



Gateway Payment Service API

Merchant Specification

Version 4.9

## Table of Contents

<b>1</b>	<b>Overview .....</b>	<b>11</b>
1.1	Audience .....	11
1.2	Payment Service.....	11
1.3	Requirements.....	11
1.4	Data Security .....	11
1.5	Revision History .....	11
<b>2</b>	<b>Authentication .....</b>	<b>14</b>
2.1	Payment Service Location .....	14
2.2	Encoding.....	14
2.3	Service Authentication Test .....	14
2.3.1	Authentication Fields .....	14
2.3.2	Authentication Request Example.....	14
2.3.3	Authentication Response Example .....	15
2.3.4	Failed Authentication Response Example.....	15
2.4	Check Service Availability.....	16
2.4.1	Check Service Availability Fields .....	16
2.4.2	Check Service Availability Request Example .....	16
2.4.3	Check Service Availability Response Example .....	16
2.4.4	Failed Check Service Availability Response Example .....	17
<b>3</b>	<b>Service Response Format.....</b>	<b>17</b>
<b>4</b>	<b>Transaction Request Fields .....</b>	<b>18</b>
4.1	Authentication Parameters.....	18
4.2	Customer Parameters .....	18
4.3	Payment and Bank Information Parameters .....	19
4.4	Line Items Parameters .....	20
4.5	Adjustment Parameters .....	20
4.6	Other Merchant Specified Parameters .....	21
4.7	Membership Adjustment Parameters .....	24
4.8	Tokenization.....	24
4.8.1	Requesting the token.....	24
4.8.2	CCAUTHCAP Request Using Tokenization.....	27

5 Credit Card Transactions.....	27
5.1 Authorization .....	27
5.1.1 Authorization Parameters.....	27
5.1.2 Authorization Request Example.....	31
5.1.3 Authorization Response Example .....	31
5.1.4 Failed Authorization Attempt Response Example .....	32
5.1.5 Service Declined Response Example.....	32
5.2 Delayed Capture.....	33
5.2.1 Delayed Capture Request .....	33
5.3 Authorization and Capture (Sale).....	34
5.3.1 Authorization Request Example (CCAUTHCAP) .....	34
5.4 Reversal.....	34
5.4.1 Reversing or voiding an Authorization (CCREVERSE).....	34
5.4.2 CREDIT_ON_FAIL Flag .....	35
5.5 Credit.....	37
5.5.1 Minimum required parameters for issuing a Credit (CCCREDIT):.....	37
5.5.2 Force Credit Request.....	37
5.5.3 Multiple Line Items .....	37
5.5.4 Example CCAUTHORIZE request with Multiple Line Items. ....	38
5.5.5 Example CCAUTHORIZE Response with Multiple Line Items .....	38
5.5.6 Example Request for Capturing a Specific Line Item .....	39
5.5.7 Capturing, Reversing or Crediting the Entire Order with Multiple Line Items.....	40
5.5.8 Example Request for Sending Reversal of a Specific Line Item .....	41
5.5.9 Line Item Types .....	43
5.6 Checking the Status of an order.....	43
5.6.1 CCSTATUS Example Request using REQUEST_REF_PO_ID.....	44
5.6.2 CCSTATUS Example Request using REQUEST_REF_PO_ID_XTL.....	44
5.6.3 CCSTATUS Example Response.....	44
5.6.4 CCSTATUS Response Fields .....	45
5.6.5 Transaction Status is PENDING .....	46
5.7 Updating Order Data.....	46
5.7.1 CCTRANSUPDATE Example Request using REQUEST_REF_PO_ID .....	46

5.7.2	CCTRANSUPDATE Example Request using REQUEST_REF_PO_ID_XTL .....	46
5.7.3	CCTRANSUPDATE Example Response .....	47
5.7.4	CCTRANSUPDATE Response Fields.....	47
<b>6</b>	<b>ACH/eCheck Transactions .....</b>	<b>48</b>
6.1	Transaction State Updates Through Postback (Optional).....	48
6.2	Transaction State using Order Detail API (Optional) .....	48
6.2.1	Transaction Status.....	48
6.3	Gateway Request Parameters .....	49
6.4	Gateway Actions .....	50
6.5	Authorization and Capture (ACHAUTHCAP) Request and Response .....	50
6.5.1	Authorization (ACHAUTHCAP) Request Example.....	50
6.5.2	Authorization (ACHAUTHCAP) Response Example (Pending Status).....	50
6.6	Account Validation via AUTHORIZATION without CAPTURE (ACHAUTHORIZE) Request and Response .....	51
6.6.1	Authorization (ACHAUTHORIZE) Request Example .....	51
6.6.2	Authorization (ACHAUTHORIZE) Response Example (Pending Status).....	52
6.6.3	Service Declined Response Example.....	52
6.7	Reversal Request and Response .....	53
6.7.1	Reversal Request Example .....	53
6.7.2	Reversal Response Example.....	53
6.8	Credit Request and Response .....	54
6.8.1	Credit Request.....	54
6.8.2	Credit Response .....	54
6.9	Status Change .....	54
<b>7</b>	<b>Boleto .....</b>	<b>54</b>
7.1	Request Parameters.....	55
7.2	Additional Response Parameters.....	56
7.2.1	Boleto Request Example .....	56
7.2.2	Boleto Response Example .....	56
<b>8</b>	<b>Pix Payment .....</b>	<b>57</b>
8.1	PIX Request Parameters.....	57
8.2	Additional Response Parameters.....	58

8.3	Pix Request Example .....	58
8.4	Pix Response Example .....	58
<b>9</b>	<b>Credilink .....</b>	<b>59</b>
9.1	How it Works.....	60
9.2	How to Use Credilink.....	60
9.3	Additional Parameters .....	60
<b>10</b>	<b>Order Insight .....</b>	<b>61</b>
10.1	Additional Request Parameters .....	61
10.2	Reporting.....	62
10.2.1	Order Insight Lookup Statuses .....	62
<b>11</b>	<b>Transaction Modes .....</b>	<b>63</b>
11.1	Recurring.....	63
11.1.1	Minimum required fields for sending a simple recurring transaction request using CUST_ID: 63	
11.1.2	Required fields for specifying which payment account to charge with the recurring transaction request using PMT_L4 parameter: .....	63
11.1.3	Required fields for specifying which payment account to charge with the recurring transaction request using PMT_ID parameter:.....	64
11.1.4	Required fields for specifying which payment account to charge with the recurring transaction request using PMT_ID_XTL parameter:.....	64
11.1.5	Required parameters for sending new billing information with the recurring transaction request: 65	
11.1.6	Parameters for sending Renewal Transactions (Rebills):.....	66
11.2	Memberships .....	66
11.2.1	Suggested fields for sending a membership transaction (new customer): .....	67
11.2.2	Parameters to create Subscription directly from Gateway API .....	68
11.2.3	Unique Customer Login Check .....	70
11.2.4	Membership Cancelation Requests (SUB_CANCEL).....	70
11.2.5	Membership Product Update Requests (Upgrade/downgrade Product).....	72
<b>12</b>	<b>Response Fields.....</b>	<b>75</b>
12.1	Response Formats.....	75
12.2	Successful Transaction Response Fields .....	75
12.3	Declined Transaction Response Fields .....	77

12.4	Service Declined Response Fields .....	79
13	Dynamic Descriptor.....	79
13.1	Dynamic Descriptor.....	79
13.1.1	Dynamic Descriptor Restrictions.....	79
14	Address Verification Service (AVS).....	80
14.1	Address Verification Service .....	80
14.1.1	Check AVS Flag (CHKAVS).....	80
14.1.2	Positive AVS Response Codes .....	80
14.1.3	Negative AVS Response Codes.....	82
14.1.4	Custom Settings (AVSMatchSet) .....	82
14.2	Card Security Code .....	82
14.2.1	Check CVV Flag (CHKCVV) .....	82
14.2.2	Positive CVV Response Codes .....	83
14.2.3	Negative CVV Response Codes .....	83
14.2.4	Custom Settings (CVVMatchSet).....	84
15	3D Secure Transactions.....	84
15.1	3D Secure .....	84
15.1.1	Low Value SCA Exemption .....	84
15.1.2	External 3DS Provider Scenario.....	84
15.1.3	Payment Gateway as 3DS Provider Using Redirect Scenario.....	85
15.1.4	Determining which outcome happened .....	89
16	Currency .....	97
16.1	Multi-currency Support.....	97
16.2	Request_currency Parameter .....	98
16.3	Currency Response Fields .....	98
16.3.1	Testing Multi-currency Response .....	98
16.3.2	Multi-Currency Response Example .....	98
17	Partial Authorization .....	99
17.1	What is Partial Authorization? .....	99
17.1.1	How It Works.....	99
17.1.2	Partial Authorization Required Parameters.....	99
17.1.3	Failed Transaction Due to Minimum Amount Response Example .....	100

17.1.4	Partial Authorization Successful Response Example .....	100
<b>18</b>	<b>External Order ID Uniqueness.....</b>	<b>101</b>
18.1	UNIQUE_XTL_ORDER_ID Parameter.....	101
18.1.1	UNIQUE_XTL_ORDER_ID Setting .....	101
18.1.2	Example Decline Response when UNIQUE_XTL_ORDER_ID parameter is set to "1":.....	102
<b>19</b>	<b>Processor Specific Fields .....</b>	<b>102</b>
19.1	Processor Specific Fields .....	102
<b>20</b>	<b>PagoEfectivo.....</b>	<b>103</b>
20.1	PagoEfectivo Features.....	103
20.2	PagoEfectivo Request Parameters.....	103
20.3	Additional Response Parameters.....	104
20.4	PagoEfectivo Request Example.....	104
20.5	PagoEfectivo Response Example .....	105
<b>21</b>	<b>Apple Pay .....</b>	<b>106</b>
21.1	Overview .....	106
21.2	Customers who can use Apple Pay .....	106
21.3	Flow.....	106
21.4	Fetch Apple Pay configuration data from the gateway .....	107
21.5	Update the configuration with the specific payment/order details.....	109
21.6	Add the Apple Pay library to your payment page.....	109
21.7	Create a JS function to validate your domain .....	109
21.8	Present the Apple Pay button.....	110
21.9	Create and start the Apple Pay session .....	110
21.9.1	Handle the Authorized Payment.....	110
21.9.2	Handle a Canceled Payment or Error.....	110
21.10	Send the authorized payment data from the session to the gateway .....	112
21.11	Important requirements for Apple Pay authorization requests .....	112
21.12	Rebiling and Credentials on File.....	112
<b>22</b>	<b>Scheme Level Tokenization .....</b>	<b>112</b>
22.1	Additional Response Parameter .....	113
<b>23</b>	<b>3<sup>rd</sup> Party Tokens: Merchant Decryption .....</b>	<b>113</b>
23.1	Overview .....	113

23.2	Request Parameters.....	113
<b>24</b>	<b>Single Euro Payments Area (SEPA).....</b>	<b>114</b>
24.1	SEPA Direct Debit Payment Process.....	114
24.2	Transaction state updates through Postback .....	114
24.3	Authorization Request (SEPA Direct Debit) - DBTAUTHORIZE .....	114
24.4	Authorization Capture Request (SEPA Direct Debit) - DBTCAPTURE .....	115
24.5	Debit Request Parameters (DBTAUTHORIZE) .....	115
24.5.1	Debit Request Example .....	116
24.5.2	Debit Response Example (Pending) .....	116
24.5.3	Debit Response Example (Success).....	117
24.6	Reversals .....	118
24.6.1	Reversing a mandate (DBTREVERSE) .....	118
24.6.2	DBTREVERSE transaction Request Example.....	118
24.7	Refunds or Credits.....	119
24.7.1	Issuing refunds or credits (DBTCREDIT) .....	119
24.8	Recurring transactions .....	119
24.8.1	Recurring transaction Request Example (DBTDEBIT & request_rebill=2) .....	119
24.8.2	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 2).....	119
24.8.3	Recurring transaction Request Example (DBTDEBIT & request_rebill=1) .....	120
24.8.4	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 1).....	121
24.9	Additional Parameters .....	122
<b>25</b>	<b>iDEAL Payments .....</b>	<b>122</b>
25.1	Debit Request Parameters (DBTAUTHORIZE) .....	122
25.1.1	Debit Request Example .....	124
25.1.2	Debit Response Example (Pending) .....	124
25.1.1	Debit Response Example (Success).....	125
25.2	Reversals .....	126
25.2.1	Reversing a mandate (DBTREVERSE) .....	126
25.2.2	DBTREVERSE transaction Request Example.....	126
25.3	Refunds or Credits.....	126
25.3.1	Issuing refunds or credits (DBTCREDIT) .....	126
25.4	Recurring transactions .....	127

25.4.1	Recurring transaction Request Example (DBTDEBIT & request_rebill=2) .....	127
25.4.2	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 2).....	127
25.4.3	Recurring transaction Request Example (DBTDEBIT & request_rebill=1) .....	128
25.4.4	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 1).....	129
25.5	Additional Parameters .....	130
<b>26</b>	<b>EPS Payments</b> .....	<b>130</b>
26.1	Debit Request Parameters (DBTAUTHORIZE) .....	130
26.1.1	Debit Request Example .....	132
26.1.2	Debit Response Example (Pending) .....	132
26.1.1	Debit Response Example (Success).....	133
26.2	Reversals .....	134
26.2.1	Reversing a mandate (DBTREVERSE) .....	134
26.2.2	DBTREVERSE transaction Request Example.....	134
26.3	Refunds or Credits.....	134
26.3.1	Issuing refunds or credits (DBTCREDIT) .....	134
26.4	Recurring transactions .....	135
26.4.1	Recurring transaction Request Example (DBTDEBIT & request_rebill=2) .....	135
26.4.2	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 2).....	135
26.4.3	Recurring transaction Request Example (DBTDEBIT & request_rebill=1) .....	136
26.4.4	Recurring transaction Response Example (DBTDEBIT & Request Rebill = 1).....	137
26.5	Additional Parameters .....	138
<b>27</b>	<b>Testing</b> .....	<b>138</b>
27.1	Using the test bank .....	138
27.2	Basic Gateway Test .....	138
27.3	Test Credit Cards .....	140
<b>Appendix A:</b>	<b>Service Request Types</b> .....	<b>141</b>
<b>Appendix B:</b>	<b>Transaction Status</b> .....	<b>142</b>
<b>Appendix C:</b>	<b>API Response Codes</b> .....	<b>143</b>
<b>Appendix D:</b>	<b>Service Response Codes</b> .....	<b>144</b>
<b>Appendix E:</b>	<b>Address Verification Service Codes</b> .....	<b>148</b>
<b>Appendix F:</b>	<b>CVV Response Codes</b> .....	<b>150</b>
<b>Appendix G:</b>	<b>Negative Option Billing- MCC 5968</b> .....	<b>150</b>

Appendix H: 3-D Secure Flow.....	153
Appendix I: Automatic Account Updater (AAU) .....	154
Appendix J: Visa Trial Processing .....	154
Appendix J: Card on File Matrix .....	155

# 1 Overview

---

## 1.1 Audience

This documentation is intended to be used by the merchant organization's technical staff. This document assumes that the reader has a working knowledge of programming languages, hardware and software requirements specified in this documentation.

## 1.2 Payment Service

The Payment Service is designed to allow merchants communicate and process online transactions with the payment gateway's transaction processing system. The API serves as a bridge between the merchant's website and various financial institutions in processing real-time payment transactions.

## 1.3 Requirements

To integrate with the Payment Service API, the following are required:

- Internet Connection
- Knowledge of HTTP Post and Get methods
- Knowledge of parsing data in any of the following formats: XML and JSON

## 1.4 Data Security

For security and guarding confidentiality, all cardholder and transaction data are sent over the internet using the TLS 1.2 cryptographic protocol. All Cardholder data is encrypted at all times. The payment gateway is **PCI-DSS Level 1** compliant.

## 1.5 Revision History

Version	Date	Description
2.0	3 Jul 2012	Added parameters for membership transactions. Updated requirement for REQUEST_API_VERSION parameter. Added Membership in Transaction Modes (Section 6). Added SUB_CANCEL and SUB_UPDATE request action types. Updated Response Fields (Section 7). Updated all gateway request and response examples. Added Cross-sale transactions descriptions. Updated Service Response table in Appendix section. Removed REQUEST_TYPE and REQUEST_SCRUB_FLAG fields from Transaction Request Fields.
2.1	18 Jul 2012	Added REQUEST_AFF_ID_SUB to the list of gateway fields. Update REQUEST_CURRENCY description. Updated BILL_ADDR_STATE description.
2.2	18 Sept 2012	Added MERCHANT_ACCT_ID as a required parameter on recurring transactions (Section 6).
3.2	21 Jan 2014	Added card type fields in gateway response (Section 8). Added Mastercard EMS Fraud Score parameters (new section, 16). Formatting changes (all sections).
3.3	28 Mar 2014	Updated Mastercard EMS Fraud Score section with gateway response fields. Added gateway response examples and EMS Reason Codes table.

<b>3.4</b>	17 Sep 2014	Added PROC_UDF fields in Transaction Request Parameters table. Added new section for merchants using eMerchantPay banks and Threatmetrix™.
<b>3.5</b>	9 Oct 2014	Added format for CUST_BIRTHDAY field. Updated Transaction Request and Response Parameters tables (Sections 4 and 8). Updated 3D Secure section. Updated Appendix C: Transaction Status. Added “Pending” status explanation (Section 5.6.5). Updated Section 17: Processor Specific Fields (this was changed from being a Threatmetrix-specific section).
<b>3.6</b>	14 May 2015	Added CARD_BALANCE in Response Fields (Section 8.2) and gateway response examples. Added REQUEST_CURRENCY field on recurring transaction request parameters (Section 7). Added Partial Auth testing using test banks (Section 18).
<b>3.7</b>	12 Feb 2016	Removed Deprecated Payment Types Appendix Extension of UDF up to XTL_UDF20
<b>3.8</b>	6 Oct 2016	Added CUST_BRCPFCNPJ parameter to send CPF or CNPJ with Brazilian transactions
<b>3.9</b>	14 Mar 2017	Added PMT_DESCRIPTOR_PHONE for dynamic support phone number (aka CITY field) on customer descriptor. Reference section 9.1
<b>3.9.1</b>	18 Aug 2017	Extended FORCE_CREDIT ability to support OCT
<b>3.9.2</b>	5 Sept 2017	Description of <a href="#">tokenization</a> feature and its use
<b>4.0</b>	1 June 2018	Description of process and parameters for 3-D Secure transactions
<b>4.1</b>	9 July 2018	Added test scenario information for multi-currency processing , and additional response assertion test scenarios
<b>4.2</b>	27 Feb 2019	Added Support for ACHAUTHORIZE for ACH authorization without transferring any funds. Also added CCTTRANSUPDATE for updating orders with Receipts, Added TRANS_REBILL_TYPE, MBSHP_ID_XTL, TRANS_TRIAL_REBILL_CUSTOMER_CONSENT, TRANS_CUSTOMER_RECEIPT Request Parameters to store MCC5968-Mastercard rule, added CARD_ON_FILE_FLAG parameter
<b>4.3</b>	12 Jul 2019	Added Automatic Account Updater (AAU) Update Date; PMT_AAU_UPDATE_DT and Update type Description; PMT_AAU_UPDATE_DESC. Also added Appendix I for AAU Services
<b>4.4</b>	13 Jul 2020	Added COF parameters including request_initiator, request_installment, pmt_numbr_cof and added option “2” on request_rebill parameter. Also Added Appendix J for Visa Trial Processing and revised TRANS_REBILL_TYPE, MBSHP_ID_XTL, TRANS_CUSTOMER_RECEIPT Request Parameters to store VISA Trial processing. Also added Service Response Code ‘519’for ‘Missing Trial Descriptor’. Added two additional parameters for 3DS (P3DS_VERSION P3DS_TRANSID). Also Added CARD_DETAIL (Credit/Debit) Field in our Response. Removed support for Double Pipes

		Delimited. Removed Card_on_file_flag field (deprecated by request_initiator). Removed SEPA Section (not supported)
4.5	20 April 2022	Added new Service Response Codes, added Appendix J: Card on File Matrix, Added Parameters to create Subscription directly from Gateway API, Added Boleto Payment Processing, Added Pix Payment Processing, Added Brazilian Credilink, added SERVICE_RESPONSE, SERVICE_ADVICE, PROCESSOR_RESPONSE, PROCESSOR_ADVICE to CCSTATUS, Added REQ_ID in all Responses, Added PMT_NUM to scrub decline response, Added 3DS 2.0 Guide
4.6	30 August 2023	Removed FORCE_NEW_CUST from PIX Request, Added REQ_LOW_VALUE_SCAEXEMPTION Field, Added a new REQUEST_ACTION - PAGSALE for PagoEfectivo, Added Digital Wallet Parameters PMT_WALLET AND PMT_CRYPTOGRAM, Added Cash Discount's CONVENIENCE_FEE Field, Added TAX_EXEMPT and TAX_AMT for Level II Processing, updated Appendix D -Service Response Codes with new Scrub decline response codes, added CCRDR Request Action, Deprecated Outcome 2: 3DS version 1 authentication, Added <a href="#">PagoEfectivo Payment Section</a> , Added <a href="#">Apple Pay Section</a>
4.7	03 January 2024	<a href="#">Added Scheme Level Tokenization Section</a> , Added Scheme Level Tokenization Parameters TRANS_NTOKEN_USED, ORIG_CARD_BRAND_TRANSID and CARD_BRAND_TRANSID, Added New Service Response Codes (700, 720, 725), updated <a href="#">Credilink Section</a> with current functionalities
4.8	06 March 2024	Added <a href="#">Single Euro Payments Area (SEPA)</a> Direct Debit Support, added new Request Actions for Debit Payment Type Support; DBTAUTHORIZE, DBTCAPTURE DBTDEBIT, DBTREVERSE and DBTCREDIT, added DEBIT_TYPE parameter to support <a href="#">Single Euro Payments Area (SEPA)</a> , <a href="#">iDEAL Payments</a> and <a href="#">EPS Payments</a> . Revised CCAUTHORIZE and CCCAUTHCAP gateway timeout from 90 Secs to 120 Secs. Also added ACHPAYOUT used for ACH Direct PAYOUT, corrected Apple Pay FetchUrl from <a href="https://api.inoviopay.com/apple-pay-poc/api/session/create">https://api.inoviopay.com/apple-pay-poc/api/session/create</a> to <a href="https://api.inoviopay.com/apple-pay-services/api/session/create">https://api.inoviopay.com/apple-pay-services/api/session/create</a> , Added SUB_UPDATE_PMT_ID Req/Resp examples
4.9	23 January 2025	Added <a href="#">Tokenization</a> updates, additional <a href="#">Response Paramaters</a> , <a href="#">Service Provider Tokens: Merchant Decryption</a> , <a href="#">Verifi: Order Insight</a> , CCRDRDELETE to <a href="#">Service Request Types</a> , new <a href="#">API Response Codes</a> , and general document cleanup

## 2 Authentication

---

### 2.1 Payment Service Location

The Payment Service is located at:

[https://api.inoviopay.com/payment/pmt\\_service.cfm](https://api.inoviopay.com/payment/pmt_service.cfm)

### 2.2 Encoding

Payment Service API uses 8-bit Unicode Transformation Format as described in the ISO/IEC 10646. Merchants are required to send Service requests with content-type, “**application/x-www-form-urlencoded**”; and charset, “iso-8859-1,\*; utf-8” using HTTP Post method.

### 2.3 Service Authentication Test

The first requirement in integrating with the Payment Service is to have a working Service login name, password and website ID.

#### 2.3.1 Authentication Fields

Field Name	Description
REQ_USERNAME	Merchant's Service username
REQ_PASSWORD	Merchant's Service password
SITE_ID	Merchant's website ID
REQUEST_ACTION	Service request action: <b>TESTAUTH</b>
REQUEST_RESPONSE_FORMAT	<a href="#">Service response format.</a>
REQUEST_API_VERSION	API Version of this document. (4.9) Must be sent on all gateway requests.

#### 2.3.2 Authentication Request Example

Below is a complete HTTP Post request example (including the header information) that contains the required Fields needed when testing basic gateway authentication:

```
POST /payment/pmt_service.cfm HTTP/1.0
User-Agent: Mozilla/117.0.1 (compatible; MS Edge 117.0.2045.36; Windows NT 5.1)
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */
Accept-Encoding: identity
Accept-Language: en-us,en
Host: my.host.invalid
Accept-Charset: iso-8859-1,*;utf-8
Content-Type: application/x-www-form-urlencoded
Content-Length: 137
req_username=test@example.com&req_password=P5sswOrD!1&request_action=TESTAUTH&site_id=0&request_response_format=XML&request_api_version=4.8
```

### 2.3.3 Authentication Response Example

Each Service request will produce a corresponding response document

```
<?xml version="4.6" encoding="UTF-8"?>
<RESPONSE>
<REQUEST_ACTION/>
<TRANS_STATUS_NAME/>
<TRANS_VALUE/>
<TRANS_ID/>
<CUST_ID/>
<XTL_CUST_ID/>
<MERCH_ACCT_ID/>
<CARD_BRAND_NAME/>
<PMT_L4/>
<API_RESPONSE>0</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE>100</SERVICE_RESPONSE>
<SERVICE_ADVICE>User Authorized</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>0</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>0</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<PROC_ID/>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>
```

### 2.3.4 Failed Authentication Response Example

Below is an example response of a failed authentication request. API\_RESPONSE “101” indicates a failed authentication attempt.

```
<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE>
<REQUEST_ACTION/>
<TRANS_STATUS_NAME/>
<TRANS_VALUE/>
<TRANS_ID/>
<CUST_ID/>
<XTL_CUST_ID/>
<MERCH_ACCT_ID/>
<CARD_BRAND_NAME/>
<PMT_L4/>
<PMT_ID>441402</PMT_ID>
<PMT_ID_XTL/>
<API_RESPONSE>101</API_RESPONSE>
<API_ADVICE>Invalid login information</API_ADVICE>
<SERVICE_RESPONSE>0</SERVICE_RESPONSE>
<SERVICE_ADVICE>Declined</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>0</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>0</INDUSTRY_RESPONSE>
```

```
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>
```

## 2.4 Check Service Availability

Use request action, “TESTGW”, to check if Payment Service is available to process requests.

### 2.4.1 Check Service Availability Fields

Field Name	Description
REQ_USERNAME	Merchant's Service username
REQ_PASSWORD	Merchant's Service password
REQUEST_ACTION	Service request action: <b>TESTGW</b>
SITE_ID	Merchant's website ID
REQUEST_RESPONSE_FORMAT	Service response format
REQUEST_API_VERSION	API Version (4.9)

### 2.4.2 Check Service Availability Request Example

```
POST /payment/pmt_service.cfm HTTP/1.0
User-Agent: Mozilla/4.6 (compatible; MSIE 7.0; Windows NT 5.1)
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-
flash, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword,
/*
Accept-Encoding: identity
Accept-Language: en-us,en
Host: Host.example.com
Accept-Charset: iso-8859-1,*,utf-8
Content-Type: application/x-www-form-urlencoded
Content-Length: 112
req_username=test@example.com&req_password=P5ssw0rd!1&request_action=TESTGW&site_id=0
&request_response_format=XML&request_api_version=4.8
```

### 2.4.3 Check Service Availability Response Example

```
<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE>
<REQUEST_ACTION/>
<TRANS_STATUS_NAME/>
<TRANS_VALUE/>
<TRANS_ID/>
<CUST_ID/>
<XTL_CUST_ID/>7
<MERCH_ACCT_ID/>
<CARD_BRAND_NAME/>
```

```

<PMT_L4/>
<PMT_ID/>
<PMT_ID_XTL/>
<API_RESPONSE>o</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE>101</SERVICE_RESPONSE>
<SERVICE_ADVICE>Service Available</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>o</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>o</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<PROC_ID/>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

#### 2.4.4 Failed Check Service Availability Response Example

```

<?xml version="1.0" encoding="UTF-8"?>
<RESPONSE>
<REQUEST_ACTION/>
<TRANS_STATUS_NAME/>
<TRANS_VALUE/>
<TRANS_ID/>
<CUST_ID/>
<XTL_CUST_ID/>
<MERCH_ACCT_ID/>
<CARD_BRAND_NAME/>
<PMT_L4/>
<PMT_ID/>
<PMT_ID_XTL/>
<API_RESPONSE>x</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE></SERVICE_RESPONSE>
<SERVICE_ADVICE>Service Unavailable </SERVICE_ADVICE>
<PROCESSOR_RESPONSE>o</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>o</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

## 3 Service Response Format

For every request, the Payment Service API returns a response message. Merchants can choose one from the following supported formats: **XML** and **JSON**. To set the Service Response format, send

the `REQUEST_RESPONSE_FORMAT` parameter with your transaction request. In this document, the response examples are in XML format.

## 4 Transaction Request Fields

A transaction request includes:

- Authentication Parameters
- Customer Parameters
- Transaction and Payment Information Parameters
- Line Item Parameters
- Adjustment Parameters (used for issuing credits, reversals and delayed captures).
- Merchant Specified Parameters

### 4.1 Authentication Parameters

Field Name	Description
<code>REQ_USERNAME</code>	Merchant's Service Username.
<code>REQ_PASSWORD</code>	Merchant's Service Password.
<code>REQUEST_ACTION</code>	Service Request Action
<code>REQUEST_API_VERSION</code>	Payment Service API Version (4.9) Must be sent on all gateway requests.
<code>REQUEST_RESPONSE_FORMAT</code>	<a href="#">Service response format</a>
<code>SITE_ID</code>	Merchant's Website ID

### 4.2 Customer Parameters

Field Name	Description
<code>BILL_ADDR</code>	Cardholder Billing Street Address
<code>BILL_ADDR_CITY</code>	Cardholder Billing City
<code>BILL_ADDR_COUNTRY</code>	Cardholder Billing Country
<code>BILL_ADDR_STATE</code>	Cardholder Billing State
<code>BILL_ADDR_ZIP</code>	Cardholder Billing Postal/ZIP code
<code>CUST_BIRTHDAY</code>	Cardholder's birthday. FORMAT: MM-DD-YYYY
<code>CUST_DLN</code>	Cardholder Driver's License ID

<b>CUST_DLN_STATE</b>	Cardholder Driver's License State (US Customers Only)
<b>CUST_EMAIL</b>	Cardholder Email Address  Note: Required by some banks.
<b>CUST_FNAME</b>	Cardholder's First Name  Note: Required by some banks.
<b>CUST_LNAME</b>	Cardholder's Last Name  Note: Required by some banks.
<b>CUST_LOGIN</b>	Cardholder's Login or User Name
<b>CUST_PASSWORD</b>	Cardholder's Password  Password must be at least 10 characters with 1 number, lower case and upper case letter.
<b>CUST_PHONE</b>	Cardholder's Phone Number  example +1 (123) 456-7890  Note: Required by some banks.
<b>CUST_SSN_L4</b>	Cardholder's Last 4 digits of Social Security Number (US Customers Only)
<b>SHIP_ADDR</b>	Cardholder's Shipping Street Address
<b>SHIP_ADDR_CITY</b>	Cardholder's Shipping City
<b>SHIP_ADDR_COUNTRY</b>	Cardholder's Shipping Country
<b>SHIP_ADDR_STATE</b>	Cardholder's Shipping State
<b>SHIP_ADDR_ZIP</b>	Cardholder's Shipping Postal/ZIP Code
<b>SITE_ID</b>	Merchant's Website ID
<b>XTL_IP</b>	Cardholder's IP Address
<b>MBSHP_ID_XTL</b>	External Membership ID
<b>MBSHP_ID_XTL</b>	External Membership ID
<b>USER_AGENT_XTL</b>	Software agent responsible for retrieving and facilitating end-user interaction with Web content

### 4.3 Payment and Bank Information Parameters

Field Name	Description
<b>PMT_DESCRIPTOR</b>	Bank Dynamic Descriptor

<b>PMT_DESCRIPTOR_PHONE</b>	Bank Dynamic Customer Support Phone Number (aka City Field)
<b>PMT_EXPIRY</b>	Credit Card Expiration Date
<b>PMT_KEY</b>	Credit Card CVV2 or CVC2 Code
<b>PMT_NUMB</b>	Credit Card Number
<b>TOKEN_GUID</b>	Token ID used in place of pmt_numbr. See <a href="#">Tokenization</a> .
<b>PMT_L4</b>	Last 4 digits of the account or credit card number
<b>PMT_ID</b>	Payment Unique Identifier
<b>PMT_ID_XTL</b>	External Payment ID
<b>REQUEST_CURRENCY</b>	Currency 3-letter Code  See <a href="#">Currency</a> section for more information.
<b>MERCH_ACCT_ID</b>	Merchant Account ID
<b>DEBIT_TYPE</b>	Debit Type used for the transaction. ( <a href="#">SEPA</a> , <a href="#">iDEAL</a> , and <a href="#">EPS</a> )

#### 4.4 Line Items Parameters

Field Name	Description
<b>LI_COUNT_1</b>	Line Item Count. Use this parameter to indicate quantity. Max. Value is "10".
<b>LI_PROD_ID_1</b>	Line Item Product ID 1
<b>LI_VALUE_1</b>	Line Item Transaction Amount 1

#### 4.5 Adjustment Parameters

Field Name	Description
<b>REQUEST_REF_PO_ID</b>	Reference Order ID is used when sending adjustment request against authorizations (i.e. Delayed Capture, Reversal and Credit requests).
<b>REQUEST_REF_PO_LI_ID</b>	Reference Line Item ID. Use this if multiple line items processing is needed.
<b>CUST_ID</b>	Customer ID created by the system after sending a successful authorization request.
<b>REQUEST_REF_PO_ID_XTL</b>	Request Reference XTL_ORDER_ID (Merchant's Order ID). Used for sending <a href="#">CCSTATUS</a> requests.

<b>CREDIT_ON_FAIL</b>	Used with reversal (void) requests. If set to 1, the system will automatically credit the transaction when the reversal request failed. See <a href="#">CREDIT_ON_FAIL</a> explanation for details.
<b>FORCE_CREDIT</b>	Used for sending <a href="#">Force Credit</a> transactions.

## 4.6 Other Merchant Specified Parameters

Field Name	Description
<b>CHKAVS</b>	Address Verification Service Flag (this is for enabling, ignoring or disabling the AVS option). <b>AVS Check is enabled by default.</b>
<b>AVSMATCHSET</b>	Used for setting the response codes to check when approving transactions based on the AVS response code. See <a href="#">AVSMatchSet</a> explanation for more details.
<b>CHKCVV</b>	CVV Check flag (this is for enabling, ignoring, or disabling the Credit Card CVV2 or CVC2 check). <b>CVV check is enabled by default. Some processors may automatically decline transactions that return a negative CVV response code.</b>
<b>CVVMATCHSET</b>	Used for setting the response codes to check when approving transactions based on the CVV response code. See <a href="#">CVVMatchSet</a> explanation for more details.
<b>CUST_BRCPFCNPJ</b>	Individual CPF/Business CPNJ Number – Specific to Brazil <ul style="list-style-type: none"> <li>• See <a href="#">Section 8</a> for use with Pix Payment and <a href="#">Section 9</a> for use with Credilink</li> </ul>
<b>CONVENIENCE_FEE</b>	Reference Field used to indicate the surcharge fee amount which is included in the amount of the transaction
<b>REQUEST_AFF_ID</b>	External Affiliate ID
<b>REQUEST_API_VERSION</b>	Payment Service API Version (4.9)
<b>REQUEST_LANGUAGE</b>	Language 3-letter Code
<b>REQUEST_RESPONSE_FORMAT</b>	Service Response Format
<b>XTL_UDFXX</b>	Merchant's User Defined Field, with <b>xx</b> having a value from 01 to 20, i.e. xtl_udf01, xtl_udf02, xtl_udf03
<b>XTL_ORDER_ID</b>	Merchant's Order ID
<b>XTL_CUST_ID</b>	Merchant's Customer ID
<b>LI_XTL_PROD_ID_X</b>	Merchant's Product ID Name. "x" indicates a dynamic number depending on the line items sent in the request. For more information see the <a href="#">Multiple Line Items</a> section.

<b>REQUEST_INITIATOR</b>	Used in Card on File (COF) transactions. Set this parameter to 'C' for Customer Initiated transactions (CIT). Set this parameter to 'M' for Merchant Initiated transactions (MIT). Default: "0"
<b>REQUEST_INSTALLMENT</b>	Set this parameter to '1' if the transaction is an installment payment. Default: "0"
<b>PMT_NUMB_COF</b>	Set this parameter to "1" to specify that a stored payment number has been used by the merchant's own system for a COF transaction. Default: "0"
<b>REQUEST_REBILL</b>	Set this parameter to "1" to specify that the transaction is a Rebill in a subscription Set this parameter to "2" to specify that the transaction is the FIRST transaction in a subscription
<b>REQUEST_XSALE</b>	Request Cross-sale transaction.
<b>PROC_GUID</b>	Processor GUID/GUWID
<b>P3DS_RETURN_URL</b>	<b>Deprecated</b> 3-D Secure Return URL after verification. <b>Deprecated</b>
<b>P3DS_PARAMS</b>	3-D Secure Verification Parameters (Payspace accounts with 3DS only) <b>Note:</b> Data returned in this field is URL-encoded.
<b>P3DS_VERIFICATION_URL</b>	<b>Deprecated</b> 3-DS Verification URL where customers should be redirected to for external verification.
<b>REQUEST_ENROLLMENT</b>	Transforms a request into a 3-D Secure enrollment check request
<b>P3DS_TRANSID</b>	Transaction ID specific to a 3-D Secure vendor
<b>P3DS_VERSION</b>	3DS Version used on transaction (2)
<b>P3DS_ECI</b>	3-D Secure ECI value
<b>P3DS_SCREEN_HEIGHT</b>	Total height of the cardholder's screen in pixels
<b>P3DS_SCREEN_WIDTH</b>	Total width of the cardholder's screen in pixels
<b>P3DS_JAVA_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents the ability of the cardholder browser to execute Java
<b>P3DS_JAVASCRIPT_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents represents the ability of the cardholder browser to execute JavaScript
<b>P3DS_BROWSER_HEADER</b>	The exact content of the HTTP accept headers sent from the cardholder's browser. Example: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
<b>P3DS_BROWSER_LANGUAGE</b>	Value represents the browser language as defined in IETF BCP47

P3DS_BROWSER_COLOR_DEPTH	Value represents the bit depth of the color palette for displaying images, in bits per pixel Possible Values: 1, 4, 8, 15, 16, 24, 32, 48
P3DS_BROWSER_TIME_ZONE	Time difference between UTC time and the cardholder browser local time, in minutes
P3DS_CHALLENGE_WINDOW	An override field that a merchant can pass in to set the challenge window size to display to the end cardholder. The ACS will reply with content that is formatted appropriately to this window size to allow for the best user experience. The sizes are width x height in pixels of the window displayed in the cardholder browser window. Possible values: 01 - 250x400, 02 - 390x400, 03 - 500x600, 04 - 600x400, 05 - Full page
REQUEST_PARES	Submitted in a 3-D Secure transaction, a value obtained from an enrollment request
REQUEST_AFF_ID_SUB	External Sub-affiliate ID
REQ_LOW_VALUE_SCAEXEMPTION	Used by Merchants to bypass 3DS on low value transactions by requesting Low-Value SCA Exemption
UNIQUE_XTL_ORDER_ID	Enables External Order ID uniqueness.
PMT_ID_XTL	External Payment /Credit Card Unique ID
PROC_UDF01	Processor Defined Field 1
PROC_UDF02	Processor Defined Field 2
PROC_SUCCESS_URL	Redirect URL after successful 3-D Secure authentication process.
PROC_ERROR_URL	Redirect URL after unsuccessful 3-D Secure authentication process.
PROC_REDIRECT_URL	3DS Authentication URL where customers should be redirected to for authentication.
PROD_NAME	Product Name
PROD_TYPE	1- Membership Cancels, 2-Membership Renews
PROD_REBILL_METRIC	M - Month D - Day Y- Year
PROD_REBILL_PERIOD	7 30 90 etc
TAX_AMT	Used to capture the tax amount charged on a transaction.
TAX_EXEMPT	A flag to indicate if a business is exempted from paying taxes or not.
TRANS_TRIAL_REBILL_CUSTOMER_CONSENT	Reports on Customer Consent for rebill
TRANS_CUSTOMER_RECEIPT	Stores Customer transaction receipt

<b>MBSHP_ID_XTL</b>	For capturing customer membership id for memberships managed by the merchant.
<b>TRANS_REBILL_TYPE</b>	Reports if rebill is of either of these types NONE, TRIAL, INITIAL, REBILL
<b>TPPE_ID</b>	Third-Party Processing Entity; this can be an e-Commerce platform, a CRM, another gateway or any other entity which transmits cardholder data on a merchant's behalf.
<b>PMT_WALLET</b>	This field shall be used to hold all the different types of wallets including e-wallet and m-wallet when available
<b>PMT_WALLET_CRYPTOGRAM</b>	Base64 URL encoded field that'll hold the payload of the authorized transaction which comes from the Apple Pay session on the payment form hosted by the merchant
<b>ORIG_CARD_BRAND_TRANSID</b>	This is the incoming card scheme transaction id from the original transaction for the card and the merchant account.

## 4.7 Membership Adjustment Parameters

When Inovio is managing your memberships/subscriptions, below are the parameters needed for any adjustments and or cancellations.

Field Name	Description
<b>REQUEST_REF_MBSHP_ID</b>	Referring Membership ID
<b>SUB_UPDATE_PROD_ID</b>	New Subscription Product ID: Used for updating the current product ID of a membership record.
<b>SUB_CANCEL_TYPE</b>	Subscription Cancel Request Type
<b>SUB_UPDATE_PMT_ID</b>	Used to Update Card used for Membership (tied to specific Membership)
<b>SUB_UPDATE</b>	Request_Action for any membership update

## 4.8 Tokenization

Rather than submitting the credit card number (**PMT\_ID**) with a transaction, a token ID may be used instead (**TOKEN\_GUID**). This unique ID is acquired by sending a request to our token service. The procedure for tokenization use is:

- 1) Submit a request to the token service (AJAX approach)
- 2) Receive the unique token ID

3) Submit a transaction request to the gateway, using **TOKEN\_GUID**

**! - The token is unique, and for one-time use. A new token must be requested for each transaction.**

In order to utilize the enhanced tokenization parameters, please contact your gateway support representative to obtain your secret key.

### 4.8.1 Requesting the token

The request for the unique token ID should be sent to the following URL:

[https://api.inoviopay.com/payment/token\\_service.cfm?](https://api.inoviopay.com/payment/token_service.cfm?)

#### 4.8.1.1 Header Request Parameters

Field Name	Description	Additional Notes	Versioning
x-timestamp	The timestamp is open for five minutes (300 seconds) before the current time. Format: YYYYmmddHHMMSS	HH: Uses 24 hour clock HHMMSS: uses UTC timezone	4.9+
x-signature	Client's generated HMAC_SHA256 Base16 signature <x-timestamp><unique_id><site_id>	Generated using x-timestamp, unique_id, card_pan and site_id with the secret key assigned by Client Support.	4.9+

#### 4.8.1.2 Body Request Parameters

Field Name	Description	Additional Notes	Versioning
card_pan	Customer card number to be tokenized		All
request_response_format	XML or JSON	Optional: Default to JSON	All
request_api_version	API Version of this document.		4.9+
site_id	Merchant's website ID		4.9+
unique_id	Alphanumeric ID linked to the request.	Generated by the client. Max of 32 characters.	4.9+

#### Token Request Example

[https://api.inoviopay.com/payment/token\\_service.cfm?CARD\\_PAN=4111111111111111&REQUEST\\_RESPONSE\\_FORMAT=json](https://api.inoviopay.com/payment/token_service.cfm?CARD_PAN=4111111111111111&REQUEST_RESPONSE_FORMAT=json)

4.9+ version example: Secret key is Password123

```
https://api.inoviopay.com/payment/token_service.cfm?CARD_PAN=4012345678901234&REQUEST_API_VERSION=4.9&UNIQUE_ID=32123456789012345678901234567890&SITE_ID=321&X-SIGNATURE=cc9ff3991d0570db933365d5d44ccb14df535fe5b6e3d506014a3c50881c0ae7&X-TIMESTAMP=20241209230900&response_request_format=json
```

#### 4.8.1.3 Token Response Parameters

##### 4.8.1.3.1 Header Response Fields

Field Name	Description	Additional Notes	Versioning
x-timestamp	Format: YYYYMMDDHHMMSS	HH: Uses 24 hour clock HHMMSS: uses UTC timezone	4.9+

<b>x-signature</b>	Inovio's generated HMAC_SHA256 Base16 signature <x-timestamp><token_reqid><JSON body response>	Generated using x-timestamp, token reqid, and the full raw JSON body response including brackets with the secret key being assigned by Client Support.	4.9+
--------------------	------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------	------

#### 4.8.1.3.2 Body Response Fields

Field Name	Description	Additional Notes	Versioning
<b>token_guid</b>	Unique token ID which should subsequently be sent in the authorization request		All
<b>token_ip</b>	IP address of token request		All
<b>token_reqid</b>	Incremental ID of the token request		All
<b>card_brand_name</b>	Credit Card Network/Brand Name		4.9+
<b>card_type</b>	Indicates the card product sub category	i.e.: VISA Business	4.9+
<b>card_bank</b>	Issuer Bank name for the BIN		4.9+
<b>card_country</b>	Issuer Bank country for the BIN	Three character alpha country code	4.9+
<b>card_account_fund_source</b>	Identifies the source of the funds associated with the primary account for the card		4.9+
<b>card_class</b>	Categorizes the BIN as a Business, Corporate, Purchase, or Consumer card		4.9+

#### Token Response Example

```
{
  "TOKEN_GUID": "7BA39EAFDAAD6B3FA8A974098A267258E6D622D9",
  "TOKEN_IP": "10.13.100.134",
  "TOKEN_REQID": "4283012"
}
```

4.9+ version example:

X-SIGNATURE: 35dab7703e959d363da14674a4f042b19b025c696da7af1325fb630bd342bae1

X-TIMESTAMP: 20241209230307

```
{
  "TOKEN_GUID": "7BA39EAFDAAD6B3FA8A974098A267258E6D622D9",
  "CARD_BRAND_NAME": "VISA",
  "CARD_TYPE": "VISA TRADITIONAL",
  "CARD_BANK": "BANK OF Inovio",
```

```
"CARD_COUNTRY": "USA",
"CARD_ACCOUNT_FUND_SOURCE": "Credit",
"CARD_CLASS": "CONSUMER",
"TOKEN_IP": "10.13.100.134",
"TOKEN_REQID": "4283012"
}
```

#### 4.8.2 CCAUTHCAP Request Using Tokenization

Example:

```
https://api.inovioipay.com/payment/pmt_service.cfm?request_action=CCAUTHCAP&request_api_version=4.8&req_username=test@example.com&req_password=Example0905&site_id=1111&request_response_format=JSON&li_value_1=10&li_prod_id_1=1001&TOKEN_GUID=7BA39EAFDAAD6B3FA8A974098A267258E6D622D9&PMT_KEY=123&PMT_EXPIRY=082025&MERCH_ACCT_ID=100&CUST_FNAME=Master&CUST_LNAME=Customer&BILL_ADDR=123MainSt.&BILL_ADDR_CITY=LosAngeles&BILL_ADDR_STATE=CA&BILL_ADDR_ZIP=90032&BILL_ADDR_COUNTRY=US&CUST_EMAIL=test@test.com&REQUEST_CURR_ENCY=USD
```

## 5 Credit Card Transactions

### 5.1 Authorization

*Authorization* request is used to confirm the availability of funds in the cardholder's bank account. This type of transaction places a temporary hold or pending "auth" in the cardholder's bank account and does not guarantee payment. For this type of transaction, merchants must send service request action, "CCAUTHORIZE", in the transaction request.

*Authorization* request is the first part of a two-stage process of "authorizing" and "capturing" funds. This two stage process is commonly used by merchants who have to do partial fulfillments of orders. In order to capture the *Authorization* request, use Service Request Action, "CCCAPTURE" (see [Section 5.2](#) for instructions).



**Important**

*Please allow a wait time of 120 seconds for CCAUTHORIZE, CCAUTHCAP and other requests. Even though the gateway usually adds only less than a second, downstream processors can take much longer to get a response from issuing banks.*

*Note that if the merchant chooses to time out their request earlier than 120 Seconds, then:*

- a. *The gateway is not responsible for lost transactions, transaction records and other failed functions*
- b. *The merchant may choose to use [CCSTATUS](#) at a later time to find out the final outcome of their request (XTL\_ORDER\_ID would have to have been provided in the original request)*
- c. *Webhooks/postbacks may or may not trigger at the time that the request is finalized at the gateway, and should not be relied upon.*

*Contact gateway support if you have questions.*

#### 5.1.1 Authorization Parameters

The table below only includes typical parameters sent in a simple Authorization transaction request. Note that merchants can send additional parameters listed in [Section 4](#) of this specification.

Field Name	Description	Data Type	Requirement
------------	-------------	-----------	-------------

<b>REQUEST_ACTION</b>	Service Request Action (Send “CCAUTHORIZE”).	<a href="#">Service Request Types</a>	Required
<b>REQ_USERNAME</b>	Service Request Username	Alphanumeric and Special Characters	Required
<b>REQ_PASSWORD</b>	Service Request Password	Alphanumeric and Special Characters	Required
<b>REQUEST_RESPONSE_FORMAT</b>	Service Response Format	Accepted values: “XML”, “PIPES” and “JSON”	Optional (default: XML)
<b>REQUEST_API_VERSION</b>	API Version	Numeric	Required
<b>SITE_ID</b>	Merchant’s Website ID	Numeric	Required
<b>CUST_FNAME</b>	Cardholder’s First Name	Alphanumeric and Special Characters	Optional
<b>CUST_LNAME</b>	Cardholder’s Last Name	Alphanumeric and Special Characters	Optional
<b>CUST_EMAIL</b>	Cardholder’s Email Address	Alphanumeric and Special Characters	Optional
<b>LI_COUNT_1</b>	Line Item Count Max value is “99”.	Numeric	Required
<b>LI_PROD_ID_1</b>	Line Item Product ID 1	Numeric	Required
<b>LI_VALUE_1</b>	Line Item Transaction Amount 1	Numeric	Required
<b>XTL_ORDER_ID</b>	Merchant’s Order ID	Alphanumeric	Optional
<b>BILL_ADDR</b>	Cardholder Billing Street Address	Alphanumeric	Optional (address fields may be required by the bank)
<b>BILL_ADDR_CITY</b>	Cardholder’s Billing City	Alphanumeric	Optional (address fields may be required by the bank)
<b>BILL_ADDR_STATE</b>	Cardholder’s Billing State	2-letter State or Territory Code	Optional (some processors may require a valid 2-letter state or territory code)

<b>BILL_ADDR_ZIP</b>	Cardholder's Billing Postal/ZIP code	Alphanumeric	Optional (address fields may be required by the bank)
<b>BILL_ADDR_COUNTRY</b>	Cardholder's Billing Country	2-letter Country Code ISO 3166-1 alpha-2	Optional (address fields may be required by the bank)
<b>PMT_NUMB</b>	Credit Card Number	Numeric	Required
<b>TOKEN_GUID</b>	Token ID	Alphanumeric	Token ID used in place of pmt_numb. See <a href="#">Tokenization</a> .
<b>PMT_KEY</b>	Credit Card CVV2 or CVC2 Code	Numeric (4)	Required
<b>PMT_EXPIRY</b>	Credit Card Expiration Date	Numeric MMYYYY Example: "122014"	Required
<b>CUST_LOGIN</b>	Cardholder's Login or User Name	Alphanumeric and Special Characters	Optional
<b>CUST_PASSWORD</b>	Cardholder's Password Password must be at least 10 characters with 1 number, lower case and upper case letter.	Alphanumeric	Optional
<b>MERCH_ACCT_ID</b>	Merchant Account ID	Numeric	If null, the system will follow merchant's bank settings.
<b>REQUEST_CURRENCY</b>	3-letter Currency Code	Example: USD	Required
<b>PMT_DESCRIPTOR</b>	Dynamic Descriptor	Alphanumeric	Optional
<b>PMT_DESCRIPTOR_PHONE</b>	Bank Dynamic Customer Support Phone Number	Numeric	Optional
<b>CUST_PHONE</b>	Cardholder's Phone Number	Numeric	Optional
<b>REQUEST_AFF_ID</b>	External Affiliate ID	Alphanumeric	Optional
<b>REQUEST_AFF_ID_SUB</b>	External Sub-affiliate ID	Alphanumeric	Optional

<b>UNIQUE_XTL_ORDER_ID</b>	Enforces External Order ID uniqueness.	Possible values: “0” – disables flag “1” – decline request “2” – returns Approval response	Optional
<b>PMT_ID_XTL</b>	External Payment Unique Identifier	Alphanumeric (max length: 64)	Optional
<b>MBSHP_ID_XTL</b>	External Membership ID	Alphanumeric	Optional (this will be required if the TRANS_REBILL_TYPE is not NONE)
<b>TRANS_REBILL_TYPE</b>	Rebill Type	One of NONE, TRIAL, INITIAL, REBILL	Optional
<b>REQUEST_INITATOR</b>	CIT (C) and MIT (M)	Used to identify transaction origination	Optional
<b>REQUEST_INSTALLMENT</b>	0 (not used), 1 (used)	Used to denote if the rebill is an installment with start and end date.	Optional
<b>PMT_NUMB_COF</b>	0 (not used), 1 (used)	Used to denote use of payment number stored by Merchant	Optional
<b>REQUEST_REBILL</b>	1 (yes for rebill), 2(Start Subscription), not used	Used to denote if the transaction is subscription card storage, a rebill or not used at all	Optional
<b>TRANS_TRIAL_REBILL_CUSTOMER_CONSENT</b>	Customer Consent for rebill after trial period	encoded character content	Required for the _first_ transaction of type REBILL following a transaction of type TRIAL on the same MBSHP_ID_XTL. Applies to MCC 5968 and Card_Brand_Name= Mastercard.
<b>TRANS_CUSTOMER RECEIPT</b>	Payment Receipt	should accept encoded character content either freeform or html or xml content representing the actual receipt document sent to the customer	Optional but subsequent transactions on same MBSHP_ID_XTL unless previous trans are received. Applies to Card_Brand_Name = VISA. Also applies to

			MCC 5968 and Card_Brand_Name = Mastercard
CARD_ON_FILE_FLAG	Flags if COF or New PMT Card	possible values are 0 (COF) or 1(New)	Optional

### 5.1.2 Authorization Request Example

```
POST /payment/pmt_service.cfm HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
Accept-Encoding: identity
Accept-Language: en-us,en
Host: Host.example.com
Accept-Charset: iso-8859-1,*;utf-8
Content-Type: application/x-www-form-urlencoded
Content-Length: 531
req_password=Test1234567&site_id=0&cust_fname=Daenarys&bill_addr_city=Los%20Angeles&b
ill_addr_state=CA&pmt_expiry=10%2F2020&xtl_cust_id=testcust11&pmt_key=123&request_res
ponse_format=XML&pmt numb=4111111111111111&cust_lname=Targaryen&request_api_version=4
.8&cust_email=useremail%40tests.com&li_value_1=1.25&req_username=merchant100%40exampl
e.com&li_prod_id_1=1001&request_currency=USD&xtl_order_id=testorder117&bill_addr_zip=
90401&request_action=CCAUTHORIZE&merch_acct_id=100&bill_addr=1 Main St
&bill_addr_country=US&pmt_id_xtl=ABCD12345&mbship_id_xtl=888920&card_on-
file_flag=0&trans_rebill_type=rebill
```

### 5.1.3 Authorization Response Example

```
<RESPONSE>
<REQUEST_ACTION>CCAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>1.25</TRANS_VALUE>
<TRANS_VALUE_SETTLED>1.25</TRANS_VALUE_SETTLED>
<CURRE_CODE_ALPHA_SETTLED>USD</CURRE_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>989898</TRANS_ID>
<CUST_ID>212121</CUST_ID>
<XTL_CUST_ID/>
<REQ_ID>30814993221</REQ_ID>
<PO_ID>17171717</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>3114</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_TYPE>VISA BUSINESS CARD</CARD_TYPE>
<PMT_AAU_UPDATE_DT>2019-07-17</PMT_AAU_UPDATE_DT>
<PMT_AAU_UPDATE_DESC>Account Number Change Message</PMT_AAU_UPDATE_DESC>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_PREPAID>1</CARD_PREPAID>
<CARD_BANK>BANK OF EUROPE</CARD_BANK>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_DETAIL>Credit</CARD_DETAIL>
```

```

<CARD_BALANCE/>
<PMT_L4>111</PMT_L4>
<PMT_ID>101011</PMT_ID>
<PMT_ID_XTL/>
<PROC_UDF01/>
<PROC_UDF02/>
<PROC_AUTH_RESPONSE>72682</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>74C8C7C8-5652-48BB-A68BGRACE46A2D1</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<PROC_REDIRECT_URL/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>8509920</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>1.25</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>1001</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>

```

#### 5.1.4 Failed Authorization Attempt Response Example

```

<RESPONSE>
<REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
<TRANS_STATUS_NAME>DECLINED</TRANS_STATUS_NAME>
<TRANS_VALUE>5.01</TRANS_VALUE>
<TRANS_ID>20005</TRANS_ID>
<CUST_ID>2</CUST_ID>
<XTL_CUST_ID>c777777777</XTL_CUST_ID>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Mastercard</CARD_BRAND_NAME>
<PMT_L4>5100</PMT_L4>
<PMT_ID>100002</PMT_ID>
<PMT_ID_XTL/>
<API_RESPONSE>0</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE>640</SERVICE_RESPONSE>
<SERVICE_ADVICE>Retry</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>501</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE>Retry</PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>0</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<REQ_ID>30814993200</REQ_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

#### 5.1.5 Service Declined Response Example

```

<RESPONSE>
<REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
<TRANS_STATUS_NAME/>
<CUST_ID/>

```

```

<XTL_CUST_ID>c777777777</XTL_CUST_ID>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME/>
<PMT_L4/>
<PMT_ID>440002</PMT_ID>
<PMT_ID_XTL/>
<API_RESPONSE>113</API_RESPONSE>
<API_ADVICE>Invalid Data</API_ADVICE>
<SERVICE_RESPONSE>o</SERVICE_RESPONSE>
<SERVICE_ADVICE>Declined</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>o</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>o</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD>cust_password</REF_FIELD>
<REQ_ID>30814993201</REQ_ID>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

The example above was declined by the API due to an invalid data in the CUST\_PASSWORD field (see **API\_ADVICE** and **REF\_FIELD** parameters). Service-declined requests will also have the “Declined” message in the **SERVICE\_ADVICE** field.

## 5.2 Delayed Capture

As described in [Section 5.1: Authorization](#), an *Authorization* does not guarantee payment. In order to capture the pending authorization, merchants must send a capture request of the *Authorization* to the Payment Service.

To send a *Delayed Capture* request, merchants must send “CCCAPTURE” in the **request\_action** parameter. The Payment Service allows merchants to send multiple partial amount capture requests or “partial captures” against a single *Authorization*, as long as the original amount of the *Authorization* is not exceeded.

### 5.2.1 Delayed Capture Request

Send a capture request by keying off the Order ID (PO\_ID) of the original authorization. Send the Order ID in the **REQUEST\_REF\_PO\_ID** parameter.

**CCCAPTURE Required Fields**

SITE_ID
REQ_USERNAME
REQ_PASSWORD
<b>REQUEST_ACTION = CCCAPTURE</b>
REQUEST_RESPONSE_FORMAT
<b>REQUEST_REF_PO_ID</b>
LI_VALUE_1
REQUEST_API_VERSION

**Important Note**

MERCHANTS may send a different amount in the LI\_VALUE\_1 parameter. However, the amount may not exceed the original authorization's total amount.

### 5.3 Authorization and Capture (Sale)

The difference between an *Authorization* only and *Authorization and Capture* request is that the latter will also request capture of the authorized funds in a single request to the Payment Service.

Use service request action, "CCAUTHCAP" for *Authorization and Capture* requests. The fields for CCAUTHCAP requests are the same as CCAUTHORIZE requests described in [Section 5.1.1: Authorization Parameters](#).

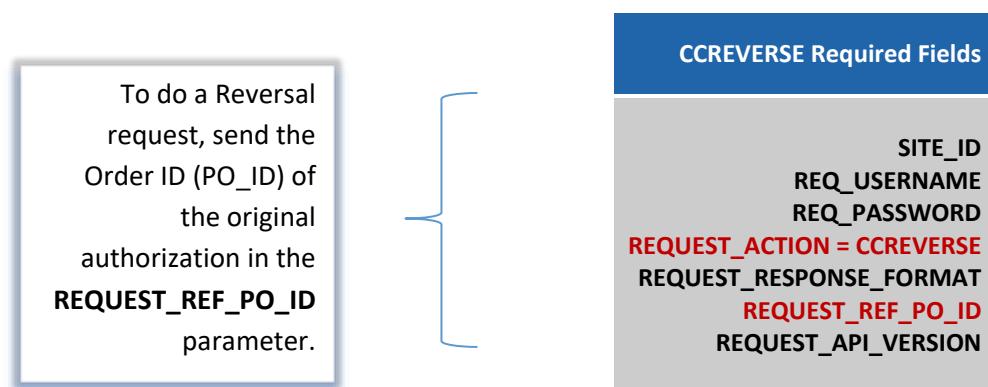
#### 5.3.1 Authorization Request Example (CCAUTHCAP)

```
POST /payment/pmt_service.cfm HTTP/1.0
User-Agent: Mozilla/4.0 (compatible; MSIE 11.0; Windows NT 5.1)
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash,
application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */
Accept-Encoding: identity
Accept-Language: en-us,en
Host: Host.example.com
Accept-Charset: iso-8859-1,*;utf-8
Content-Type: application/x-www-form-urlencoded
Content-Length: 53
req_username=test@example.com&req_password=Password!1&request_action=CCAUTHCAP&site_id=0&merch_acct_id=100&cust_fname=John&cust_lname=Doe&cust_email=user5@example.com&cust_login=username1&cust_password=12345678Xx&xtl_cust_id=c777777777&xtl_order_id=o111111111&li_prod_id_1=1001&li_value_1=19.95&li_count_1=1&bill_addr=123 Main Street Apt. 1&bill_addr_city=Los Angeles&bill_addr_state=CA&bill_addr_zip=90066&bill_addr_country=US&pmt_num=5105105105105100&pmt_key=123&pmt_expiry=12/2025&request_response_format=XML&xtl_ip=10.00.00.0.90&request_api_version=4.8
```

### 5.4 Reversal

#### 5.4.1 Reversing or voiding an Authorization (CCREVERSE).

MERCHANTS may request to reverse the original authorization by sending CCREVERSE in the request\_action parameter.

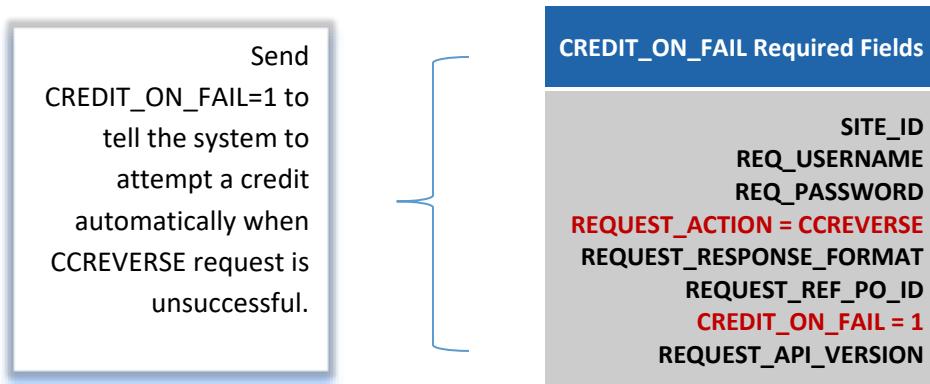


**Important Note**

Partial amount reversal requests are not supported by all processors. Please contact [Client Services](#) before implementing partial amount requests.

### 5.4.2 CREDIT\_ON\_FAIL Flag

In any case when the reversal request has failed, the system can automatically attempt a credit without the merchant sending a separate credit request. To enable this feature, merchants must set the CREDIT\_ON\_FAIL parameter to “1” and send this parameter with the CCREVERSE request. Set this field to null to disable.



The merchant should expect a CCREVERSE response if the reversal request is successful; if not successful, the API will return a CCCREDIT response instead.

#### 5.4.2.1 Example Successful CCREVERSE Response

```
<RESPONSE>
<REQUEST_ACTION>CCREVERSE</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>-1.22</TRANS_VALUE>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>-1.22</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>69462</TRANS_ID>
<CUST_ID>1547</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>261148</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>1102</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_TYPE>VISA BUSINESS</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>0</CARD_PREPAID>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>1111</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL/>
```

```

<PROC_UDFO1/>
<PROC_UDFO2/>
<PROC_AUTH_RESPONSE>74282</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>TESTRETRIEVE123</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>300383</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>-1.22</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>1001</PO_LI_PROD_ID_1>
</RESPONSE>

```

#### 5.4.2.2 Example Successful CCCREDIT Response

```

<RESPONSE>
<REQUEST_ACTION>CCCRECREDIT</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>-1.22</TRANS_VALUE>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>-1.22</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>21089</TRANS_ID>
<CUST_ID>1503</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>215257</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>1000</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_TYPE>VISA CLASSIC</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>1</CARD_PREPAID>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>2244</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL/>
<PROC_UDFO1/>
<PROC_UDFO2/>
<PROC_AUTH_RESPONSE>33086</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>B51URX2-6GRACE123</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>251038</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>-1.22</PO_LI_AMOUNT_1>

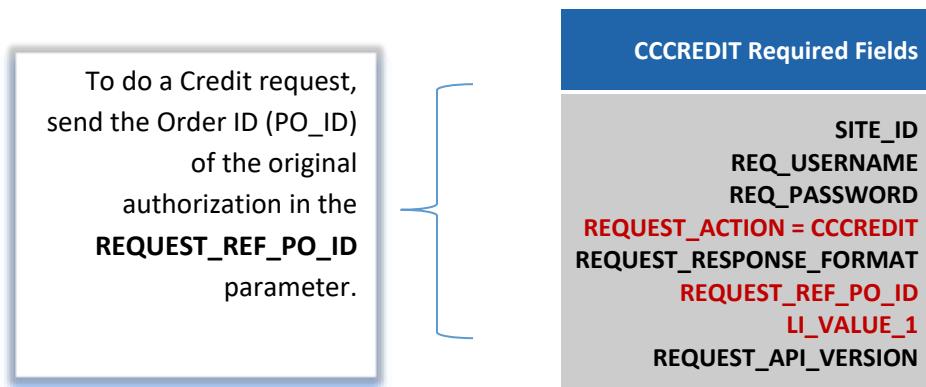
```

```
<PO_LI_PROD_ID_1>1001</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>
```

## 5.5 Credit

### 5.5.1 Minimum required parameters for issuing a Credit (CCCRECREDIT):

MERCHANTS may request to return a captured authorization by sending CCCREDIT in the request\_action parameter.



#### *Important Note*

MERCHANTS may only credit transactions that have been captured or settled.

A different amount from the original authorization maybe sent in the LI\_VALUE\_1 parameter. However, the amount may not exceed the original authorization's total amount. The exception is a force credit.

### 5.5.2 Force Credit Request

Force Credit is a type of transaction where the credit request is sent directly to the settlement file. MERCHANTS do not need to send the REQUEST\_REF\_PO\_ID parameter on this type of request. However, the FORCE\_CREDIT parameter is required, together with the full credit card information.

#### *Important Note*

Force Credits are only available to certain Merchant Accounts. Please contact your gateway support representative for more information.

### 5.5.3 Multiple Line Items

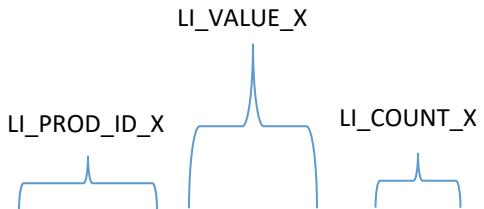
**Note:** If your business does not intend to support purchase orders with more than one line item. Please skip this section. For more information, contact your gateway support representative.

The Payment Service supports purchase orders with more than one line item information. This feature will be useful for merchants using shopping cart like business models.

MERCHANTS may use this feature by sending the Line Item Parameters: LI\_PROD\_ID\_X, LI\_COUNT\_X, and LI\_VALUE\_X. "X" indicates that this value is dynamic and should be replaced by a number

depending on how many line items the merchant wants to send (i.e. for 2 line items the request should contain: LI\_VALUE\_1, LI\_PROD\_ID\_1, LI\_COUNT\_1, LI\_VALUE\_2, LI\_PROD\_ID\_2, and LI\_COUNT\_2).

#### 5.5.4 Example CCAUTHORIZE request with Multiple Line Items.



Item	Prod ID	Unit Price	Qty.	Total
Product Name 1	# 1001	19.95	1	19.95
Product Name 2	# 2001	10.50	1	10.50

The request for the shopping items above should look like:

```
/pmt_service.cfm?req_username=test@example.com&req_password=TestPassword1&request_action=CCAUTHORIZE&request_api_version=4.8&site_id=0&merch_acct_id=100&cust_fname=John&cust_lname=Doe&cust_email=test@example.com&cust_login=username1&cust_password=12345678Xx&bill_addr=1 Main Street&bill_addr_city=Hollywood&bill_addr_state=CA&bill_addr_zip=90078&bill_addr_country=US&pmt numb=5105105105105100&pmt_key=123&pmt_expiry=12/2025&request_response_form at=XML&xltl_ip=10.00.000.99&li_prod_id_1=1001&li_value_1=19.95&li_count_1=1&li_prod_id_2=2001&li_value_2=10.50&li_count_2=1
```

**Note**

Replace “CCAUTHORIZE” with “CCAUTHCAP” in the request above to do a CCAUTHCAP request instead of CCAUTHORIZE.

#### 5.5.5 Example CCAUTHORIZE Response with Multiple Line Items

The Line Item response parameters below are highlighted in gray.

```
<RESPONSE>
<REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE>30.45</TRANS_VALUE>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>30.45</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>30000</TRANS_ID>
<CUST_ID>2000</CUST_ID>
```

```

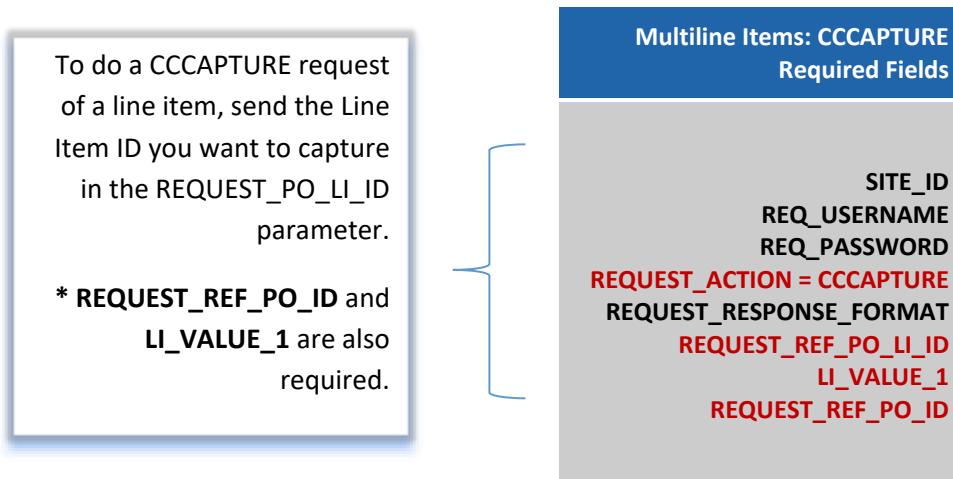
<XTL_CUST_ID>
<PO_ID>77777</PO_ID>
<XTL_ORDER_ID>
<BATCH_ID>1001</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_TYPE>VISA BUSINESS</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>o</CARD_PREPAID>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>1111</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL/>
<PROC_UDF01/><PROC_UDF02/>
<PROC_AUTH_RESPONSE>51402</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>7169CEF3-64EF-46BB-55A333956994D2D</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<REQ_ID>30814003200</REQ_ID>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>123455</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>19.95</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>1001</PO_LI_PROD_ID_1>
<PO_LI_ID_2>123456</PO_LI_ID_2>
<PO_LI_COUNT_2>1</PO_LI_COUNT_2>
<PO_LI_AMOUNT_2>10.5</PO_LI_AMOUNT_2>
<PO_LI_PROD_ID_2>2001</PO_LI_PROD_ID_2>
<MBSHP_ID_1/>
</RESPONSE>

```

LINE ITEM

RESPONSE

### 5.5.6 Example Request for Capturing a Specific Line Item



**MERCHANTS SHOULD ONLY USE THE REQUEST\_REF\_PO\_LI\_ID WHEN SENDING A DELAYED CAPTURE REQUEST AGAINST A SPECIFIC LINE ITEM.**

## 5.5.7 Capturing, Reversing or Crediting the Entire Order with Multiple Line Items

To capture, reverse, or credit the entire purchase order, just send the correct REQUEST\_ACTION and REQUEST\_REF\_PO\_ID with other required parameters in the API request.

### 5.5.7.1 Multiline Items: CCCAPTURE Example Request

Note that this request will capture the second line item from the CCAUTHORIZE example above:

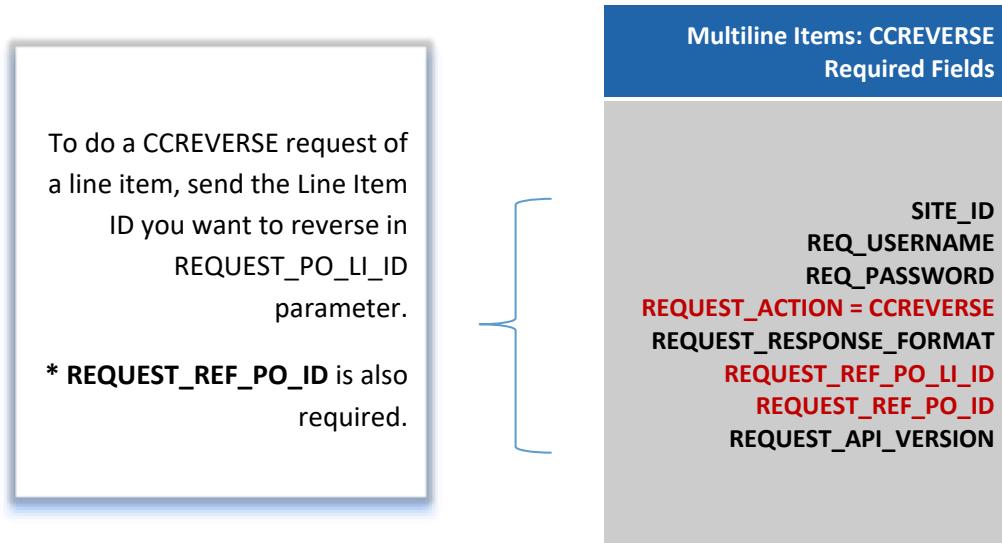
```
/pmt_service.cfm?req_password=TestPassword1&site_id=0&request_ref_po_li_id=123456&request_ref_po_id=77777&chkavvs=F&request_response_format=JSON&req_username=test%40example.com&li_value_1=10.50&request_action=CCCAPTURE&request_api_version=4.8
```

### 5.5.7.2 Multiline Items: CCCAPTURE Example Approved Response

```
<RESPONSE>
<REQUEST_ACTION>CCCAPTURE</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>10.5</TRANS_VALUE>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>10.5</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE>
<TRANS_ID>124447</TRANS_ID>
<CUST_ID>2000</CUST_ID>
<XTL_CUST_ID>
<PO_ID>77777</PO_ID>
<XTL_ORDER_ID>
<BATCH_ID>1101</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_TYPE>VISA BUSINESS</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>0</CARD_PREPAID>
<CARD_BANK>
<CARD_BALANCE>
<PMT_L4>1111</PMT_L4>
<PMT_ID>
<PMT_ID_XTL>
<PROC_UDF01><PROC_UDF02>
<PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>9287005b-0740-4049-8526-9dcc6f06e610</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM>5478723</PROC_REFERENCE_NUM>
<AVS_RESPONSE>
<CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>123456</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>10.5</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>2001</PO_LI_PROD_ID_1>
<MBSHP_ID>
```

&lt;/RESPONSE&gt;

## 5.5.8 Example Request for Sending Reversal of a Specific Line Item



### 5.5.8.1 Multiline Items: CCREVERSE Example Request

```
/pmt_service.cfm?req_password=TestPassword1&site_id=0&request_ref_po_li_id=123456&request_ref_po_id=77777&chkavs=F&request_response_format=XML&req_username=test%40example.com&li_value_1=10.50&request_api_version=4.8&request_action=CCREVERSE
```

### 5.5.8.2 Multiline Items: CCREVERSE Example Response

```
<RESPONSE>
<REQUEST_ACTION>CCREVERSE</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>-10.50</TRANS_VALUE>
<CURREN_CODE_ALPHA>USD</CURREN_CODE_ALPHA>
<TRANS_VALUE_SETTLED>-10.50</TRANS_VALUE_SETTLED>
<CURREN_CODE_ALPHA_SETTLED>USD</CURREN_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE></TRANS_EXCH_RATE>
<TRANS_ID>100002</TRANS_ID>
<CUST_ID>2000</CUST_ID>
<XTL_CUST_ID></XTL_CUST_ID>
<PO_ID>77777</PO_ID>
<XTL_ORDER_ID></XTL_ORDER_ID>
<BATCH_ID>1001</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_TYPE>VISA BUSINESS</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>o</CARD_PREPAID>
<CARD_BANK></CARD_BANK>
<CARD_BALANCE></CARD_BALANCE>
```

```

<PMT_L4>5100</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL/>
<PROC_UDFO1/><PROC_UDFO2/>
<PROC_AUTH_RESPONSE>69803</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>31B22CB-A453-GRA3CE5</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>123456</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>-10.50</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>2001</PO_LI_PROD_ID_1>
<MBSHP_ID>
</RESPONSE>

```

#### 5.5.8.3 Crediting a specific line item

##### 5.5.8.4

To do a CCCREDIT request of a line item, send the Line Item ID you want to credit in REQUEST\_PO\_LI\_ID parameter.

\* REQUEST\_REF\_PO\_ID and LI\_VALUE\_1 are also required.

*Simple Request*

#### Multiline Items: CCCREDIT Required Fields

SITE_ID
REQ_USERNAME
REQ_PASSWORD
REQUEST_API_VERSION
<b>REQUEST_ACTION = CCCREDIT</b>
REQUEST_RESPONSE_FORMAT
REQUEST_REF_PO_LI_ID
LI_VALUE_1
REQUEST_REF_PO_ID

```
/pmt_service.cfm?req_password=TestPassword1&site_id=0&request_ref_po_li_id=123456&request_ref_po_id=77777&chkavsf=F&request_response_format=XML&req_username=test%40example.com&li_value_1=10.50&request_api_version=4.8&request_action=CCREDIT
```

#### 5.5.8.5 Multiline Items: CCCREDIT Example Response

```

<RESPONSE>
<REQUEST_ACTION>CCREDIT</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>-10.5</TRANS_VALUE>
<TRANS_ID>100002</TRANS_ID>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>-10.5</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<CUST_ID>2000</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>77777</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>1001</BATCH_ID>

```

```

<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_TYPE>VISA BUSINESS</CARD_TYPE>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_DETAIL>Credit</CARD_DETAIL>
<CARD_PREPAID>o</CARD_PREPAID>
<CARD_BANK>BANK OF TEST BANKS</CARD_BANK>
<CARD_BALANCE/>
<PMT_L4>5100</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL/>
<PROC_UDF01/><PROC_UDF02/>
<PROC_AUTH_RESPONSE>69803</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>31B22CB-A453-GRA3CE5</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>123457</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>10.5</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>2001</PO_LI_PROD_ID_1>
<MBSHP_ID/>
</RESPONSE>

```

A new Line Item ID is created for credit transactions.

### 5.5.9 Line Item Types

The Line Item Type is not returned in the gateway API response, however, this data will be available on some of the client reports. The table below lists different type of line items:

Line Item Type ID	Description
1	Goods and Services
2	Discount
3	Tax
4	Shipping
5	Chargeback
6	Refund

### 5.6 Checking the Status of an order

To check the status of an order, merchants may set the REQUEST\_ACTION parameter to "CCSTATUS" along with the Order ID in the REQUEST\_REF\_PO\_ID parameter.

If REQUEST\_REF\_PO\_ID is not available, merchants may use the XTL\_ORDER\_ID that was sent in the original authorization instead. Send this value in the REQUEST\_REF\_PO\_ID\_XTL parameter.

### 5.6.1 CCSTATUS Example Request using REQUEST\_REF\_PO\_ID

```
/pmt_service.cfm?req_password=P5ssword1&site_id=1234&request_ref_po_id=1234567&request_response_format=JSON&req_username=test1%40example.net&request_api_version=4.8&request_action=CCSTATUS
```

### 5.6.2 CCSTATUS Example Request using REQUEST\_REF\_PO\_ID\_XTL

```
/pmt_service.cfm?req_password=
P5ssword1&site_id=1234&request_ref_po_id_xtl=order123&request_response_format=JSON&req_username=test1%40example.net&request_api_version=4.8&request_action=CCSTATUS
```

### 5.6.3 CCSTATUS Example Response

The response example below shows that the order was captured and then reversed. Merchants may refer to the **TRANS\_STATUS\_NAME** in checking the status of a specific transaction request.

```
<RESPONSE>
<TRANSACTION TRANS_ID="43500">
  <REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
  <TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
  <TRANS_VALUE>10</TRANS_VALUE>
  <CUST_ID>1500</CUST_ID>
  <XTL_CUST_ID/>
  <PO_ID>1234567</PO_ID>
  <XTL_ORDER_ID>order123</XTL_ORDER_ID>
  <BATCH_ID>1001</BATCH_ID>
  <PROC_NAME>Test Processor</PROC_NAME>
  <MERCH_ACCT_ID>100</MERCH_ACCT_ID>
  <CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
  <PMT_L4>3762</PMT_L4>
  <PROC_UDF01/>
  <PROC_UDF02/>
  <PROC_AUTH_RESPONSE>46177</PROC_AUTH_RESPONSE>
  <PROC_RETRIEVAL_NUM>56789AB</PROC_RETRIEVAL_NUM>
  <PROC_REFERENCE_NUM/>
  <AVS_RESPONSE>M</AVS_RESPONSE>
  <CVV_RESPONSE>M</CVV_RESPONSE>
</TRANSACTION>
<TRANSACTION TRANS_ID="43502">
  <REQUEST_ACTION>CCREVERSE</REQUEST_ACTION>
  <TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
  <TRANS_VALUE>-10</TRANS_VALUE>
  <CUST_ID>1500</CUST_ID>
  <XTL_CUST_ID/>
  <PO_ID>1234567</PO_ID>
  <XTL_ORDER_ID>order123</XTL_ORDER_ID>
  <BATCH_ID>1001</BATCH_ID>
  <PROC_NAME>Test Processor</PROC_NAME>
  <MERCH_ACCT_ID>100</MERCH_ACCT_ID>
  <CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
  <CARD_DETAIL>DEBIT</CARD_DETAIL>
  <CARD_TYPE>VISA BUSINESS</CARD_TYPE>
  <CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
  <CARD_COUNTRY>CRI</CARD_COUNTRY>
  <CARD_PREPAID>0</CARD_PREPAID>
  <CARD_BANK/>
```

```

<REQ_ID>304993200</REQ_ID>
<PMT_L4>3762</PMT_L4>
<PMT_ID>43333</PMT_L4>
<PMT_ID_XTL/>
  <PROC_UDF01/>
  <PROC_UDF02/>
    <PROC_AUTH_RESPONSE>62088</PROC_AUTH_RESPONSE>
    <PROC_RETRIEVAL_NUM>123456789A</PROC_RETRIEVAL_NUM>
    <PROC_REFERENCE_NUM/>
    <AVS_RESPONSE>M</AVS_RESPONSE>
    <CVV_RESPONSE>M</CVV_RESPONSE>
  </TRANSACTION>
<TRANSACTION TRANS_ID="43501">
  <REQUEST_ACTION>CCCAPTURE</REQUEST_ACTION>
  <TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
  <TRANS_VALUE>10</TRANS_VALUE>
  <CUST_ID>1500</CUST_ID>
  <XTL_CUST_ID/>
  <PO_ID>1234567</PO_ID>
  <XTL_ORDER_ID>order123</XTL_ORDER_ID>
  <BATCH_ID>1001</BATCH_ID>
  <PROC_NAME>Test Processor</PROC_NAME>
  <MERCH_ACCT_ID>100</MERCH_ACCT_ID>
  <CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
  <CARD_DETAIL>DEBIT</CARD_DETAIL>
  <CARD_TYPE>VISA BUSINESS</CARD_TYPE>
  <CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
  <CARD_COUNTRY>CRI</CARD_COUNTRY>
  <CARD_PREPAID>0</CARD_PREPAID>
  <CARD_BANK/>
    <PMT_L4>3762</PMT_L4>
    <PROC_UDF01/>
    <PROC_UDF02/>
      <PROC_AUTH_RESPONSE>63106</PROC_AUTH_RESPONSE>
      <PROC_RETRIEVAL_NUM>F56789</PROC_RETRIEVAL_NUM>
      <PROC_REFERENCE_NUM/>
      <AVS_RESPONSE>M</AVS_RESPONSE>
      <CVV_RESPONSE>M</CVV_RESPONSE>
    </TRANSACTION>
</RESPONSE>

```

#### 5.6.4 CCSTATUS Response Fields

- TRANS\_ID – Unique transaction ID.
- REQUEST\_ACTION – This will return the Service Request Action the merchant sent in the transaction request.
- TRANS\_STATUS\_NAME – Transaction Status.
- TRANS\_VALUE – Total transaction amount for all line items.
- CUST\_ID – Customer ID.
- XTL\_CUST\_ID- Merchant's Customer ID.
- PO\_ID – Purchase order ID.
- XTL\_ORDER\_ID – Merchant's order ID.
- BATCH\_ID – Settlement Batch ID.
- PROC\_NAME – Merchant Processor Name (Example: "Payvision").

- MERCH\_ACCT\_ID - Merchant Bank's Account ID.
- CARD\_BRAND\_NAME – Bank account or credit card type (Example: “Mastercard”).
- PMT\_L4 – Payment account or credit card's last 4 digits.
- PMT\_ID – Payment ID
- PMT\_ID\_XTL – External Payment ID
- PROC\_UDF01 – Processor User Defined Field 1
- PROC\_UDF02 – Processor User Defined Field 2
- PROC\_AUTH\_RESPONSE – Processor Authorization Response Code.
- PROC\_RETRIEVAL\_NUM – Processor Retrieval Number or GUID
- PROC\_REFERENCE\_NUM – Processor Reference Number.
- AVS\_RESPONSE – Address Verification Service Response Code.
- CVV\_RESPONSE – Card Verification Value Response Code.
- REQ\_ID – Request Identifier
- REQUEST\_API\_VERSION – Payment Service API Version (4.9).
- PO\_LI\_ID\_X – Purchase Order Line Item ID.
- PO\_LI\_COUNT\_X – Purchase Order Line Item Count.
- PO\_LI\_AMOUNT\_X – Purchase Order Line Item Total Amount.
- PO\_LI\_PROD\_ID\_X – Purchase Order Line Item Product ID.
- MBSHP\_ID – Membership ID (returned on membership transactions)

### 5.6.5 Transaction Status is PENDING

Some authorization requests may get a transaction status of “Pending”. This usually happens if the processor insists that the customer goes through an extra credit card authentication step such as 3-D Secure (i.e. Verified by Visa or Mastercard SecureCode). Merchants should also expect a redirect URL in `PROC_REDIRECT_URL` field in the gateway response in case of 3-D Secure authentication.

ACH and European Direct Debit authorization requests will also return “Pending” status.

## 5.7 Updating Order Data

“CCTRANSUPDATE” can be used to update specific data elements of an order. Currently, only `TRANS_CUST_RECEIPT` is available to be updated.

To update an order with a receipt, merchants may set the `REQUEST_ACTION` parameter to “CCTRANSUPDATE” along a reference to the order, either in the `REQUEST_REF_PO_ID` (referencing the `PO_ID` from a previous order) or `REQUEST_REF_PO_ID_XTL` (referencing the `XTL_ORDER_ID` from a previous order) parameter.

### 5.7.1 CCTRANSUPDATE Example Request using REQUEST\_REF\_PO\_ID

```
/pmt_service.cfm?req_password=P5ssword1&site_id=1234&request_ref_po_id=1234567&request_response_format=JSON&req_username=test1%40example.net&request_api_version=4.8&request_action=CCTRANSUPDATE&mbshp_id_xtl=1502&trans_customer_receipt=aHRocHM6Ly9udXRyaXN5cy5yZWMudHJhbnNfaWQyMzQ=
```

### 5.7.2 CCTRANSUPDATE Example Request using REQUEST\_REF\_PO\_ID\_XTL

```
https://api.inovio.com/payment/pmt_service.cfm ?req_password=P5ssword1&site_id=1234&request_ref_po_id_xtl=order123&request_response_format=JSON&req_username=test1%40example.net&request_api_version=4.8&request_action=CCTRANSUPDATE&mbshp_id_xtl=1502&trans_customer_receipt=aHRocHM6Ly9udXRyaXN5cy5yZWMudHJhbnNfaWQyMzQ=
```

### 5.7.3 CCTRANSUPDATE Example Response

The response example below shows that the order was captured and then reversed. Merchants may refer to the `TRANS_STATUS_NAME` in checking the status of a specific transaction request.

```
<RESPONSE>
<TRANSACTION TRANS_ID="43500">
<REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>10</TRANS_VALUE>
<CUST_ID>1500</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>1234567</PO_ID>
<XTL_ORDER_ID>order123</XTL_ORDER_ID>
<BATCH_ID>1001</BATCH_ID>
<PROC_NAME>Test Processor</PROC_NAME>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<PMT_L4>3762</PMT_L4>
<PROC_UDF01/>
<PROC_UDF02/>
<PROC_AUTH_RESPONSE>46177</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>56789AB</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>M</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<MBSHP_ID_XTL>1502</MBSHP_ID_XTL>
</TRANSACTION>
```

### 5.7.4 CCTRANSUPDATE Response Fields

- `TRANS_ID` – Unique transaction ID.
- `REQUEST_ACTION` – This will return the Service Request Action the merchant sent in the transaction request.
- `TRANS_STATUS_NAME` – Transaction Status.
- `TRANS_VALUE` – Total transaction amount for all line items.
- `CUST_ID` – Customer ID.
- `XTL_CUST_ID` – Merchant's Customer ID.
- `PO_ID` – Purchase order ID.
- `XTL_ORDER_ID` – Merchant's order ID.
- `BATCH_ID` – Settlement Batch ID.
- `PROC_NAME` – Merchant Processor Name (Example: "Payvision").
- `MERCH_ACCT_ID` – Merchant Bank's Account ID.
- `CARD_BRAND_NAME` – Bank account or credit card type (Example: "Mastercard").
- `PMT_L4` – Payment account or credit card's last 4 digits.
- `PMT_ID` – Payment ID
- `PMT_ID_XTL` – External Payment ID
- `PROC_UDF01` – Processor User Defined Field 1
- `PROC_UDF02` – Processor User Defined Field 2
- `PROC_AUTH_RESPONSE` – Processor Authorization Response Code.
- `PROC_RETRIEVAL_NUM` – Processor Retrieval Number or GUID
- `PROC_REFERENCE_NUM` – Processor Reference Number.
- `AVS_RESPONSE` – Address Verification Service Response Code.

- CVV\_RESPONSE – Card Verification Value Response Code.
- REQ\_ID – Request Identifier
- REQUEST\_API\_VERSION – Payment Service API Version (4.9).
- PO\_LI\_ID\_X – Purchase Order Line Item ID.
- PO\_LI\_COUNT\_X – Purchase Order Line Item Count.
- PO\_LI\_AMOUNT\_X – Purchase Order Line Item Total Amount.
- PO\_LI\_PROD\_ID\_X – Purchase Order Line Item Product ID.
- MBSHP\_ID – Membership ID (returned on membership transactions)
- MBSHP\_ID\_XTL – Customer membership ID received from the Merchant

## 6 ACH/eCheck Transactions

The following steps describe the lifecycle of a transaction processed with an ACH account:

1. Customer initiates purchase on the payment page.
2. Merchant submits transaction request to Payments Service. This should include the following data:
  - a. Customer and payment information.
  - b. Merchant gateway credentials and other gateway parameters as needed.
3. Payments Service transmits transaction request to the bank.
4. Merchant receives gateway response. Transaction status should be “Pending”.
5. Post processing transaction status update (both methods are optional):
  - a. Postback API: Payments Service sends Postback to the merchant as the transaction **state** changes. Postback is real-time.
  - b. Order Detail Report: Merchant downloads order and transaction data using Order Detail Report service.

**!** If you choose not to use methods 5a or 5b to receive updates to the status of an ACH order, you can log into our portal and check the status by searching for the order ID.

### 6.1 Transaction State Updates Through Postback (Optional)

To keep merchants updated on the status of their pending ACH transactions, we can send real-time Postbacks to the merchant.

### 6.2 Transaction State using Order Detail API (Optional)

Merchants may also download transaction status updates as an alternative to Postback by using Order Detail Report service.

#### 6.2.1 Transaction Status

TRANS_STATUS_NAME	Description
APPROVED	Transaction has been approved.
PENDING	Transaction is in pending status.
RUNNING	Transaction processing was not completed or is waiting completion usually because of gateway error.

### 6.3 Gateway Request Parameters

The table below describes parameters needed in sending ACH purchase request to the gateway. Merchants may send additional parameters, as described in the Payments Payment Service API, such as customer billing address, email, website login and other pass-through data.

Field Name	Description
<b>REQUEST_ACTION</b>	Service Request Action (Send “ACHAUTHCAP”)
<b>REQ_USERNAME</b>	Service Request Username
<b>REQ_PASSWORD</b>	Service Request Password
<b>REQUEST_RESPONSE_FORMAT</b>	Service Response Format (XML or JSON)
<b>REQUEST_API_VERSION</b>	API Version (example: “4.6”)
<b>SITE_ID</b>	Merchant’s Website ID
<b>LI_PROD_ID_1</b>	Line Item Product ID 1
<b>LI_VALUE_1</b>	Line Item Transaction Amount 1
<b>PMT_NUMB</b>	Bank Account Number
<b>BANK_IDENTIFIER</b>	Routing Number
<b>MERCH_ACCT_ID</b>	Merchant Account ID
<b>REQUEST_CURRENCY</b>	3-letter Currency Code
<b>CUST_FNAME</b>	Customer First Name
<b>CUST_LNAME</b>	Customer Last Name

<b>BILL_ADDR</b>	Billing Street Address
<b>BILL_ADDR_CITY</b>	Billing City Name
<b>BILL_ADDR_STATE</b>	Billing State 2-letter Code
<b>BILL_ADDR_COUNTRY</b>	Billing Country 2-letter Code
<b>BILL_ADDR_ZIP</b>	Billing ZIP or Postal Code

## 6.4 Gateway Actions

Below are the only supported gateway actions for ACH. Any other gateway action or feature will not work.

REQUEST_ACTION	Description
<b>ACHAUTHCAP</b>	Used for authorization and capture requests.
<b>ACHAUTHORIZE</b>	Used for Authorizations without Capture
<b>ACHREVERSE</b>	Used for Authorization Capture Reversal
<b>ACHCREDIT</b>	Used for transaction credit requests.
<b>ACHPAYOUT</b>	Used for ACH direct deposit payout

## 6.5 Authorization and Capture (ACHAUTHCAP) Request and Response

Authorization and capture occurs when a merchant processes the transaction for the entire amount therefore placing a hold on the funds and then later requests the authorized (captured) funds to be transferred to their account.

### 6.5.1 Authorization (ACHAUTHCAP) Request Example

```
https://api.inoviopay.com/payment/pmt_service.cfm ?
request_action=ACHAUTHCAP&request_api_version=4.8&req_username=merchant@example.com&req_p
assword=Gatewaypwd1&site_id=242424&request_response_format=XML&request_currency=USD&merch
_acct_id=10001&bank_identifier=987654321&pmt_numb=01234567&cust_fname=John&cust_lname=Tester
&li_value_1=10.00&li_prod_id_1=12345&bill_addr=124%20Test%20Address&bill_addr_country=US&bill_ad
dr_city=Los%20Angeles&bill_addr_state=CA&bill_addr_zip=90904&xlt_ip=
```

### 6.5.2 Authorization (ACHAUTHCAP) Response Example (Pending Status)

```
<RESPONSE>
<REQUEST_ACTION>ACHAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME>PENDING</TRANS_STATUS_NAME>
<TRANS_VALUE>10 </TRANS_VALUE>
```

```

<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>10</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>022480</TRANS_ID>
<CUST_ID>1234567</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>323232</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>10999</BATCH_ID>
<PROC_NAME>ACH Processor</PROC_NAME>
<MERCH_ACCT_ID>10001</MERCH_ACCT_ID>
<CARD_BRAND_NAME/>
<CARD_DETAIL/>
<CARD_TYPE/>
<CARD_PREPAID/>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>4567</PMT_L4>
<PMT_ID>505050</PMT_ID>
<PMT_ID_XTL/>
<PROC_UDF01/>
<PROC_UDF02/>
<PROC_AUTH_RESPONSE/>
<PROC_RETRIEVAL_NUM/>
<PROC_REFERENCE_NUM/>
<PROC_REDIRECT_URL/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQ_ID>30814003200</REQ_ID>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>1555538</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>10</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>12345</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>

```

## 6.6 Account Validation via AUTHORIZATION without CAPTURE (ACHAUTHORIZE) Request and Response

This Request/Response is used to validate ACH parameters provided by the customer without capturing any amount. This means that there are no funds captured or transferrable to a Merchant's account.

### 6.6.1 Authorization (ACHAUTHORIZE) Request Example

```

https://api.inoviopay.com/payment/token_service.cfm?
request_action=ACHAUTHORIZE&request_api_version=4.8&req_username=merchant@example.com&re
q_password=Gatewaypwd1&site_id=242424&request_response_format=XML&request_currency=USD&m
erch_acct_id=10001&bank_identifier=987654321&pmt_numb=01234567&cust_fname=John&cust_lname=
Tester&li_value_1=10.00&li_prod_id_1=12345&bill_addr=124%20Test%20Address&bill_addr_country=US
&bill_addr_city=Los%20Angeles&bill_addr_state=CA&bill_addr_zip=90904&xtl_ip=

```

## 6.6.2 Authorization (ACHAUTHORIZE) Response Example (Pending Status)

```
<RESPONSE>
<REQUEST_ACTION>ACHAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME>PENDING</TRANS_STATUS_NAME>
<TRANS_VALUE>10 </TRANS_VALUE>
<CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
<TRANS_VALUE_SETTLED>10</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>022480</TRANS_ID>
<CUST_ID>1234567</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>323232</PO_ID>
<XTL_ORDER_ID/>
<BATCH_ID>10999</BATCH_ID>
<PROC_NAME>ACH Processor</PROC_NAME>
<MERCH_ACCT_ID>10001</MERCH_ACCT_ID>
<CARD_BRAND_NAME/>
<CARD_DETAIL/>
<CARD_TYPE/>
<CARD_PREPAID/>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>4567</PMT_L4>
<PMT_ID>505050</PMT_ID>
<PMT_ID_XTL/>
<PROC_UDFO1/>
<PROC_UDFO2/>
<PROC_AUTH_RESPONSE/>
<PROC_RETRIEVAL_NUM/>
<PROC_REFERENCE_NUM/>
<PROC_REDIRECT_URL/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQ_ID>30814003200</REQ_ID>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>1555538</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>10</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>12345</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>
```

## 6.6.3 Service Declined Response Example

The example above was declined by the API due to an invalid data in the `PMT_NUMB` field.

```
<RESPONSE>
<REQUEST_ACTION>ACHAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME/>
<CUST_ID/>
<XTL_CUST_ID>c777777777</XTL_CUST_ID>
<MERCH_ACCT_ID>100</MERCH_ACCT_ID>
<CARD_BRAND_NAME/>
<PMT_L4/>
```

```

<PMT_ID/>
<PMT_ID_XTL/>
<API_RESPONSE>113</API_RESPONSE>
<API_ADVICE>Invalid Data</API_ADVICE>
<SERVICE_RESPONSE>o</SERVICE_RESPONSE>
<SERVICE_ADVICE>Declined</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>o</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE></PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>o</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD>PMT_NUMB</REF_FIELD>
<PROC_NAME/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

## 6.7 Reversal Request and Response

### 6.7.1 Reversal Request Example

[https://api.inoviopay.com/payment/prmt\\_service.cfm  
?request\\_action=ACHREVERSE&request\\_api\\_version=4.8&req\\_username=merchant@example.com&req\\_p assword=Gatewaypwd1&site\\_id=242424&request\\_response\\_format=XML&request\\_ref\\_po\\_id=323232&li\\_v alue\\_1=10](https://api.inoviopay.com/payment/prmt_service.cfm?request_action=ACHREVERSE&request_api_version=4.8&req_username=merchant@example.com&req_p assword=Gatewaypwd1&site_id=242424&request_response_format=XML&request_ref_po_id=323232&li_v alue_1=10)

### 6.7.2 Reversal Response Example

```

<RESPONSE>
  <REQUEST_ACTION>ACHREVERSE</REQUEST_ACTION>
  <TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
  <TRANS_VALUE>10</TRANS_VALUE>
  <CURR_CODE_ALPHA>USD</CURR_CODE_ALPHA>
  <TRANS_VALUE_SETTLED>-10</TRANS_VALUE_SETTLED>
  <CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
  <TRANS_EXCH_RATE/>
  <TRANS_ID>1635078</TRANS_ID>
  <CUST_ID>1234567</CUST_ID>
  <XTL_CUST_ID/>
  <PO_ID>547848</PO_ID>
  <XTL_ORDER_ID/>
  <BATCH_ID>10999</BATCH_ID>
  <PROC_NAME>ACHProcessor</PROC_NAME>
  <MERCH_ACCT_ID>10001</MERCH_ACCT_ID>
  <CARD_BRAND_NAME/>
  <CARD_DETAIL/>
  <CARD_TYPE/>
  <CARD_PREPAID/>
  <CARD_BANK/>
  <PMT_L4>4567</PMT_L4>
  <PMT_ID>505050</PMT_ID>
  <PMT_ID_XTL/>
  <PROC_UDF01/>
  <PROC_UDF02/>
  <PROC_AUTH_RESPONSE/>
  <PROC_RETRIEVAL_NUM/>

```

```

<PROC_REFERENCE_NUM/>
<PROC_REDIRECT_URL/>
<AVS_RESPONSE/>
<CVV_RESPONSE/>
<REQ_ID>30224993200</REQ_ID>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>1555539</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>-10</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>12345</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>

```

## 6.8 Credit Request and Response

### 6.8.1 Credit Request

[https://api.inoviopay.com/payment/pmt\\_service.cfm  
?REQUEST\\_REF\\_PO\\_ID=591570&req\\_username=example@this.net&req\\_password=Password111&site\\_id=12345&request\\_response\\_format=JSON&request\\_action=ACHCREDIT&request\\_api\\_version=20&merch\\_acct\\_id=30603&CREDIT\\_ON\\_FAIL=0&client\\_id=15904](https://api.inoviopay.com/payment/pmt_service.cfm?REQUEST_REF_PO_ID=591570&req_username=example@this.net&req_password=Password111&site_id=12345&request_response_format=JSON&request_action=ACHCREDIT&request_api_version=20&merch_acct_id=30603&CREDIT_ON_FAIL=0&client_id=15904)

### 6.8.2 Credit Response

Like ACHAUTHCAP, most processors will return a status of PENDING, and the status will later be updated to APPROVED at a later date, sometimes up to ten days.

## 6.9 Status Change

After the ACH order has been approved, our system will update the status of the order from PENDING to APPROVED. Notification of this change can be acquired by pulling transaction data through our Order Detail Report, or by receiving postback data.

## 7 Boleto

Boleto (boleto bancário) is similar to the previous two transaction types in that there is still an additional step after the authorization. The difference is that the URL in the response yields the boleto, which the customer must then print and submit with payment at a qualifying location.



**Brazilian addresses are composed of the following fields and must be entered in this format.**

- **The name of the logradouro (which is equivalent to an avenue or street), the number of the place in the street and the number of the floor and/or apartment (when necessary). These fields must be explicitly entered in this order on the bill\_addr field.**
- **The neighbourhood**
- **The city**
- **The state (which must be informed by an abbreviation)**
- **The postal code (numbers only).**

**Example:**

```

bill_addr: "Av. Paulista, 1098, 1º andar apto. 101"
Bill_addr_district: "Bela Vista"

```

```
bill_addr_city "São Paulo"
bill_addr_state "SP"
bill_addr_country: ""
bill_addr_zip : "01310000"
```

## 7.1 Request Parameters

Field Name	Description	Mandatory
<b>request_action</b>	<b>BOLETOAUTHCAP</b>	Yes
<b>li_prod_id_x</b>	Line Item Count - Max value is "10"	Yes
<b>li_value_x</b>	Line Item Transaction Amount	Yes
<b>cust_brcpfcnpj</b>	Customer's CPF number	Yes
<b>merch_acct_id</b>	Merchant Account ID	Yes
<b>cust_fname</b>	Customer's First Name	Yes
<b>cust_lname</b>	Customer's Last Name	Yes
<b>cust_birthday</b>	Customer's date of birth	No
<b>bill_addr</b>	Customer's Billing Street Address <i>&lt;street name&gt;, &lt;street number&gt;, [complement – Optional]&gt;</i>	Yes
<b>bill_addr_city</b>	Customer's Billing City	Yes
<b>Bill_addr_district</b>	Customer's Neighborhood	Yes
<b>bill_addr_state</b>	Customer's Billing State	Yes

<b>bill_addr_zip</b>	Customer's Billing Postal/ZIP code	Yes
<b>bill_addr_country</b>	Customer's Billing Country	Yes
<b>cust_email</b>	Customer's Email Address	No
<b>request_currency</b>	3-letter Currency Code. Ex: BRL	No

## 7.2 Additional Response Parameters

Field Name	Description
<b>proc_success_url</b>	The address to which a Customer is redirected once the authorization is complete
<b>proc_error_url</b>	The address to which a Customer is redirected when an authorization fails to be completed
<b>Proc_redirect_url</b>	The URL to display the Boleto for printing.
<b>Proc_barcode</b>	Contains formatted Boleto barcode data in the gateway API response

### 7.2.1 Boleto Request Example

```
https://api.inoviopay.com/payment/pmt_service.cfm
?cust_fname=Testy6730960&cust_lname=Tester6730960&li_count_1=1&li_prod_id_1=32805&li_
value_1=.53&cust_login=Test6730960&cust_password=Tests6730960&xtl_cust_id=Test6730960
&request_action=BOLETOAUTHCAP&merch_acct_id=128812&cust_email=Here@isatest.org&req
uest_currency=BRL&bill_addr_country=BR&cust_brcpfcnpj=629.698.570.30&bill_addr_zip=6990
0000&bill_addr=6566+Any+street&cust_phone=6566566666&bill_addr_city=Some+City&bill_add
r_state=AC&req_username=custusername&req_password=CustPassword123&site_id=11191&requ
est_response_format=JSON&request_api_version=4.8
```

### 7.2.2 Boleto Response Example

Below is a sample of a response to a BOLETOAUTHCAP request. The `PROC_REDIRECT_URL` is the URL to display the Boleto for printing.

```
{
: "REQUEST_ACTION":"BOLETOAUTHCAP",
: "TRANS_STATUS_NAME":"PENDING",
: "TRANS_VALUE":1.2,
: "CURR_CODE_ALPHA":"BRL",
: "TRANS_VALUE_SETTLED":1.2,
: "CURR_CODE_ALPHA_SETTLED":"BRL",
: "TRANS_EXCH_RATE":1.0,
: "TRANS_ID":124973612,
: "CUST_ID":45752655,
: "XTL_CUST_ID":"",
: "PO_ID":111351972,
```

```

    : "XTL_ORDER_ID":"",
    : "BATCH_ID":1652714,
    : "PROC_NAME":"Bradesco",
    : "MERCH_ACCT_ID":52618,
    : "CARD_BRAND_NAME":"BOL",
    : "CARD_TYPE":"",
    : "CARD_PREPAID":0,
    : "CARD_BANK":"",
    : "CARD_BALANCE":"",
    : "PMT_L4":"3456",
    : "PMT_ID":49166813,
    : "PMT_ID_XTL":"",
    : "PROC_UDF01":"CA",
    : "PROC_UDF02":"",
    : "PROC_AUTH_RESPONSE":"124973612",
    : "PROC_RETRIEVAL_NUM":"2379276625600113519572003852208675100000000120",
    : "PROC_BARCODE":"2379276625600113519572003852208675100000000120",
    : "PROC_REFERENCE_NUM":"35cc6ac8-5ef7-4a49-b8dd-72609e2aa627",
    : "PROC_REDIRECT_URL": "https://transaction.pagador.com.br/post/pagador/reenvia.asp/35cc6ac8-5ef7-4a49-b8dd-72609e2aa627",
    : "AVS_RESPONSE":"",
    : "CVV_RESPONSE":"",
    : "REQ_ID":"30814113221",
    : "REQUEST_API_VERSION":"4.6",
    : "PO_LI_ID_1":"60915685",
    : "PO_LI_COUNT_1":"1",
    : "PO_LI_AMOUNT_1":"1.2",
    : "PO_LI_PROD_ID_1":"20923",
    : "MBSHP_ID_1": ""
}
}

```

## 8 Pix Payment

Pix is a payment method for instant direct bank transfers in Brazil. Pix is owned and regulated by Central Bank (Banco Central) and operated by Brazilian banks, digital accounts, and wallets.

For online shoppers using Pix, the funds are sent directly from the shopper bank account and then transferred to the merchant's bank account instantly.

### 8.1 PIX Request Parameters

Field Name	Description	Mandatory
request_action	PIXSALE	Yes
li_prod_id_x	Line Item Count - Max value is "10"	Yes
li_value_x	Line Item Transaction Amount	Yes
cust_brcpfcnpj	Customer's CPF number	Yes

<b>merch_acct_id</b>	Merchant Account ID	Yes
<b>cust_fname</b>	Customer's First Name	Yes
<b>cust_lname</b>	Customer's Last Name	Yes
<b>cust_birthday</b>	Customer date of birth	No
<b>bill_addr</b>	Customer Billing Street Address	No
<b>bill_addr_city</b>	Customer's Billing City	No
<b>bill_addr_district</b>	Customer's Neighborhood	No
<b>bill_addr_state</b>	Customer's Billing State	No
<b>bill_addr_zip</b>	Customer's Billing Postal/ZIP code	No
<b>bill_addr_country</b>	Customer's Billing Country	No
<b>cust_email</b>	Customer's Email Address	No
<b>request_currency</b>	3-letter Currency Code. Ex: BRL	No

## 8.2 Additional Response Parameters

Field Name	Description
<b>Proc_redirect_url</b>	The address to which a Customer is redirected once the QR Code has been generated
<b>proc_retrieval_num</b>	Processor tracking number ( <i>pedido.numero</i> )
<b>Proc_reference_num</b>	Contains Pix transaction verification number ( <i>pix.token</i> )

## 8.3 Pix Request Example

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=PIXSALE&li_count_1=1&li_prod_id_1=32221&li_value_1=1.00&bill_addr_zip=014
18102&xtl_udf01=Test1&BILL_ADDR_CITY=Sao+Paulo&BILL_ADDR=277+Domingos+Sergio+dos+A
njos&BILL_ADDR_STATE=SP&XTL_IP=000.000.001&BILL_ADDR_DISTRICT=Pirituba&CUST_B
RCPFCNPJ=483.649.730-29
&req_username=pixuser%40Testpay.com&req_password=Password12345555&site_id=22619&req
uest_response_format=JSON&request_api_version=4.8&request_currency=BRL&merch_acct_id
=31219&cust_email=test%40test.com&cust_fname=Pessoa&cust_lname=de+Testing
```

## 8.4 Pix Response Example

Below is a sample response for a PIXSALE request. The `PROC_REDIRECT_URL` is the URL to display the QR Code for scanning and further payment.

```
{
  "REQUEST_ACTION": "PIXSALE",
  "REQ_ID": "6960183",
  "TRANS_STATUS_NAME": "PENDING",
  "TRANS_VALUE": "1",
  "TRANS_VALUE_SETTLED": "1",
  "CURR_CODE_ALPHA_SETTLED": "BRL",
  "TRANS_EXCH_RATE": "",
  "TRANS_ID": "362000021",
  "CUST_ID": "3546515",
  "XTL_CUST_ID": "",
  "CUST_BRCPFCNPJ": "483.649.730-29",
  "PO_ID": "35739669",
  "XTL_ORDER_ID": "",
  "BATCH_ID": "30323",
  "PROC_NAME": "BRADESCO PIX",
  "MERCH_ACCT_ID": "11119",
  "CARD_BRAND_NAME": "PIX",
  "PMT_L4": "2488",
  "PMT_ID": "331179",
  "PMT_ID_XTL": "",
  "PMT_AAU_UPDATE_DT": "",
  "PMT_AAU_UPDATE_DESC": "",
  "PROC_UDF01": "",
  "PROC_UDF02": "",
  "PROC_AUTH_RESPONSE": "",
  "PROC_RETRIEVAL_NUM": "3212739",
  "PROC_REFERENCE_NUM": "abcdhsU2xuUFZnNmRXa25CUIMoQWJ4OEhMc2htb1o2dDNsVStDcacgsjdNOHIWTEtZZUYzxbstsZkR2R3FkUUFoNA==",
  "PROC_BARCODE": "",
  "PROC_REDIRECT_URL": "https://meiosdepagamento.bradesco.com.br/apipix/formulario?token=abcdesdW5SaVVFcFVGeFFGMnBFSGFvakdzWGFiaktmMkI3T1JkkpvSDN4NmZWUEdTKit3ovYUErMERmMmVLMStIVVdPRFdaWUdxRGxyz==",
  "AVS_RESPONSE": "",
  "CVV_RESPONSE": "",
  "REQ_ID": "30819993221",
  "REQUEST_API_VERSION": "4.6",
  "P3DS_VENDOR": "",
  "P3DS_RESPONSE": "",
  "PO_LI_ID_1": "37943",
  "PO_LI_COUNT_1": "1",
  "PO_LI_AMOUNT_1": "1",
  "PO_LI_PROD_ID_1": "32221",
  "MBSHP_ID_1": ""
}
```

## 9 Credilink

Credilink validates the Brazilian identification numbers, CPF (individuals) or CNPJ (businesses) for use on different business rules. Credilink Service is only available in the Brazilian market. For more information on technical usage, please contact your gateway support representative.

## 9.1 How it Works

Merchant sends transaction data, including the CPF. This can be sent with the other optional parameters like phone number, date of birth and customer name. Credilink will check the CPF and attempt to confirm the validity of the CPF/CNPJ against requested services as supplied to the gateway. If there is no match for the CPF/CNPJ, the user is a minor where restricted or if there is a death record associated with the CPF, then a decline response will be returned and other adverse actions may be enforced for the declined CPF.

- Credilink Identity check will be applicable on all transaction types (Credit Card, Pix and Boleto)
- The service will validate if the CPF belongs to a minor or not. (No date of birth will be returned)
  - If the CPF is for a minor (person of less than 18 YO as per the date of check), then a Response indicating the Customer as a minor will be returned
  - If the CPF is not for a minor, then additional checks will be performed
- The Service will also validate if the CPF is associated with a death record.
  - If the CPF is associated with a death record, a response indicating a death record will be returned
  - If there is a death record associated with the CPF, then the transaction will be declined and CPF is blacklisted.
- The Service also checks for CPF validity, if CPF is reported as invalid, then a response is returned indicating invalid cpf
- The service will check if the CPF is valid but canceled, if the CPF is canceled, then a response is returned indicating invalid cpf.

## 9.2 How to Use Credilink

- Submit a request to Inovio to configure and activate Credilink on a merchant account that'll use the credilink services
- Pass in the CUST\_BRCPFCNPJ parameter in every transaction
- Receive response:

Service_Response	Description
700	Scrub Decline
706	Failed Age Validation
707	Invalid CPF

## 9.3 Additional Parameters

Field Name	Condition	Description
CUST_BRCPFCNPJ	Required	Customer's cpf/cnpj
CUST_BIRTHDAY	Optional	Customer's Date of Birth
CUST_PHONE	Optional	Customer's Phone number
CUST_FNAME	Optional	Customer's First Name

<b>CUST_LNAME</b>	Optional	Customer's Last Name
<b>Transaction Fields</b>		
<p>In addition to CUST_BRCPFCNPJ the phone number or birthday can be sent.</p> <p>A match check is run against these values.</p>		CUST_BIRTHDAY CUST_BRCPFCNPJ <b>CUST_PHONE</b> SITE_ID REQ_USERNAME REQ_PASSWORD REQUEST_ACTION REQUEST_RESPONSE_FORMAT REQUEST_API_VERSION LI_PROD_ID_1 LI_VALUE_1 LI_COUNT_1 <b>PMT_ID</b> MERCH_ACCT_ID REQUEST_CURRENCY

## 10 Order Insight

Order Insight is a platform that allows merchants to share transaction information with issuers in real time to resolve disputes and prevent chargebacks. It's part of Visa's initiative to modernize the dispute management process and is integrated into Visa's issuer dispute management platform, Visa Resolve Online (VROL). Verifi is a payment protection and management company that helps merchants, issuers, and acquirers reduce financial loss and fraud.

- Order Insight allows Merchants to receive real-time alerts when a transaction dispute is initiated by a customer. The Order Insight service will share and receive detailed transaction data on the customer, the product or service purchased, and the payment method used.
- Compelling Evidence (CE) is an additional tool of Order Insight that helps merchants use a cardholder's purchase history to prove that a disputed transaction is legitimate. This takes place over 2-5 separate API requests to confirm the consistency of the recurring purchase data.

### 10.1 Additional Request Parameters

The below request parameters are needed to fully unlock the benefits of Verifi's Order Insight product. All transaction linking through API calls is completed on behalf of the merchant's account by Inovio. These are optional fields within the Inovio Gateway API and will not stop a transaction even if Order Insight is enabled on the Merchant Account. When an Order Insight request comes in, Inovio will respond with all transactional information given by the merchant at the time of the original transaction.

Order Insight requires a history of 120 days of the product parameters described below. CE requires a history of 365 days of the customer and device parameters described below.

For more information on technical usage, please contact your gateway support representative to obtain a copy of the form to be filled out to expedite the historical update of transactional information and use of Order Insight products.

Field Name	Description	Data Type	Char Limit
XTL_CUST_ID	Merchant's Customer ID	Alphanumeric and special characters	max of 24
PROD_DESC_XTL	Detailed description of the product (merchandise or service) purchased	Alphanumeric and special characters	up to 1,000
DEVICE_ID_XTL	Device ID of the device used to submit the order. Examples are IMEI or MEID values from the device. Data cannot be hashed.	Alphanumeric and special characters	15-32
DEVICE_FINGERPRINT_XTL	Device fingerprint information is generated by a third-party service provider or the Merchant's own algorithm to combine browser or device attributes to form a unique fingerprint to identify the device. Data can be hashed.	Alphanumeric and special characters	20-45

## 10.2 Reporting

### 10.2.1 Order Insight Lookup Statuses

As an Order Insight lookup ages, postbacks will be sent to show the progression throughout the transaction's lifetime. A lookup can have many statuses depending on the outcome of the API calls between Inovio and Verifi over the transaction's chargeback window.

Status	Description
Responded	Inovio has received the API call from Verifi and responded with the data requested within SLA. This is followed by a "ORDER_INSIGHT_LOOKUP_EVENT" postback.
Failed (visible in Portal and Order Detail API)	Inovio never received the API request from Verifi. Lookup event was discovered during the SFTP catch up process. We will fight to prevent this charge from Verifi.
Deflection	The lookup request did not lead to a chargeback. This is followed by a "ORDER_INSIGHT_SETTLED_EVENT" postback.
Negation	The lookup request led to a chargeback. This is followed by a "ORDER_INSIGHT_NEGATION_EVENT" postback.
Reversal	The lookup request that was previously led to a deflected dispute was reversed. This is followed by a "ORDER_INSIGHT_REVERSAL_EVENT" postback.

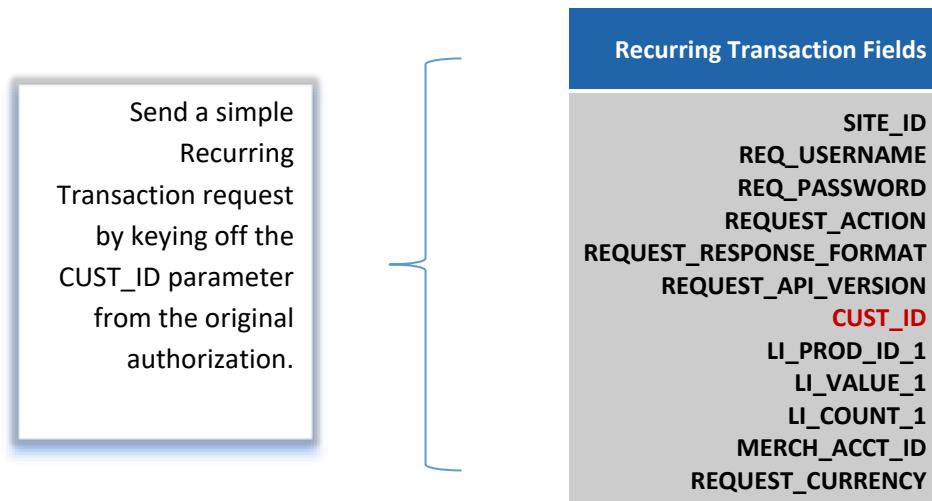
# 11 Transaction Modes

## 11.1 Recurring

MERCHANTS may sometimes need to charge their customers periodically using a payment account that a customer has previously used. In this case, the succeeding transaction after the original authorization is called a *recurring* transaction. In recurring transaction mode, the merchant is not required to re-send the full payment information. When a client intends to use this functionality, the first customer initiated transaction should be flagged as `request_rebill=2`. This makes the processor aware that this is the initial transaction of an upcoming recurring card on file transaction or set of transactions. See [Appendix K](#) for more details.

### 11.1.1 Minimum required fields for sending a simple recurring transaction request using CUST\_ID:

A recurring transaction request can be made by sending the cardholder's Customer ID or `cust_id` in the transaction request. The Payment Service will charge the most recent payment account or credit card that was used. The customer's most recently used address record will be used in the transaction request. Non-USD rebill requests should include the `REQUEST_CURRENCY` with the non-USD currency 3-letter code.



### 11.1.2 Required fields for specifying which payment account to charge with the recurring transaction request using PMT\_L4 parameter:

Some customers may have more than one payment account in the system. In order to specify which payment account to charge, merchants may send the last 4 digits of the payment account or credit card number in the `pmt_l4` parameter.

Sending the last 4 digits of the payment account will let the Payment Service know which account to charge for the recurring payment.

\*As an option, merchants may also send a new Credit Card Expiration Date in the request.

#### Recurring Transaction Fields

SITE_ID
REQ_USERNAME
REQ_PASSWORD
REQUEST_ACTION
REQUEST_RESPONSE_FORMAT
REQUEST_API_VERSION
CUST_ID
LI_PROD_ID_1
LI_VALUE_1
LI_COUNT_1
<b>PMT_L4</b>
*PMT_EXPIRY
MERCH_ACCT_ID
REQUEST_CURRENCY

#### 11.1.3 Required fields for specifying which payment account to charge with the recurring transaction request using PMT\_ID parameter:

Alternatively, the merchant may use Payment ID (PMT\_ID) instead of PMT\_L4, in specifying which payment record to use in the transaction request. The Payment Service will return an error if it cannot find the payment record.

Sending the Payment ID of the payment account will let the Payment Service know which account to charge for the recurring payment.

#### Recurring Transaction Fields

SITE_ID
REQ_USERNAME
REQ_PASSWORD
REQUEST_ACTION
REQUEST_RESPONSE_FORMAT
REQUEST_API_VERSION
CUST_ID
LI_PROD_ID_1
LI_VALUE_1
LI_COUNT_1
<b>PMT_ID</b>
MERCH_ACCT_ID
REQUEST_CURRENCY

#### 11.1.4 Required fields for specifying which payment account to charge with the recurring transaction request using PMT\_ID\_XTL parameter:

Merchants may also use their External Payment ID (PMT\_ID\_XTL) instead of PMT\_ID or PMT\_L4.

Sending the External Payment ID works the same way as sending the PMT\_L4 or PMT\_ID. The service will return an error if it cannot find the customer's payment record.

#### Recurring Transaction Fields

SITE\_ID  
REQ\_USERNAME  
REQ\_PASSWORD  
REQUEST\_ACTION  
REQUEST\_RESPONSE\_FORMAT  
REQUEST\_API\_VERSION  
CUST\_ID  
LI\_PROD\_ID\_1  
LI\_VALUE\_1  
LI\_COUNT\_1  
**PMT\_ID\_XTL**  
MERCH\_ACCT\_ID  
REQUEST\_CURRENCY

#### 11.1.5 Required parameters for sending new billing information with the recurring transaction request:

New billing information including the address fields may be sent with the Recurring Transaction request.

Card expiration date and CVV2/CVC2 are required when sending the full credit card number in the transaction.

The **\*Address Fields** are optional. The Payment Service will use the most recent address information on file if new address information were not included in the request.

#### Recurring Transaction Fields

SITE\_ID  
REQ\_USERNAME  
REQ\_PASSWORD  
REQUEST\_ACTION  
REQUEST\_RESPONSE\_FORMAT  
REQUEST\_API\_VERSION  
**CUST\_ID**  
LI\_PROD\_ID\_1  
LI\_VALUE\_1  
LI\_COUNT\_1  
**PMT\_NUMB**  
**PMT\_EXPIRY**  
**PMT\_KEY**  
PMT\_ID\_XTL  
**\*BILL\_ADDR**  
**\*BILL\_ADDR\_CITY**  
**\*BILL\_ADDR\_STATE**  
**\*BILL\_ADDR\_COUNTRY**  
**\*BILL\_ADDR\_ZIP**  
MERCH\_ACCT\_ID  
REQUEST\_CURRENCY

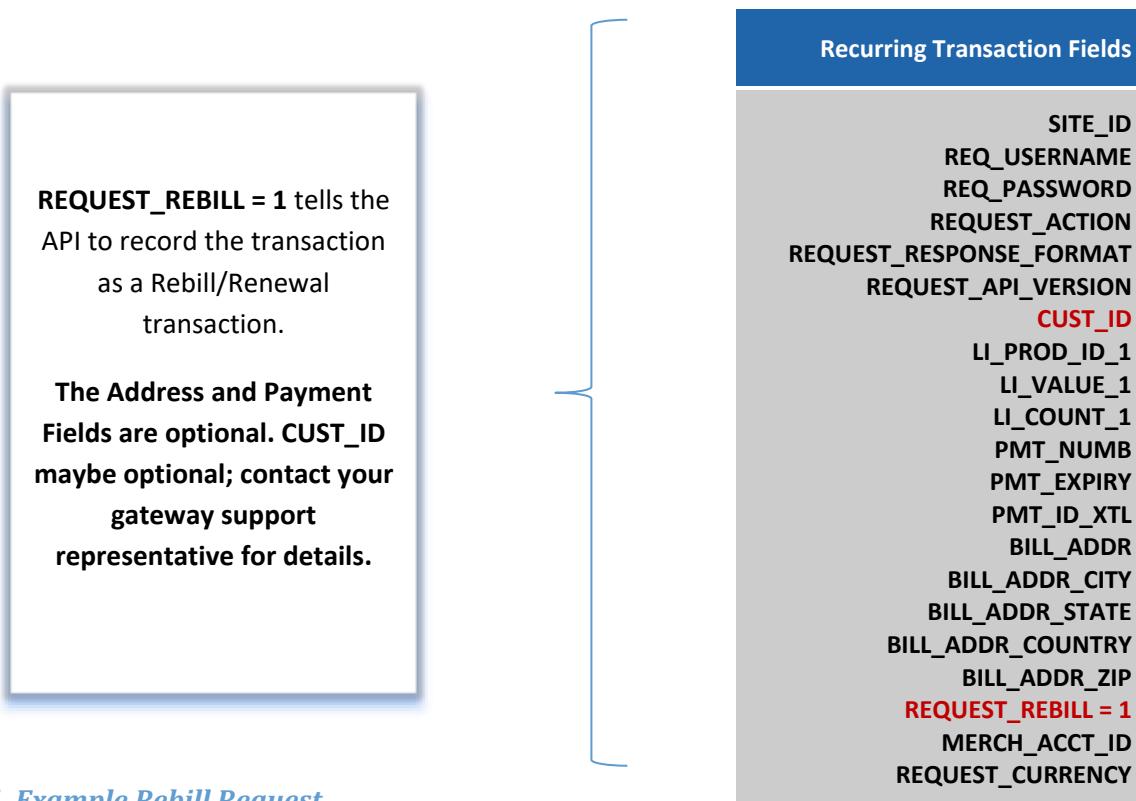
### 11.1.6 Parameters for sending Renewal Transactions (Rebills):

NOTE: This is for merchant-managed Membership products and should not be confused with Renewals managed by the gateway.

To specify that a request is a Renewal or Rebill, set the REQUEST\_REBILL parameter to “1”.

**MERCHANTS ARE ENCOURAGED TO ALWAYS SET THE REQUEST\_REBILL PARAMETER FOR RENEWAL TRANSACTIONS AS IT IS IMPORTANT IN GENERATING CUSTOMER AND SUBSCRIPTION REPORTS.**

Non-USD rebill request should include the REQUEST\_CURRENCY with the non-USD currency 3-letter code.



#### 11.1.6.1 Example Rebill Request

MERCHANTS CAN SEND THE CUST\_ID TO CHARGE AN EXISTING CUSTOMER IN THE SYSTEM, OR SEND THE COMPLETE PAYMENT AND CUSTOMER INFORMATION IN THE REQUEST, IF CUST\_ID IS NOT AVAILABLE.

```
Pmt_service.cfm?req_password=TestPassword1&site_id=0&request_response_format=JSON&req_username=test%40example.com&li_value_1=25.75&li_prod_id_1=1001&request_currency=EUR&xtr_order_id=test&cust_id=1501&pmt_14=0026&request_action=CCAUTHCAP&merch_acct_id=100&request_rebill=1
```

## 11.2 Memberships

The payment system supports website membership management, commonly known as “subscriptions”. Product IDs (li\_prod\_id\_1) should be created before implementing this type of transaction, which can be created using the gateway’s Product Service. For more information on technical usage, please contact your gateway support representative.

Some processors may require customer billing address data.

### 11.2.1 Suggested fields for sending a membership transaction (new customer):

Field Name	Description	Data Type
<b>REQUEST_ACTION</b>	Service Request Action (Send “CCAUTHCAP”).	<a href="#">Service Request Types</a>
<b>REQ_USERNAME</b>	Service Request Username	Alphanumeric
<b>REQ_PASSWORD</b>	Service Request Password	Alphanumeric and Special Characters
<b>REQUEST_RESPONSE_FORMAT</b>	Service Response Format	Accepted values: “XML”, “PIPES” and “JSON”
<b>REQUEST_API_VERSION</b>	Payment Service API Version	Numeric
<b>SITE_ID</b>	Merchant’s Website ID	Numeric
<b>CUST_FNAME</b>	Customer First Name	Alphanumeric and Special Characters
<b>CUST_LNAME</b>	Customer Last Name	Alphanumeric and Special Characters
<b>CUST_EMAIL</b>	Customer Email Address	Alphanumeric
<b>LI_COUNT_1</b>	Line Item Count Max value is “99”.	Numeric
<b>LI_PROD_ID_1</b>	Line Item Product ID 1 (product type should be set to membership)	Numeric
<b>LI_VALUE_1</b>	Line Item Transaction Amount 1	Numeric
<b>BILL_ADDR</b>	Customer Billing Street Address	Alphanumeric
<b>BILL_ADDR_CITY</b>	Customer Billing City	Alphanumeric

<b>BILL_ADDR_STATE</b>	Customer Billing State	2-letter State or Territory Code
<b>BILL_ADDR_ZIP</b>	Customer Billing Postal/ZIP code	Alphanumeric
<b>BILL_ADDR_COUNTRY</b>	Customer Billing Country	2-letter Country Code ISO 3166-1 alpha-2
<b>PMT_NUMB</b>	Credit Card Number	Numeric
<b>PMT_KEY</b>	Credit Card CVV2 or CVC2 Code	Numeric (4)
<b>PMT_EXPIRY</b>	Credit Card Expiration Date	Numeric MMYYYY Example: "122014"
<b>CUST_LOGIN</b>	Customer Login or User Name	Alphanumeric and Special Characters
<b>CUST_PASSWORD</b>	Cardholder's Password Password must be at least 10 characters with 1 number, lower case and upper case letter.  NOTE: Contact Client Services to request change of password requirements.	Alphanumeric
<b>MERCH_ACCT_ID</b>	Merchant Account ID	Numeric
<b>REQUEST_CURRENCY</b>	3-letter Currency Code	Example: "USD"
<b>REQUEST_AFF_ID</b>	External Affiliate ID	Alphanumeric
<b>PMT_ID_XTL</b>	External Payment/Credit Card ID	Alphanumeric

### 11.2.2 Parameters to create Subscription directly from the Gateway API

For more information on technical usage, please contact your gateway support representative about creating subscriptions directly from the Gateway API.

Field Name	Description
	Inovio Payment Service API v4.9    68

<b>PROD_NAME</b>	Product name
<b>PROD_TYPE</b>	1 - Membership Cancels 2 - Membership Renews
<b>PROD_REBILL_METRIC</b>	M Month D Day Y Year
<b>PROD_REBILL_PERIOD</b>	7, 30, 90 etc

#### 11.2.2.1 Example Request creating Subscription directly from Gateway API

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=CCAUTHCAP&pmt numb=4000000000000002&pmt expiry=122025&li_value_1=1.00&bill_addr_state=CA&bill_addr_zip=63146&pmt key=123&xtl_udf01=Test01&prod_name=The+Name&prod_type=2&prod_rebill_metric=D&prod_rebill_period=1&req_username=api@fauxcom.net&req_password=Password9993&site_id=23803&request_response_format=JSON&request_api_version=20&request_currency=USD&argus_debug=0
```

#### 11.2.2.2 Example Response creating Subscription directly from Gateway API

```
{
  "CUST_ID": 110750,
  "REQUEST_API_VERSION": "4.6",
  "CURR_CODE_ALPHA_SETTLED": "USD",
  "CARD_PREPAID": 0,
  "XTL_CUST_ID": "",
  "CARD_TYPE": "",
  "CARD_DETAIL": "",
  "P3DS_RESPONSE": "",
  "PMT_ID": 525551,
  "BATCH_ID": 124104,
  "TRANS_VALUE_SETTLED": 1,
  "TRANS_ID": 1797217,
  "CURR_CODE_ALPHA": "USD",
  "PMT_AAU_UPDATE_DT": "",
  "PROC_REFERENCE_NUM": "TEST828185469",
  "MERCH_ACCT_ID": 123403,
  "REQUEST_ACTION": "CCAUTHCAP",
  "PO_ID": 988074,
  "PROC_NAME": "Test Processor",
  "PMT_L4": "0002",
  "MBSHP_ID_1": "73102",
  "TRANS_STATUS_NAME": "APPROVED",
  "XTL_ORDER_ID": "",
  "TRANS_EXCH_RATE": "",
  "P3DS_VENDOR": "",
  "AVS_RESPONSE": "M",
  "REQ_ID": "15104062",
  "PMT_ID_XTL": "",
  "PO_LI_ID_1": "1829990",
  "PMT_AAU_UPDATE_DESC": ""}
```

```

"PROC_UDF01": "",  

"PROC_RETRIEVAL_NUM": "29259606-88FD-4545-808FBD6936846650",  

"CARD_BALANCE": "",  

"CARD_BRAND_NAME": "Visa",  

"CARD_CLASS": "Consusmer Credit",  

"CARD_DETAIL": "Credit",  

"CARD_COUNTRY": "USA",  

"TRANS_VALUE": 1,  

"PROC_UDF02": "",  

"PO_LI_COUNT_1": 1,  

"PROC_REDIRECT_URL": "",  

"CARD_BANK": "",  

"PROC_AUTH_RESPONSE": "TEST31436",  

"CVV_RESPONSE": "M",  

"PO_LI_AMOUNT_1": "1.00",  

"PO_LI_PROD_ID_1": 127617  

}

```

### 11.2.3 Unique Customer Login Check

For new customers, Payment Service will decline requests if the customer login (CUST\_LOGIN) information sent in the request already exists within the same website (SITE\_ID). The gateway will return Service Response “695” on failed Unique Customer Login Check.

#### 11.2.3.1 Example Declined Response: Username Unavailable

```

<RESPONSE>
  <REQUEST_ACTION>CCAUTHCAP</REQUEST_ACTION>
  <TRANS_STATUS_NAME/>
  <TRANS_VALUE/>
  <TRANS_ID/>
  <CUST_ID/>
  <XTL_CUST_ID/>
  <MERCH_ACCT_ID>100</MERCH_ACCT_ID>
  <CARD_BRAND_NAME/>
  <PMT_L4/>
  <API_RESPONSE>0</API_RESPONSE>
  <API_ADVICE></API_ADVICE>
  <SERVICE_RESPONSE>695</SERVICE_RESPONSE>
  <SERVICE_ADVICE>Site Username Unavailable</SERVICE_ADVICE>
  <PROCESSOR_RESPONSE>0</PROCESSOR_RESPONSE>
  <PROCESSOR_ADVICE></PROCESSOR_ADVICE>
  <INDUSTRY_RESPONSE>0</INDUSTRY_RESPONSE>
  <INDUSTRY_ADVICE></INDUSTRY_ADVICE>
  <REF_FIELD/>
  <PROC_NAME/>
  <AVS_RESPONSE/>
  <CVV_RESPONSE/>
  <REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>

```

### 11.2.4 Membership Cancelation Requests (SUB\_CANCEL)

MERCHANTS may send cancelation requests using the Payment Service by sending `request_action=SUB_CANCEL` in the gateway request. Below are the minimum required parameters for cancelation requests:

#### 11.2.4.1 SUB\_CANCEL Request Parameters

Field Name	Description	Data Type
REQ_USERNAME	Service Request Username	Alphanumeric
REQ_PASSWORD	Service Request Password	Alphanumeric and Special Characters
REQUEST_RESPONSE_FORMAT	Service Response Format	Accepted values: "XML" and "JSON"
REQUEST_API_VERSION	Payment Service API Version (API Version needs to be 2 or higher.)	Numeric
SITE_ID	Merchant's Website ID	Numeric
REQUEST_ACTION	Service Request Action: send "SUB_CANCEL"	
REQUEST_REF_MBSHP_ID	Referring Membership ID.  Note: MBSHP_ID is returned on successful membership authorization requests.	Numeric
SUB_CANCEL_TYPE	<p>Types:</p> <p><b>"1" = CANCEL NOW</b>            "1" will set the membership cancelation date to today's date.</p> <p><b>"2" = CANCEL ON NEXT UPCOMING REBILL DATE</b>            Send "2" to request membership cancelation on the rebill/renewal date.</p>	

#### 11.2.4.2 SUB\_CANCEL Gateway Request Example

```
request_ref_mbshp_id=9149&req_password=Testpass1234&site_id=1234&sub_cancel_type=1&request_api_version=4.8&req_username=testmerchant@example.com&cust_id=291087&request_action=SUB_CANCEL
```

#### 11.2.4.3 SUB\_CANCEL Response Parameters

Field Name	Description

<b>REQUEST_ACTION</b>	Request Action Type: "SUB_CANCEL"
<b>API_RESPONSE</b>	API Response Code
<b>API_ADVICE</b>	API Response Advice
<b>SERVICE_RESPONSE</b>	Service Response Code
<b>SERVICE_ADVICE</b>	Service Response Advice
<b>REF_FIELD</b>	Reference Field (this will contain the field in question in case of data validation errors)
<b>CUST_ID</b>	Customer ID
<b>MBSHP_ID</b>	Membership ID
<b>MBSHP_CANCEL_TS_UTC</b>	Membership Cancelation Date in UTC
<b>MBSHP_REBILL_TS_UTC</b>	Membership Rebill Date in UTC

#### 11.2.4.4 SUB\_CANCEL Gateway Response Example

```
<RESPONSE>
<REQUEST_ACTION>SUB_UPDATE</REQUEST_ACTION>
<API_RESPONSE>0</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE>102</SERVICE_RESPONSE>
<SERVICE_ADVICE>Membership Updated</SERVICE_ADVICE>
<REF_FIELD/>
<CUST_ID>291061</CUST_ID>
<MBSHP_ID>1234</MBSHP_ID>
<MBSHP_CANCEL_TS_UTC>ts '2023-09-03 20:46:19'</MBSHP_CANCEL_TS_UTC>
<MBSHP_REBILL_TS_UTC/>
</RESPONSE>
```

#### 11.2.5 Membership Product Update Requests (Upgrade/downgrade Product)

MERCHANTS may send product update requests using the Payment Service by sending `request_action=SUB_UPDATE` in the gateway request. Note that if the update product is a membership, a new membership record (with the same customer information) is created when the customer's current membership product has been successfully changed. The customer is charged

for the update product on the next upcoming rebill date of the customer's current membership. The customer's current membership will be canceled at the end of the subscription date.

Below are the minimum required parameters for SUB\_UPDATE requests:

#### **11.2.5.1 SUB\_UPDATE Request Parameters**

Field Name	Description	Data Type
<b>REQ_USERNAME</b>	Service Request Username	Alphanumeric
<b>REQ_PASSWORD</b>	Service Request Password	Alphanumeric and Special Characters
<b>REQUEST_RESPONSE_FORMAT</b>	Service Response Format	Accepted values: "XML" and "JSON"
<b>REQUEST_API_VERSION</b>	Payment Service API Version (4.9)	Numeric
<b>SITE_ID</b>	Merchant's Website ID	Numeric
<b>REQUEST_ACTION</b>	Service Request Action: send "SUB_UPDATE"	
<b>REQUEST_REF_MBSHP_ID</b>	Referring Membership ID.	Numeric
<b>SUB_UPDATE_PROD_ID</b>	Membership will be updated to the Product ID sent in this field.	Numeric
<b>SUB_UPDATE_PMT_ID</b>	Used to Update Card used for Membership (tied to specific Membership)	

#### **11.2.5.2 SUB\_UPDATE Response Parameters**

Field Name	Description
<b>REQUEST_ACTION</b>	Request Action Type: "SUB_UPDATE"
<b>API_RESPONSE</b>	API Response Code

<b>API_ADVICE</b>	API Response Advice
<b>SERVICE_RESPONSE</b>	Service Response Code
<b>SERVICE_ADVICE</b>	Service Response Advice
<b>REF_FIELD</b>	Reference Field (this will contain the field in question in case of data validation errors)
<b>CUST_ID</b>	Customer ID
<b>MBSHP_ID</b>	Membership ID
<b>MBSHP_CANCEL_TS_UTC</b>	Membership Cancelation Date in UTC
<b>MBSHP_REBILL_TS_UTC</b>	Membership Rebill Date in UTC

#### 11.2.5.3 SUB\_UPDATE Gateway Request Example (SUB\_UPDATE\_PMT\_ID)

MERCHANTS may update a Payment Card (PMT\_ID) through the gateway services for an existing Membership without having to cancel the Membership by sending **request\_action=SUB\_UPDATE** in the gateway request together with **SUB\_UPDATE\_PMT\_ID** Parameter with the replacement PMT\_ID. If the customer is adding a new card that doesn't exist in our system for that customer, clients shall run a **CCAUTHORIZE** for zero amount (\$0.00) to validate the card and create a PMT\_ID that'll be used with SUB\_UPDATE\_PMT\_ID.

```
/pmt_service.cfm?request_action=SUB_UPDATE&req_username=testmerchant@example.com&req_password=TestPass1234&request_ref_mbshp_id=1234&site_id=1234&SUB_UPDATE_PROD_ID=78890&request_api_version=4.8&SUB_UPDATE_PMT_ID=546789&cust_id=291061
```

#### 11.2.5.4 SUB\_UPDATE Gateway Response Example (SUB\_UPDATE\_PMT\_ID)

```
{
  "REQUEST_ACTION": "SUB_UPDATE",
  "API_RESPONSE": "0",
  "API_ADVICE": " ",
  "SERVICE_RESPONSE": 102,
  "SERVICE_ADVICE": "Membership Updated",
  "REF_FIELD": "",
  "CUST_ID": 7538691,
  "MBSHP_ID": 78245,
  "MBSHP_CANCEL_TS_UTC": "June, 28 2024 22:00:59",
  "MBSHP_REBILL_TS_UTC": ""
}
```

# 12 Response Fields

---

## 12.1 Response Formats

- Extensive Mark-up Language (XML)
- JavaScript Object Notation (JSON)

## 12.2 Successful Transaction Response Fields

These are the fields returned to the merchant when a transaction has been approved by the Payment Service.

Field Name	Description
<b>REQUEST_ACTION</b>	This will return the Service Request Action the merchant sent in the transaction request.
<b>TRANS_STATUS_NAME</b>	Transaction Status
<b>TRANS_VALUE</b>	Total requested transaction amount for all line items.
<b>CURR_CODE_ALPHA</b>	Requested Currency 3-letter Code
<b>TRANS_VALUE_SETTLED</b>	Transaction Settled Amount (after conversion to settled currency).
<b>CURR_CODE_ALPHA_SETTLED</b>	Settled Currency 3-letter Code
<b>TRANS_EXCH_RATE</b>	Currency Exchange Rate
<b>TRANS_ID</b>	Transaction ID
<b>CUST_ID</b>	Customer ID
<b>XTL_CUST_ID</b>	Merchant's Customer ID
<b>PO_ID</b>	Purchase order ID
<b>XTL_PO_ID</b>	Merchant's Order ID
<b>BATCH_ID</b>	Settlement Batch ID
<b>PROC_NAME</b>	Merchant Processor Name (Example: "EPX")
<b>MERCH_ACCT_ID</b>	Merchant Bank's Account ID
<b>CARD_BRAND_NAME</b>	Credit Card Network/Brand Name
<b>CARD_DETAIL</b>	Credit or Debit Card

<b>CARD_TYPE</b>	Credit Card Type
<b>CARD_CLASS</b>	Categorizes the BIN as a Business, Corporate, Purchase, or Consumer card
<b>CARD_COUNTRY</b>	Issuer Bank country for the BIN
<b>CARD_PREPAID</b>	Indicates that the credit card is a prepaid card if value returned is "1".
<b>CARD_BANK</b>	Credit Card Issuing Bank Name
<b>CARD_BALANCE</b>	Prepaid card balance (this is a processor-specific feature). This field will return the card's available balance.
<b>PMT_L4</b>	Payment account or credit card's last 4 digits.
<b>PMT_ID</b>	Payment Unique Identifier
<b>PMT_ID_XTL</b>	External Unique Identifier
<b>PROC_UDF01</b>	Processor User Defined Field 1
<b>PROC_UDF02</b>	Processor User Defined Field 2
<b>PROC_AUTH_RESPONSE</b>	Processor Authorization Response Code
<b>PROC_RETRIEVAL_NUM</b>	Processor Retrieval Number or GUID
<b>PROC_REFERENCE_NUM</b>	Processor Reference Number
<b>PROC_REDIRECT_URL</b>	URL where customers are redirected to for external verification (i.e. 3D Secure page)
<b>AVS_RESPONSE</b>	Address Verification Service Response Code
<b>CVV_RESPONSE</b>	Card Verification Value Response Code
<b>REQUEST_API_VERSION</b>	Payment Service API Version
<b>PO_LI_ID_X</b>	Purchase Order Line Item ID
<b>PO_LI_COUNT_X</b>	Purchase Order Line Item Count
<b>PO_LI_AMOUNT_X</b>	Purchase Order Line Item Total Amount
<b>PO_LI_PROD_ID_X</b>	Purchase Order Line Item Product ID
<b>MBSHP_ID</b>	Membership ID (returned on membership transactions)

<b>TRANS_NTOKEN_USED</b>	Used to indicate on whether a Network Token was used or a PAN was used. (It will be set to "1" if a scheme token was used or set to "0" otherwise)
<b>CARD_BRAND_TRANSID</b>	This is the card scheme transaction id for the current transaction

**Important Note**

If there is more than one line item in the transaction request you sent, the API will return the succeeding line item response fields for each line item that was created (PO\_LI\_ID\_X, PO\_LI\_COUNT\_X, PO\_LI\_AMOUNT\_X, and PO\_LI\_PROD\_ID\_X ("x" is to indicate a dynamic number depending on the number of line items sent in the request)). For example, **sending two line items** in your gateway request will get a response like this:

```
<<RESPONSE>
<PO_LI_ID_1>123455</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>19.95</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>1001</PO_LI_PROD_ID_1>
<PO_LI_ID_2>123456</PO_LI_ID_2>
<PO_LI_COUNT_2>1</PO_LI_COUNT_2>
<PO_LI_AMOUNT_2>10.5</PO_LI_AMOUNT_2>
<PO_LI_PROD_ID_2>2001</PO_LI_PROD_ID_2>
</RESPONSE>
```



**LINE ITEM FIELDS**

## 12.3 Declined Transaction Response Fields

These are the fields returned to the merchant when the transaction request has been declined on the Merchant Account Processor (bank) level:

Field Name	Description
<b>REQUEST_ACTION</b>	This will return the Service Request Action the merchant sent in the transaction request.
<b>TRANS_STATUS_NAME</b>	Transaction Status
<b>TRANS_VALUE</b>	Total transaction amount
<b>TRANS_ID</b>	Unique transaction ID
<b>CUST_ID</b>	Customer ID
<b>XTL_CUST_ID</b>	Merchant's Customer ID.

<b>MERCH_ACCT_ID</b>	[WL] Account ID
<b>CARD_BRAND_NAME</b>	Bank account or credit card type (Example: "MasterCard").
<b>PMT_L4</b>	Payment account or credit card's last 4 digits.
<b>API_RESPONSE</b>	Payment Service API Response Code. This is the higher level response information used for authentication and checking Service availability.
<b>API_ADVICE</b>	Payment Service API Response details.
<b>SERVICE_RESPONSE</b>	Service Response Code. The information in this field is for all transaction related requests (example: authorization requests and scrub-only transaction requests.)
<b>SERVICE_ADVICE</b>	Service Response details
<b>PROCESSOR_RESPONSE</b>	Merchant Account Processor Response Code
<b>PROCESSOR_ADVICE</b>	Merchant Account Service Response details
<b>INDUSTRY_RESPONSE</b>	Issuing Bank's Response Code
<b>INDUSTRY_ADVICE</b>	Issuing Bank's Response details
<b>REF_FIELD</b>	Reference field (useful when getting "missing required parameter" messages).
<b>PROC_NAME</b>	Merchant Processor Name (Example: Payvision)
<b>AVS_RESPONSE</b>	Address Verification Service Response Code
<b>CVV_RESPONSE</b>	Card Verification Value Response Code
<b>REQUEST_API_VERSION</b>	Payment Service API Version

## 12.4 Service Declined Response Fields

These are the fields returned to the merchant when the transaction request has been declined by the gateway before sending the transaction to the bank.

**NOTE:** Transactions that have been blocked by the gateway due to invalid format or missing data will not be recorded and will not show up in merchant-facing reports on the Portal.

- REQUEST\_ACTION – This will return the Service Request Action the merchant sent in the transaction request.
- TRANS\_STATUS\_NAME – Transaction Status.
- CUST\_ID – Customer ID.
- XTL\_CUST\_ID – Merchant's Customer ID.
- MERCH\_ACCT\_ID – Inovio Account ID.
- CARD\_BRAND\_NAME – Bank account or credit card type (Example: "Mastercard").
- PMT\_L4 – Payment account or credit card's last 4 digits.
- PMT\_ID – Payment Unique Identifier
- PMT\_ID\_XTL – External Unique Identifier
- API\_RESPONSE – Payment Service API Response Code. This is the higher level response information used for authentication and checking Service availability.
- API\_ADVICE – Payment Service API Response details.
- SERVICE\_RESPONSE – Service Response Code. The information in this field is for all transaction related requests (example: authorization requests and scrub-only transaction requests.)
- SERVICE\_ADVICE – Service Response details.
- PROCESSOR\_RESPONSE – Merchant Account Processor Response Code.
- PROCESSOR\_ADVICE – Merchant Account Service Response details.
- INDUSTRY\_RESPONSE – Issuing Bank's Response Code.
- INDUSTRY\_ADVICE – Issuing Bank's Response details.
- REF\_FIELD – Reference field (useful when getting "missing required parameter" messages).
- PROC\_NAME – Processor Name (returned as Null).
- AVS\_RESPONSE – Address Verification Service Response Code.
- CVV\_RESPONSE – Card Verification Value Response Code.
- REQUEST\_API\_VERSION – Payment Service API Version.

**Note**

Please check the [Appendix Section](#) for a list of different response codes and descriptions.

## 13 Dynamic Descriptor

---

### 13.1 Dynamic Descriptor

If supported by the Merchant Account Processor, merchants may send a Dynamic Descriptor in the transaction request using the **PMT\_DESCRIPTOR** parameter and/or the **PMT\_DESCRIPTOR\_PHONE** parameter. These parameters will replace the static statement descriptor that appears in the cardholder's bank statement.

#### 13.1.1 Dynamic Descriptor Restrictions

In very few cases will Dynamic descriptors function properly without prior arrangements being made with your Payment Processor. Dynamic descriptors may be restricted by your payment processor. Dynamic descriptors may also have format and length restrictions. Dynamic

descriptors may only take effect after a transaction has been settled. All of these issues should be reviewed with your payment processor before proceeding.

For more information on technical usage, please contact your gateway support representative about using Dynamic Descriptors.

## 14 Address Verification Service (AVS)

### 14.1 Address Verification Service

The Address Verification System is a system used to verify the address provided by the customer. AVS is generally supported by most banks in the United States and some foreign countries.

By default, the Payment Service will check AVS and return the AVS response code in the AVS\_RESPONSE field (as described in [Service Response Fields](#) section). Note that the response is ignored by the system unless AVS Check is enabled by the merchant.

#### 14.1.1 Check AVS Flag (CHKAVS)

As an option, merchants may choose to ignore, disable, or even set conditions on how transactions should be approved based on the AVS and CVV response codes returned by the processor.

CHKAVS Setting	Description
	AVS check is enabled.
T	Note: Transaction is approved if the AVS Response Code is any of the codes described in the <a href="#">Positive AVS Response Codes</a> table unless a custom AVSMATCHSET has been set by the merchant.
F	Do not check for AVS.
I	Check AVS but ignore the response. <b>(This is the default behavior.)</b>
	AVS response code will be ignored based on the result of the CVV check.
C	e.g. A transaction that returned the following response codes will be approved if CHKAVS parameter was set to "conditional": AVS RESPONSE CODE = N (No match) CVV2 RESPONSE CODE = M (MATCH)

#### 14.1.2 Positive AVS Response Codes

By default when AVS Check is enabled, the Payment Service will approve transactions that return any of the AVS Response Codes in the table below.

**Important Note:** Merchants may send a custom set of codes using the AVSMATCHSET parameter. See [Section 13.1.4](#) below for details.

AVS Response Code	Description
A	Street address matches, but 5-digit and 9-digit postal code do not match.
B	Street address matches, but postal code not verified.
D	Street address and postal code match. Code "M" is equivalent.
E	AVS data is invalid or AVS is not allowed for this card type.
F	Card member's name does not match, but billing postal code matches.
G	Non-U.S. issuing bank does not support AVS.
H	Card member's name does not match. Street address and postal code match.
I	Address not verified.
J	Card member's name, billing address, and postal code match.
L	Card member's name and billing postal code match, but billing address does not match.
M	Street address and postal code match. Code "D" is equivalent.
O	Card member's name and billing address match, but billing postal code does not match.
P	Postal code matches, but street address not verified.
Q	Card member's name, billing address, and postal code match.
R	System unavailable.
S	Bank does not support AVS.
T	Card member's name does not match, but street address matches.
U	Address information unavailable. Returned if the U.S. bank does not support non-U.S. AVS or if the AVS in a U.S. bank is not functioning properly.
V	Card member's name, billing address, and billing postal code match.
W	Street address does not match, but 9-digit postal code matches.
X	Street address and 9-digit postal code match.
Y	Street address and 5-digit postal code match.
Z	Street address does not match, but 5-digit postal code matches.

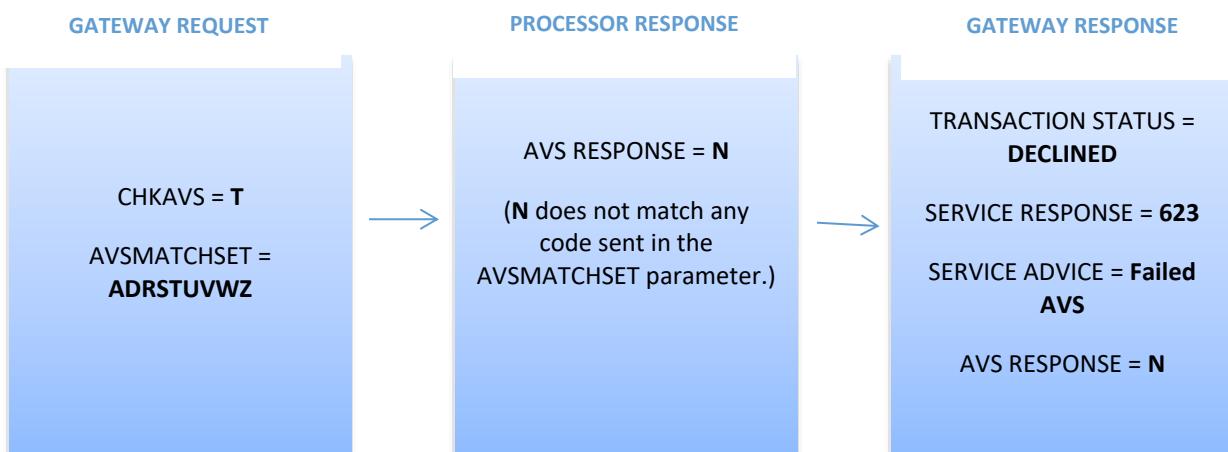
### 14.1.3 Negative AVS Response Codes

By default when AVS Check is enabled by the merchant, the Payment Service will decline all transactions that return any of the AVS Response Codes in the table below.

AVS Response Code	Description
C	Street address and postal code do not match.
K	Card member's name matches but billing address and billing postal code do not match.
N	Standard Response (US). Street address and postal code do not match.

### 14.1.4 Custom Settings (AVSMATCHSET)

The “AVSMATCHSET” parameter allows merchants to send a list of [AVS Response Codes](#). The Payment Service will return a “declined” response if the AVS response code returned by the processor **does not match** any of the codes sent in the AVSMATCHSET parameter.



## 14.2 Card Security Code

The Card Security Code (sometimes called CVV, CVV2, CVVC, CVC, CVC2 and V-code) is the 3 or 4-digit number located on the credit or debit card. The Payment Service requires that the merchant send these numbers in the `PMT_KEY` parameter by default.

**Important Note:** Some Merchant Account Processors may automatically decline transactions that return a negative CVV2 response. In this case, the merchant should see a “declined” transaction response message.

### 14.2.1 Check CVV Flag (CHKCVV)

As an option, merchants may choose to ignore, disable, or even set conditions on how transactions should be approved based on the CVV response codes returned by the processor.

CHKCVV Setting	Description
	CVV check is enabled.
T	Note: Transaction is approved if the CVV Response Code is any of the codes described in the <a href="#">Positive CVV Response Codes</a> table unless a custom <b>CVVMATCHSET</b> has been set by the merchant.
F	Do not send CVV to the bank.
I	Check CVV but ignore the response. <b>(This is the default behavior.)</b> CVV response code will be ignored based on the result of the AVS check.
C	e.g. A transaction that returned the following response codes will be approved if CHKCVV parameter was set to "conditional": CVV2 RESPONSE CODE = N (CVV Match) AVS RESPONSE CODE = M (No match)

#### 14.2.2 Positive CVV Response Codes

By default when CVV Check is enabled by the merchant, the Payment Service will approve all transactions that return any of the CVV Response Codes in the table below.

CVV Response Code	Description
M	Match
P	Not Processed
S	Not Supported
U	Service Not Available
X	No CVC/CVV/CVV2/CID Response Data Available
(empty)	No CVC/CVV/CVV2/CID Response Data Available

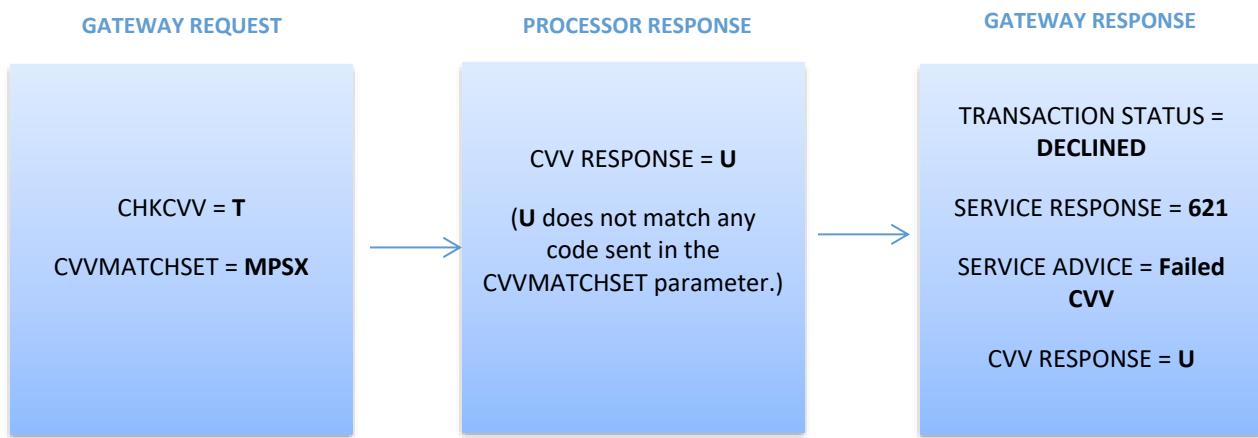
#### 14.2.3 Negative CVV Response Codes

By default when CVV Check is enabled by the merchant, the Payment Service will decline all transactions that return any of the CVV Response Codes in the table below.

CVV Response Code	Description
N	No match

#### 14.2.4 Custom Settings (CVVMATCHSET)

The “CVVMATCHSET” parameter allows merchants to send a list of [CVV Response Codes](#). The Payment Service will return a “declined” response if the CVV response code returned by the processor **does not match** any of the codes sent in the CVVMATCHSET parameter.



## 15 3D Secure Transactions

### 15.1 3D Secure

The payment gateway supports 3DS transactions for both 3DS version 1.x and 3DS version 2.x. The user of this document is assumed to understand the differences between these versions. Contact your gateway support representative for more information. The payment gateway supports 3DS for all card brands and in different scenarios.

#### 15.1.1 Low Value SCA Exemption

Where SCA is mandated but transactions qualify for SCA Exemptions for reasons like low value transactions, the Gateway has no way to know when such exemptions have been met. Where the exemption is supported by the Processor, the gateway shall make available the `REQ_LOW_VALUE_SCAEXEMPTION` parameter for merchants to use when they want a transaction to bypass SCA.

#### 15.1.2 External 3DS Provider Scenario

In this scenario, the merchant uses an external 3DS provider, and the 3DS authentication is completed before authorization requests are sent to the gateway.

##### 15.1.2.1 3DS Authentication Data Gateway Parameters.

The payment gateway has defined parameters for 3DS on authorization request and these parameters are named in accordance with the 3DS standards which have been established in the industry. These are described in the table below. These parameters should be sent (with data provided by the external 3DS provider) along with the authorization request.

Field Name	Description
P3DS_CAVV	Cardholder Authentication Verification Value
P3DS_ECI	Electronic Commerce Indicator

<b>P3DS_XID</b>	Transaction ID
<b>P3DS_VERSION</b>	Reports on 3DS Version used to process Transaction (Required for Mastercard Identity Check transactions in Authorization on 3DS 2)
<b>P3DS_TRANSID</b>	Unique transaction identifier assigned by the Directory Server (DS)-( Required for Mastercard Identity Check transactions in Authorization IF P3DS_VERSION is 3DS 2)
<b>P3DS_SCREEN_HEIGHT</b>	Total height of the cardholder's screen in pixels
<b>P3DS_SCREEN_WIDTH</b>	Total width of the cardholder's screen in pixels
<b>P3DS_JAVA_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents the ability of the cardholder browser to execute Java
<b>P3DS_JAVASCRIPT_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents represents the ability of the cardholder browser to execute JavaScript
<b>P3DS_BROWSER_HEADER</b>	The exact content of the HTTP accept headers sent from the cardholder's browser. Example: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
<b>P3DS_BROWSER_LANGUAGE</b>	Value represents the browser language as defined in IETF BCP47
<b>P3DS_BROWSER_COLOR_DEPTH</b>	Value represents the bit depth of the color palette for displaying images, in bits per pixel Possible Values: 1, 4, 8, 15, 16, 24, 32, 48
<b>P3DS_BROWSER_TIME_ZONE</b>	Time difference between UTC time and the cardholder browser local time, in minutes
<b>P3DS_CHALLENGE_WINDOW</b>	An override field that a merchant can pass in to set the challenge window size to display to the end cardholder. The ACS will reply with content that is formatted appropriately to this window size to allow for the best user experience. The sizes are width x height in pixels of the window displayed in the cardholder browser window. Possible values: 01 - 250x400, 02 - 390x400, 03 - 500x600, 04 - 600x400, 05 - Full page
<b>USER_AGENT_XTL</b>	Software agent responsible for retrieving and facilitating end-user interaction with Web content
<b>XTL_IP</b>	Cardholder's IP Address

### 15.1.3 Payment Gateway as 3DS Provider Using Redirect Scenario

The following steps and examples provide a guide on how to use 3DS 2.0 with Inovio's payment gateway through a Merchant's own payment page:

#### 15.1.3.1 Make a request to Inovio's API to retrieve Device Data Collection (DDC) references and a JWT

15.1.3.1.1 Inovio API request – DDC - See the process {A} in Figure 1 below  
**Endpoint:** <https://api.inoviopay.com/payment/3dsrequest.cfm>

**Parameters:**

Field Name	Description
------------	-------------

<b>REQ_USERNAME*</b>	API credential username
<b>REQ_PASSWORD*</b>	API credential password
<b>MERCH_ACCT_ID</b>	Numeric merchant account identifier
<b>REQUEST_API_VERSION=4.8*</b>	Version of the API
<b>BILL_ADDR_COUNTRY*</b>	2-letter ISO 3166-1 Alpha-2 code
<b>REQUEST_CURRENCY</b>	3-letter ISO-4217 code
<b>SITE_ID</b>	Merhcnat's site identifier
<b>PMT_BIN</b>	First 6 digits of the card number
<b>CUST_ID</b>	Customer ID Required when using a tokenized card. Cannot be sent with <b>PMT_BIN</b>
<b>PMT_ID</b>	Payment ID Required when using a tokenized card. Cannot be sent with <b>PMT_BIN</b>
<b>P3DS_SCREEN_HEIGHT</b>	Total height of the cardholder's screen in pixels
<b>P3DS_SCREEN_WIDTH</b>	Total width of the cardholder's screen in pixels
<b>P3DS_JAVA_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents the ability of the cardholder browser to execute Java
<b>P3DS_JAVASCRIPT_ENABLED</b>	A Boolean value (TRUE/FALSE) that represents represents the ability of the cardholder browser to execute JavaScript The exact content of the HTTP accept headers sent from the cardholder's browser. Example: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
<b>P3DS_BROWSER_HEADER</b>	
<b>P3DS_BROWSER_LANGUAGE</b>	Value represents the browser language as defined in IETF BCP47
<b>P3DS_BROWSER_COLOR_DEPTH</b>	Value represents the bit depth of the color palette for displaying images, in bits per pixel Possible Values: 1, 4, 8, 15, 16, 24, 32, 48
<b>P3DS_BROWSER_TIME_ZONE</b>	Time difference between UTC time and the cardholder browser local time, in minutes
<b>P3DS_CHALLENGE_WINDOW</b>	An override field that a merchant can pass in to set the challenge window size to display to the end cardholder. The ACS will reply with content that is formatted appropriately to this window size to allow for the best user experience. The sizes are width x height in pixels of the window displayed in the cardholder browser window. Possible values: 01 - 250x400, 02 - 390x400, 03 - 500x600, 04 - 600x400, 05 - Full page

<b>USER_AGENT_XTL</b>	Software agent responsible for retrieving and facilitating end-user interaction with Web content
<b>XTL_IP</b>	Cardholder's IP Address

**Note:**

\* = REQUIRED

IF **MERCH\_ACCT\_ID** IS NOT PROVIDED, THEN THE FOLLOWING PARAMETERS ARE REQUIRED SO THAT MERCHANT ACCOUNT DISTRIBUTION RULES CAN BE ACTIVATED:

- **BILL\_ADDR\_COUNTRY**
- **REQUEST\_CURRENCY**
- **SITE\_ID**

**15.1.3.1.2 Inovio API response – DDC - See the data {B} in Figure 1 below**

{

```
"JWT": "eyJhbGciOiJIUzI1NiJ9.eyJpYXQiOjE2MDgwNjQ4NjksIk9yZ1VuaXRJZCI6IjVhZjRhNTRiMGU0MjNkMjY4ODA4MmRjZCIsI1J1ZmVyzW5jZUlkijoINTU1MTA0MS0yMDAxMyIsImlzcyI6IjVkmjc5OGIyNmZlM2QxMje1Y2M2ZDg4YyIsImp0aSI6Iju1NTEwNDEifQ.HYta+F3+1Im7yw0N1BFSiAEFctSE9er0A1cd2px6Nl0",
"MERCH_ACCT_ID": "20013",
"DDC_URL": "https://centinelapistag.cardinalcommerce.com/V2/Cruise/Collect",
"DDC_REFERENCEID": "5551041-20013",
"PMT_BIN",
"MERCH_ACCT_ID"
```

}

**Note:**

- DDC\_REFERENCEID SHOULD BE STORED FOR LATER USE.
- MERCH\_ACCT\_ID, IF NOT ALREADY KNOWN, SHOULD BE STORED FOR LATER USE.
- PMT\_BIN SHOULD BE USED IN THE NEXT STEP.
- THE DATA ABOVE IS FOR EXAMPLE USE ONLY – THE DATA RETURNED IN THE MERCHANT'S ENVIRONMENT WILL BE DIFFERENT.

**15.1.3.2 DDC Starts - See the process {C} in Figure 1 below**

DDC takes place in an iframe and requires the JWT and BIN number of the customer's card. This form should be submitted **before** the checkout form is submitted.

```
<form name="ddcForm" id="ddcForm" method="POST">
    <input type="hidden" name="JWT" id="JWT"/>
    <input type="hidden" name="Bin" id="Bin"/>
</form>
```

An eventListener should be added so that the DDC can finish before proceeding with submission of the checkout form. An example of this is as follows:

```
request.onload = function (req) {
    var json = JSON.parse(req.currentTarget.response || req.target.responseText);

    window.addEventListener("message", function(event) {
        if (event.origin === "https://centinelapistag.cardinalcommerce.com") {
            //Here you call your server side to make the call to 3dsrequest.cfm;
        }
    }, false);
    //populate DDC form
};

request.send();
```

**NOTE:** If the DDC is not allowed time to complete, the transaction will revert to **3DS 1.0**.

### 15.1.3.3 DDC Complete

Once the DDC finishes, and the checkout form is submitted, the merchant should send a transaction request to inovio's gateway API

#### 15.1.3.3.1 Inovio Payment Gateway API Request - *See the process {D} in Figure 1 below*

Please refer to the Inovio Gateway API Documentation for full usage and parameter details

Request parameters that are specific to 3DS and specific values to use are:

```
P3DS_VERSION:2
    //TELLS THE GATEWAY WE WANT 3DS 2.0
REQUEST_ENROLLMENT:1
    //TELLS THE GATEWAY TO CHECK THE CARD'S 3DS ENROLLMENT
P3DS_RETURN_URL: HTTPS://MERCHANTWEBSITE.COM/SOME_ENDPOINT
    //URL USED TO RECEIVE REDIRECT BACK FROM CARD ISSUER'S ACS
DDC_REFERENCEID
    //THE REFERENCE ID WHICH WAS RETURNED FROM THE DDC API REQUEST
```

### 15.1.3.3.2 Inovio Payment Gateway API Response

There are three possible outcomes in the response:

- Outcome 1: The most likely, which is that 3DS version 2 authentication will be required
  - Outcome 2: In some rare cases, the card's issuing bank may force the older 3DS version 1 authentication to be required
  - Outcome 3: In some cases, the issuer will allow a “frictionless” authentication, or the issue will not support any authentication; in either of these cases, the transaction will be fully completed with no further action needed.

### 15.1.4 Determining which outcome happened

The table shows how to determine which of the three outcomes happened, according to parameters returned by the API: `TRANS_STATUS_NAME` and `P3DS_PROCVERSION`

	TRANS_STATUS_NAME	P3DS_PROCVERSION
Outcome 1 (3DS 2)	PENDING	2.x
Outcome 2 (3DS 1)	PENDING	1.x
Outcome 3 (no action)	APPROVED or DECLINED	n/a

#### **15.1.4.1 Outcome 1: 3DS version 2 authentication - See the data {K} in Figure 1 below**

### Note:

THE DATA ABOVE IS FOR EXAMPLE USE ONLY – THE DATA RETURNED IN THE MERCHANT'S ENVIRONMENT WILL BE DIFFERENT.

### **15.1.4.2 Outcome 2: 3DS version 1 authentication (Deprecated)**

This is outlined in elements {F} through {J} in Figure 1 below

#### 15.1.4.2.1 Payment Gateway as 3DS Provider Using Redirect Scenario

In this scenario, the payment gateway can be the 3DS provider and enrollment checks will be done via the gateway prior to the authorization request.

Note that this scenario provides a service which must be provisioned at the gateway before it can be used. If this is not yet done or it is not certain, please contact your gateway support representative for more information.

#### 15.1.4.2.2 Payment Gateway as 3DS Provider Using Redirect Scenario Workflow

This 3-D secure transaction process consists of two requests from the merchant. The first is the enrollment check, and the second is the transaction attempt. The full process occurs in this order:

- Merchant submits a CCAUTHCAP or CCAUTHORIZE for the enrollment check. The parameter request\_enrollment must be set to 1 for the request to be treated as an enrollment check.
- The gateway sends the card information to a downstream 3DS ACI and receives a response which indicates the card's 3DS enrollment status.
- If the card is not eligible for 3DS (usually this means that the issuer is not participating in 3DS), then the gateway will immediately attempt an authorization without 3DS and this will be processed in the usual manner described earlier in this document
- If the card is either eligible for or enrolled in 3DS, then the gateway will respond with the following important 3DS-specific parameters:
  - PROC\_REDIRECT: This is a URL to which the customer must be redirected to perform 3DS authentication.
  - P3DS\_PAREQ: This is a parameter which should be included as an HTTP POST variable on the redirect.
  - P3DS\_TRANSID: This must be saved and re-transmitted back to the gateway after the 3DS authentication process is completed.
- The merchant should redirect the customer (this can be done in the same window, a separate window or an iframe) to URL they received in PROC\_REDIRECT\_URL via HTTP POST method (GET will result in an error) and the merchant should include the following data as POST parameters on the redirect:
  - PaReq: The value of P3DS\_PAREQ from the prior gateway response. NOTE: the parameter name is case sensitive and must be exactly "PaReq".
  - TermUrl: This is the merchant's URL to which the customer will be redirected back after the 3DS authentication process is completed. NOTE: the parameter name is case sensitive and must be exactly "TermUrl".
  - MD: This is an optional parameter. It can be used by the merchant to maintain session identifier(s) in case this is needed to maintain the customer's session with the merchant site between redirects.
- The customer will complete the authentication process inside the PROC\_REDIRECT URL. Note that there are many different outcomes from the authentication process. In all outcomes, the customer should be redirected via HTTP POST back to the merchant on the TERMURL.
- When the customer returns back to the merchant site on the TERMURL, this will be done via HTTPD POST. The merchant will capture the following data as POST parameters on the redirect:
  - PARES: This value will be used to complete the 3DS authentication process at the gateway.
  - MD: This is the optional session data described above.

- Once the customer returns from the 3DS authentication process, the merchant will again submit a CCAUTHCAP or CCAUTHORIZE, and include the following extra parameters:
  - P3DS\_TRANSID: This is the value of the P3DS\_TRANSID parameter which was received in first gateway response described above.
  - REQUEST\_PARES: This is the value of the PARES parameter which was passed back via HTTP POST after the 3DS process was completed.
- The payment gateway will extract the PARES value into 3DS authentication data. Then the gateway will process the authorization request as usual with the 3DS authentication data appended.
- The gateway response data may include extra parameters to describe any 3DS conditions which might apply. See the full parameter list below for more information.

#### **15.1.4.3 Payment Gateway as 3DS Provider Using Redirect Scenario 3-D Secure Parameters**

These parameters are specific to the Payment Gateway as 3DS Provider Using Redirect Scenario.

##### **15.1.4.3.1 Enrollment Check**

###### **15.1.4.3.1.1 Request Parameters**

Field Name	Description
REQUEST_ENROLLMENT	0 (off) 1(on)

###### **15.1.4.3.1.2 Enrollment Check Request Example**

```
https://api.inoviopay.com/payment/pmt_service.cfm
?cust_fname=John&cust_lname=Sample&pmt_expiry=012023&li_count_1=1&li_prod_id_1=32144&
li_value_1=3.00&bill_addr_country=US&merch_acct_id=25109&request_currency=USD&request_enrollment=1&req_username=api%40example.com&req_password=Password123&site_id=23803&r
equest_response_format=JSON&request_action=CCAUTHCAP&request_api_version=4.8&pmt numb
=400000000001091&PROC_SUCCESS_URL=http://www.example.com&PROC_ERROR_URL=http://www.e
xample.com
```

###### **15.1.4.3.1.3 Enrollment Check Response Parameters**

Field Name	Description
PROC_REDIRECT_URL	3D Secure Verification URL where the customer should be redirected to for authentication.
P3DS_TRANSID	3D Secure Vendor's Transaction ID
P3DS_PAREQ	Payment Authentication Request

###### **15.1.4.3.1.4 Enrollment Check Response Example**

```
{
: "REQUEST_ACTION":"CCAUTHCAP",
: "REQ_ID":"4888098",
: "TRANS_STATUS_NAME":"PENDING",
: "TRANS_VALUE":"",
: "TRANS_VALUE_SETTLED":"",
: "CURR_CODE_ALPHA_SETTLED":"",
: "TRANS_EXCH_RATE":"",
: "TRANS_ID":""}
```

```

: "CUST_ID":"",
: "XTL_CUST_ID":"",
: "CUST_BRCPFCNPJ":"",
: "PO_ID":"",
: "XTL_ORDER_ID":"",
: "BATCH_ID":"",
: "PROC_NAME":"",
: "MERCH_ACCT_ID":"12345",
: "CARD_BRAND_NAME":"",
: "PMT_L4":"",
: "PMT_ID":"",
: "PMT_ID_XTL":"",
: "PROC_UDF01":"",
: "PROC_UDF02":"",
: "PROC_AUTH_RESPONSE":"",
: "PROC_RETRIEVAL_NUM":"",
: "PROC_REFERENCE_NUM":",

: "PROC_REDIRECT_URL":"https://3ds.example.com/merchantacsfrontend/pareq.jsp?vaa=b&gold=AAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAA",
: "AVS_RESPONSE":"",
: "CVV_RESPONSE":"",
: "REQUEST_API_VERSION":"4.6",
: "P3DS_RESPONSE":"Full Authentication",
: "P3DS_VENDOR":"Cardinal",
: "P3DS_ECI":"07",
: "P3DS_TRANSID":"EnK4GI8WOhkBRVzQDUIo",
:

"P3DS_PAREQ":"eNpVUV1vgjAUfe+vIGbPtAVUYq5N/JpjH+pQNNkbgWaQadFSjNuvX4sw3ds9997TnnMu
bDLJ+XTNkopyBm+8LONPbuXpsDMTL9488HfL7Gscbn/ep1FAOgxWo5CfGJy5LPNCMGoT2wHcQqSfkEk
WC8UgTk7jYMGo43rdXt8H3DQQHLgMpneD8i3u4hf74h57yw2esDS8V4HqGICkqoeQ38zoCuAUIKrlnm
VLHcoBxLgzfTooDYNNHgG+iVpWpSu3lkqcs3HVF7PoefmYruK5mkT7j14Uhdv1ZjQEbDYQpLHizCHUJz1
CLeoPPDIgfcB1HoF8MCLYg2sTrahBCI7mn9EVuWZy39A2Kim5SFofLULAL8dCcL2jQ/2rtYOb7smTiTZRJ
kHX9SilJtka1/RcB+IQQmt+XqeDDQc3l8PNkXX17/i/rCWomA=="
}

```

### 15.1.4.3.2 3D Secure Purchase

#### 15.1.4.3.2.1 Request Parameters

Field Name	Description
P3DS_TRANSID	Transaction ID returned in first response, such as in the above example: P3DS_TRANSID:EnK4GI8WOhkBRVzQDUIo
REQUEST_PARES	The value returned from the enrollment check, which is passed to the designated <b>TERMURL</b> described in Step 4 above

#### 15.1.4.3.2.2 3D Secure Purchase Example

```

https://api.inoviopay.com/payment/pmt_service.cfm
?cust_fname=James&cust_lname=Sample&pmt_expiry=012020&li_count_1=1&li_prod_id_1=32805
&li_value_1=3.00&xtl_udf01=udfex&bill_addr_country=US&merch_acct_id=25109&request_currency=USD&P3DS_TRANSID=EnK4GI8WOhkBRVzQDUI0&REQUEST_PARES=eNqdmFuTqsiygn%2BN8D%2BsmP3
ozACKtwm7IwoobgrKXXzjJncQQUb%2B%2FS519eqetVfsPef0SxdpV1ZWZeZXCRs9ugYBowXe7Rq8b6Sgrp0w

```

```
%2BBb7b7%2FBYktywsraRym1moPCGAL%2B2%2FvmANSgfio8R8suECrdiwI6uJ261XOYDEWWFTQp1m9Iuw2ud
Vw&req_username=api@example.com&req_password=Password123&site_id=12345&request_respon
se_format=JSON&request_action=CCAUTHCAP&request_api_version=4.8&pmt numb=400000000000
0002& PROC_SUCCESS_URL=http://www.example.com&PROC_ERROR_URL=http://www.example.com
```

#### 15.1.4.3.2.3 3D Secure Response Parameters

MERCHANTS using 3D SECURE VERIFICATION will receive these additional parameters in the gateway response.

Field Name	Description
<b>SERVICE_RESPONSE</b>	This is the standard gateway service response, but may contain a 3DS-specific code in certain decline cases
<b>SERVICE_ADVICE</b>	This is the standard gateway service response, but may contain 3DS-specific information in certain decline cases
<b>P3DS_RESPONSE</b>	3D Secure service response
<b>P3DS_VENDOR</b>	3D Secure downstream provider ( <i>informational only</i> )

#### 15.1.4.4 Outcome 3: No further action needed

This is the process found immediately prior to data {K} in Figure 1 below

- After the payment gateway request is completed, the actual 3DS authentication process can proceed on the checkout page.  
ALL OF THE FOLLOWING STEPS FOLLOW FROM OUTCOME 1 IN THE PRECEDING SECTION
- A form will need to be submitted to the PROC\_REDIRECT\_URL with the parameters shown in the below image.

See the data {L} in Figure 1 below

#### NOTE:

PLEASE NOTE THAT FOR 3DS 2.0 TRANSACTIONS, A JWT IS REQUIRED. IF A JWT IS NOT PRESENT, YOU WILL NEED TO REVERT TO 3DS 1.0.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
<script type="text/javascript">
window.onload = function () {
    document.forms['stepUpForm'].submit();
}
</script>
</head>
<body>
<form id="stepUpForm" method="post" action="${url}">
    <input type="hidden" name="JWT" value="${jwt}" />
    <input type="hidden" name="MD" value="${md}" />

```

```
</form>
</body>
</html>
```

#### FORM POST ACTION

//THIS SHOULD BE THE VALUE OF PROC\_REDIRECT\_URL FROM THE INOVIO API RESPONSE

DESCRIBED ABOVE

#### HIDDEN INPUT FIELD JWT

//THIS SHOULD BE THE VALUE OF P3DS\_JWT FROM THE INOVIO API RESPONSE DESCRIBED ABOVE

#### HIDDEN INPUT FIELD MD

//THIS IS AN OPTIONAL PARAMETER WHICH CAN BE USED BY THE MERCHANT TO STORE ITS OWN  
REFERENCE DATA/IDENTIFIER

- After the above form is submitted to the PROC\_REDIRECT\_URL, the user will authenticate with the ACS, and then be redirected back to the merchant's website via the P3DS\_RETURN\_URL.

*See the data {M} in Figure 1 below*

- The redirect back from PROC\_REDIRECT\_URL to P3DS\_RETURN\_URL will be a POST, and will contain the POST parameters TRANSACTIONID, RESPONSE, and MD. At this point, the merchant should use the data in those parameters to make a final request to Inovio novio

Gateway API to perform a transaction.

*See the data {N} in Figure 1 below*

Please refer to the Inovio Gateway API Documentation for full usage and parameter details

Request parameters that are specific to 3DS and specific values to use are:

#### P3DS\_PROCTRANSID

//THIS SHOULD BE THE VALUE OF TRANSACTIONID FROM THE ACS

#### REQUEST\_PARES

//THIS SHOULD BE THE VALUE OF RESPONSE FROM THE ACS

#### P3DS\_VERSION: 2

//TELLS THE GATEWAY WE WANT 3DS 2.0

#### Note:

THE VALUE OF THE "RESPONSE" PARAMETER FROM THE REDIRECT BACK TO THE RETURN URL MAY BE EMPTY. THIS IS NOT AN ERROR. THAT EMPTY VARIABLE SHOULD BE PASSED AS THE EMPTY VALUE FOR REQUEST\_PARES ON THE SUBSEQUENT GATEWAY REQUEST.

THE RESPONSE FROM THE GATEWAY API WILL CONTAIN THE RESULT OF THE TRANSACTION. IF SUCCESSFUL, THE TRANSACTION WILL HAVE THE 3DS AUTHENTICATION ATTACHED TO IT AT THE PROCESSOR.

*See the data {O} in Figure 1 below*

#### 15.1.4.5 Special cases

##### 15.1.4.5.1 Using Inovio API 3DS 2.0 Guide With Tokenized Cards

THIS IS USED FOR ANY CIT (CUSTOMER INITIATED TRANSACTION) USING A TOKEN OR "ONE-CLICK" WITH 3DS 2.0

To use tokenized cards, adhere to the following practices in the documented flow:

- Make sure to send CUST\_ID and PMT\_ID values when making request to <https://api.inoviopay.com/payment/3dsrequest.cfm>  
This will allow Inovio's system to look up the card value and provide the PMT\_BIN value which will be necessary in later steps
- Make sure to capture the PMT\_BIN value from the response from <https://api.inoviopay.com/payment/3dsrequest.cfm>
- Make sure to send the same CUST\_ID and PMT\_BIN values to the gateway API <https://api.inoviopay.com/payment/3dsrequest.cfm> in all subsequent steps. See the Inovio Payment Gateway API documentation for more details.

#### 15.1.4.5.2 Using Inovio API 3DS 2.0 Guide with Inovio's Merchant Account Distribution

Inovio's MERCHANT ACCOUNT DISTRIBUTION ENABLES A MERCHANT TO INTELLIGENTLY USE ALL OF THEIR MERCHANT ACCOUNTS AND ROUTE TRANSACTIONS TO THE APPROPRIATE MERCHANT ACCOUNT, BASED ON CARD ISSUER, LOCALE, CURRENCY, CARD TYPE, AND OTHER AVAILABLE DATA.

To use Merchant Account Distribution, adhere to the following practices in the documented flow:

- Note that the value for MERCH\_ACCT\_ID will not be known at the start of this process, but will be determined in the first step, and then must be captured and used in later steps.
- Be sure to send BILL\_ADDR\_COUNTRY, REQUEST\_CURRENCY, and SITE\_ID values when making request to <https://api.inoviopay.com/payment/3dsrequest.cfm>  
This will allow Inovio's system to look up the card value and provide the MERCH\_ACCT\_ID value which will be necessary in later steps
- Make sure to capture the MERCH\_ACCT\_ID value from the response from <https://api.inoviopay.com/payment/3dsrequest.cfm>
- Make sure to send the MERCH\_ACCT\_ID value to the gateway API <https://api.inoviopay.com/payment/3dsrequest.cfm> in all subsequent steps. See the Inovio Payment Gateway API documentation for more details.

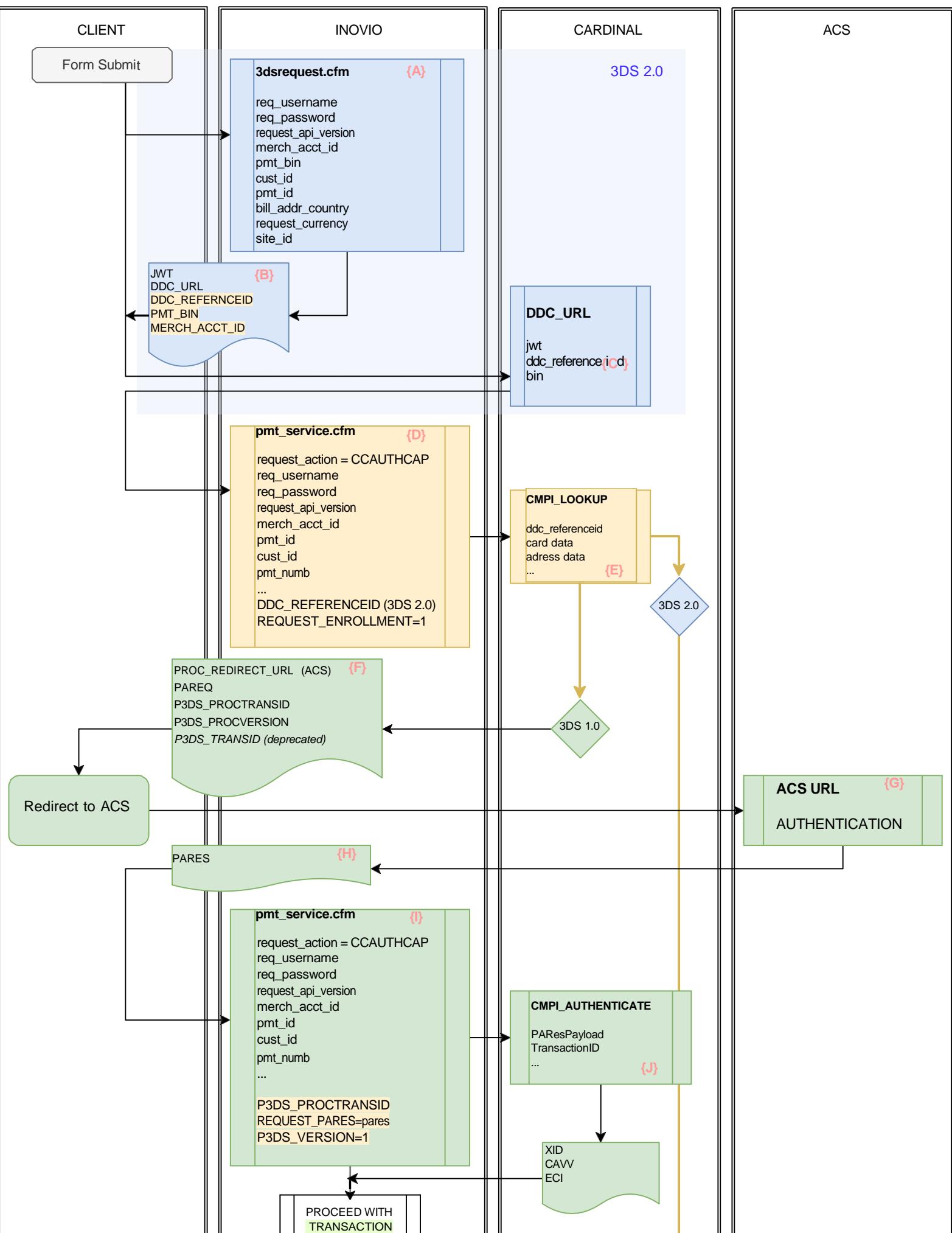
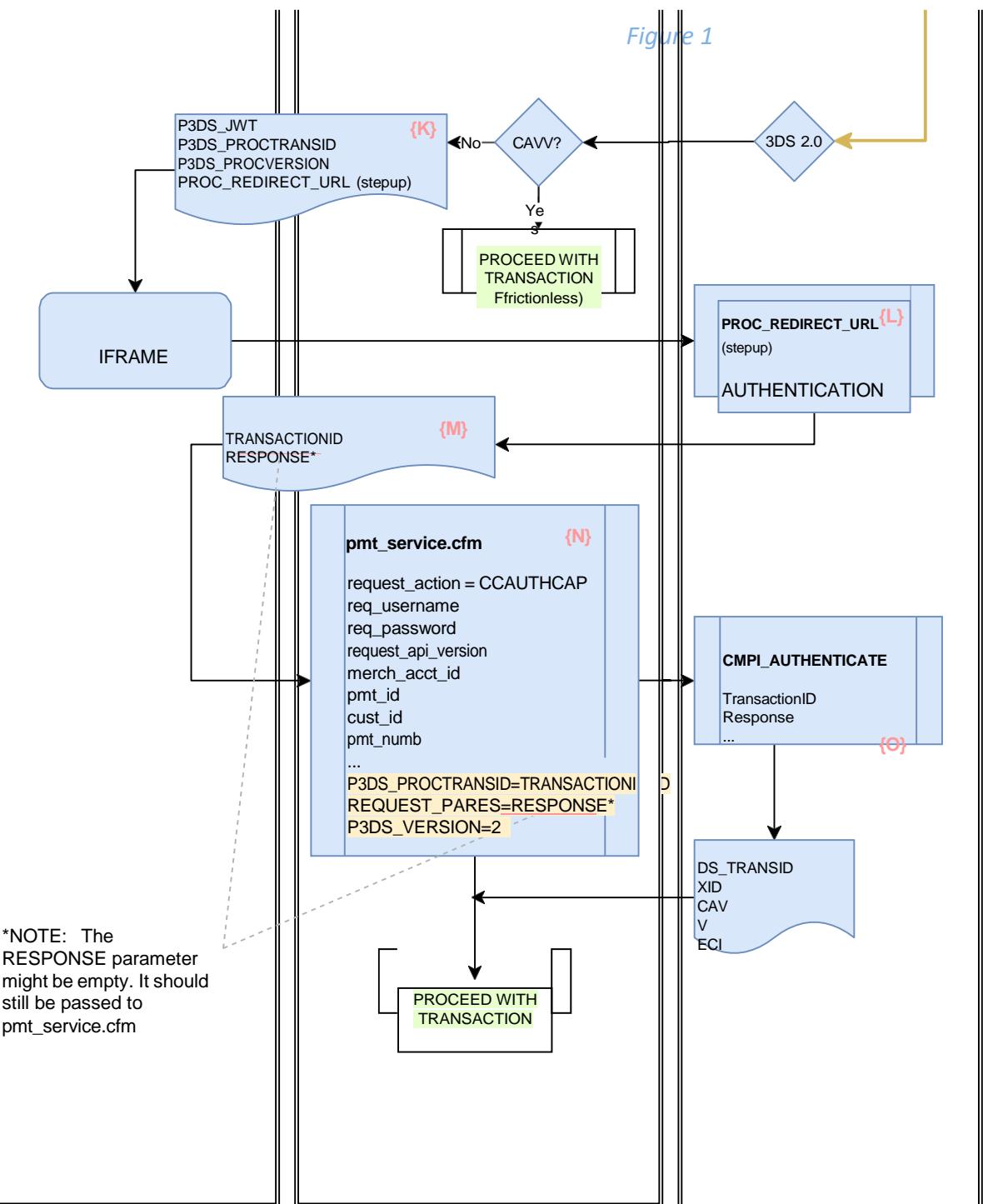


Figure 1



## 16 Currency

### 16.1 Multi-currency Support

The payment gateway supports all foreign currencies; however, availability to settle transactions in different currencies depends on the merchant account provider. Send an inquiry to your gateway support representative before implementing non-US Dollar transactions in your system.

## 16.2 Request\_currency Parameter

To set the currency requested in your transaction request, send the 3-letter ISO-4217 ([http://en.wikipedia.org/wiki/ISO\\_4217](http://en.wikipedia.org/wiki/ISO_4217)) currency code in the **request\_currency** parameter.

## 16.3 Currency Response Fields

The table below explains currency related gateway response fields (available on Payment Service API version 2.8 and above).

Field Name	Description
<b>TRANS_VALUE</b>	Requested transaction amount
<b>CURR_CODE_ALPHA</b>	Requested Currency (3-letter code)
<b>TRANS_VALUE_SETTLED</b>	Settled amount after currency conversion
<b>CURR_CODE_ALPHA_SETTLED</b>	Settled Currency (3-letter code)
<b>TRANS_EXCH_RATE</b>	Currency Conversion Rate (for more information, contact your gateway support representative)

### 16.3.1 Testing Multi-currency Response

An example of multi-currency response can be tested by using the Test Processor. Submitting a transaction to the test processor specifically with a EUR amount of 7.00 will yield an example of currency conversion in the response.

**!** Because this is a fictitious test scenario for demonstration purposes, the amounts in the response will always be the same, and *will not reflect* the actual EUR to USD conversion rate. This testing scenario will only work with the Test Processor.

### 16.3.2 Multi-Currency Response Example

In the response example below, the amount submitted was

```
{
: "REQUEST_ACTION":"CCAUTHORIZE",
: "REQ_ID":"4934041",
: "TRANS_STATUS_NAME":"APPROVED",
: "TRANS_VALUE":7,
: "CURR_CODE_ALPHA":"EUR",
: "TRANS_VALUE_SETTLED":"5.686415",
: "CURR_CODE_ALPHA_SETTLED":"USD",
: "TRANS_EXCH_RATE":0.812345,
: "TRANS_ID":2342342,
: "CUST_ID":242342235,
: "XTL_CUST_ID":"",
: "PO_ID":345345345,
: "XTL_ORDER_ID":"",
: "BATCH_ID":23423423,
: "PROC_NAME":"Test Processor",
: "MERCH_ACCT_ID":333333,
: "CARD_BRAND_NAME":"Visa",
```

```

: "CARD_TYPE":"",
: "CARD_PREPAID":"",
: "CARD_BANK":"",
: "CARD_BALANCE":"",
: "PMT_L4":"0002",
: "PMT_ID":5345453434,
: "PMT_ID_XTL":"",
: "PROC_UDFO1":"",
: "PROC_UDFO2":"",
: "PROC_AUTH_RESPONSE":"TEST23969",
: "PROC_RETRIEVAL_NUM":"7781C275-625C-4E10-A25651241742F7D8",
: "PROC_REFERENCE_NUM":"TEST976291250",
: "PROC_REDIRECT_URL":"",
: "AVS_RESPONSE":"M",
: "CVV_RESPONSE":"M",
: "REQUEST_API_VERSION":"4.0",
: "P3DS_RESPONSE":"",
: "P3DS_VENDOR":"",
: "PO_LI_ID_1":"2342342",
: "PO_LI_COUNT_1":"1",
: "PO_LI_AMOUNT_1":"7.00",
: "PO_LI_PROD_ID_1":"22222",
: "MBSHP_ID_1":""
}

```

## 17 Partial Authorization

### 17.1 What is Partial Authorization?

Partial Authorization is a processor-specific feature that allows merchants to accept partial amount authorizations usually on prepaid and debit cards.

For inquiries regarding Partial Authorization support on your merchant account, contact your gateway support representative.

#### 17.1.1 How It Works

- Merchant sends the Partial Authorization (PARTIAL\_AUTH parameter) flag in their gateway transaction request, along with the minimum amount the merchant will accept for authorization (PARTIAL\_AUTH\_MIN parameter).
- Payment Service will attempt to authorize the full requested amount sent in LI\_VALUE\_1 parameter. If the requested minimum or full amount is available on the customer's card, the merchant will receive an “Approved” transaction response.

**NOTE:** The amount available for authorization will be returned in TRANS\_VALUE gateway response field. Merchants must parse the value in this field in order to determine the actual authorization amount.

#### 17.1.2 Partial Authorization Required Parameters

Field Name	Description
PARTIAL_AUTH	To enable Partial Authorization, set this parameter to “1” and send it with your CCAUTHCAP or CCAUTHORIZE gateway request.

	Set to "0" to disable or do not send the parameter at all.
PARTIAL_AUTH_MIN	<p>This parameter sets the minimum amount that the merchant would accept for partial authorization.  *Example: PARTIAL_AUTH_MIN=25.00  If a minimum of 25.00 is not available on the customer's card, the transaction request will be declined by the Payment Service.</p>

### 17.1.3 Failed Transaction Due to Minimum Amount Response Example

```

<RESPONSE>
<REQUEST_ACTION>CCAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME>DECLINED</TRANS_STATUS_NAME>
<TRANS_VALUE>21.00</TRANS_VALUE>
<TRANS_ID>989898</TRANS_ID>
<CUST_ID>101010</CUST_ID>
<XTL_CUST_ID/>
<MERCH_ACCT_ID>123</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_DETAIL>DEBIT</CARD_DETAIL>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_TYPE>VISA CLASSIC</CARD_TYPE>
<CARD_PREPAID>1</CARD_PREPAID>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>1111</PMT_L4>
<PMT_ID/>
<PMT_ID_XTL>
<API_RESPONSE>0</API_RESPONSE>
<API_ADVICE></API_ADVICE>
<SERVICE_RESPONSE>660</SERVICE_RESPONSE>
<SERVICE_ADVICE>Partial Approval</SERVICE_ADVICE>
<PROCESSOR_RESPONSE>10</PROCESSOR_RESPONSE>
<PROCESSOR_ADVICE>ApprvLesserAmt</PROCESSOR_ADVICE>
<INDUSTRY_RESPONSE>0</INDUSTRY_RESPONSE>
<INDUSTRY_ADVICE></INDUSTRY_ADVICE>
<REF_FIELD/>
<PROC_NAME>Processor Name</PROC_NAME>
<AVS_RESPONSE>U</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>878787</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>21.00</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>32320</PO_LI_PROD_ID_1>
</RESPONSE>

```

### 17.1.4 Partial Authorization Successful Response Example

```

<RESPONSE>
<REQUEST_ACTION>CCAUTHCAP</REQUEST_ACTION>
<TRANS_STATUS_NAME>APPROVED</TRANS_STATUS_NAME>
<TRANS_VALUE>25</TRANS_VALUE>

```

```

<TRANS_VALUE_SETTLED>25</TRANS_VALUE_SETTLED>
<CURR_CODE_ALPHA_SETTLED>USD</CURR_CODE_ALPHA_SETTLED>
<TRANS_EXCH_RATE/>
<TRANS_ID>989898</TRANS_ID>
<CUST_ID>1782177</CUST_ID>
<XTL_CUST_ID/>
<PO_ID>6889345</PO_ID>
<XTL_ORDER_ID>mytest1</XTL_ORDER_ID>
<BATCH_ID>12345</BATCH_ID>
<PROC_NAME>Processor Name</PROC_NAME>
<MERCH_ACCT_ID>123</MERCH_ACCT_ID>
<CARD_BRAND_NAME>Visa</CARD_BRAND_NAME>
<CARD_TYPE>VISA CLASSIC</CARD_TYPE>
<CARD_DETAIL>CREDIT</CARD_DETAIL>
<CARD_CLASS>CONSUMER CREDIT</CARD_CLASS>
<CARD_COUNTRY>CRI</CARD_COUNTRY>
<CARD_PREPAID>1</CARD_PREPAID>
<CARD_BANK/>
<CARD_BALANCE/>
<PMT_L4>1111</PMT_L4>
<PROC_UDF01/>
<PROC_UDF02/>
<PROC_AUTH_RESPONSE>XXYYXX123</PROC_AUTH_RESPONSE>
<PROC_RETRIEVAL_NUM>083073744911846</PROC_RETRIEVAL_NUM>
<PROC_REFERENCE_NUM/>
<AVS_RESPONSE>U</AVS_RESPONSE>
<CVV_RESPONSE>M</CVV_RESPONSE>
<REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
<PO_LI_ID_1>455555</PO_LI_ID_1>
<PO_LI_COUNT_1>1</PO_LI_COUNT_1>
<PO_LI_AMOUNT_1>25</PO_LI_AMOUNT_1>
<PO_LI_PROD_ID_1>77770</PO_LI_PROD_ID_1>
<MBSHP_ID_1/>
</RESPONSE>

```

## 18 External Order ID Uniqueness

### 18.1 UNIQUE\_XTL\_ORDER\_ID Parameter

This parameter allows the merchant to control how the Payment Service handles External Order ID's (XTL\_ORDER\_ID) uniqueness the payment system. This parameter should only be sent with CCAUTHCAP or CCAUTHORIZE requests.

For inquiries regarding this parameter, contact your gateway support representative.

#### 18.1.1 UNIQUE\_XTL\_ORDER\_ID Setting

Value	Description
0	Send this value (or do not send the parameter at all) to disable XTL_ORDER_ID uniqueness enforcement. Default behavior.

	<p>Send this value to enable XTL_ORDER_ID uniqueness enforcement; this will also tell the gateway to decline the request.</p> <p><b>1</b></p> <p>If the XTL_ORDER_ID value already exists in a previously <b>approved</b> order, the gateway is going to decline the transaction request.</p>
	<p><b>2</b></p> <p>Send this value to enable XTL_ORDER_ID uniqueness enforcement; if the XTL_ORDER_ID value already exists in a previously <b>approved</b> order, the gateway is going to return the Service Response of the matching order/transaction.</p>

### 18.1.2 Example Decline Response when UNIQUE\_XTL\_ORDER\_ID parameter is set to "1":

```

<RESPONSE>
  <REQUEST_ACTION>CCAUTHORIZE</REQUEST_ACTION>
  <TRANS_STATUS_NAME/>
  <TRANS_VALUE/>
  <TRANS_ID/>
  <CUST_ID/>
  <XTL_CUST_ID>1234567MMBBR</XTL_CUST_ID>
  <MERCH_ACCT_ID>100</MERCH_ACCT_ID>
  <CARD_BRAND_NAME/>
  <PMT_L4/>
  <PMT_ID/>
  <PMT_ID_XTL/>
  <API_RESPONSE>o</API_RESPONSE>
  <API_ADVICE></API_ADVICE>
  <SERVICE_RESPONSE>685</SERVICE_RESPONSE>
  <SERVICE_ADVICE>Duplicate Order Detected</SERVICE_ADVICE>
  <PROCESSOR_RESPONSE>o</PROCESSOR_RESPONSE>
  <PROCESSOR_ADVICE></PROCESSOR_ADVICE>
  <INDUSTRY_RESPONSE>o</INDUSTRY_RESPONSE>
  <INDUSTRY_ADVICE></INDUSTRY_ADVICE>
  <REF_FIELD>XTL_ORDER_ID</REF_FIELD>
  <PROC_NAME/>
  <AVS_RESPONSE/>
  <CVV_RESPONSE/>
  <REQUEST_API_VERSION>4.6</REQUEST_API_VERSION>
</RESPONSE>
```

## 19 Processor Specific Fields

### 19.1 Processor Specific Fields

These parameters can be used by merchants to send processor-specific data in their gateway transaction request.

Field Name	Used For
<b>PROC_UDF01</b>	eMerchantPay: ThreatMetrix™ eMerchantPay will occasionally require merchants to add ThreatMetrix™ THM_ID to the transaction data. Use the PROC_UDF01 parameter in all initial CCAUTHCAP and CCAUTHORIZE requests for this purpose.
<b>PROC_UDF02</b>	Other processor-specific data. Reserved for NATS Affiliate Software.

For more information regarding these fields, contact your gateway support representative.

## 20 PagoEfectivo

Pago Efectivo is an online payment method in Peru that allows online shoppers to pay through online banking or in cash at authorized centers. Customers are able to check out with a unique payment identification code – **CIP** (Código de identificación de pago) instead of a credit or debit card. Where supported, a QR Code will be generated and the Customer will be able to scan and pay through their bank or copy the QR Code URL and pay through a computer browser.

### 20.1 PagoEfectivo Features

Type of Payment	Offline
Chargeback	No
Consumer Currency	PEN (USD supported)
Consumer Country	Peru
Expiry Period	2.5 hrs
Recurring Payment	No
Partial Payment Support	No
Refunds	No (Check w/Merchant)

### 20.2 PagoEfectivo Request Parameters

Field Name	Description
<b>request_action</b>	<b>PAGSALE</b>
<b>li_prod_id_x</b>	Line Item Count Max value is “10”
<b>li_value_x</b>	Line Item Transaction Amount
<b>userDocumentType</b>	Customer ID Type
<b>userDocumentNumber</b>	Customer Document number

<b>merch_acct_id</b>	Merchant Account ID
<b>cust_fname</b>	Customer's First Name
<b>cust_lname</b>	Customer's Last Name
<b>bill_addr</b>	Customer Billing Street Address
<b>bill_addr_city</b>	Customer's Billing City
<b>bill_addr_state</b>	Customer's Billing State
<b>bill_addr_zip</b>	Customer's Billing Postal/ZIP code
<b>bill_addr_country</b>	Customer's Billing Country
<b>cust_email</b>	Customer's Email Address
<b>request_currency</b>	3-letter Currency Code. Ex: PEN

### 20.3 Additional Response Parameters

Field Name	Description
<b>Proc_redirect_url</b>	The address to which a Customer is redirected once the QR Code has been generated
<b>proc_retrieval_num</b>	Processor tracking number
<b>Proc_reference_num</b>	Contains CIP number
<b>dateExpiry</b>	CIP Expiry Date

### 20.4 PagoEfectivo Request Example

```
https://api.inovioipay.com/payment/pmt_service.cfm
?site_id=328776&li_prod_id_1=183238&cust_fname=customer&cust_lname=Tester&bill_addr_state=CA&request_response_format=JSON&request_api_version=20&cust_email=test@test.com&bill_addr_zip=90401&bill_addr_country=US&request_action=PAGSALE&req_username=testing@example&req_password=Maxell0022.&&bill_addr_city=Los Angeles&bill_addr=1
>MainSt&userDocumentType=NAN&merch_acct_id=157923&userDocumentNumber=123456789123456&li_value_1=15&request_currency=USD&cust_phone=+1 202-918-213212345
{
  "site_id": "328776",
  "li_prod_id_1": "183238",
  "cust_fname": "customer",
  "cust_lname": "Tester",
  "bill_addr_state": "CA",
  "request_response_format": "JSON",
  "request_api_version": "20",
  "cust_email": "test@test.com",
  "bill_addr_zip": "90401",
```

```

"bill_addr_country":"US",
"request_action":"PAGSALE",
"req_username":"testero@example.com",
"req_password":"Maxeloo22.#",
"bill_addr_city":"Los Angeles",
"bill_addr":"1 Main St",
"userDocumentType":"DNI",
"merch_acct_id":"050023",
"userDocumentNumber":"123456789123456",
"li_value_1":"60",
"request_currency":"PEN",
"cust_phone":"1 202-918-213212345"
}

```

## 20.5 PagoEfectivo Response Example

Below is a sample response for a PAGSALE request. The `PROC_REDIRECT_URL` is the URL to display the CIP and QR Code for further payment.

```
{
  "REQUEST_ACTION": "PAGSALE",
  "REQ_ID": "58931022114",
  "TRANS_STATUS_NAME": "PENDING",
  "TRANS_VALUE": 60,
  "CURR_CODE_ALPHA": "PEN",
  "TRANS_VALUE_SETTLED": 60,
  "CURR_CODE_ALPHA_SETTLED": "PEN",
  "TRANS_EXCH_RATE": "",
  "TRANS_ID": 3943973448,
  "CUST_ID": 361302333,
  "XTL_CUST_ID": "",
  "PO_ID": 3887678760,
  "XTL_ORDER_ID": "",
  "BATCH_ID": 7146505,
  "PROC_NAME": "PagoEffectivo",
  "MERCH_ACCT_ID": 157923,
  "CARD_BRAND_NAME": "PagoEfectivo",
  "CARD_TYPE": "",
  "CARD_PREPAID": "",
  "CARD_BANK": "",
  "CARD_BALANCE": "",
  "CARD_DETAIL": "",
  "PMT_L4": "2345",
  "PMT_ID": 487448490,
  "PMT_ID_XTL": "",
  "PMT_AAU_UPDATE_DT": "",
  "PMT_AAU_UPDATE_DESC": "",
  "PROC_UDF01": "CA",
  "PROC_UDF02": "",
  "PROC_AUTH_RESPONSE": "",
  "PROC_RETRIEVAL_NUM": "",
  "PROC_REFERENCE_NUM": "161680789",
  "PROC_BARCODE": "",
  "PROC_REDIRECT_URL": "https://payment.pagoeffectivo.pe/C80oBC64-47Do-4EE8-BF1F-7B36D8206524.html"
}
```

```

    "AVS_RESPONSE": "",  

    "CVV_RESPONSE": "",  

    "REQUEST_API_VERSION": "20",  

    "P3DS_VENDOR": "",  

    "P3DS_RESPONSE": "",  

    "PO_LI_ID_1": "2658110558",  

    "PO_LI_COUNT_1": 1,  

    "PO_LI_AMOUNT_1": "60",  

    "PO_LI_PROD_ID_1": "183238",  

    "MBSHP_ID_1": ""  

}  

}

```

## 21 Apple Pay

---

### 21.1 Overview

This section discusses how to send credit card transactions to the Payment Gateway using Apple Pay. In order to do so, the web pages which host the payment forms must have their domains registered. See the document *Apple Pay Registration Process* for more details.

Additionally, the Processor and Merchant Account being used must also support Apple Pay. For more information on technical usage, please contact your gateway support representative.

### 21.2 Customers who can use Apple Pay

Customers who are using Apple devices (iPhone, iPad, MacBook, etc.) and Apple browser (e.g. Safari) are able to use Apple Pay to make purchases. This section discusses how to set up an Apple Pay purchase on your hosted Payment Page, and then send that as a transaction to the gateway API.

### 21.3 Flow

This is a list of the order in which you can set up the things needed to process transactions with Apple Pay.

- Fetch the Apple Pay configuration data from the gateway
- Update the configuration with the specific payment/order details
- Present the Apple Pay button
- Add the Apple Pay library to the payment page
- Create a JS function to validate your domain
- Create and start the Apple Pay session
- Get the authorized payment information from the customer's browser
- Send the transaction to the gateway

Each item in the list is discussed in the following sections, and some example code is provided. Note that all code provided is for use as an example only, and does not represent code that can be used in your production environment.

**NOTE:** Real cards must be used when testing in production environment. Test cards will not work.

## 21.4 Fetch Apple Pay configuration data from the gateway

You will need to get configuration data from the gateway in order to set up the Apple Pay session when the customer clicks the Apple Pay button. This comes from a request to the following endpoint:

**Endpoint:** <https://api.inoviopay.com/payment/applepay.cfm>

**Parameters:**

Field Name	Description
REQUEST_USERNAME	API CREDENTIAL USERNAME
REQUEST_PASSWORD	API CREDENTIAL PASSWORD
DOMAIN_NAME	DOMAIN NAME WHERE THE PAYMENT PAGE IS HOSTED/SERVED FROM
CLIENT_ID	CLIENT ID
APPLEPAYCONFIG (REQUEST_ACTION)	REQUEST_ACTION TO INSTRUCT THE ENDPOINT TO PROVIDE THE APPLE PAY CONFIGURATION

**Note: All parameters are required**

The response will be a JSON-formatted string. Here is an example of the structure:

```
{
  "APPLEPAY_CLI_CONF_ID": "xx", // internal unique id; do not change
  "REQ_ID": "1111111111", // trace ID, used for troubleshooting
  "CLIENT_ID": "111111", // your gateway api client_id
  "INITIATIVE": "web", // identifies e-commerce transaction
  "DISPLAY_NAME": "xxxxxx", // Short, localized description of the merchant.
  "PARTNER_MERCH_NAME": "xxxxxx", // name of the merchant
  "PARTNER_INTERNAL_MERCH_ID": "xxxxxxxxxxxxxx", // identifies the merchant to apple
  "ENCRYPT_TO": "xxxxxxxxxxxxxxxxxxxx", // merchant ID to Apple
  "DOMAIN_LIST": [ // domain registered in Inovio portal serving this page
    {
      "DOMAIN_NAME": "paymentpagedomain.com",
      "DOMAIN_STATUS": "ACTIVE"
    }
  ],
  "paymentRequest": { //
    "countryCode": "US",
    "currencyCode": "USD",
    "merchantCapabilities": [
      "supports3DS"
    ],
    "supportedNetworks": [
      "visa",
      "masterCard"
    ],
    "requiredBillingContactFields": [

```

```

"postalAddress",
"name",
"phoneticName",
"phone",
"email"
"total": {
  "label": "" // required; a short, localized description of the line item
  "type": "final", // do not change
  "amount": 0 // must be greater than or equal to zero
},
},
"fetchUrl": "https://api.inoviopay.com/apple-pay-services/api/session/create"
}

```

This should be provided on your payment page in the parameter: merchantConfig

### Example:

```

<script>
this.merchantConfig = {
  "APPLEPAY_CLI_CONF_ID": 11,
  "REQ_ID": 1111111111,
  "CLIENT_ID": 1111111,
  "INITIATIVE": "web",
  "DISPLAY_NAME": "xxxxxx",
  "PARTNER_MERCH_NAME": "xxxxxx",
  "PARTNER_INTERNAL_MERCH_ID": "xxxxxxxxxxxx",
  "ENCRYPT_TO": "xxxxxxxxxxxxxxxxxxxx",
  "DOMAIN_LIST": [
    {
      "DOMAIN_NAME": "paymentpagedomain.com",
      "DOMAIN_STATUS": "ACTIVE"
    }
  ],
  "paymentRequest": {
    "countryCode": "US",
    "currencyCode": "USD",
    "merchantCapabilities": [
      "supports3DS"
    ],
    "supportedNetworks": [
      "visa",
      "masterCard"
    ],
    "requiredBillingContactFields": [
      "postalAddress",
      "name",
      "phoneticName",
      "phone",
      "email"
    ],
    "total": {
      "label": ""
      "type": "final",
    }
  }
}

```

```

        "amount": 0
    },
},
"fetchUrl": "https://api.inoviopay.com/apple-pay-services/api/session/create"
}
</script>

```

## 21.5 Update the configuration with the specific payment/order details

Once you have the configuration stored in the merchantConfig parameter on your payment page, you can update the details which are specific to the purchase.

**Example:**

```

<script>
merchantConfig.total.label = "Some purchase description total";
merchantConfig.total.amount = 49.95
</script>

```

## 21.6 Add the Apple Pay library to your payment page

You must add the Apple Pay library to your payment page to make calls to the Apple Pay API. To do that, source the Apple Pay library script source in the page's source html

**Example:**

```
<script src="https://applepay.cdn-apple.com/jsapi/v1/apple-pay-sdk.js"></script>
```

Note that this library will include functionality to validate the user/browser/device as eligible for Apple Pay.

## 21.7 Create a JS function to validate your domain

The Apple Pay JS API library will need to call a function that you define in order to perform a validation on the domain where your payment page is hosted. Note that all of this must load directly from your servers on this domain (no proxies, iframes, etc.)

**Example:**

```

<script>
function validateMerchant(merchantConfig) {
  console.log("validateMerchant: ", merchantConfig.DOMAIN_LIST)
  const data={
    method: "POST",
    headers: {
      "Content-type": "application/json",
      "Accept": "*/*"
    },
    body: JSON.stringify({
      "merchantIdentifier": merchantConfig.PARTNER_INTERNAL_MERCH_ID,
      "displayName": merchantConfig.DISPLAYNAME,
      "initiative": merchantConfig.INITIATIVE,
    })
  }
}

```

```

    "initiativeContext": merchantConfig.DOMAIN_LIST[o].DOMAIN_NAME
  })
};

return fetch(merchantConfig.fetchUrl,data);
}

<script>

```

## 21.8 Present the Apple Pay button

Once the Apple Pay library is added, you can add an HTML element. Please note that it is required to name the element “apple-pay-button”.

**Example:**

```

<style>
.apple-pay-button {
  --apple-pay-button-width: 150px;
  --apple-pay-button-height: 30px;
  --apple-pay-button-border-radius: 3px;
  --apple-pay-button-padding: 0px 0px;
  --apple-pay-button-box-sizing: border-box;
}
</style>

```

The customer will see the Apple Pay button:



## 21.9 Create and start the Apple Pay session

After presenting the button, you must add an action to it which will start up the session where the browser will pop up a dialog for the user to select their card and approve the transaction.

For this, add an onClick event to the Apple Pay button. When referring to “paymentRequest”, we are looking at the JSON object returned from `APPLEPAYCONFIG`. An example of this is included below

### 21.9.1 Handle the Authorized Payment

After the customer has authorized the payment via the Apple Pay overlay, Safari will return an object (Token) with the data needed to transfer to the gateway API to authorize a payment.

For this, you must create the `session.onpaymentauthorized` method to receive the token object. An example of this is included below

### 21.9.2 Handle a Canceled Payment or Error

You should also handle the case where the customer does not authorize the payment or in case an error is thrown. For this, you must create the `session.onCancel`. An example of this is included below. Here is an example of all of these elements described above:

```

<script>
  async function onApplePayClick() {
    // Check for valid ApplePay session
    if(!ApplePaySession){
      return;
    }
    // Get the paymentRequest data from the merchantConfig object
    const paymentRequest = this.merchantConfig.paymentRequest;
    // Validate the merchant.
    // Note that this.merchantConfig is defined in Step 3
    const session = new ApplePaySession(3, paymentRequest);
    session.onvalidatemerchant = async (event) => {
      const merchantSession = await
        this.validateMerchant(this.merchantConfig).then(res=>res.json());
      session.completeMerchantValidation(merchantSession);
    };
    // Handle the authorized payment
    // Note, this must come inside the onApplePayClick() function, and
    // before session.begin
    session.onpaymentauthorized = async (event) => {
      // Define ApplePayPaymentAuthorizationResult
      const result = {
        status: ApplePaySession.STATUS_SUCCESS,
      };
      var applePaymentToken = event.payment;
      var tokenEncoded = btoa(JSON.stringify(applePaymentToken));
      // Now that the payment is authorized in Apple Pay, you can
      // initiate a service call to the gateway passing the
      // applePaymentToken into the gateway API parameter: PMT_WALLET_CRYPTOGRAM
      //
      // Do that here, through your server

      // If the payment is declined at the processor, then you should
      // put an appropriate message in place to the customer
      // result.status = ApplePaySession.STATUS_FAILURE

      // Now we can gracefully complete the Apple Pay browser interaction
      session.completePayment(result);
    };
    // In case the customer cancels or in case of an unexpected error
    // Note, this must come inside the onApplePayClick() function, and
    // before session.begin
    session.oncancel = (event) => {
      // Payment canceled by WebKit
      // error handling
      console.log("Payment canceled by WebKit: "+JSON.stringify(event.error))
    }
    // now the user will interact to approve the purchase
    // Note, this must come at the END of the onApplePayClick()
    session.begin();
  }
</script>

```

## 21.10 Send the authorized payment data from the session to the gateway

Here is an example of a request to authorize payment with Apple Pay:

```
https://api.inoviopay.com/payment/pmt_service.cfm ?request_action=CCAUTHCAP&
li_count_1=1&li_prod_id_1=111&li_value_1=49.95&req_username=GATEWAY
USER&req_password=GATEWAY
PASS&site_id=11111&request_response_format=JSON&request_api_version=4.8&request_cu
rrency=USD&PMT_WALLET=applepay &PMT_WALLET_CRYPTOGRAM=xxxxxxxx
```

## 21.11 Important requirements for Apple Pay authorization requests

You must NOT include any of the following parameters, which will cause the gateway to reject your request:

```
PMT_NUMB
PMT_KEY
PMT_EXPIRY
TOKEN_GUID
PMT_ID
PMT_LAST4
PMT_ID_XTL
PMT_NUMB_COF
REQUEST_INITIATOR
```

Other gateway parameters can be included, but MUST match the values used in the Apple Pay session.

- Parameter `li_value_1` must match the value of `paymentRequest.total` in the `merchantConfig` data.
- Parameter `request_currency` must match the value of `paymentRequest.currencyCode` in the `merchantConfig` data

## 21.12 Rebiling and Credentials on File

The Inovio response will return a `CUST_ID` and `PMT_ID` on a successful authorization using an Apple Pay token (in the same way that it does for normal transactions).

To authorize on the generated ApplePay payment record, the same `CUST_ID` and `PMT_ID` should be passed in on all subsequent rebills or unscheduled merchant initiated card-on-file transaction requests.

# 22 Scheme Level Tokenization

Scheme tokenization is a service provided by card networks (schemes) where a payment account number (PAN) is replaced by a token. The generated network token is stored by the gateway and used for subsequent or recurring transactions. Adding scheme level tokenization service does not require additional system changes for the client.

Merchant accounts can be enrolled for scheme tokenization. For enrollment information, please contact your gateway support representative. Once a merchant account is enrolled, the gateway will automatically provision and store the network tokens for clients' customers' cards for that merchant account. Any time the card account is changed (new number, new expiry), the token will be automatically updated. There is no action needed in the gateway API request to cause this to happen and there are no additional integration changes necessary.

Note that the gateway will relate a scheme token to a card. So if an API transaction request references a card with PMT\_ID, the token will be automatically used.

The gateway response will indicate if a card has been tokenized with the PMT\_WALLET parameter. It will be set to "NETWORK TOKEN" if the card was tokenized. *Note that just because a card is tokenized does not always mean that a transaction will use the token, so it is necessary to look at TRANS\_NTOKEN\_USED (see the next paragraph for details).*

The gateway response will indicate if a token was used instead of a card for a transaction with the TRANS\_NTOKEN\_USED parameter. It will be set to 1 if a scheme token was used or set to 0 if otherwise. (*Note that just because a card was tokenized does not always mean that a transaction used the token, so it is necessary to look at TRANS\_NTOKEN\_USED*)

## 22.1 Additional Response Parameter

Field Name	Description
TRANS_NTOKEN_USED	Used to indicate whether a PAN was used instead of a Network Token when the Token fails

# 23 Service Provider Tokens: Merchant Decryption

## 23.1 Overview

This section discusses how to send credit card transactions to the Payment Gateway using service provider party wallet tokens that has been decrypted by the client. The Inovio Gateway platform supports tokens from 3rd party service providers in accordance with the EMVCo specifications. The integrated application is responsible for decoding the data-string produced by the digital wallet / e-wallet output and sending the appropriate data with the transaction request.

## 23.2 Request Parameters

Below is a list of transaction request parameters that are needed to route third party tokens properly.

Field Name	Description	Additional Notes
<b>WALLET_TOKEN</b>	Field to tell that the following transaction is using a service provider wallet provider.	Field is required. Acceptable values: ApplePay
<b>WALLET_TAVV</b>	Contains the cryptogram returned by the token provider	Field is required.
<b>WALLET_ECI</b>	Contains the ECI response value returned by the Token Provider.	Variable Type: Numeric Max Length: 2

<b>WALLET_TID</b>	Contains the token providers token transaction identifier (TID).	Required to be sent with the transaction if returned by the Token Provider
-------------------	------------------------------------------------------------------	----------------------------------------------------------------------------

Here is an example of a request to authorize payment with Apple Pay:

```
/payment/pmt_service.cfm?request_action=CCAUTHCAP&merch_acct_id=120603&site_id=100103&li_prod_id_1=111205&pmt numb=*****1600%20&pmt_expiry=122029&pmt_key=123&li_value_1=1.00&request_currency=USD&cust_fname=Gilberto&cust_lname=Gilberto&cust_email=test%40test.com&bill_addr_country=US&bill_addr_state=CA&bill_addr_city=Los%20Angeles&bill_addr=4022%20Randolph%20Street&bill_addr_zip=90401&XTL_IP=11.11.11.11&request_api_version=20&request_response_format=JSON&req_username=ellenscoffee%40gmail.com&req_password=Password123456789&argdebug=1&WALLET_ECI=07&WALLET_TID=tyujvcyug5432jksdkfjskdfjk&WALLET_TAVV=%2F1QRHvYAB+88c3MOKPZ3MAACAA%3D&WALLET_TOKEN=applepay
```

Product functions only for applicable processors. For more information on technical usage, please contact your gateway support representative.

## 24 Single Euro Payments Area (SEPA)

The Payment Service supports transactions processed through Single Euro Payments Area (SEPA) Direct Debit.

Single Euro Payments Area (SEPA) Direct Debit allows Europeans with a bank account to do cashless payments online. It also provides a standardized process for payment transactions in the Euro area. For further reading and information, you may visit [SEPA's website](#).

### 24.1 SEPA Direct Debit Payment Process

The following describes the lifecycle of SEPA Direct Debit transactions:

1. Customer initiates a purchase on Merchant's payment page
2. Merchant's site makes a Prepare mandate request (DBTDEBIT {"pay now" sent to trustpay}) with purchase order reference and mandate details to the gateway.
3. Gateway Response with PROC\_REDIRECT\_URL, transaction status is PENDING
4. Customer is redirected to PROC\_REDIRECT\_URL and fills up the SEPA Direct Debit mandate form in the iframe loaded on the merchant's site and confirms the data.
5. Customer is redirected to either proc\_success\_url or proc\_error\_url that were passed into the DBTDEBIT request.
6. The gateway receives a callback on the result and does a postback to merchant with the status change.

### 24.2 Transaction state updates through Postback

To keep merchants updated on the status of their "PENDING" SEPA transactions, the gateway will send real-time Postback to the merchant through the provided notification url.

### 24.3 Authorization Request (SEPA Direct Debit) - DBTAUTHORIZE

When a Mandate has been signed, A *Debit* request is used to make an authorization of payment in the customer's bank account. Merchants must send service request action, "DBTAUTHORIZE" for this type of request. 'Pay Now' is not an option on Authorization Request.

## 24.4 Authorization Capture Request (SEPA Direct Debit) - DBTCAPTURE

When a Mandate has been approved by the customer's bank, a capture request is made to capture the DBTAUTHORIZE 'Pay Now' for a one off sale. Merchants must send service request action, "DBTCAPTURE" for this type of request

## 24.5 Debit Request Parameters (DBTAUTHORIZE)

The table below describes the parameters needed to complete a **DBTAUTHORIZE** transaction request. Merchants can always send additional parameters listed in [Section 4](#) of this specification as needed.

Field Name	Description	Data Type	Requirement
<b>request_action</b>	DBTAUTHORIZE	<a href="#">Service Request Types</a>	Required
<b>req_username</b>	Service Request Username	Alphanumeric	Required
<b>req_password</b>	Service Request Password	Alphanumeric and Special Characters	Required
<b>request_response_format</b>	Service Response Format	Accepted values: "XML", "PIPES" and "JSON"	Optional (Default: XML)
<b>request_api_version</b>	API Version: version 4.8 and above supports European Direct Debit	Numeric	Required
<b>site_id</b>	Merchant's Website ID	Numeric	Required
<b>cust_fname</b>	Cardholder's First Name	Alphanumeric and Special Characters	Required
<b>cust_lname</b>	Cardholder's Last Name	Alphanumeric and Special Characters	Required
<b>cust_email</b>	Cardholder's Email Address	Alphanumeric	Required
<b>li_count_1</b>	Line Item Count Max value is "99".	Numeric	Required
<b>li_prod_id_1</b>	Line Item Product ID 1	Numeric	Required

<b>li_value_1</b>	Line Item Transaction Amount 1	Numeric	Required
<b>bill_addr</b>	Customer Billing Street Address	Alphanumeric	Required
<b>bill_addr_city</b>	Customer Billing City	Alphanumeric	Required
<b>bill_addr_state</b>	Customer Billing State	2-letter State or Territory Code	Some processors may require a valid 2-letter state/territory code
<b>bill_addr_zip</b>	Customer Billing Postal/ZIP code	Alphanumeric	Required
<b>bill_addr_country</b>	Customer Billing Country	2-letter Country Code ISO 3166-1 alpha-2	Required
<b>cust_login</b>	Customer Login or User Name	Alphanumeric and Special Characters	Optional
<b>cust_password</b>	Customer Password Password must be at least 10 characters with 1 number, lower case and upper case letter.	Alphanumeric	Optional
<b>merch_acct_id</b>	Merchant Account ID	Numeric	If null, the system will follow merchant's bank load balancer settings.
<b>request_currency</b>	3-letter Currency Code	All SEPA transactions are in "EUR"	Required
<b>pmt_numb</b>	Bank Account Number, <u>IBAN</u>	Alphanumeric	Required

#### 24.5.1 Debit Request Example

```
/pmt_service.cfm?request_action=DBTAUTHORIZE&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_numb=500105175422315031&li_value_1=0.01&request_currency=EUR&cust_fname=Melinda&cust_lname=Melinda&cust_email=melinda.bozassvai%40gmail.com&bill_addr_country=DE&bill_addr_state=TW&bill_addr_city=Trierweiler&bill_addr=Unter%20den%20Buchen%201&bill_addr_zip=54311&request_api_version=4.8&request_response_format=JSON&req_username=apiuser%40example.com&req_password=Pa33w@rd&DEBIT_TYPE=SEPA'
```

#### 24.5.2 Debit Response Example (Pending)

```
{
  "REQUEST_ACTION": "DBTAUTHORIZE",
  "REQ_ID": "18785674",
```

```

    "TRANS_STATUS_NAME": "PENDING",
    "TRANS_VALUE": 0.01,
    "CURR_CODE_ALPHA": "EUR",
    "TRANS_VALUE_SETTLED": 0.01,
    "CURR_CODE_ALPHA_SETTLED": "EUR",
    "TRANS_EXCH_RATE": "",
    "TRANS_ID": 2001878902,
    "CUST_ID": 7534577,
    "XTL_CUST_ID": "",
    "PO_ID": 18103630,
    "XTL_ORDER_ID": "",
    "BATCH_ID": 140418,
    "PROC_NAME": "TrustPay",
    "MERCH_ACCT_ID": 142376,
    "CARD_BRAND_NAME": "SEPA",
    "CARD_TYPE": "",
    "CARD_PREPAID": "",
    "CARD_BANK": "",
    "CARD_DETAIL": "",
    "CARD_BALANCE": "",
    "PMT_L4": "5031",
    "PMT_ID": 4354240,
    "PMT_ID_XTL": "",
    "PMT_AAU_UPDATE_DT": "",
    "PMT_AAU_UPDATE_DESC": "",
    "PROC_UDF01": "",
    "PROC_UDF02": "",
    "PROC_AUTH_RESPONSE": "",
    "PROC_RETRIEVAL_NUM": "",
    "PROC_REFERENCE_NUM": "",
    "PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",
    "AVS_RESPONSE": "",
    "CVV_RESPONSE": "",
    "CARD_BRAND_TRANSID": "",
    "REQUEST_API_VERSION": "4.8",
    "P3DS_VENDOR": "",
    "P3DS_RESPONSE": "",
    "PO_LI_ID_1": "1901856",
    "PO_LI_COUNT_1": 1,
    "PO_LI_AMOUNT_1": "0.01",
    "PO_LI_PROD_ID_1": "128917",
    "MBSHP_ID_1": "",
    "TRANS_NTOKEN_USED": 0
}

```

#### 24.5.3 Debit Response Example (Success)

```
{
    "REQUEST_ACTION": "DBTCAPTURE",
    "REQ_ID": "18787081",
    "TRANS_STATUS_NAME": "APPROVED",
    "TRANS_VALUE": 0.01,
    "CURR_CODE_ALPHA": "EUR",
    "TRANS_VALUE_SETTLED": 0.01,
```

```

"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001878992,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103673,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"CARD_BRAND_NAME": "SEPA",
"CARD_TYPE": "",
"CARD_PREPAID": "",
"CARD_BANK": "",
"CARD_DETAIL": "",
"CARD_BALANCE": "",
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "",
"PMT_AAU_UPDATE_DT": "",
"PMT_AAU_UPDATE_DESC": "",
"PROC_UDFo1": "",
"PROC_UDFo2": "",
"PROC_AUTH_RESPONSE": "",
"PROC_RETRIEVAL_NUM": "6214013397",
"PROC_REFERENCE_NUM": "",
"PROC_REDIRECT_URL": "",
"AVS_RESPONSE": "",
"CVV_RESPONSE": "",
"CARD_BRAND_TRANSID": "",
"REQUEST_API_VERSION": "4.8",
"P3DS_VENDOR": "",
"P3DS_RESPONSE": "",
"PO_LI_ID_1": "1901896",
"PO_LI_COUNT_1": 1,
"PO_LI_AMOUNT_1": "0.01",
"PO_LI_PROD_ID_1": "128917",
"MBSHP_ID_1": "",
"TRANS_NTOKEN_USED": 0
}

```

## 24.6 Reversals

### 24.6.1 Reversing a mandate (DBTREVERSE)

Merchants or customers can choose to cancel existing mandate whenever they want. To cancel an existing Mandate, Merchants should use DBTREVERSE Request\_Action. This request\_action also stops future rebills and cancels subscriptions.

### 24.6.2 DBTREVERSE transaction Request Example

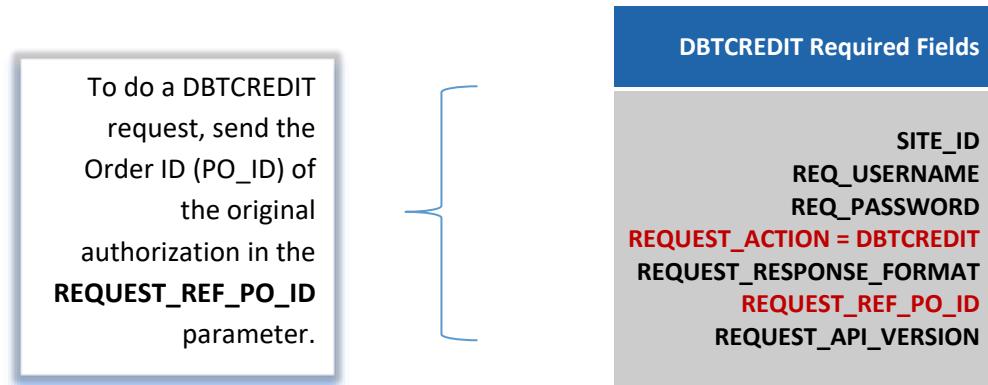
/payment/pmt\_service.cfm?request\_action=DBTREVERSE&merch\_acct\_id=123456&site\_id=456878&li\_prod\_id\_1=14512&pmt\_numbr=200125175423005031&li\_value\_1=0.01&request\_currency=EUR&cust\_fname=CustomerFN&cust\_lname=Masha&cust\_email=msepauser%40gmail.com&bill\_addr\_country=DE&bill\_addr\_state=TW&bill\_addr\_city=Trierweiler&bill\_addr=Unter%20den%20mystreetn%201&bill\_addr\_zip=54311&XTL\_IP=11.11.11.11&request\_api\_version=4.8&request\_response\_format=JSON&req\_username=puser%40exam

```
ple.com&req_password=Testingpassword&argdebug=1&PROC_SUCCESS_URL=testingSuccess&PROC_ER
ROR_URL=testingError&request_ref_po_id=18103694'
```

## 24.7 Refunds or Credits

### 24.7.1 Issuing refunds or credits (DBTCREDIT)

MERCHANTS may request to credit a direct debit order by sending **DBTCREDIT** in the **request\_action** parameter. Note that the request may not be processed successfully by the bank for certain transaction states (e.g. you cannot credit a payment that has not been collected by the bank).



## 24.8 Recurring transactions

SEPA Direct Debit supports recurring payments. Recurring payment Mandate Type authorizes subsequent rebills until canceled. Request Action **DBTDEBIT** is used for initial and subsequent rebills as authorized by Mandate. In summary these are the Recurring transactions scenarios.

- DBTDEBIT & request\_rebill=2: start of rebill subscription
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #1
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #2
- DBTREVERSE: cancel mandate, end of subscription

### 24.8.1 Recurring transaction Request Example (DBTDEBIT & request\_rebill=2)

```
/payment/pmt_service.cfm?request_action=DBTDEBIT&merch_acct_id=142376&site_id=116215&li_prod_id
_1=128917&pmt numb=300105172002111031&li_value_1=0.01&request_currency=EUR&cust_fname=Masha
&cust_lname=cust&cust_email=customer.sepa%40example1.com&bill_addr_country=DE&bill_addr_state=
TW&bill_addr_city=Trierweiler&bill_addr=Unter%20den%20mystreetn%201&bill_addr_zip=54311&XTL_IP
=11.11.11.11&request_api_version=4.8&request_response_format=JSON&req_username=puser%40example.c
om&req_password=Oosii499&request_ref_po_id=18103687&request_rebill=2'
```

### 24.8.2 Recurring transaction Response Example (DBTDEBIT & request\_rebill=2)

```
"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18785674",
"TRANS_STATUS_NAME": "PENDING",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": "2001878902,
```

```

"CUST_ID": 7534577,
"XTL_CUST_ID": "", 
"PO_ID": 18103630,
"XTL_ORDER_ID": "", 
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"CARD_BRAND_NAME": "EUDD",
"CARD_TYPE": "", 
"CARD_PREPAID": "", 
"CARD_BANK": "", 
"CARD_DETAIL": "", 
"CARD_BALANCE": "", 
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "", 
"PMT_AAU_UPDATE_DT": "", 
"PMT_AAU_UPDATE_DESC": "", 
"PROC_UDF01": "", 
"PROC_UDF02": "", 
"PROC_AUTH_RESPONSE": "", 
"PROC_RETRIEVAL_NUM": "", 
"PROC_REFERENCE_NUM": "", 
"PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",
"AVS_RESPONSE": "", 
"CVV_RESPONSE": "", 
"CARD_BRAND_TRANSID": "", 
"REQUEST_API_VERSION": "4.8",
"P3DS_VENDOR": "", 
"P3DS_RESPONSE": "", 
"PO_LI_ID_1": "1901856",
"PO_LI_COUNT_1": 1,
"PO_LI_AMOUNT_1": "0.01",
"PO_LI_PROD_ID_1": "128917",
"MBSHP_ID_1": "", 
"TRANS_NTOKEN_USED": o
}

```

#### 24.8.3 Recurring transaction Request Example (DBTDEBIT & request\_rebill=1)

/payment/pmt\_service.cfm?request\_action=DBTDEBIT&merch\_acct\_id=142376&site\_id=116215&li\_prod\_id\_1=128917&pmt\_numb=200100075433315031&li\_value\_1=0.01&request\_currency=EUR&cust\_fname=Masha da&cust\_lname=Sam&cust\_email=customer%40gmail.com&bill\_addr\_country=DE&bill\_addr\_state=TW&bill\_addr\_city=Trierweiler&bill\_addr=Unter%20den%20mystreet%201&bill\_addr\_zip=54311&request\_api\_version=4.8&request\_response\_format=JSON&req\_username=apiuser%40example.com&req\_password=Pass wordmed@98&request\_rebill=1&CUST\_ID=7534577&PMT\_ID=4354240'

#### Recurring Transaction Fields

CUST\_ID and PMT\_ID are passed into the Gateway. Gateway finds the original mandate, and charges the customer

SITE\_ID  
REQ\_USERNAME  
REQ\_PASSWORD  
REQUEST\_ACTION  
REQUEST\_RESPONSE\_FORMAT  
REQUEST\_API\_VERSION  
**CUST\_ID**  
LI\_PROD\_ID\_1  
LI\_VALUE\_1  
LI\_COUNT\_1  
**PMT\_ID**  
MERCH\_ACCT\_ID  
REQUEST\_CURRENCY

#### 24.8.4 Recurring transaction Response Example (DBTDEBIT & request\_rebill=1)

```
"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18787586",
"TRANS_STATUS_NAME": "APPROVED",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001879023,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103688,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"CARD_BRAND_NAME": "EUDD",
"CARD_TYPE": "",
"CARD_PREPAID": "",
"CARD_BANK": "",
"CARD_DETAIL": "",
"CARD_BALANCE": "",
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "",
"PMT_AAU_UPDATE_DT": "",
"PMT_AAU_UPDATE_DESC": "",
"PROC_UDF01": "",
"PROC_UDF02": "",
"PROC_AUTH_RESPONSE": "",
"PROC_RETRIEVAL_NUM": "5602944007",
"PROC_REFERENCE_NUM": "",
"PROC_REDIRECT_URL": ""
```

```

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901910",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

## 24.9 Additional Parameters

Field Name	Description
PROC_SUCCESS_URL	Used to indicate the URL to which the customer is redirected after successful payment (if PayNow was chosen)
PROC_ERROR_URL	Used to indicate the URL to which the customer is redirected after an error or failed payment (if PayNow was chosen)
PMT_NUMB	Used for IBAN Field (Required)
DEBIT_TYPE	Used to indicate the Debit type used for payment (SEPA, iDEAL, EPS) - Required

## 25 iDEAL Payments

iDEAL is a Dutch (Netherlands-NL) -based payment method that allows customers to complete transactions online using their bank credentials. As such, the customer's bank account has to be in the Netherlands for iDEAL payment availability. When a customer selects to pay with iDEAL, they're presented with a list of banks that support iDEAL. After the customer selects their bank, they're redirected to the bank's website or mobile app to complete the payment. iDEAL is the most commonly used online banking method in the Netherlands, supported by all the major Dutch consumer banks.

### 25.1 Debit Request Parameters (DBTAUTHORIZE)

The table below describes the parameters needed to complete a **DBTAUTHORIZE** transaction request. Merchants can always send additional parameters listed in [Section 4](#) of our API Documentation as needed.

Field Name	Description	Data Type	Requirement
request_action	DBTAUTHORIZE	<a href="#">Service Request Types</a>	Required

<b>req_username</b>	Service Request Username	Alphanumeric	Required
<b>req_password</b>	Service Request Password	Alphanumeric and Special Characters	Required
<b>request_response_format</b>	Service Response Format	Accepted values: "XML", "PIPES" and "JSON"	Optional (Default: XML)
<b>request_api_version</b>	API Version: version 4.8 and above supports European Direct Debit	Numeric	Required
<b>site_id</b>	Merchant's Website ID	Numeric	Required
<b>cust_fname</b>	Cardholder's First Name	Alphanumeric and Special Characters	Required
<b>cust_lname</b>	Cardholder's Last Name	Alphanumeric and Special Characters	Required
<b>cust_email</b>	Cardholder's Email Address	Alphanumeric	Required
<b>li_count_1</b>	Line Item Count Max value is "99".	Numeric	Required
<b>li_prod_id_1</b>	Line Item Product ID 1	Numeric	Required
<b>li_value_1</b>	Line Item Transaction Amount 1	Numeric	Required
<b>bill_addr</b>	Customer Billing Street Address	Alphanumeric	Required
<b>bill_addr_city</b>	Customer Billing City	Alphanumeric	Required
<b>bill_addr_state</b>	Customer Billing State	2-letter State or Territory Code	Some processors may require a valid 2-letter state/territory code
<b>bill_addr_zip</b>	Customer Billing Postal/ZIP code	Alphanumeric	Required

<b>bill_addr_country</b>	Customer Billing Country	2-letter Country Code ISO 3166-1 alpha-2	Required
<b>cust_login</b>	Customer Login or User Name	Alphanumeric and Special Characters	Optional
<b>cust_password</b>	Customer Password Password must be at least 10 characters with 1 number, lower case and upper case letter.	Alphanumeric	Optional
<b>merch_acct_id</b>	Merchant Account ID	Numeric	If null, the system will follow merchant's bank load balancer settings.
<b>request_currency</b>	3-letter Currency Code	All iDEAL transactions are in "EUR"	Required
<b>pmt_numb</b>	Bank Account Number, <a href="#">IBAN</a>	Alphanumeric	Required

### 25.1.1 Debit Request Example

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=DBTAUTHORIZE&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_nu
mb=500105175422315031&li_value_1=0.01&request_currency=EUR&cust_fname=Melinda&cust_lname=Mel
inda&cust_email=melinda.bozassvai%40gmail.com&bill_addr_country=NL&bill_addr_state=TW&bill_addr
_city=NLCITY&bill_addr=Unter%20den%20Buchen%201&bill_addr_zip=54311&request_api_version=4.8&r
equest_response_format=JSON&req_username=apiuser%40example.com&req_password=Pa33w@rd&DEBI
T_TYPE=iDEAL'
```

### 25.1.2 Debit Response Example (Pending)

```
{
  "REQUEST_ACTION": "DBTAUTHORIZE",
  "REQ_ID": "18780074",
  "TRANS_STATUS_NAME": "PENDING",
  "TRANS_VALUE": 0.01,
  "CURR_CODE_ALPHA": "EUR",
  "TRANS_VALUE_SETTLED": 0.01,
  "CURR_CODE_ALPHA_SETTLED": "EUR",
  "TRANS_EXCH_RATE": "",
  "TRANS_ID": "2001878902",
  "CUST_ID": "7534577",
  "XTL_CUST_ID": "",
  "PO_ID": "18103630",
  "XTL_ORDER_ID": "",
  "BATCH_ID": "140418",
  "PROC_NAME": "TrustPay",
  "MERCH_ACCT_ID": "142376",
  "DEBIT_TYPE": "iDEAL",
```

```

"CARD_TYPE": "",  

"CARD_PREPAID": "",  

"CARD_BANK": "",  

"CARD_DETAIL": "",  

"CARD_BALANCE": "",  

"PMT_L4": "5031",  

"PMT_ID": 4354240,  

"PMT_ID_XTL": "",  

"PMT_AAU_UPDATE_DT": "",  

"PMT_AAU_UPDATE_DESC": "",  

"PROC_UDFO1": "",  

"PROC_UDFO2": "",  

"PROC_AUTH_RESPONSE": "",  

"PROC_RETRIEVAL_NUM": "",  

"PROC_REFERENCE_NUM": "",  

"PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",  

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901856",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}
}

```

### 25.1.1 Debit Response Example (Success)

```

"REQUEST_ACTION": "DBTCAPTURE",  

"REQ_ID": "18787081",  

"TRANS_STATUS_NAME": "APPROVED",  

"TRANS_VALUE": 0.01,  

"CURR_CODE_ALPHA": "EUR",  

"TRANS_VALUE_SETTLED": 0.01,  

"CURR_CODE_ALPHA_SETTLED": "EUR",  

"TRANS_EXCH_RATE": "",  

"TRANS_ID": 2001878992,  

"CUST_ID": 7534577,  

"XTL_CUST_ID": "",  

"PO_ID": 18103673,  

"XTL_ORDER_ID": "",  

"BATCH_ID": 140418,  

"PROC_NAME": "TrustPay",  

"MERCH_ACCT_ID": 142376,  

"DEBIT_TYPE": "iDEAL",  

"CARD_TYPE": "",  

"CARD_PREPAID": "",  

"CARD_BANK": "",  

"CARD_DETAIL": ""
}

```

```

"CARD_BALANCE": "",  

"PMT_L4": "5031",  

"PMT_ID": 4354240,  

"PMT_ID_XTL": "",  

"PMT_AAU_UPDATE_DT": "",  

"PMT_AAU_UPDATE_DESC": "",  

"PROC_UDFo1": "",  

"PROC_UDFo2": "",  

"PROC_AUTH_RESPONSE": "",  

"PROC_RETRIEVAL_NUM": "6214013397",  

"PROC_REFERENCE_NUM": "",  

"PROC_REDIRECT_URL": "",  

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901896",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

## 25.2 Reversals

### 25.2.1 Reversing a mandate (DBTREVERSE)

Merchants or customers can choose to cancel existing mandate whenever they want. To cancel an existing Mandate, Merchants should use DBTREVERSE Request\_Action. This request\_action also stops future rebills and cancels subscriptions.

### 25.2.2 DBTREVERSE transaction Request Example

```

https://api.inoviopay.com/payment/pmt_service.cfm  

?request_action=DBTREVERSE&merch_acct_id=123456&site_id=456878&li_prod_id_1=14512&pmt_numbr=400125175423005031&li_value_1=0.01&request_currency=EUR&cust_fname=CustomerFN&cust_lname=Ma sha&cust_email=msepauser%40gmail.com&bill_addr_country=NL&bill_addr_state=TW&bill_addr_city=NL CITY&bill_addr=Unter%20den%20mystreetn%201&bill_addr_zip=54311&XTL_IP=11.11.11.11&request_api_version=4.8&request_response_format=JSON&req_username=puser%40example.com&req_password=Testingpassword&argdebug=1&PROC_SUCCESS_URL=testingSuccess&PROC_ERROR_URL=testingError&request_ref_po_id=18103694

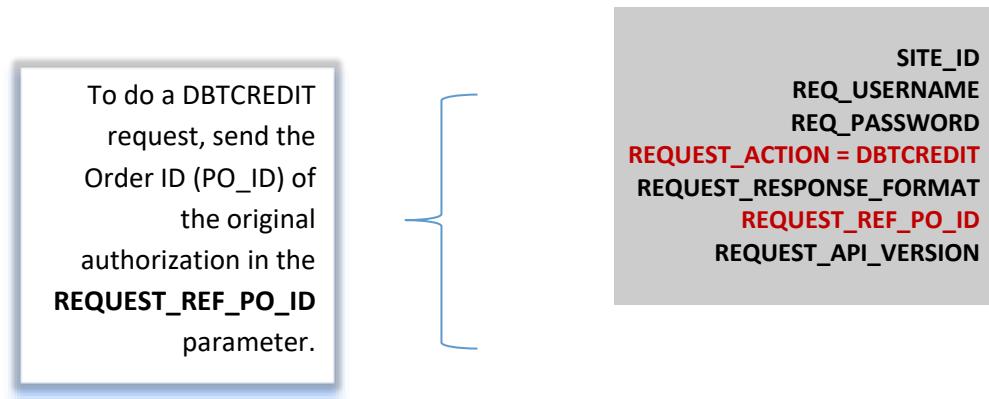
```

## 25.3 Refunds or Credits

### 25.3.1 Issuing refunds or credits (DBTCREDIT)

Merchants may request to credit a direct debit order by sending **DBTCREDIT** in the request\_action parameter. Note that the request may not be processed successfully by the bank for certain transaction states (e.g. you cannot credit a payment that has not been collected by the bank).

**DBTCREDIT Required Fields**



## 25.4 Recurring transactions

iDEAL supports recurring payments through SEPA. Recurring payment Mandate Type authorizes subsequent rebills until canceled. Request Action **DBTDEBIT** is used for initial and subsequent rebills as authorized by Mandate. In summary these are the Recurring transactions scenarios.

- DBTDEBIT & request\_rebill=2: start of rebill subscription
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #1
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #2
- DBTREVERSE: cancel mandate, end of subscription

### 25.4.1 Recurring transaction Request Example (DBTDEBIT & request\_rebill=2)

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=DBTDEBIT&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_numbr=30
0105172002111031&li_value_1=0.01&request_currency=EUR&cust_fname=Masha&cust_lname=cust&cust_email=customer.sepa%40example1.com&bill_addr_country=NL&bill_addr_state=TW&bill_addr_city=NLCITY&bill_addr=Unter%20den%20mystreetn%201&bill_addr_zip=54311&XTL_IP=11.11.11.11&request_api_version=4.8&request_response_format=JSON&req_username=puser%40example.com&req_password=Oosii499&request_ref_po_id=18103687&request_rebill=2'
```

### 25.4.2 Recurring transaction Response Example (DBTDEBIT & request\_rebill=2)

```
"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18785674",
"TRANS_STATUS_NAME": "PENDING",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001878902,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103630,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "iDEAL",
"CARD_TYPE": "",
```

```

"CARD_PREPAID": "",  

"CARD_BANK": "",  

"CARD_DETAIL": "",  

"CARD_BALANCE": "",  

"PMT_L4": "5031",  

"PMT_ID": 4354240,  

"PMT_ID_XTL": "",  

"PMT_AAU_UPDATE_DT": "",  

"PMT_AAU_UPDATE_DESC": "",  

"PROC_UDF01": "",  

"PROC_UDF02": "",  

"PROC_AUTH_RESPONSE": "",  

"PROC_RETRIEVAL_NUM": "",  

"PROC_REFERENCE_NUM": "",  

"PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",  

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901856",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

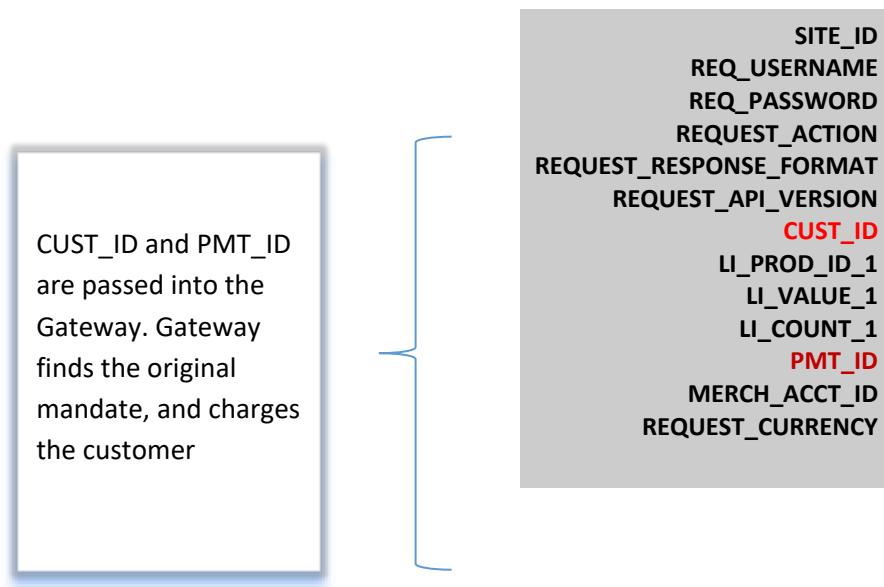
"TRANS_NTOKEN_USED": 0
}

```

#### 25.4.3 Recurring transaction Request Example (DBTDEBIT & request\_rebill=1)

[https://api.inovio.com/payment/pmt\\_service.cfm?request\\_action=DBTDEBIT&merch\\_acct\\_id=142376&site\\_id=116215&li\\_prod\\_id\\_1=128917&pmt\\_numbr=40010007543315031&li\\_value\\_1=0.01&request\\_currency=EUR&cust\\_fname=Mashada&cust\\_lname=Sam&cust\\_email=customer%40gmail.com&bill\\_addr\\_country=NL&bill\\_addr\\_state=TW&bill\\_addr\\_city=NLCITY&bill\\_addr=Unter%20den%20mystreet%201&bill\\_addr\\_zip=54311&request\\_api\\_version=4.8&request\\_response\\_format=JSON&req\\_username=apiuser%40example.com&req\\_password=Passwormed@98&request\\_rebill=1&CUST\\_ID=7534577&PMT\\_ID=4354240](https://api.inovio.com/payment/pmt_service.cfm?request_action=DBTDEBIT&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_numbr=40010007543315031&li_value_1=0.01&request_currency=EUR&cust_fname=Mashada&cust_lname=Sam&cust_email=customer%40gmail.com&bill_addr_country=NL&bill_addr_state=TW&bill_addr_city=NLCITY&bill_addr=Unter%20den%20mystreet%201&bill_addr_zip=54311&request_api_version=4.8&request_response_format=JSON&req_username=apiuser%40example.com&req_password=Passwormed@98&request_rebill=1&CUST_ID=7534577&PMT_ID=4354240)

Recurring Transaction Fields



#### 25.4.4 Recurring transaction Response Example (DBTDEBIT & request\_rebill=1)

```

"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18787586",
"TRANS_STATUS_NAME": "APPROVED",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001879023,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103688,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "iDEAL",
"CARD_TYPE": "",
"CARD_PREPAID": "",
"CARD_BANK": "",
"CARD_DETAIL": "",
"CARD_BALANCE": "",
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "",
"PMT_AAU_UPDATE_DT": "",
"PMT_AAU_UPDATE_DESC": "",
"PROC_UDF01": "",
"PROC_UDF02": "",
"PROC_AUTH_RESPONSE": "",
"PROC_RETRIEVAL_NUM": "5602944007",
"PROC_REFERENCE_NUM": "",
"PROC_REDIRECT_URL": ""

```

```

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901910",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

## 25.5 Additional Parameters

Field Name	Description
PROC_SUCCESS_URL	Used to indicate the URL to which the customer is redirected after successful payment (if PayNow was chosen)
PROC_ERROR_URL	Used to indicate the URL to which the customer is redirected after an error or failed payment (if PayNow was chosen)
PMT_NUMB	Used for IBAN Field (Required)
DEBIT_TYPE	Used to indicate the Debit type used for payment (SEPA, iDEAL, EPS) - Required

## 26 EPS Payments

Electronic Payment Standard (EPS) is an Austrian online banking payment method operated by the Austrian Banking Association. EPS is a guaranteed payment method which lets customers make online payments through their online banking. EPS Payment checkout process works as follows;

1. Customer Bank Account must be located in Austria for EPS payment availability
2. Merchant can be located anywhere within the approved country with local currency presentment
3. Merchant offers EPS as payment method on their checkout page
4. Customer selects EPS from the list of payment methods, then selects their bank.
5. Customer is redirected to the bank website to enter their bank details.
6. Customer confirms the payment and is redirected back to checkout page.

### 26.1 Debit Request Parameters (DBTAUTHORIZE)

The table below describes the parameters needed to complete a **DBTAUTHORIZE** transaction request. Merchants can always send additional parameters listed in [Section 4](#) of our API Documentation as needed.

Field Name	Description	Data Type	Requirement

<b>request_action</b>	DBTAUTHORIZE	<a href="#">Service Request Types</a>	Required
<b>req_username</b>	Service Request Username	Alphanumeric	Required
<b>req_password</b>	Service Request Password	Alphanumeric and Special Characters	Required
<b>request_response_format</b>	Service Response Format	Accepted values: "XML", "PIPES" and "JSON"	Optional (Default: XML)
<b>request_api_version</b>	API Version: version 4.8 and above supports European Direct Debit	Numeric	Required
<b>site_id</b>	Merchant's Website ID	Numeric	Required
<b>cust_fname</b>	Cardholder's First Name	Alphanumeric and Special Characters	Required
<b>cust_lname</b>	Cardholder's Last Name	Alphanumeric and Special Characters	Required
<b>cust_email</b>	Cardholder's Email Address	Alphanumeric	Required
<b>li_count_1</b>	Line Item Count Max value is "99".	Numeric	Required
<b>li_prod_id_1</b>	Line Item Product ID 1	Numeric	Required
<b>li_value_1</b>	Line Item Transaction Amount 1	Numeric	Required
<b>bill_addr</b>	Customer Billing Street Address	Alphanumeric	Required
<b>bill_addr_city</b>	Customer Billing City	Alphanumeric	Required
<b>bill_addr_state</b>	Customer Billing State	2-letter State or Territory Code	Some processors may require a valid 2-letter state/territory code

<b>bill_addr_zip</b>	Customer Billing Postal/ZIP code	Alphanumeric	Required
<b>bill_addr_country</b>	Customer Billing Country	2-letter Country Code ISO 3166-1 alpha-2	Required
<b>cust_login</b>	Customer Login or User Name	Alphanumeric and Special Characters	Optional
<b>cust_password</b>	Customer Password Password must be at least 10 characters with 1 number, lower case and upper case letter.	Alphanumeric	Optional
<b>merch_acct_id</b>	Merchant Account ID	Numeric	If null, the system will follow merchant's bank load balancer settings.
<b>request_currency</b>	3-letter Currency Code	All EPS transactions are in "EUR"	Required
<b>pmt_numb</b>	Bank Account Number, <u>IBAN</u>	Alphanumeric	Required

### 26.1.1 Debit Request Example

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=DBTAUTHORIZE&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_nu
mb=500105175422315031&li_value_1=0.01&request_currency=EUR&cust_fname=Melinda&cust_lname=Mel
inda&cust_email=melinda.bozassvai%40gmail.com&bill_addr_country=AT&bill_addr_state=TW&bill_addr
_city=Trierweiler&bill_addr=Unter%20den%20Buchen%201&bill_addr_zip=54311&request_api_version=4.8
&request_response_format=JSON&req_username=apiuser%40example.com&req_password=Pa33w@rd&DE
BIT_TYPE=EPS'
```

### 26.1.2 Debit Response Example (Pending)

```
{
  "REQUEST_ACTION": "DBTAUTHORIZE",
  "REQ_ID": "18785674",
  "TRANS_STATUS_NAME": "PENDING",
  "TRANS_VALUE": 0.01,
  "CURR_CODE_ALPHA": "EUR",
  "TRANS_VALUE_SETTLED": 0.01,
  "CURR_CODE_ALPHA_SETTLED": "EUR",
  "TRANS_EXCH_RATE": "",
  "TRANS_ID": "2001878902",
  "CUST_ID": "7534577",
  "XTL_CUST_ID": "",
  "PO_ID": "18103630",
  "XTL_ORDER_ID": "",
  "BATCH_ID": "140418",
```

```

"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "EPS",
"CARD_TYPE": "",
"CARD_PREPAID": "",
"CARD_BANK": "",
"CARD_DETAIL": "",
"CARD_BALANCE": "",
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "",
"PMT_AAU_UPDATE_DT": "",
"PMT_AAU_UPDATE_DESC": "",
"PROC_UDF01": "",
"PROC_UDF02": "",
"PROC_AUTH_RESPONSE": "",
"PROC_RETRIEVAL_NUM": "",
"PROC_REFERENCE_NUM": "",
"PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",
"AVS_RESPONSE": "",
"CVV_RESPONSE": "",
"CARD_BRAND_TRANSID": "",
"REQUEST_API_VERSION": "4.8",
"P3DS_VENDOR": "",
"P3DS_RESPONSE": "",
"PO_LI_ID_1": "1901856",
"PO_LI_COUNT_1": 1,
"PO_LI_AMOUNT_1": "0.01",
"PO_LI_PROD_ID_1": "128917",
"MBSHP_ID_1": "",
"TRANS_NTOKEN_USED": 0
}

```

### 26.1.1 Debit Response Example (Success)

```

"REQUEST_ACTION": "DBTCAPTURE",
"REQ_ID": "18787081",
"TRANS_STATUS_NAME": "APPROVED",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001878992,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103673,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "EPS",
"CARD_TYPE": ""

```

```

"CARD_PREPAID": "",  

"CARD_BANK": "",  

"CARD_DETAIL": "",  

"CARD_BALANCE": "",  

"PMT_L4": "5031",  

"PMT_ID": 4354240,  

"PMT_ID_XTL": "",  

"PMT_AAU_UPDATE_DT": "",  

"PMT_AAU_UPDATE_DESC": "",  

"PROC_UDF01": "",  

"PROC_UDF02": "",  

"PROC_AUTH_RESPONSE": "",  

"PROC_RETRIEVAL_NUM": "6214013397",  

"PROC_REFERENCE_NUM": "",  

"PROC_REDIRECT_URL": "",  

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901896",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

## 26.2 Reversals

### 26.2.1 Reversing a mandate (DBTREVERSE)

Merchants or customers can choose to cancel existing mandate whenever they want. To cancel an existing Mandate, Merchants should use DBTREVERSE Request\_Action. This request\_action also stops future rebills and cancels subscriptions.

### 26.2.2 DBTREVERSE transaction Request Example

```

https://api.inoviopay.com/payment/pmt_service.cfm  

?request_action=DBTREVERSE&merch_acct_id=123456&site_id=456878&li_prod_id_1=14512&pmt_numbe  

200125175423005031&li_value_1=0.01&request_currency=EUR&cust_fname=CustomerFN&cust_lname=Ma  

sha&cust_email=msepauser%40gmail.com&bill_addr_country=AT&bill_addr_state=TW&bill_addr_city=Tri  

erweiler&bill_addr=Unter%20den%20mystreetn%201&bill_addr_zip=54311&XTL_IP=11.11.11.11&request_api  

_version=4.8&request_response_format=JSON&req_username=puser%40example.com&req_password=Te  

stingpassword&argdebug=1&PROC_SUCCESS_URL=testingSuccess&PROC_ERROR_URL=testingError&req  

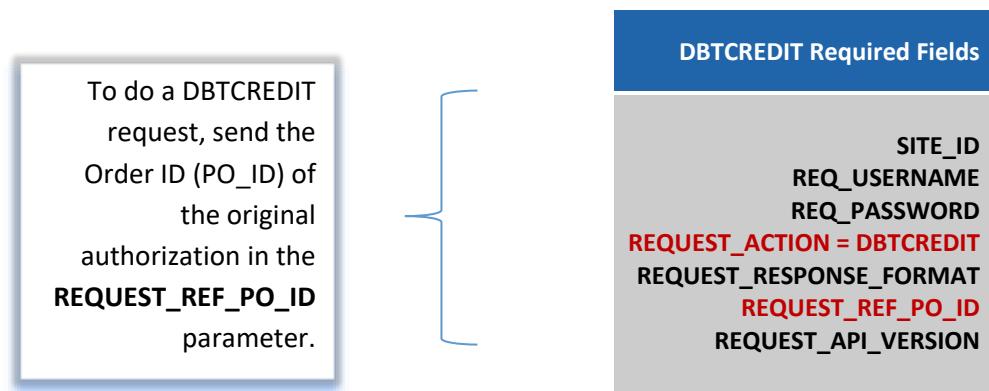
uest_ref_po_id=18103694'

```

## 26.3 Refunds or Credits

### 26.3.1 Issuing refunds or credits (DBTCREDIT)

Merchants may request to credit a direct debit order by sending **DBTCREDIT** in the request\_action parameter. Note that the request may not be processed successfully by the bank for certain transaction states (e.g. you cannot credit a payment that has not been collected by the bank).



## 26.4 Recurring transactions

EPS supports recurring payments through SEPA. Recurring payment Mandate Type authorizes subsequent rebills until canceled. Request Action **DBTDEBIT** is used for initial and subsequent rebills as authorized by Mandate. In summary these are the Recurring transactions scenarios.

- DBTDEBIT & request\_rebill=2: start of rebill subscription
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #1
- DBTDEBIT & request\_rebill=1 & cust\_id & pmt\_id: rebill payment #2
- DBTREVERSE: cancel mandate, end of subscription

### 26.4.1 Recurring transaction Request Example (DBTDEBIT & request\_rebill=2)

```
https://api.inoviopay.com/payment/pmt_service.cfm
?request_action=DBTDEBIT&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt_numbr=30
0105172002111031&li_value_1=0.01&request_currency=EUR&cust_fname=Masha&cust_lname=cust&cust_e
mail=customer.sepa%40example1.com&bill_addr_country=AT&bill_addr_state=TW&bill_addr_city=Trierw
eiler&bill_addr=Unter%20den%20mystreetn%201&bill_addr_zip=54311&XTL_IP=11.11.11.11&request_api_ver
sion=4.8&request_response_format=JSON&req_username=puser%40example.com&req_password=Oosii49
9&request_ref_po_id=18103687&request_rebill=2'
```

### 26.4.2 Recurring transaction Response Example (DBTDEBIT & request\_rebill=2)

```
"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18785674",
"TRANS_STATUS_NAME": "PENDING",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001878902,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103630,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "EPS",
"CARD_TYPE": "",
```

```

"CARD_PREPAID": "",  

"CARD_BANK": "",  

"CARD_DETAIL": "",  

"CARD_BALANCE": "",  

"PMT_L4": "5031",  

"PMT_ID": 4354240,  

"PMT_ID_XTL": "",  

"PMT_AAU_UPDATE_DT": "",  

"PMT_AAU_UPDATE_DESC": "",  

"PROC_UDF01": "",  

"PROC_UDF02": "",  

"PROC_AUTH_RESPONSE": "",  

"PROC_RETRIEVAL_NUM": "",  

"PROC_REFERENCE_NUM": "",  

"PROC_REDIRECT_URL": "https://aapi.trustpay.eu/SepaDirectDebitMandates/input/659b33f4-f5d8-4acb-9dc7-6673e43e57ae",  

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901856",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

#### 26.4.3 Recurring transaction Request Example (DBTDEBIT & request\_rebill=1)

[https://api.inoviopay.com/payment/pmt\\_service.cfm?request\\_action=DBTDEBIT&merch\\_acct\\_id=142376&site\\_id=116215&li\\_prod\\_id\\_1=128917&pmt numb=60010007543315031&li\\_value\\_1=0.01&request\\_currency=EUR&cust\\_fname=Mashada&cust\\_lname=Sam&cust\\_email=customer%40gmail.com&bill\\_addr\\_country=AT&bill\\_addr\\_state=TW&bill\\_addr\\_city=Trierweiler&bill\\_addr=Unter%20den%20mystreet%201&bill\\_addr\\_zip=54311&request\\_api\\_version=4.8&request\\_response\\_format=JSON&req\\_username=apiuser%40example.com&req\\_password=Passwormed@98&request\\_rebill=1&CUST\\_ID=7534577&PMT\\_ID=4354240](https://api.inoviopay.com/payment/pmt_service.cfm?request_action=DBTDEBIT&merch_acct_id=142376&site_id=116215&li_prod_id_1=128917&pmt numb=60010007543315031&li_value_1=0.01&request_currency=EUR&cust_fname=Mashada&cust_lname=Sam&cust_email=customer%40gmail.com&bill_addr_country=AT&bill_addr_state=TW&bill_addr_city=Trierweiler&bill_addr=Unter%20den%20mystreet%201&bill_addr_zip=54311&request_api_version=4.8&request_response_format=JSON&req_username=apiuser%40example.com&req_password=Passwormed@98&request_rebill=1&CUST_ID=7534577&PMT_ID=4354240)

Recurring Transaction Fields

CUST\_ID and PMT\_ID are passed into the Gateway. Gateway finds the original mandate, and charges the customer

SITE\_ID  
REQ\_USERNAME  
REQ\_PASSWORD  
REQUEST\_ACTION  
REQUEST\_RESPONSE\_FORMAT  
REQUEST\_API\_VERSION  
**CUST\_ID**  
LI\_PROD\_ID\_1  
LI\_VALUE\_1  
LI\_COUNT\_1  
**PMT\_ID**  
MERCH\_ACCT\_ID  
REQUEST\_CURRENCY

#### 26.4.4 Recurring transaction Response Example (DBTDEBIT & request\_rebill=1)

```
"REQUEST_ACTION": "DBTDEBIT",
"REQ_ID": "18787586",
"TRANS_STATUS_NAME": "APPROVED",
"TRANS_VALUE": 0.01,
"CURR_CODE_ALPHA": "EUR",
"TRANS_VALUE_SETTLED": 0.01,
"CURR_CODE_ALPHA_SETTLED": "EUR",
"TRANS_EXCH_RATE": "",
"TRANS_ID": 2001879023,
"CUST_ID": 7534577,
"XTL_CUST_ID": "",
"PO_ID": 18103688,
"XTL_ORDER_ID": "",
"BATCH_ID": 140418,
"PROC_NAME": "TrustPay",
"MERCH_ACCT_ID": 142376,
"DEBIT_TYPE": "EPS",
"CARD_TYPE": "",
"CARD_PREPAID": "",
"CARD_BANK": "",
"CARD_DETAIL": "",
"CARD_BALANCE": "",
"PMT_L4": "5031",
"PMT_ID": 4354240,
"PMT_ID_XTL": "",
"PMT_AAU_UPDATE_DT": "",
"PMT_AAU_UPDATE_DESC": "",
"PROC_UDF01": "",
"PROC_UDF02": "",
"PROC_AUTH_RESPONSE": "",
"PROC_RETRIEVAL_NUM": "5602944007",
"PROC_REFERENCE_NUM": "",
"PROC_REDIRECT_URL": ""
```

```

"AVS_RESPONSE": "",  

"CVV_RESPONSE": "",  

"CARD_BRAND_TRANSID": "",  

"REQUEST_API_VERSION": "4.8",  

"P3DS_VENDOR": "",  

"P3DS_RESPONSE": "",  

"PO_LI_ID_1": "1901910",  

"PO_LI_COUNT_1": 1,  

"PO_LI_AMOUNT_1": "0.01",  

"PO_LI_PROD_ID_1": "128917",  

"MBSHP_ID_1": "",  

"TRANS_NTOKEN_USED": 0
}

```

## 26.5 Additional Parameters

Field Name	Description
PROC_SUCCESS_URL	Used to indicate the URL to which the customer is redirected after successful payment (if PayNow was chosen)
PROC_ERROR_URL	Used to indicate the URL to which the customer is redirected after an error or failed payment (if PayNow was chosen)
PMT_NUMB	Used for IBAN Field (Required)
DEBIT_TYPE	Used to indicate the Debit type used for payment (SEPA, iDEAL, EPS) - Required

# 27 Testing

## 27.1 Using the test bank

MERCHANTS may run test authorizations and other gateway actions using the test bank provided in the Payment Service Welcome Email.

If you did not receive your test bank information, contact your gateway support representative.

## 27.2 Basic Gateway Test

Use the table below as a reference for basic gateway testing scenarios and for testing your gateway response parser. The test bank should be used when running the tests.

**NOTE:** Transactions that have been blocked by the gateway due to invalid format or missing data will not be recorded and will not show up in merchant-facing reports on the Portal.

Case	Description	Expected Result
INVALID DATA	Send "1234" in PMT_EXPIRY	API_ADVISE: Invalid Data
REQUIRED FIELD	Send null in SITE_ID	API_ADVISE: Required field REF_FIELD: SITE_ID

<b>INVALID MERCHANT_ACCT_ID</b>	Send "9999" in MERCH_ACCT_FIELD	SERVICE_ADVICE: No merchant account configured
<b>APPROVED AUTHORIZATION</b>	Send "CCAUTHORIZE" in REQUEST_ACTION field along with valid data	TRANS_STATUS_NAME: APPROVED
<b>APPROVED CAPTURE AUTHORIZATION</b>	Send "CCCAPTURE" (follow this <a href="#">section</a> for the required fields).	TRANS_STATUS_NAME: APPROVED
<b>APPROVED SALE (AUTH + CAPTURE)</b>	Send "CCAUTHCAP" in REQUEST_ACTION field along with valid data	TRANS_STATUS_NAME: APPROVED
<b>EXPIRED CARD</b>	Send "5.02" in LI_VALUE_1	SERVICE_ADVICE: Expired Card PROCESSOR_ADVICE: Expired Card
<b>FAILED CVV</b>	Send "5.03" in LI_VALUE_1	SERVICE_ADVICE: Failed CVV PROCESSOR_ADVICE: Failed CVV
<b>FAILED AVS</b>	Send "5.04" in LI_VALUE_1	SERVICE_ADVICE: Failed AVS PROCESSOR_ADVICE: Failed AVS
<b>BANK DECLINED RESPONSE</b>	Send "5.05" in LI_VALUE_1	SERVICE_ADVICE: Declined PROCESSOR_ADVICE: Declined
<b>FRAUD</b>	Send "5.06" in LI_VALUE_1	SERVICE_ADVICE: Fraud PROCESSOR_ADVICE: Fraud
<b>PARTIAL AUTH TEST</b>	Send "PARTIAL_AUTH=1" "PARTIAL_AUTH_MIN=5.00" "LI_VALUE_1=6.60"	TRANS_STATUS_NAME: APPROVED TRANS_VALUE: 5.5
<b>OVER LIMIT</b>	Send "5.07" in LI_VALUE_1	PROCESSOR_ADVICE: Overlimit
<b>AVS NO MATCH</b>	Send "5.08" in LI_VALUE_1	AVS_RESPONSE: N
<b>MISSING REQUIRED FIELD</b>	Send "5.18" in LI_VALUE_1	PROCESSOR_ADVICE: Missing Required Field
<b>DOWNSTREAM PROCESSOR UNAVAILABLE</b>	Send "5.30" in LI_VALUE_1	PROCESSOR_ADVICE: Downstream Processor Unavail
<b>GENERAL DECLINE</b>	Send "6.00" in LI_VALUE_1	PROCESSOR_ADVICE: General Decline
<b>INSUFFICIENT FUNDS</b>	Send "6.35" in LI_VALUE_1	PROCESSOR_ADVICE: Insufficient Funds
<b>PICKUP CARD</b>	Send "6.10" in LI_VALUE_1	PROCESSOR_ADVICE: Pickup Card
<b>STOLEN CARD</b>	Send "6.05" in LI_VALUE_1	PROCESSOR_ADVICE: Stolen Card
<b>EXPIRED CARD</b>	Send "6.24" in LI_VALUE_1	PROCESSOR_ADVICE: Expired Card
<b>INVALID CVV</b>	Send "6.20" in LI_VALUE_1	PROCESSOR_ADVICE: Invalid CVV

<b>CONVERT CURRENCY</b>	Send REQUEST_CURRENCY=EUR and LI_VALUE_1=7.00	TRANS_VALUE: 7 CURR_CODE_ALPHA: EUR TRANS_VALUE_SETTLED: 5.686415 CURR_CODE_ALPHA_SETTLED: USD TRANS_EXCH_RATE: 0.812345
<b>CVV NO MATCH</b>	Send "7.25" in LI_VALUE_1	CVV_RESPONSE:N
<b>R1 REVOCATION OF ALL AUTHORIZATIONS</b>	Send "9.91" in LI_VALUE_1	PROCESSOR_RESPONSE: R1 PROCESSOR_ADVICE: Revocation of All Authorizations1
<b>R3 REVOCATION OF ALL AUTHORIZATIONS</b>	Send "9.93" in LI_VALUE_1	PROCESSOR_RESPONSE: R3 PROCESSOR_ADVICE: Revocation of All Authorizations
<b>R1 REVOCATION OF ALL AUTHORIZATIONS</b>	Send "8.81" in LI_VALUE_1	INDUSTRY_RESPONSE: R1 INDUSTRY_ADVICE: Revocation of All Authorizations
<b>R3 REVOCATION OF ALL AUTHORIZATIONS</b>	Send "8.83" in LI_VALUE_1	INDUSTRY_RESPONSE: R3 INDUSTRY_ADVICE: Revocation of All Authorizations

### 27.3 Test Credit Cards

Use the credit cards below with any address, expiration date and CVV2 data.

Network	Credit Card Number
MASTERCARD	5105105105105100
MASTERCARD	5555555555554444
MASTERCARD	5546989999990033
VISA	4111111111111111
VISA	4907639999990022
AMERICAN EXPRESS	378282246310005
DINERS CLUB	38520000023237
DINERS CLUB	30569309025904
DISCOVER	6011111111111117
DISCOVER	6011000990139424

JCB	3530111333300000
JCB	3566002020360505

## Appendix A: Service Request Types

REQUEST_ACTION	Description
ACHAUTHCAP	Used for authorization and capture requests.
ACHAUTHORIZE	Used for Authorizations without Capture
ACHVERSE	Used for Authorization Capture Reversal
ACHCREDIT	Used for transaction credit requests.
APPLEPAYCONFIG	Instructs the endpoint to provide Apple Pay Configuration
CCAUTHORIZE	Used for sending transaction authorization-only requests.
CCCAPTURE	Used for sending transaction capture previous authorization requests.
CCAUTHCAP	Used for sending transaction “authorization and capture” requests.
CCREVERSE	Used for sending transaction reversal or void requests. Sending this will reverse the <b>original authorization</b> . For reversing CCCAPTURE transactions, merchants should use CCREVERSECAP as the request action.
CCREVERSECAP	Used for sending transaction reversal or void requests against a “CCCAPTURE” transaction.
CCCREDIT	Used for issuing transaction returns or credits.
CCRDR	Used for RDR Dispute Processing
CCRDRDELETE	Used for RDR Dispute Processing to show a case that has been removed by the customer/issuer
CCTC40	Used for TC40 Alerts Processing
CCSTATUS	Used for checking the status of a previous transaction or order.
CCTRANSUPDATE	Used to add receipts on transaction that was previously run (approved or declined)
DBTAUTHORIZE	Used for preparing Mandate without charging
DBTCAPTURE	Used to charge the Mandate for the submitted amount

<b>DBTCREDIT</b>	Used for SEPA Direct Debit Refund/Credit request
<b>DBTDEBIT</b>	Used for SEPA Direct Debit Pay Immediately 'Pay Now'
<b>DBTREVERSE</b>	Used for canceling existing mandate and end subscription
<b>TESTGW</b>	Used for testing gateway availability.
<b>TESTAUTH</b>	Used for testing basic authentication.
<b>SUB_CANCEL</b>	Used for requesting cancelation of an active membership record.
<b>SUB_UPDATE</b>	Used for updating the Product ID of an existing membership record.
<b>ACHAUTHCAP</b>	Request a transaction for Electronic Funds Transfer
<b>ACHAUTHORIZE</b>	Authorize/Validate Check without funds transfer
<b>ACHREVERSE</b>	Reverse an ACHAUTHCAP
<b>BOLETOAUTHCAP</b>	Used for Brazilian Boleto Payment type
<b>PIXSALE</b>	Used for Brazilian Pix Payment type
<b>PAGSALE</b>	Used for Peru's PagoEfectivo Payment type

## Appendix B: Transaction Status

TRANS_STATUS_NAME	Description
<b>APPROVED</b>	Transaction has been approved.
<b>DECLINED</b>	Transaction has been declined.
<b>PENDING</b>	Transaction is in pending status (expected on 3-D Secure, and preauthorization of online check transactions (i.e. Boleto, ACH, Pix etc.)).
<b>RUNNING</b>	Transaction processing is not completed or is waiting completion.
<b>FAILED</b>	Transaction did not finish payment completion (used in European Direct Debit transactions)

## Appendix C: API Response Codes

API_RESPONSE	Description
<b>100</b>	Invalid login information (throttle)
<b>101</b>	Invalid login information
<b>102</b>	User not active
<b>103</b>	Invalid site
<b>104</b>	Invalid service
<b>105</b>	Invalid service action
<b>106</b>	Invalid service object
<b>110</b>	Required field
<b>111</b>	Invalid length
<b>112</b>	Not numeric
<b>113</b>	Invalid Data
<b>115</b>	Customer not found
<b>116</b>	User MUST change password
<b>118</b>	New password must not match the previous 5 passwords
<b>119</b>	request_ref_po_id and request_po_li_id mismatch
<b>120</b>	System Error
<b>125</b>	Duplicate Login
<b>130</b>	Same Product ID found on different line items.
<b>135</b>	Duplicate Company Name
<b>136</b>	Duplicate Site Name
<b>150</b>	Product Not Found
<b>152</b>	Product Type Not Found

	<b>153</b>	Duplicate XTL product id
	<b>155</b>	Selected currency not configured
	<b>160</b>	Invalid product amount
	<b>165</b>	Currency not supported
	<b>170</b>	Duplicate product amount and currency
	<b>175</b>	Language not supported
	<b>176</b>	Duplicate product description and language
	<b>180</b>	Invalid transaction limit type
	<b>181</b>	Invalid limit type
	<b>183</b>	Payment Type is required
	<b>205</b>	No Permissions on requested object
	<b>210</b>	Merchant Account not found
	<b>211</b>	Currency not found
	<b>215</b>	Invalid Card Brand
	<b>260</b>	Invalid Process Ref ID
	<b>261</b>	Invalid Process Type
	<b>410</b>	Field not supported with wallet payment
	<b>411</b>	REQUEST_CURRENCY mismatch with Cryptogram

## Appendix D: Service Response Codes

SERVICE_RESPONSE	Description
<b>100</b>	User Authorized
<b>101</b>	Service Available
<b>102</b>	Membership Updated

<b>150</b>	Product Not Found
<b>152</b>	Product Type Not Found
<b>155</b>	Selected currency not configured
<b>157</b>	MID has RDR Status OFF
<b>190</b>	Invalid Product Configuration
<b>192</b>	Product Not Active
<b>200</b>	CVV required by processor
<b>201</b>	Country required by processor
<b>202</b>	DOB required by processor
<b>203</b>	SSN required by processor
<b>204</b>	Address required by processor
<b>205</b>	City required by processor
<b>206</b>	State required by processor
<b>207</b>	Postal Code required by processor
<b>208</b>	Phone required by processor
<b>209</b>	IP required by processor
<b>210</b>	CPF required by processor
<b>211</b>	Email required by processor
<b>212</b>	FName required by processor
<b>213</b>	LName required by processor
<b>215</b>	Activity limit exceeded
<b>216</b>	Invalid amount
<b>217</b>	No such issuer
<b>218</b>	Wrong PIN entered

	<b>219</b>	R0: Stop recurring payments
	<b>220</b>	R1: Stop recurring payments
	<b>221</b>	System malfunction
	<b>500</b>	No merchant account configured
	<b>501</b>	Customer not found
	<b>502</b>	Transaction error
	<b>503</b>	Service Unavailable
	<b>505</b>	Order adjusted to zero
	<b>506</b>	Capture amount exceeds order value
	<b>507</b>	Order fully captured
	<b>510</b>	Order already reversed
	<b>511</b>	Order already charged back
	<b>512</b>	Order not found
	<b>515</b>	Order fully credited
	<b>516</b>	Credit amount exceeds order value
	<b>518</b>	Missing required field
	<b>520</b>	Unsupported Currency
	<b>522</b>	Unsupported card brand
	<b>525</b>	Batch Closed: Please credit
	<b>526</b>	ApplePay is not supported on this merch_acct_id
	<b>527</b>	No ApplePay merch_acct_id configured
	<b>528</b>	ApplePay MCC Restricted
	<b>530</b>	Downstream Processor Unavailable
	<b>536</b>	Order not settled: Please reverse

	<b>540</b>	Maximum Auth Limit Exceeded
	<b>555</b>	Call Center
	<b>560</b>	Invalid Service Action
	<b>564</b>	Invalid Terminal
	<b>565</b>	Invalid Amount
	<b>570</b>	Invalid Card Type
	<b>580</b>	Unsupported Request
	<b>600</b>	Declined
	<b>601</b>	Scrub Decline
	<b>603</b>	Fraud
	<b>605</b>	Stolen Card
	<b>610</b>	Pickup Card
	<b>615</b>	Lost Card
	<b>620</b>	Invalid CVV
	<b>621</b>	Failed CVV
	<b>622</b>	Invalid AVS
	<b>623</b>	Failed AVS
	<b>624</b>	Expired Card
	<b>625</b>	Excessive Use
	<b>630</b>	Invalid Card Number
	<b>635</b>	Insufficient Funds
	<b>640</b>	Retry
	<b>650</b>	Do Not Honor
	<b>660</b>	Partial Approval

	<b>670</b>	Additional Authentication Required
	<b>675</b>	Invalid Card Number, failed Mod 10 validation
	<b>680</b>	Duplicate Transaction Detected
	<b>685</b>	Duplicate Order Detected
	<b>690</b>	Active Membership Exists
	<b>692</b>	Invalid Rebill Product
	<b>695</b>	Site Username Unavailable
	<b>697</b>	Membership Not Active
	<b>698</b>	Membership Not Found
	<b>699</b>	Membership Not Set for Rebill
	<b>700</b>	Scrub Decline
	<b>706</b>	Failed Age Validation
	<b>707</b>	Invalid CPF

## Appendix E: Address Verification Service Codes

Code	Description	Credit Card Network
A	Street address matches, but 5-digit and 9-digit postal code do not match.	Standard domestic (US)
B	Street address matches, but postal code not verified.	Standard international
C	Street address and postal code do not match.	Standard international
D	Street address and postal code match. Code "M" is equivalent.	Standard international
E	AVS data is invalid or AVS is not allowed for this card type.	Standard domestic (US)

<b>F</b>	Card member's name does not match, but billing postal code matches.	American Express only
<b>G</b>	Non-U.S. issuing bank does not support AVS.	Standard international
<b>H</b>	Card member's name does not match. Street address and postal code match.	American Express only
<b>I</b>	Address not verified.	Standard international
<b>J</b>	Card member's name, billing address, and postal code match.	American Express only
<b>K</b>	Card member's name matches but billing address and billing postal code do not match.	American Express only
<b>L</b>	Card member's name and billing postal code match, but billing address does not match.	American Express only
<b>M</b>	Street address and postal code match. Code "D" is equivalent.	Standard international
<b>N</b>	Street address and postal code do not match.	Standard domestic (US)
<b>O</b>	Card member's name and billing address match, but billing postal code does not match.	American Express only
<b>P</b>	Postal code matches, but street address not verified.	Standard international
<b>Q</b>	Card member's name, billing address, and postal code match.	American Express only
<b>R</b>	System unavailable.	Standard domestic (US)
<b>S</b>	Bank does not support AVS.	Standard domestic (US)
<b>T</b>	Card member's name does not match, but street address matches.	American Express only
<b>U</b>	Address information unavailable. Returned if the U.S. bank does not support non-U.S. AVS or if the AVS in a U.S. bank is not functioning properly.	Standard domestic (US)

V	Card member's name, billing address, and billing postal code match.	American Express only
W	Street address does not match, but 9-digit postal code matches.	Standard domestic (US)
X	Street address and 9-digit postal code match.	Standard domestic (US)
Y	Street address and 5-digit postal code match.	Standard domestic (US)
Z	Street address does not match, but 5-digit postal code matches.	Standard domestic (US)

## Appendix F: CVV Response Codes

Code	Description
M	Match
N	No Match
P	Not Processed
S	Not Supported
U	Service Not Available
X	No CVC/CVV/CVV2/CID Response Data Available
(empty)	No CVC/CVV/CVV2/CID Response Data Available

## Appendix G: Negative Option Billing- MCC 5968

Effective April 12, 2019, all MasterCard acquirers and processors are required to ensure their merchants which use Negative Option Trial Continuity billing models (MCC 5968) providing physical products are compliant with revised standards published in AN 2202.

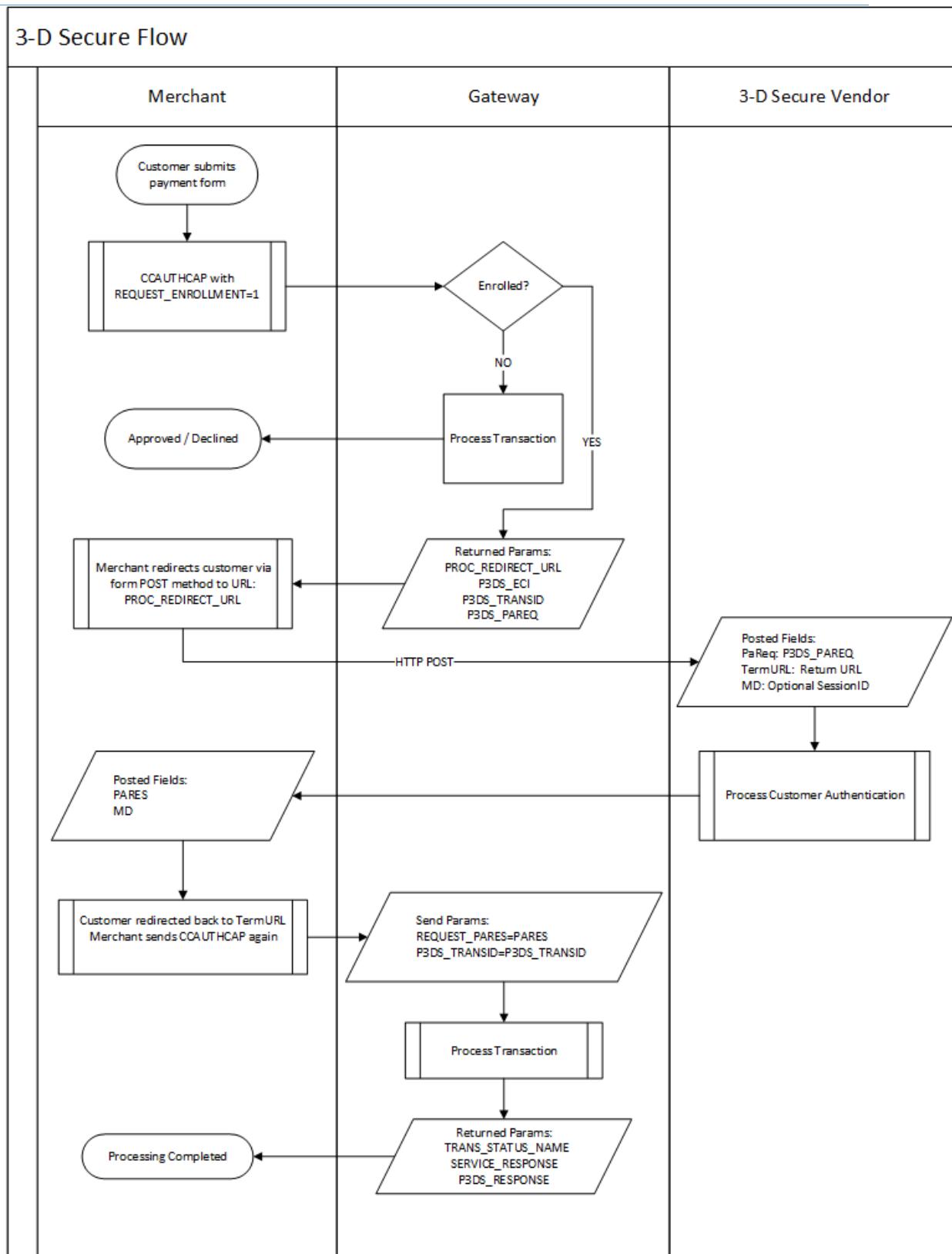
The gateway provides functionality and accepts data in support of these standards, and this appendix provides guidance on how the gateway API should be used for this billing model and which parameters are relevant.

Please note that this functionality may only be in place for some ISOs. Merchants should contact their merchant account providers or gateway support representative for additional information on whether this section is relevant to them.

Field Name	Description
<b>REQUEST_ACTION</b>	<p>Use the “CCTRANSUPDATE” value to add a receipt to a transaction that was previously run.</p> <p>Because MasterCard’s guidelines state that a customer receipt must be sent in both the case of an authorized rebill OR a declined rebill, it is not possible to know the exact content of the receipt at the time of the authorization request (since a receipt for an authorization will be much different from a receipt for a decline).</p> <p>Therefore, this action is necessary to update the order with the appropriate customer receipt AFTER the authorization request has been completed.</p>
<b>MBSHP_ID_XTL</b>	<p>Customer Membership ID.</p> <p>Use this parameter to identify a specific subscription to a customer.</p>
<b>TRANS_REBILL_TYPE</b>	<p>Use this parameter with CCAUTHCAP and CCAUTHORIZE to indicate one of the following conditions:</p> <p>“NONE – send this value if the authorization is not related to a subscription. Valid CVV will be required. (note that a “Card on File” transaction without CVV is not supported for MC 5968 Trial accounts and will be declined)</p> <p>“TRIAL” – send this value if the authorization is for the start of a trial-based subscription. A new, unique MBSHP_ID_XTL value and a valid CVV will be required.</p> <p>“INITIAL” – Send this value if the authorization is for the start of a NON-trial-based subscription. A new, unique MBSHP_ID_XTL value and a valid CVV will be required.</p> <p>“REBILL” – send this value if the authorization is for the continuation of an existing subscription. An existing MBSHP_ID_XTL is required. If the rebill follows a trial, then TRANS_TRIAL_REBILL_CUSTOMER_CONSENT is required. It is required that any previous rebills MUST have been updated with a TRANS_CUSTOMER_RECEIPT.</p> <p>First transaction after trial with subsequent transactions possible as rebill/membership service , REBILL-Rebill transaction with prior transaction of Initial type previously captured)</p>
<b>TRANS_CUSTOMER_RECEIPT</b>	<p>The entire URL encoded contents of the receipt which merchant transmitted to the customer. Use REQUEST_ACTION=CCTRANSUPDATE to record this parameter after the authorization attempt is completed. This will be used to store the receipt for a successful authorization OR the receipt for</p>

	an unsuccessful authorization. This is required prior to the next subsequent rebill or else the attempt for the next rebill will be rejected.
<b>TRANS_TRIAL_REBILL_CUSTOMER_CONSENT</b>	This is to store Customer Consent for subsequent rebill after a trial. This should be sent with the authorization request for that rebill and is required.
<b>TPPE_ID</b>	ID of Third-Party Processing Entity (e.g. CRM, eCommerce Platform, etc.) which is transmitting the request to the gateway on the merchant's behalf. Required when merchant is not transmitting cardholder data directly from their own systems.

## Appendix H: 3-D Secure Flow



## Appendix I: Automatic Account Updater (AAU)

In the event there is a change on Credit Card on record that might impact future transactions, AAU-enrolled Merchants will not experience approval changes caused by Credit card number changes. Changes like a new expiration date, account status change or a new account number will be automatically updated before there's any effect on future transactions.

When a Card is updated with a new expiration date, the new expiration date, Update date and description will be reported. Occasionally, since Credit Card numbers are encrypted/tokenized, a new Credit Card will only be reported by an Update Date and Description. Where available, the Last 4 digits will be updated but the PMT\_ID remains the same. The table below outlines different scenarios and their respective examples. For more information regarding this service, contact our Client Services.

Update Response Action	Card Type	Description/ Element Name
Closed Account Advise	VISA	(PMT_AAU_UPDATE_DESC: "Closed Account Advise", PMT_AAU_UPDATE_DT: "2019-07-24")
Expiration Date Change Message	VISA	(PMT_AAU_UPDATE_DESC: Expiration Date Change Message", PMT_EXPIRY - "012022", PMT_AAU_UPDATE_DT: "2019-07-24")
Account Number Change Message	VISA.	(PMT_AAU_UPDATE_DESC: "Account Number Change Message", PMT_EXPIRY: "012022", PMT_L4:"6183", PMT_AAU_UPDATE_DT: "2019-07-24")
Match Made, Account Closed	MasterCard	(PMT_AAU_UPDATE_DESC: "Match Made, Account Closed", PMT_AAU_UPDATE_DT: "2019-07-24")
Match Made, Expiration Date Changed	MasterCard	(PMT_AAU_UPDATE_DESC: " Match Made, Expiration Date Changed", PMT_EXPIRY: "012022", PMT_L4:"1860", PMT_AAU_UPDATE_DT: "2019-07-24")
Match Made, Update Data Provided	MasterCard	(PMT_AAU_UPDATE_DESC: " Match Made, Update Data Provided", PMT_EXPIRY:" 012022", PMT_L4:"1920", PMT_AAU_UPDATE_DT: 2019-07-24")

## Appendix J: Visa Trial Processing

Effective April 18, 2020, Visa has updated rules for merchants that offer free trials or introductory offers as part of an ongoing subscription service.

The gateway provides functionality and accepts data in support of these rules, and this appendix provides guidance on how the gateway API should be used for this billing model and which parameters are relevant.

Please note that this functionality may not function properly for all processors (e.g. dynamic descriptor support may not allow trial purchases to be identified). Merchants should contact their merchant account providers or gateway support representative with any questions.

Field Name	Requirement	Description
<b>MBSHP_ID_XTL</b>	OPTIONAL	<p>Customer Membership ID. Use this parameter to identify a specific subscription to a customer.</p>
<b>TRANS_REBILL_TYPE</b>	OPTIONAL (Unless Transaction Type is “TRIAL”)	<p>Use this parameter with CCAUTHCAP and CCAUTHORIZE to indicate one of the following conditions:</p> <p>“NONE” – send this value if the authorization is not related to a subscription. Valid CVV will be required</p> <p>“TRIAL” – send this value if the authorization is for the start of a trial-based subscription. A new, unique MBSHP_ID_XTL value and a valid CVV will be required.</p> <p>“INITIAL” – Send this value if the authorization is for the start of a NON-trial-based subscription. A new, unique MBSHP_ID_XTL value and a valid CVV will be required.</p> <p>“REBILL” – send this value if the authorization is for the continuation of an existing subscription. An existing MBSHP_ID_XTL is required.</p>
<b>TRANS_CUSTOMER_RECEIPT</b>	OPTIONAL	The entire URL encoded contents of the receipt which merchant transmitted to the customer.

## Appendix K: Card on File Matrix

A Card on File (*Credential on File, COF or Stored Credential*) is payment information like an account number or payment token that is stored by a merchant or its agent to process future purchases for a cardholder. Future purchases may include Installments, subscription rebills and occasional one time purchases initiated by the Merchant (MIT) or the Client (CIT).

The table below outlines the new parameters mandated and added to our gateway for Card on File transactions.

New Parameter	Values	Description
<b>REQUEST_INITATOR</b>	CIT (C) and MIT (M)	Used to identify transaction origination

<b>REQUEST_INSTALLMENT</b>	0 (not used), 1 (used)	Used to denote if the rebill is an installment with start and end date.
<b>PMT_NUMB_COF</b>	0 (not used), 1 (used)	Used to denote use of payment number stored by Merchant
<b>REQUEST_REBILL</b>	1 (yes for rebill), 2(Start Subscription), not used	Used to denote if the transaction is subscription card storage, a rebill or not used at all