



API Specification

Version 2.0

Version Date: 11/01/2018

Notice: The information in this document is confidential and proprietary to UniversalPay and is only intended for use by merchant customers of UniversalPay, internal staff, and authorised business partners of UniversalPay.

This document is protected by copyright restricting its use, replication in any format, and distribution. No part of this document may be reproduced in any form by any means without the express permission of UniversalPay.

UniversalPay reserves the right to amend, delete or add to the contents of the document, at any time, and to make improvements and/or changes to the products and/or programmes described in this document.

Every reasonable attempt will be made to ensure that the contents of the document are accurate, and a true reflection of the products and programmes described herein. However, UniversalPay will not be held liable for any inaccuracies of any nature, however communicated by UniversalPay.

UniversalPay and other trademarks are trademarks or registered trademarks of their respective owners.

All other product names mentioned in this document are the trademarks of their respective owners.

© UniversalPay 2017

Contents

- Contents..... 2
- 1 Introduction 8
 - 1.1 Document Purpose 8
 - 1.2 Audience 8
 - 1.3 Definitions 9
- 2 Integration Options..... 10
 - 2.1 PCI Compliant Payment Form Integration 10
 - 2.2 PCI Compliant Form Plugin 10
 - 2.3 Non-PCI Compliant Direct API Integration 10
 - 2.4 Virtual Terminal Integration..... 11
 - 2.5 Standalone Backoffice and Virtual Terminal..... 11
- 3 Gateway Interface..... 12
 - 3.1 Addresses 12
 - 3.1.1 User Acceptance Testing Addresses 12
 - 3.1.2 Production Addresses 12
 - 3.2 HTTP Specification..... 12
 - 3.2.1 Example HTTP Request 12
 - 3.2.2 POST data 12
- 4 API Operations 13
 - 4.1 Process Overview 13
 - 4.2 Transaction Statuses 14
- 5 TOKENIZE – API Operation 15
 - 5.1 Session Token Request..... 15
 - 5.1.1 Format..... 15
 - 5.1.2 Definition..... 15
 - 5.1.3 Example..... 15
 - 5.2 Session Token Response - Processed..... 15
 - 5.2.1 Format..... 15
 - 5.2.2 Definition..... 15
 - 5.2.3 Example..... 15
 - 5.3 Session Token Response – Not Processed 15
 - 5.3.1 Format..... 15
 - 5.3.2 Definition..... 15
 - 5.3.3 Example..... 15
 - 5.4 Tokenize Request 16
 - 5.4.1 Format..... 16
 - 5.4.2 Definition..... 16
 - 5.4.3 Example..... 16
 - 5.5 Tokenize Response – Processed 17
 - 5.5.1 Format..... 17

- 5.5.2 Definition..... 17
- 5.5.3 Example..... 17
- 5.6 Tokenize Response – Not Processed..... 17
 - 5.6.1 Format..... 17
 - 5.6.2 Definition..... 17
 - 5.6.3 Example..... 17
- 6 AUTH/PURCHASE – API Operation..... 18
 - 6.1 Transaction Status..... 18
 - 6.2 3DS Redirection..... 18
 - 6.3 Process 19
 - 6.4 Session Token Request..... 20
 - 6.4.1 Format..... 20
 - 6.4.2 Definition..... 20
 - 6.4.3 Example..... 24
 - 6.5 Session Token Response - Processed..... 25
 - 6.5.1 Format..... 25
 - 6.5.2 Definition..... 25
 - 6.5.3 Example..... 25
 - 6.6 Session Token Response – Not Processed 25
 - 6.6.1 Format..... 25
 - 6.6.2 Definition..... 25
 - 6.6.3 Example..... 25
 - 6.7 Load Payment Form Request 26
 - 6.7.1 Format..... 26
 - 6.7.2 Definition..... 26
 - 6.8 Sample Payment Form Call 26
 - 6.9 Auth/Purchase Request 27
 - 6.9.1 Format..... 27
 - 6.9.2 Definition..... 27
 - 6.9.3 Example..... 27
 - 6.10 Redirection Response 28
 - 6.10.1 Format..... 28
 - 6.10.2 Definition..... 28
 - 6.10.3 Example..... 28
 - 6.11 Auth/Purchase Response – Processed..... 29
 - 6.11.1 Format..... 29
 - 6.11.2 Definition..... 29
 - 6.11.3 Example..... 29
 - 6.12 Auth/Purchase Response – Not Processed..... 30
 - 6.12.1 Format..... 30
 - 6.12.1 Definition..... 30

- 6.12.2 Example..... 30
- 7 CAPTURE – API Operation..... 31
 - 7.1 Transaction Status..... 31
 - 7.2 Funds Transfer 31
 - 7.3 Session Token Request..... 32
 - 7.3.1 Format..... 32
 - 7.3.2 Definition..... 32
 - 7.3.3 Example..... 32
 - 7.4 Session Token Response - Processed..... 32
 - 7.4.1 Format..... 32
 - 7.4.2 Definition..... 32
 - 7.4.3 Example..... 32
 - 7.5 Session Token Response – Not Processed 32
 - 7.5.1 Format..... 32
 - 7.5.2 Definition..... 32
 - 7.5.3 Example..... 32
 - 7.6 Capture Request 33
 - 7.6.1 Format..... 33
 - 7.6.2 Definition..... 33
 - 7.6.3 Example..... 33
 - 7.7 Capture Response – Processed 33
 - 7.7.1 Format..... 33
 - 7.7.2 Definition..... 33
 - 7.7.3 Example..... 33
 - 7.8 CAPTURE Response – Not Processed 34
 - 7.8.1 Format..... 34
 - 7.8.2 Definition..... 34
 - 7.8.3 Example..... 34
- 8 VOID – API Operation..... 35
 - 8.1 Session Token Request..... 35
 - 8.1.1 Format..... 35
 - 8.1.2 Definition..... 35
 - 8.1.3 Example..... 35
 - 8.2 Session Token Response – Processed 35
 - 8.2.1 Format..... 35
 - 8.2.2 Definition..... 35
 - 8.2.3 Example..... 35
 - 8.3 Session Token Response – Not Processed 35
 - 8.3.1 Format..... 35
 - 8.3.2 Definition..... 35
 - 8.3.3 Example..... 35

- 8.4 Void Request 36
 - 8.4.1 Format..... 36
 - 8.4.2 Definition..... 36
 - 8.4.3 Example..... 36
- 8.5 Void Response - Processed 36
 - 8.5.1 Format..... 36
 - 8.5.2 Definition..... 36
 - 8.5.3 Example..... 36
- 8.6 VOID Response – Not Processed..... 37
 - 8.6.1 Format..... 37
 - 8.6.2 Definition..... 37
 - 8.6.3 Example..... 37
- 9 REFUND – API Operation 38
 - 9.1 Session Token Request..... 38
 - 9.1.1 Format..... 38
 - 9.1.2 Definition..... 38
 - 9.1.3 Example..... 38
 - 9.2 Session Token Response - Processed 39
 - 9.2.1 Format..... 39
 - 9.2.2 Definition..... 39
 - 9.2.3 Example..... 39
 - 9.3 Session Token Response – Not Processed 39
 - 9.3.1 Format..... 39
 - 9.3.2 Definition..... 39
 - 9.4 Refund Request..... 40
 - 9.4.1 Format..... 40
 - 9.4.2 Definition..... 40
 - 9.4.3 Example..... 40
 - 9.5 Refund Response – Processed 40
 - 9.5.1 Format..... 40
 - 9.5.2 Definition..... 40
 - 9.5.3 Example..... 40
 - 9.6 REFUND Response – Not Processed 41
 - 9.6.1 Format..... 41
 - 9.6.2 Definition..... 41
 - 9.6.3 Example..... 41
- 10 GET STATUS – API Operation 42
 - 10.1 Session Token Request..... 42
 - 10.1.1 Format..... 42
 - 10.1.2 Definition..... 42
 - 10.1.3 Example..... 42

- 10.2 Session Token Response – Processed 42
 - 10.2.1 Format..... 42
 - 10.2.2 Definition..... 42
 - 10.2.3 Example..... 42
- 10.3 Session Token Response – Not Processed 42
 - 10.3.1 Format..... 42
 - 10.3.2 Definition..... 42
- 10.4 Status Check Request..... 43
 - 10.4.1 Format..... 43
 - 10.4.2 Definition..... 43
 - 10.4.3 Example..... 43
- 10.5 Status Check Response - Processed 43
 - 10.5.1 Format..... 43
 - 10.5.2 Definition..... 43
 - 10.5.3 Example..... 43
- 10.6 GET_STATUS Response – Not Processed 43
 - 10.6.1 Format..... 43
 - 10.6.2 Definition..... 43
- 11 GET AVAILABLE PAYMENT SOLUTIONS – API Operation 44
 - 11.1 Session Token Request..... 44
 - 11.1.1 Format..... 44
 - 11.1.2 Definition..... 44
 - 11.1.3 Example..... 44
 - 11.2 Session Token Response - Processed..... 44
 - 11.2.1 Format..... 44
 - 11.2.2 Definition..... 44
 - 11.2.3 Example..... 44
 - 11.3 Session Token Response – Not Processed 44
 - 11.3.1 Format..... 44
 - 11.3.2 Definition..... 44
 - 11.3.3 Example..... 44
 - 11.4 Get Available Payment Solutions Request..... 45
 - 11.4.1 Format..... 45
 - 11.4.2 Definition..... 45
 - 11.4.3 Example..... 45
 - 11.5 Get Available Payment Solutions Response – Processed 45
 - 11.5.1 Format..... 45
 - 11.5.2 Definition..... 45
 - 11.5.3 Example..... 45
 - 11.6 Get Available Payment Solutions Response – Not Processed 45
 - 11.6.1 Format..... 45

11.6.2	Definition.....	45
11.6.3	Example.....	45
12	Transaction Result Call.....	46
12.1	Transaction Result Response	46
Appendix A	Payment Page Customisation Policy.....	48
Appendix B	Card Type Values.....	49
Appendix C	PCI Compliant Payment Form Code Examples.....	50
C.1	Sample code for redirection based integration	50
C.2	Sample code for JavaScript based integration:.....	51

1 Introduction

1.1 Document Purpose

This specification provides Small/Medium Enterprise (SME) merchant developers with the necessary information to integrate their sales systems with the UniversalPay Gateway Application Programming Interfaces (APIs).

Larger Enterprise Merchants have different versions to the functionality detailed herein.

1.2 Audience

This API Specification is intended to enable member merchant business technology staff to integrate their systems with the UniversalPay Gateway APIs.

The document defines the external interfaces to the UniversalPay Gateway necessary to:

- Request card tokens
- Request payment authorisation
- Make purchases
- Capture funds from customers' accounts
- Void authorised payment requests
- Refund purchases (captured payments)
- Request transactions statuses
- Integrate PCI Compliant Payment Forms
- Integrate the UniversalPay Gateway's Virtual Terminal
- Receive Transaction Call results
- Test integrations

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

1.3 Definitions

Acronym or term	Description
Processed	<p>In this document, the Response sections that are defined as Processed indicate that the UniversalPay Gateway processed the transaction Request.</p> <p>The transaction status will change.</p> <p>Although the <result> field = “success”, the outcome may result in a transaction failure.</p> <p>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.</p> <p>The exception is the Session Token Responses. A Session Token will always be successfully issued if the Request was processed.</p>
Not Processed	<p>In this document, the Response sections that are defined as Not Processed indicate that the IPG Gateway failed to process the transaction Request.</p> <p>The status of the transaction will not change as a result.</p> <p>Processing failures are generally due to technical issues. The request should be re-submitted.</p>
Merchant's Server IP Addresses	<p>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 UniversalPay Gateway.</p> <p>During the API Operation, the IP Address of the requesting server is validated against that stored in the UniversalPay Gateway for the Merchant ID, along with the Password provided.</p> <p>If the IP Address does not match, the request is rejected.</p>
Session Tokens	<p>All API Operations require a Session Token before a payment API Operation can be performed.</p> <p>The Session Token is a randomly generated, 18-character, hexadecimal string.</p> <p>The Session Token is used by the UniversalPay Gateway to validate an incoming request and to connect the Session Token Request with the API Operation Request.</p> <p>The subsequent API Operation Request must contain the Session Token, that is associated with the API Operation, sent in the Session Token Response JSON file by the UniversalPay Gateway.</p> <p>Session Tokens have a limited lifetime of a few minutes. Therefore, the API Operation Request must be timely, otherwise the Session Token may expire.</p> <p>The Session Token should be retained by the merchant's system for any queries about the API Operation in the future, should problems arise. This provides high-level detail about the overall transaction.</p>
Result IDs	<p>The Result ID is included in all Response JSON files, received from the UniversalPay Gateway.</p> <p>The Result ID is a randomly generated, 18-character, hexadecimal string.</p> <p>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.</p> <p>Combined with the Session Token it provides a complete reference to the transaction in the UniversalPay Gateway.</p>
Customer IDs	<p>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 <i>customerId</i> provided, thus enabling the merchant to associate the transaction with the customer in their own system.</p> <ul style="list-style-type: none"> • If the <i>customerId</i> is provided, the customer will be set up in the UniversalPay Gateway once, and all subsequent transactions will be associate with that same customer. • If the <i>customerId</i> field is left blank/empty, the UniversalPay Gateway will generate a random number identifier that will only be relevant to the API operation in the IPG Gateway. <p>Therefore, a single customer can appear in the UniversalPay Gateway database several times.</p> <p>In the UniversalPay Gateway Back-Office application, the <i>customerId</i> 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 UniversalPay Gateway.</p>

2 Integration Options

The Intelligent Payments API offers the following types of integrations for payment processing:

- PCI Compliant Payment Form Integration (hosted payment pages)
- PCI Compliant Payment Form Plugin (for third party shopping cart platforms)
- Non-PCI Compliant Direct API Integration (Merchant's own payment form)
- Virtual Terminal Integration (Within Merchant's Back-Office systems)
- Standalone Backoffice and Virtual Terminal (UniversalPay Gateway supplied application)

2.1 PCI Compliant Payment Form Integration

The Payment Form is also known as the Cashier UI, or just Cashier.

- The Payment Form is called by the merchant shopping cart platform directly after obtaining a valid Session Token from the UniversalPay Gateway
- The Payment Form is PCI Compliant as it resides within the UniversalPay Gateway, which is a Level 1 PCI Compliant environment
- The Payment Form has a pre-defined layout and presentation. The form can be customised by introducing the CSS code when the payment form (Cashier) is loaded
- The Payment Form is to be used ONLY for E-Commerce Type Card Payments
- The Payment Form will also provide 3D-Secure authentication, if 3D-Secure is enabled in the Back-Office, or the merchant forces the secure payment in the Session Token Request

See section Appendix C - PCI Compliant Payment Form for code examples.

2.2 PCI Compliant Form Plugin

- The Payment Form is installed as a plugin within the merchant's shopping cart platform
- The Payment Form contains the same features as the PCI Compliant Payment Form Integration, above
- The plugin for the Intelligent Payments Payment Form is obtained from [SellXed Website](#); select "Intelligent Payments" from the Payment Service Provider list

2.3 Non-PCI Compliant Direct API Integration

- The merchant takes total control of what is displayed to the customers on their Payment Page, i.e. the merchant develops own payment form
- The merchant calls the UniversalPay Gateway services directly through the API
- Direct API Integration will also provide 3D-Secure authentication, if the 3D-Secure is enabled in the Back-Office, or the merchant forces the secure payment in the Session Token Request
- With Direct API Integration, the merchant can choose which payment operation to perform. These are:
 - TOKENIZE Securely store card details in the UniversalPay Gateway, a Card Token is used for transactions
 - AUTH Perform an authorisation only
 - CAPTURE Perform a capture operation on a previous AUTH type payment
 - VOID Undo a previous AUTH type payment, i.e. prevent the transaction from being captured
 - PURCHASE Perform a full authorise and capture payment in one-step
 - REFUND Perform a refund (partial or full) on a previously captured transaction
- For each of the above operations, a separate Session Token is to be obtained from the UniversalPay Gateway

2.4 Virtual Terminal Integration

- The merchant integrates the UniversalPay Gateway Virtual terminal within the merchant system
- The Virtual Terminal may only be used for Mobile Order/Telephone Order (MOTO) type payments
- The Virtual Terminal comes in a pre-set layout and presentation and may not be modified or customised

2.5 Standalone Backoffice and Virtual Terminal

- The Standalone Backoffice and Virtual Terminal does not require any integration work from the merchant. It may only be used for MOTO type payments
- On the Standalone Backoffice and Virtual Terminal, the following operations may be performed:
 - AUTH - an authorisation only type payment (would need to be captured in a separate operation);
 - CAPTURE - a capture operation on a previous AUTH type payment;
 - VOID - Undo a previous AUTH type payment, so that it is not captured
 - PURCHASE - a full authorise and capture payment in one-step
 - REFUND - a refund (partial or full) on a previously captured transaction

3 Gateway Interface

3.1 Addresses

3.1.1 User Acceptance Testing Addresses

Session Token Request URL: <https://api.test.universalpay.es/token>

Payment Operation Action URL: <https://api.test.universalpay.es/payments>

3.1.2 Production Addresses

Session Token Request URL: <https://api.universalpay.es/token>

Payment Operation Action URL: <https://api.universalpay.es/payments>

3.2 HTTP Specification

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

3.2.1 Example HTTP Request

- POST: <https://api.universalpay.es/token>
- Content-Type: application/x-www-form-urlencoded
- Content-Length: 415

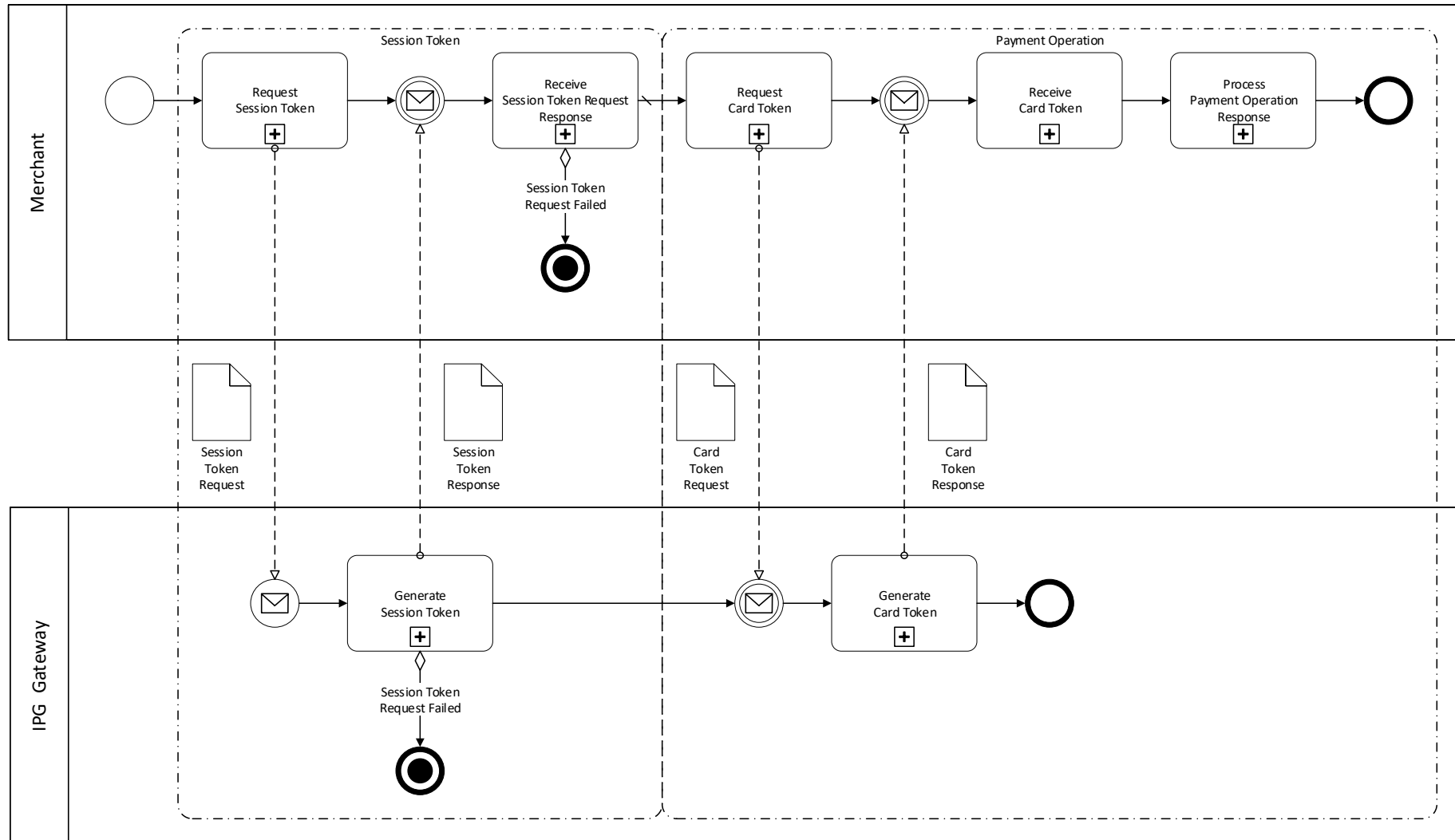
3.2.2 POST data

merchantId=160001&action=PURCHASE&password=password&allowOriginUrl=www.merchantsite.com×tamp=1459767453376&channel=ECOM&userDevice=DESKTOP&amount=25.96¤cy=GBP&country=DE&paymentSolutionId=500&specinCreditCardToken=123456781111&customerId=9876543&brandId=670&merchantNotificationUrl=https%3A%2F%2Fwww.posttestserver.com%2Fpost.php%2FipgTesting%3Fdir%3DJCTesting&merchantLandingPageUrl=https://www.merchantsite.com%2FlandingPage&forceSecurePayment=true

4 API Operations

4.1 Process Overview

Shown below is a generic view of how the UniversalPay Gateway API process operate. The primary feature to note is that each Payment Operation Request is preceded by a Session Token Request, which authenticates the merchant system before the Request Action is processed by the UniversalPay Gateway.

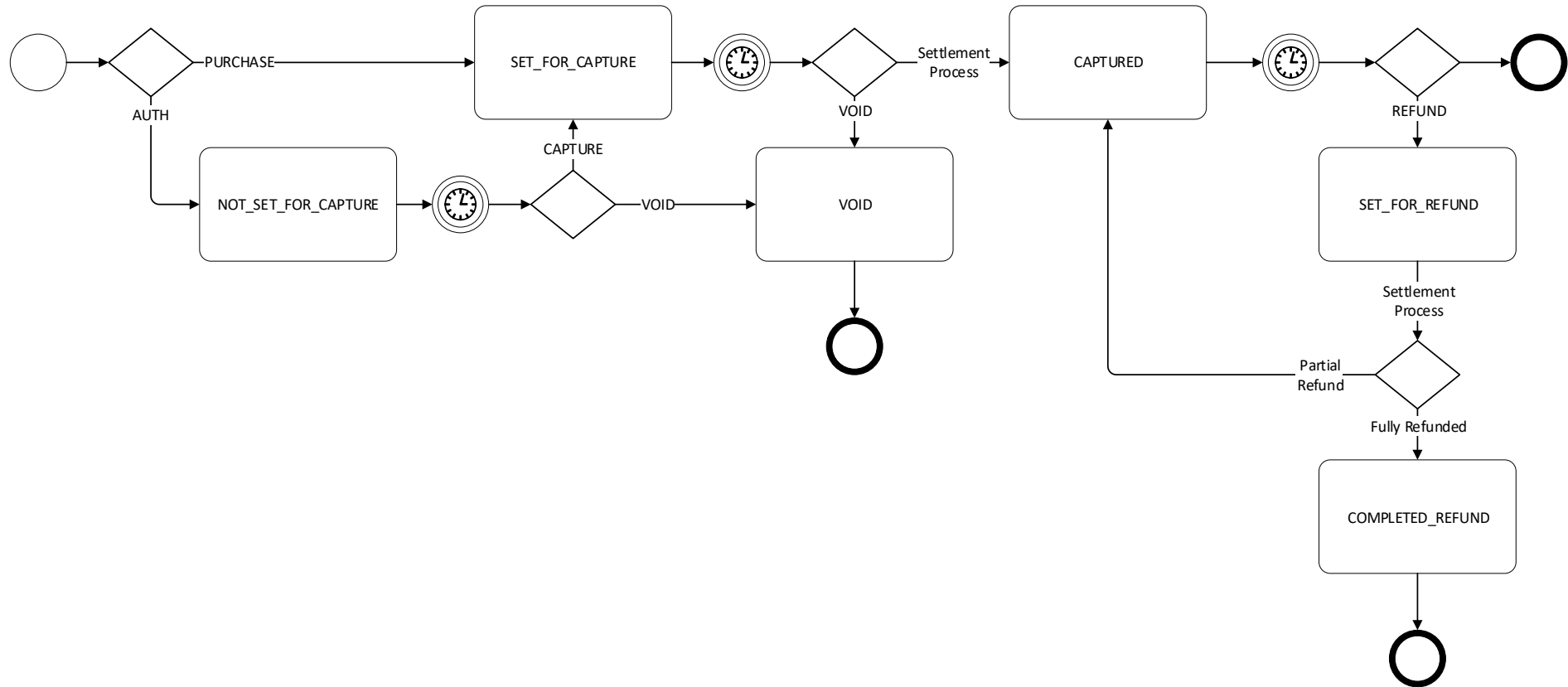


4.2 Transaction Statuses

Transactions in the UniversalPay 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 6).

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



References:

1. AUTH & PURCHASE: see section 6 - AUTH/PURCHASE – API Operation
2. CAPTURE: see section 7CAPTURE – API Operation
3. VOID: see section 8 - VOID – API Operation
4. Settlement Process: this is an internal process to the IPG Gateway, that communicates the transactions to the acquiring banks to settle funds between the merchants and their customers
5. REFUND: see section 9 - REFUND – API Operation

5 TOKENIZE – API Operation

The UniversalPay Gateway stores card details in a PCI Level 1 compliant environment. Therefore, the card details are encrypted, and the merchant is provided with a token for the card.

This facility removes the requirement for merchants to manage, store and communicate this sensitive data.

The Card Token is the only card identifier accepted as an input when a payment operation takes place.

5.1 Session Token Request

5.1.1 Format

POST Request

5.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway
action	String (enum)	Y	“TOKENIZE”
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
allowOriginUrl	String (255)	Y	Merchant's URL that will make the Tokenize Request (see Section 5.4). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
customerId	String (20)	N	Customer identifier in merchant system. If no value provided, a value will be auto-generated and returned in the Session Token Response - (see section 5.2). The received or generated value is stored against the card token.

5.1.3 Example

```
merchantId=111111&password=klw74U6yt40mNo&action=TOKENIZE&timestamp=1459767453376&allowOriginUrl=www.merchantsite.com&customerId=123456
```

5.2 Session Token Response - Processed

5.2.1 Format

JSON

5.2.2 Definition

Parameter	Data Type	Description
result	String (40)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (see Section 5.1)
token	String (40)	One-time use Session Token – hexadecimal string
resultId	String (40)	Hexadecimal string that is to be used in any support request calls.

5.2.3 Example

```
{"result":"success","merchantId":111111,"token":"abcde12345abcde12345"}
```

5.3 Session Token Response – Not Processed

5.3.1 Format

JSON

5.3.2 Definition

Parameter	Data Type	Description
result	String (40)	“failure”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (see Section 5.1)
errors	String Array	List of issues

5.3.3 Example

```
{"result":"failure","merchantId":111111,"errors":["Access denied"]}
```

5.4 Tokenize Request

5.4.1 Format

POST Request

5.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Request (see Section 5.1)
number	String (100)	Y	Card number - Used to determine the card type
nameOnCard	String (150)	Y	Cardholder name
expiryMonth	String (2)	Y	Card expiration month, e.g. "04"
expiryYear	String (4)	Y	Card expiration year, e.g. "2020"
cardDescription	String (50)	N	Free form text field for the merchant's use that is used for reconciliation. E.g. merchant's internal card's sequence id
startMonth	String (2)	C	Card issued month, e.g. "04" Conditional on the card type, determined from the [number], where Issuer Identification Number (IIN) Range is 50, 56, 57, 58, or 6 <ul style="list-style-type: none"> • Mandatory if card type is Maestro • Optional for all other cards
startYear	String (4)	C	Card issued year, e.g. "2014" Conditional on the card type, determined from the [number], where Issuer Identification Number (IIN) Range is 50, 56, 57, 58, or 6 <ul style="list-style-type: none"> • Mandatory if card type is Maestro • Optional for all other cards
issueNumber	String (2)	C	Card issue number Conditional on the card type, determined from the [number], where Issuer Identification Number (IIN) Range is 50, 56, 57, 58, or 6 <ul style="list-style-type: none"> • Mandatory if card type is Maestro • Optional for all other cards

5.4.3 Example

merchantId=111111&token=abcde12345abcde12345&number=1111222233334444&nameOnCard=NAME+OF+CARD+OWNER&expiryMonth=04&expiryYear=2020&cardDescription=Customer ID 123456

5.5 Tokenize Response – Processed

5.5.1 Format

JSON

5.5.2 Definition

Parameter	Data Type	Description
result	String (40)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 5.1)
cardToken	String (100)	Credit Card Token – hexadecimal string
customerId	String (20)	Customer identifier from Session Token Request (section 5.1)
cardType	Integer (enum)	Card type code – see Appendix B
cardIssuer	String (100)	Card issuer name
country	String (enum)	Card issuer country The value is the alpha-2 code as defined in the ISO 3166 standard

5.5.3 Example

```
{“result”:“success”,“merchantid”:111111,“cardToken”:“45ae201ghy23498FjMj701”,“customerId”:“123456”,“cardType”:400,“cardIssuer”:“HSBC”,“country”:“GB”}
```

5.6 Tokenize Response – Not Processed

5.6.1 Format

JSON

5.6.2 Definition

Parameter	Data Type	Description
result	String (40)	“failure”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 5.1)
errors	String Array	List of errors.

5.6.3 Example

```
{“result”:“failure”,“merchantid”:111111,“errors”:[“invalid_card_number”,“invalid_expiry_year”]}
```

6 AUTH/PURCHASE – API Operation

The difference between the AUTH and PURCHASE API Operations is the final status of the transaction.

- An AUTH API Operation simply gains confirmation of the customer's ability to pay; a subsequent CAPTURE – API Operation (section 7) is required to transfer funds
- A PURCHASE API Operation combines the AUTH and CAPTURE – API Operations in one operation; as with the CAPTURE – API Operation, the fund transfer occurs later in the day through a batch operation

6.1 Transaction Status

The AUTH/PURCHASE transaction will not have a status until processing is complete:

- AUTH transactions only certify that the card is valid, and the funds are available
 - The funds are not moved from the customer's card to the merchant account until a CAPTURE – API Operation (see Section 7) is performed.
 - The transaction status after a successful AUTH transaction is NOT_SET_FOR_CAPTURE this means the payment is authorised, but no funds are transferred; the CAPTURE – API Operation (see Section 7) must be called to transfer funds, which will set the transaction status to SET_FOR_CAPTURE
- PURCHASE operations combine the AUTH and CAPTURE – API Operations
 - The funds are moved from the customer's account to the merchant's account
 - The transaction status after a successful PURCHASE transaction is SET_FOR_CAPTURED

An AUTH or PURCHASE can be unsuccessful for two reasons:

- The transaction status is set to ERROR if there was an error in the API Operation
 - usually the result of a communication error between the customer browser and the IPG Gateway
- The transaction status is set to DECLINED if:
 - the card payment was refused by the acquirer, or
 - the 3DS authentication failed

A failure response is sent.

N.B. the transfer of funds from the customers' accounts to the merchant account is not instantaneous. All transactions, with the SET_FOR_CAPTURE status, are settled at the end of the working day. This allows for the VOID – API Operation (section 8) to be performed before that time. The REFUND – API Operation (section 8) is used to recompense customers after the settlement process.

Only the full authorised amount of the transaction can be captured, based on the amount authorised in the AUTH API operation. Partial captures are currently not supported.

6.2 3DS Redirection

Some card issuers or acquirers require additional customer authentication, known as 3D Secure. The customer's browser must be redirected to the card issuer's site. This adds an extra security layer to the transaction.

In this case, the Redirection Response (section 6.10) is returned, which contains the parameter *redirectionUrl*. The merchant site must redirect the customer browser to the that URL. The customer completes the required information to confirm payment.

- If identification is successful, the UniversalPay Gateway processes the payment.

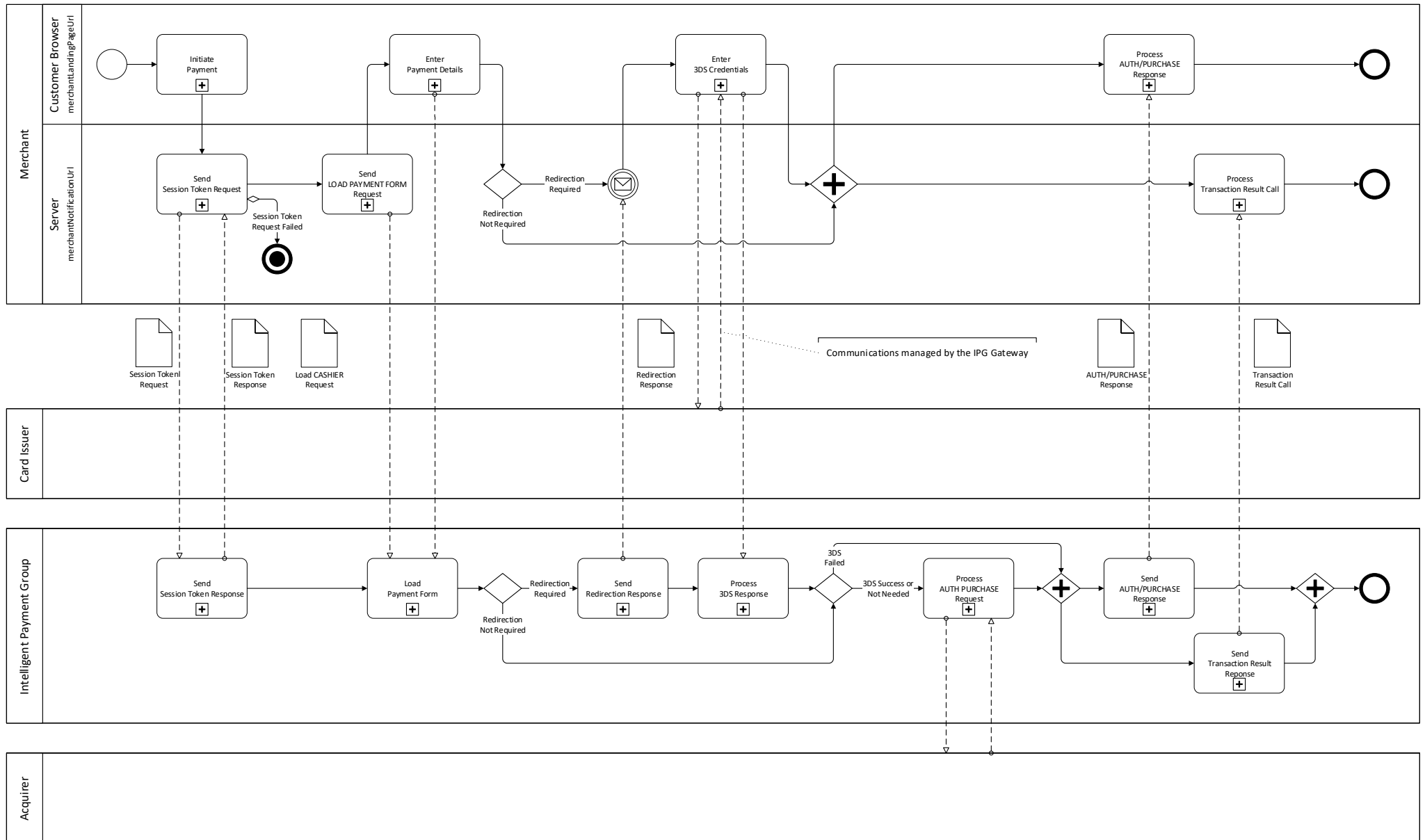
The customer is redirected back to merchant URL provided in the *merchantLandingPageUrl* parameter, specified in the Session Token Request (section 6.4).

- If payment is not authorised, the UniversalPay Gateway does not process the payment and the transaction terminates at this point.

The customer is redirected back to merchant URL provided in the *merchantLandingPageUrl* parameter specified in the Session Token Request (section 6.4).

The UniversalPay Gateway sends a Transaction Result Call (section 12) to the merchant notification URL, specified in the *merchantNotificationUrl* parameter, given in the Session Token Request (section 6.4).

6.3 Process



6.4 Session Token Request

6.4.1 Format

POST Request

6.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway
merchantTxId	String (50)	N	Transaction identifier or Order ID in merchant system. If NULL, auto-generated by the UniversalPay Gateway
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Auth/Purchase Request (see Section 6.9). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	"AUTH" or "PURCHASE"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
customerId	String (20)	C	Customer identifier in merchant system or identifier generated in the TOKENIZE – API Operation (section 5). <ul style="list-style-type: none"> • Mandatory if payment method is Credit Cards • Optional if other method If NULL, auto-generated by the UniversalPay Gateway
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 Brand Id supplied by the UniversalPay Gateway when the merchant account was set up If not provided a default value will be used
channel	String (enum)	Y	Transaction channel which decides if Virtual Terminal (VT) or Cashier will be used in this transaction Possible values: <ul style="list-style-type: none"> • "ECOM" (for e-commerce type transactions) • "MOTO" (for card not present transactions)
userDevice	String (enum)	N	Type of device used: <ul style="list-style-type: none"> • "MOBILE" • "DESKTOP" • "UNKNOWN" (default value)
userAgent	String (1024)	N	The user agent of the browser from which the transaction was performed
amount	BigDecimal (15.2 or 15.3)	Y	The total transaction amount, including tax, shipping, surcharge and discount amounts
taxAmount	BigDecimal (15.2 or 15.3)	N	Tax amount
shippingAmount	BigDecimal (15.2 or 15.3)	N	Shipping amount

Parameter	Data Type	Mandatory	Description
chargeAmount	BigDecimal (15.2 or 15.3)	N	Surcharge amount
discountAmount	BigDecimal (15.2 or 15.3)	N	Discounts applied
currency	String (enum)	Y	Currency alphabetic code as defined in the ISO 4217 standard
country	String (enum)	Y	The country where the transaction takes place. If this is not known or unavailable, the <i>customerAddressCountry</i> should be used (see below). The value is the alpha-2 code as defined in the ISO 3166 standard
paymentSolutionId	Integer (18)	C	Payment solution identifier in the UniversalPay Gateway – currently the only value supported is 500 - Cards Conditions: <ul style="list-style-type: none"> • Must be provided for Direct API integration • Can be omitted in PCI Compliant Payment Form (see section Appendix C) In this case, all the suitable payment solutions will be offered in the Cashier UI
language	String (enum)	N	The language that the Cashier will be presented in, when loaded to the merchant's iFrame. <ul style="list-style-type: none"> • If provided, the Cashier will be provided in the language requested • If not provided, the Cashier will be provided in the merchant's default language The value must be a 2-letter code as defined in ISO 639-1 standard Currently supported languages are: <ul style="list-style-type: none"> • English en • Hungarian hu • Spanish es • German de • Polish pl • Greek el If an unsupported language is provided, either in the field or as the merchant's default language, English (en) will be used.
s_text1, s_text2... s_text5	String (200)	N	5 Text fields for general use, example "late payment"
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. The accepted values are "true" and "false"
n_num1, n_num2... n_num5	BigDecimal (7.2)	N	5 Numeric fields for general use. Use decimal dot "." and not the comma ",", and avoid thousand separators.
merchantNotificationUrl	String (200)	Y	The merchant's server-to-server communications URL, where the merchant will receive Transaction Result Call (section 12).
merchantLandingPageUrl	String (200)	N	The merchant's page URL that is loaded in the customer's browser, for success/failure messaging to the customer.

Parameter	Data Type	Mandatory	Description
firstTimeTransaction	Boolean	N	Customer's first-time transaction flag, which affects 3D Secure processing. <ul style="list-style-type: none"> • if <i>customerId</i> provided, false • if <i>customerId</i> not provided, true Note: the last case will always be true if the UniversalPay Gateway generates the <i>customerId</i> . Therefore, the customer will always be subject to 3D Secure, if it is set up for the card.
customerDocumentType	String (enum)	N	Type of document used to identify the customer by the merchant, accepted values: <ul style="list-style-type: none"> • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UNIQUE_TAXPAYER_REFERENCE • OTHER
customerDocumentNumber	String (30)	C	Customer document number <ul style="list-style-type: none"> • Mandatory if <i>customerDocumentType</i> provided
merchantReference	String (200)	N	Customer merchant reference
customerFirstName	String (50)	N	Customer first name
customerLastName	String (100)	N	Customer last name
customerSex	String (enum)	N	Customer sex: <ul style="list-style-type: none"> • M (male) • F (female)
customerDateOfBirth	Date	N	Customer date of birth. Format DD/MM/YYYY
customerRegistrationDate	Date	N	Customer registration date on merchant's site. Format DD/MM/YYYY
customerEmail	String (60)	N	Customer email address
customerPhone	String (100)	N	Customer phone number
customerIPAddress	String (39)	N	Customer IP address from where purchase is made. Only IPv4 supported
customerAddressHouseName	String (50)	N	Customer address house name
customerAddressHouseNumber	String (5)	N	Customer address house number
customerAddressFlat	String (5)	N	Customer address flat
customerAddressStreet	String (50)	N	Customer address street
customerAddressCity	String (50)	N	Customer address city
customerAddressDistrict	String (50)	N	Customer address district
customerAddressPostalCode	String (30)	N	Customer address postal code
customerAddressCountry	String (enum)	N	Customer address country The value is the alpha-2 code as defined in the ISO 3166 standard
customerAddressState	String (40)	N	Customer address state
customerAddressPhone	String (100)	N	Customer address phone
customerShippingAddressHouseName	String (50)	N	Customer shipping address house name – can be omitted if same as home address
customerShippingAddressHouseNumber	String (5)	N	Customer shipping address house number – can be omitted if same as home address

Parameter	Data Type	Mandatory	Description
customerShippingAddressFlat	String (5)	N	Customer shipping address flat – can be omitted if same as home address
customerShippingAddressStreet	String (50)	N	Customer shipping address street – can be omitted if same as home address
customerShippingAddressCity	String (50)	N	Customer shipping address city – can be omitted if same as home address
customerShippingAddressDistrict	String (50)	N	Customer shipping address district – can be omitted if same as home address
customerShippingAddressPostalCode	String (30)	N	Customer shipping address postal code – can be omitted if same as home address
customerShippingAddressCountry	String (enum)	N	Customer shipping address country – can be omitted if same as home address The value is the alpha-2 code as defined in the ISO 3166 standard
customerShippingAddressState	String (40)	N	Customer shipping address state – can be omitted if same as home address
customerShippingAddressPhone	String (100)	N	Customer shipping address phone – can be omitted if same as home address
customerBillingAddressHouseName	String (50)	N	Customer billing address house name – can be omitted if same as home address
customerBillingAddressHouseNumber	String (5)	N	Customer billing address house number – can be omitted if same as home address
customerBillingAddressFlat	String (5)	N	Customer billing address flat – can be omitted if same as home address
customerBillingAddressStreet	String (50)	N	Customer billing address street – can be omitted if same as home address
customerBillingAddressCity	String (50)	N	Customer billing address city – can be omitted if same as home address
customerBillingAddressDistrict	String (50)	N	Customer billing address district – can be omitted if same as home address
customerBillingAddressPostalCode	String (30)	N	Customer billing address postal code – can be omitted if same as home address
customerBillingAddressCountry	String (enum)	N	Customer billing address country – can be omitted if same as home address The value is the alpha-2 code as defined in the ISO 3166 standard
customerBillingAddressState	String (40)	N	Customer billing address state – can be omitted if same as home address
customerBillingAddressPhone	String (100)	N	Customer billing address phone – can be omitted if same as home address
payerFirstName	String (50)	N	Payer first name – can be omitted if same as customer
payerLastName	String (100)	N	Payer last name – can be omitted if same as customer
payerEmail	String (60)	N	Payer email – can be omitted if same as customer
payerDateOfBirth	Date	N	Payer date of birth – can be omitted if same as customer. Format DD/MM/YYYY
payerPhone	String (100)	N	Payer phone – can be omitted if same as customer
payerDocumentType	String (enum)	N	Type of document used by the payer to identify himself in merchant's side, accepted values: <ul style="list-style-type: none"> • PASSPORT • NATIONAL_ID • DRIVING_LICENSE • UTR • OTHER Can be omitted if same as customer
payerDocumentNumber	String (30)	C	Payer document number – can be omitted if same as customer <ul style="list-style-type: none"> • Mandatory if <i>payerDocumentType</i> provided
payerCustomerId	String (20)	N	Customer identifier of the payee in the merchant's system – can be omitted if same as customer

Parameter	Data Type	Mandatory	Description
forceSecurePayment	Boolean	N	For Card transactions: <ul style="list-style-type: none"> • if True: forces 3D Secure processing no matter the routing rules • If False, not provided or NULL: the 3D Secure routing rules in the UniversalPay Gateway are used If 3D Secure processing is required, the Redirection Response (section 6.10) is sent
processUnknownSecurePayment	Boolean	N	For Card transactions, this determines how "U" (Unknown) responses from the 3DS process are managed: <ul style="list-style-type: none"> • True: success and AUTH/PURCHASE is requested • False: transaction fails at that stage • NULL: value from 3D Secure routing rules are used
specinCreditCardToken	String (100)	N	Credit Card Token Applies to Credit/Debit Card payments only, see TOKENIZE – API Operation (section 5)
specinProcessWithoutCvv2	Boolean	N	Allows to process Credit Cards transactions excluding security code CVV2 Default: false. This requires prior authorization by the UniversalPay Gateway and acquirer.
bankMid	String (50)	N	If merchant wishes to control which acquirer bank MID will be used for any given transaction, this is the place to put it. This is considered advanced feature only for very specific scenarios.
storeCard	Boolean	N	If False the UniversalPay Gateway does not store the card for the customer If True the UniversalPay Gateway stores card If Null the UniversalPay Gateway stores the card

6.4.3 Example

merchantId=1111111&password=klw74U6yt40mNo&merchantTxId=XYZ123456789ABC&allowOriginUrl=www.merchantsite.com&action=AUTH×tamp=1249751864238&customerId=ABD123&operatorId=brian01&brandId=987654321&channel=ECOM&userDevice=DESKTOP&amount=120&taxAmount=10&shippingAmount=15&chargeAmount=5&discountAmount=10¤cy=GBP&country=GB&paymentSolutionId=500&language=en&merchantNotificationUrl=www.merchantsite.com&merchantLandingPageUrl=www.merchant.com&firstTimeTransaction=Y&customerDocumentType=PASSPORT&customerDocumentNumber=12345678&merchantReference=ABC123546&customerFirstName=John&customerLastName=Smith&customerSex=M&customerDateOfBirth=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&forceSecurePayment=True&processUnknownSecurePayment=True&specinCreditCardToken=45ae201ghy23498FjMj701&specinProcessWithoutCvv2=False&bankMid=01000320_MOTO+EUR+Test+Bank+MID&storeCard=False

6.5 Session Token Response - Processed

6.5.1 Format

JSON

6.5.2 Definition

Parameter	Data Type	Description
result	String (40)	"success"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 6.4)
token	String (40)	One-time use Session Token – hexadecimal string

6.5.3 Example

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

6.6 Session Token Response – Not Processed

6.6.1 Format

JSON

6.6.2 Definition

Parameter	Data Type	Description
result	String (40)	Failure
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 6.4)
errors	Array	List of issues

6.6.3 Example

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

6.7 Load Payment Form Request

The Payment Form, also known as Cashier, is a PCI Compliant Payment Form provided by the UniversalPay Gateway. The Payment Form is used to enter the customer's card details at the point of sale. If the customer's card has not been tokenised and/or is not held on file, the Payment Form will tokenize the card using a process like that shown in section 5 - TOKENIZE – API Operation.

The Payment Form is loaded inside an iFrame through in response to the Load Payment Form Request (detailed below) from the merchant's payment URL, using the Session Token from the AUTH/PURCHASE – API Operation (section 6).

The Payment Form can be customised to suit the merchant's payment screen – see Appendix A - Payment Page Customisation Policy. The customised file is subject to review and sign off from a security perspective. As it needs to be put on the IPG Gateway server to work.

6.7.1 Format

POST Request

6.7.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 6.5)
containerId	String	C	ID of the HTML element to wrap the Payment Form. Required for JavaScript based integration.
successCallback	String	C	JS function name to be called in the case of successful transaction Sample function: function handleSuccess(data){ /* code */ }
failureCallback	String	C	JS function name to be called in case of failed/declined transaction Sample function: function handleFailure(data){ /* code */ }
cancelCallback	String	C	JS function name to be called in case of user cancelled transaction Sample function: function handleCancel(data){ /* code */ }
bannerUrl	String	N	A valid URL of a resource (html page, image) to replace the default footer in the Payment Form, with logo
integrationMode	String (enum)	C	Possible values: <ul style="list-style-type: none"> • Inject • Iframe • Standalone

6.8 Sample Payment Form Call

UAT JavaScript URL: <https://cashierui.test.universalpay.es/js/api.js>

UAT baseUrl: <https://cashierui.test.universalpay.es/ui/cashier>

LIVE JavaScript URL: <https://cashierui.universalpay.es/js/api.js>

LIVE baseUrl: <https://cashierui.universalpay.es/ui/cashier>

Template for Cashier CSS customisation:

<https://cashierui.test.universalpay.es/cashier/css/cashier-customisation-template.css>

See also Appendix C - PCI Compliant Payment Form Code Examples

6.9 Auth/Purchase Request

6.9.1 Format

POST Request

6.9.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response (section 6.5)
freeText	String (200)	N	Merchant free text or comments, if not received in the Session Token Request (section 6.4)
numberOfInstallments	Integer (2)	N	This value must always be 1 The field is included in preparation for recurring payments processing This value will be ignored if it was received in the Session Token Request (section 6.4).
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 Session Token Request (section 6.4).
fraudToken	String (50)	C	Antifraud token, if an antifraud tool has been executed before an analysis identifier is required by payment acquirer. Mandatory for transactions conducted in LATAM countries, and only when the merchant wishes the transaction to be conducted as direct integration (server-to-server), as opposed to browser-redirection based integration.
paymentSolutionId	Integer (18)	C	Payment solution identifier in the UniversalPay Gateway. Mandatory, if not received in the Session Token Request (section 6.4), otherwise ignored.
specinCreditCardCVV	String (5)	C	Credit card CVV, if payment solution is credit card through the ECOM channel. This is configurable in the UniversalPay Gateway.

6.9.3 Example

merchantId=1111111&token=abcde12345abcde12345&numberOfInstallments=1&paymentSolutionId=500&specinProcessWithoutCvv2=False

6.10 Redirection Response

The Redirection Response is sent if:

- *forceSecurePayment* parameter = True, in the Session Token Request (section 6.4), or
- the 3D Secure routing rules held in the UniversalPay Gateway require 3D

output response depends on whether customer redirection is required.

6.10.1 Format

JSON

6.10.2 Definition

Parameter/Label	Data Type	Description
result	String (enum)	"redirection"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 6.4)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 6.4).
txId	Integer (18)	Transaction id in the UniversalPay Gateway.
redirectionUrl	String (URL)	URL to which the customer's browser must be redirected.

6.10.3 Example

```
{"result":"redirection","merchantId":111111,"merchantTxId":"abc123","txId":123,"redirectionUrl":"https://mpi.bank.com/123123123-abc-123123123"}
```

6.11 Auth/Purchase Response – Processed

6.11.1 Format

JSON

6.11.2 Definition

Parameter	Data Type	Description
result	String (40)	"success"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 6.4)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 6.4).
txId	Integer (18)	Transaction id in the UniversalPay Gateway
acquirerTxId	String (100)	Transaction identifier in acquirer system, if acquirer returns it.
amount	BigDecimal (15.2 or 15.3)	Transaction amount, including tax, shipping, surcharge and discount amounts.
currency	String (enum)	Currency alphabetic code as defined in the ISO 4217 standard
customerId	String (20)	Customer identifier in merchant system. If NULL, auto-generated by the UniversalPay Gateway
action	String (enum)	Action executed as provided in the Session Token Request (section 6.4).
pan	String (100)	Customer account value or number used in the transaction. If payment solution is Credit Cards, the Credit Card Token is used (as created in the TOKENIZE – API Operation, section 5), not real card number.
brandId	Integer (18)	Brand Id as received in Session Token Response (section 6.5) or default value if not provided there.
paymentSolutionId	Integer (18)	Payment solution id.
freeText	String (200)	Merchant free text.
language	String (enum)	Customer's preferred language. The value must be a 2-letter code as defined in ISO 639-1 standard This value will be sent to some payment acquirers. So they can show their site to the customer in the language requested. If not provided, language will be the payment acquirer's default.
acquirerAmount	BigDecimal (15.2 or 15.3)	Amount processed by payment acquirer. May be different than initial amount requested.
acquirerCurrency	String (enum)	Transaction currency in the payment acquirer's system. May be different to the transaction currency requested (e. g. if a currency conversion applied). Currency alphabetic code as defined in the ISO 4217 standard
paymentSolutionDetails	JSON block	Specific payment solution content; for card payments, which is the only payment solution supported, this will be the Transaction Authorisation Code as received from acquirer { "authCode":"" }
status	String (enum)	Transaction status: <ul style="list-style-type: none"> • AUTH, if authorised – NOT_SET_FOR_CAPTURE • PURCHASE, if sale made – SET_FOR_CAPTURE • If not authorised or sale not made – DECLINED • Else ERROR if any error happened
errors	String (400)	Any errors that occurred during the successful processing of a transaction

6.11.3 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": "NPOT_SET_FOR_CAPTURE"
}
```

6.12 Auth/Purchase Response – Not Processed

6.12.1 Format

JSON

6.12.1 Definition

Parameter	Data Type	Description
result	String (40)	failure
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 6.4)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 6.4).
txId	Integer (18)	Transaction id in the UniversalPay Gateway
errors	Array	List of issues

6.12.2 Example

```
{"result":"failure","merchantId":1231231,"merchantTxId":"abc-123","txId":123,"errors":["insufficient funds"]}
```

7 CAPTURE – API Operation

A CAPTURE – API Operation is required after an AUTH Request, not a PURCHASE Request.

7.1 Transaction Status

All transactions that can be captured have the status SET_FOR_CAPTURE, which means the payment is authorised, through the AUTH/PURCHASE – API Operation section 6).

The successful CAPTURE – API Operation will set the transaction status to CAPTURED.

A CAPTURE – API Operation can be unsuccessful for two reasons:

- The transaction status is set to ERROR if there was an error in the API Operation
 - usually the result of a communication error between the customer browser and the UniversalPay Gateway
- The transaction status is set to DECLINED if:
 - the card payment was refused by the acquirer, or
 - the 3DS authentication failed

A failure response is sent.

7.2 Funds Transfer

The transfer of funds from the customers' accounts to the merchant account is not instantaneous.

The UniversalPay Gateway CAPTURE – API Operation flags transactions for batch processing later in the day, according to the acquirers' requirements. This batch process informs the acquirer that the funds should be transferred.

The actual transfer of funds is performed by the acquirer.

This delay allows for the VOID – API Operation (section 8) to be performed before the CAPTURE – API Operation. The REFUND – API Operation (section 8) is used to recompense customers after the acquirer's settlement process.

Only the full authorised amount of the transaction can be captured, based on the amount authorised in the AUTH API operation. Partial captures are currently not supported.

7.3 Session Token Request

7.3.1 Format

POST Request

7.3.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway.
originalTxId	Integer (18)	N	Transaction Id in the UniversalPay Gateway.
originalMerchantTxId	String (50)	Y	Transaction identifier or Order ID in merchant system. If NULL, auto-generated by the UniversalPay Gateway
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Capture Request (see Section 7.6). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	"CAPTURE"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
agentId	String (18)	N	Id of the merchant's representative who requested the capture.
amount	BigDecimal (10.2 or 10.3) BigDecimal (15.2 or 15.3)	M	Amount to capture If not equal to the original transaction amount, the CAPTURE action will be refused. Partial captures are not supported.

7.3.3 Example

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

7.4 Session Token Response - Processed

7.4.1 Format

JSON

7.4.2 Definition

Parameter	Data Type	Description
result	String (40)	"success"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 7.3)
token	String (40)	One-time use Session Token – hexadecimal string

7.4.3 Example

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

7.5 Session Token Response – Not Processed

7.5.1 Format

JSON

7.5.2 Definition

Parameter	Data Type	Description
result	String (40)	Failure
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 7.3)
errors	Array	List of issues

7.5.3 Example

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


7.6 Capture Request

7.6.1 Format

POST Request

7.6.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response (section 7.4)

7.6.3 Example

merchantid=1111111&token=abcde12345abcde12345

7.7 Capture Response – Processed

7.7.1 Format

JSON

7.7.2 Definition

Parameter	Data Type	Description
result	String (40)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 7.3)
originalMerchantTxId	String (50)	Transaction identifier or Order ID in merchant system If NULL, auto-generated by the UniversalPay Gateway
originalTxId	Integer (18)	Transaction Id in the UniversalPay Gateway
amount	BigDecimal (15.2 or 15.3)	Amount captured
currency	String (enum)	Currency alphabetic code as defined in the ISO 4217 standard
customerId	String (20)	Customer identifier in merchant system. If NULL, auto-generated by the UniversalPay Gateway
action	String (enum)	“CAPTURE”
pan	String (100)	Customer account value or number used in the transaction. If payment solution is Credit Cards, the Credit Card Token is used (as created in the TOKENIZE – API Operation, section 5), not real card number.
brandId	Integer (18)	The Brand Id supplied by the UniversalPay Gateway when the merchant account was set up
paymentSolutionId	Integer (18)	Payment solution id.
status	String (enum)	Transaction status: <ul style="list-style-type: none"> SET_FOR_CAPTURE if capture action was executed successfully. The transaction now is on a queue to be captured by a batch ERROR if any error happened so capture did not take place
errors	String (400)	Only applies to ERROR transactions. A brief description of error cause.

7.7.3 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”}
```

7.8 CAPTURE Response – Not Processed

7.8.1 Format

JSON

7.8.2 Definition

Parameter	Data Type	Description
result	String (40)	"failure"
merchantid	Integer (18)	Merchant ID received in the Session Token Request (section 7.3)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 7.3).
txId	Integer (18)	Transaction id in the UniversalPay Gateway.
errors	String Array	List of errors.

7.8.3 Example

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

8 VOID – API Operation

VOID – API Operations can only be performed on transactions with a status of NOT_SET_FOR_CAPTURE, after an AUTH API operation. The VOID – API Operation effectively cancels the AUTH API operation.

The status after a successful VOID – API Operation is VOIDED.

8.1 Session Token Request

8.1.1 Format

POST Request

8.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway.
originalTxId	Integer (18)	N	Transaction Id if the transaction to be voided in the UniversalPay Gateway
originalMerchantTxId	String (50)	Y	Transaction identifier or Order ID in merchant system of the transaction that is going to be voided.
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Void Request (see Section 8.4). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	"VOID"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
agentId	String (18)	N	Id of the merchant's representative that requests the void.

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)	"success"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 8.1)
token	String (40)	One-time use Session Token – hexadecimal string

8.2.3 Example

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

8.3 Session Token Response – Not Processed

8.3.1 Format

JSON

8.3.2 Definition

Parameter	Data Type	Description
result	String (40)	"failure"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 8.1)
errors	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	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response (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)	"success"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 8.1)
originalMerchantTxId	String (50)	Merchant transaction identifier or Order ID of the transaction that is going to be voided.
originalTxId	Integer (18)	Transaction Id of the refund transaction in the UniversalPay Gateway
amount	BigDecimal (15.2 or 15.3)	Amount voided.
currency	String (enum)	Currency alphabetic code as defined in the ISO 4217 standard
customerId	String (20)	Customer identifier in merchant system. If NULL, auto-generated by the UniversalPay Gateway
action	String (enum)	"VOID"
pan	String (100)	Customer account value or number used in the transaction. If payment solution is Credit Cards, the Credit Card Token is used (as created in the TOKENIZE – API Operation, section 5), not real card number.
brandId	Integer (18)	The Brand Id supplied by the UniversalPay Gateway when the merchant account was set up
paymentSolutionId	Integer (18)	Payment solution id.
status	String (enum)	Transaction status: <ul style="list-style-type: none"> VOID if void was executed successfully ERROR if any error happened so void did not take place
errors	String (400)	Only applies to ERROR transactions. A brief description of errors

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)	"failure"
merchantid	Integer (18)	Merchant ID received in the Session Token Request (section 8.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the IPG Gateway, if not received in the Session Token Request (section 7.3)
txId	Integer (18)	Transaction id in the IPG Gateway.
errors	String Array	List of errors.

8.6.3 Example

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

9 REFUND – API Operation

The UniversalPay Gateway caters for full or partial refunds. More than one refund can be performed on a single transaction. However, the refund amount cannot exceed the original transaction amount. Therefore, if a partial refund has been made, and a second is attempted that added to the first is more than the original transaction amount, the REFUND – API Operation will result in a failed Refund Response – Processed (section 9.5) with errors. Refunds can only be performed on transactions with the status of CAPTURED.

9.1 Session Token Request

9.1.1 Format

POST Request

9.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway
originalTxId	Integer (18)	N	Transaction Id of the transaction to be refunded in the UniversalPay Gateway
originalMerchantTxId	String (50)	Y	Transaction identifier or Order ID in merchant system of the transaction that is going to be refunded.
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Refund Request (see Section 9.4). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	"REFUND"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
agentId	String (18)	N	Id of the merchant's representative that requests the refund.
amount	BigDecimal (10.2 or 10.3) BigDecimal (15.2 or 15.3)	Y	Amount to refund.

9.1.3 Example

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

9.2 Session Token Response - Processed

9.2.1 Format

JSON

9.2.2 Definition

Parameter	Data Type	Description
result	String (enum)	"success"
token	String (40)	One-time use Session Token – hexadecimal string
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 9.1)

9.2.3 Example

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

9.3 Session Token Response – Not Processed

9.3.1 Format

JSON

9.3.2 Definition

Parameter	Data Type	Description
result	String (40)	"failure"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 9.1)
errors	Array	List of issues

9.3.3 Example

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

9.4 Refund Request

9.4.1 Format

POST Request

9.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 9.2).

9.4.3 Example

merchantId=1111111&token=fghij67890fghij67890

9.5 Refund Response – Processed

9.5.1 Format

JSON

9.5.2 Definition

Parameter/Label	Data Type	Description
result	String (40)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 9.1)
originalMerchantTxId	String (50)	Merchant transaction identifier or Order ID of the transaction that is going to be refunded.
originalTxId	Integer (18)	Transaction Id of the transaction to be refunded in the UniversalPay Gateway
txId	Integer (18)	Transaction Id of the refund transaction in the UniversalPay Gateway
amount	BigDecimal (15.2 or 15.3)	Amount refunded.
currency	String (enum)	Currency alphabetic code as defined in the ISO 4217 standard
customerId	String (20)	Customer identifier in merchant system. If NULL, auto-generated by the UniversalPay Gateway
action	String (enum)	“REFUND”
pan	String (100)	Customer account value or number used in the transaction. If payment solution is Credit Cards, the Credit Card Token is used (as created in the TOKENIZE – API Operation, section 5), not real card number.
brandId	Integer (18)	The Brand Id supplied by the UniversalPay Gateway when the merchant account was set up
paymentSolutionId	Integer (18)	Payment solution id.
status	String (enum)	Transaction status: <ul style="list-style-type: none"> SET_FOR_REFUND if refund was created successfully. Will be automatically executed at a further stage (fulfilment) ERROR if any error happened so refund did not take place
Errors	String (400)	Only applies to ERROR transactions. A brief description of errors.

9.5.3 Example

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


9.6 REFUND Response – Not Processed

9.6.1 Format

JSON

9.6.2 Definition

Parameter	Data Type	Description
result	String (40)	"failure"
merchantid	Integer (18)	Merchant ID received in the Session Token Request (section 9.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 9.1)
txId	Integer (18)	Transaction id in the UniversalPay Gateway
errors	String Array	List of errors.

9.6.3 Example

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

10 GET STATUS – API Operation

The GET STATUS – API Operation provides a method to check the status of a transaction in the UniversalPay Gateway. Although, each API Call defined above provides the resulting status after a transaction has been processed, the merchant may want to reconcile data in their local database to that in the UniversalPay Gateway. The transaction is queried by the Merchant’s transaction identifier or by the transaction identifier the UniversalPay Gateway.

10.1 Session Token Request

10.1.1 Format

POST Request

10.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant id that identifies the merchant in the UniversalPay Gateway.
password	String (64)	Y	Merchant password in the UniversalPay Gateway.
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Status Check Request (section 10.2). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	“GET_STATUS”
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00

10.1.3 Example

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

10.2 Session Token Response – Processed

10.2.1 Format

JSON

10.2.2 Definition

Parameter	Data Type	Description
token	String (40)	One-time use Session Token – hexadecimal string
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 10.1)

10.2.3 Example

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

10.3 Session Token Response – Not Processed

10.3.1 Format

JSON

10.3.2 Definition

Parameter	Data Type	Description
result	String (40)	“failure”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 10.1)
errors	Array	List of issues

10.3.3 Example

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

10.4 Status Check Request

10.4.1 Format

POST Request

10.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant id that identifies the merchant in the UniversalPay Gateway.
token	String (40)	Y	Session Token received in the Session Token Response – Processed (section 10.2)
action	String (enum)	Y	“GET_STATUS”
txId	Integer (18)	C	transaction ID in the UniversalPay Gateway
merchantTxId	Integer (50)	C	transaction ID in merchant’s system

10.4.3 Example

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

10.5 Status Check Response - Processed

10.5.1 Format

JSON

10.5.2 Definition

Parameter	Data Type	Description
result	String (enum)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 10.1)
merchantTxId	String (50)	Merchant transaction identifier or Order ID of the transaction.
txId	Integer (18)	Transaction identifier in the UniversalPay Gateway.
status	String (enum)	The Status of the transaction in the UniversalPay Gateway (see section 4.2 - Transaction Statuses)

10.5.3 Example

```
{“result”:“success”,“merchantId”:111111,“MerchantTxId”:“abc123”,“txId”:546,“status”:“SET_FOR_CAPTURE”}
```

10.6 GET_STATUS Response – Not Processed

10.6.1 Format

JSON

10.6.2 Definition

Parameter	Data Type	Description
result	String (40)	“failure”
merchantid	Integer (18)	Merchant ID received in the Session Token Request (section 10.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 10.1)
txId	Integer (18)	Transaction id in the UniversalPay Gateway
errors	String Array	List of errors.

10.6.3 Example

```
{“result”:“failure”,“merchantId”:111111,“merchantTxId”:“abc123”,“txId”:123,“errors”:["communications failure"]}
```

11 GET AVAILABLE PAYMENT SOLUTIONS – API Operation

The GET AVAILABLE PAYMENT SOLUTIONS – API Operation allows the merchant to check the payment solutions that are available to them, for a given country, currency and brand.

11.1 Session Token Request

11.1.1 Format

POST Request

11.1.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
password	String (64)	Y	Merchant password in the UniversalPay Gateway.
allowOriginUrl	String (253)	Y	Merchant's web page URL that will make the Get Available Payment Solutions Request (section 11.4). Cross-Origin Resource Sharing (CORS) headers will allow only this origin.
action	String (enum)	Y	"GET_AVAILABLE_PAYSOLS"
timestamp	Integer (18)	Y	Milliseconds since 1970-01-01 00:00:00
currency	String (enum)	Y	Currency alphabetic code as defined in the ISO 4217 standard
country	String (enum)	Y	Country alpha-2 code as defined in the ISO 3166 standard
brandId	Integer (18)	N	The Brand Id supplied by the UniversalPay Gateway when the merchant account was set up If not provided a default value will be used

11.1.3 Example

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

11.2 Session Token Response - Processed

11.2.1 Format

JSON

11.2.2 Definition

Parameter	Data Type	Description
token	String (40)	One-time use Session Token – hexadecimal string
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 11.1)

11.2.3 Example

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

11.3 Session Token Response – Not Processed

11.3.1 Format

JSON

11.3.2 Definition

Parameter	Data Type	Description
result	String (40)	"failure"
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 11.1)
errors	Array	List of issues

11.3.3 Example

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

11.4 Get Available Payment Solutions Request

11.4.1 Format

POST Request

11.4.2 Definition

Parameter	Data Type	Mandatory	Description
merchantId	Integer (18)	Y	Merchant ID that identifies the merchant in the UniversalPay Gateway
token	String (40)	Y	Session Token received in the Session Token Response - Processed (section 11.2)

11.4.3 Example

merchantId=1111111&token=fghij67890fghij67890

11.5 Get Available Payment Solutions Response – Processed

11.5.1 Format

JSON

11.5.2 Definition

Parameter	Data Type	Description
result	String (enum)	“success”
merchantId	Integer (18)	Merchant ID received in the Session Token Request (section 11.1)
resultId	String	Hexadecimal string that is to be used in any support request calls to the UniversalPay Gateway Service Desk, should there be an issue with or questions about this request.
data	Array	List of payment solutions available.
data:ID	Integer (18)	Payment solution identifier in the UniversalPay Gateway
data:NAME	String (150)	Payment solution name.

11.5.3 Example

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

11.6 Get Available Payment Solutions Response – Not Processed

11.6.1 Format

JSON

11.6.2 Definition

Parameter	Data Type	Description
result	String (40)	“failure”
merchantid	Integer (18)	Merchant ID received in the Session Token Request (section 11.1)
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. Generated by the UniversalPay Gateway, if not received in the Session Token Request (section 11.1).
txId	Integer (18)	Transaction id in the UniversalPay Gateway.
errors	String Array	List of errors.

11.6.3 Example

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

12 Transaction Result Call

A Transaction Result Call is a secure, server-to-server transmission between the UniversalPay Gateway and the merchant's systems. When an API operation is completed (successfully or not), a Transaction Result Call is sent to inform the merchant about the result and the status of the payment.

The Transaction Result Call is sent to the URL that is provided in *merchantNotificationUrl* in the Session Token Request of the AUTH/PURCHASE – API Operation (section 6.4). Transaction Result Calls to subsequent API Operations, (VOID, REFUND, CAPTURE) will use this same *merchantNotificationUrl* value as it is saved against the original transaction in the UniversalPay Gateway.

If the *merchantNotificationUrl* is not provided in the original transaction, no Transaction Result Call is sent. Merchants must reconcile their payment data by running the detailed transaction reports from the Back-Office.

12.1 Transaction Result Response

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

Parameter	Data Type	Description
merchantId	Integer (18)	Merchant ID that identifies the merchant in the UniversalPay Gateway
merchantTxId	String (50)	Transaction identifier or Order ID in merchant system. If NULL, auto-generated by the UniversalPay Gateway.
txId	Integer (18)	Transaction id in the UniversalPay Gateway.
acquirerTxId	String (100)	Transaction identifier in acquirer system, if acquirer returns it.
amount	BigDecimal (15.2 or 15.3)	Transaction amount, including tax, shipping, surcharge and discount amounts
currency	String (enum)	Currency alphabetic code as defined in the ISO 4217 standard
customerId	String (20)	Customer identifier in merchant system. If NULL, auto-generated by the UniversalPay Gateway
action	String (enum)	Action executed: <ul style="list-style-type: none"> • AUTH • PURCHASE • REFUND • CAPTURE • VOID
pan	String (100)	Customer account value or number used in the transaction. If payment solution is Credit Cards, the Credit Card Token is used (as created in the TOKENIZE – API Operation, section 5), not real card number.
brandId	Integer (18)	The Brand Id supplied by the UniversalPay Gateway when the merchant account was set up If not provided a default value will be used
paymentSolutionId	Integer (18)	Payment solution id.
status	String (enum)	Transaction status: <ul style="list-style-type: none"> • SET_FOR_CAPTURE • NOT_SET_FOR_CAPTURE • DECLINED • CAPTURED • SET_FOR_REFUND • COMPLETED_REFUND • VOID • ERROR
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 initial amount requested.
acquirerCurrency	String (enum)	The payment acquirer's transaction currency. May be different to the transaction currency, e.g. if a currency conversion applied. Currency alphabetic code as defined in the ISO 4217 standard .
country	String (enum)	This is the country where the transaction takes place. The value is the alpha-2 code as defined in the ISO 3166 standard .

Parameter	Data Type	Description
freeText	String (200)	Merchant free text.
language	String (enum)	Customer's preferred language. The value must be a 2-letter code as defined in the ISO 639-1 standard .
errorMessage	String (400)	Only applies to ERROR transactions. It is a brief description of the cause of the error.
paymentSolutionDetails	JSON block	Specific payment solution content; for card payments, which is the only payment solution supported, this will be the Transaction Authorisation Code as received from acquirer <pre>{ "authCode":"" }</pre>

Appendix A Payment Page Customisation Policy

To adhere to payment security standards and PCI compliance, the following regulations apply for payment page customisations:

- Customisation must be limited to:
 - Font used in texts
 - Background colours
 - Text colours
 - Element colours (boxes, shapes etc...)
 - Element styling (e.g. box style, round/sharp edges)
- The following items are not permitted:
 - Elements outside of the existing elements in the payment page;
 - External links
 - Hidden elements
 - Element layout changes (e.g. box positioning in the iframe);

Appendix B Card Type Values

The values are determined by the Issuer Identification Number (IIN) of the customer's card.

Value	Card Type/Payment Solution
100	MAESTRO
200	MASTERCARD CREDIT
300	MASTERCARD DEBIT
400	VISA CREDIT
500	VISA DEBIT
600	VISA ELECTRON

Appendix C PCI Compliant Payment Form Code Examples

C.1 Sample code for redirection based integration

```
<!DOCTYPE html>
<html><head>
<script type="text/javascript">
window.onerror = function myErrorHandler (errorMsg, url, lineNumber) {
    alert ("Error occurred: " + errorMsg + "\nurl: " + url + "\nline: " + lineNumber); return false;
    }
</script>
<style>
label {
    width: 10em;
    display: inline-block;
    margin: 0 0 0.5em 0;
    }
input {
    width: 15em;
    }
</style>
</head>
<body>
    <h3>Redirection example</h3>
    <h4>redirectedFullWindow</h4>
<form method="get" action="https://cashierui.test.universalpay.es/ui/cashier">
    <label>token:</label><input name="token"/><br/>
    <label>merchantId:</label><input name="merchantId" value="666"/><br/>
    <label>paymentSolutionId:</label><input name="paymentSolutionId" value="500"/><br/>
    <input type="hidden" name="integrationMode" value="standalone"/>
    <button type="submit" >Pay </button>
</form>
</body>
</html>
```

C.2 Sample code for JavaScript based integration:

```
<!DOCTYPE html>
<html><head>
<style>
#ipgCashierDiv{
    width: 600px;
    height: 400px;
    border: 1px solid gray;
    margin: 10px;
    }
label {
    width: 10em;
    display: inline-block;
    margin: 0 0 0.5em 0;
    }
input {
    width: 15em;
    }
</style>
<script>document.write ('<script src="https://cashierui.test.universalpay.es/js/api.js?ts=' + Date.now () +
""\><\script>');</script>
<script type="text/javascript">
var cashier = com.myriadpayments.api.cashier();
cashier.init (
    { baseUrl:'https://cashierui.test.universalpay.es/ui/cashier' }
    );
function handleResult (data){
    alert (JSON.stringify (data));
    }
function pay (){
    var token = document.getElementById ("tokenIn").value;
    var merchantId = document.getElementById ("merchantIdIn").value;
    cashier.show (
        {
        containerId:"ipgCashierDiv",
        merchantId: merchantId,
        token:token,
        successCallback: handleResult,
        failureCallback: handleResult,
        cancelCallback: handleResult,
        styleSheetUrl: "/cashier/css/optional-customisation.css"
        }
    );
};
</script>
```

```
</head>
<body>
  <h3>Simple Javascript integration example</h3>
  <h4>simpleJsIntegration</h4>
  <div>
    <label>token:</label><input id="tokenIn"/><br/>
    <label>merchantId:</label><input id="merchantIdIn" value="666"/><br/>
    <button onclick="pay ()">Pay</button>
  </div>
  <div id="ipgCashierDiv"></div>
</body>
</html>
```