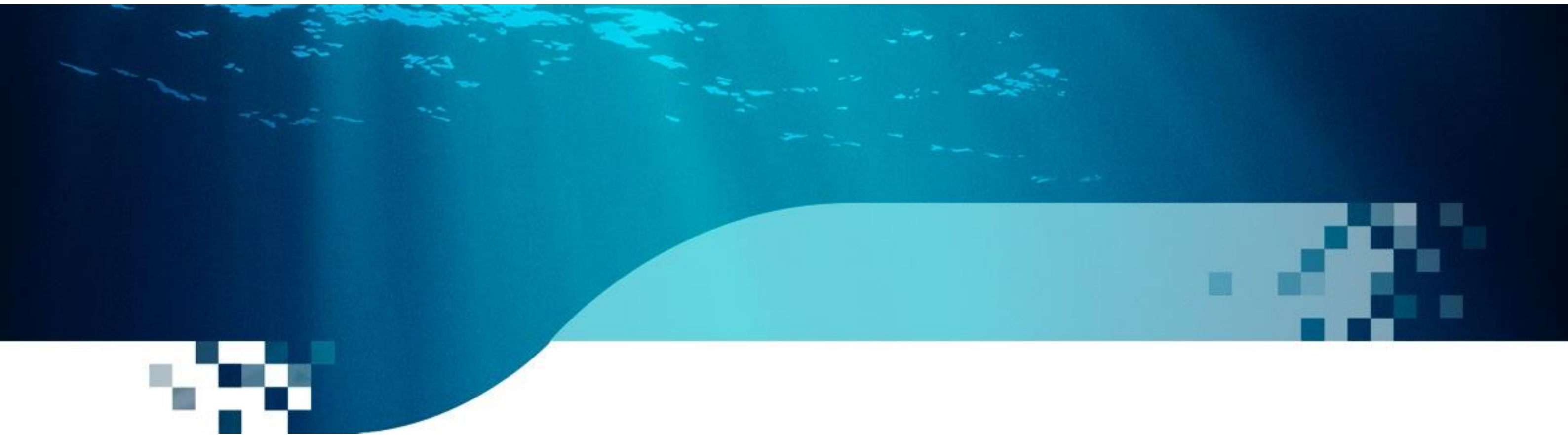


Technical Reference EVO Gateway

API Operations Overview

June 28, 2021

Simplifying Payments AROUND THE GLOBE

Changes in contrast to the previous document version

Chapter no.	Key word	Content of change	Author (initials)	Date
All		New document	CE	06/28/2021

About this document

Document purpose

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

Intended audience

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

- Merchant business and technology staff
- Shopping Cart Plugin providers

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

- Request payment card tokens
- Integrations with 3DS versions 1.0 or 2.0 for payment card authentication
- Submit authorization transactions
- Submit purchases / sales transactions
- Capture (full or partial) funds from customers' accounts as a result of successful authorization transactions
- Void authorized 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 EVO Gateway) in those payment processes.

Explanation of role-related terms

Three main roles exist in the complex scenario of payment processing. Depending on the perspective and situation at hand, two of these can act as customers, or two of them as service providers. To enable a clear distinction, systematic use will be made of the following terms:

Merchant or Client

Uses, in its capacity as merchant or services provider, the products of EVO Payments to settle payments for the goods or services it offers. Is the contract partner and thus the direct "customer" of EVO Payments.

Customer or end customer

Customers who purchase goods from the merchant or services from the service provider; are contractual partners of the merchant / service partner and not of EVO Payments.

EVO Payments

EVO Payments provides services connected with payment processing and acts as the link between the parties, especially the merchant / service provider and other establishments such as card organizations and other institutions involved in the processing of payments.

Terms in the EVO Payments XML language

All terms belonging to the EVO Payments XML language are displayed in “Consolas” font in this document (e. g. **`additionalDetails`**). If such XML terms are separated by line breaks, they do not have a hyphen, as this might lead to misunderstandings in the spelling of the programming terms. Thus, if an XML term is separated by a line break in this document, the line break must be ignored when programming.

Example:
Text in this document text in this document text in this document text in this document text in **`additional`**
`Details` document text in this document text in this Dokument Text in …

Text when programming: **`additionalDetails`**

Note bulleting conventions

Two forms of note bulleting are used in this document:

- Notes providing useful (but not absolutely vital) information:

  > ➡ Please note …

- Notes that must be observed under all circumstances to avoid the risk of functional problems:

  > **!** It is imperative that …

Table of Contents

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 EVO Gateway about which integration method will be employed. This is to assist with correctly configuring the merchant account in the EVO Gateway and for future support purposes.

The EVO 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 EVO Gateway to process payment, integrate with 3DS (version 1.0 or 2.0) 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 e-commerce 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 EVO Gateway manages a Level-1 PCI Compliant environment that is certified and regularly audited. The merchant will integrate the EVO 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 EVO Gateway servers, hidden from the merchant's webpages and servers, and 3DS authentication (version 1.0 or 2.0) is invoked when required. The payment card data will not be exposed to the merchant's system.

The EVO Gateway can also provide other Alternate Payment Methods (APM), non-card payment methods to merchants. The EVO 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 EVO 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 e-commerce web pages by providing streamlined application that integrates with a merchant's website.

The primary feature of this integration method is that the EVO 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, and 3DS authentication (version 1.0 or 2.0) is invoked when required.

The EVO 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 EVO Gateway.

2.1 Tokenize

The **Tokenize** API Operation is only used by Direct API Integration merchants.

The 370188API Operation provides a hexadecimal string that represents the customer's payment card in the EVO Gateway. Only Card Token can be used in all other API Operations, not the actual payment card data. The 370188API Operation is the only operation that accepts real card data.

The payment card details are provided to the EVO Gateway in the 370188API Operation and stored in the EVO Gateway's own PCI Level 1 compliant environment, in which the card details are encrypted.

The Tokenize API Operation is described in the **Technical Reference EVO Gateway – Tokenize**.

2.2 Auth / Purchase / Verify

The **Auth / Purchase / Verify** API Operation combines the Authorize, 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 EVO Gateway Hosted Payment Page.

Depending on the configuration of the merchant's payment form or the EVO 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 EVO 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.

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

01. The Direct API Integration merchants will
    - Take the customer's payment details
    - Send the **Session Token** Request, and receive the **Session Token** Response
    - Merchant, transaction and customer data is used to invoke 3DS authentication if required
    - Send an **Authorize / Purchase / Verify** Request on receipt of the **Session Token**, and receive the appropriate Response files

02. The Hosted Payment Page Integration merchants will:
   - Send the **Session Token** Request, and receive the **Session Token** Response
   - Send a **Load Payment Form** Request
   - The EVO Gateway Hosted Payment Page loads into the merchant's web page (a parameter in the **Session Token** Request)
   - The customer inputs their payment card data or selects an APM
   - Merchant, transaction and customer data is used to invoke 3DS authentication if required
   - The EVO Gateway Hosted Payment Page processes the payment as selected and returns the appropriate Authorize / Purchase / Verify Response and Transaction Result Call
   - The merchant's webpage and system will receive and process the response as required

These differences are documented in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Hosted Payment Page Integration)**.

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 EVO Gateway back-office (section 5).

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 Authorize transactions.

The EVO 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 **Technical Reference EVO Gateway – Refund**.

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 EVO Gateway back-office (section 5).

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 Authorize transactions.

The **Void** API Operation is described in the **Technical Reference EVO Gateway – Void**.

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 EVO Gateway back-office (section 5).

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 Authorize transactions.

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

The **Capture** API Operation is described in the **Technical Reference EVO Gateway – Capture**.

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 EVO Gateway and the merchant's server. In the all the above API Operations the `merchantNotificationUrl` parameter tells the EVO 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 EVO Gateway.

The **Transaction Result Call** is described in the **Technical Reference EVO Gateway – Transaction Result Call**.

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 EVO Gateway to check the status of the transaction in the EVO Gateway.

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

The **Get Status** API Operation is described in the **Technical Reference EVO Gateway – Get Status**.

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 EVO 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 **Technical Reference EVO Gateway – Get Payment Solutions**.

2.9 OneClick

The **OneClick** functionality allows the merchant to present back to a customer a payment card that the customer has previous requested to be saved for future purchases. The customer must actively choose to save the card and has the facility to remove a card from the function. Thus, regulatory requirements are met.

By its nature, **OneClick** functionality is only available to e-commerce merchants and must be initiated by the customer / cardholder. It must not be incorporated into a merchant's own Back-Office function.

2.9.1 Saving a payment card for OneClick

The customer elects to save the card during the payment process in the payment page:

Direct API Integration merchants can use this EVO Gateway functionality by providing the customer choices in their own payment forms, then set the appropriate value in the **`setOneClickValueSettingForCard`** parameter in the Authorize or Purchase Request (see **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)**).

Hosted Payment Page Integration merchants who have been configured to use the function have no integration requirements as it is built into the EVO Gateway Hosted Payment Page as a checkbox.

Shopping Cart Plugins merchants can use this functionality in the same as Hosted Payment Page Integration merchants if the plugin supports **OneClick**.

2.9.2 Get OneClick Payment Methods

Direct API Integration and Hosted Payment Page Integration merchants can retrieve a customer's **OneClick** payment cards and design their own UI. The Get **OneClick Payment Methods** API Operation retrieves the card data from the EVO Gateway that is required to initiate a payment. The EVO Gateway Card Token is returned, not the real payment card number. Notes:

➡ 01. The merchant's webpage must have a method of securely and positively identifying the customer, as the API Operation requires the customer ID as stored in the EVO Gateway.

02. When the customer clicks on the OneClick payment method, as presented by the merchant webpage, the action is to initiate the **Auth / Purchase / Verify** API Operation to initiate an authorize or purchase transaction using the card data retrieved.

03. The Get **OneClick Payment Methods** API Operation is an independent operation, from the Hosted Payment Page, for example, to allow the merchant to use the method as a "Buy-It-Now" type feature on a product page, not just in the checkout page.

The Get **OneClick Payment Methods** API Operation is described in the **Technical Reference EVO Gateway – Get Oneclick Payment Methods API Specification**.

2.9.3 Remove OneClick Payment Method

The Remove **OneClick Payment Method** API Operation compliments the Get **OneClick Payment Methods** API Operation. The customer must have the option to remove a payment card from the **OneClick** method. Therefore, the merchant's solution must provide this function and use the Remove **OneClick Payment Method** API Operation when required.

2.9.4 Load OneClick Buttons

The EVO Gateway provides a pre-designed form containing the customers chosen **OneClick** payment cards using the Load **OneClick** Buttons API Operation.

The Load **OneClick** Buttons API Operation loads the **OneClick Buttons** Form securely into the merchant's webpage in the same way that the Payment Form is loaded for Hosted Payment Page Integration merchants. This is because all the processing is performed on the EVO Gateway servers, removing the need for API Calls when the customer selects a **OneClick** payment card or wishes to remove the card from **OneClick**.

➡ When the cardholder clicks the **OneClick** button to make a purchase / sale, the merchant system must send an **Auth / Purchase / Verify** Request as described in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)** regardless of the integration method employed.

This is one event where a Hosted Payment Page integrated merchant will use the Direct API Integration method instead of loading a hosted payment page:

- If the Hosted Payment Page has been loaded, the same **Session Token** can be used to send the **Auth / Purchase / Verify** Request API Operation
- If the Hosted Payment Page has not been loaded, a **Session Token** will need to be requested as described in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)**

3. Card On File Functionality

Card On File (COF) Transactions are those that are initiating or are initiated from stored payment card data.

By their nature, COF transactions will not have payment card or cardholder authentication data accompanying the transactions. To enable the Schemes and Issuers to assess risk and determine potential fraud accurately, indicators and processes have been introduced to provide greater clarity into transactions using stored credentials.

In all COF scenarios explicit customer consent for the future use of the payment card data must be gained through an initial transaction, where the cardholder is made aware of the potential use of their data. This is usually done during the first Authorize or Purchase transaction, but can also be done in a Verify action.

Subsequent COF Transactions can be initiated by the cardholder or the merchant, all initial transactions must involve the cardholder:

- Cardholder Initiated Transactions (CITs): Anonymized payment card data is presented to the cardholder to select a payment card to initiate a transaction. The cardholder is not required to input the card details. CITs do not require the card security code (CSC/CVV) to be entered and 3DS is not required. The cardholder must be positively identified using suitable authentication in the merchant's shopping website before the card data is presented to the customer.

- Merchant Initiated Transactions (MITs): Transactions are periodically initiated by merchants on behalf of the cardholder with prior agreement from the cardholder. MIT do not require CVV/CSC, which must never be stored, and 3DS is not required.

The following Card On File e-commerce scenarios are offered by the EVO Gateway that require the **`cardOnFile`** prefixed parameters to be included in the **Session Token** Request:

- Merchant Managed Recurring Payments Plans transactions: are MITs where the merchant sends payment requests to the EVO Gateway using stored payment card data. The merchant manages the timing / frequency and amount of the payment request and is simply using the EVO Gateway to execute the payment. These types of transactions use the **`cardOnFile`** and **`mmrp`** parameters (section 3.1).
- EVO Gateway Managed Recurring Payments Plans transactions: are MITs where the EVO Gateway creates payment requests using stored payment card data. The EVO Gateway manages the frequency and amounts on behalf of the merchant set up in the initial transaction. The merchant will use the **Auth / Purchase / Verify** API Operation to create the initial transaction using the **`rpPlan`** parameters. This is the only transaction the merchant submits to the EVO Gateway. These types of transactions use the **`cardOnFile`** and **`rpPlan`** parameters (section 3.2).
- **OneClick** transactions: are CITs where the merchant provides a method for the cardholder to pay for an item or group of items with one click of a button. The payment card data associated with that button is then used to build the **Auth / Purchase / Verify** API Operation **Session Token** Request.

  > ➡ The cardholder must be positively identified in the merchant's website before the **OneClick** buttons are presented to them.

  These types of transactions only use the **`cardOnFile`** parameters

- Stored Credentials transactions: are CITs where the anonymized payment card data is presented. The cardholder can select a card to pre-fill the Payment Form's fields. The CVV/CSC may be entered, but it is not necessary. 3DS processing is also not required.

  > ➡ The cardholder must be positively identified in the merchant's website before the Stored Credentials are presented to them.

  These types of transactions only use the **`cardOnFile`** parameters.

  > ➡ Each scenario must have its own unique initiating transaction. For example, an initial transaction cannot be used to gain the customer consent for a recurring payment plan and **OneClick** series of transactions. Similarly, each recurring payment plan for a cardholder must have its own initiating transaction.

3.1 Merchant Managed Recurring Payment Plans

Merchants may manage their own recurring payment plans with their customers and simply use the EVO Gateway to execute the payments.

A Recurring Payment Plan is an agreement between the merchant and the cardholder, where the cardholder provides explicit consent for a merchant to periodically charge his/her account number for recurring goods or services. These may include payment of charges such as insurance premiums, subscriptions, membership fees, tuition, utility charges or a loan on a large amount purchase.

A Merchant Managed Recurring Payment Plan can be created and maintained using:

- The Direct API integration (section 1.1) only, where the merchant takes the first payment using their own payment form and sends the first and subsequent payment requests using the API operation described in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)**.

- A combination of the Hosted Payment Page integration (section 1.2) and Direct API integration (section 1.1), where the merchant:
  - Takes the first payment using the EVO Gateway's Hosted Payment Page described in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Hosted Payment Page Integration)**, and
  - Sends the subsequent payment requests using the API operation described in the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)**.

  > ➡ In this scenario the Payment Card Token will have been received in the first payment response.

Merchant must complete the **`cardOnFile`** and **`mmrp`** prefixed fields in the **Session Token** Request appropriately for all Merchant Managed Recurring Payment Plan Transactions Requests. These data are required by the Acquirer, Issuers and Card Schemes to recognize that:

- A Recurring Payment Plan is being created (**`cardOnFileType = “First”`**), and
- Subsequent transactions are related to the initiating transaction (**`cardOnFileType = “Repeat”`**).

3.2 EVO Gateway Managed Recurring Payment Plans

The EVO Gateway offers the facility to manage Recurring Payment Plan on behalf of their merchants. This is an automated service that creates payment requests using the cardholder data using the frequency and amount data provided by the merchant in the initial transaction.

The fields prefixed with **`rpPlan`**, in the **Session Token** Request of the **Auth / Purchase / Verify** API Operation (**Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)** and **Technical Reference EVO Gateway – Auth / Purchase / Verify (Hosted Payment Page Integration)**), are provided for the merchant to be able to set up an EVO Gateway Managed Recurring Payment Plan.

Merchant must complete the **`cardOnFile`** and **`rpPlan`** prefixed fields appropriately for all EVO Gateway Managed Recurring Payment Plan Transactions Requests. These data are required by the Acquirer, Issuers and Card Schemes to recognize that a Recurring Payment Plan is being created (**`cardOnFileType`** = **`First`**).

The data must only be sent once in the **Session Token** Request for the transaction that initiates the Recurring Payment Plan. All subsequent payment requests will be generated by the EVO Gateway.

The transaction results for the subsequent payments are returned to the merchant in a Transaction Result call. The EVO Gateway Managed Recurring Payment Plans created by this process can be seen and managed in the **EVO Gateway Back-Office / Virtual Terminal Recurring Payments** menu option.

➡ A **Session Token** Response – Not Processed will be returned with an error stating that the merchant is not authorized for Recurring Payments and the payment will not be processed:

01. If the merchant has not been configured for Recurring Payments in the EVO Gateway and the **`rpPlan`** fields have been completed
02. If **`quickSale = True`** and the **`rpPlan`** fields have been completed

4. Gateway Interface

Detailed below are the URL access points for all API Calls to the EVO 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 EVO Gateway
  - be received from an IP Address that has been whitelisted in the EVO 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 EVO 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 EVO 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 EVO Gateway back-office (see section 5). The application is a public application that is accessed using username and password credentials supplied at the time of on-boarding.

4.1 User Acceptance Testing Addresses

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

4.1.1 Production Addresses

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

4.1.2 HTTP Specification

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

4.1.3 Example HTTP Request

- POST: https://api.evopaymentgateway.com/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%2FevoTesting%3Fdir%3DJCTesting&merchantLandingPageUrl=https://www.merchantsite.com%2FlandingPage&forceSecurePayment=true

5. EVO Gateway back-office

The EVO Gateway back-office compliments the API Operations by providing some API Operations functionality, namely:

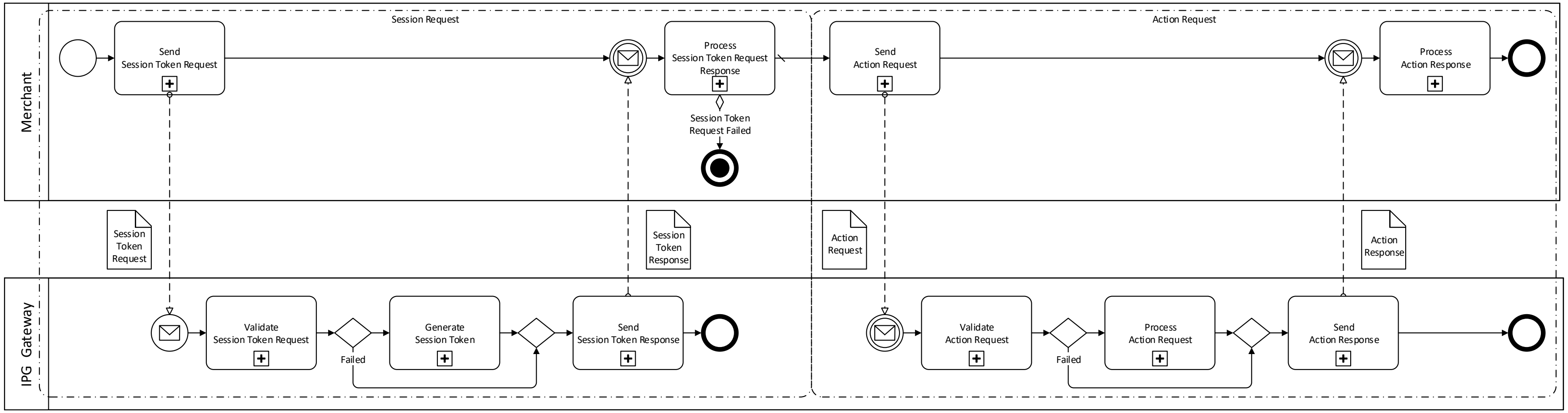
- Transaction Management
  provides a list of all customers' transactions that can be filtered, sorted and searched; a transaction can be selected from the list to show the full detail
- Refund
  both full and partial on the initial Purchase transaction amount; multiple refunds can be performed on a single transaction up to the full amount
- Void
  for both Authorize and Purchase transactions
- Capture
  both full and partial on the initial Authorize transaction amount; multiple captures are not yet supported
- Recurring Payments Management
  allows for plan amendment and changes to the payment card used in the plan
- Recurring Payments Scheme
  allows for the creation and management of template recurring payment plans that are offered to the merchant's customers
- 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 EVO Gateway. The EVO Gateway Back-Office provides for an initial or permanent solution to customer transaction management.

6. API Operations overview

6.1 Process overview

Shown below is a generic view of how all EVO 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 EVO Gateway before the Action Request can be processed by the EVO Gateway.



6.2 Process

01. The merchant system sends the appropriate **Session Token** Request for the API Operation to the EVO Gateway
02. The EVO Gateway validates the **Session Token** Request and authenticates the merchant
    - If the validation or authentication fails:
        - The EVO Gateway returns a **Session Token** Response – Not Processed to the merchant system
        - The merchant system must process the error
        - The Process Terminates Here
        - The merchant must rectify the issues and submit a new **Session Token** Request
    - If the validation and authentication succeed:
        - The EVO Gateway generates a **Session Token**
        - The EVO Gateway returns a **Session Token** Response – Processed to the merchant system that contains the **Session Token** in the **`token`** parameter
03. The merchant system sends the required **Action** Request for the API Operation to the EVO Gateway
04. The EVO Gateway validates the **Action** Request and authentications the **Action** Request to the **`action`** parameter
    - If the validation or authentication fails:
        - The EVO Gateway returns an **Action** Response – Not Processed to the merchant system
        - The merchant system must process the error
        - The merchant must rectify the issues and submit a new **Session Token** Request, i.e. restart the process from the beginning
    - If the validation and authentication succeed:
        - The EVO Gateway processes the **Action** Request
        - The EVO 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 EVO Gateway processes. The merchant system must react appropriately

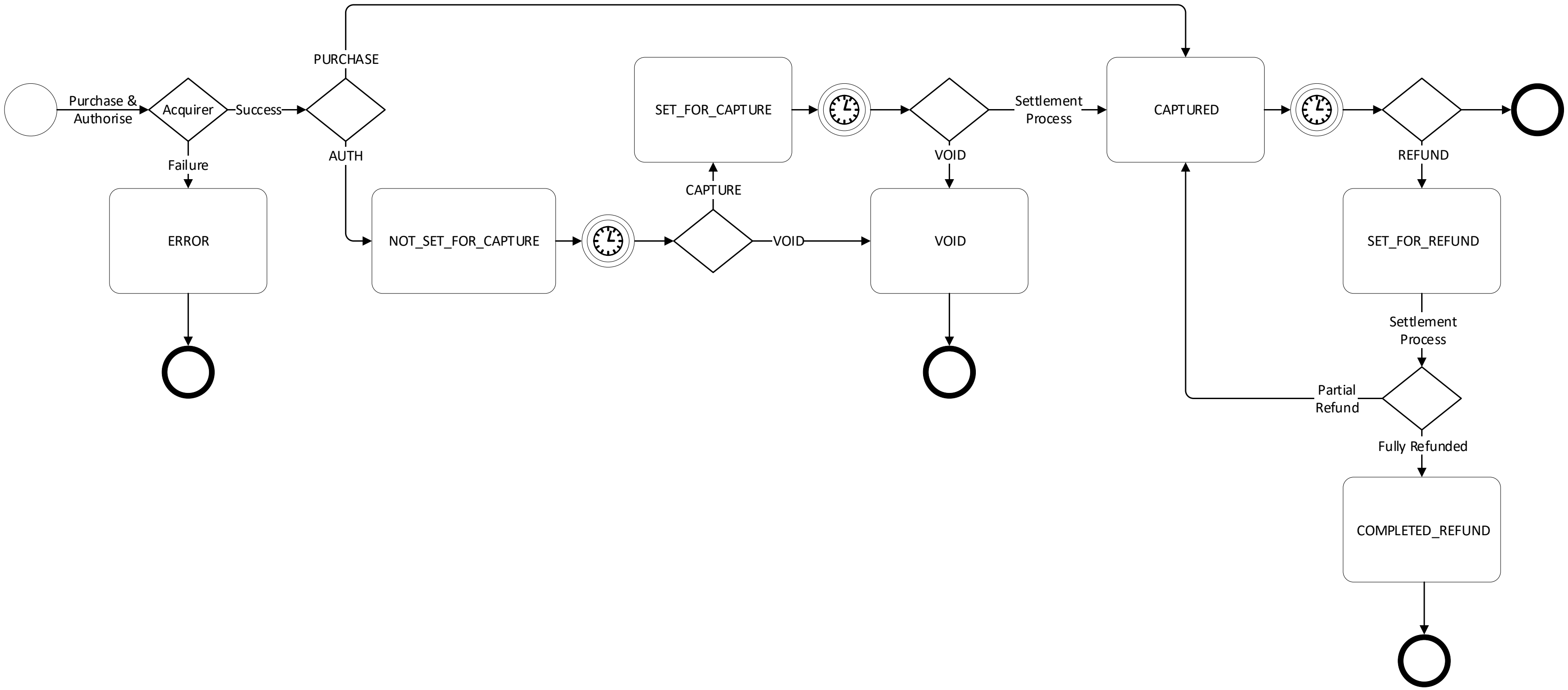
          For some API Operations a **Transaction Result Call** will be sent to the merchant's servers, provided in the **`merchantNotificationUrl`** parameter.

6.3 Transaction Statuses

Payment Transactions in the EVO 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:



7. 3rd Domain Secure Authentication

7.1 Version 1.0

3rd Domain Secure (3DS) is used by the Issuing Banks to provide an additional Authentication Layer to prevent payment card fraud and misuse. The EVO Gateway is integrated with a number of third-party suppliers to present a 3DS challenge window to the cardholder, at the point of sale, when required.

Integration is based on the following key factors:

- The Issuing Bank supports 3DS
- The merchant is enrolled in the scheme
- The transaction meets criteria that require 3DS authentication
- Is 3DS enforced in the EVO Gateway, a choice by the Acquirer based on the merchant's preference and risk

If 3DS is enabled / required when the payment card data is received, the EVO Gateway identifies redirects the customer's browser to the Issuing Bank's Customer Authentication Window (CAW). The cardholder is required to input their security data, registered with the Issuing Bank. This is processed by the Issuing Bank, and neither the EVO Gateway nor the merchant's webpages can detect or read the data input by the cardholder.

If a successful response is received from the Issuing Bank's authentication processes, the payment process continues to the authorization of the transaction. If a failed response is returned, the transaction fails and no authorization attempt is made by the EVO Gateway. The appropriate **Auth / Purchase / Verify** Response is returned to the merchant's webpage, and a matching **Transaction Result Call** follows.

7.2 Strong Customer Authentication (3DS Version 2.0)

Strong Customer Authentication (SCA) is in the process of being implemented to strengthen the authentication of a cardholder at the point of sale. 3DS Version 2.0 is the upgrade to Version 1.0 in support of this initiative.

SCA is defined as 'authentication based on the use of two or more elements categorized as:

- Knowledge: something only the cardholder knows, such as Pass-Phrase, PIN or Password, etc.
- Possession: something only the cardholder possesses, such as providing a one-time password (OTP) to a cardholder's registered a mobile telephone, or reading a hardware token on the cardholder's device
- Inherence: something the user is, such as facial or fingerprint recognition.

SCA has required the integration of the new flows to the Acquirers and Issuing Banks' existing Payment Processes. The EVO Gateway has implemented these new processes on behalf of the merchants. However, additional data is required from the merchants in the **Auth / Purchase / Verify** API Operation to enable the new processes, which will be detailed in updates to the **Technical Reference EVO Gateway – Auth / Purchase / Verify (Direct API Integration)** and **Technical Reference EVO Gateway – Auth / Purchase / Verify (Hosted Payment Page Integration)**.

Appendix A: API Operations Definitions

Acronym or term	Description
Processed	In this document, the Response sections that are defined as Processed indicate that the EVO 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 EVO 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 resubmitted.
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 EVO Gateway. During the API Operation, the IP Address of the requesting server is validated against that stored in the EVO 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 EVO 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 EVO Gateway
Result IDs	The Result ID is included in all Response JSON files, received from the EVO 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 EVO Gateway.

Acronym or term	Description
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 EVO Gateway once, and all subsequent transactions will be associate with that same customer. - If the **CustomerId** field is left blank / empty, the EVO Gateway will generate a random number identifier that will only be relevant to the API operation in the EVO Gateway. Therefore, a single customer can appear in the EVO Gateway database several times. In the EVO 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 EVO Gateway.