1.	2.	3.
Relevanter Prozessschritt	Käufer ist im Bestellprozess	Käufer hat paydirekt im Bestellprozess ausgewählt und die Bestellung final abgesandt und wurde zu paydirekt weitergeleitet, hat dort aber noch nichts gemacht.	Käufer hat Abbrechen geklickt (in der paydirekt Zahlstrecke)
Status paydirekt	-	OPEN	CANCELLED
Notify	Nein	Ja (OPEN)	Ja (CANCELLED)
Rückleitung	Nein	Nein	Ja (Cancelled URL)
Status Backend Bestell / Zahlung	Hidden	Hidden	Hidden
Anzeige Backend	Nein	Nein	Nein
Status Frontend	Käufer hat Produkte im Warenkorb	Nicht relevant, Käufer ist nicht mehr im Shop	Warenkorb gefüllt, Meldung wird angezeigt

	4.	5.	6.
Relevanter Prozessschritt	Käufer hat Zahlung bestätigt und diese ist genehmigt (in der paydirekt Zahlstrecke)	Käufer wurde der Directsale (Zahlung) oder die ORDER (Vormerkung) nicht genehmigt (in der paydirekt Zahlstrecke)	Käufer hat den Warenkorb abandoned (Browser geschlossen bei zuletzt geöffneter paydirekt Zahlstrecke)
Status paydirekt	APPROVED	REJECTED	EXPIRED
Notify	Ja (APPROVED)	Ja (REJECTED)	Ja (EXPIRED)
Rückleitung	Ja (Success URL)	Ja (Rejected URL)	Nein
Status Backend Bestell / Zahlung	Offen / Komplette bezahlt	Hidden	Hidden
Anzeige Backend	Ja	Nein	Nein
Status Frontend	„Danke“ Seite / Erfolgreiche Bestellung	Käufer wird auf Warenkorb zurückgeleitet und erhält eine kurze Info	Wenn Shop unterstützt, sollte der Warenkorb noch gefüllt sein (Shop anhängig)

	7.	8.	
Relevanter Prozessschritt	Käufer ist mit Zurück-Button des Browsers(!) in den Shop zurückgekommen (von der paydirekt Zahlstrecke)	Zahlung wurde wegen fehlgeschlagener Altersprüfung von paydirekt nicht genehmigt.	
Status paydirekt	EXPIRED	REJECTED	
Notify	Ja (EXPIRED)	Ja (REJECTED)	
Rückleitung	Nein	Ja (AgeVerification URL)	
Status Backend Bestell / Zahlung	Hidden	Hidden	
Anzeige Backend	Nein	Nein	
Status Frontend	Käufer hat Produkte im Warenkorb	Käufer wird auf Warenkorb zurückgeleitet und erhält eine kurze Info	

6. Build-System

Das Build-System ist ein ANT-derivat namens Phing (<https://www.phing.info/>). Dieses wird zum Zusammenstellen der auslieferbaren Version des jeweiligen Systems verwendet.

Folgende Einstellungen sind pro Plug-In in der build.xml vorzunehmen:

Property name	Value Beschreibung
TAG: xmlproperty	Pfad zur plugin.xml
core.commit.id	Git commit ID der zugehörigen Core Version
build.dist.special.folder	Relativer Pfad, in den der Sourcecode kopiert werden soll (einige Systeme benötigen einen speziellen Pfad)
build.src.lib.folder	Relativer Pfad zum Lib Verzeichnis
build.compress.filename.prefix	Dateinamens-Präfix. Bsp.: shopware-5.2.2-5.3.4-
build.zip	Wahrheitswert, ob eine .zip Datei erstellt werden
build.tgz	Wahrheitswert, ob eine .tgz Datei erstellt werden
shopsystem.id	Name des Shop-Systems
shopsystem.version	Version des Shop-Systems (wird verwendet zum Ersetzen der Version)
shopsystem.tag	Kurzform des Shopsystems inkl. Version Bsp.: sw522

7. Tests

Die Shop-Systeme sind in der Regel nicht automatisiert testbar, da es sich um integrierte Plug-Ins handelt. Tests beschränken sich hier auf manuelle Tests.

Das Core Modul beinhaltet PHP-Unit Tests, die durch den Build-Server ausgeführt werden. Jede Anpassung am Core muss auch in einem Test abgebildet sein.

8. Entwicklungs-Standards

Da es sich bei den Entwicklungen um Shop-Plug-Ins handelt, hält sich die Programmierung an den vorgegebenen Shop-System „Best Practices“. Die eingesetzte Programmiersprache ist PHP. Ob ein objektorientierter Ansatz gewählt werden kann, hängt von dem jeweiligen Shop-Framework ab. Manch ältere Systeme erlauben nur eine eingeschränkte, rein funktionale Programmierung.

Die Dokumentation der einzelnen Funktionen sind im PHPDoc Format als Kommentar über der jeweiligen Funktion beschrieben.

9. Hilfe und Support

9.1. Fragen/Support zum Plugin (z.B. Installation, Konfiguration, Betrieb...)

Bei allen technischen Fragen rund um das Plugin steht Ihnen Fulino jederzeit gerne, nach Maßgabe der Support-Bedingungen, kostenlos zur Verfügung:

9.1.1. Support-Zeiten

Montag – Freitag (außer an gesetzlichen Feiertagen, sowie jährlich dem 24.12. und 31.12.):

8:00 Uhr bis 18:00 Uhr.

9.1.2. Support Mail

Paydirekt-info@logata.com

9.1.3. Support Hotline

+49 (0) 2871 2346 291

9.2. Fragen/Support zu paydirekt (z.B. Abrechnung, Händlerportal, Transaktionsanalyse, Betrugsfälle...)

Bei allen Fragen zu paydirekt steht Ihnen jederzeit gerne der paydirekt Händler-Support kostenlos zur Verfügung:

9.2.1. Support-Zeiten

Montag – Freitag (außer an gesetzlichen Feiertagen, sowie jährlich dem 24.12. und 31.12.):

8:00 Uhr bis 18:00 Uhr.

9.2.2. Support Mail

haendler@paydirekt.de

9.2.3. Support Hotline

+49 (0)69 2475382223