|  |  |  |
| --- | --- | --- |
| **S. No.** | **API** | **Description** |
| **1** | Checkout API | Checkout API |
| **2** | Status Enquiry | To get status of the transaction |
| **3** | Webhook | Callback initiated to merchant URL |

**1. Overview**

This document provides a detailed guide for integrating with the PGH Crypto Payment Gateway. It outlines the API specifications, request formats, and response handling mechanisms to ensure a smooth transaction process.

**2. API Endpoints**

**2.1 Checkout API**

**Description**

The Checkout API is used to initiate payment requests. The first request is submitted via form submission from the browser, redirecting the user to the PGH payment page for processing.

**Integration Process**

1. **Create an order form** with the required parameters.
2. **Submit the form via POST** to redirect the user to the payment gateway.
3. **User completes the transaction** on the PGH payment page.
4. **Merchant receives the response** via Webhook or Status Enquiry API.

**Endpoint**

POST {baseUrl}/pgui/jsp/paymentrequest

**Required Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **Data Type** | **Mandatory** | **Description** |
| APP\_ID | Numeric | Yes | Unique merchant identifier |
| ORDER\_ID | Alphanumeric | Yes | Merchant's order reference |
| TXNTYPE | Alphabet | Yes | Transaction type (SALE) |
| CURRENCY\_CODE | Numeric | Yes | Currency code (ISO 4217) |
| CUST\_FIRST\_NAME | Alphabet | Yes | Customer's first name |
| CUST\_LAST\_NAME | Alphabet | Yes | Customer's last name |
| CUST\_STREET\_ADDRESS1 | Alphanumeric | Yes | Customer's street address |
| CUST\_CITY | Alphabet | Yes | Customer's city |
| CUST\_STATE | Alphabet | Yes | Customer's state |
| CUST\_COUNTRY | Alphabet | Yes | Customer's country |
| CUST\_ZIP | Numeric | Yes | Customer's zip code |
| CUST\_PHONE | Numeric | Yes | Customer's phone number |
| CUST\_EMAIL | Alphanumeric | Yes | Customer's email |
| AMOUNT | Numeric | Yes | Amount in minor currency units (e.g., 1000 for USD 10.00) |
| RETURN\_URL | URL | Yes | Merchant URL to receive the response |
| HASH | Alphanumeric | Yes | Security hash (see hash generation) |

**Important Notes**

* The **first request must be a form submission** from the browser, redirecting the user to the PGH payment page.
* Amounts should be provided in minor currency units (e.g., USD 1.20 should be sent as 120).
* **A transaction is considered successful only if RESPONSE\_CODE = 000 and STATUS = Captured**.
* Always **validate the response hash** to ensure data integrity.
* **Do not rely on messages other than RESPONSE\_CODE and STATUS** for transaction success determination.

**2.2 Status Enquiry API**

**Description**

Status Enquiry API allows merchants to retrieve the transaction status when no response is received due to network errors or other exceptions.

**Endpoint**

POST {baseUrl}/pgui/services/paymentServices/transactionStatus

**Required Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter Name | Data Type | Mandatory | Description |
| APP\_ID | Numeric | Yes | Unique merchant identifier |
| TXN\_ID | Numeric | Optional | Transaction ID received in response |
| ORDER\_ID | Alphanumeric | Yes | Order reference used at checkout |
| CURRENCY\_CODE | Numeric | Yes | Currency code |
| AMOUNT | Numeric | Yes | Transaction amount |
| HASH | Alphanumeric | Yes | Security hash |

**Sample Status Enquiry Response**

[

 {

 "RESPONSE\_CODE": "000",

 "APP\_ID": "1000221213112408",

 "AMOUNT\_DUE": "0.00369902",

 "TXN\_ID": "1000240517133901",

 "EXCHANGE\_RATE": "2703.42",

 "MOP\_TYPE": "ETH",

 "CRYPTO\_DEPOSIT\_ADDRESS": "75f60814d4e597563315f8da67521ba954e23cb37a3faa28",

 "ACQ\_ID": "1003240517133850",

 "ACQUIRER\_RESPONSE": "Transaction Successful",

 "HASH": "BDA522BD2DC30935B85E9A9F031995C4BF792B66AD865644D3FF9876406B58D5",

 "PAYMENT\_TYPE": "CR",

 "STATUS": "Captured",

 "CURRENCY\_CODE": "840",

 "CREATE\_DATE": "2024-05-17 13:39:13",

 "ORDER\_ID": "SIGN1715933328571",

 "AMOUNT": "10.00",

 "RESPONSE\_MESSAGE": "SUCCESS",

 "AMOUNT\_UNDERPAID": "N",

 "TXID": "921eadbabf2566945595342b01f41bf8f39357d9818740e7e910250492156e6e",

 "AMOUNT\_OVERPAID": "N",

 "AMOUNT\_RECEIVED": "0.00369902"

 },

 {

 "API\_RESPONSE": "000"

    }

]

**Important Notes**

* Either ORDER\_ID or TXN\_ID can be used for status enquiry.
* Amounts must be provided in minor currency units.
* The **valid response combination for success**: RESPONSE\_CODE = 000 and STATUS = Captured.
* Status Pending means the transaction is still in progress.

**2.3 Webhook / Callback**

**Description**

The PGH system triggers a webhook as soon as there is an update in the transaction status. Merchants must provide a URL to receive JSON-based callbacks.

**Sample Webhook Endpoint**

https://www.merchant.com/webhook

**Sample Webhook Response**

{

 "BILLING\_CUST\_NAME": "Shaikh Zayed",

 "RESPONSE\_DATE\_TIME": "2024-05-24 12:17:17",

 "RESPONSE\_CODE": "015",

 "MERCHANT\_CITY": "Delhi",

 "CUST\_PHONE": "0987654321",

 "MOP\_TYPE": "ETH",

 "EXPIRY": "172",

 "CURRENCY\_CODE": "840",

 "CUST\_FIRST\_NAME": "Shaikh",

 "CURRENCY\_CODE\_ALPHA": "USD",

 "STATUS": "Pending",

 "CUST\_STREET\_ADDRESS1": "Shaikh Zayed Road",

 "CUST\_CITY": "Dubai",

 "AMOUNT": "1000",

 "RESPONSE\_MESSAGE": " Status is yet to be received from Bank",

 "CUST\_STATE": "UAE",

 "MERCHANT\_COUNTRY": "Greece",

 "CUST\_EMAIL": "capital@yolo.com",

 "CUST\_ZIP": "122016",

 "MERCHANT\_POSTAL\_CODE": "110001",

 "AMOUNT\_DECIMAL": "10.00",

 "APP\_ID": "1000221213112408",

 "CUST\_LAST\_NAME": "Zayed",

 "AMOUNT\_DUE": "0.00304206",

 "TXN\_ID": "1003240524121715",

 "EXCHANGE\_RATE": "3287.241",

 "MERCHANT\_BUSINESS\_NAME": "TestMerchant",

 "CRYPTO\_DEPOSIT\_ADDRESS": "0cdf66b064e1628611e541ff9fc2df1c9f8e0c124f14d4b0",

 "ACQ\_ID": "132094",

 "TXNTYPE": "SALE",

 "CUST\_COUNTRY": "AE",

 "DUPLICATE\_YN": "N",

 "HASH": "A8C7D4394647A36B43FA44EAD4981CEB41CA8B61C9318653B1330331A409E39C",

 "PAYMENT\_TYPE": "CR",

 "RETURN\_URL": "http://merchant.com/test/returnUrl",

 "ORDER\_ID": "SIGN1716533216675",

 "MERCHANT\_ADDRESS": "Delhi",

 "MERCHANT\_STATE": "Delhi",

 "MERCHANT\_MOBILE\_NO": "9876543210",

 "CUST\_NAME": "Shaikh Zayed"

}

**Important Notes**

* Merchants must **validate the hash** before processing the webhook response.
* Only **RESPONSE\_CODE = 000 + STATUS = 'Captured'** confirms a successful transaction.
* If no response is received, use the **Status Enquiry API**.

**3. Hash Generation Mechanism**

Every request is accompanied by a secure Hash that verifies the request’s Integrity as well as merchant identity. The secure Hash algorithm is SHA-512.

* To generate secure Hash, use the following method: Take all the request parameters that are to be sent and sort them in ascending order according to the name of each field.
* Now prepare a String of these sorted name value params with Tiled (~) as the separator like:

AMOUNT=1000~APP\_ID=1000221129001154~CURRENCY\_CODE=840~CUST\_CITY=Winterfell~CUST\_COUNTRY=US~CUST\_EMAIL=johnsnow@test.com~CUST\_NAME=John Snow~CUST\_PHONE=9454243567~CUST\_STREET\_ADDRESS1=Great Wall~CUST\_ZIP=32546 ~ORDER\_ID=7773428492547592~RETURN\_URL=https://www.merchant.com/response.jsp~TXNTYPE=SALE

* Next step is to append the Salt Key at the end of the parameter string given to you. After adding you will get the following output (If salt key is 5cd925b2d0254be0)

AMOUNT=1000~APP\_ID=1000221129001154~CURRENCY\_CODE=840~CUST\_CITY=Winterfell~CUST\_COUNTRY=US~CUST\_EMAIL=johnsnow@test.com~CUST\_NAME=John Snow~CUST\_PHONE=9454243567~CUST\_STATE=The North~CUST\_STREET\_ADDRESS1=Great Wall~CUST\_ZIP=32546~ORDER\_ID=7773428492547592~RETURN\_URL=https://www.merchant.com/response.jsp~TXNTYPE=SALE5cd925b2d0254be0

Note: Please append the secret key in lower case without any separator after the last value.

* After completing the above-mentioned process, you will have to call SHA 512 algorithm and pass the parameter string to the same and the SHA will return you the desired result as

Hash Value

1ed4fc3edf91e2d991a47a3d0492fe21e21195ff5ea20322c27fbc1bb3d09f56f03d6a24bf1138eaaceeffa9b6d16b381731ab330b1654e39d1c9711862744bc

* Now you must convert the generated value to the Upper Case, and you will get the result.

Final Hash Value

1ED4FC3EDF91E2D991A47A3D0492FE21E21195FF5EA20322C27FBC1BB3D09F56F03D6A24BF1138EAACEEFFA9B6D16B381731AB330B1654E39D1C9711862744BC

**4. Response Codes**

|  |  |  |
| --- | --- | --- |
| **Response Code** | **Response Message** | **Comments** |
| 000  | Success  | Transaction captured  |
| 001  | Acquiring Error  | Acquirer bank not available  |
| 002  | Denied  | Denied by Bank  |
| 003  | Timeout  | Transaction timed out  |
| 004  | Declined  | Declined by bank  |
| 005  | Authentication not available  | Authentication issue  |
| 006  | Transaction processing  | Transaction in process  |
| 007  | Rejected by acquirer  | Rejected by bank  |
| 008  | Duplicate  | Duplicate submission  |
| 009  | Response signature did not match  | Response from acquirer did not match, re-check transaction status  |
| 010  | Cancelled by user  | User pressed cancel button  |
| 012  | Denied due to fraud detection  |   |
| 015  | Transaction is pending  | Transaction is pending  |
| 017  | Acquiring service down  |   |
| 018  | Duplicate order ID  |   |
| 022  | Failed at acquirer  | Transaction was not processed by bank  |
| 026  | Pending  | Transaction in process  |
| 100  | User not found  | Invalid credentials used  |
| 105  | User not approved  | User not active  |
| 113  | Payment option not supported  | Payment method not allowed for the user  |
| 130  | Currency not supported  | Currency not supported |
| 300  | Invalid Request  | Invalid Request  |
| 302  | No transaction found  | No transaction found  |
| 323  | Invalid Hash  | Invalid Hash  |
| 327  | Invalid request (Request URL)  | Invalid request (Request URL)  |
| 900  | Internal system error  | Internal system error  |
| 999  | Unknown Error | Unknown Error |

**5. Currency Codes and Minor Units**

|  |  |  |  |
| --- | --- | --- | --- |
| **Currency** | **Alphabetical Code** | **Numeric Code** | **Decimals** |
| United Arab Emirates Dirham | AED | 784 | 2 |
| Australian dollar | AUD | 036 | 2 |
| Canadian Dollar | CAD | 124 | 2 |
| Euro | EUR | 978 | 2 |
| Pound sterling | GBP | 826 | 2 |
| Indian rupee | INR | 356 | 2 |
| United States Dollar | USD | 840 | 2 |
| Bangladeshi Taka | BDT | 050 | 2 |
| Bahraini dinar | BHD | 048 | 3 |
| Swiss Franc | CHF | 756 | 2 |
| Japanese yen | JPY | 392 | 0 |
| Kenyan shilling | KES | 404 | 2 |
| Kuwaiti dinar | KWD | 414 | 3 |
| Sri Lankan rupee | LKR | 144 | 2 |
| Nepalese rupee | NPR | 524 | 2 |
| Qatari Riyal | QAR | 634 | 2 |
| South African Rand | ZAR | 710 | 2 |
| Nigerian Naira | NGN | 566 | 2 |
| Vietnamese Dong | VND | 704 | 0 |
| Saudi Riyal | SAR | 682 | 2 |
| Omani Rial | OMR | 512 | 3 |
| Singapore Dollar | SGD | 702 | 2 |
| Philippine Peso | PHP | 608 | 2 |
| Egyptian Pound | EGP | 818 | 2 |
| Zambian Kwacha | ZMW | 967 | 2 |
| Russian Ruble | RUB | 643 | 2 |
| Ghanaian Cedi | GHS | 936 | 2 |
| Ugandan Shilling | UGX | 800 | 0 |
| Tunisian Dinar | TND | 788 | 3 |
| Ukrainian Hryvnia | UAH | 980 | 2 |
| Jordanian Dinar | JOD | 400 | 3 |
| Moroccan Dirham | MAD | 504 | 2 |
| Iraqi Dinar | IQD | 368 | 3 |
| Turkish lira | TRY | 949 | 2 |
| Chinese Yuan | CNY | 156 | 2 |