Quickstart API CreditCard v1.0

G. Franke, Micropayment™

Juli 2016

Contents

Einleitung

Für Zahlungen per Kreditkarte benötigen Sie einen Account bei Micropayment und ein eingerichtetes Project. Um die API nutzen zu können, benötigen Sie auch einen accesskey, siehe controlcenter.

Eine Zahlung per Kreditkarte kann auf 2 unterschiedlichen Wegen erfolgen. Wir unterscheiden Folgendes:

- Purchase
  - die Zahlung wird sofort durchgeführt.
- Authorization / Capture
  - die Zahlung erfolgt in 2 Schritten. Mit Authorization wird ein Betrag für einen gewissen Zeitraum reserviert, d.h. das Kreditlimit des Kunden wird um diesen Betrag reduziert. Der reservierte oder ein kleinerer Betrag wird dann zu einem späteren Zeitpunkt mit Capture gebucht.

Bei der Kreditkartenzahlung unterscheiden wir weiterhin zwischen der Einmal- bzw. Erstzahlung und möglichen Folgezahlungen nach einer Erstzahlung. Speziell für visa/master kann eine Zahlung als “Recurring Billing” geflaggt werden. Dies erfolgt bei den betreffenden Funktionen über den Parameter “recurring”. Der account muß für die Nutzung von “recurring” separat freigeschaltet werden.

Grundsätzlich erfordert jede Zahlung (Purchase / Authorization) den cvc Code. Bei Folgezahlungen nach einer erfolgreichen Erstzahlung kann auf den cvc Code verzichtet werden.

Kreditkartenzahlungen laufen bei Micropayment wie folgt ab (siehe API-Funktionen):

Ablauf Erstzahlung:

1. Customer erzeugen. Beschrieben mit customerId.
   - **customerCreate**
2. Kreditkartendaten dem Customer zuweisen
   - **creditcardDataSet**
3. optionale Daten dem Customer zuweisen (Anschrift etc.)
   - **addressSet**
4. Session erzeugen. wichtige Parameter: customerId, amount, currency
   - **sessionCreate**
5. Transaction erzeugen in Form einer Zahlung per Purchase oder Authorization/Capture. Grundlage ist die erzeugte Session, referenziert über die sessionId. Zu einer Session kann es also mehrere Transaktionen geben (z.B. Authorization und Capture)
   - **transactionPurchase**
   - **transactionAuthorization**
   - **transactionCapture**

Kreditkarteninformationen werden immer über den Customer referenziert. Somit ist für jede

Buchung auch ein gültiger Customer notwendig, referenziert über die customerId. Für eine Folgezahlung ohne Eingabe des cvc Codes benötigen Sie einen Customer, mit dem bereits eine erfolgreiche Buchung gemacht wurde.

Ablauf Folgezahlung:

1. Session erzeugen.
    - **sessionCreate**
2. Zahlung per Purchase oder Authorization/Capture.
    - **transactionPurchase**
    - **transactionAuthorization**
    - **transactionCapture**

Wenn Kreditkartendaten durch Scripte des Händlers erfaßt/bearbeitet werden sollen, ist eine entsprechende PCI Zertifizierung erforderlich. Um dies zu umgehen, stellt Micropayment eine IFramelösung in Verbindung mit einer Tokenisation der Kreditkartendaten bereit. Das Ganze ist als JsBridge implementiert und seperat dokumentiert. Bei der Tokenisation werden die Kreditkartendaten einmalig entgegengenommen und dann durch einen generierten temporären Token, d.h. einer Zeichenkette, referenziert. Dieses Token wird bei einigen Funktionen als Referenz auf die Kreditkarten benötigt. Eine dauerhafte Speicherung ist nicht vorgesehen, den der Token dient letztendlich nur dazu, einem Customer Kreditkartendaten zuzuweisen. Durch Verwendung der JsBridge kommen die Händlerscripte nie unmittelbar mit pan und cvc in Berührung.

Funktionen die nur mit PCI Zertifizierung genutzt werden können:

- **creditcardCheckValidity**
- **creditcardDataSet**
- **shortCustomerCreate**
- **shortSessionCreate**
- **shortTransactionAuthorization**
- **shortTransactionPurchase**

Funktionen die nur bei der Tokenisation mit der JsBridge verwendet werden können da sie den Token oder eine verschlüsselte Transaktonsnummer (siehe cryptId) als Parameter erwarten die ausschließlich bei der JsBridge enstehen:

- **tokenAuthorization**.
- **tokenCustomerCreate**.
- **tokenPurchase**.
- **tokenSessionCreate**.

- **transactionGetCrypt**.

Hilfsfunktionen

Neben den bereits genannten Funktionen, stellt die API noch diverse Helper zur Verfügung. Einige erkennt man bereits am Prefix. Das sind folgende:

1. **short-Prefix**

   Hier werden Aggregate bestehender Funktionen bereitgestellt. Diese dienen in erster Linie der Vereinfachung. Da aber einiges an Netzwerkoverhead entfällt, kommt es auch zu einer Beschleunigung des Zahlprozesses.

   - **shortCustomerAuthorization**.

     Kombiniert die Funktionen ‘sessionCreate’ und ‘transactionAuthorization’. Benötigt einen Customer.

   - **shortCustomerCreate**.

     Kombiniert die Funktionen ‘customerCreate’ und ‘creditCardDataSet’.

   - **shortCustomerPurchase**.

     Kombiniert die Funktionen ‘sessionCreate’ und ‘transactionPurchase’. Benötigt einen Customer.

   - **shortSessionCreate**.

     Kombiniert die Funktionen ‘customerCreate’, ‘creditCardDataSet’ und ‘sessionCreate’.

   - **shortTransactionAuthorization**.

     Kombiniert die Funktionen ‘customerCreate’, ‘creditCardDataSet’, ‘sessionCreate’ und ‘transactionAuthorization’.

   - **shortTransactionPurchase**.

     Kombiniert die Funktionen ‘customerCreate’, ‘creditCardDataSet’, ‘sessionCreate’ und ‘transactionPurchase’.

2. **token-Prefix**

   Funktionen die man bei der Tokenisation verwenden kann. Grundsätzlich wird der Token als Parameter erwartet, der bei der Tokenisation mit der JsBridge erzeugt wurde. Wie die short-Prefix Funktionen stellen sie Zusammenfassungen einfacher Funktionsaufrufe bereit.

   - **tokenAuthorization**.

     Kombiniert die Funktionen ‘customerCreate’, ‘creditCardDataSet’, ‘sessionCreate’ und ‘transactionAuthorization’.

   - **tokenCustomerCreate**.

     Entspricht der Kombination aus ‘customerCreate’ und ‘creditCardDataSet’, wobei die benötigten Daten aus dem token generiert werden.

- **tokenPurchase**.

  Kombiniert die Funktionen ‘customerCreate’, ‘creditCardDataSet’, ‘sessionCreate’ und ‘transactionPurchase’.

- **tokenSessionCreate**.

  Kombiniert die Funktionen ‘tokenCustomerCreate’, und ‘sessionCreate’

3. **Doublet**

   Funktionen die die Mehrfache Nutzung von Kreditkartennummern kontrollieren. Damit kann man prüfen oder verhindern, dass eine Kreditkartennummer von mehreren Customern genutzt wird.

   - **customerDoubletList**

   Liefert Liste mit Kunden die die gleiche Kreditkarte nutzen wie der zu prüfende Kunde.

   - **customerList**

   Liefert Liste mit bestehenden Kunden. Mit dem Parameter “onlyDoublet” kann man festlegen, dass nur Customer gelistet werden, die eine Kreditkarte verwenden, die auch noch bei einem anderen Customer verwendet wird. Bei Abfragen bitte immer eine sinnvolle Begrenzung des Ergebnisses durch geeignete Parameterwahl vornehmen (customerId, onlyNewer, from, to, limit).

   - **shortCustomerCreate**.

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **shortSessionCreate**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **shortTransactionAuthorization**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **shortTransactionPurchase**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **tokenAuthorization**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **tokenCustomerCreate**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **tokenPurchase**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

   - **tokenSessionCreate**

   Dubletten bei Erstellung erlauben/ablehnen. Parameter “doublet”.

Appendix

doublet

Parameter “doublet”. Wenn 1 sind Dubletten erlaubt, d.h. eine Kreditkartennummer kann von mehreren Customern verwendet werden. Wenn 0, dann scheitert die Funktion, siehe code 4105. Default ist 1.

recurring

Wiederholende Zahlungen, typischerweise ein ABO. Betrag und Zeitintervall können variieren. Der gleichlautende Parameter bei einigen Funktionen kann eine Transaktion als “Recurring Billing” bei visa/master markieren. Der account muß jedoch dafür freigeschaltet sein. Sie können natürlich auch ohne seperate markierung Folgebuchungen mit einem bestehenden customer machen.

API-Funktion

damit sind Funktionen der Kreditkarten API gemeint. Tokenfunktionen sind ab Version 1.6 verfügbar https://sipg.micropayment.de/public/creditcard/v1.6/nvp/ .

Weiter Einzelheiten sind unter https://techdoc.micropayment.de/ verlinkt.

Tokenisation

Damit ist der Vorgang der Tokenerstellung gemeint. Der Token repräsentiert die erfaßten Kartendaten und ist eine bestimmte Zeit gültig (z.Z. 2 Stunden). In dieser Zeit müssen die Daten weiterverarbeitet werden. Die Verarbeitung des Token kann auf mehrere Arten erfolgen. Zum einen stehen die Funktionen der API zur Verfügung, vor allem die Funktionen mit dem Prefix “token”, z.B. tokenCustomerCreate, zum anderen stellt das JsBridge eine Möglichkeit zur direkten Buchung bereit (Billingfunktionen). Momentan ist eine 3DSecurebuchung nur über die Billingfunktionen des JsBridgees möglich.

3DSecure

3-D Secure ist ein Verfahren, das für zusätzliche Sicherheit bei Online-Kreditkartentransaktionen eingesetzt wird. Es steht stellvertretend für die dementsprechenden Dienste von VISA und MASTER.

IFramelösung

siehe JsBridge

JsBridge

Script und IFrame gehostet bei Micropayment. Mindestens pan und cvc werden hiermit erfaßt, wobei für jedes Feld ein eigenes IFrame generiert wird. Das Javascript stellt beim Einbinden das Object “Micropayment” bereit. Dies stellt zur IFramesteuerung einige Funktionen, Events und Parameter bereit.

cryptId

An die Rücksprungadresse nach der Tokenisation wird die verschlüsselte Transaktionsnummer als GET-Parameter cryptId gehängt. Die wird für die API-Funktion transactionGetCrypt() benötigt. Die erste Stelle der cryptId gibt den Testmodus an. Wenn cryptId mit 1 beginnt wurde cryptId im Testmodus erzeugt, bei 0 war der Testmodus aus. Diese Stelle der cryptId können Sie auswerten, wenn Sie in der Rücksprungadresse den Testmodus nicht extra angeben wollen. Für die API-Funktion transactionGetCrypt() wird der richtige Wert für Testmodus erwartet.

Account

Anmeldung bzw. Registrierung bei Micropayment erforderlich. siehe controlcenter.

controlcenter

Verwaltungsprogramm wenn Sie sich unter https://www.micropayment.de/ einloggen mit ihrem Account.

accesskey

accesskey des Projectaccounts. Sie finden ihn unter: controlcenter: Setup/Overview ->AccessKey Achtung! Aus Sicherheitsgründen darf der accesskey nicht mit javascript verwendet werden sondern nur auf Serverseite.

Testmodus

Wenn aktiviert, wird eine eventuelle Buchung nur simuliert, also keine reale Buchung ausgelößt. Die resultierende Transaktion erscheint nicht in der Statistik. Bei einer Notification ist testmode ebenfalls entsprechend gesetzt. Für das Project muß im controlcenter der Testmodus aktiviert sein.

Project

Muss über das controlcenter für die Zahlart eingerichtet werden. Der Testmodus kann sofort benutzt werden, muss aber aktiviert werden.

Customer

Datensatz der den Kunden representiert. Neben Kreditkartendaten kann er noch Informationen wie email oder Anschrift haben.

customerId

Mit der customerId wird der Customer referenziert und damit indirekt die Kreditkartendaten.

Purchase

Sorfortige Buchung bei Zahlung mit Kreditkarte.

Authorization

Ein bestimmter Betrag wird reserviert, d.h. der Kreditrahmen des Kunden wird um diesen Betrag reduziert. Der reservierte oder ein kleinerer Betrag kann dann mit Capture zu einem späteren Zeitpunkt gebucht werden.

Capture

Ein (vor)autorisierter Betrag, siehe Authorization, wird gebucht.

pan

die Kreditkartennummer.

cvc

steht für Card Validation Code und tritt auch noch unter anderen Bezeichnungen auf. Gemeint ist die auf der Kreditkarte aufgedruckte Prüfnummer.

holder

steht für credit card holder, also der Kreditkarteninhaber

Session

Datensatz der den Zahlvorgang beschreibt, enthält Informationen über den zu buchenden Betrag, Währung und sonstiger Informationen nebst einer Refernz auf den Customer. Einer Transaktion können mehrere Transaktonen je nach Art zugeordnet sein, z.B. Authorization, Capture und dann ein Refund. Referenziert über sessionId.

Transaction

beschreibt den konkreten Vorgang oder die Vorgänge einer Session, also Purchase, Authorization, Capture etc.. Je Vorgang wird ein Transaktonssatz erzeugt. Referenziert über transactionId.

transactionId

siehe Transaction.

sessionId

siehe Session.

title

bei allen Zahlprozessen über Micropayment kann mit diesem Parameter ein eindeutiger Bezeichner für den Vorgang mitgegeben werden. Der Wert erscheint in der Statistik, wird für die Sesession-suche indiziert.