IPG Gateway

API Operations Overview v 3.0 May 1, 2019

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

Intelligent Payments Group 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, Intelligent Payments Group will not be held liable for any inaccuracies of any nature, however communicated by Intelligent Payments Group.

Table of Contents

Document Control 3
Document Purpose 3
Intended Audience 3
1 Merchant Integration Methods 4
1.1 Direct API Integration 4
1.2 Hosted Payment Page Integration 4
1.3 Shopping Cart Plugins 4
2 API Operations Overview 5
2.1 TOKENIZE 5
2.2 AUTH/PURCHASE/VERIFY 5
2.3 REFUND 6
2.4 VOID 6
2.5 CAPTURE 6
2.6 TRANSACTION RESULT CALL 6
2.7 GET STATUS 7
2.8 GET AVAILABLE PAYMENT SOLUTIONS 7
3 Gateway Interface 8
3.1 User Acceptance Testing Addresses 9
3.1.1 Production Addresses 9
3.1.2 HTTP Specification 9
3.1.3 Example HTTP Request 9
4 IPG Gateway Back-Office 10
5 API Operations Overview 11
5.1 Process Overview 11
5.2 Process 12
5.3 Transaction Statuses 13
Appendix A API Operations Definitions 14

Document Control

Document Purpose

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

Intended Audience

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

- Merchant business and technology staff
- Shopping Cart Plugin providers

The document defines the external interfaces to the IPG 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 IPG Gateway) in those payment processes.

1 Merchant Integration Methods

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

The IPG Gateway supports three integration methods:

1.1 Direct API Integration

Direct API Integration is designed for the merchant that has a fully functional, PCI Compliant payment environment.

The primary feature of this integration method is the merchant's capability to develop their own payment form, where their customers to input their sensitive payment card information and expose alternate non-card payment methods to their customers.

In this scenario the merchant is using the IPG Gateway to process payment and supporting transactions through their acquirer.

This method is intended for more technologically sophisticated merchants who manage complex systems that provide a full customer shopping experience.

1.2 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 IPG Gateway manages a Level-1 PCI Compliant environment that is certified and regularly audited. The merchant will integrate the IPG 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 IPG Gateway servers. The payment card data will not be exposed to the merchant's system.

The IPG Gateway can also provide other Alternate Payment Methods (APM), non-card payment methods to merchants. The IPG 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 IPG Gateway, where the data is stored securely and confidentially.

1.3 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 IPG 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 IPG 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 IPG Gateway.

2.1 TOKENIZE

The TOKENIZE API Operation is only used by Direct API Integration merchants.

The TOKENIZE API Operation provides a hexadecimal string that represents the customer's payment card in the IPG Gateway. Only Card Token can be used in all other API Operations, not the actual payment card data. The TOKENIZE API Operation is the only operation that accepts real card data.

The payment card details are provided to the IPG Gateway in the TOKENIZE API Operation and stored in the IPG Gateway's own PCI Level 1 compliant environment, in which the card details are encrypted.

The TOKENIZE API Operation is described in the *IPG Gateway – 1 – TOKENIZE* document

2.2 AUTH/PURCHASE/VERIFY

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

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

Depending on the configuration of the merchant's payment form or the IPG Gateway Hosted Payment Page, the AUTH/PURCHASE/VERIFY 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 IPG Gateway.

Direct API Integration merchants must manage the processing differences between the payment methods offered. This may not require the AUTH/PURCHASE/VERIFY API Operation at all.

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

The differences between Direct API Integration merchants and Hosted Payment Page Integration merchants require a different internal AUTH/PURCHASE/VERIFY API Operation process:

1. The Direct API Integration merchants will
    a. Take the customer's payment details
    b. Send the Session Token Request, and receive the Session Token Response
    c. Send an Authorise/Purchase/Verify Request on receipt of the Session Token, and receive the appropriate Response files
2. The Hosted Payment Page Integration merchants will:
    a. Send the Session Token Request, and receive the Session Token Response
    b. Send a Load Payment Form Request
    c. The IPG 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 IPG Gateway Hosted Payment Page processes the payment as selected and returns the appropriate Authorise/Purchase/Verify Response and Transaction Result Call
    f. The merchant's webpage and system will receive and process the response as required

These differences are documented in the *IPG Gateway – 2 – AUTH-PURCHASE-VERIFY – Direct API* and *IPG Gateway – 2 – AUTH-PURCHASE-VERIFY – Hosted Payment Page* documents.

2.3 REFUND

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

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

- By Direct API Integration, 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 IPG 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 the *IPG Gateway – 3 – REFUND* document

2.4 VOID

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

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

- By Direct API Integration, 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 the *IPG Gateway – 4 – VOID* document

2.5 CAPTURE

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

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

- By Direct API Integration, 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 IPG 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 the *IPG Gateway – 5 – CAPTURE* document

2.6 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 IPG Gateway and the merchant's server. In the all the above API Operations the *merchantNotificationUrl* parameter tells the IPG 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 IPG Gateway.

The TRANSACTION RESULT CALL is described in the *IPG Gateway – 6 – TRANSACTION RESULT CALL* document

2.7 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 IPG Gateway to check the status of the transaction in the IPG Gateway.

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

The GET STATUS API Operation is described in the *IPG Gateway – 7 – GET STATUS* document.

2.8 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 IPG 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 the *IPG Gateway – 7 – GET AVAILABLE PAYMENT SOLUTIONS* document.

3 Gateway Interface

Detailed below are the URL access points for all API Calls to the IPG Gateway applications:

- **Session Token Request URL**: All Session Token Requests must be sent to this URL regardless of the API Action type being executed. All Session Token Requests must
    - Contain a valid Merchant ID and Merchant Password in the *merchantId* and *password* parameters, which will have been provided at the time of on-boarding in the IPG Gateway
    - Be received from an IP Address that has been whitelisted in the IPG Gateway, which will have been done at the time of on-boarding
- **Action Request URL**: All Action Requests must be sent to this URL (except the Load Payment Form Request – see below). All Action Requests must
    - Contain the same Merchant ID sent in the Session Token Request
    - Contain the Session Token received in the Session Token Response – Processed
    - Be received from an IP Address that has been whitelisted in the IPG Gateway, which will have been done at the time of on-boarding
      The IP Address does not need to be the same address used in the Session Token Request
- **Payment Form URL**: For Hosted Payment Page integrations, the Load Payment Form Request is sent to its own application. The Load Payment Form Request must
    - Contain the same Merchant ID sent in the Session Token Request
    - Contain the Session Token received in the Session Token Response – Processed
    - Be received from an IP Address that has been whitelisted in the IPG Gateway, which will have been done at the time of on-boarding
      The IP Address does not need to be the same address used in the Session Token Request
- **Back-Office URL**: Is used by a merchant to access the Merchant’s instance of the IPG Gateway Back-Office (see section 4). The application is a public application that is accessed using username and password credentials supplied at the time of on-boarding

3.1 User Acceptance Testing Addresses

Session Token URL:	https://apiuat.test.evomexico.mx/token
Action Request URL:	https://apiuat.test.evomexico.mx/payments
Payment Form URL:	https://cashierui-apiuat.test.evomexico.mx/
Back-Office URL:	https://backofficeui-apiuat.test.evomexico.mx/

3.1.1 Production Addresses

Session Token URL:	https://api.evomexico.mx/token
Action Request URL:	https://api.evomexico.mx/payments
Payment Form URL:	https://cashierui-api.evomexico.mx
Back-Office URL:	https://backofficeui-api.evomexico.mx

3.1.2 HTTP Specification

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

3.1.3 Example HTTP Request

- POST: https://api.evomexico.mx/token
- Content-Type: application/x-www-form-urlencoded
- Content-Length: 415
- 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&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 IPG Gateway Back-Office

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

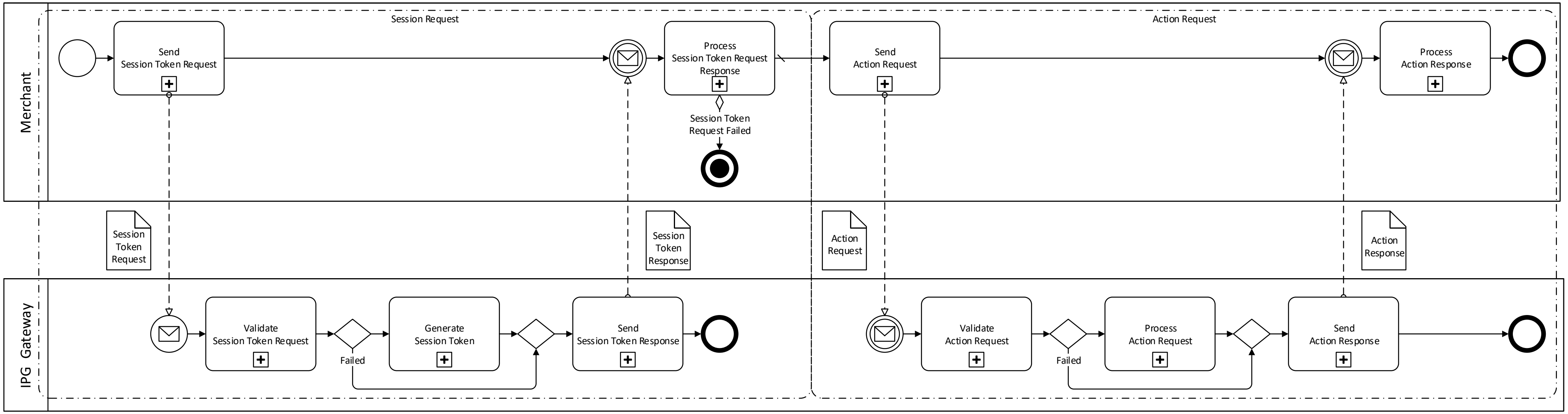
- Transaction Management provides a list of all customers' transactions that can be filtered, sorted and searched; a transaction can be selected from the list to show the full detail
- Refund both full and partial on the initial Purchase transaction amount; multiple refunds can be performed on a single 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 IPG Gateway. The IPG 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 IPG 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 IPG Gateway before the Action Request can be processed by the IPG Gateway.



5.2 Process

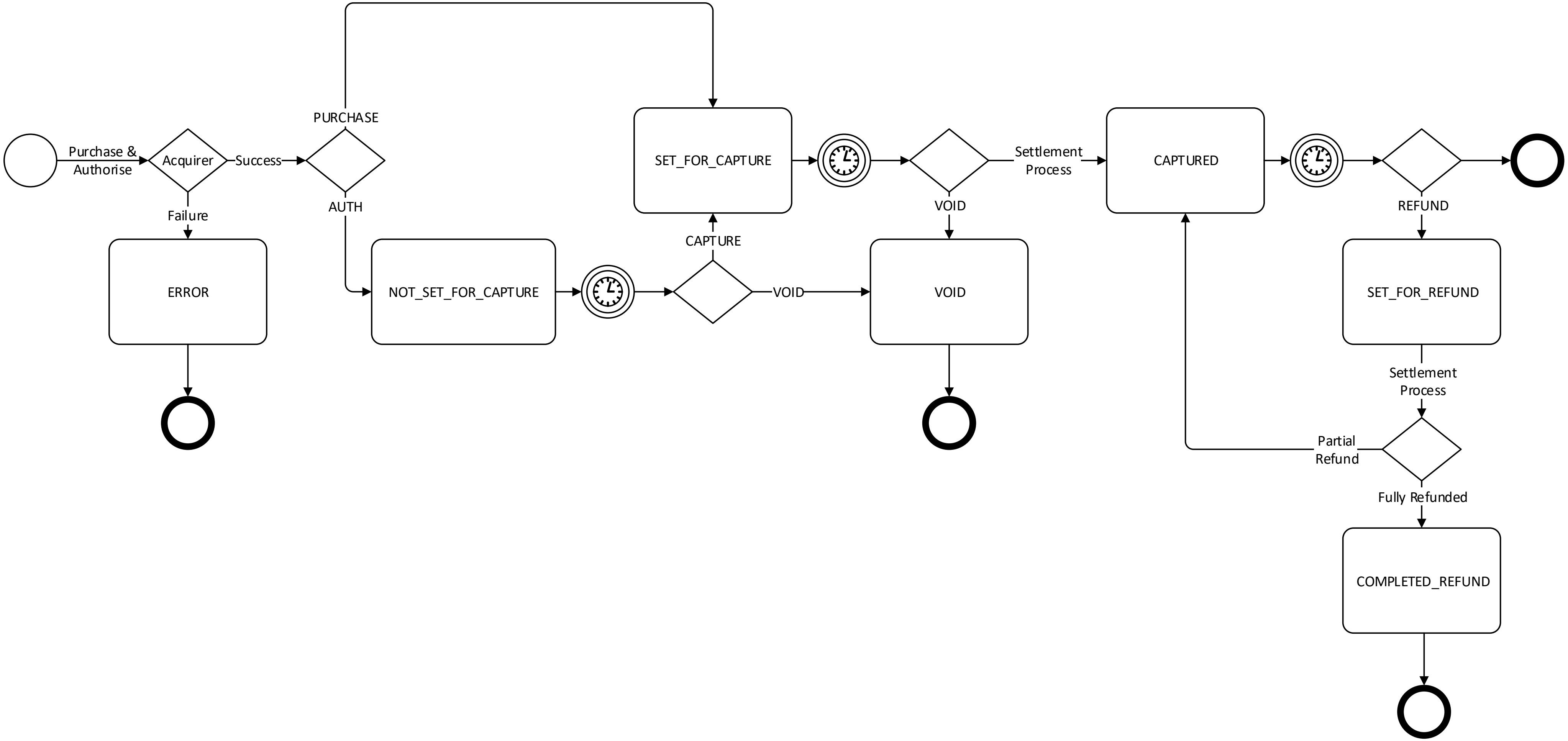
1. The merchant system sends the appropriate Session Token Request for the API Operation to the IPG Gateway
2. The IPG Gateway validates the Session Token Request and authenticates the merchant
   a. If the validation or authentication fails:
      i. The IPG Gateway returns a Session Token Response – 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 IPG Gateway generates a Session Token
      ii. The IPG 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 IPG Gateway
4. The IPG Gateway validates the Action Request and authentications the Action Request to the *action* parameter
   a. If the validation or authentication fails:
      i. The IPG 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 IPG Gateway processes the Action Request
      ii. The IPG 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 IPG 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 IPG 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/VERIFY API Operation (see section 2.2).

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



Appendix A API Operations Definitions

Acronym or term	Description
Processed	In this document, the Response sections that are defined as Processed indicate that the IPG 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 *IPG 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 IPG Gateway. During the API Operation, the IP Address of the requesting server is validated against that stored in the IPG 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 IPG 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 IPG Gateway
Result IDs	The Result ID is included in all Response JSON files, received from the IPG 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 IPG 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 IPG Gateway once, and all subsequent transactions will be associate with that same customer. • If the *customerId* field is left blank/empty, the IPG Gateway will generate a random number identifier that will only be relevant to the API operation in the *IPG Gateway*. Therefore, a single customer can appear in the IPG Gateway database several times. In the IPG 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 IPG Gateway.