

1. Introduction

This guide outlines how to work with the provider's API for processing and encrypting credit card data. It includes examples of XML requests and responses, descriptions of required fields, and PHP code examples.

Table of Contents:

- 1. Introduction
- 2. Authentication
- 3. Request and Response Format
- 4. Types of Requests
 - 4.1 Credit Card Tokenization
 - 4.2 Transaction Status Check
- 5. XML Request and Response Examples
- 6. Errors and Response Codes

2. Authentication

To use the provider's API, you need a unique merchant account identifier (`merchant-account-id`). This identifier must be included in every request.

Example:

```
<merchant-account-id>c3671cf9-6366-4e39-8d67-31ce24094655</merchant-account-id>
```

Note: Ensure that the `merchant-account-id` is kept confidential and is never exposed in public-facing documentation or logs.

3. Request and Response Format

All requests and responses must be sent in XML format. Requests are made via HTTP POST with `Content-Type: application/xml`.

- Requests must be well-formed XML documents.
- Responses are returned in XML format.

Important: Ensure that the XML body is properly encoded in UTF-8 and follows the format specified in the XSD schema.

4. Request Types

4.1 Credit Card Tokenization

This request is used to tokenize credit card data. You need to send the card details, including the card number, expiration date, and security code, to receive a unique token for subsequent transactions.

Request Parameters:

- `merchant-account-id` (required) — Merchant account identifier.
- `request-id` (required) — Unique request identifier.
- `transaction-type` (required) — Type of transaction (e.g., `tokenize`).
- `card` (required) — Card information, including card number, expiration date, and security code.

Sample Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<payment>
  <payment-methods>
    <payment-method name="creditcard"/>
  </payment-methods>
  <merchant-account-id>c3671cf9-6366-4e39-8d67-31ce24094655</merchant-account-id>
  <request-id>7ac7f07f-fec5-48a3-afa0-f8ca41312a39</request-id>
  <transaction-type>tokenize</transaction-type>
  <card>
    <account-number>4444333322221111</account-number>
    <expiration-month>12</expiration-month>
    <expiration-year>2026</expiration-year>
    <card-type>visa</card-type>
    <card-security-code>123</card-security-code>
  </card>
</payment>
```

Sample Response:

```
<payment>
  <merchant-account-id>c3671cf9-6366-4e39-8d67-31ce24094655</merchant-account-id>
  <transaction-id>f96a2be7-88a5-4a2b-8d3e-34e0666e6c8c</transaction-id>
  <request-id>7ac7f07f-fec5-48a3-afa0-f8ca41312a39</request-id>
  <transaction-type>tokenize</transaction-type>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2024-10-10T10:47:31.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
  <card-token>
    <token-id>4920619521161111</token-id>
    <masked-account-number>4444331111</masked-account-number>
  </card-token>
</payment>
```

Note: The `card-token` is used for subsequent transactions without needing to resend card details. Always store the token securely.

4.2 Transaction Status Check

This request is used to check the status of an existing transaction using a unique transaction ID.

Request Parameters:

- `merchant-account-id` (required) — Merchant account identifier.
- `transaction-id` (required) — Unique transaction identifier.

Sample Request:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<status-request>
  <merchant-account-id>c3671cf9-6366-4e39-8d67-31ce24094655</merchant-account-id>
  <transaction-id>f96a2be7-88a5-4a2b-8d3e-34e0666e6c8c</transaction-id>
</status-request>
```

Sample Response:

```
<status-response>
  <merchant-account-id>c3671cf9-6366-4e39-8d67-31ce24094655</merchant-account-id>
  <transaction-id>f96a2be7-88a5-4a2b-8d3e-34e0666e6c8c</transaction-id>
  <transaction-state>success</transaction-state>
  <completion-time-stamp>2024-10-10T10:47:31.000Z</completion-time-stamp>
  <statuses>
    <status code="201.0000" description="The resource was successfully created." severity="information"/>
  </statuses>
</status-response>
```

5. XML Request and Response Examples

This section contains examples for all major request types and corresponding responses.

1. Credit Card Tokenization
2. Transaction Status Check

6. Errors and Response Codes

If errors occur, the API returns an XML response with an error description and status code.

Example of an error response:

```
<statuses>
  <status code="404.0001" description="Transaction not found" severity="error"/>
</statuses>
```

Status Code Explanations:

- `200.0000` — Success.
- `404.0001` — Transaction not found.
- `500.0001` — Internal server error.

- `400.0001` — Invalid card number.
- `402.0002` — Insufficient funds.

Note: Always check the `severity` field to determine whether the response indicates a success or failure.