BOI Payment Acceptance
A division of EVO

BOIPA gateway

AUTH/PURCHASE/VERIFY (Direct API Integration)
Version 5.8 3DS V2.x, September 23, 2020

Contents

Document Purpose

The purpose of this document is to describe the AUTH/PURCHASE/VERIFY (Direct API Integration) API Operation to enable merchant developers to integrate their webpages with the BOIPA gateway. Refer to the *BOIPA gateway – 0 – Overview* document for how this API Operation is used in the merchant processes.

The AUTH/PURCHASE/VERIFY (Direct API Integration) API Operation allows the merchant using the Direct API Integration Method to send customer authorise and purchase payment card transactions, or payment card details for verification through the BOIPA gateway.

PSD2, SCA & 3DSV2.x Considerations

Changes to the Payment Services Directive (PSD2), embodied in Strong Customer Authentication (SCA) and the updated Third Domain Secure Version 2.1 & 2.2 (3DSV2.x), have added to the data required by Card Schemes. Issuers, Acquirers and Payment Service Providers (PSPs), including the BOIPA gateway have been upgrading their systems to take account of the new data requirements.

The one overriding change to card payment transactions that should be understood by all merchants is that all card payment transactions will now be processed through 3DS Authentication. Therefore, merchants will not be able to switch off Authentication processing, except under exceptional circumstances agreed with the Acquirer.

The new data requirements are primarily focussed on providing improved security to the cardholder in the prevention of fraud and card misuse.

Therefore, additional data parameters are provided for in the Session Token Request (section 1.1). In addition, the requirements for some existing parameters have changed in that some parameters that were optional are now mandatory for 3DSV2.x processing. The failure to provide these parameters will automatically channel the transaction through the current 3DS Version 1.0 authentication method.

At the time of writing, it is not known when 3DS Version 1.0 will be retired. Although the Card Schemes have stated that it will be retired, they have not yet provided and firm indication of when this may happen.

To assist the merchant's business analysis of the Session Token Request (section 1.1), the parameters have been grouped with heading rows to provide an overview of those parameters.

To assist the development of integration the new and changed parameters have been shaded in green.

Note: as much information should be supplied as is available to the merchant to assist the Issuer with providing a Frictionless Flow, i.e. to authenticate a payment card transaction without the need to challenge the cardholder.

Change Log

Version	Date	Author	Description of Change
5.0	21/04/20	Vaughan Morgan-Jones	Section 2.1.2: SCA/3DS V2.x parameters added
5.1	25/05/20	Vaughan Morgan-Jones	Section 2.1.2: • Changed *cardOnFileReason* to be completed by *all* merchants Required for authentication purposes • Added *cardOnFileMaxPayments* All Sections: Examples removed – to be reworked in future version App F: Reworded explanation for *merchantAuthData*
5.2	12/06/20	Vaughan Morgan-Jones	Section 2.1.2: Changes made to External Authentication parameters to provide enumerated values for protocolVersion and require the data for all MPIs, not just Redsys.
5.3	07/07/20	Vaughan Morgan-Jones	Section 2.1.2: • *sdkAppInfo*: Added to support App Flow • *cardOnFileInitialTransactionId*: Added note
5.4	24/07/20	Vaughan Morgan-Jones	Section 1.1: Added *mmrpOrderNumber* Corrupted document rebuilt
5.5	03/09/20	Vaughan Morgan-Jones	Section 1.1: Removed values 07 & 08 from merchantChallengeInd
5.6	15/09/20	Vaughan Morgan-Jones	Section 1.1: • Changed rules for customer address data • Added "**Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information." to Customer Address & IP Address parameters • Removed all references to Quick Sales, which are not offered in the EU
5.7	17/09/20	Vaughan Morgan-Jones	Section 1.1.2: Changed rpDueDate = 0 when rpFrequency = 20 & 23
5.8	23/09/20	Vaughan Morgan-Jones	Section 1.1.2: Changed Requirement for *merchantNotificationUrl* to 'N'

1 Session Token API Operation

1.1 Session Token Request

1.1.1 Format

POST Request to Session Token Request URL (see Section 3 of the BOIPA gateway – 0 – Overview document)

1.1.2 Definition

Parameter	Data Type	Required	Description
Security Data Mandatory to identify the merchant in the BOIPA gateway			
merchantId	Integer (18)	Y	The merchant's account identifier for the merchant in the BOIPA gateway provided at on-boarding
password	String (64)	Y	The merchant's account password for API Operations in the BOIPA gateway provided at on-boarding
Transaction Data The Transaction Data defines the type of transaction the merchant is requesting the BOIPA gateway to perform, how the transaction result will be managed, and complimentary data required by the Authentication and Authorisation Processes. The transaction result can be the Authentication or Authorisation response.			
action	String (enum)	Y	Must be "AUTH", "PURCHASE" or "VERIFY" For Recurring Payments, i.e. where *rpPlanType* > 0 "AUTH" or "PURCHASE" can be used for any *rpPlanType* "VERIFY" can only be used for *rpPlanType* = 2 (Direct Debit) or 4 (Pay Per Use) In the case of free-trial period for Pay Per Use Plan Types, or deferred first payment for Direct Debits
firstTimeTransaction	Boolean	N	A flag to indicate if the transaction is the customer's first. For some merchant configurations, this forces 3D Secure processing. **Note**: if a *customerId* value is not provided, first-time transaction is assumed
timestamp	Integer (13)	Y	Milliseconds since 1970-01-01 00:00:00

Parameter	Data Type	Required	Description
merchantChallengeInd	String (enum)	N	Merchant Challenge Indicator: Indicates whether the merchant is requesting a challenge for this transaction, for local/regional mandates or other reasons. **It is highly recommended that this parameter is supplied, even if there is no preference ('01')** For example: for Payment Authorisations (*action* = 'AUTH' or 'PURCHASE'), a merchant may have concerns about the transaction, and request a challenge. Some BOIPA gateway rules will override a merchant's requirement not to challenge the cardholder: 1. A challenge will always be requested for Non-Payment Authorisations (*action* = 'VERIFY') 2. A challenge will always be requested for *cardOnFileType* = 'First' 3. A challenge may be requested for if the Acquirer's Transaction Risk Analysis has been performed and requires a challenge request **Values accepted**: 01 = No preference – **Default if parameter not provided** 02 = No challenge requested 03 = Challenge requested (merchant preference) 09 = Challenge requested - the merchant requests a whitelist prompt if a challenge is required **Note**: Values '04', '05', '06', '07', '08' and are reserved for BOIPA gateway use **Netcetera Constraint**: Value '09' is only available when Netcetera initiates authentication with EMV 3DS 2.2.0 version or greater. In this instance, the *threeDSPreferredProtocolVersion* and *enforcethreeDSPreferredProtocolVersion* parameters should be set appropriately
merchantDecReqInd	String (enum)	N	Merchant Decoupled Request Indicator: Indicates whether the merchant requests the Issuer to utilise Decoupled Authentication and agrees to utilise Decoupled Authentication if the Issuer confirms its use. **Values accepted**: Y = Decoupled Authentication is supported and preferred if challenge is necessary N = Do not use Decoupled Authentication - **Default if not provided** **Netcetera Constraint**: Parameter is only available when Netcetera initiates authentication with EMV 3DS 2.2.0 version or greater. In this instance, the *threeDSPreferredProtocolVersion* and *enforcethreeDSPreferredProtocolVersion* parameters should be set appropriately
merchantDecMaxTime	Integer (5)	N	Merchant Decoupled Request Maximum Wait Time: Indicates the maximum amount of time that the merchant will wait for an Issuer to provide the results of a Decoupled Authentication transaction (in minutes). Valid values are between 1 and 10080. If not provided, it is expected that the Issuer will use 10080 minutes (7 days) as a default. **Netcetera Constraint**: Parameter is only available when Netcetera initiates authentication with EMV 3DS 2.2.0 version or greater. In this instance, the *threeDSPreferredProtocolVersion* and *enforcethreeDSPreferredProtocolVersion* parameters should be set appropriately
channel	String (enum)	Y	The transaction channel through which the payment was taken: "ECOM" for card present e-commerce type transactions that are customer initiated, usually through a website checkout screen "MOTO" for card not present transactions that are merchant initiated, usually through a virtual terminal type application developed by the merchant

Parameter	Data Type	Required	Description
country	String (enum)	Y	The ISO alpha-2 code country in which the transaction takes place, as defined in the ISO 3166 standard If this is not known or unavailable, the *customerAddressCountry* will be used.
allowOriginUrl	String (256)	Y	The merchant's URL that will make the Auth/Purchase/Verify Request (see Section 2.1) This will usually be the URL of the customer's browser. Cross-Origin Resource Sharing (CORS) headers will allow only this origin
merchantNotificationUrl	String (200)	N	The merchant's server-to-server communications URL, to which the Transaction Result Call will be sent It is highly recommended that this parameter is provided, so that the merchant receives a timely result of the payment authentication and authorisation in the Transaction Result Call. If not provided, no immediate notification will be sent to the merchant. The transaction result will be shown in the BOIPA gateway Back-Office or it can be retrieved using the GET STATUS API Operation.
merchantLandingPageUrl	String (200)	N	The URL to which the customer's browser is redirected for success or failure messaging
merchantLandingPageRedirectMethod	String (enum)	N	Determines the method by which the customer is redirected to *merchantLandingPageUrl* **Values accepted:** 'POST' – **Default** if omitted 'GET'

Parameter	Data Type	Required	Description
External Authentication The following parameters are provided for the merchant to provide evidence of a transaction that has been authenticated in the 3DS Process with an external supplier. Provision of this data allows the BOIPA gateway to send the transaction straight to the Payment Authorisation Service for the merchant. The BOIPA gateway will not validate the values or attempt to determine whether the merchant's Authentication process completed successfully. Merchants should generally only make application for payment authorisation if the cardholder authentication was successful, i.e. Transaction Status from the Authentication process was 'Y', 'A' or 'I', although some 'U' conditions may be accepted by the Issuer.			
authenticationValue	String (28)	N	The Authentication Value used to provide proof of authentication. This is a Payment System-specific value provided by the 3DS Process. **Value accepted**: A 20-byte value that has been Base64 encoded, giving a 28-byte result.
authenticationECI	String (2)	N	Payment System-specific value provided by the ACS or DS to indicate the results of authentication of the cardholder. **Values accepted** are Payment System specific
protocolVersion	String (8)	C	The 3DS Protocol Version used to authenticate the transaction, returned by the authentication service **Values accepted**: '1.0' '2.1.0' '2.2.0' **Condition**: Required if transaction has been authenticated by an external authentication service
threeDSServerTransId	String (28)	C	The 3DS Server Transaction Identifier that was returned by the authentication service **Condition**: Required if transaction has been authenticated by an external authentication service
dsTransID	String (36)	C	The Directory Server (DS) Transaction Identifier that was returned by the authentication service **Conditions**: • Required if transaction has been authenticated by an external authentication service • Required if *protocolVersion* = '2.1.0'
acsTransId	String (36)	C	The Access Control Server (ACS) Transaction Identifier that was returned by the authentication service **Conditions**: • Required if transaction has been authenticated by an external authentication service, AND • Required if *cardOnFileInitiator* = 'Merchant'
authenticationType	String (2)	C	The type of authentication **Conditions**: • Required if transaction has been authenticated by an external authentication service • Optional if *protocolVersion* = '2.1.0'

Parameter	Data Type	Required	Description
authenticationFlow	String (1)	C	A flag to indicate whether the authentication occurred as a result of a Frictionless or Challenge Flow **Values accepted**: 'F' = Frictionless Flow 'C' = Challenge Flow 'A' = AVS Only **Condition**: • Required if transaction has been authenticated by an external authentication service • Optional if *protocolVersion* = '2.1.0'
transStatus	String (1)	C	Transaction Authentication Status that was returned by the authentication service **Values accepted**: 'Y' Authentication Verification Successful 'N' Not Authenticated/Account Not Verified; Transaction denied 'U' Authentication/Account Verification Could Not Be Performed; Technical or other problem 'A' Authentication Attempted; Not Authenticated/Verified, but a proof of attempt provided 'C' Challenge Required 'D' Decoupled Authentication Challenge Required 'R' Authentication/Account Verification Rejected 'I' Informational Only; 3DS Requestor challenge preference acknowledged **Condition**: Required if transaction has been authenticated by an external authentication service
authenticationDateTime	String (12)	C	Date and time in UTC of the cardholder authentication **Format**: YYYYMMDDHHMM **Condition**: Required if transaction has been authenticated by an external authentication service

Parameter	Data Type	Required	Description
Payment Method Data The Payment Method Data defines how the merchant's customer wishes to pay for an Authorisation or Purchase (*action* = 'AUTH' or 'PURCHASE') The *action* = 'VERIFY' can only be performed on payment cards. The following parameters are required for payment cards: • *paymentSolutionId* = 500 • *specinCreditCardToken*			
paymentSolutionId	Integer (18)	N	The BOIPA gateway Payment Solution Identifier See *BOIPA gateway – 7 – GET AVAILABLE PAYMENT SOLUTIONS* for valid values
specinCreditCardToken	String (100)	C	The payment card token received in the TOKENIZE API Operation, see *BOIPA gateway – 1 – TOKENIZE* document **Conditions**: 1. This parameter is required for Card Payments 2. For OneClick transactions this **must** be the *data.oneClickPaymentMethods.payToken* returned in the Get OneClick Payment Methods Response – Processed
specinProcessWithoutCvv2	Boolean	N	A flag that indicates whether the payment card transaction is to be processed with or without the Card Verification Value [CVV]. The CVV is provided in the *specinCreditCardCVV* parameter in the Auth/Purchase/Verify Request (Section 2.1). If not provided, a true value is assumed. If the *specinCreditCardCVV* parameter is then not provided, the Auth/Purchase/Verify Request will be rejected by the BOIPA gateway. This requires prior authorization by the BOIPA gateway and acquirer. **Note**: The CVV is also known as Card Security Code (CSC), Card Verification Data [CVD], Card Verification Number, Card Verification Value Code, Card Verification Code [CVC], Verification Code [V-code or V code], or Signature Panel Code [SPC])
forceSecurePayment	Boolean	C	For payment card transactions only, if the merchant has 3D Secure disabled for all transactions as a rule, this field can be used to force 3D Secure processing for individual transactions: • If True: forces 3D Secure processing no matter the routing rules • If False, not provided or NULL: the 3D Secure routing rules in the BOIPA gateway are used If 3D Secure processing is required, the 3DS Redirection Response (section 2.2) is sent **Condition** • This parameter is only valid for 3DS Version 1.0. In 3DS Version 2.x processing, the *merchantChallengeInd* is used to determine the merchant's preference for Authentication processing • If *cardOnFileType* = "Repeat" the *forceSecurePayment* parameter should be omitted. If it is included with any value (true, false, or empty) the parameter will be ignored
processUnknownSecurePayment	Boolean	N	Determines how 3DSV1.0 Authentication Response "U" (Unknown) value is processed: If True and 'U' is returned: a Session Token Response – Not Processed (section 1.3) is returned If 3DS Version 2.x Authentication is used, this parameter is ignored. The processing of the 'U' Authentication response is determined by the transaction status reason provided in the 3DS Authentication process.

Parameter	Data Type	Required	Description
Merchant Transaction Data Merchant Transaction Data provides information about the merchant's bank account, information needed to recognise the merchant in the acquirer and settlement systems, and data that the merchant wants to add to the transaction for post settlement reconciliation and processing.			
merchantTxId	String (50)	N	The merchant's reference for the transaction. If the parameter is empty or omitted, a reference will be generated by the BOIPA gateway as a hexadecimal string, and returned in the transaction responses **It is highly recommended that a value is supplied to reconcile transactions in the BOIPA gateway with the merchant's own order management system**
operatorId	String (20)	N	Identifier of the merchant's operator or agent on behalf of the end customer, if the operation is not performed by the merchant, and the merchant wants to track the operator who performed the transaction
brandId	Integer (18)	N	The BOIPA gateway Brand Id for the merchant's goods or services that was supplied at on-boarding If not provided the merchant's default BOIPA gateway Brand Id will be used
bankMid	String (50)	N	The merchant's Bank MID with the Acquirer. Used by the merchant to control which acquirer bank MID will be used for the transaction.
limitMin	BigDecimal (15.2 or 15.3)	N	Sets a minimum transaction value allowed to be processed in the BOIPA gateway This overrides the minimum value set in the BOIPA gateway merchant configuration It is the merchant's responsibility to set a value that will be accepted by the Acquirer and Issuer involved in the transaction process
limitMax	BigDecimal (15.2 or 15.3)	N	Sets a maximum transaction value allowed to be processed in the BOIPA gateway This overrides the maximum value set in the BOIPA gateway merchant configuration It is the merchant's responsibility to set a value that will be accepted by the Acquirer and Issuer involved in the transaction process
freeText	String (200)	N	A free text field for use by the merchant that is returned in the Transaction Result Call (*see BOIPA gateway - 6 - TRANSACTION RESULT CALL*)
customParam1_OR … customParam20_OR	String (50)	N	20 Text Fields that used by merchants to reconcile transactions performed through mobile applications with results from the acquirer. **Currently only available for EVO Poland merchants.**
s_text1, s_text2... s_text5	String (200)	N	5 Text fields for general use
d_date1, d_date2... d_date5	Date/Time	N	5 Date fields for general use. Format: DD/MM/YYYY hh:mm:ss – the time part can be omitted, resulting in 00:00:00
b_bool1, b_bool2... b_bool5	Boolean	N	5 Boolean fields for general use – accepted values are "true" and "false"
n_num1, n_num2... n_num5	BigDecimal (7.2)	N	5 Numeric fields for general use – a dot "." must be used as a decimal separator, not the comma "," and a thousand separator must not be used
Customer Browser/App/Device Data The Customer Browser/App/Device Data is required to support Strong Customer Authentication (SCA) and 3DS V2.x when an Authentication Challenge (3DS) is required. Although the parameters are non-mandatory in the initial release, as much information should be supplied as is available. This will enable card issuers to provide more Frictionless Flows in the Authentication processes, where the cardholder is not challenged during the transaction.			
userDevice	String (enum)	C	Type of device used, accepted values: • "MOBILE" • "DESKTOP" **Condition**: Required for 3DSV2.x. If not supplied, 3DSV1.0 Authentication will be used

Parameter	Data Type	Required	Description
userAgent	String (2048)	C	Browser User-Agent: Exact content of the HTTP user-agent header from the browser in which the transaction was performed **Note**: If the total length of the User-Agent sent by the browser exceeds 2048 characters, the excess content will be truncated. **Conditions**: • Required for 3DSV2.x. If not supplied, 3DSV1.0 Authentication will be used • Required if *customerBrowser.browserJavascriptEnabled* = true
customerIPAddress	String (45)	C	Browser IP Address: IP address of the customer's browser, where the transaction is initiated, as returned by the HTTP headers to the merchant **Value accepted**: • IPv4 address is represented in the dotted decimal format of 4 sets of decimal numbers separated by dots. The decimal number in each and every set is in the range 0 to 255. Example IPv4 address: 1.12.123.255 • IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two octets. The groups are separated by colons (:). Example IPv6 address:2011:0db8:85a3:0101:0101:8a2e:0370:7334 **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.

customerBrowser	JSON Object	C	Customer Browser Information: Information about the customer's browser that is required by the 3DS Process to facilitate the Challenge Flow, if required.
			Accurate Browser Information is required for an Issuer to determine the ability to support authentication on a particular Cardholder browser for each transaction. The data must be unique to each transaction.
			This data can be provided to the merchant on request or through for example, remote JavaScript calls.
			Format:
			"customerBrowser" : {
			"browserAcceptHeader":" ... ",
			"browserJavaEnabled":" ... ",
			"browserJavascriptEnabled":" ... ",
			"browserLanguage":" ... ",
			"browserColorDepth":" ... ",
			"browserScreenHeight":" ... ",
			"browserScreenWidth":" ... ",
			"browserTZ":" ... "
			}
			Condition: This object is mandatory for 3DS V2.x; if not provided 3DS V1.0 will be applied
			See Appendix C - customerBrowser Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Required	Description

sdkAppInfo	JSON Object	C	Mobile Application Information: Information about the mobile application installed on the customer's device that is required by the 3DS Process to facilitate the Challenge Flow, if required.
			Accurate App Information is required for an Issuer to determine the ability to support authentication on a particular Cardholder device for each transaction. The data must be unique to each transaction.
			Format:
			"sdkApp" : {
			"sdkAppId":" ... ",
			"sdkEncryptedData":" ... ",
			"sdkPublicKey":" ... ",
			"sdkMaxTimeout":" ... ",
			"sdkReferenceNumber":" ... ",
			"sdkTransactionId":" ... ",
			"sdkInterface":" ... ",
			"sdkUiType":" ... "
			}
			Condition: This object is mandatory for 3DS V2.x; if not provided 3DS V1.0 will be applied
			See Appendix D - sdkAppInfo Data Element Definitions for the data elements' definitions.

Transaction Amount Data Transaction Amount Data provides the values of the sale.			
amount	BigDecimal (15.2 or 15.3)	C	The total transaction amount, including tax, shipping, surcharge and discount amounts **Conditions**: If *action* = "AUTH" or "PURCHASE", if a value is supplied this must be > 0.00 If *action* = "VERIFY", this must be 0.00 or omitted See Appendix A - UAT Trigger Values
currency	String (enum)	Y	The ISO alpha-3 code for the currency as defined in the ISO 4217 standard
taxAmount	BigDecimal (15.2 or 15.3)	N	Tax amount as a currency value (not percentage) If *action* = "VERIFY", this must be 0.00 or omitted
shippingAmount	BigDecimal (15.2 or 15.3)	N	Shipping amount If *action* = "VERIFY", this must be 0.00 or omitted
chargeAmount	BigDecimal (15.2 or 15.3)	N	Surcharge amount If *action* = "VERIFY", this must be 0.00 or omitted
discountAmount	BigDecimal (15.2 or 15.3)	N	Discount amount If *action* = "VERIFY", this must be 0.00 or omitted
Customer Personal Data Customer Personal Data identifies the customer involved in the transaction. The supply and storage of this data is subject to regional restrictions (such as GDPR in the EU). Although all fields are non-mandatory, the minimum data that should be supplied are *customerFirstName* and *customerLastName*, which will allow the merchant to easily identify transactions for their customers in the BOIPA gateway Back-Office Transactions Lists.			
customerFirstName	String (50)	N	First name of the customer
customerLastName	String (100)	N	Last name, surname or family name of the customer

Parameter	Data Type	Required	Description
customerSex	String (enum)	N	Customer sex: • M (male) • F (female)
customerDateOfBirth	Date	N	Customer date of birth – format DD/MM/YYYY
customerEmail	String (60)	N	Customer email address
customerPhone	String (100)	N	Customer phone number
customerDocumentType	String (enum)	N	Type of document used to confirm the customer's identification BOIPA gateway accepted values: • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UNIQUE_TAXPAYER_REFERENCE • OTHER
customerDocumentNumber	String (30)	C	Customer document number **Condition**: Mandatory if *customerDocumentType* provided
customerDocumentState	String (2)	C	For EVOUS Sales Channel Merchants, the alpha-2 code for the State that issued the Driver's Licence. **Condition**: Mandatory if merchant Sales Channel is 'EVOUS' and *customerDocumentType* = 'DRIVING_LICENSE' and if *country* = • 'US' alpha-2 code for the US State that issued the licence – see Section B.1 • 'CA' alpha-2 code for the Canadian State that issued the licence – see Section B.2 • 'MX' alpha-2 code for the Mexican State that issued the licence – see Section B.3 • Else set to NULL
Payer Data The payer data is required by some regions and payment services, e.g. PayU Latam in Brazil, and so should only be completed if required by regulation. This data is not used to differentiate between the customer and someone else paying for the transaction. No checking or validation is performed by the BOIPA gateway.			
payerFirstName	String (50)	N	Payer first name, if the Payee is different to the Customer
payerLastName	String (100)	N	Payer last name, if the Payee is different to the Customer
payerEmail	String (60)	N	Payer email, if the Payee is different to the Customer
payerDateOfBirth	Date	N	Payer date of birth, if the Payee is different to the Customer
payerPhone	String (100)	N	Payer phone, if the Payee is different to the Customer
payerDocumentType	String (enum)	N	Type of document used to confirm the payer's identification, if the Payee is different to the Customer BOIPA gateway accepted values: • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UTR • OTHER

Parameter	Data Type	Required	Description
payerDocumentNumber	String (30)	C	Payer document number, if the Payee is different to the Customer **Condition**: Mandatory if *payerDocumentType* provided
payerCustomerId	String (20)	C	Customer identifier of the payee in the merchant's system **Condition**: Required if the payee is also a customer of the merchant

Parameter	Data Type	Required	Description
<u>Customer Account Data with the Merchant</u> Customer Account Data is used in the BOIPA gateway to supplement the transaction data to support Frictionless Flows in Strong Customer Authentication (SCA) and 3DS V2.x. **This parameter is optional, but it is recommended that it is provided if this information is available. Although individual data elements are optional, as much available information should be provided as is available.**			
customerId	String (20)	N	Customer identifier in the merchant system or the value generated by the BOIPA gateway in the TOKENIZE API Operation (see the *BOIPA gateway – 1 – TOKENIZE* document). If supplied, this must be the value supplied in or by the TOKENIZE API Operation. The value is used to validate that the payment card token is for the correct customer. If the *customerId* value is not the same held against the payment card token the payment request will not be processed. If the parameter is omitted: • For payment cards the value stored during the TOKENIZE Operation will be used • For Alternate Payment Methods a value will be generated by the BOIPA gateway
merchantReference	String (200)	N	Merchant's supplementary information about customer **Note**: this information is only stored in the BOIPA gateway, and not used in the payment process
customerRegistrationDate	Date	N	Customer registration date on merchant's site – format DD/MM/YYYY **This parameter is optional, but it is recommended that it is provided if the information is available.** **Notes**: 1. Used in the 3DS V2.x Authentication process as part of the *customerAccountInfo* 2. Used for reporting and in some risk tools where required

Parameter	Data Type	Required	Description

customerAccountInfo	JSON Object	N	Customer Account Information: Additional information about the Cardholder's account provided by the merchant.
			This parameter is optional, but it is recommended that it is provided if the information is available.
			Format:
			"customerAccountInfo" : {
			"custAccAgeInd":" … ",
			"custAccChange":" … ",
			"custAccChangeInd":" … ",
			"custAccPwChange":" … " ,
			"custAccPwChangeInd":" … " ,
			"custPurchaseCount":" … " ,
			"custProvisionAttemptsPerDay":" ... ",
			"custTxnActivityDay":" ... ",
			"custTxnActivityYear":" ... ",
			"custPaymentAccAge":" ... ",
			"custPaymentAccInd":" ... ",
			"custShipAddressUsage":" ... ",
			"custShipAddressUsageInd":" ... ",
			"custShipNameIndicator":" ... ",
			"custSuspiciousAccActivity":" ... "
			}
			See below for the data elements' definitions.
			Note: Cardholder Account Information data elements used to define a time period can be included as either the specific date or an approximate indicator for when the action occurred. Merchants can use either parameter, e.g. use *chAccAgeInd* or *chAccChange*, using both is unnecessary.
			See Appendix E - customerAccountInfo Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Required	Description
Customer Address Data Customer address data are required for 3DSV2.x Authentication unless market or regional mandate restricts sending this information. If address is included, at least one of *customerAddressHouseName*, *customerAddressHouseNumber* or *customerAddressFlat* should be provided. The *customerBillingAddress* and *customerShippingAddress* parameters are marked as Not Require (N) to allow for merchant flexibility in their data encoding: 1. If *customerBillingAddress* data are omitted, the *customerAddress* data will be used for the customer billing address 2. If *customerShippingAddress* data are omitted, the *customerAddress* data will be used for the customer shipping address Therefore: A. To use the customerAddress parameters as the customer's billing and shipping address, omit the customerBillingAddress and customerShippingAddress parameters B. To use the customerBillingAddress as the customer's shipping address, but different to the customerAddress values, complete the customerShippingAddress parameters with the same data C. To use the customerAddress parameters as the customer's billing address and have a different shipping address, omit the customerBillingAddress and complete the customerShippingAddress parameters D. To use the *customerAddress* parameters as the customer's shipping address and have a different billing address, omit the *customerShippingAddress* and complete the *customerBillingAddress* parameters			
customerAddressHouseName	String (50)	C	Customer correspondence address house name **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressHouseNumber	String (5)	C	Customer correspondence address house number **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressFlat	String (5)	C	Customer correspondence address flat **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressStreet	String (50)	C	Customer correspondence address street The customer's street should be supplied whenever possible as it is used with the *customerAddressPostalCode* value for AVS (Address Verification System) Checks, and so reduce the possibility of a payment decline **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressCity	String (50)	C	Customer correspondence address city **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressDistrict	String (50)	N	Customer correspondence address district
customerAddressPostalCode	String (30)	C	Customer correspondence address postal code **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressCountry	String (2)	C	Customer correspondence address country: The ISO alpha-2 code as defined in the ISO 3166 standard **Note**: this will be used if *country* field is not supplied **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressState	String (40)	C	Customer correspondence address state, county or province **Condition**: Required for 3DSV2.x unless market or regional mandate restricts sending this information.
customerAddressPhone	String (100)	N	Customer correspondence address phone
customerBillingAddressHouseName	String (50)	N	Customer billing address house name
customerBillingAddressHouseNumber	String (5)	N	Customer billing address house number
customerBillingAddressFlat	String (5)	N	Customer billing address flat
customerBillingAddressStreet	String (50)	N	Customer billing address street

Parameter	Data Type	Required	Description
customerBillingAddressCity	String (50)	N	Customer billing address city
customerBillingAddressDistrict	String (50)	N	Customer billing address district
customerBillingAddressPostalCode	String (30)	N	Customer billing address postal code
customerBillingAddressCountry	String (enum)	N	Customer billing address country The ISO alpha-2 code as defined in the ISO 3166 standard
customerBillingAddressState	String (40)	N	Customer billing address state
customerBillingAddressPhone	String (100)	N	Customer billing address phone
customerShippingAddressHouseName	String (50)	N	Customer shipping address house name
customerShippingAddressHouseNumber	String (5)	N	Customer shipping address house number
customerShippingAddressFlat	String (5)	N	Customer shipping address flat
customerShippingAddressStreet	String (50)	N	Customer shipping address street
customerShippingAddressCity	String (50)	N	Customer shipping address city
customerShippingAddressDistrict	String (50)	N	Customer shipping address district
customerShippingAddressPostalCode	String (30)	N	Customer shipping address postal code
customerShippingAddressCountry	String (enum)	N	Customer shipping address country The ISO alpha-2 code as defined in the ISO 3166 standard
customerShippingAddressState	String (40)	N	Customer shipping address state, county or province
customerShippingAddressPhone	String (100)	N	Customer shipping address phone

Parameter	Data Type	Required	Description
Additional Authentication Data The Additional Authentication Data has been introduced by the Secure Customer Authentication (SCA) and 3DS V2.x processes to combat fraud and increase electronic payment security for customers. Although the parameters are non-mandatory in the initial release, it is highly recommended to provide as much information as possible. This will enable card issuers to provide more Frictionless Flows in the Authentication processes, where the cardholder is not challenged during the transaction.			

merchantAuthInfo	JSON Object	N	Merchant Authentication Information: Information about how the merchant authenticated the cardholder before or during the transaction.
			This parameter is optional, but it is recommended that it is provided if the information is available.
			Also, although the individual data elements are optional, as much available information should be provided as is available.
			Format:
			"merchantAuthInfo" : {
			"merchantAuthData":" ... ",
			"merchantAuthMethod":" … ",
			"merchantAuthTimestamp":" … "
			}
			See Appendix F - merchantAuthInfo Data Elements Definitions for the data elements' definitions.

merchantPriorAuthInfo	JSON Object	N	Merchant Prior Transaction Authentication Information: Information about how the merchant authenticated the cardholder as part of a previous 3DS transaction.
			This parameter is optional, but it is recommended that it is provided if the information is available.
			Also, although the individual data elements are optional, as much available information should be provided as is available.
			Format:
			"merchantPriorAuthInfo" : {
			"merchantPriorAuthData":" ... ",
			"merchantPriorAuthMethod":" … ",
			"merchantPriorAuthTimestamp":" … ",
			"merchantPriorRef":" … "
			}
			If any data element is not provided, this object will not be included in the Authentication Request
			See Appendix G - merchantPriorAuthInfo Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Required	Description

merchantRiskIndicator	JSON Object	N	Merchant Risk Indicator: Merchant's assessment of the level of fraud risk for the specific authentication for both the cardholder and the authentication being conducted.
			This parameter is optional, but it is recommended that it is provided if this information is available.
			Also, although the individual data elements are optional, as much available information should be provided as is available.
			Format:
			"merchantRiskIndicator" : {
			"deliveryEmailAddress":" ... ",
			"deliveryTimeframe":" ... ",
			"giftCardAmount":" ... ",
			"giftCardCount":" ... ",
			"giftCardCurr":" ... ",
			"preOrderDate":" ... ",
			"preOrderPurchaseInd":" ... ",
			"reorderItemsInd":" ... ",
			"shipIndicator":" ... "
			}
			See Appendix H - merchantRiskIndicator Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Required	Description
Card On File Transactions Required Parameters Transactions that are initiated by stored payment card credentials, stored either by the merchant or in the BOIPA gateway, must be identified in the payment process through to the Card Issuers and Card Schemes. By their nature, these transactions, where the cardholder is not present at the point of initiation, will not have card or cardholder authentication data accompanying the transaction. To enable the Schemes and Issuers to assess risk and determine potential fraud accurately, new indicators and processes have been introduced to provide greater clarity into transactions using stored credentials. The following '*cardOnFile*' prefixed parameters are provided to comply with these requirements. These parameters must be provided for: • Recurring Payments Plans transactions – these are Plans managed by the merchant, either initiated using this API or the BOIPA gateway's Hosted Payment Page • Stored Credential Payments – these are where the cardholder has consented to the merchant storing the card details (except the CVV/CSC), which will be presented back to the cardholder in future payments, so that the customer does not have to re-enter the payment card information The field rules are: • For the initial transaction: o *cardOnFileType* is set to 'First' only o *cardOnFileInitiator* and *cardOnFileInitialTransactionId* parameters are omitted **Note**: if the *cardOnFileInitiator* and *cardOnFileInitialTransactionId* parameters are included they will be ignored by the BOIPA gateway • Subsequent (recurring) payment requests must have the following values: o *cardOnFileType* is set to 'Repeat' o *cardOnFileInitiator* is set to ▪ 'Merchant' for Recurring Payments ▪ 'Cardholder' for OneClick *cardOnFileInitialTransactionId* is set to the *merchantTxId* value returned in the Auth/Purchase/Verify Response – Processed (section 2.3) of the initial transaction			
cardOnFileType	String (10)	C	Indicates if the transaction is the first in a series of COF transactions or a transaction from already stored credentials **Conditions**: Mandatory if the payment originates from stored payment card credentials, i.e. the cardholder or merchant user did not input the card data during the transaction process, e.g. OneClick or pre-populated payment pages from stored card data **Permitted Values** "First": If the transaction is starting a series of COF transactions "Repeat": If the transaction is a subsequent transaction
cardOnFileInitiator	String (10)	C	Indicates if the COF transaction is either a: • Cardholder Initiated Transaction (CIT) where the cardholder actively selects the card to use, and completes the transaction using previously stored details. • Merchant Initiated Transaction (MIT) where a merchant submits a transaction using previously stored detailed without the cardholder's participation. For example, a recurring payment. **Condition**: Mandatory if *cardOnFileType* = "Repeat" **Permitted Values** "Cardholder": If a Cardholder Initiated Transaction "Merchant": If a Merchant Initiated Transaction

Parameter	Data Type	Required	Description
cardOnFileInitialTransactionId	String (50)	C	The merchant's transaction identifier[1] for the transaction that started the COF series of payments, i.e. the transaction where *cardOnFileType* = "First"; the *merchantTxId* value sent in the original Session Token Request or returned in the Auth/Purchase/Verify Response – Processed **Note**: this *must* be the transaction identifier for the specific set of transactions. For example, if the customer has multiple recurring payments plans with the merchant, this value for the payment being request must be the initial payment for the plan **Conditions**: • Mandatory if *cardOnFileType* = "Repeat" • For OneClick transactions this **must** be the *data.oneClickPaymentMethods.originalTransactionId* returned in the Get OneClick Payment Methods Response – Processed **Note**: If the initial transaction identifier is not known, the value "999999999999999" should be provided. This situation can arise from a migration to a new payment service provider or during the transition to the new Card On File requirements when continuing legacy recurring payments. This should not be used for any other purposes, otherwise an increase in payment decline rate could be experienced
cardOnFileReason	String (1)	C	Indicates the type of series of COF transactions **Condition**: Mandatory if *cardOnFileType* = "First" or "Repeat" **Values**: • "I": Installments • "R": Recurring • "H": Reauthorization • "E": Resubmission • "D": Delayed • "M": Incremental • "N": No Show • "C": Other
cardOnFileMaxPayments	Integer (3)	C	Indicates the maximum number of authorisations permitted for instalment payments, where cardOnFileReason = 'I'. Must be greater than 1. **Condition**: Mandatory if the Merchant and Cardholder have agreed to instalment payments, i.e. cardOnFileReason = 'I'

[1] **Note**: this is used to match the constraint in the REFUND API Operation where the *originalMerchantTxId* is mandatory, whereas the *originalTxId* (the EVO Gateway transaction identifier) is non-mandatory. Therefore, it is more likely that the merchant would already have a method for their ID. See Section 1.1 of the *API Specification - 3 - REFUND*

Parameter	Data Type	Required	Description
Merchant Managed Recurring Payment Plan Required Parameters The following fields prefixed with "*mmrp*" are provided for the merchant to be able to send transaction data from Merchant Managed Recurring Payment Plans. These data are required by the Acquirer, Issuers and Card Schemes to recognise that a Recurring Payment Plan is being created and to accept subsequent transactions in a plan as being related to the initiating transaction. A Recurring Payment transaction is a transaction for which a cardholder provides written permission to a merchant to periodically charge his/her account number for recurring goods or services. These may include payment of charges such as insurance premiums, subscriptions, membership fees, tuition or utility charges. The recurring transaction indicator must be present in the authorization/initial purchase/sale. Address verification must be obtained with the initial transaction and is not required in the subsequent recurring transactions that contain the recurring indicator. Address verification is required to be obtained yearly. **Notes**: 1. The data values must be as stated in the Description 2. The data must be accompanied with the "*cardOnFile*" prefixed data above The data are not required if the merchant is setting up an BOIPA gateway Managed Recurring Payment Plan in the BOIPA gateway (see the "*rp*" prefixed fields below)			
mmrpBillPayment	String (10)	N	For the initial and subsequent transactions must be set to "Recurring"
mmrpCustomerPresent	String (12)	C	For the initial and subsequent transactions must be set to "BillPayment" **Condition**: required if mmrpBillPayment = "Recurring"
mmrpRecurringExpiry	Date	C	Date after which no further recurring payments authorisations shall be performed, i.e. the expected date of the final payment of the Recurring Payments Plan. **Format**: YYYYMMDD **Condition**: required if mmrpBillPayment = "Recurring"
mmrpRecurringFrequency	Integer (4)	C	The minimum number of days between Plan payments. Examples: • Daily Plans: 1 • Weekly Plans: 7 • Monthly Plans 28 **Condition**: required if *mmrpBillPayment* = "Recurring"
mmrpContractNumber	String (50)	C	Contract number is managed by the merchant and must be unique for each contractual agreement between the merchant and cardholder. Required for the initial and subsequent transactions **Conditions**: Required if mmrpBillPayment = "Recurring" **Required for Banamex (EVO MX) merchants only**
mmrpExistingDebt	String (15)	C	For the initial and subsequent transactions must be set to "NotExistingDebt" **Conditions**: required if mmrpBillPayment = "Recurring" **Required for Banamex (EVO MX) merchants only**
mmrpCurrentInstallmentNumber	Number (1)	C	For the initial and subsequent transactions must be set to "1" **Condition**: required if mmrpBillPayment = "Recurring" **Required for Banamex (EVO MX) merchants only**
mmrpOrderNumber	String (50)	C	The merchant's reference for the transaction. This will generally be the same as the *merchantTxId* above, if provided, but can be another unique reference for the merchant's own reconciliation. **Condition**: required if *mmrpBillPayment* = "Recurring" **Required for Banamex (EVO MX) merchants only**

Parameter	Data Type	Required	Description
mmrpOriginalMerchantTransactionId	String (50)	C	For the initial and subsequent transactions must be set to the *merchantTxId* parameter value **Condition**: required if mmrpBillPayment = "Recurring"
BOIPA gateway Recurring Payment Plan Setup Required Parameters The following fields prefixed with "rp" are provided for the merchant to be able to set up an BOIPA gateway Managed Recurring Payment Plan with their customer in the BOIPA gateway. The data must only be sent with the Request for the payment/verification that will initiate the Recurring Payment Plan series of payments. All subsequent payment requests will be generated by the BOIPA gateway. The transaction results will be returned to the merchant in a Transaction Result call when complete. The BOIPA gateway Managed Recurring Payment Plans created by this process can be seen and managed in the BOIPA gateway Back-Office/Virtual Terminal Recurring Payments menu option. **Notes**: 1. If the parameters are completed, the 'cardOnFileType' parameter must be set to "First". If not, an error will be returned stating that the parameter is missing. 2. If the merchant has not been configured for Recurring Payments in the BOIPA gateway and data is present where *rpPlanType* > 0, an error will be returned stating that the merchant is not authorised for Recurring Payments and the payment will not be processed. 3. Errors will be returned in the Session Token Response – Not Processed (section 1.3) Therefore, for merchants that have not been configured for Recurring Payment Plans all these fields must be omitted or empty (*rpPlanType* can be set to '0').			
rpPlanType	Number (1)	C	Defines the type of Recurring Payment to be created **Condition**: Only required in the initial transaction to create the recurring payment plan in the BOIPA gateway **Permitted Values:** 0 or missing = None (all Recurring Payments fields must be empty/will be ignored) 1 = Subscription 2 = Direct Debit 3 = Repayment 4 = Pay Per Use
rpPlanName	String (200)	C	The name of the Recurring Payments Plan given by the merchant **Condition**: Required if *rpPlanType* > 0 **Permitted Values:** free text for the merchant's easy reference in the BOIPA gateway Back-Office/Virtual Terminal

Parameter	Data Type	Required	Description
rpFrequency	Number (2)	C	Indicates how often payments are taken. **Condition**: Required if *rpPlanType* > 0 **Permitted Values**: The value is dependent on the *rpPlanType* value: If *rpPlanType* = 4 must be 0 Ad hoc or not known Else one of the following 20 Daily 23 Every 3 Days 1 Weekly 22 Every 2 Weeks 2 Monthly 3 Every 3 Months / Quarterly 4 Every 6 Months 5 Yearly
rpNoOfPayments	Number (3)	C	The total number of payments to be taken **Condition**: Required if *rpPlanType* > 0 **Permitted Values**: The value is dependent on the *rpPlanType* value: 1 can be 0 or > 1 2 can be 0 or > 1 3 must be > 1 4 must be 0 If the plan is open-ended then *rpNoOfPayments* must be 0 If the plan is fixed term, then *rpNoOfPayments* must be > 1 (the first payment counts as 1)
rpDueDay	Number (2)	C	Defines the date on which the payment is due. This value is used to calculate the next payment due date after a payment is taken. Only for the second payment after the initial payment, this can be overridden by *rpNextPaymentDate*, but the third and subsequent payments will be calculated from the *rpFrequency* and *rpDueDay* values provided. **Note**: these can be changed in the Back-Office/Virtual terminal Recurring Payments Plan menu option. **Condition**: Required if *rpPlanType* > 0 **Permitted Values**: The value is dependent on the *rpFrequency* value: If *rpFrequency* = 0, 20 or 23 0 If *rpFrequency* = 1 or 22 >= 1 and <= 7 the day of the week (where Monday = 1) If *rpFrequency* = 2, 3, 4 or 5 >= 1 and <= 28 the day of the month, or 32 the last day of the month

Parameter	Data Type	Required	Description
rpNextPaymentDate	Date	C	Used to force a specific date when the second payment of the Recurring Payment Plan must be taken. **Condition**: Can be provided if *rpPlanType* > 0 If not provided the next *rpNextPaymentDate* will be calculated from the *rpFrequency* and *rpDueDay* **Permitted Values**: a date in the format DD/MM/YYYY
rpAmount	BigDecimal (15.2 or 15.3)	C	The amount to be recovered from the payment card for each subsequent Recurring Payment. This can be different from the initial payment provided in the *amount* field above. **Condition**: Required if *rpPlanType* > 0 **Permitted Values**: The value is dependent on the *rpPlanType* value: 1 > 0.00 2 can be 0.00 or greater 3 > 0.00 4 can be 0.00 or greater If *rpAmount* = 0.00, the merchant will provide the values to the BOIPA gateway in text files supplied to the SFTP folder
rpFinalAmount	BigDecimal (15.2 or 15.3)	C	The final amount to be recovered from the payment card when a fixed term AUTH/PURCHASE/VERIFY (Direct API Integration) Plan ends. **Condition**: Required if *rpPlanType* > 0 **Permitted Values**: The value is dependent on the *rpPlanType* value: 1 must be 0.00 2 must be 0.00 3 must be > 0.00 can be the same as *rpAmount* 4 must be 0.00
rpContractNumber	String (50)	C	The unique Contract Number between the merchant and cardholder for the Recurring Payment Plan **Condition**: Required if *rpPlanType* is provided and merchant's sales channel is Banamex (EVO MX) **Only used by merchants from the EVO MX Sales Channel**
rpReceiptEmail	String (100)	C	The email address to which receipts should be sent for all the subsequent recurring payments. A receipt will be sent for all results of those transactions, i.e. whether successful, declined or an error. **Condition:** if *rpReceiptRequired* = 1 this field must be completed
rpCardUpdaterInterval	Integer	C	Denotes the time interval in days between successive processing of payment cards through the Card Updater Service. The maximum interval allowed by the Card Schemes is 6 months, 180 days. **Condition**: Can be provided if *rpPlanType* > 0 This is a value that is applied to the Recurring Payments Plan and will override the default value configured for the merchant in the BOIPA gateway. The field allows the merchant to change the time interval for selected Recurring Payment Plans. **Permitted Values**: Must be an integer <=180

Parameter	Data Type	Required	Description
Merchant Managed eGlobal Instalments Parameters The following parameters prefixed with "*mmip*" are provided for EVO MX/Banamex merchant's to be able to send transaction data that includes the cardholder's chosen Issuing Bank Instalment Plan. The parameters are provided for Direct API Integrated merchants who manage the Instalments Plans data in their own back-offices or virtual terminals. All parameters must be completed. **Only used by merchants from the EVO MX Sales Channel**			
mmipPlanID	String (50)	N	The merchant's identifier in the merchant's system for the Instalment Plan chosen by the cardholder **Condition**: none
mmipIssuerName	String (100)	C	The name of the Instalments Plan Issuer in the merchant's system **Condition**: required if *mmipPlanId* exists
mmipPlanName	String (25)	C	The name given to the Instalment Plan in the merchant's system **Condition**: required if *mmipPlanId* exists
mmipStartDate	Date	C	The date, in the format DD/MM/YYYY, from which the Instalments Plan is available to the merchant's customers **Condition**: required if *mmipPlanId* exists
mmipEndDate	Date	C	The date, in the format DD/MM/YYYY > *mmipStartDate*, up and until which the Instalments Plan is available to the merchant's customers **Condition**: required if *mmipPlanId* exists
mmipCurrency	String (3)	C	The currency of the instalments amount offered by the Issuer, as a 3-alpha code as defined in the ISO-4217 standard **Condition**: required if *mmipPlanId* exists
mmipMinimumAmount	Number (15,2)	C	The minimum amount, > 0.00, that can be paid in the Instalments Plan **Condition**: required if *mmipPlanId* exists
mmipNoOfPayments	Number (3)	C	The number of months, > 1, that the Instalments Plan will be for **Condition**: required if *mmipPlanId* exists
BOIPA gateway Managed eGlobal Instalments Parameter The following parameter is provided for EVO MX/Banamex merchant's to be able to select an Installment Plan from the data stored in the BOIPA gateway. The data will have been input in the BOIPA gateway Back-Office using the 'Instalments Plans' option.			
selectedInstallmentsPlanId	String (7)	N	The Plan ID of the chosen Instalments Plan If not included, the Request will be processed as a single purchase transaction **Only used by merchants from the EVO MX Sales Channel**

1.2 Session Token Response - Processed

1.2.1 Format

JSON

1.2.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “success”
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 1.1)
token	String (40)	The Session Token that is a one-time use, hexadecimal string The Token that must only be used for the Auth/Purchase/Verify Request (section 2.1) Session tokens are valid for 3600 second (1 hour) after which they expire Any requests with expired session tokens will be rejected
resultId	String (40)	Hexadecimal string that is to be used in any support request calls
processingTime	Integer (6)	The time in seconds for the process to complete
additionalDetails	Array	Not used – will always be “{}” or not included

1.3 Session Token Response – Not Processed

1.3.1 Format

JSON

1.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “failure”
errors	String Array	List of issues
resultId	String (40)	Hexadecimal string that is to be used in any support request calls
processingTime	Integer (6)	The time in seconds for the process to complete
additionalDetails	Array	Not used – will always be “{}” or not included

2 AUTH/PURCHASE/VERIFY API Operation

2.1 Auth/Purchase/Verify Request

2.1.1 Format

POST Request to Action Request URL (see Section 3 of the *BOIPA gateway – 0 – Overview* document)

2.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the BOIPA gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 1.1)
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 1.2)
freeText	String (200)	N	A free text field for use by the merchant that is returned in the Transaction Result Call (see *BOIPA gateway - 6 - TRANSACTION RESULT CALL*), can be used if not supplied in the Session Token Request (section 1.1)
customerId	String (20)	C	Customer identifier in the merchant system, or the value generated by the BOIPA gateway in a previous original payment transaction using the payment card or method. The value is used to validate that the payment card token is for the correct customer. If the *customerId* value is not the same held against the payment card token in the BOIPA gateway database a Auth/Purchase/Verify Response – Not Processed (section 2.4) is returned. This must be the value supplied in or by the TOKENIZE API Operation. The value is used to validate that the payment card token is for the correct customer. • Mandatory for payment cards method • Optional for alternative payment methods If the parameter is omitted or no value is provided for a first time use of the payment card, the BOIPA gateway will generate a value, which will be stored internally against the payment method and returned in the Auth/Purchase/Verify Response – Processed (section 2.3) **Condition**: Mandatory, if not received in the Session Token Request (section 1.1), otherwise ignored
customerIPAddress	String (39)	C	Customer IP address from where purchase is made. Only IPv4 supported **Condition**: Mandatory, if not received in the Session Token Request (section 1.1), otherwise ignored
fraudToken	String (50)	N	Antifraud token If an antifraud tool has been executed before an analysis identifier is required by payment acquirer. Mandatory for transactions conducted in LATAM countries, and only when the merchant wishes the transaction to be conducted as direct integration (server-to-server), as opposed to browser-redirection based integration.
paymentSolutionId	Integer (18)	C	Payment solution identifier in the BOIPA gateway. **Condition**: Mandatory, if not received in the Session Token Request (section 1.1), otherwise ignored

Parameter	Data Type	Mandatory	Description
setOneClickValueSettingForCard	Boolean	N	If TRUE flags that the cardholder wishes to save the card stored in the *specinCreditCardToken* parameter for future OneClick transactions • Must be TRUE if the payment card is to be saved **Note**: the card will only be available for use as a OneClick Payment Method, if the current transaction is successful. Otherwise, the payment card will not be available in the future. The customer will have to make another transaction that is successful.
specinCreditCardCVV	String (4)	C	Credit card CVV, if payment solution is credit card through the ECOM channel. **Condition**: Mandatory, if not received in the Session Token Request (section 1.1), otherwise ignored
specinCreditCardToken	String (100)	C	The payment card token received in the TOKENIZE API Operation, see *BOIPA gateway – 1 – TOKENIZE* **Condition**: Mandatory, if not received in the Session Token Request (section 1.1), otherwise ignored
ipPlanId	String (7)	N	The Plan ID of the chosen Instalments Plan If not included in the context of Instalments Plans, the API Operation will be treated as a normal single purchase transaction **Only used by merchants from the EVO MX Sales Channel**

2.2 3DS Redirection Response

The 3DS Redirection Response is used by the merchant's system to open the 3DS challenge window in the customer's browser, for the customer to enter their security information to confirm their identity.

The 3DS Redirection Response is sent if:

- *forceSecurePayment* parameter = True, in the Session Token Request (section 1.1), or
- the 3D Secure routing rules held in the BOIPA gateway for the merchant require that card payment transactions are subject to 3DS Version 1.0

2.2.1 Format

JSON

2.2.2 Definition

Parameter/Label	Data Type	Description
result	String (enum)	Will always be "redirection"
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 1.1)
merchantTxId	String (50)	The merchant's reference for the transaction provided in the Session Token Request (section 1.1) or that generated by the BOIPA gateway
txId	Integer (18)	The unique identifier for the transaction in the BOIPA gateway
redirectionUrl	String (URL)	The URL to which the customer's browser must be redirected after the 3D Secure processing is completed

2.3 Auth/Purchase/Verify Response – Processed

2.3.1 Format

JSON

2.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “success”
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 1.1)
merchantTxId	String (50)	The merchant’s reference for the transaction provided in the Session Token Request (section 1.1) or that generated by the BOIPA gateway
txId	Integer (18)	The unique identifier for the transaction in the BOIPA gateway
acquirerTxId	String (100)	The transaction identifier in acquirer system, if returned
amount	BigDecimal (15.2 or 15.3)	The transaction amount, including tax, shipping, surcharge and discount amounts, provided in the Session Token Request (section 1.1)
currency	String (enum)	The transaction ISO alpha-3 currency code as defined in the ISO 4217 standard, provided in the Session Token Request (section 1.1)
customerId	String (20)	The customer identifier provided in the Session Token Request (section 1.1), or that generated by the BOIPA gateway
action	String (enum)	Action executed as provided in the Session Token Request (section 1.1) (“AUTH”, “PURCHASE” or “VERIFY”)
pan	String (100)	The customer account value/number used in the transaction If a payment card was used this will be the *specinCreditCardToken* value provided in the Session Token Request (section 1.1)
brandId	Integer (18)	The *brandId* value received in Session Token Response, or the default value used by the BOIPA gateway, if not provided
paymentSolutionId	Integer (18)	The *paymentSolutionId* value received in the Session Token Request (section 1.1)
freeText	String (200)	A free text field for use by the merchant that is returned in the Transaction Result Call (see BOIPA gateway - 6 - TRANSACTION RESULT CALL), used if not supplied in the Session Token Request (section 1.1)
language	String (enum)	{not used for Direct API merchant}
acquirerAmount	BigDecimal (15.2 or 15.3)	Amount processed by payment acquirer. May be different than the *amount* in the Session Token Request (section 1.1)
acquirerCurrency	String (enum)	The ISO alpha-3 currency code, as defined in the ISO 4217 standard, of the currency processed by the payment acquirer, which maybe different to the *currency* in the Session Token Request (section 1.1)
paymentSolutionDetails	JSON block	For payment cards only: The Transaction Authorisation Code received from the acquirer, format: { “authCode”:””, “expiryDate”:””, “cardType”:””, “maskedPan”:””, “nameOnCard”:”” , "avsPostCode":"", "addrResultCode":"" } **Note**: the *maskedPan* value format is “999999******9999”
rpId	Integer (18)	The identifier for the BOIPA gateway Managed Recurring Payment Plan that was requested in the Session Token Request (section 6.4) through the ‘rp’ prefixed parameters If no Plan was requested this field will be empty If there was an error setting up the Plan the errors will be detailed in the *errors* field

<table>
<tr><th>Parameter</th><th>Data Type</th><th>Description</th></tr>
<tr><td>status</td><td>String (enum)</td><td>The status of the transaction in the BOIPA gateway:
<table>
<tr><th>Status</th><th>Condition</th></tr>
<tr><td>NOT_SET_FOR_CAPTURE</td><td>If “AUTH” successful</td></tr>
<tr><td>SET_FOR_CAPTURE</td><td>If “PURCHASE” successful</td></tr>
<tr><td>VERIFIED</td><td>If “VERIFY” successful</td></tr>
<tr><td>DECLINED</td><td>If “AUTH” or ”PURCHASE” was declined/refused</td></tr>
<tr><td>ERROR</td><td>If an error was returned by the payment process</td></tr>
</table></td></tr>
<tr><td>errors</td><td>String (400)</td><td>Any errors that occurred during the successful processing of a transaction</td></tr>
<tr><td>customParameter1Or … customParameter20Or</td><td>String (50)</td><td>The original 20x (50 character) free text fields provided by the merchant in the Session Token Request (section 1.1)</td></tr>
<tr><td>customParameter1 … customParameter20</td><td>String (50)</td><td>20 x (50 character) free text fields provided by the merchant in the Session Token Request (section 1.1), with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.</td></tr>
</table>

2.4 Auth/Purchase/Verify Response – Not Processed

2.4.1 Format

JSON

2.4.1 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “failure”
errors	String Array	List of issues
resultId	String (40)	Hexadecimal string that is to be used in any support request calls
processingTime	Integer (6)	The time in seconds for the process to complete
additionalDetails	Array	Not used – will always be “{}” or not included
errors	String Array	List of errors
customParameter1Or … customParameter20Or	String (50)	The original 20x (50 character) free text fields provided by the merchant in the Session Token Request (section 1.1)
customParameter1 … customParameter20	String (50)	20 x (50 character) free text fields provided by the merchant in the Session Token Request (section 1.1), with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.

Appendix A UAT Trigger Values

When integrating with the BOIPA gateway in the User Acceptance Testing (UAT) environment, certain *amount* values in the Session Token Request (section 1.1) can be used to elicit status and error messages. This facility is provided to merchants so that testing can be confirmed against these expected errors.

Amount	Status	Error Message
0.00	SUCCESS	{none}
0.01	SUCCESS	{none}
0.02	SUCCESS	{none}
0.03	ERROR	Refer to card issuer
0.04	ERROR	Refer to card issuer, special condition
0.05	ERROR	Invalid merchant
0.06	SUCCESS	{none}
0.07	ERROR	Pick-up card
0.08	ERROR	Do not honor
0.09	ERROR	Error
0.10	ERROR	Pick-up card, special condition
0.11	ERROR	Invalid transaction
0.12	ERROR	Invalid amount
0.13	ERROR	Invalid card number
0.14	ERROR	No such issuer
0.15	ERROR	Re-enter transaction
0.16	ERROR	Not sufficient funds
0.17	ERROR	Unable to locate record
0.18	ERROR	Format error
0.19	ERROR	Bank not supported
0.20	ERROR	Expired card, pick-up
0.21	ERROR	Suspected fraud, pick-up
0.22	ERROR	Contact acquirer, pick-up
0.23	ERROR	Restricted card, pick-up
0.24	ERROR	Call acquirer security, pick-up
0.25	ERROR	PIN tries exceeded, pick-up
0.26	ERROR	No savings account
0.27	ERROR	No card record
0.28	ERROR	Lost card, pick-up
0.29	ERROR	Stolen card, pick-up
0.30	ERROR	Contact acquirer
0.31	ERROR	Exceeds withdrawal limit
0.32	ERROR	Original amount incorrect
0.33	ERROR	Expired card
0.34	SUCCESS	{none}
0.35	ERROR	Incorrect PIN
0.36	ERROR	Transaction not permitted to cardholder
0.37	ERROR	Transaction not permitted on terminal
0.38	ERROR	Suspected fraud
0.39	ERROR	Restricted card
0.40	ERROR	Exceeds withdrawal frequency
0.41	ERROR	Call acquirer security
0.42	ERROR	PIN tries exceeded
0.43	ERROR	Hard capture
0.44	ERROR	Cut-off in progress
0.45	ERROR	Issuer or switch inoperative
0.46	ERROR	Duplicate transaction
0.47	ERROR	System malfunction
0.48	ERROR	Wrong PIN, allowable number of PIN tries exceeded
0.49	ERROR	Time out
0.50	ERROR	Cryptographic failure

Amount	Status	Error Message
0.51	ERROR	Routing error
0.52	ERROR	Exceeds cash limit
0.53	ERROR	TVR check failure
0.54	ERROR	TVR configuration error
0.55	ERROR	Unacceptable PIN
0.56	ERROR	Cashback service not available
0.57	ERROR	Cash request exceeds Issuer limit
0.58	SUCCESS	{none}
0.59	SUCCESS	{none}
0.60	SUCCESS	{none}
0.61	SUCCESS	{none}
0.62	SUCCESS	{none}
0.63	SUCCESS	{none}
0.64	SUCCESS	{none}
0.65	SUCCESS	{none}
0.66	SUCCESS	{none}
0.67	SUCCESS	{none}
0.68	SUCCESS	{none}
0.69	SUCCESS	{none}
0.70	SUCCESS	{none}
0.71	SUCCESS	{none}
0.72	SUCCESS	{none}
0.73	SUCCESS	{none}
0.74	SUCCESS	{none}
0.75	SUCCESS	{none}
0.76	SUCCESS	{none}
0.77	SUCCESS	{none}
0.78	SUCCESS	{none}
0.79	SUCCESS	{none}
0.80	SUCCESS	{none}
0.81	SUCCESS	{none}
0.82	SUCCESS	{none}
0.83	SUCCESS	{none}
0.84	SUCCESS	{none}
0.85	SUCCESS	{none}
0.86	SUCCESS	{none}
0.87	SUCCESS	{none}
0.88	SUCCESS	{none}
0.89	SUCCESS	{none}
0.90	SUCCESS	{none}
0.91	SUCCESS	{none}
0.92	SUCCESS	{none}
0.93	ERROR	ERROR
0.94	ERROR	ERROR
0.95	ERROR	Communication Error
0.96	SUCCESS	{none}
0.97	SUCCESS	{none}
0.98	SUCCESS	{none}
0.99	SUCCESS	{none}

Appendix B Country States

The following table shows the codes for the US, Canadian and Mexican States used in the *customerDocumentState* parameter of the Session Token Request (section 1.1).

B.1 United States

State	Abbr	State	Abbr	Territories	Abbr
Alabama	AL	Montana	MT	American Samoa	AS
Alaska	AK	Nebraska	NE	Guam	GU
Arizona	AZ	Nevada	NV	Norther Mariana Islands	MP
Arkansas	AR	New Hampshire	NH	Puerto Rico	PR
California	CA	New Jersey	NJ	U.S. Virgin Islands	VI
Colorado	CO	New Mexico	NM		
Connecticut	CT	New York	NY		
Delaware	DE	North Carolina	NC		
District of Columbia	DC	North Dakota	ND		
Florida	FL	Ohio	OH		
Georgia	GA	Oklahoma	OK		
Hawaii	HI	Oregon	OR		
Idaho	ID	Pennsylvania	PA		
Illinois	IL	Rhode Island	RI		
Indiana	IN	South Carolina	SC		
Iowa	IA	South Dakota	SD		
Kansas	KS	Tennessee	TN		
Kentucky	KY	Texas	TX		
Louisiana	LA	Utah	UT		
Maine	ME	Vermont	VT		
Maryland	MD	Virginia	VA		
Massachusetts	MA	Washington	WA		
Michigan	MI	West Virginia	WV		
Minnesota	MN	Wisconsin	WI		
Mississippi	MS	Wyoming	WY		
Missouri	MO				

B.2 Canada

State	Abbr
Alberta	AB
British Columbia	BC
Manitoba	MB
New Brunswick	NB
Newfoundland and Labrador	NL
Northwest Territories	NT
Nova Scotia	NS
Nunavut	NU
Ontario	ON
Prince Edward Island	PE
Quebec	QC
Saskatchewan	SK
Yukon	YT

B.3 Mexico

State	Abbr
Aguascalientes	AG
Baja California	BJ
Baja California Sur	BS
Campeche	CP
Chiapas	CH
Chihuahua	CI
Coahuila	CU
Colima	CL
Distrito Federal	DF
Durango	DG
Guanajuato	GJ
Guerrero	GR
Hidalgo	HG
Jalisco	JA
Mexico	EM
Michoacán	MH
Morelos	MR
Nayarit	NA
Nuevo Leon	NL
Oaxaca	OA
Puebla	PU
Queretaro	QA
Quintana Roo	QR
San Luis Potosi	SL
Sinaloa	SI
Sonora	SO
Tabasco	TA
Tamaulipas	TM
Tlaxcala	TL
Veracruz	VZ
Yucatan	YC
Zacatecas	ZT

Appendix C customerBrowser Data Elements Definitions

All parameters are required if the *customerBrowser* object is included

Data Element	Data Type	Req	Description
browserAcceptHeader	String (2048)	Y	Exact content of the HTTP accept headers as sent to the merchant from the Cardholder's browser. **Value accepted**: If the total length of the accept header sent by the browser exceeds 2048 characters, the excess content will be truncated.
browserJavaEnabled	Boolean	Y	Is the Browser Java Enabled: Flag that represents the ability of the cardholder browser to execute Java. Value is returned from the navigator.javaEnabled property. **Values accepted**: false – Not supported true – Supported
browserJavascriptEnabled	Boolean	Y	Is the Browser JavaScript Enabled: Flag that represents the ability of the cardholder browser to execute JavaScript. **Values accepted**: false – Not supported true – Supported
browserLanguage	String (8)	Y	Browser Language: Value representing the browser language as defined in IETF BCP47. Returned from navigator.language property.
browserColorDepth	String (enum)	Y	Browser Colour Depth: Value representing the bit depth of the colour palette for displaying images, in bits per pixel. Obtained from Cardholder browser using the screen.colorDepth property. **Values accepted**: 1 = 1 bit 4 = 4 bits 8 = 8 bits 15 = 15 bits 16 = 16 bits 24 = 24 bits 32 = 32 bits 48 = 48 bits
browserScreenHeight	Integer (6)	Y	Browser Screen Height: Total height of the Cardholder's screen in pixels. Value is returned from the screen.height property.
browserScreenWidth	Integer (6)	Y	Browser Screen Width: Total width of the cardholder's screen in pixels. Value is returned from the screen.width property.
browserTZ	Integer (6)	Y	Browser Time Zone: Time-zone offset in minutes between UTC and the Cardholder browser local time. Value is returned from the getTimezoneOffset() method. Note that the offset is positive if the local time zone is behind UTC and negative if it is ahead. **Value accepted**: Example time zone offset values in minutes: If UTC -5 hours: 300 or +300 If UTC +5 hours: 300

Data Element	Data Type	Req	Description
challengeWindowSize	String (enum)	Y	Challenge Window Size: Dimensions of the challenge window that has been displayed to the Cardholder. The Issuer will reply with content that is formatted to appropriately render in this window to provide the best possible user experience. Preconfigured sizes are width X height in pixels of the window displayed in the Cardholder browser window. This is used only if a Challenge is required by the Issuer. **Values accepted**: 01 – 250 x 400 02 – 390 x 400 03 – 500 x 600 04 – 600 x 400 05 – Full screen

Appendix D sdkAppInfo Data Element Definitions

All parameters are required if the sdkAppInfo object is included

Data Element	Data Type	Req	Description
sdkAppId	String (36)	Y	SDK App Id: Universally unique ID created upon all installations of the 3DS Requestor App on a Consumer Device. This will be newly generated and stored by the 3DS SDK for each installation. **Value accepted**: Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions as long as the output meets specified requirements.
sdkEncryptedData	String (64000)	Y	JWE Object (represented as a string) that contains data encrypted by the SDK for the Directory Server to decrypt See Section 6.2.2.1 of EMVCo's "EMV 3-D Secure – Protocol and Core Functions Specification" for additional detail
sdkPublicKey	JWK Object (Max 256 Chars)	Y	Public key component of the ephemeral key pair generated by the 3DS SDK and used to establish session keys between the 3DS SDK and ACS See Section 6.2.3.1 of EMVCo's "EMV 3-D Secure – Protocol and Core Functions Specification" for additional detail
sdkMaxTimeout	Integer (2)	Y	Indicates maximum amount of time (in minutes) for all exchanges.
sdkReferenceNumber	String (32)	Y	Identifies the vendor and version for the 3DS SDK that is integrated in a 3DS Requestor App, assigned by EMVCo when the 3DS SDK is approved.
sdkTransactionId	String (26)	Y	Universally unique transaction identifier assigned by the 3DS SDK to identify a single transaction. Canonical format as defined in IETF RFC 4122. This may utilise any of the specified versions if the output meets specified requirements.
sdkInterface	String (enum)	Y	SDK Interface: Lists all of the SDK Interface types that the device supports for displaying specific challenge user interfaces within the SDK. **Values accepted**: 01 = Native 02 = HTML 03 = Both
sdkUiType	String (enum) Array	Y	SDK UI Type: Lists all UI types that the device supports for displaying specific challenge user interfaces within the SDK. **Values accepted for each *sdkInterface* value**: Native UI = 01–04 HTML UI = 01–05 **Note**: Currently, all SDKs need to support all UI Types. In the future, however, this may change (for example, smart watches may support a UI Type not yet defined by this specification). **Values accepted**: 01 = Text 02 = Single Select 03 = Multi Select 04 = OOB 05 = HTML Other (valid only for HTML UI)

Appendix E customerAccountInfo Data Elements Definitions

All parameters are optional, but should be supplied if the data is available to facilitate a Frictionless Flow

Data Element	Data Type	Req	Description
custAccAgeInd	String (enum)	N	Cardholder Account Age Indicator: Length of time that the cardholder has had the account with the merchant. **Values accepted:** 01 = No account (guest check-out) 02 = Created during this transaction 03 = Less than 30 days 04 = 30−60 days 05 = More than 60 days
custAccChange	String (8)	N	Cardholder Account Change: Date that the cardholder's account with the merchant was last changed, including Billing or Shipping address, new payment account, or new user(s) added. **Date format** = YYYYMMDD
custAccChangeInd	String (enum)	N	Cardholder Account Change Indicator: Length of time since the cardholder's account information with the merchant was last changed, including Billing or Shipping address, new payment account, or new user(s) added. **Values accepted:** 01 = Changed during this transaction 02 = Less than 30 days 03 = 30−60 days 04 = More than 60 days
custAccPwChange	String (8)	N	Cardholder Account Password Change: Date that cardholder's account with the merchant had a password change or account reset **Date format** = YYYYMMDD
custAccPwChangeInd	String (enum)	N	Indicates the length of time since the cardholder's account with the Merchant had a password change or account reset. **Values accepted:** 01 = No change 02 = Changed during this transaction 03 = Less than 30 days 04 = 30−60 days 05 = More than 60 days
custPurchaseCount	Integer (4)	N	Cardholder Account Purchase Count: Number of purchases with this cardholder account during the previous six months.
custProvisionAttemptsPerDay	Integer (3)	N	Number of Provisioning Attempts Per Day: Number of Add Card attempts in the last 24 hours.
custTxnActivityDay	Integer (3)	N	Number of Transactions Per Day: Number of transactions (successful and abandoned) for this cardholder account with the merchant across all payment accounts in the previous 24 hours.
custTxnActivityYear	Integer (3)	N	Number of Transactions Per Year: Number of transactions (successful and abandoned) for this cardholder account with the merchant across all payment accounts in the previous year.
custPaymentAccAge	Integer (8)	N	Payment Account Age: Date that the payment account was enrolled in the cardholder's account with the merchant. **Date format** = YYYYMMDD

Data Element	Data Type	Req	Description
custPaymentAccInd	String (enum)	N	Payment Account Age Indicator: Indicates the length of time that the payment account was enrolled in the cardholder's account with the merchant. **Values accepted:** 01 = No account (guest check-out) 02 = During this transaction 03 = Less than 30 days 04 = 30−60 days 05 = More than 60 days
custShipAddressUsage	String (8)	N	Shipping Address Usage: Date when the shipping address used for this transaction was first used with the merchant. **Date format** = YYYYMMDD
custShipAddressUsageInd	String (enum)	N	Shipping Address Usage Indicator: Indicates when the shipping address used for this transaction was first used with the merchant. **Values accepted:** 01 = This transaction 02 = Less than 30 days 03 = 30−60 days 04 = More than 60 days
custShipNameIndicator	String (enum)	N	Shipping Name Indicator: Indicates if the Cardholder Name on the account is identical to the shipping Name used for this transaction. **Values accepted:** 01 = Account Name identical to shipping Name 02 = Account Name different than shipping Name
custSuspiciousAccActivity	String (enum)	N	Suspicious Account Activity: Indicates whether the merchant has experienced suspicious activity (including previous fraud) on the cardholder account. **Values accepted:** 01 = No suspicious activity has been observed 02 = Suspicious activity has been observed

Appendix F merchantAuthInfo Data Elements Definitions

All parameters are required if the merchantAuthInfo object is included, except merchantAuthData, which is undefined in 3DS V2.x (See Description).

Data Element	Data Type	Req	Description
merchantAuthData	String (20000)	N	Merchant Authentication Data: Data that documents and supports a specific authentication process. For example, if *merchantAuthMethod* = 03, this element can carry information about the provider of the federated ID and related information. 06, this element can carry the FIDO attestation data (including the signature). 07, this element can carry FIDO Attestation data with the FIDO assurance data signed. 08, this element can carry the SRC assurance data. In the current version of the 3DS V2.x specification, this data element is not defined in detail, and therefore is optional. However, the intention is that for each merchant Authentication Method, this field should carry data that the ACS can use to verify the authentication process.
merchantAuthMethod	String (enum)	Y	Merchant Authentication Method: Mechanism used by the merchant to authenticate Cardholder. **Values accepted**: 01 = No merchant authentication occurred (i.e. cardholder "logged in" as guest) 02 = Login to the cardholder account in the merchant's system using merchant's own credentials 03 = Login to the cardholder account in the merchant's system using federated ID 04 = Login to the cardholder account in the merchant's system using issuer credentials 05 = Login to the cardholder account in the merchant's system using third-party authentication 06 = Login to the cardholder account in the merchant's system using FIDO Authenticator 07 = Login to the cardholder account in the merchant's system using FIDO Authenticator (FIDO assurance data signed) 08 = SRC Assurance Data **Netcetera Constraint**: Values '07' and '08' are only available when Netcetera initiates authentication with EMV 3DS 2.2.0 version or greater. In this instance, the *threeDSPreferredProtocolVersion* and *enforcethreeDSPreferredProtocolVersion* parameters should be set appropriately
merchantAuthTimestamp	String (12)	Y	Merchant Authentication Timestamp: Date and time in UTC of the cardholder authentication. Date format = YYYYMMDDHHMM

Appendix G merchantPriorAuthInfo Data Elements Definitions

All parameters are required if the merchantPriorAuthInfo object is included, except merchantPriorAuthData, which is undefined in 3DS V2.x (See Description)

Data Element	Data Type	Req	Description
merchantPriorAuthData	String (2048)	N	Merchant Prior Transaction Authentication Data: Data that documents and supports a specific authentication process. In the current version of the specification this data element is not defined in detail, however the intention is that for each Merchant Authentication Method, this field carry data that the ACS can use to verify the authentication process. In future versions of the specification, these details are expected to be included.
merchantPriorAuthMethod	String (enum)	N	Merchant Prior Transaction Authentication Method: Mechanism used by the merchant to previously authenticate the Cardholder **Values accepted**: 01 = Frictionless authentication occurred 02 = Cardholder challenge occurred 03 = AVS verified 04 = Other Issuer methods
merchantPriorAuthTimestamp	String (12)	N	Merchant Prior Transaction Authentication Timestamp: Date and time in UTC of the prior cardholder authentication. Date format = YYYYMMDDHHMM
merchantPriorRef	String (36)	N	Merchant Prior Transaction Reference: This data element provides additional information to the Issuer to determine the best approach for handing a request. This data element contains the original *merchantTxId* for a prior authenticated transaction (for example, the first recurring transaction that was authenticated with the cardholder).

Appendix H merchantRiskIndicator Data Elements Definitions

All parameters are optional, but should be supplied if the data is available to facilitate a Frictionless Flow

Data Element	Data Type	Req	Description
deliveryTimeframe	String (enum)	N	Delivery Timeframe: Indicates the merchandise delivery timeframe. **Values accepted:** 01 = Electronic Delivery 02 = Same day shipping 03 = Overnight shipping 04 = Two-day or more shipping
giftCardAmount	BigDecimal (15.2 or 15.3)	N	Gift Card Amount: For prepaid or gift card purchase, the purchase amount total of prepaid or gift card(s)
giftCardCount	Integer (2)	N	Gift Card Count: For prepaid or gift card purchase, total count of individual prepaid or gift cards/codes purchased.
giftCardCurr	String (3)	N	Gift Card Currency: For prepaid or gift card purchase, the ISO alpha-3 code for the currency as defined in the ISO 4217 standard
preOrderDate	String (8)	N	Pre-Order Date: For a pre-ordered purchase, the expected date that the merchandise will be available. **Date format** = YYYYMMDD
preOrderPurchaseInd	String (enum)	N	Pre-Order Purchase Indicator: Indicates if the Cardholder is placing an order for merchandise with a future availability or release date. **Values accepted:** 01 = Merchandise available 02 = Future availability
reorderItemsInd	String (enum)	N	Reorder Items Indicator: Indicates whether the cardholder is reordering previously purchased merchandise. **Values accepted:** 01 = First time ordered 02 = Reordered
shipIndicator	String (enum)	N	Shipping Indicator: Indicates shipping method chosen for the transaction. Merchants must choose the Shipping Indicator code that most accurately describes the cardholder's specific transaction, not their general business. If one or more items are included in the sale, use the Shipping Indicator code for the physical goods, or if all digital goods, use the Shipping Indicator code that describes the most expensive item. **Values accepted:** 01 = Ship to cardholder's billing address 02 = Ship to another verified address on file with merchant 03 = Ship to address that is different than the cardholder's billing address 04 = "Ship to Store" / Pick-up at local store (Store address shall be populated in shipping address fields) 05 = Digital goods (includes online services, electronic gift cards and redemption codes) 06 = Travel and Event tickets, not shipped 07 = Other (for example, Gaming, digital services not shipped, emedia subscriptions, etc.)