



Der Weg durch die Technik

Technische Modul-Dokumentation

Der wohl größte Vorteil der eCommerce-Plattform Magento liegt in seiner offenen und erweiterbaren Struktur. Module der Modulwerft können und sollen nicht alle erdenklichen Funktionen abdecken, aber sie sollen einfach und ohne Probleme für Magento-Entwickler erweiterbar sein.

Mit Hilfe dieser technischen Modul-Dokumentation sollen erfahrene Magento-Entwickler weitere Funktionalitäten einfach und schnell hinzufügen können, ohne die Update-Fähigkeit des Moduls zu beeinträchtigen.

Kompetenz, Leidenschaft, Zuverlässigkeit – gemeinsam über die Weltmeere!

Gutschein-Restwerte

Einleitung

Diese technische Modul-Dokumentation wendet sich an Magento-Entwickler, die mit den gängigen Programmier-Techniken in Magento vertraut sind. Hierzu zählen allgemein die Anpassung des Systems in Form von Modulen, der Magento-Rewrite-Mechanismus sowie das Event-Observer-Muster.

Die vorliegende Dokumentation beschreibt, wie das vorliegende Modul auf korrekte Art und Weise erweitert werden kann.

Release Notes

Datum	Modul-Version	Änderungen
29.04.2014	1.2.0	+ Erste Version dieser Dokumentation + Absatz „Allgemeiner Modulaufbau“ + Absatz „Datenbank-Struktur“ + Absatz „Neue Events“ + Absatz „Benutzte Events“ + Absatz „Frontend-Integration“ + Absatz „Erweiterbarkeit“ + Absatz „Unterstützung durch die Modulwerft“

Allgemeiner Modul-Aufbau

Das Modul integriert seine Funktionalität vollständig über Events in das Magento-Grundsystem. Auf Rewrites von Core-Klassen wurde zugunsten der Updatefähigkeit verzichtet. Die benutzten Events werden in einem der nachfolgenden Absätze genauer beschrieben.

Vom Modul gespeicherte Daten werden in der Datenbank abgelegt. Logging-Informationen werden in der Datei **`/var/log/Modulwerft_CouponRemainingValue.log`** hinterlegt.

Das Modul ist ausführlich über das Magento-Backend konfigurierbar. Details hierzu finden Sie in der Konfigurationsanleitung.

Aufgabe des Moduls ist es, Gutscheine, die nicht vollständig eingelöst wurden, weitere Male einlösbar zu halten und den Restwert korrekt zu berechnen und zur Verfügung zu stellen. Hierfür wird an diversen Stellen per Observer in das System eingegriffen, um auch bei Stornierungen und Gutschriften ein korrektes Systemverhalten zu garantieren.

Datenbank-Struktur

In der Magento-Datenbank werden insgesamt drei Tabellen angelegt:

modulwerft_couponremainingvalue_coupon

In dieser Tabelle werden die Restwerte zu den konkreten Gutschein-Codes (Coupons) abgelegt. Gespeichert werden der ursprüngliche und der derzeit noch verbleibende Wert.

modulwerft_couponremainingvalue_history

In dieser Tabelle wird der Verlauf der Gutschein-codes gespeichert. Es wird genau dokumentiert, welcher Coupon im Rahmen welcher Bestellung verringert oder erhöht wurde (z.B. Gutschrift). Die Listen in den Warenkorb-Preisregeln basieren auf dieser Tabelle.

modulwerft_couponremainingvalue_rule

In dieser Tabelle wird die Konfiguration der Regeln gespeichert. Wird eine Warenkorb-Preisregel als Restwert-Regel definiert, wird diese Information in dieser Tabelle abgelegt.

Neue Events

Das Modul erstellt ein paar Events, die sowohl vom Modul selbst benutzt werden als auch für die Anpassung und Erweiterung des Moduls genutzt werden können.

Event:	modulwerft_couponremainingvalue_coupon_service_reset_coupon_value
Kontext:	Der Gutschein wurde erfolgreich nach einer Stornierung oder Gutschrift hochgesetzt
Parameter:	<p>rule (Mage_SalesRule_Model_Rule): Rule-Objekt des betroffenen Coupons</p> <p>coupon (Modulwerft_CouponRemainingValue_Model_Coupon): CRV-Daten des betroffenen Coupons</p> <p>coupon_history (Modulwerft_CouponRemainingValue_Model_History): History-Model des CRV-Moduls</p> <p>order (Mage_Sales_Model_Order): Betroffene Bestellung</p> <p>creditmemo (Mage_Sales_Model_Order_Creditmemo): Betroffene Gutschrift (bei Stornierung nicht vorhanden)</p>

Event:	modulwerft_couponremainingvalue_coupon_service_used_coupon_value
Kontext:	Der Gutschein wurde erfolgreich eingelöst und der Restwert berechnet
Parameter:	<p>rule (Mage_SalesRule_Model_Rule): Rule-Objekt des betroffenen Coupons</p> <p>coupon (Modulwerft_CouponRemainingValue_Model_Coupon): CRV-Daten des betroffenen Coupons</p> <p>coupon_history (Modulwerft_CouponRemainingValue_Model_History): History-Model des CRV-Moduls</p> <p>order (Mage_Sales_Model_Order): Betroffene Bestellung</p>

Event:	modulwerft_couponremainingvalue_coupon_service_redeemed_coupon
Kontext:	Der Gutschein wurde erfolgreich eingelöst, der Restwert berechnet und die zugehörige Bestellung ist in den State In Verarbeitung (Processing) gegangen
Parameter:	<p>rule (Mage_SalesRule_Model_Rule): Rule-Objekt des betroffenen Coupons</p> <p>coupon (Modulwerft_CouponRemainingValue_Model_Coupon): CRV-Daten des betroffenen Coupons</p> <p>coupon_history (Modulwerft_CouponRemainingValue_Model_History): History-Model des CRV-Moduls</p> <p>order (Mage_Sales_Model_Order): Betroffene Bestellung</p>

Event:	modulwerft_couponremainingvalue_notification_send_before
Kontext:	Ereignis kurz bevor der Kunde per E-Mail über den neuen Restwert seines Gutscheines informiert wird
Parameter:	<p>order (Mage_Sales_Model_Order): Betroffene Bestellung</p> <p>rule (Mage_SalesRule_Model_Rule): Rule-Objekt des betroffenen Coupons</p> <p>coupon_history (Modulwerft_CouponRemainingValue_Model_History): History-Model des CRV-Moduls</p> <p>mail (Mage_Core_Model_Email_Template): Objekt des E-Mail-Templates, das verschickt werden soll</p> <p>creditmemo (Mage_Sales_Model_Order_Creditmemo): Betroffene Gutschrift (bei Stornierung nicht vorhanden)</p>

Event:	modulwerft_couponremainingvalue_notification_send_after
Kontext:	Ereignis kurz nachdem der Kunde per E-Mail über den neuen Restwert seines Gutscheines erfolgreich informiert wurde
Parameter:	<p>order (Mage_Sales_Model_Order): Betroffene Bestellung</p> <p>rule (Mage_SalesRule_Model_Rule): Rule-Objekt des betroffenen Coupons</p> <p>coupon_history (Modulwerft_CouponRemainingValue_Model_History): History-Model des CRV-Moduls</p> <p>mail (Mage_Core_Model_Email_Template): Objekt des E-Mail-Templates, das verschickt werden soll</p> <p>creditmemo (Mage_Sales_Model_Order_Creditmemo): Betroffene Gutschrift (bei Stornierung nicht vorhanden)</p>

Benutzte Events

Event:	sales_order_place_after
Event-Typ:	Core-Event (Sales-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	salesOrderPlaceAfter
Aufgabe:	Berechnung des Gutschein-Restwertes nach Eingang einer neuen Bestellung

Event:	salesrule_rule_save_before
Event-Typ:	Core-Event (Salesrule-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	ruleSaveBefore
Aufgabe:	Zurücksetzen der Limits Verwendungen pro Rabattcode (UsesPerCoupon) und Verwendungen pro Kunde (UsesPerCustomer) vor dem Speichern einer Warenkorb-Preisregel, um Mehrfach-Einlösung zu ermöglichen

Event:	salesrule_rule_save_after
Event-Typ:	Core-Event (Salesrule-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	ruleSaveAfter
Aufgabe:	Speicherung beim Speichern einer Warenkorb-Preisregel, ob die Restwert-Funktionalität in der Warenkorb-Preisregel aktiviert wurde, in der Modul-Datenbank-Tabelle

Event:	salesrule_rule_load_after
Event-Typ:	Core-Event (Salesrule-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	ruleLoadAfter
Aufgabe:	Laden der Einstellung zur Aktivierung der Restwert-Funktionalität beim Laden einer Warenkorb-Preisregel

Event:	sales_order_save_after
Event-Typ:	Core-Event (Sales-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	redeemCoupon
Aufgabe:	Versand einer E-Mail beim Speichern einer Bestellung, falls diese einen Restwert-Gutschein enthält und im State auf Verarbeitung (Processing) umspringt

Event:	sales_order_creditmemo_refund
Event-Typ:	Core-Event (Sales-Modul)
Observer-Klasse:	Modulwert_CouponRemainingValue_Model_Observer
Observer-Methode:	creditmemoRefundResetCouponValue
Aufgabe:	Aufbuchung des eingesetzten Gutscheinbetrages auf den Gutscheincode beim Erstellen einer Gutschrift

Event:	sales_quote_save_before
Event-Typ:	Core-Event (Sales-Modul)
Observer-Klasse:	Modulwert_CouponRemainingValue_Model_Observer
Observer-Methode:	validateRemainingValueOfCoupon
Aufgabe:	Überprüfung im Warenkorb, ob noch ein Restwert auf dem eingelösten Coupon vorhanden ist

Event:	order_cancel_after
Event-Typ:	Core-Event (Sales-Modul)
Observer-Klasse:	Modulwert_CouponRemainingValue_Model_Observer
Observer-Methode:	orderCancelResetCouponValue
Aufgabe:	Aufbuchung des eingesetzten Gutscheinbetrages auf den Gutscheincode beim Stornieren einer Bestellung

Event:	core_collection_abstract_load_before
Event-Typ:	Core-Event (Core-Modul)
Observer-Klasse:	Modulwert_CouponRemainingValue_Model_Observer
Observer-Methode:	addUseRemainingValueFlag
Aufgabe:	Laden der Konfiguration für Restwert-Funktionalität auch in Rule-Collections

Event:	core_collection_abstract_load_after
Event-Typ:	Core-Event (Core-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	applyCouponValueToRule
Aufgabe:	Anwenden des Gutschein-Restwertes im Warenkorb anstelle des ursprünglichen Gutscheinwertes der Warenkorb-Preisregel

Event:	core_abstract_save_commit_after
Event-Typ:	Core-Event (Core-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	couponSaveAfter
Aufgabe:	Übertragen von Daten aus der Warenkorb-Preisregel in die Modul-Datenbank-Tabellen beim Speichern der Regel

Event:	modulwerft_couponremainingvalue_coupon_service_redeemed_coupon
Event-Typ:	Modul-Event (CouponRemainingValue-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	couponSendNotificationUsedValue
Aufgabe:	Versand der E-Mail an den Kunden mit Restwert-Informationen

Event:	modulwerft_couponremainingvalue_coupon_service_reset_coupon_value
Event-Typ:	Modul-Event (CouponRemainingValue-Modul)
Observer-Klasse:	Modulwerft_CouponRemainingValue_Model_Observer
Observer-Methode:	couponSendNotificationResetValue
Aufgabe:	Versand der E-Mail an den Kunden mit Information über Aufbuchung des Gutschein-Restwertes nach Stornierung oder Gutschrift

Frontend-Integration

Der Kunde erhält im Frontend eine direkte Rückmeldung nur über verschickte Benachrichtigungsmails, sofern dies entsprechend der Konfigurationsanleitung eingerichtet wurde. Eine weitere Integration, etwa im Kundenmenü oder im Checkout, findet derzeit nicht statt.

Erweiterbarkeit

Das Modul lässt sich an den unterschiedlichsten Stellen erweitern. Wir empfehlen den Verzicht auf Class-Rewrites, um das Modul updatefähig zu halten. Weitere Events können wir in weiteren Modul-Versionen auf Wunsch gerne integrieren.

Unterstützung durch die Modulwerft

Sollten Sie Fragen zur Erweiterbarkeit des Moduls haben, melden Sie sich gerne per E-Mail bei unserem Support unter support@modulwerft.com.

Vielen Dank, dass Sie sich für ein Modul der Modulwerft entschieden haben. Sie haben damit ein Qualitätsprodukt erworben. Sollten Sie einmal nicht zufrieden sein oder haben Ideen für Erweiterungen oder Verbesserungen, würden wir uns über eine E-Mail an support@modulwerft.com freuen.