cardlink

Cardlink Payment Gateway
Redirect Model

Implementation Specifications

Contents

Introduction

This document briefly describes the Cardlink Payment Gateway Redirect Model specifications.

Overview

Cardlink Payment Gateway is a payment application that is designed for processing merchant payments in Electronic Commerce environment. The inputs to Cardlink Payment Gateway are requests from the merchant shopping solution and from there the payment process is controlled by Cardlink Payment Gateway until the payment has completed successfully or has failed and the information will be sent back to the merchant shopping solution.

The Cardlink Payment Gateway core design enables multiple types of merchant interfaces to be implemented. The Redirect Model default interface and MPI (Merchant Plug in) integrated version is provided in the next pages, for reference. Merchants can easily attach their look and feel to payment pages by supplying their own custom CSS stylesheet.

Cardlink Payment Gateway interface for merchant shop

Communication between the Cardlink Payment Gateway and the Merchant shopping cart software can be implemented via HTTP post protocol following the specifications below.

Merchant shop request to initiate payment with Cardlink Payment Gateway

The Merchant shopping cart sends to Cardlink e-Commerce via http POST the appropriate HTML code with the relative information of the transaction that starts. The following table describes the parameters of the POST from the payment page to Cardlink Payment Gateway.

Counter	Field (HTTP POST parameter)	Required / Optional	Description
1.	version	R	Value 2
2.	mid	R	Merchant id supplied (integer number) will be supplied to merchant, max length 30
3.	lang	O	Language selection for payment page (ISO 639-1 language code en, fi, sv...)
4.	deviceCategory	O	Optional user device category (default 0 www browser, 1 mobile browser)
5.	orderid	R	Merchant shop order id provided by merchant shop max length 50 chars (string 1..50 – only letters and numbers are accepted with no any space between them)
6.	orderDesc	R	Order description text (string 1..128 – special characters are accepted)
7.	orderAmount	R	Order amount (decimal number >0.0) max length 15 with decimal point
8.	currency	R	Order amount currency (string 3 ISO ISO 4217 alphabetic code (EUR, USD))
9.	payerEmail	O	Order payer email address (string 1..64)
10.	payerPhone	O	Order payer phone number, optional but strongly recommended (string..30)
11.	billCountry	R	Billing address country code (string 2 ISO 3166-1-alpha-2 code (US, FI, GB))
12.	billState	R	Billing address state (string..50)
13.	billZip	R	Billing address zip code (string..16)

14.	billCity	R	Billing address city (string..64)
15.	billAddress	R	Billing address street (string..100)
16.	weight	O	Order shipping weight (kg) if item is shippable and shipping needs to be calculated by Cardlink Payment Gateway (decimal number >0) max length 12 with decimal point)
17.	dimensions	O	Order shipping dimensions (mm) in format width:height:depth for example a box 200:200:200 (string..25) can be used for shipping calculation if implemented so
18.	shipCountry	O/R	Shipping address country code (string 2 ISO 3166-1-alpha-2 code (US, FI, GB)) Optional, required when parameter weight or dimensions are present.
19.	shipState	O/R	Shipping address state (string..50) Optional, required when parameter weight or dimensions are present.
20.	shipZip	O/R	Shipping address zip code (string..16) Optional, required when parameter weight or dimensions are present. Optional, required when parameter weight or dimensions are present
21.	shipCity	O/R	Shipping address city (string..64) Optional, required when parameter weight or dimensions are present.
22.	shipAddress	O/R	Shipping address street (string..100) Optional, required when parameter weight or dimensions are present.
23.	addFraudScore	O	Incoming starting risk score (integer) max length 12
24.	maxPayRetries	O	Maximum payment retries allowed before user is sent back to merchant, overrides specific profile setting if present (integer) max length 2
25.	reject3dsU	O	Should 3-D Secure return U status, merchant has option of continuing the transaction without liability shift or reject the transaction. >If this value is true, the transaction will not be accepted. (string 1 Y/N)

26.	payMethod	O	For pre selection of payment method. Paymethod id, can be used to preselect payment method at merchant site, so user cannot select other payment method later (string..20), exact values will depend of implemented methods on service provider side.
27.	trType	O	Optional transaction type default assumed payment, valid values 1 - payment, 2 - pre authorization (applicable only to card payments only)
28.	extInstallmentoffset	O	Optional. In case installments are supported by the processing system then this parameter of installments can be used to indicate initial offset in months when first payment will be submitted (by acquirer). Applicable for card payments only. Integer max length 3
29.	extInstallmentperiod	O/R	Optional, required in case previous parameter is present. In case installments are supported by the processing system then this parameter of installments is used to indicate the number of payments/months the merchant requests for installments. Applicable for card payments only. Value must be >1. Max length 3 Installment parameters and recurring parameters together are not allowed on same request
30.	extRecurringfrequency	O	Optional. In case recurring payments are supported by the processing system then this parameter can be used to indicate frequency of recurring payments, defines minimum number of days between any two subsequent payments. The number of days equal to 28 is special value indicating that transactions are to be initiated on monthly basis. Applicable for card payments only. Max length 3

31.	extRecurringenddate	O/R	Optional, required in case previous parameter is present. In case recurring payments are supported by the processing system then this parameter can be used to indicate date after which recurring ends and no more transactions are initiated. The format is YYYYMMDD. Applicable for card payments only. Installment parameters and recurring parameters together are not allowed on same request.
32.	blockScore	O	Optional block score parameter that will be used to block the transaction if transaction riskScore reaches this value or above. (Positive Integer number) max length 9.
33.	cssUrl	O	The absolute or relative (to Cardlink Payment Gateway location on server) url of custom CSS stylesheet, to be used to display payment page styles. (string..128) Note: if absolute and payment page is SSL secured make sure the url is also SSL secured as browsers will not show unsecure element object warning.
34.	confirmUrl	R	Confirmation url where to send payment confirmation in case payment was successful (string..128)
35.	cancelUrl	R	Cancel url where to send payment feedback in case payment has failed or was canceled by user (string..128)
36.	var1	O	Optional merchant and acquirer agreed free variable type string ..255
37.	var2	O	Optional merchant and acquirer agreed free variable type string ..255
38.	var3	O	Optional merchant and acquirer agreed free variable type string ..255
39.	var4	O	Optional merchant and acquirer agreed free variable type string ..255
40.	var5	O	Optional merchant and acquirer agreed free variable type string ..255
41.	var6	O	Optional merchant and acquirer agreed free variable type string ..255
42.	var7	O	Optional merchant and acquirer agreed free variable type string ..255

43.	var8	O	Optional merchant and acquirer agreed free variable type string ..255
44.	var9	O	Optional merchant and acquirer agreed free variable type string ..255
45.	digest	R	Message digest to ensure and verify message security and integrity. SHA256 digest of all the field values above concatenated together with the shared secret (see section Calculation of the Digest).

Table 1: Payment Page fields

If extension parameters extInstallmentoffset, extInstallmentperiod are present and valid then the request in considered an installment payment (parent).

If extension parameters extRecurringfrequency, extRecurringenddate are present and valid then the request in considered a recurring payment (parent).

Note: Installment parameters and recurring parameters together are not allowed on same request. For this reason the merchant can send:

Either the installment fields:

<input type="hidden" name="extInstallmentoffset" size="3" value="2">
<input type="hidden" name="extInstallmentperiod" size="3" value="2">

Or the recurring fields:

<input type="hidden" name="extRecurringfrequency" size="3" value="12"> <input type="hidden" name="extRecurringenddate" size="8" value="20140712">

All parameters in the post must be in form default encoding (application/x-www-form-urlencoded) and form must be submitted with utf-8 encoding.

form.action="{supplied Cardlink Payment Gateway service url}"

form.method="POST"

form.enctype="application/x-www-form-urlencoded"

form.accept-charset="UTF-8"

Example of Sale transaction with Visa

<BEGINNING OF PAYMENT FORM>
<form id="form1" name="test" method="POST"
action="https://cardlink.test.modirum.com/vpos/shophandlermpi" accept-charset="UTF-8" >
<input type="hidden" name="version" value="2"/>
<input type="hidden" name="mid" value="0101119349"/>
<input type="hidden" name="lang" value="en"/>
<input type="hidden" name="deviceCategory value="0"/>
<input type="hidden" name="orderid" value="O170911143656"/>
<input type="hidden" name="orderDesc" value="Test order some items"/>

```
<input type="hidden" name="orderAmount" value="0.12"/>
<input type="hidden" name="currency" value="EUR"/>
<input type="hidden" name="payerEmail" value="cardlink@cardlink.gr”/>
<input type="hidden" name="payerPhone" value="6900000000”/>
<input type="hidden" name="payMethod" value="visa"/>
<input type="hidden" name="confirmUrl"
value="https://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=confirm"/>
<input type="hidden" name="cancelUrl"
value="https://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=cancel"/>
<input type="hidden" name="digest" value="Xw19+XA5lQXbzEHvvFYe1Zrm7N+rvpdIvzuIyM9HY3Q="/>
</form>
<END OF PAYMENT FORM>
```

Calculation of the Digest

Digest is the safety valve for the transactions between Company and Cardlink e- Commerce. Digest is one of the fields that the payment page sends and certifies the safe data transferring between the company and the Bank.

Requirements for Digest creation:

1) **The field values that the POST form sends (post_fields_values).**

   Based on the values of the payment page fields, the company needs to create a string with all the field values sent by payment page in the bank system.

   a. Concatenate all the values of all the possible fields listed in the table, the same order as parameters are listed in **Payment page fields** table.

   b. If a parameter is omitted, empty string is concatenated.

   c. The string must be encoded using UTF – 8 char encoding. This can be achieved by using the functions provided by the language that implements the solution (eg utf8_encode for **PHP** or Encoding.Convert for **.NET**).

2) **The SHARED SECRET**
   The bank communicates to every merchant a unique code which is called SHARED SECRET. The SHARED SECRET must be included at the end of the previous string.

3) **The base64 and sha-256 functions.**

Digest in the request POST (and in the return POST) is calculated by the following rule:

1) Concatenate all the values of all the possible fields listed in the table, the same order as parameters are listed in **Payment page fields** table.
2) Add SHARED SECRET
3) Encryption and Encoding using base64 and sha 256 functions
   a. Calculate SHA256 digest of step 1 (using of UTF-8 char encoding when converting string to bytes).
   b. Return the SHA256 digest.
   c. Encode digest bytes with Base64 encoding.

Digest=base64(sha256(utf8bytes(value1|value2|...|secret)))

Note: '|' indicates concatenation of data in formula and must not be added to data.

Never implement the digest calculation in browser using javascipt or similar as this way you expose your shared secret to the world. Only implement it in server side executed code as (jsp/servlet/asp/php etc).

In the previous example of Sale Transaction (page8), the digest value sent was calculated as « Xw19+XA5lQXbzEHvvFYe1Zrm7N+rvpdIvzuIyM9HY3Q=». According to the digest calculation rules this value is delivered as follows:

1) **Concatenate all the values of all the possible fields listed in the table, the same order as parameters are listed in Payment page fields table.**

Counter	Post Field	Post field value
1	version	2
2	mid	0101119349
3	lang	en
4	deviceCategory	0
5	orderid	O170911143656
6	orderDesc	Test order some items
7	orderAmount	0.12
8	currency	EUR
9	payerEmail	cardlink@cardlink.gr
...	...	...
...	...	...
26	payMethod	visa
...	...	...
...	...	...
...	...	...
34	confirmUrl	https://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=confirm
35	cancelUrl	https://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=cancel

Table 2: Sale Example Http Post Request field values

So, the **concatenated string** of the above values is the following:

01011193492en0O170911143656Test order some items0.12EURcardlink@cardlink.gr6900000000GRUKvisahttps://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=confirmhttps://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=cancel

Note: The concatenated string must contain all the values the merchant has sent, not just the required ones (the above string contains the optional values of **lang** and **deviceCategory**)**.**

2) Add SHARED SECRET

SHARED SECRET is the password between the bank and the merchant and must be added at the end of the previous string. Assume SECRET is «Cardlink1», then the produced sting is the following:

01011193492en0O170911143656Test order some items0.12EURcardlink@cardlink.gr6900000000GRUKvisahttps://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=confirmhttps://euro.test.modirum.com/vpostestsv4/shops/shopdemo.jsp?cmd=cancelCardlink1

3) Encryption and Encoding using base64 and sha256 functions

The final step of digest calculation is the use of base64 and sha256 functions according to the following rule:

Digest=base64(sha256(utf8bytes(value1|value2|...|secret)))

This produces the following result:

Xw19+XA5lQXbzEHvvFYe1Zrm7N+rvpdIvzuIyM9HY3Q=

Return message POST to inform merchant shopabout payment success or failure.

The following table describes the parameters of the POST from Cardlink Payment Gateway back to merchant shop.

Counter	Field (HTTP POST parameter)	Required / Optional	Description
1	version	R	Value 2
2	mid	R	Merchant id supplied (integer number) max length 30
3	orderid	R	Merchant shop order id string max length 50
4	status	R	Payment status (string..16) see section 2.4 payment statuses
5	orderAmount	R	Order amount (decimal number >0.0) same as in request
6	currency	R	Order amount currency (string 3, ISO ISO 4217 alphabetic code (EUR, USD)) same as in request
7	paymentTotal	R	Order amount plus fees and shipping and additional service charges if applicable (decimal number >0,0) Required when payment was a success, can be omitted when payment was failed or canceled
8	message	O	Optional message (string..128) can provide optional information about payment or error description.
9	riskScore	O	Optional information about the possible risk with transaction (integer number)
10	payMethod	O	Optional information about payment method used to complete transaction (string20).Is present only when payment was success
11	txId	O	Optional system assigned transaction reference id (integer number)
12	paymentRef	O	Optional end payment system reference or approval code. String 1..64
13	extData	O	Acquirer defined field may be encoded and contain subfields in format p1=v1&p2=v2.. (url encoded value) string .. 1024
14	digest	R	Message digest to ensure and verify message security and integrity. SHA256 digest of all the field values above concatenated together with the shared secret.

Note1: When payment is success the message is returned back to the url that merchant has defined as **confirmUrl** in request. If payment fails or it is canceled the message is returned to **cancelUrl** parameter provided in merchant request.

Note2: Since the latest Cardlink Payment Gateway version there is also available configurable option that server sends in background delayed (5..120 seconds) independent confirmation message (copy of redirection confirmation) to merchant server without user browser interaction as sometimes user browser may fail to reach back to merchant site. So it is recommended that merchant systems are prepared to handle multiple confirmations for same order properly due this feature and also possible user browser reloads, back buttoning etc. Background confirmation http request can be identified by having user-agent header set to value “Modirum VPOS”.

Recurring notification POST

The following table describes the parameters of the direct POST from Cardlink Payment Gateway back to merchant shop (recurring notifications url) in case of scheduled recurring child is processed by Cardlink server.

Counter	Field (HTTP POST parameter)	Required / Optional	Description
1.	version	O/R	Value 2
2.	mid	R	Merchant id supplied (integer number) max length 30
3.	orderid	R	Merchant shop order id string max length 50
4.	status	R	Payment status (string..16) see section 2.4 payment statuses
5.	orderAmount	R	Order amount (decimal number >0.0) same as in request
6.	currency	R	Order amount currency (string 3, ISO ISO 4217 alphabetic code (EUR, USD)) same as in request
7.	paymentTotal	R	Order amount plus fees and shipping and additional service charges if applicable (decimal number >0,0) Required when payment was a success, can be omitted when payment was failed or canceled
8.	message	O	Optional message (string..128) can provide optional information about payment or error description.
9.	riskScore	O	Optional information about the possible risk with transaction (integer number)
10.	payMethod	O	Optional information about payment method used to complete transaction (string20).Is present only when payment was success
11.	txId	O	Optional system assigned transaction reference id (integer number)
12.	Sequence	R	Sequence number or recurring (parent has sequence number 1, the first recurring child will have sequence number 2, etc)
13.	SeqTxId	R	The sequence transaction unique id in system (Integer)

14.	paymentRef	O	Optional end payment system reference or approval code. String 1..64
15.	digest	R	Message digest to ensure and verify message security and integrity. SHA256 digest of all the field values above concatenated together with the shared secret.

Table 4:Http Post Response fields with recurring

Payment statuses in response message

AUTHORIZED, CAPTURED	Payment was successful (accept order)
CANCELED	Payment failed, user canceled the process (deny order)
REFUSED	Payment failed, payment was denied for card or by bank (deny order)
ERROR	Non recoverable system or other error occurred during payment process (deny order)

Table 5: Payment Statuses in Response message

Digest Calculation in response message

Merchants may use the received digest to validate the Cardlink Payment Gateway response.
Digest in received message is calculated with the same rules as in request message.

Assuming that the sale transaction of the example was successful and the following field values where returned by Payment Gateway:

Counter	Post field	Post field value
1	version	2
2	mid	0101119349
3	orderid	O170911143656
4	status	CAPTURED
5	orderAmount	0.12
6	currency	EUR
7	paymentTotal	0.12
8	message	OK, 00 - Approved
9	riskScore	0
10	payMethod	visa
11	txId	926012471
12	paymentRef	138104
13	digest	FpwgGyCRwhmF6CWtRFLqfkuQpdPyX8Xh3tJg3E891SA=

Table 6: Sale Example Http Post Response field values

Note: The above field values are listed in the same order as parameters are listed in table.

a. The concatenated string of the above values is the following:
20101119349O170911143656CAPTURED0.12EUR0.12OK, 00 - Approved0visa926012471138104

b. By adding the SHARED SECRET, the following string is produced
20101119349O170911143656CAPTURED0.12EUR0.12OK, 00 - Approved0visa926012471138104Cardlink1

c. By using base64 and SHA256

Digest=base64(sha256(utf8bytes(value1|value2|...|secret))) =
FpwgGyCRwhmF6CWtRFLqfkuQpdPyX8Xh3tJg3E891SA=

This is the same as the digest field value the merchant received.