API Operations Specification

Version 2.5

Version Date: 25.01.2021

Document Purpose

This specification provides merchant developers with the necessary information to integrate their sales systems with the eService Gateway Application Programming Interfaces (APIs).

Intended Audience

This API Operations Overview is intended enable planning and integration with the eService Gateway APIs by:

- Merchant business and technology staff
- Shopping Cart Plugin providers

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 eService 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 6.1). 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 6.1), 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.

Appendix A The document defines the external interfaces to the eService Gateway necessary to:

- Request payment card tokens
- Submit authorisation transactions
- Submit purchases/sales transactions
- Capture (full or partial) funds from customers' accounts as a result of successful authorisation transactions
- Void authorised payment requests
- Refund (full or partial) purchases (captured payments)
- Request transactions statuses
- Integrate PCI Compliant Payment Forms
- Receive Transaction Call results

The reader should have the knowledge and understanding of the payments industry processes, and the role of the payment processer (the eService Gateway) in those payment processes.

Contents

1 Merchant Integration Methods

Merchants' integration methods are agreed with their Acquirer during their on-boarding process. It is essential that the merchant informs the eService Gateway about which integration method will be employed. This is to assist with correctly configuring the merchant account in the eService Gateway and for future support purposes.

The eService Gateway supports three integration methods:

1.1 Hosted Payment Page Integration

Hosted Payment Page Integration is designed for the merchant that wants to focus on providing an ecommerce web presence to offer their goods or services to their customers, and not concern themselves with the complexities of managing PCI Compliant environments that are required to manage sensitive customer payment card information in secure, often encrypted environments. Features include:

The primary feature of this integration method is that the eService Gateway manages a Level-1 PCI Compliant environment that is certified and regularly audited. The merchant will integrate the eService Gateway's own hosted payment form into their checkout pages. The payment form is loaded in such a way that all the processing takes place on the eService Gateway servers. The payment card data will not be exposed to the merchant's system.

The eService Gateway can also provide other Alternate Payment Methods (APM), non-card payment methods to merchants. The eService Gateway manages all the integrations with the APM providers, returning transaction results to the merchants' systems. In some, instances, the merchant will have been required to register accounts with these APM providers, and to supply their credentials to the eService Gateway, where the data is stored securely and confidentially.

1.2 Shopping Cart Plugins

Shopping Cart Plugins simplify the creation and building of a merchant ecommerce web pages by providing streamlined application that integrates with a merchant's website.

The primary feature of this integration method is that the eService Gateway payment processing is already incorporated into the Shopping Cart Plugin. The merchant integrates with the best suited to their purposes. Shopping Cart Plugins reduce the requirement for merchants to understand the complexities of web design and development by supplying a ready-made method of presenting their goods and services to their customers and enabling the taking of payments, card or alternate methods.

The eService Gateway provides its own shopping cart plugins and is integrated into many other third-party providers.

The reader should refer to the Shopping Cart Plugins supplier for the integration methods.

2 API Operations Overview

This section contains the list and descriptions of the API Operations that are available in the eService Gateway.

2.1 AUTH/PURCHASE

The AUTH/PURCHASE API Operation combines the Authorise and Purchase actions into one API Operation, due to the similarities between them.

The AUTH/PURCHASE API Operation processes merchant's customers' payments taken either in the merchant's own payment form or in the eService Gateway Hosted Payment Page.

Depending on the configuration of the merchant's payment form or the eService Gateway Hosted Payment Page, the AUTH/PURCHASE API Operation will cater for payment card and non-payment card payment processing. Some non-payment card method, also known as Alternate Payment Methods (APMs) have their own API Operations that bypass or do not require processing through the eService Gateway.

eService Gateway Hosted Payment Page Integration merchants will loaded the payment pages that are preconfigured in the eService Gateway. When the merchant's customer selects a payment method the eService Gateway will react appropriately.

1. The Hosted Payment Page Integration merchants will not send an Authorise/Purchase Request on receipt of the Session Token. Instead, as the Action Request, the merchant will:
    a. Send the Session Token Request, and receive the Session Token Response
    b. Send a Load Payment Form Request
    c. The eService Gateway Hosted Payment Page loads into the merchant's web page (a parameter in the Session Token Request)
    d. The customer inputs their payment card data or selects an APM
    e. The eService Gateway Hosted Payment Page processes the payment as selected and returns the appropriate Authorise/Purchase Response and Transaction Result Call
    f. The merchant's webpage and system will receive and process the response as required

2.2 REFUND

The REFUND API Operation is available to Hosted Payment Page Integration and Shopping Cart Plugins merchants. The functionality is also available in the eService Gateway Back-Office (section 4).

The REFUND API Operation should not be a merchant customer-facing function. It is used either:

- By Hosted Payment Page Integration merchants who have built their own back-office application
- By Shopping Cart Plugins, where the functionality has been built into the plugin

The REFUND API Operation can be performed on all Purchase transactions and captured Authorise transactions.

The eService Gateway offers full or partial refunds. More than one partial refund can be performed up to the full amount of the original transaction amount.

The REFUND API Operation is described in section 7 - REFUND API Operation

2.3 VOID

The VOID API Operation is available to Hosted Payment Page Integration and Shopping Cart Plugins merchants. The functionality is also available in the eService Gateway Back-Office (section 4).

The VOID API Operation should not be a merchant customer-facing function. It is used either:

- By Hosted Payment Page Integration merchants who have built their own back-office application.

- By Shopping Cart Plugins, where the functionality has been built into the plugin

The VOID API Operation can be performed on unsettled Purchase transactions and un-captured Authorise transactions.

The VOID API Operation is described in section 8 - VOID API Operation

2.4 CAPTURE

The CAPTURE API Operation is available to Hosted Payment Page Integration and Shopping Cart Plugins merchants. The functionality is also available in the eService Gateway Back-Office (section 4).

The CAPTURE API Operation should not be a merchant customer-facing function. It is used either:

- By Hosted Payment Page Integration merchants who have built their own back-office application
- By Shopping Cart Plugins, where the functionality has been built into the plugin

The CAPTURE API Operation can be performed on un-captured Authorise transactions.

The eService Gateway offers full or partial captures. Currently, only one partial capture can be performed on an Authorise transaction, where the residual amount is released back to the customer's account.

The CAPTURE API Operation is described in section 9 - CAPTURE API Operation

2.5 TRANSACTION RESULT CALL

The TRANSACTION RESULT CALL is not an API Operation, but a result of the above API Operations.

The TRANSACTION RESULT CALL is a server-to-server call between the eService Gateway and the merchant's server. In the all the above API Operations the *merchantNotificationUrl* parameter tells the eService Gateway where to send the TRANSACTION RESULT CALL.

If this parameter is left empty or not included in the API Call from the merchant a TRANSACTION RESULT CALL is not sent by the eService Gateway.

The TRANSACTION RESULT CALL is described in section 10 - Transaction Result Call

2.6 GET STATUS

The GET STATUS API Operation is a utility available to the merchants.

The GET STATUS API Operation allows the merchant to send a transaction reference to the eService Gateway to check the status of the transaction in the eService Gateway.

The Operation can be used to reconcile transactions statuses between the merchant's transactions database and the eService Gateway database.

The GET STATUS API Operation is described in section 11 - GET STATUS API Operation

2.7 GET AVAILABLE PAYMENT SOLUTIONS

The GET AVAILABLE PAYMENT SOLUTIONS API Operation is a utility to the merchants.

The GET AVAILABLE PAYMENT SOLUTIONS API Operation allows the merchant to dynamically query the eService Gateway as to which payment solutions are available to a merchant's customer depending on the currency, country and merchant's brand.

The GET AVAILABLE PAYMENT SOLUTIONS API Operation is described in section 12 - GET AVAILABLE PAYMENT SOLUTIONS API Operation

3 Gateway Interface

3.1 Addresses

3.1.1 User Acceptance Testing Addresses

Session Token URL: https://apiuat.test.secure.eservice.com.pl/token
Action Request URL: https://apiuat.test.secure.eservice.com.pl/payments
Payment Form URL: https://cashierui-apiuat.test.secure.eservice.com.pl/
Back-Office URL: https://backofficeui-apiuat.test.secure.eservice.com.pl/

3.1.2 Production Addresses

Session Token URL: https://api.secure.eservice.com.pl/token
Action Request URL: https://api.secure.eservice.com.pl/payments
Payment Form URL: https://cashierui-api.secure.eservice.com.pl/
Back-Office URL: https://backofficeui-api.secure.eservice.com.pl/

3.2 HTTP Specification

- Protocol: https
- Method: POST
- Content Type: application/x-www-form-urlencoded

3.2.1 Example HTTP Request

- POST: https://api.secure.eservice.com.pl/token
- Content-Type: application/x-www-form-urlencoded
- Content-Length: 415

3.2.2 POST data

merchantId=160001&action=PURCHASE&password=password&allowOriginUrl=www.merchantsite.com&timestamp=1459767453376&channel=ECOM&userDevice=DESKTOP&amount=25.96&currency=GBP&country=DE&paymentSolutionId=500&customerId=9876543&brandId=670&merchantNotificationUrl=https%3A%2F%2Fwww.posttestserver.com%2Fpost.php%2FipgTesting%3Fdir%3DJCTesting&merchantLandingPageUrl=https://www.merchantsite.com%2FlandingPage

4 eService Gateway Back-Office

The eService Gateway Back-Office compliments the API Operations by providing some API Operations functionality, namely:

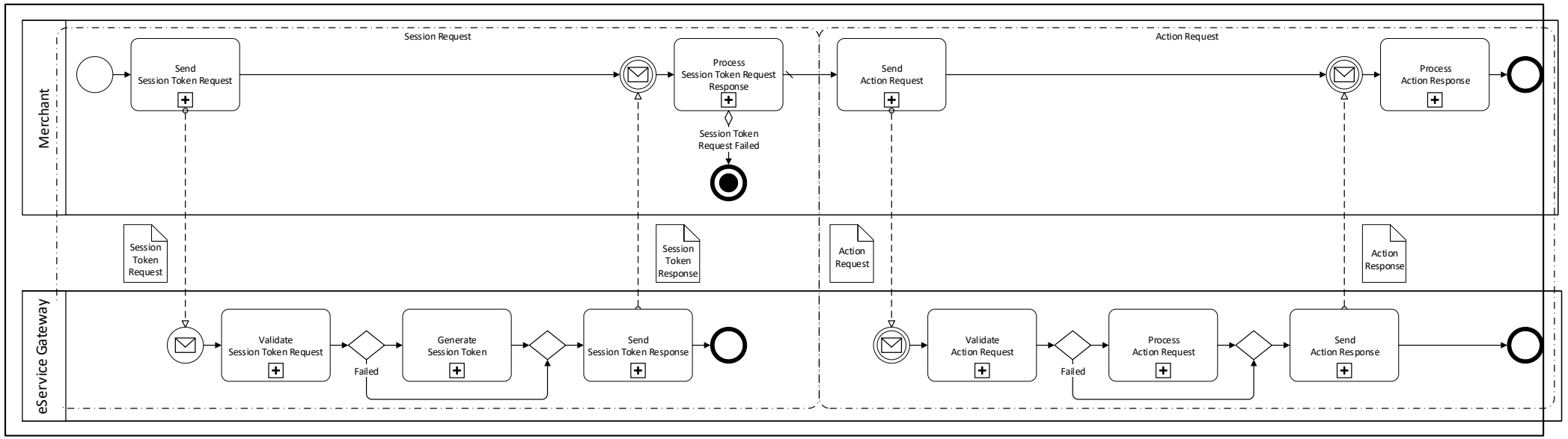
- Transaction Management — provides a list of all customers' transactions that can be filtered, sorted and searched; from the list a transaction can be selected to show the full detail
- Refund — both full and partial on the initial Purchase transaction amount; multiple refunds can be performed on a single transaction up to the full amount
- Void — for both Authorise and Purchase transactions
- Capture — both full and partial on the initial Authorise transaction amount; multiple captures are not yet supported
- Summary Reports and Detailed Reports — that show summary and detailed reports of the transaction over time

The above functionality can be replicated by the merchants' systems, if required, by using the API Operations or managing the data received from their customers and the eService Gateway. The eService Gateway Back-Office provides for an initial or permanent solution to customer transaction management.

5 API Operations Overview

5.1 Process Overview

Shown below is a generic view of how all eService Gateway API processes operate. The primary feature to note is that each API Operation has two components: the Session Token Request that authenticates the merchant system in the eService Gateway before the Action Request can be processed by the eService Gateway.



5.2 Process

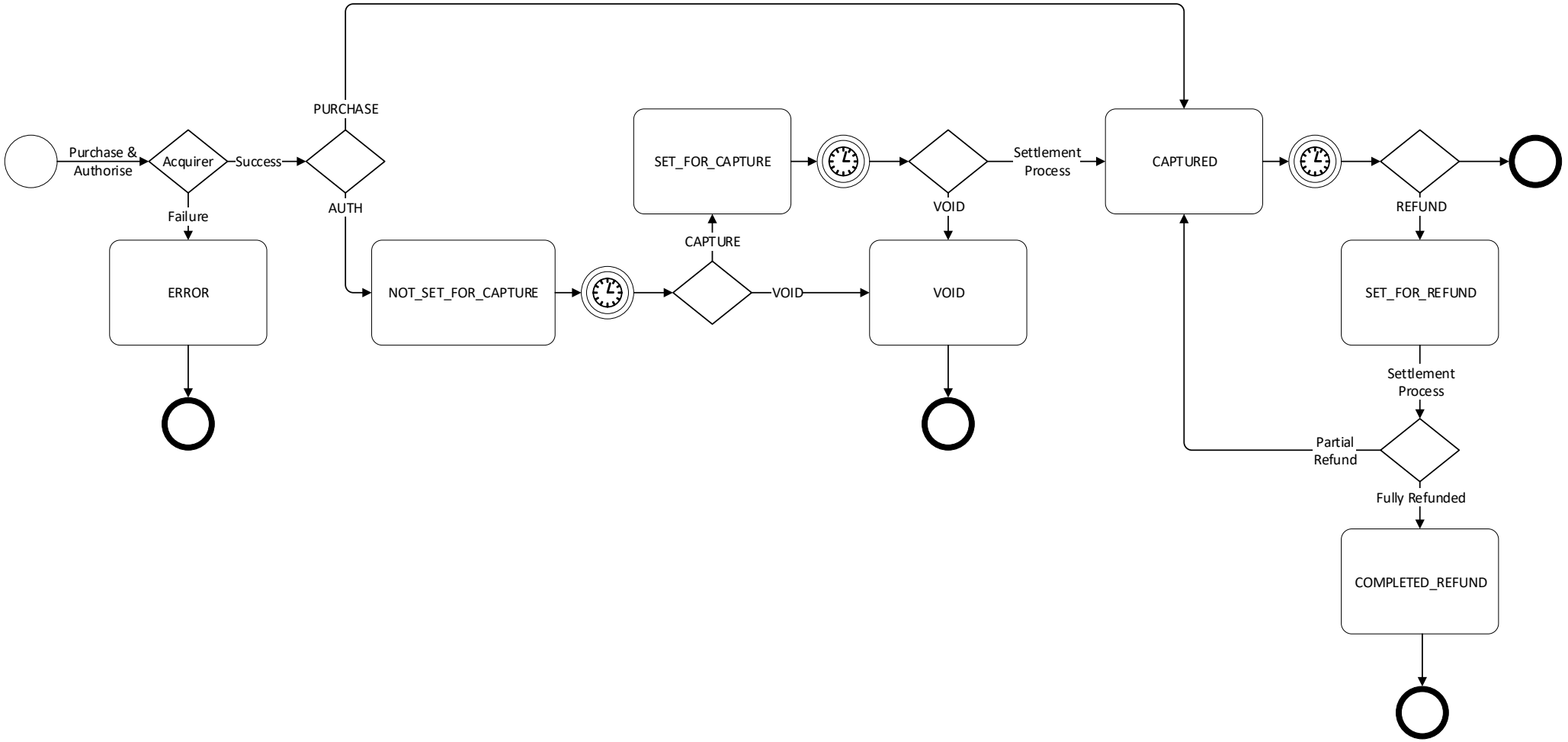
1. The merchant system sends the appropriate Session Token Request for the API Operation to the eService Gateway
2. The eService Gateway validates the Session Token Request and authenticates the merchant
    a. If the validation or authentication fails:
        i. The eService Gateway returns a Session Token Response – Not Processed to the merchant system
        ii. The merchant system must process the error
        iii. **The Process Terminates Here**
        iv. The merchant must rectify the issues and submit a new Session Token Request
    b. If the validation and authentication succeed:
        i. The eService Gateway generates a Session Token
        ii. The eService Gateway returns a Session Token Response –Processed to the merchant system that contains the Session Token in the *token* parameter
3. The merchant system sends the required Action Request for the API Operation to the eService Gateway
4. The eService Gateway validates the Action Request and authentications the Action Request to the *action* parameter
    a. If the validation or authentication fails:
        i. The eService Gateway returns an Action Response – Not Processed to the merchant system
        ii. The merchant system must process the error
        iii. The merchant must rectify the issues and submit a new Session Token Request, i.e. restart the process from the beginning
    b. If the validation and authentication succeed:
        i. The eService Gateway processes the Action Request
        ii. The eService Gateway returns an Action Response – Processed to the merchant system that contains the results of the processing
        The Action Response – Processed may also contain errors in the *errors* parameter. These are errors from the payment transaction process, not the internal eService Gateway processes. The merchant system must react appropriately
        For some API Operations a Transaction Result Call will be sent to the merchant's servers, provided in the *merchantNotificationUrl* parameter.

5.3 Transaction Statuses

Payment Transactions in the eService Gateway are acted upon by the API Operations during the payments process. At the end of operation, the transaction acquires a status, provided the operation process ended correctly. If the API Operation did not process correctly, there is no change to the transaction's status.

All transactions are created by the AUTH/PURCHASE API Operation (see section 2.1).

The following diagram shows the status flow of a transaction – statuses are the boxes, the operations that act on the transaction are the connectors:



6 AUTH/PURCHASE API Operation – Hosted Payment Page Integration

6.1 Session Token Request

6.1.1 Format

POST Request

6.1.2 Definition

Parameter	Data Type	Mandatory	Description
Security Data Mandatory to identify the merchant in the eService Gateway			
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String (64)	Y	The merchant's password in the eService Gateway provided at on-boarding
Transaction Data The Transaction Data defines the type of transaction the merchant is requesting the eService 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" or "PURCHASE"
firstTimeTransaction	Boolean	N	A flag to indicate if the transaction is the customer's first. This affects the behaviour of the 3D Secure processing. **Note**: if a *customerId* value is not provided, the eService Gateway will always treat the transaction as a first-time transaction for the customer
timestamp	Integer (13)	Y	Milliseconds since 1970-01-01 00:00:00
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
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 (see Section 6.4) This will usually be the URL of the customer's browser. Cross-Origin Resource Sharing (CORS) headers will allow only this origin

Parameter	Data Type	Mandatory	Description
merchantNotificationUrl	String (200)	Y	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 eService 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
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')			
paymentSolutionId	Integer (18)	N	The eService Gateway Payment Solution Identifier See section 12 - GET AVAILABLE PAYMENT SOLUTIONS API Operation for valid values
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 eService 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 eService 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 eService Gateway Brand Id for the merchant's goods or services supplied at on-boarding If not provided the merchant's default eService Gateway Brand Id will be used
freeText	String (200)	N	A free text field for use by the merchant that is returned in the Transaction Result Call (see 10 - Transaction Result Call)
customParameter1Or ... customParameter20Or	String (50)	N	20 Text Fields that used by merchants to reconcile transactions performed through mobile applications with results from the acquirer.
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

Parameter	Data Type	Mandatory	Description
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
userAgent	String (1024)	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
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.
language	String (enum)	N	The ISO alpha-2 language code, as defined in ISO 639-1 standard, for the language to be used in the Hosted Payment Page, when loaded to the merchant's webpage. • If a supported language code is provided, the language translation will be provided • If not provided or an unsupported language code is provided, the merchant's default language is used [Please consult your eCommerce Support Team for currently supported languages]
Transaction Amount Data Transaction Amount Data provides the values of the sale.			

Parameter	Data Type	Mandatory	Description
amount	BigDecimal (15.2 or 15.3)	Y	The total transaction amount, including tax, shipping, surcharge and discount amounts. If action = "AUTH" or "PURCHASE", this must be > 0.00 See Appendix C - 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)
shippingAmount	BigDecimal (15.2 or 15.3)	N	Shipping amount
chargeAmount	BigDecimal (15.2 or 15.3)	N	Surcharge amount
discountAmount	BigDecimal (15.2 or 15.3)	N	Discount amount
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 eService Gateway Back-Office Transactions Lists. **Conditional Parameters:** 3DS V2.x requires these parameters "unless market or regional mandate restricts sending this information". Therefore it is the merchant's responsibility to assess whether they are able or not able to send this information. 'Market or regional mandate' also covers situations where the merchant's own processes do not require this data to be captured, as well as for regulatory restrictions such as GDPR. **However, it is highly recommended, if possible, to send this data, if it is available, to enable card issuers to immediately authenticate a transaction – Frictionless Flow** Enabling a Frictionless Flow is not solely dependent on these parameters, but the issuers' decision are enabled with more information			
customerFirstName	String (50)	C	First name of the customer **Condition:** See above statement
customerLastName	String (100)	C	Last name, surname or family name of the customer **Condition:** See above statement

Parameter	Data Type	Mandatory	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)	C	Customer email address **Condition:** See above statement
customerPhone	String (100)	C	Customer phone number **Condition:** See above statement
customerDocumentType	String (enum)	N	Type of document used to confirm the customer's identification eService Gateway accepted values: • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UNIQUE_TAXPAYER_REFERENCE • OTHER
customerDocumentNumber	String (30)	C	Customer document number **Condition**: Mandatory if *customerDocumentType* provided
Payer Data The payer data is required by some regions or payment services 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 eService 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 eService Gateway accepted values: • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UNIQUE_TAXPAYER_REFERENCE OTHER

Parameter	Data Type	Mandatory	Description
payerDocumentNumber	String (30)	C	Payer document number, if the Payee is different to the Customer **Condition**: Mandatory if *payerDocumentType* provided
payerCustomerId	String (20)	N	Customer identifier of the payee in the merchant's system
Customer Account Data with the Merchant Customer Account Data is used in the eService 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)	C	Customer identifier in the merchant system, or the value generated by the eService 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. • Mandatory for payment cards method • Optional for alternative payment methods If the parameter is omitted or no value is provided, the eService Gateway will generate a value, which will be stored internally against the payment method and returned in the Auth/Purchase Response – Processed (section 6.5)
merchantReference	String (200)	N	Merchant's supplementary information about customer Note: this information is only stored in the eService 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:** • Used in the 3DS V2.x Authentication process as part of the customerAccountInfo • Used for reporting and in some risk tools where required

Parameter	Data Type	Mandatory	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 D - customerAccountInfo Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Mandatory	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 **Conditional Parameters:** 3DS V2.x requires these parameters "unless market or regional mandate restricts sending this information". Therefore it is the merchant's responsibility to assess whether they are able or not able to send this information. 'Market or regional mandate' also covers situations where the merchant's own processes do not require this data to be captured, as well as for regulatory restrictions such as GDPR. **However, it is highly recommended, if possible, to send this data, if it is available, to enable card issuers to immediately authenticate a transaction – Frictionless Flow** Enabling a Frictionless Flow is not solely dependent on these parameters, but the issuers' decision are enabled with more information			
customerAddressHouseName	String (50)	C	Customer correspondence address house name **Condition:** See above statement
customerAddressHouseNumber	String (5)	C	Customer correspondence address house number **Condition:** See above statement
customerAddressFlat	String (5)	C	Customer correspondence address flat **Condition:** See above statement
customerAddressStreet	String (50)	C	Customer correspondence address street **Condition:** See above statement
customerAddressCity	String (50)	C	Customer correspondence address city **Condition:** See above statement
customerAddressDistrict	String (50)	N	Customer correspondence address district

Parameter	Data Type	Mandatory	Description
customerAddressPostalCode	String (30)	C	Customer correspondence address postal code **Condition:** See above statement
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:** See above statement
customerAddressState	String (3)	C	Customer correspondence address state, county or province: The ISO alpha-3 code as defined in the ISO 3166-2 standard **Condition:** See above statement
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
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 (2)	N	Customer billing address country The ISO alpha-2 code as defined in the ISO 3166 standard
customerBillingAddressState	String (3)	N	Customer billing address state The ISO alpha-3 code as defined in the ISO 3166-2 standard
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

Parameter	Data Type	Mandatory	Description
customerShippingAddressCountry	String (2)	N	Customer shipping address country The ISO alpha-2 code as defined in the ISO 3166 standard
customerShippingAddressState	String (3)	N	Customer shipping address state, county or province The ISO alpha-3 code as defined in the ISO 3166-2 standard
customerShippingAddressPhone	String (100)	N	Customer shipping address phone
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 E - merchantAuthInfo Data Elements Definitions for the data elements' definitions.

Parameter	Data Type	Mandatory	Description

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 F - merchantPriorAuthInfo Data Elements Definitions for the data elements' definitions.

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 G - merchantRiskIndicator Data Elements Definitions for the data elements' definitions.

6.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&merchantTxId=XYZ123456789ABC&allowOriginUrl=www.merchantsite.com&action=AUTH&timestamp=1249751864238&customerId=ABD123&brandId=987654321&channel=ECOM&userDevice=DESKTOP&amount=120&taxAmount=10&shippingAmount=15&chargeAmount=5&discountAmount=10&currency=GBP&country=GB&paymentSolutionId=500&language=en&merchantNotificationUrl=www.merchantsite.com&merchantLandingPageUrl=www.merchant.com&customerFirstName=John&customerLastName=Smith&customerSex=M&customerDateOfBirth0=01/01/1999&customerRegistrationDate=01/01/2017&customerEmail=john.smith@email.com&customerPhone=079525551234&customerIPAddress=111.111.111.111&customerAddressHouseName=House+Name&customerAddressHouseNumber=1&customerAddressFlat=3&customerAddressStreet=Street+Name&customerAddressCity=London&customerAddressDistrict=Mayfair&customerAddressPostalCode=W1A+A11&customerAddressCountry=United+Kingdom&customerAddressState=London&customerAddressPhone=00442025551234

6.2 Session Token Response - Processed

6.2.1 Format

JSON

6.2.2 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 6.1)
token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Auth/Purchase (see Section 6.4) 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

6.2.3 Example

```
{
    "result":"success",
    "merchantId":1111111,
    "token":"abcde12345abcde12345",
    "resultId":"fghij67890fghij67890"
}
```

6.3 Session Token Response – Not Processed

6.3.1 Format

JSON

6.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 6.1)
errors	String Array	List of issues

6.3.3 Example

```
{
    "result":"failure",
    "merchantId":1111111,
    "errors": [
      {
          "messageCode": "This field is required in [REQUEST]",
          "fieldName": "password"
      }
    ]
}
```

6.4 Auth/Purchase Request

6.4.1 Format

POST Request

6.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 6.1).
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 6.2).
freeText	String (200)	N	Merchant' free text or comments received in the Session Token Request (section 6.1).
customerIPAddress	String (39)	N	Customer IP address from where purchase is made. Only IPv4 supported. This value will be ignored if it was received in the customerIPAddress field in the Session Token Request (section 6.1).
paymentSolutionId	Integer (18)	C	Payment solution identifier in the eService Gateway. Mandatory, if not received in the Session Token Request (section 6.1), otherwise ignored.
integrationMode	String (enum)	C	The method that the merchant's webpage will use to load the Hosted Payment Page, accepted values: Inject HostedPayPage Standalone

6.5 Auth/Purchase Response – Processed

6.5.1 Format

JSON

6.5.2 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 6.1)
merchantTxId	String (50)	The merchant's reference for the transaction provided in the Session Token Request (section 6.1) or that generated by the eService Gateway
txId	Integer (18)	The unique identifier for the transaction in the eService 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 6.1)
currency	String (enum)	The transaction ISO alpha-3 currency code as defined in the **ISO 4217 standard**, provided in the Session Token Request (section 6.1)
customerId	String (20)	The customer identifier provided in the Session Token Request (section 6.1), or that generated by the eService Gateway
action	String (enum)	Action executed as provided in the Session Token Request (section 6.1) ("AUTH" or "PURCHASE")
pan	String (100)	The customer account value/number used in the transaction

Parameter	Data Type	Description
brandId	Integer (18)	The *brandId* value received in Session Token Request (section 6.1), or the default value used by the eService Gateway, if not provided
paymentSolutionId	Integer (18)	The *paymentSolutionId* value received in the Session Token Request (section 6.1)
freeText	String (200)	{not used for Hosted Payment Page or Shopping Cart Plugins Integration Merchants}
language	String (enum)	The ISO alpha-2 language code provided in the Session Token Request (section 6.1)
acquirerAmount	BigDecimal (15.2 or 15.3)	Amount processed by payment acquirer. May be different than the *amount* in the Session Token Request (section 6.1)
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 6.1)
paymentSolutionDetails	JSON block	For payment cards only: the Transaction Authorisation Code received from the acquirer, format: {"authCode":""}
status	String (enum)	The status of the transaction in the eService Gateway: Status: NOT_SET_FOR_CAPTURE — Condition: If "AUTH" successful Status: SET_FOR_CAPTURE — Condition: If "PURCHASE" successful Status: DECLINED — Condition: If "AUTH" or "PURCHASE" was declined/refused Status: ERROR — Condition: If an error was returned by the payment process
errors	String (400)	Any errors that occurred during the successful processing of a transaction
customParameter1Or ... customParameter20Or	String (50)	The original 20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation
customParameter1 ... customParameter20	String (50)	20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation, with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.

6.5.3 Example

{"result":"success","merchantId":111111,"merchantTxId":"abc123","txId":"123","acquirerTxId":"0009312","amount":12.50,"currency":"GBP","customerId":"mgn456","action":"PURCHASE","pan":"45ae201ghy23498FjMj701","brandId":3,"paymentSolutionId":500,"freeText":"Added+10%+discount+on+the+item","language":"en","acquirerAmount":16.7,"acquirerCurrency":"EUR","paymentSolutionDetails":{"authCode":"1234"},"status":"NOT_SET_FOR_CAPTURE"}

6.6 Auth/Purchase Response – Not Processed

6.6.1 Format

JSON

6.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"

merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 6.1)
merchantTxId	String (50)	The merchant's reference for the transaction provided in the Session Token Request (section 6.1) or that generated by the eService Gateway
txId	Integer (18)	The unique identifier for the transaction in the eService Gateway
errors	String Array	List of errors

Example


```
{
        "result":"failure",
        "merchantId":1231231,
        "merchantTxId":"abc-123",
        "txId":123,
        "errors": [
            {
                "messageCode": "This field is required in [REQUEST]",
                "fieldName": "password"
            }
        ]
}
```

7 REFUND API Operation

7.1 Session Token Request

7.1.1 Format

POST Request

7.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer(18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String(64)	Y	The merchant's password to the eService Gateway provided at on-boarding
action	String (enum)	Y	Must be "REFUND"
timestamp	Integer(18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String(253)	Y	The merchant's URL that will make the Refund Request (section 7.4) Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
originalTxId	Integer(18)	N	The eService Gateway transaction Id of the transaction to be refunded This will have been returned in the *txId* field of the *Auth/Purchase Response – Processed* (see sections 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
originalMerchantTxId	String(50)	Y	The merchant's original transaction identifier of the transaction to be refunded, that was provided in the *merchantTxId* field of the Auth/Purchase *Session Token Request* and *Auth/Purchase Request* (see sections 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
agentId	String(18)	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
amount	BigDecimal (10.2 or 10.3) BigDecimal (15.2 or 15.3)	Y	The amount to refund Can be less than or equal to the original transaction amount Cannot be more than the original transaction amount

7.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&originalTxId=1234&originalMerchantTxId=XYZ123ABC&allowOriginUrl=www.merchantsite.com&action=REFUND&timestamp=1249751864238&agentId=john04&amount=120

7.2 Session Token Response - Processed

7.2.1 Format

JSON

7.2.2 Definition

Parameter	Data Type	Description
result	String (enum)	Will always be “success”
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 7.1)
token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Refund Request (see Section 7.4) 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

7.2.3 Example

{"result":"success","merchantId":1111111,"token":"fghij67890fghij67890",”resultId”:”fghij67890fghij67890”}

7.3 Session Token Response – Not Processed

7.3.1 Format

JSON

7.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “failure”
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 7.1)
errors	String Array	List of issues

9.3.3 Example

{"result":"failure","merchantId":1111111,"errors":["Access denied"]}

7.4 Refund Request

7.4.1 Format

POST Request

7.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 7.1)
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 7.2)

7.4.3 Example

merchantId=1111111&token=fghij67890fghij67890

7.5 Refund Response – Processed

7.5.1 Format

JSON

7.5.2 Definition

Parameter/Label	Data Type	Description
result	String (40)	Will always be "success"
merchantId	Integer (18)	The *merchantId* value sent in the Session Token Request (section 7.1)
action	String (enum)	Will always be "REFUND"
originalMerchantTxId	String (50)	The merchant transaction identifier of the transaction that was refunded
originalTxId	Integer (18)	The eService Gateway transaction Id of the transaction refunded sent in the Session Token Request (section 7.1)
txId	Integer (18)	The unique identifier for the refund transaction in the eService Gateway
amount	BigDecimal (15.2 or 15.3)	The transaction amount refunded sent in the Session Token Request (section 7.1)
currency	String (enum)	The transaction ISO alpha-3 currency code, as defined in the ISO 4217 standard, used in the original and refund transactions
customerId	String (20)	The eService Gateway customer identifier used in the original and refund transactions
pan	String (100)	The customer account value/number used in the original and the refund transactions
brandId	Integer (18)	The *brandId* value for the original and the refund transaction
paymentSolutionId	Integer (18)	The *paymentSolutionId* value used in the original and refund transactions
status	String (enum)	The status of the transaction in eService Gateway: Status: SET_FOR_REFUND — Condition: If "REFUND" successful Status: ERROR — Condition: If an error was returned by the payment process
errors	String (400)	Any errors that occurred during the successful processing of a transaction
customParameter1Or ... customParameter20Or	String (50)	The original 20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation
customParameter1 ... customParameter20	String (50)	20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation, with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.

7.5.3 Example

{"result":"success","merchantId":111111,"originalMerchantTxId":"abc123","originalTxId":123,"txId":546, "amount":12.50,"currency":"GBP","customerId":"mgn456","action":"REFUND","pan":"45ae201ghy23498 FjMj701","brandId":3,"paymentSolutionId":500,"status":"SET_FOR_CAPTURE"}

7.6 Refund Response – Not Processed

7.6.1 Format

JSON

7.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"

merchantid	Integer (18)	The *merchantId* value received in the Session Token Request (section 7.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the eService Gateway, if not received in the Session Token Request (section 7.1)
txId	Integer (18)	The unique identifier for the refund transaction attempt in the eService Gateway
errors	String Array	List of errors

7.6.3 Example

{"result":"failure","merchantId":1111111,"merchantTxId":"abc123","txId":123,"errors":["communications failure"]}

8 VOID API Operation

8.1 Session Token Request

8.1.1 Format

POST Request

8.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer(18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String(64)	Y	The merchant's password to the eService Gateway provided at on-boarding
action	String (enum)	Y	Must be "VOID"
timestamp	Integer(18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String(253)	Y	The merchant's URL that will make the Void Request (section 8.4) Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
originalTxId	Integer(18)	N	The eService Gateway transaction Id of the transaction to be voided This will have been returned in the *txId* field of the *Auth/Purchase Response – Processed* (see sections and 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
originalMerchantTxId	String(50)	Y	The merchant's original transaction identifier of the transaction to be voided, that was provided in the *merchantTxId* field of the Auth/Purchase *Session Token Request* and *Auth/Purchase Request (*see sections 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
agentId	String(18)	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

8.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&originalTxId=1234&originalMerchantTxId

8.2 Session Token Response – Processed

8.2.1 Format

JSON

8.2.2 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 8.1)

token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Void Request (see Section 8.4) 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

8.2.3 Example

{"result":"success","merchantId":1111111,"token":"fghij67890fghij67890",”resultId”:”fghij67890fghij67890”}

8.3 Session Token Response – Not Processed

8.3.1 Format

JSON

8.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “failure”
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 8.1)
errors	String Array	List of issues

8.3.3 Example

{"result":"failure","merchantId":1111111,"errors":["Access denied"]}

8.4 Void Request

8.4.1 Format

POST Request

8.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 8.1)
token	String (40)	Y	Session Token received in the Session Token Response – Processed (section 8.2)

8.4.3 Example

merchantId=1111111&token=fghij67890fghij67890

8.5 Void Response - Processed

8.5.1 Format

JSON

8.5.2 Definition

Parameter/Label	Data Type	Description
result	String (40)	Will always be “success”
merchantId	Integer (18)	The *merchantId* value sent in the Session Token Request (section 8.1)
action	String (enum)	Will always be “VOID”

Parameter/Label	Data Type	Description
originalMerchantTxId	String (50)	The merchant transaction identifier of the transaction that was voided
originalTxId	Integer (18)	The eService Gateway transaction Id of the transaction voided sent in the Session Token Request (section 8.1)
txId	Integer (18)	The unique identifier for the void transaction in the eService Gateway
amount	BigDecimal (15.2 or 15.3)	The transaction amount voided sent in the Session Token Request (section 8.1)
currency	String (enum)	The transaction ISO alpha-3 currency code, as defined in the ISO 4217 standard, used in the original and void transactions
customerId	String (20)	The eService Gateway customer identifier used in the original and void transactions
pan	String (100)	The customer account value/number used in the original and the void transactions
brandId	Integer (18)	The *brandId* value for the original and the void transaction
paymentSolutionId	Integer (18)	The *paymentSolutionId* value used in the original and void transactions
status	String (enum)	The status of the transaction in eService Gateway: Status: VOID — Condition: If "VOID" successful Status: ERROR — Condition: If an error was returned by the payment process
errors	String (400)	Any errors that occurred during the successful processing of a transaction
customParameter1Or ... customParameter20Or	String (50)	The original 20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation
customParameter1 ... customParameter20	String (50)	20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation, with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.

8.5.3 Example

{"result":"success","merchantId":111111,"originalMerchantTxId":"abc123","originalTxId":123,"txId":546,"amount":12.50,"currency":"GBP","customerId":"mgn456","action":"VOID","pan":"45ae201ghy23498FjMj701","brandId":3,"paymentSolutionId":500,"status":"VOID"}

8.6 Void Response – Not Processed

8.6.1 Format

JSON

8.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantid	Integer (18)	The *merchantId* value received in the Session Token Request (section 8.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the eService Gateway, if not received in the Session Token Request (section 8.1)
txId	Integer (18)	The unique identifier for the void transaction attempt in the eService Gateway

errors	String Array	List of errors.

8.6.3 Example

{"result":"failure","merchantId":1111111,"merchantTxId":"abc123","txId":123,"errors":["communications failure"]}

9 CAPTURE API Operation

9.1 Session Token Request

9.1.1 Format

POST Request

9.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer(18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String(64)	Y	The merchant's password to the eService Gateway provided at on-boarding
action	String (enum)	Y	Must be "CAPTURE"
timestamp	Integer(18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String(253)	Y	The merchant's URL that will make the Capture Request (section 9.4) Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
originalTxId	Integer(18)	N	The eService Gateway transaction Id of the transaction to be captured This will have been returned in the *txId* field of the *Auth/Purchase Response – Processed* (see sections 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
originalMerchantTxId	String(50)	Y	The merchant's original transaction identifier of the transaction to be captured, that was provided in the *merchantTxId* field of the Auth/Purchase *Session Token Request* and *Auth/Purchase Request (*see sections 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method)
agentId	String(18)	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
amount	BigDecimal (10.2 or 10.3) BigDecimal (15.2 or 15.3)	M	The amount to capture Can be less than or equal to the original transaction amount Cannot be more than the original transaction amount

9.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&originalTxId=123456789&originalMerchantTxId=XYZ123456789ABC&allowOriginUrl=www.merchantsite.com&action=CAPTURE&timestamp=1249751864238&agentId=brian01&amount=120

9.2 Session Token Response - Processed

9.2.1 Format

JSON

9.2.2 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 9.1)
Token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Capture Request (see Section 9.4) 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

9.2.3 Example

{"result":"success","merchantId":1111111,"token":"abcde12345abcde12345",”resultId”:”fghij67890fghij67890”}

9.3 Session Token Response – Not Processed

9.3.1 Format

JSON

9.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 9.1)
errors	String Array	List of issues

9.3.3 Example

{"result":"failure","merchantId":1111111,"errors":["Access denied"]}

9.4 Capture Request

9.4.1 Format

POST Request

9.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 9.1)
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 9.2)

9.4.3 Example

merchantid=1111111&token=abcde12345abcde12345

9.5 Capture Response – Processed

9.5.1 Format

JSON

9.5.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "success"
merchantId	Integer (18)	The *merchantId* value sent in the Session Token Request (section 9.1)
action	String (enum)	Will always be "CAPTURE"
originalMerchantTxId	String (50)	The merchant transaction identifier of the transaction that was captured
originalTxId	Integer (18)	The eService Gateway transaction Id of the transaction captured sent in the Session Token Request (section 9.1)
		The unique identifier for the capture transaction in the eService Gateway
amount	BigDecimal (15.2 or 15.3)	The transaction amount captured sent in the Session Token Request (section 9.1)
currency	String (enum)	The transaction ISO alpha-3 currency code, as defined in the ISO 4217 standard, used in the original and capture transactions
customerId	String (20)	The eService Gateway customer identifier used in the original and capture transactions
pan	String (100)	The customer account value/number used in the original and the capture transactions
brandId	Integer (18)	The *brandId* value for the original and the capture transaction
paymentSolutionId	Integer (18)	The *paymentSolutionId* value used in the original and capture transactions
status	String (enum)	The status of the transaction in eService Gateway: Status: SET_FOR_CAPTURE — Condition: If "CAPTURE" successful All capture transactions are batch processed, when the status will be set to CAPTURED Status: ERROR — Condition: If an error was returned by the payment process
errors	String (400)	Any errors that occurred during the successful processing of a transaction
customParameter1Or ... customParameter20Or	String (50)	The original 20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation
customParameter1 ... customParameter20	String (50)	20 free text fields provided by the merchant in the Session Token Request of the Auth/Purchase API Operation, with non-Basic Latin characters replaced by a space character. These values will have been sent for payment processing.

9.5.3 Example

{"result":"success","merchantId":111111,"originalMerchantTxId":"abc123","originalTxId":123,"amount":12.50,"currency":"GBP","customerId":"mgn456","action":"CAPTURE","pan":"45ae201ghy23498FjMj701","brandId":3,"paymentSolutionId":500,"status":"SET_FOR_CAPTURE"}

9.6 Capture Response – Not Processed

9.6.1 Format

JSON

9.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantid	Integer (18)	The *merchantId* value received in the Session Token Request (section 9.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the eService Gateway, if not received in the Session Token Request (section 9.1)
txId	Integer (18)	The unique identifier for the capture transaction attempt in the eService Gateway
errors	String Array	List of errors.

9.6.3 Example

{"result":"failure","merchantId":1111111,"merchantTxId":"abc123","txId":123,"errors":["communications failure"]}

10 Transaction Result Call

The notification or result call is sent as a POST request message with the following parameters:

Parameter	Data Type	Description
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request of the API Operation
action	String (enum)	The *action* received in the Session Token Request of the API Operation
merchantTxId	String (50)	The merchant's reference for the transaction provided in the *merchantTxId* parameter in the API Operation If the *merchantTxId* parameter was empty or omitted, a value will have been generated by the eService Gateway as a hexadecimal string
txId	Integer (18)	The unique identifier for the transaction in the eService Gateway
acquirerTxId	String (100)	The transaction identifier in acquirer system, if acquirer returns it
amount	BigDecimal (15.2 or 15.3)	The total transaction amount, including tax, shipping, surcharge and discount amounts
currency	String (enum)	The ISO alpha-3 code for the currency of the transaction, as defined in the ISO 4217 standard
customerId	String (20)	The merchant's reference for the customer of the transaction provided in the *customerId* parameter in the API Operation If the *customerId* parameter was empty or omitted, a value will have been generated by the eService Gateway
pan	String (100)	Customer payment account number or eService Gateway payment card token number used in the transaction
brandId	Integer (18)	The Brand Id used in the transaction, which was supplied by the eService Gateway when the merchant account was set up, or the default set up in the eService Gateway if none was provided
paymentSolutionId	Integer (18)	The eService Gateway Payment Solution Identifier used in the transaction
status	String (enum)	The transaction status in the eService Gateway after the API Operation is completed
acquirer	String (100)	The acquirer name in case of a Credit Card payment or the payment solution name if an alternative payment method has been used
acquirerAmount	BigDecimal (15.2 or 15.3)	Amount processed by payment acquirer May be different to *amount* in the original transaction
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 original transaction, e.g. if a currency conversion was applied
country	String (enum)	The ISO alpha-2 code country in which the transaction took place, as defined in the ISO 3166 standard
freeText	String (200)	Merchant free text
language	String (enum)	The ISO alpha-2 language code, as defined in ISO 639-1 standard, for the language that was provided in the API Operation
errorMessage	String (400)	Only applies to ERROR transactions It is a brief description of the cause of the error
paymentSolutionDetails	JSON block	For payment cards only: the Transaction Authorisation Code received from the acquirer, format: {"authCode":""}

11 GET STATUS API Operation

11.1 Session Token Request

11.1.1 Format

POST Request

11.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String (64)	Y	The merchant's password to the eService Gateway provided at on-boarding
action	String (enum)	Y	Must be "GET_STATUS"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String (253)	Y	The merchant's URL that will make the Get Status Request (section 11.4) Cross-Origin Resource Sharing (CORS) headers will allow only this origin.

11.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&allowOriginUrl=www.merchantsite.com&action=GET_STATUS&timestamp=1249751864238

11.2 Session Token Response – Processed

11.2.1 Format

JSON

11.2.2 Definition

Parameter	Data Type	Description
result	String (enum)	Will always be "success"
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 11.1)
token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Get Status Request (see Section 11.4) 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

11.2.3 Example

{"result":"success","merchantId":1111111,"token":"fghij67890fghij67890","resultId":"fghij67890fghij67890"}

11.3 Session Token Response – Not Processed

11.3.1 Format

JSON

11.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantId	Integer (18)	The *merchantId* value received in the Session Token Request (section 11.1)
errors	String Array	List of issues

11.3.3 Example

{"result":"failure","merchantId":1111111,"errors":["Access denied"]}

11.4 Get Status Request

11.4.1 Format

POST Request

11.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 11.1)
token	String (40)	Y	Session Token received in the Session Token Response – Processed (section 11.2)
action	String (enum)	Y	"GET_STATUS"
txId	Integer (18)	C	The eService Gateway transaction Id of the transaction for which the status is requested This will have been returned in the *txId* field of the *Auth/Purchase Response – Processed* (see section 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method) **Condition**: Must be provided if the *merchantTxId* is not provided
merchantTxId	Integer (50)	C	The merchant's transaction identifier of the transaction for which the status is requested, that was provided in the *merchantTxId* field of the Auth/Purchase *Session Token Request* and *Auth/Purchase Request (*see section 6 - AUTH/PURCHASE API Operation – Hosted Payment Page Integration, as appropriate to the integration method) **Condition**: Must be provided if the *txId* is not provided

11.4.3 Example

MerchantId=111111&token=fghij67890fghij67890&action=GET_STATUS&txId=546&MerchantTxId=abc123

11.5 Get Status Response - Processed

11.5.1 Format

JSON

11.5.2 Definition

Parameter	Data Type	Description
result	String (enum)	Will always be "success"

merchantId	Integer (18)	The *merchantId* value sent in the Session Token Request (section 11.1)
txId	Integer (18)	The eService Gateway transaction identifier provided in the in the Get Status Request (section 11.4)
merchantTxId	String (50)	The merchant's transaction identifier provided in the in the Get Status Request (section 11.4)
status	String (enum)	The status of the transaction in the eService Gateway (see *API Operations – 0 – Overview*)

11.5.3 Example

{“result”:”success”,"merchantId":111111,"merchantTxId":"abc123",”txId”:546,”status”:”SET_FOR_CAPTURE”}

11.6 Get Status Response – Not Processed

11.6.1 Format

JSON

11.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be “failure”
merchantid	Integer (18)	The *merchantId* value received in the Session Token Request (section 11.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the eService Gateway, if not received in the Session Token Request (section 11.1)
txId	Integer (18)	The unique identifier for the refund transaction attempt in the eService Gateway
errors	String Array	List of errors

11.6.3 Example

{"result":"failure","merchantId":1111111,"merchantTxId":"abc123","txId":123,"errors":["communications failure"]}

12 GET AVAILABLE PAYMENT SOLUTIONS API Operation

12.1 Session Token Request

12.1.1 Format

POST Request

12.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding
password	String (64)	Y	The merchant's password to the eService Gateway provided at on-boarding
action	String (enum)	Y	Must be "GET_AVAILABLE_PAYSOLS"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String (253)	Y	The merchant's URL that will make the Get Available Payment Solutions Request (section 12.4) Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
currency	String (enum)	Y	The ISO alpha-3 code, as defined in the ISO 4217 standard, for the currency in which a Payment Solution is available
country	String (enum)	Y	The ISO alpha-2 code, as defined in the ISO 3166 standard, for the currency in which a Payment Solution is available
brandid	Integer (18)	N	The Brand Id supplied by the eService Gateway when the merchant account was set up for which a Payment Solution is available

12.1.3 Example

merchantId=1111111&password=klw74U6yt40mNo&allowOriginUrl=www.merchantsite.com&action=GET_AVILABLBE_SOLS&timestamp=1249751864238&currency=EUR&country=IE&brandId=1234567

12.2 Session Token Response - Processed

12.2.1 Format

JSON

12.2.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "success"
merchantId	Integer (18)	The *merchantId* received in the Session Token Request (section 12.1)
token	String (40)	The Session Token that is a one-time use, hexadecimal string The Session Token that must only be used for the Get Available Payment Solutions Request (see Section 12.4) 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

12.2.3 Example

{"result":"success","merchantId":1111111,"token":"fghij67890fghij67890","resultId":"fghij67890fghij67890"}

12.3 Session Token Response – Not Processed

12.3.1 Format

JSON

12.3.2 Definition

Parameter	Data Type	Description
result	String (40)	Will always be "failure"
merchantId	Integer (18)	The *merchantId* received in the Session Token Request (section 12.1)
errors	String Array	List of issues

12.3.3 Example

{"result":"failure","merchantId":1111111,"errors":["Access denied"]}

12.4 Get Available Payment Solutions Request

12.4.1 Format

POST Request

12.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	The identifier for the merchant in the eService Gateway provided at on-boarding This must be the same as that sent in the Session Token Request (section 12.1)
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 12.2)

12.4.3 Example

merchantId=1111111&token=fghij67890fghij67890

12.5 Get Available Payment Solutions Response – Processed

12.5.1 Format

JSON

12.5.2 Definition

Parameter	Data Type	Description
result	String (enum)	Will always be "success"
merchantId	Integer (18)	The *merchantId* received in the Session Token Request (section 12.1)
data	String Array	List of payment solutions available
ID	Integer (18)	The unique identifier for the Payment Solution in the eService Gateway
NAME	String (150)	The name of the Payment Solution in the eService Gateway
resultId	String	Hexadecimal string that is to be used in any support request calls

12.5.3 Example

{"result":"success","merchantid":"1111111","data":[{"NAME":"CreditDebitCards","ID":"500"},{"NAME":"Neteller","ID":"100"}],"additionalDetails":{},"resultId":"e79e9506-a38a-403e-b435-be0c91b436db"}

12.6 Get Available Payment Solutions Response – Not Processed

12.6.1 Format

JSON

12.6.2 Definition

Parameter	Data Type	Description
result	String (enum)	Will always be “failure”
merchantid	Integer (18)	The *merchantId* received in the Session Token Request (section 12.1)
errors	String Array	List of errors

12.6.3 Example

{"result":"failure","merchantId":1111111,"errors":["communications failure"]}

Appendix B API Operations Definitions

Acronym or term	Description
Processed	In this document, the Response sections that are defined as Processed indicate that the eService Gateway processed the transaction Request. The transaction status will change. Although the <result> field = "success", the outcome may result in a transaction failure. For example, a CAPTURE Request may result in a successful capture of the funds, or it may fail, because the funds are unavailable, or the requested amount may not equal the original amount of the AUTH transaction. The exception is the Session Token Responses. A Session Token will always be successfully issued if the Request was processed.
Not Processed	In this document, the Response sections that are defined as Not Processed indicate that the eService Gateway failed to process the transaction Request. The status of the transaction will not change as a result. Processing failures are generally due to technical issues. The request should be re-submitted.
Merchant's Server IP Addresses	When the merchant is set up, the IP Addresses of the merchant's servers that will make the HTTP POST Requests, are stored in the eService Gateway. During the API Operation, the IP Address of the requesting server is validated against that stored in the eService Gateway for the Merchant ID, along with the Password provided. If the IP Address does not match, the request is rejected.
Session Tokens	All API Operations require a Session Token before a payment API Operation can be performed. The Session Token that is a one-time use, hexadecimal string that must only be used for the Action Request, that is used by the eService Gateway to validate an incoming request and to connect the Session Token Request with the API Operation Request. The subsequent API Operation Request must contain the Session Token that is associated with the API Operation. Session Tokens are valid for 3600 second (1 hour) after which they expire Any requests with expired session tokens will be rejected and ignored by the eService Gateway
Result IDs	The Result ID is included in all Response JSON files, received from the eService Gateway. The Result ID is a randomly generated, 18-character, hexadecimal string. The Result ID should be retained by the merchant's system for any queries about the API Operation in the future, should problems arise. This provides low-level detail about the overall transaction. Combined with the Session Token it provides a complete reference to the transaction in the eService Gateway.
Customer IDs	A merchant may have a customer management system that has customer account identifiers. These identifiers should be included in relevant Request files. The Response files will reference the *customerId* provided, thus enabling the merchant to associate the transaction with the customer in their own system. • If the *customerId* is provided, the customer will be set up in the eService Gateway once, and all subsequent transactions will be associate with that same customer. • If the *customerId* field is left blank/empty, the eService Gateway will generate a random number identifier that will only be relevant to the API operation in the eService Gateway. Therefore, a single customer can appear in the eService Gateway database several times. In the eService Gateway Back-Office application, the *customerId* field can be used for filtering and searching, along with other customer details. It is more efficient to find a customer using the merchant's known identifier than the one randomly generated by the eService Gateway.

Appendix C UAT Trigger Values

When integrating with the eService Gateway in the User Acceptance Testing (UAT) environment, certain *amount* values 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

Amount	Status	Error Message
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
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}

Amount	Status	Error Message
0.97	SUCCESS	{none}
0.98	SUCCESS	{none}
0.99	SUCCESS	{none}

Appendix D 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	String (4)	N	Cardholder Account Purchase Count: Number of purchases with this cardholder account during the previous six months.
custProvisionAttemptsPerDay	String (3)	N	Number of Provisioning Attempts Per Day: Number of Add Card attempts in the last 24 hours.

Data Element	Data Type	Req	Description
custTxnActivityDay	String (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	String (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	String (8)	N	Payment Account Age: Date that the payment account was enrolled in the cardholder's account with the merchant. **Date format** = YYYYMMDD
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 E 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 F 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 G 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

Data Element	Data Type	Req	Description
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.)