API CERTIFICATION GUIDE

Instructions for use

Medellín, August, 2025

Version 1

CONFIDENTIAL USE

Document Statement

Document Statement This document was prepared by, for, and shall remain under the property of Evertec® for its confidential use.
The client agrees, by their acceptance or use of these documents, to return them at the request of Evertec® and not to reproduce, copy, lend, or otherwise disclose or dispose of their contents, directly or indirectly, and not to use them for any purpose other than that for which they were specifically prepared.

CONTENT

Note: This document is intended to guide the client in configuring the API Gateway service of PlacetoPay. The public documentation for the configurations can be found at the following link: Placetopay Docs. You should navigate to the “API Services” section and select “Gateway.”

Once the integration with Evertec PlacetoPay is completed, a certification process is required, where various aspects are evaluated to ensure the correct functioning of the integration and to protect data integrity. Therefore, test links and data must be free of programming errors.

Below are the different aspects to consider in the certification process that will be evaluated with the PlacetoPay API service.

1. TRADE ACTIVITY

It is verified that the website / App / Portal is only processing products and/or services related to the commercial activities established at the beginning of the negotiation. If the site processes products not permitted by Evertec PlacetoPay, certification will not be granted.

2. PLACETOPAY LOGOS

The system where the integration was carried out must display the PlacetoPay logo to the user. This logo can be obtained from one of the following URLs:

- https://static.placetopay.com/placetopay-logo.svg
- https://static.placetopay.com/placetopay-logo-dark- background.svg
- https://static.placetopay.com/placetopay-logo-square.svg
- https://static.placetopay.com/placetopay-logo-square-dark- background.svg
  Additionally, it must contain a hyperlink to our main informational page: https://placetopay.dev/.

It is also recommended to add the logos of the payment networks available to process payments so that customers are aware of the payment methods enabled for the business.

3. FREQUENTLY ASKED QUESTIONS (FAQ)

A Frequently Asked Questions (FAQ) section must be included in the site, mentioning payments through Evertec PlacetoPay. If the business does not have an FAQ section, the frequently asked questions provided in the documentation must still be included.

Note: The FAQ section must not mention or reference terms such as "online payments" or "online transactions." Additionally, these frequently asked questions will be shared at the beginning of the integration process.

4. TERMS, CONDITIONS AND PRIVACY POLICIES

To ensure transparency and compliance with applicable regulations, the business must establish clear Terms and Conditions as well as a user information management policy that is easily accessible. These should be added in a dedicated section, such as the footer, etc. The following guidelines must be met:

- The Terms & Conditions (T&C) must be visible to the user before processing the payment.
- The business must allow users to review and explicitly accept the general conditions before completing the purchase.
- The business is responsible for defining the general terms and conditions that govern its operations.

5. PAYMENT PROCESS

The system must allow the buyer to view the total amount to be paid before proceeding to the implemented payment service. This amount must match the value sent to Evertec PlacetoPay within the transaction messaging.

The business will define the process for breaking down the product or service cost, including taxes and fees charged to the cardholder. This can be done through a form, shopping cart, virtual store, receipt selection, or other methods.

The system must ensure that no additional charges are generated when updating the status of a transaction if the **process** response already indicates a final status. This means that if there is already a final status, a new **query** to verify the payment status should not be made, except when the status is PENDING. In line with this, when providing a payment summary and updating the transaction in the business system, it must be done comprehensively within the database, maintaining traceability and avoiding additional requests based on the status provided by Evertec PlacetoPay.

6. USE OF TAXES

If, according to the country's regulations and business model, the remittance of taxes to banks is mandatory, the merchant must itemize these taxes for the products in the transaction. The base amount, tax type, and tax value must be sent.

If taxes are sent, it is verified that the merchant includes these values in the payment request and within the TAX BREAKDOWN, where they must be displayed to the user.

To use this functionality, each tax type must be sent according to the documentation: Amounts and currencies - Placetopay Docs.

NOTE: If the Fiscal Control Number is enabled and will be generated as part of the Transaction Process, the corresponding Tax Breakdown must be sent. Otherwise, the Fiscal Control Number may be generated incorrectly.

7. SECURITY REQUIREMENTS

The configuration data for the Evertec PlacetoPay connection must be stored as parameters in the database or in a configuration file such as **.ini, .json, .xml,** etc. This is a best practice in programming and makes it easier to update the key when needed.

For services that use JavaScript-based validations, it must be ensured that the operation is not disrupted when accessed from a browser with JavaScript execution disabled. This can be achieved by preventing the page from loading improperly or by implementing server-side validation.

It is recommended to implement AES256 encrypted passwords for user authentication on the merchant's site.

Under no circumstances shall the merchant store or display sensitive cardholder card data. The data that are considered sensitive are:

- **Credit card number**
- **CVV (Verification Code).**

8. PAY BUTTON CONTROL AND RETURN HANDLING

Multiple user requests must be prevented and controlled in case the service takes too long to respond when the pay button is pressed. The button must be disabled while the payment is being processed.

Additionally, once the payment is completed, if the user clicks a button to return to the merchant's website/app/portal, validation must be performed to ensure the payment is not processed again. This action prevents duplicate payments. This topic is further explained in section 15 of this guide.

9. ACCOUNT VALIDATION – ACH (DIRECT DEBIT TO BANK ACCOUNTS)

According to **NACHA** regulations, ACH payments (direct debits from bank accounts) must validate the bank account at least the first time. To meet this requirement, the
merchant can use either an external account validation service or Evertec PlacetoPay's own validation service, described in the following paragraphs.
Evertec PlacetoPay provides an in-house account validation service, which offers two types of validation:

- **Instant validation:** Provides an immediate verification status, streamlining the process for both the user and the merchant.
- **Micro-deposit validation:** This method takes **1 to 2 days**, as small deposits are made to the user's bank account for confirmation. During this period, the account remains in **pending status** until verification is complete. (This is further detailed in **section 10.2** of this guide.)

This service allows the user to validate their bank accounts using the **CreateSession** method.

With this method, a URL is generated containing the session with the account validator, which is then displayed to the user for validation. Once the user completes the account validation, they will be redirected to the URL provided in the “returnUrl” parameter. To check the validator session status, a query must be executed using the **CheckSessionStatus** method.

Once the account is validated, a **“verificationCode”** (Validation Signature) will be generated. This value must be sent within the Payment Request.

Additionally, after the account is validated, further validations may be performed if necessary (for this, the Validate Existing Account method should be used). Another option is for the merchant to securely store the payer’s bank account details, bank code, routing number, and account type along with the payment method token for future transactions.

Once these validations are completed and a transaction needs to be processed, the request must be sent as follows:

```
   {
    "auth": {},
    "payer": {
        "name": "Test",
        "surname": "Test",
        "email": "pruebasp2p.juanserna@gmail.com",
        "mobile": 3006108300
    },
    "payment": {
        "reference": "Test_ach_1",
        "description": "Pruebas ",
        "amount": {
            "currency": "USD",
            "total": 20
        }
    },
    "instrument": {
        "account": {
            "bankCode": "021502011",
            "bankName": "FIRSTBANK (PUERTO RICO)",
            "accountType": "CCD",
            "accountNumber": "4111111111111111",
            "franchise": "_021502011_",
            "verificationCode": "eyJhbGciOiJSUzI1NiJ9"
        }
    },
    "ipAddress": "127.0.0.1",
    "userAgent": "Testing"
}
```

Note: It is essential to ensure that the retrieved data is stored in an encrypted manner. This procedure is crucial to protect sensitive information and prevent risks associated with unauthorized access, data breaches, potential fraud attempts, and other security threats.


```
        "account": {
            "bankCode": "221571473",
            "bankName": "FIRSTBANK (PUERTO RICO)",
            "accountType": "DDA",
            "accountNumber": "4111111111111111",
            "franchise": "_221571473_",
            "verificationCode": "…"
}
```


9.1. EVENT WEBHOOK

Since ACH account validations are performed through a session, it is necessary to obtain the validation result. For this, a notification URL (Webhook) must be implemented to receive a JSON structure indicating the session status. The JSON structure will contain the following data:

- Approved sesión


```
{
  "status": {
    "status": "APPROVED",
    "reason": "00",
    "message": "The request has been successfully
approved"
  },
  "requestId":  "53eccc5c-bbb8-
36aea4f0dfc8a791e736",    "signature":
"0cd51ffd7cb58c0a36fb7b926838b1feb6336512" }
```

- Failure due to Maximum Failed Attempts


```
{
"status": { "status": "REJECTED",
"reason": "38", "message": "The
maximum number of failed attempts
has been reached" }, "requestId":
"53eccc5c-bbb8-36ae-a4f0-
dfc8a791e736", "signature":
"0cd51ffd7cb58c0a36fb7b926838b1feb
6336512"}
```


- Expired session


```
{
  "status": {
    "status": "REJECTED",
    "reason": "XD",
    "message": "The request has expired"
  },
  "requestId":  "53eccc5c-bbb8-
36aea4f0dfc8a791e736",    "signature":
"0cd51ffd7cb58c0a36fb7b926838b1feb6336512" }
```

- Session canceled by the user


```
{
"status": {
"status": "REJECTED", "reason":
"?C", "message": "Verification
cancelled by the user"
},
"requestId":"53eccc5c-bbb8
36ae□a4f0dfc8a791e736",

"signature":"0cd51ffd7cb58c0a36f
b7b926838b1feb6336512" }
```


Note: Once the notification is received, follow the steps outlined in the following documentation:

Event Webhook

9.2. LIGHTBOX FOR ACH BANK ACCOUNT VALIDATION

PlacetoPay allows this service to be implemented within an **iframe/lightbox** generated by the merchant. It is essential to validate the messages sent from the iframe for events such as:

Validation Approved

```
{
    "type": "close",
    "payload": {
    "status": {
      "status": "OK",
       "reason": "00",
    },
    "account": {
      "status": {
          "status": "OK",
          "reason": "00",
          "message": "La solicitud ha sido aprobada
exitosamente",            "date": "…"
      },
        "bankCode": "POPULAR",
        "bankName": "Banco Popular",
        "accountType": "SAV",
        "accountNumber": "…",
        "verificationCode": "…"
    },
  } }
```

Validation Canceled

```
{
    "type": "close",
    "payload": {
      "status": {
        "status": "REJECTED",
        "reason": "?C"
        },
      "account": null
    }
  }
```

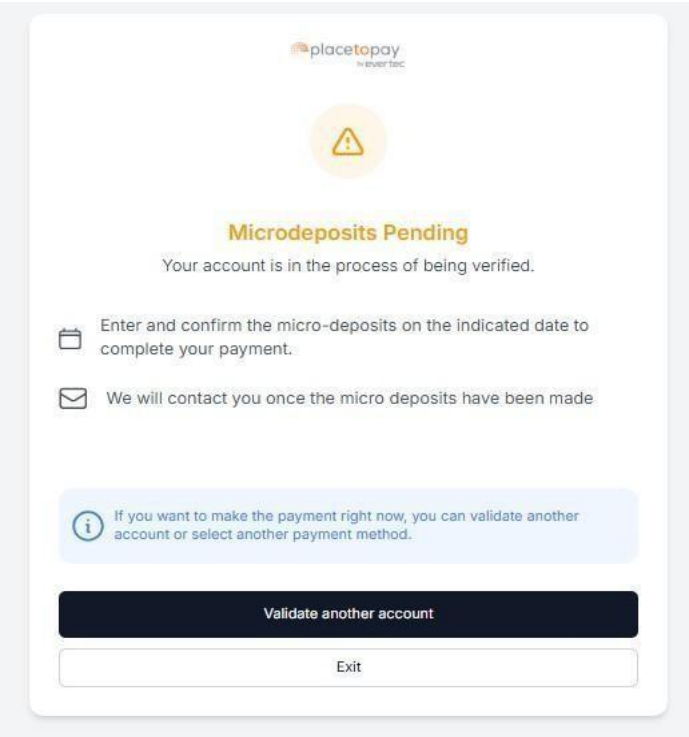
Validation Rejected

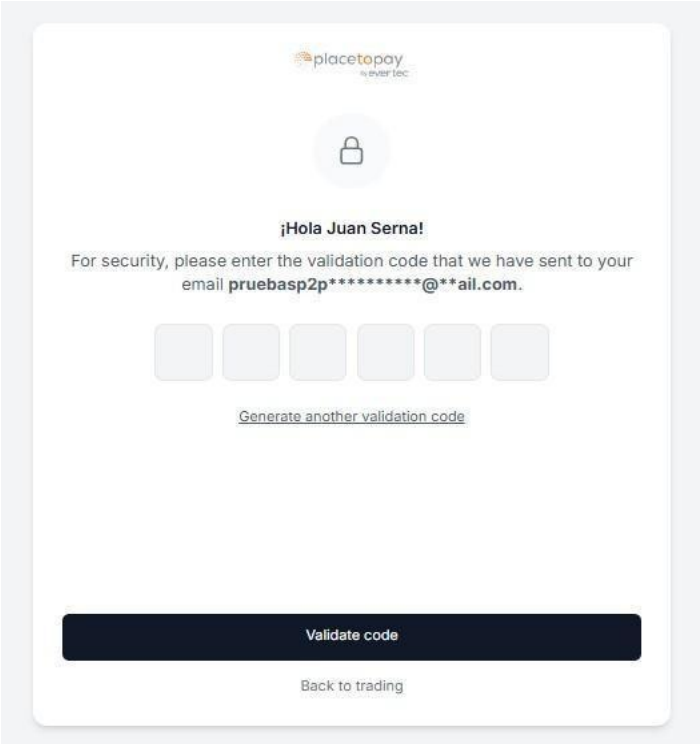
```
{
 "type": "close",
  "payload": {
"status": {
      "status": "REJECTED",
     "reason": "38",
   },
   "account": null
  }
```

NOTE: The verification status details must be captured. If not executed, it will not be possible to update the process if the user completes the validation. Please refer to the official documentation for implementation: Lightbox Support.

9.3. HANDLING OF PENDING STATUS

For scenarios where validation is performed using micro-deposits (which can take 1 to 2 business days), note that when querying the session, the service response may be "rejected." Subsequently, the user has the option to generate a new session to confirm the initial validation. During this process, the service sends an OTP to the user to verify their identity. Below is a detailed user flow for this type of scenario.


placetopay
by evertec
Microdeposits Pending
Your account is in the process of being verified.
Enter and confirm the micro-deposits on the indicated date to complete your payment.
We will contact you once the micro deposits have been made
If you want to make the payment right now, you can validate another account or select another payment method.
Validate another account
Exit


placetopay
by evertec
¡Hola Juan Serna!
For security, please enter the validation code that we have sent to your email pruebasp2p**********@**ail.com.
Generate another validation code
Validate code
Back to trading

Hello Juan Serna! Your confirmation code is:

700957

Sincerely,
Placetopay | ACH Account Verification TEST
placetopay
by evertec

placetopay
by evertec

Successful!

The verification was successful

••••1111
FIRSTBANK (PUERTO RICO)

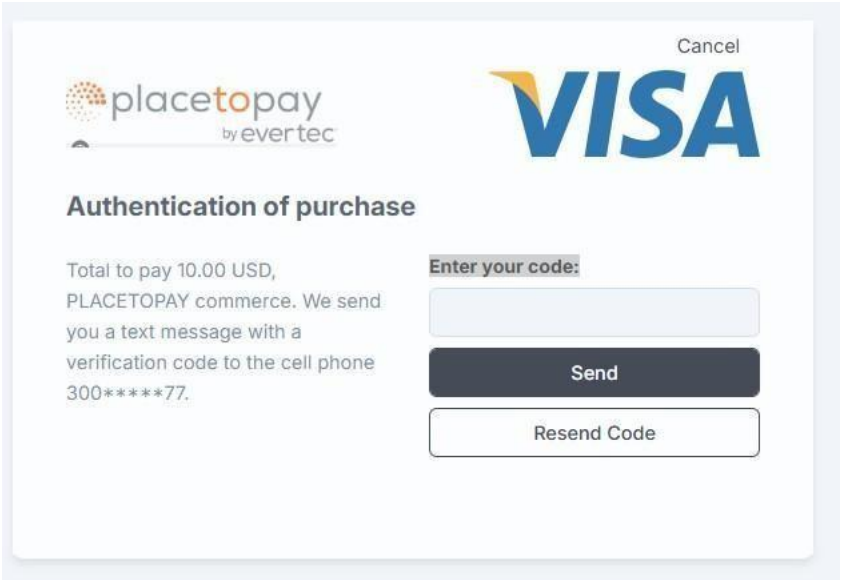
Continue

Once the user returns from the previously initiated validation, the merchant will be able to retrieve the account details if the validation was successfully approved.


```
  {
    "status": {
        "status": "APPROVED",
        "reason": "00",
        "message": "La verificación ha sido exítosa",
        "date": "2025-01-17T17:14:21+00:00"
    },
    "data": {
        "requestId": "…",
        "account": {
            "bankCode": "221571473",
            "bankName": "FIRSTBANK (PUERTO RICO )",
            "accountType": "SAV",
            "accountNumber": "…",
            "franchise": "_221571473_",
            "verificationCode": "…"
        },
        "expiresAt": "2025-01-17 17:28:45"
    }
}
```


10. 3DS GENERATION AND VALIDATION

The merchant must implement a 3DS authentication flow for the user, either through a modal window or a redirection, according to the initial request for card information validation: **Information Request**.

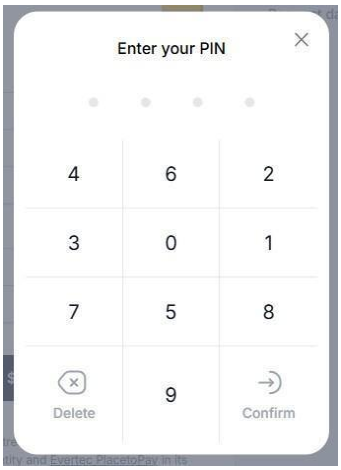


Once the user returns from validation, the **3DSQuery** method must be used to retrieve the authentication information. The response will provide all the details needed for the **Transaction Processing** under the **threeDS** parameter.

Additionally, please refer to the official documentation: 3DS Generation

11. PINPAD-PINBLOCK GENERATION AND VALIDATION

The merchant must generate, request, and provide the user with a PINPAD, either through a modal window or embedded within the interface. This input field must be required, allowing only numeric values, in accordance with the initial request for card information validation: **Information Request**.



Once the user enters the PIN, a PinBlock is generated based on the positions provided by the user. Using the **PinBlock** method, the response will return the PinBlock, which must be sent in **Transaction Processing** under the pin parameter.

Additionally, please refer to the official documentation: PinPad Request.

For transactions using PinPad, the merchant can implement the library using the following documentation: PinPad SDK - Placetopay Docs.

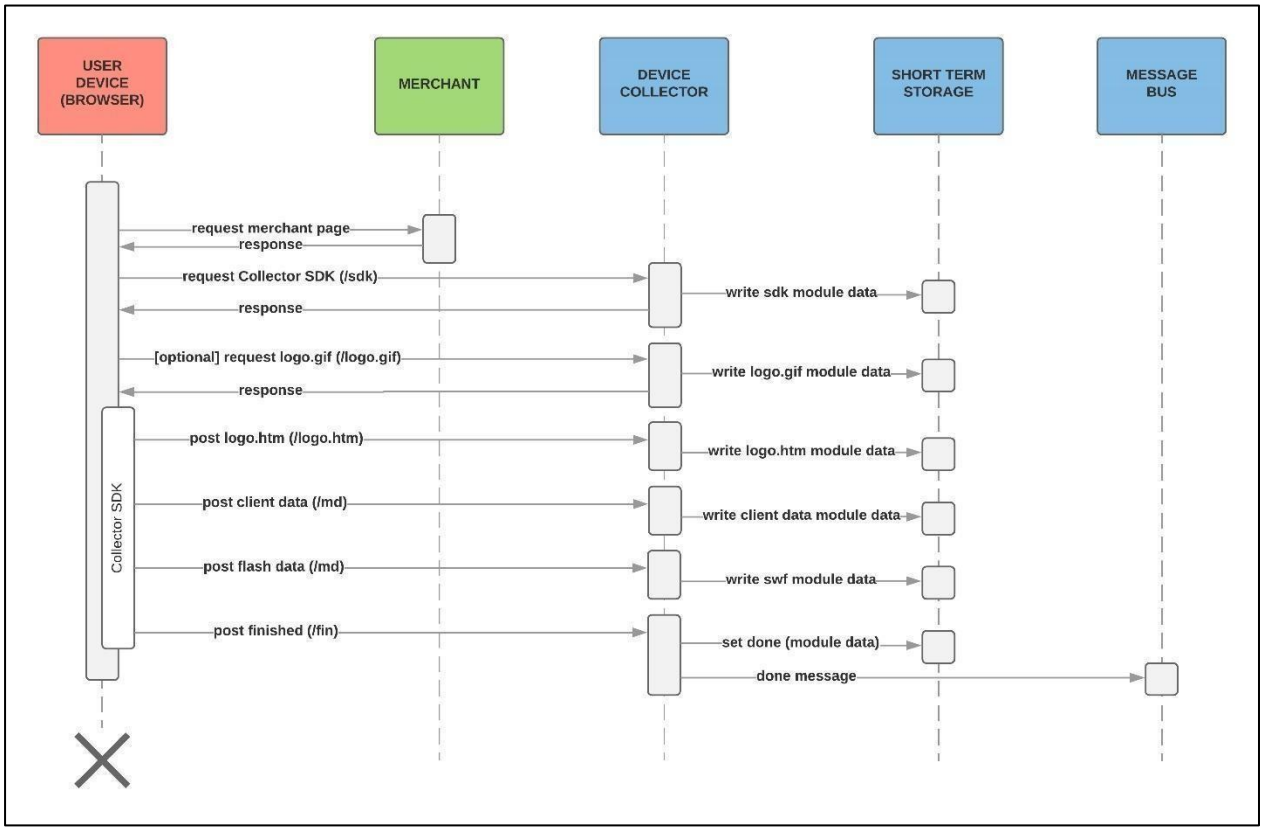
12. KOUNT AUTHENTICATION

The system must send a Kount object in the **Transaction Processing** message, containing the attributes of a session, which is a unique random value per transaction.

```
"kount": {
  "session": "CO485"
}
```

Additionally, please refer to the official documentation: Placetopay Docs and Kount - Placetopay Docs.

This service helps prevent fraud, reduce friction for cardholders, and increase trust in transactions processed through PlacetoPay's fraud prevention system



For integrations requiring Kount security validation for fraud control, the following steps must be completed:

1. After registering for Kount services and receiving a Client ID, add the Kount Web Client SDK to your website to integrate the Device Data Collector (DDC). For more details, see the documentation: Integrating the Web Client SDK for Device Data Collection into Your Website – Kount Developer

2. After retrieving the information, create the device session ID as outlined in: How to
Create a Session ID for the Device Data Collector (DDC) – Kount Developer

3. The Kount session must be created with the following values: **clientId= 201000,**

Environment=TEST (sandbox) o **Environment=PROD (production)** Once the Device Data Collector (DDC) connection is established, send the Kount session in the Kount object.

```
"kount": {
  "session": "ba2ccc0c27d84921ae9034da92ebc74d"   }x
```

4. Below, we share a repository guide to guide you through the kount process: https://github.com/luisfelipegh/kount-ddc?tab=readme-ov-file

13. FIELD VALIDATION

The merchant must send at least the following required fields in the payment request:

Buyer and payer data (Buyer – Payer)

- Name of Buyer-Payer.
- Surname of Buyer-Payer.
- Email of Buyer-Payer.
- Cell phone of Buyer-payer.

Payment Data (Payment)

- Reference (Unique per Transaction).
- Description
- Currency
- Amount

- Taxes (Optional)
- Refund base (Optional)
- Recurring (if the service is enabled)
  ○ Periodicity ○ Interval ○ Next payment ○ Max periods ○ Due date for recurrence ○ Notification URL Card Data (Instrument) ○f Card
  - Number or PAN
  - Expiration
  - CVV ○ ThreeDS
  - Id
  - Enrolled
  - Authenticated
  - Eci
  - Cavv Xid Extra
  - transStatusReas on
  - acsTransId
  - threeDSServerTr ansID ○ Pinpad
  - (if the service is enabled)
  - transactionId
  - positions
  - pinBlock
  - length ○ Pin ○ Account (if the service is enabled)
  - BankCode
  - BankName

- AccountType
- AccountNumber
- Franchise
- Verificationcode ○
- Kount ○ AVS (if the service is enabled)
- Session
- Type
- Additional Data ○ IP ○ UserAgent

Important: Each field must contain consistent information. Additionally, please refer to the official documentation for any changes: Placetopay Docs

- The system requires the user to enter the card number as a mandatory field, accepting only numeric values with a length between 13 and 19 characters, validating the entered value through the Luhn algorithm.
- The application requires or presents the expiration date as a mandatory field, ensuring that the expiration date is not expired and only accepts values from January to December (01 to 12, always two digits). The maximum selection allowed is 10 years from the current date.
- The system requires the user to enter the CVV2 as a mandatory masked field, accepting only numeric values with a maximum length of 4 characters (American

  Express allows up to 4 characters, while Visa, Mastercard, and Discover allow only 3).
- The reference must have a maximum length of 32 alphanumeric characters and must not contain or store the card number or special characters, being unique for each request.
- For buyer and buyer-payer data, each of the fields being sent to Evertec Placetopay must be validated as required at the time the user is entering the information.
- The first and last name fields must not allow numbers or special characters, but must allow accents, spaces, and the letter Ñ. For companies, only the legal business name should be used, and numbers are allowed.
- Numeric fields such as mobile or phone numbers must not allow letters or special characters.
- The email field must have a valid structure: [user] @ [domain].[originType]. [extension].
- The IP address and browser agent must belong to the end user's device.

In the payment process, the buyer is not always the cardholder, so this information must be considered when sending data to Evertec Placetopay.

Note: Asking for card ownership is considered a best practice and is optional. However, if the merchant decides not to implement this, they must ensure that they always send the payer or cardholder's information.

14. IDEMPOTENCY CONTROL (DUPLICATE PAYMENTS)

The validation to avoid duplicate transactions must be performed during the Payment

Process Flow, this process can be done by sending the Idempotency Key

“IdempotenceKey", which is a unique value defined by the merchant for each transaction and must be included in the Payment Request.

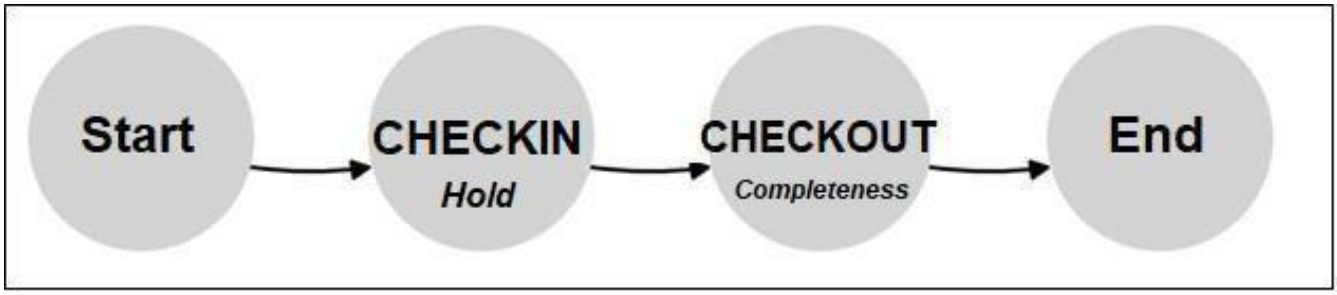
```
{ ...
"idempotenceKey": "ABCD1234",
"instrument": {
...
}
```

For more details, please refer to the official documentation: Idempotence Control - Placetopay Docs

NOTE: The Merchant must generate and send the IdempotenceKey parameter with a unique value based on the Business Logic for each transaction. The system will prevent transactions with the same IdempotenceKey from being processed.

15. PREAUTHORIZATION PAYMENT PROCESSING

For processing preauthorized payments, the user/cardholder must complete the process by reserving the requested amount on their credit card. Once the reservation is made, this amount can be confirmed, modified, or canceled.



Note: Please note that this item applies if the Preauthorization service was acquired.

15.1. CHECKIN

It is used as a security deposit to use a good or a service. To reserve the amount, it must be sent in the request as follows:

```
"action": "checkin",
"payment": {
       "reference": "pay_checkin_1",
       "description": "Payment with pre-auth",
       "amount": {
              "currency": "USD",
              "total": 10
         }
    }
```

15.2. CHECKOUT

To confirm / capture the preauthorized amount from the check-in, the following information must be sent:

```
"internalReference":     11012331,  //Internal      Reference  Code
"authorization": "000000", //Authorization number provided by the financial
institution.     "amount": {
        "currency": "USD",
        "total": 15
    },
    "action": "checkout"
```

Note: The internalReference value is delivered on the service response.

15.3. CANCEL A PREATHORIZATION

To cancel a previously authorized check-in, the action on the checkout must be sent with a value equal to 0.

```
"internalReference": 11012331,
"authorization": "000000",
    "amount": {
        "currency": "USD",
        "total": 0
    },
    "action": "checkout"
```

The preauthorization is cancelled and releases the retained amount from the previous sessions. For more information regarding preauthorization: Placetopay Docs

15.4. REATTEMPTS CAPTURE (CHECKOUT)

In case of encounter an error or a rejection at the moment of capturing the amount of a preauthorized transaction, the merchant must make a maximum of 3 reattempts to capture. We recommend retrying those reattempts each 60 seconds. In the case that in the third reattempt the error remains, you must cancel the operation sending the checkout “action” on 0 (Canceling a preauthorization).

Date	Amount	Status	Type
2024-04-04 21:18:45	0,00	Approved	Checkout
2024-04-04 21:18:16	201,00	Declined	Checkout
2024-04-04 21:18:12	201,00	Declined	Checkout
2024-04-04 21:18:02	201,00	Declined	Checkout
2024-04-04 21:12:22	190,00	Approved	Checkin

16. PAYMENT PROCESSING WITH TOKENIZATION

To perform a payment with tokenization, the user must type the card information, so the system can encrypt the card information (number and expiration date) this in order to perform the charge over that payment method.To perform the tokenization is necessary to send the information of the card in the “tokenize” method, For more information regarding preauthorization: Placetopay Docs.

Note: Please note that this item applies if the tokenization service was acquired.

16.1. CHARGE TOKENIZED PAYMENT METHOD

After obtaining the token or subtoken from the tokenization process, the following information must be sent:

```
"payment": {
        "reference": "1122334455",
        "description": "Testing",
        "amount": {
            "currency": "USD",
            "total": 100
        }
    },
    "instrument": {
        "token": {
            "token":
"e07ca9986cf0ecac8a557fa11c07bf37ea35e9e3e3a4180c49"
        }
    },
```

The token or subtoken can be obtained from the response of the method on the instrument object.

The token or encrypted key, generated by a tokenization process, allows to generate charges without the user's interaction, or one-click payments.

Important: For payments that require a pin (Puerto Rico), it is necessary to ask the card holder this security information for its processing:

```
"payment": {
    "reference": "1234567890",
    "description": "Token payment with pin",
    "amount": {
      "currency": "USD",
      "total": 19.9
    }
  },
"instrument": {
   "pin": "0B880E2326F6409E",
        "token":
"ee1d56a192dc07e4e403cdaa2569407118a0e19dc185b1e63bad2b2edc2a3bf0",
      "subtoken": "5172915969800005",
      "franchise": "ath_card",
      "franchiseName": "ath_card ",
      "lastDigits": "0005",
      "validUntil": "2018-12-31"
    }
}
```


It is important to ensure that the payer's information is sent in the charge service with a token, as it is validated in the transactional security and data validation processing for the card holder.

16.2. TOKENIZATION OF ATH PAYMENT METHOD WITH PIN

When performing a tokenization process for the ATH payment method with a PIN, it is important to keep in mind that the charge should only be made when the user uses a wallet and is prompted to enter the PIN again. This means that a PIN pad must be provided to the customer so that they can enter it in the interface.

Note: It is important to consider that for this payment method, the merchant cannot generate a recurring or periodic charge, as this operation does not allow charges without PIN entry (PINless operation).

TOKEN INVALIDATION

The management of tokens and keys for payment instruments is important, storing this information safely and controlling the states for those keys in databases.

To invalidate an existing token stored in your database, it is necessary to send this information:

```
{
     "auth": {…},
    "locale": "en_PR",
    "instrument": {
      "token": {
        "token":
"a3bfc8e2afb9ac5583922eccd6d2061c1b0592b099f04e352a894f37ae51cf1a"
      }
    }
  }
```

See documentation to invalidate a token: Tokenization (Invalidate)

Note: If a payment method is being tokenized, a button or a section where the user can disable/delete the payment method must be displayed

16.3. TOKEN REMOVAL DUE TO REJECTIONS

It is important to maintain control over tokenized payment methods. If any of these generate consecutive rejections (3 times), the payment method must be automatically removed using the Invalidate Token method.

This implementation is necessary because brands such as Visa, Mastercard, among others, impose fines or penalties if attempts are made to charge a payment method that continuously gets rejected.

17. RESPONSE HANDLING FOR TRANSACTIONAL STATUSES

After the user finishes the payment process, the transaction details must be displayed, showing at least the following:

- Reference (Required).
- Transaction's final status (Required).
    - Approved. ○ Rejected. ○ Pending.
    - Failed.
- Reason and message.
- Date and time (Required).
- Total amount (Required).
- Receipt.
- Authorization.
- Card's last 4 digits.

If you want to display more information to the final user, you can follow the template from RESPONSE TEMPLATE FORMAT described below on this document, showing the agreement status with the response given by Evertec Placetopay.

Recommendations aligned to the user experience:

- Approved response: It is possible to add a link or button so the user or client can return to the home payment site, validating the reception as well.

Back to Home

- Denied or failed response: It is recommended to add a link or button that allows the user to retry the payment.

Retry Payment

17.1. PENDING TRANSACTION CONTROL:

It is possible to visualize a transaction with pending status by the direct financial institution's response or by not receiving a response from Evertec Placetopay (TimeOut).

There are three ways to obtain the final status of a transaction: The first is through the service's response during the **TransactionProcessing**; the second, in case of receiving a PENDING status, through the **TransactionQuery** method by using the internal reference; and the third, through the **TransactionSearch** service, which acts as a contingency plan in case of communication loss when creating a transaction during the **TransactionProcessing**.

Note: The request towards the service to know the final status of a pending transaction must be made using the **TransactionQuery** method, the **TransactionSearch** method is only for lost connection cases during the transaction.

There are two main methods to obtain the final status of a transaction, using the notification URL (webhook) and a scheduled task (cronjob):

- **Notification URL:** The purpose of this process is to inform your system when the transactions change their status from pending to final status. The notification URL is configured by the merchant under ports 80 or 443 and must be programmed to receive a POST request on Evertec Placetopay's behalf. The structure looks like this:


```
{
  "status": {
    "status": "APPROVED",
    "reason": "00",
    "message": "APPROVED",
    "date": "2024-07-11T15:22:37-05:00"
  },
  "internalReference": 1,
  "reference": "5834381",
  "signature": "9c0f8ff164d0af4a795f71ee127d8926f56d05fb"
}
```


After the Transaction Processing service consumption, a payment network timeout can be automatically generated if a response is not given on time.

- **Sonda or Cronjob (optional)**: This process consists of a scheduled task that consumes the **TransactionQuery** method for transactions that remain pending on your records. This process must be executed every 12 minutes to verify which transactions have more than 5 minutes in pending status.

When a transaction is found on the database with pending or ongoing status, the system must inform and present the following event:

- A **double charge control message** informing the user of existing pending transactions before making a new payment attempt. For example:
  - “At this moment, your order with reference number *#Reference* and value of *#Amount* is PENDING. In case of not receiving a confirmation from your financial institution, please wait for a couple of minutes and check again to verify if your payment is successfully confirmed. If you want more information regarding your transaction’s status, you can get in touch with us through our contact channels *000-00-00* or emailing to * email@email.com, asking for the status of the transaction <#CUS/Authotizationn> “.

17.2. 32 SECOND TIMEOUT

For the responses given through the webhook notification process, a bottleneck may occur; this can happen due to a delay on the response from the processing network. In this regard, the merchant’s development team must configure a timeout with a maximum waiting time of 32 seconds to receive this notification. In case this time is exceeded, the merchant must mark the transaction as failed and give a response to the user or final client as “Declined transaction.”.

18. SALES RECEIPT

In case of sending custom sales receipts (printed, emailed, exported file, voice message) from the merchant’s system according to each one of the transaction statuses and that payment-related information is displayed, verify that the information provided to the user consistently matches with the payment information. It is suggested that the user visualize the following information:

- Reference.
- Date and time of the transaction.
- Transaction status.
- Total paid.
- Type of credit and deferred (Ecuador) or selected installment.
- Card’s last four digits.
- Bank.

If the merchant doesn't count with its own proof of sales, consider using the reference provided by Evertec Placetopay.

19. TRANSACTIONAL HISTORY

If the final user must authenticate in the merchant's site in order to start a payment process, the user must be capable of seeing at least the last 10 transactions. Each registry must contain at least the following information sorted descending by date: ● Date and time of the transaction.

- Reference number (issued by the merchant).
- Authorization/CUS (optional).
- Transaction status.
- Amount (must be joined with the currency according to ISO 4217).

DATE AND HOUR	REFERENCE	STATUS	AUTHORIZATION/CUS	TOTAL
31/01/2025 6:50 P.M.	ORDER_420	PENDING	5555	USD $200
31/01/2025 1:00 P.M.	ORDER_419	APPROVED	5439	USD $289
29/01/2025 10:50 A.M	ORDER_418	REJECTED	5276	USD $276
28/01/2025 3:00 P.M.	ORDER_417	FAILED	5002	USD $526

Note: If authentication is not required on the merchant's website and if by the business rules it is not possible to implement a module to look up for the previous user's transactions, it is necessary to inform the reason to the Evertec Placetopay analyst so it can be analyzed to give an exception on the case.

20. INFORMATION SECURITY

The system and the merchant must comply with the following criteria:

- The merchant must be certified in PCI (current).
- The system must not store, under any circumstance, cardholder's sensitive information.
- The system does not show to the user in the payment summary, sales receipt, or in any other scenario the card number (maximum four digits and BIN).
- The system does not show to the user in the payment summary, sales receipt, or in any other scenario the card CVV.

- Store in the database the minimum required information for any future claim, such as the transaction's date and time, receipt number, amount, authorization number, up to 4 card number last digits, reference, and promissory note if applicable.
- The system must have a valid digital certificate, gathering information under the HTTPS safe protocol.
- Must use TLS 1.2 protocol or greater.
- If the development uses a web interface (even intranet), the URL of information capture is not masked (hidden) using techniques such as IFRAMES.

21. INFORMATION PROCESSING

When making a transaction, it is sent to the Evertec Placetopay application with the same data that was validated, typed, or selected on the user interface:

- Card number.
- CVV.
- Expiration date.
- Total amount.
- Currency.
- Reference.
- Description.
- Tax (optional).
- Return Basis (optional if no taxes are used).
- User information.
- Security protocol information.

22. CONSISTENCY OF INFORMATION

The information sent to Evertec Placetopay should not have inconsistencies across the interfaces where payment-related information is presented. Accordingly, the following aspects must be met:

- The transaction date and time must be consistent between the commerce development database, the Evertec Placetopay console query, and the sales receipt generated by the system when transactions are carried out.
- The last four digits of the card must appear uniquely and consistently between the commerce development database and the Evertec Placetopay console query.

23. IP ADDRESS AND BROWSER AGENT

When sending a request to process a transaction using the **Transaction Processing** method, it is important to include the ipAddress and userAgent parameters as mentioned in the FIELD VALIDATION section (processTransaction). These are additional required data that must be sent for the transactional security filter control.

Example

```
…
"ipAddress": "192.168.1.100",
"userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0
Safari/537.36"
…
```

The IP address must be obtained from the customer's device, either by executing a function in your code or through a script depending on your programming language. The same applies to the browser agent, whether it is a web page or an application where the service is integrated.

24. SENGIND EXTRADATA

You can send additional information in the requests using extradata. If additional references or other relevant data need to be included in the payment process, this can be done using an array of objects called **additional**.

Example:

```
…
"additional": {
    "merchantCode": "468231",
    "terminalNumber": "00990101"
    },
…
```

24.1. CUSTOMER ACCOUNT NUMBER

If you need to send specific additional information about a customer account where services are linked and identified in your system by an account number, you must send the extradata as follows:

```
" additional ": {
    " CustomerAccountNumber ": "8100312356677"
   }
 ]
```

Note: The following keys are NOT allowed: _accountNumber, userAgent, fingerprint, sourcePlatform, tokenizationID, trazabilyCode, transactionCycle, RequestId, PartnerAuthCode.

25. REVERSE TRANSACTIONS

A transaction reversal process must be defined. If the merchant intends to process reversals through the API provided by Placetopay Evertec, this must be confirmed with an analyst to validate functionality and ensure proper operation. On the other hand, if the administrative console is to be used, this must be communicated via email within the analyst's assigned thread.

Note: A reversal is only possible before the reconciliation time. If the process needs to be executed after the defined time, a refund must be processed instead. Additionally, it must be communicated whether reversals will be performed via API or through the administrative console.

25.1. REFUND TRANSACTION

If refunds are to be processed using the API provided by Placetopay Evertec, this must be confirmed with an analyst to validate functionality and ensure proper operation. If the administrative console is used instead, this must be communicated via email within the analyst's assigned thread.

Note: Refunds will only be effective if processed after 3:00 PM (standard cutoff time).

25.2. TYPES OF REFUND

Full Refund

A full refund is used to reimburse the total amount approved in the transaction.

```
{
    "auth": {},
    "internalReference": int,
    "authorization": "int",
    "action": "refund",
}
```

Note: The "refunded" object will be present, indicating with true that the transaction has been refunded.

```
"refunded": true,
```

Partial Refund

A partial refund is used to reimburse only part of the approved transaction amount. The **"amount"** property must be included in the request, specifying the amount to be refunded.

```
{
    "auth": {},
    "internalReference": int,
    "authorization": "int",
    "action": "refund",
    "payment": {
        "amount": {
            "currency": "string",
            "total": int
        }
    }
}
```

Partial refunds can be processed multiple times until the total approved transaction amount is fully refunded, once a transaction has been fully refunded, no further refunds can be processed.

Note: Partial refunds can only be processed once the transaction has passed the cut-off time and has been reconciled, which usually happens the next business day.

Below are key points related to the response of a transaction queried after a refund:

- The **"refunded"** object will be present, indicating with true that the transaction has been refunded.

```
"refunded": true,
```

- Within the **"additional"** array, a structure will be present specifying the refunded amount of the transaction.

```
{
       "merchantCode": "int",
        "terminalNumber": "string",
        "bin": "int",
        "_wcTransactionId_": "int",
        "amountRefunded": int }
```

26. RELEVANT INFORMATION

27. Webhook for ACH Refunds (PR Optional)

This item applies when the merchant uses the ACH payment method. In this case, a webhook must be available to manage and configure refunds related to this payment method.

The merchant must provide a public URL/URI where PlacetoPay can send refund notifications. This URL must be enabled to accept HTTP POST requests with a JSON payload structure, as shown below:

```
{
"time": "2024-11-14T05:50:15-05:00",
"type": "chargeback.created",
"data": {
       "status": {
       "status": "APPROVED",
       "reason": "R01",
       "message": "Fondos insuficientes",
       "date": "2024-07-03T22:59:00-05:00"
       },
       "date": "2024-07-03T22:59:00-05:00",
       "transactionDate": "2024-07-03T22:59:00-
       05:00",
       "internalReference": 2,
       "reference": "9123418",
       "paymentMethod": "EBACH",
       "franchise": "ach",
       "franchiseName": "Ebus ACH",
       "issuerName": null,
       "amount": {
       "taxes": [
              {
              "kind": "valueAddedTax",
              "amount": 0,
              "base": 0
              }
       ],
       "currency": "USD",
       "total": 20000
       },
       "conversion": {
       "from": {
              "currency": "USD",
              "total": 20000
       },
       "to": {
              "currency": "USD",
              "total": 20000
       },
       "factor": 1
       },
       "authorization": "10000123",
       "receipt": null,
       "type": "CHARGEBACK",
       "refunded": false,
       "lastDigits": null,
       "provider": null,
       "discount": null,
       "processorFields": {
       "id": "b66f02ffff79b79ea9587b3cd3507a50",
       "b24": "R01"
       },
       "additional": {
       "merchantCode": "4549106521651",
       "terminalNumber": "19371021",
       "bankName": "Test Bank",
       "accountNumber": "6789"
       }
}
}
```

27.1. Validate the Signature

To ensure the authenticity of each notification, PlacetoPay includes the XSignature field in the HTTP request header. You can validate this signature using an HMAC SHA-256 hash, taking the content of the notification body and your site's transaction key (tranKey) as input.

```
const crypto = require('crypto');

// Body de la solicitud y firma recibida
const body = JSON.stringify(notificationBody);
const receivedSignature = headers['X-Signature'];

const tranKey = 'YOUR_SITE_TRANKEY';

// Generar la firma localmente
const generatedSignature = crypto
  .createHmac('sha256', tranKey)
  .update(body)
  .digest('hex');

// Validar la firma
if (generatedSignature === receivedSignature) {
  console.log("La notificación es auténtica.");
} else {
  console.log("Firma inválida: la notificación no es auténtica.");
}
```

An HTTPS notification is sent when an ACH refund is generated. Webhooks – PlacetoPay Docs

evertec®

evertecinc.com