RinggitPay

PAYMENT GATEWAY INTEGRATION & TESTING GUIDE

VERSION 1.19

COMPANY INFORMATION

RinggitPay Sdn Bhd

(Company Registration No.: 1340946-U)

CORPORATE ADDRESS

HEADQUARTERS:

Unit A-13-13A & Unit A-13-15
Level 13, Tower A, Menara UOA Bangsar
No. 5, Jalan Bangsar Utama 1
59000 Kuala Lumpur
Malaysia

CONTACT INFORMATION

Telephone: +603-2201 6498 / +6017-989 8253

Fax: +603-2280 0313

Email: careline@ringgitpay.com

Website: www.ringgitpay.biz

SOCIAL MEDIA AND MESSAGING CHANNELS

Stay connected with us through our official platforms

Facebook: facebook.com/ringgitpay

Instagram: instagram.com/ringgitpay

WhatsApp: wa.me/60179898253

LinkedIn: linkedin.com/company/ringgitpay

Version History

Project Name	RinggitPay Integration And Testing			
Document Name	RinggitPay Payment Gateway Integration And Testing Guide			
Document Status	Reviewed			
Version	**Date**	**Author**	**Reviewer**	**Version Summary**
1.10	24-06-2021	Joseph	Sagar	New CC test data, Status code, payer account verification added
1.11	20-02-2023	Joseph	Sagar	Payer Account Verification - RPP
1.12	08-05-2023	Joseph	Sagar	Payer Account Verification Updated
1.13	01-07-2023	Joseph	Sagar	Payment mode, card response updated, API Header notes added
1.14	01-12-2023	Joseph	Sagar	Settlement response message added
1.15	18-01-2024	Joseph	Sagar	Merchant custom reference
1.16	05-04-2024	Joseph	Sagar	Card test data updated
1.17	15-11-2024	Joseph		Ref 4 update, Merchant notes
1.18	20-01-2025	Joseph	Vineeth	Ref 5, Ref 6 updated
1.19	21-04-2025	Vineeth	Joseph	i) Revamped the overall document design, including updates to the font type, header, and footer for improved readability and consistency. ii) Revised test card details for credit and debit transactions in Section 5.5.2 - Credit/Debit Card. iii) Replaced and enhanced RinggitPay Gateway checkout screen shots in Section 5.5 - UAT Payment Interface Overview to reflect the latest interface updates.

TABLE OF CONTENTS

1.Introduction

In today's digital economy, accepting online payments is no longer a luxury it's a necessity. However, navigating the complexities of integrating with banks and third-party payment gateway providers can be overwhelming, time-consuming, and costly. Many merchants, particularly those without a technical background, find it challenging to implement even a single payment method on their existing platforms.

At **RinggitPay Sdn Bhd,** we recognize these challenges and have developed a robust, user-friendly, and secure solution to simplify the online transaction process for merchants of all sizes. Our flagship product, the RinggitPay Secure Online Transaction Service, is designed to help businesses grow and scale across Southeast Asia by enabling fast, safe, and flexible digital payments.

"Empowering Businesses with Seamless, Secure and Scalable Online Payment Solutions"

1.1. Purpose of the Document

This document serves as a comprehensive technical and operational guide for merchants and developers integrating with RinggitPay's secure online payment platform. It outlines the necessary procedures, API specifications, message formats, authentication mechanisms, and transaction workflows required to initiate, process, and verify online payments through RinggitPay.

By following this guide, merchants will be equipped to:

- ✓ Seamlessly implement the RinggitPay payment gateway in their online platforms
- ✓ Understand and utilize RinggitPay's request/response protocols
- ✓ Ensure secure data exchange and transaction verification
- ✓ Successfully complete User Acceptance Testing (UAT) and go live in production

This guide is intended for technical teams, including developers, engineers, and integration specialists, as well as project stakeholders who oversee payment implementation and security compliance.

1.2. What is RinggitPay?

RinggitPay is a comprehensive payment gateway solution provider based in Malaysia. Our platform is built with a focus on reliability, security, and user centric design, enabling merchants to effortlessly accept and manage online payments through a unified platform. We cater to businesses in e-commerce, retail, services, and digital industries offering customized solutions to suit various operational needs, business models, and market segments.

1.3. Our Solution: RinggitPay Secure Online Transaction

The RinggitPay Secure Online Transaction service is an end-to-end solution that handles both the front-end and back-end aspects of digital payments. It provides a seamless experience for both merchants and their customers, ensuring secure, real-time, and transparent payment processing.

Core Features and Capabilities

Front-End (Customer-Facing Interface)

- **Responsive Payment Page Design**
  Our payment pages are designed to work flawlessly across all devices desktops, tablets, and mobile phones ensuring a smooth and professional checkout experience for your customers.

- **Custom Branding Options**
  Easily tailor the look and feel of your payment page to match your brand identity, building trust and improving user experience.

- **Secure Checkout Experience**
  All transactions are encrypted and follow industry-standard security protocols to protect customer data and reduce fraud risk.

- **Real-Time Payment Tracking**
  Merchants can log in to their dedicated dashboard at any time to monitor transaction statuses and manage customer payments in real time.

- **Instant Notifications**
  Get immediate alerts on successful or failed transactions and other payment activities.

- **Detailed Visual Reports**
  Gain valuable insights through interactive charts and graphs that allow you to monitor payment collections, analyze invoice aging trends, and review e-Mandate collection history with ease.

Back-End (Technical & Operational Features)

- **Server-to-Server Notifications (IPN)**
  Our system sends instant server-to-server notifications to ensure that all transaction statuses are updated accurately and without delays minimizing errors and manual checks.

- **Flexible API Integration**
  RinggitPay offers comprehensive, well-documented APIs for developers to easily integrate payment functionality into websites, mobile apps, and other platforms.

- **Support for Multiple Payment Methods**
  Accept a wide range of payment types, including credit/debit cards, FPX (online banking), e-wallets, and more ensuring convenience for your customers.

- **Scalable Infrastructure**
  Whether you're handling hundreds or thousands of transactions, RinggitPay is designed to grow with your business without compromising on speed or performance.

1.4. Why Choose RinggitPay?

- **Faster Time-to-Market:** Quick and easy integration with minimal technical dependency.
- **Reduced Operational Overhead:** Automated processes minimize manual reconciliation and payment tracking.
- **Enhanced Customer Experience:** Deliver a smooth, fast, and secure payment journey that increases customer satisfaction and reduces cart abandonment
- **Regulatory Compliance:** Adheres to local financial regulations and global security standards (PCI DSS).
- **Dedicated Local Support:** Our Malaysian-based support team is ready to assist with integration, maintenance, and troubleshooting.

Start Accepting Payments the Smarter Way
Join hundreds of businesses already using RinggitPay to power their online sales. Whether you're launching a new e-commerce store or looking to optimize your current payment system, RinggitPay offers the flexibility, reliability, and support you need.

"Contact us today to schedule a demo or get started with your RinggitPay integration"

1.5. Privacy Notice

This document contains confidential and proprietary information intended solely for the use of RinggitPay merchants and authorized partners.

By accessing or using this document, you agree to:

1. Use the information only for the purpose of system integration with RinggitPay
2. Not distribute, reproduce, or share any part of this document without prior written consent from RinggitPay
3. Handle all data and content with strict confidentiality in accordance with applicable data protection and privacy regulations

If you have received this document in error, please notify RinggitPay immediately and delete it from your system.

1.6. Definition

This section outlines key terms and concepts used throughout this document to ensure clarity and consistency.

1.6.1. Abbreviation

The following abbreviations are used throughout this document to ensure clarity and consistency in the terminology related to payment systems, technology integration, and financial services. These abbreviations are commonly referenced in both technical and business contexts.

S. No	Abbreviation	Description
1	PCI DSS	Payment Card Industry Data Security Standard
2	API	Application Program Interface
3	URL	Uniform Resource Locator
4	UAT	User Acceptance Testing
5	SHA	Secure Hash Algorithm
6	HTTPS	Hypertext Transfer Protocol Secure
7	NVP	Name Value Pair
8	CVV	Card Verification Value
9	UTC	Universal Time Coordinated
10	CC	Credit Card
11	DC	Debit Card

2.Payment Flow Overview

RinggitPay provides a secure and simplified hosted payment page solution, enabling merchants to accept online payments without the need for complex or PCI DSS-compliant infrastructure within their own systems.

The integration is straightforward: merchants submit essential transaction details to RinggitPay via an HTTPS POST request. This method minimizes technical complexity, accelerating onboarding for both technical and non-technical teams.

Once redirected to the RinggitPay interface, buyers complete their transactions using their preferred payment method either Internet Banking (FPX) or Credit/Debit Card.

Following a successful transaction, RinggitPay manages the post-payment process through two simultaneous steps:

1. **Buyer Redirection:** The buyer is securely redirected back to the merchant’s front end via a form POST.
2. **Server-to-Server Notification:** A detailed API response is sent to the merchant’s back end to confirm the transaction status in real time.

This dual-notification system ensures reliable status updates, even if the buyer does not complete the redirection step.

3.Pre-Requisite

Before initiating integration with the RinggitPay payment system, merchants must complete the following steps and gather the required credentials and configuration details.

Merchant & Application Registration

Ensure your merchant account and at least one application have been successfully registered with RinggitPay. Once this is completed, reach out to the RinggitPay support team to receive your essential access credentials and keys.

1. PORTAL ACCESS CREDENTIALS

Used to manage your merchant profile and monitor transactions via the RinggitPay Portal.

Item	Description
Merchant User Credentials	Email & Password used to log in to the RinggitPay Merchant Portal to view profile information, manage applications, and access transaction history.

2. APPLICATION INTEGRATION KEYS

Required to authenticate and securely communicate with RinggitPay's hosted payment system.

Item	Description
App Id	A unique identifier assigned to your application. Used in all payment-related API calls.
Request Key	An alphanumeric key used to digitally sign payment requests sent to RinggitPay.
Response Key	An alphanumeric key used to validate the authenticity of payment responses from RinggitPay
⚠ NOTE	Always store these keys securely on your server. Avoid hard coding them into client-side applications.

3. CALL BACK AND RE-DIRECT URL’s

These URLs allow RinggitPay to return payment results back to your system

URL Type	Description	Example
API URL	A back end server-to-server callback endpoint. This is the primary and most reliable method for receiving payment results.	https://bills2u-api.ringgitpay.co/api/payment/directresponse
Redirect URL	A front end redirect URL, where the buyer is redirected after payment. Useful for customer experience but less reliable due to client-side issues.	https://payer.ringgitpay.co/payment/transactionreceipt
Return URL	Used primarily for mobile/web app integrations. If specified in a request, it will override the Redirect URL	
⚠ NOTE	Always use API URL as your system’s source of truth for payment confirmation. Redirect URL should only enhance the customer experience.	

4. TRANSACTION ENQUIRY & ENVIRONMENT URLs

Merchants can manually or programmatically verify the status of any transaction using RinggitPay's Transaction Enquiry endpoints.

Type	Payment Request	Transaction Enquiry
UAT	https://ringgitpay.co/payment	https://ringgitpay.co/transactionenquiry
PRODUCTION	https://ringgitpay.com/payment	https://ringgitpay.com/transactionenquiry
⚠ NOTE	Use the UAT URLs for testing your integration before going live.	

4.Security and Data Integrity

RinggitPay prioritizes the security and integrity of all transactions processed through its platform. To safeguard data during transmission between merchants and the payment gateway, RinggitPay utilizes a cryptographic hashing mechanism that leverages several key field values. These include the **App Id, Request Key, Response Key** and other essential parameters.

By generating and validating secure hash strings, the system ensures:

- **Data Integrity:** Prevents tampering of request and response data during transmission.
- **Authentication:** Confirms that communication is between trusted parties.
- **Non-repudiation:** Assures that requests and responses cannot be denied after being sent or received.

4.1. Request Key

Request Key is a confidential, shared secret assigned to each RinggitPay merchant. It is used to generate a one-time hash string that is embedded in all outgoing API requests to RinggitPay. This hash acts as a digital signature, verifying that the request originated from an authenticated source and that the content has not been altered in transit.

Key Characteristics:

- ✓ Unique per merchant account.
- ✓ Used to sign request payloads.
- ✓ Acts as a seed value in the hash generation algorithm.
- ✓ Should never be exposed publicly or hard-coded into client-side code.

Best Practices:

- **Keep it confidential:** Do not share the key publicly or store it in publicly accessible environments (e.g., frontend code, public repositories).
- **Rotate regularly:** Periodically update the key to minimize the risk of exposure.
- **Monitor access:** Log and audit access to the key within your system.

How to Obtain the Request Key:

1. Log in to the RinggitPay Merchant Portal.
2. Navigate to Merchant Profile.
3. Scroll down to the section labeled Request Key.
4. Alternatively, retrieve the key from the credentials sent to you by the RinggitPay Application Team.
5. Copy the key and store it securely in your back end system or key vault.

⚠ **Important:** If you suspect that your Request Key has been exposed or compromised, contact RinggitPay Support immediately to revoke and regenerate the key.

4.2. Response Key

Response Key is another merchant-specific shared secret. It is used to verify the authenticity of the response data returned by RinggitPay. Upon receiving a response, merchants must validate the cryptographic hash using the Response Key to ensure the response was generated by RinggitPay and has not been altered.

Key Characteristics:

- ✓ Unique for each merchant.
- ✓ Used for validating server responses.
- ✓ Complements the Request Key for full request-response validation.
- ✓ Must be stored securely on the server side.

Best Practices:

- ❖ Never disclose this key in any client-facing environment.
- ❖ Validate all responses from RinggitPay using the hash generated with this key.
- ❖ Audit and monitor for unexpected usage.

How to Obtain the Response Key:

1. Log in to the RinggitPay Merchant Portal.
2. Navigate to Merchant Profile.
3. Locate the field labeled Response Key.
4. Alternatively, use the key provided by the RinggitPay App Team.
5. Copy and store the key in a secure location, such as your server's environment variables or secure vault.

⚠ **Note:** In the event of suspected misuse or compromise of the Response Key, you must contact RinggitPay Support to initiate a key reset.

4.3. X-API-Key (Optional Advanced Authentication)

For merchants requiring additional validation mechanisms, RinggitPay supports the inclusion of a custom X-API-Key or similar key-value pairs in response headers. This feature enhances security by introducing a secondary verification step beyond cryptographic hashing.

Key Characteristics:

- ✓ Optional but recommended for sensitive implementations.
- ✓ Configurable per merchant via request to RinggitPay Support.
- ✓ Injected into all RinggitPay response headers upon setup.
- ✓ Can be used to validate response authenticity before proceeding with processing.

Setup Procedure:

- ❖ Generate a secure key-value pair (e.g., X-API-Key: your-secure-key) on your end.
- ❖ Contact RinggitPay Support and provide the key-value pair.
- ❖ Once configured, RinggitPay will include this key in all future response headers for your merchant account.
- ❖ Implement middleware or logic in your back end to verify the presence and correctness of the key.

⚠ **Note:** Store your API key in a secure server-side location and avoid exposing it in any front end code or public-facing resources.

5.Payment Request Message

To initiate a payment transaction through RinggitPay, merchants must send a request to the appropriate endpoint with all required and optional fields properly formatted. This section provides comprehensive details about the request format, required fields, optional data elements, and submission endpoints for both testing and production environments.

5.1. Endpoints

ENVIRONMENT	URL
UAT	https://ringgitpay.co/payment
PRODUCTION	https://ringgitpay.com/payment

5.2. Request Format

The payment request must be submitted using an HTTP POST method with the content type set to application/x-www-form-urlencoded. The following table describes each parameter included in the request:

REQUEST PARAMETERS						
S.No	Field Name	Parameter Name	Mandatory	Type	Max Length	Description
1	AppId	appId	Yes	C	20	Unique identifier assigned to each merchant by RinggitPay.
2	Currency	currency	Yes	C	4	Transaction Currency code (default: MYR)
3	Amount	amount	Yes	N	16,2	Transaction Amount

4	Order Id	orderId	Yes	C	20	Unique transaction identifier provided by the merchant
5	CheckSum	checkSum	Yes	C	-	Checksum value
6	Buyer Email	buyerEmail	No	C	50	Email address of the payer
7	Return URL	returnURL	No	C	150	URL to which the user is redirected after payment (must be URL-encoded).
8	Payer Name	accName	No	C	50	Name of the payer
9	Reference 1	ref1	No	C	50	Merchant custom reference 1
10	Reference 2	ref2	No	C	50	Merchant custom reference 2
11	Reference 3	ref3	No	C	50	Merchant custom reference 3
12	Reference 4	ref4	No	C	50	Merchant custom reference 4
13	Reference 5	ref5	No	C	50	Merchant custom reference 5
14	Reference 6	ref6	No	C	50	Merchant custom reference 6
15	Payer Account Type	accType	No	N		Bank account type of the payer. (Required if Payer Verification is enabled) Ex: 01
16	Payer Account Number	accNumber	No	N	20	Bank account number of the payer. (Required if Payer Verification is enabled)
17	Payer Buyer Id	accBuyerId	No	N	20	Unique buyer identifier. (Required if Payer Verification is enabled)

⚠ **Note:** ref1 to ref6 are optional custom reference fields. Any values sent in these fields will be returned in the payment API response, allowing merchants to track custom meta data.

5.3. Sample Request

Below is a sample HTML form using Name-Value Pair (NVP) format to initiate a payment request to the RinggitPay UAT environment:

SAMPLE REQUEST

```
<form method=”post” action = “https://ringgitpay.co/payment”>
<input type=hidden name=”appId” value="RPA-XXXXXX-XXX">
<input type=hidden name=”currency” value = “MYR”>
<input type=hidden name=”amount” value = “5000.00”>
<input type=hidden name=”orderId” value = "RP07022025001">
<input type=hidden name=”checkSum” value =
“7DAEC32208803C7ED132ABAC6987FFF8D5777CD3B688E34DFD4125E572FD3FEC”>
<input type=hidden name=”buyerEmail” value = “user@ringgitpay.com”>
<input type=hidden name=”returnURL” value =
“http%3A%2F%2Fwww.merchantsite.com%2Freceipt”>
<input type=hidden name="accName" value ="Rakesh Sharma G">
<input type=hidden name=”ref1” value = "Agri Motor">
<input type=hidden name=”ref2” value = "2025 Model">
<input type=hidden name=”ref3” value = "Petrol Engine">
<input type=hidden name=”ref4” value = "Energy Efficient">
<input type=hidden name=”ref5” value = "1000 KM">
<input type=hidden name=”ref6” value = "Free Service">
<input type=hidden name=”accType” value = “01”>
<input type=hidden name=”accBuyerId” value = “9001014332342”>
<input type=hidden name=”accNumber” value = “12345678901234567890”>
</form>
```

⚠ **Note:**

- ✓ URL (Plain): http://www.merchantsite.com/receipt
- ✓ URL Encoded: http%3A%2F%2Fwww.merchantsite.com%2Freceipt
- ✓ The returnURL must be URL-encoded.
- ✓ Ensure the return URL remains within your application's domain for security and validation purposes.

5.3.1. Payer Verification (Optional)

If Payer Verification is enabled for your merchant account, the following three fields must be included:

```
<input type="hidden" name="accType" value="01">
<input type="hidden" name="accBuyerId" value="9001014332342">
<input type="hidden" name="accNumber" value="12345678901234567890">
```

Field Usage by Payment Channel		
Field	**FPX**	**DuitNow**
accType		
accBuyerId	✓	✓
accNumber	✓	✓

5.4. Checksum Calculation

To ensure the integrity and authenticity of each transaction request, the RinggitPay system requires the inclusion of a cryptographic checksum (checkSum). This checksum is calculated using a SHA-256 hashing algorithm based on a structured combination of selected data fields and a merchant-specific REQUESTKEY.

Purpose

The checksum serves as a tamper-proof signature that verifies the request has not been altered during transmission. It protects against unauthorized changes and ensures the request originates from a legitimate source.

Checksum Generation:

To generate the checkSum value:

1. Concatenate the required field values in the exact order as documented.
2. Separate each value with a pipe (|) character.
3. Append the REQUESTKEY (provided by RinggitPay) at the end of the string.
4. Apply SHA-256 hashing to the entire string.
5. Convert the resulting hash to uppercase.

⚠ **Note:** The checksum must be generated using the raw values only. Do not include field names or labels. Avoid white space or additional formatting.

SOURCE STRING FORMAT
checkSum = appId|currency|amount|orderId|REQUESTKEY

Example

Input Values:

1. **appId:** RPA-XXXXXX-XXX
2. **currency:** MYR
3. **amount:** 5000.00
4. **orderId:** RP07022025001
5. **REQUESTKEY:** DKFJE34KL9C2HDH5

CONSTRUCTED SOURCE STRING
RPA-XXXXXX-XXX|MYR|5000.00|RP07022025001|DKFJE34KL9C2HDH5

SHA-256 Output (Uppercase)
7DAEC32208803C7ED132ABAC6987FFF8D5777CD3B688E34DFD4125E572FD3FEC

⚠ **Note:**

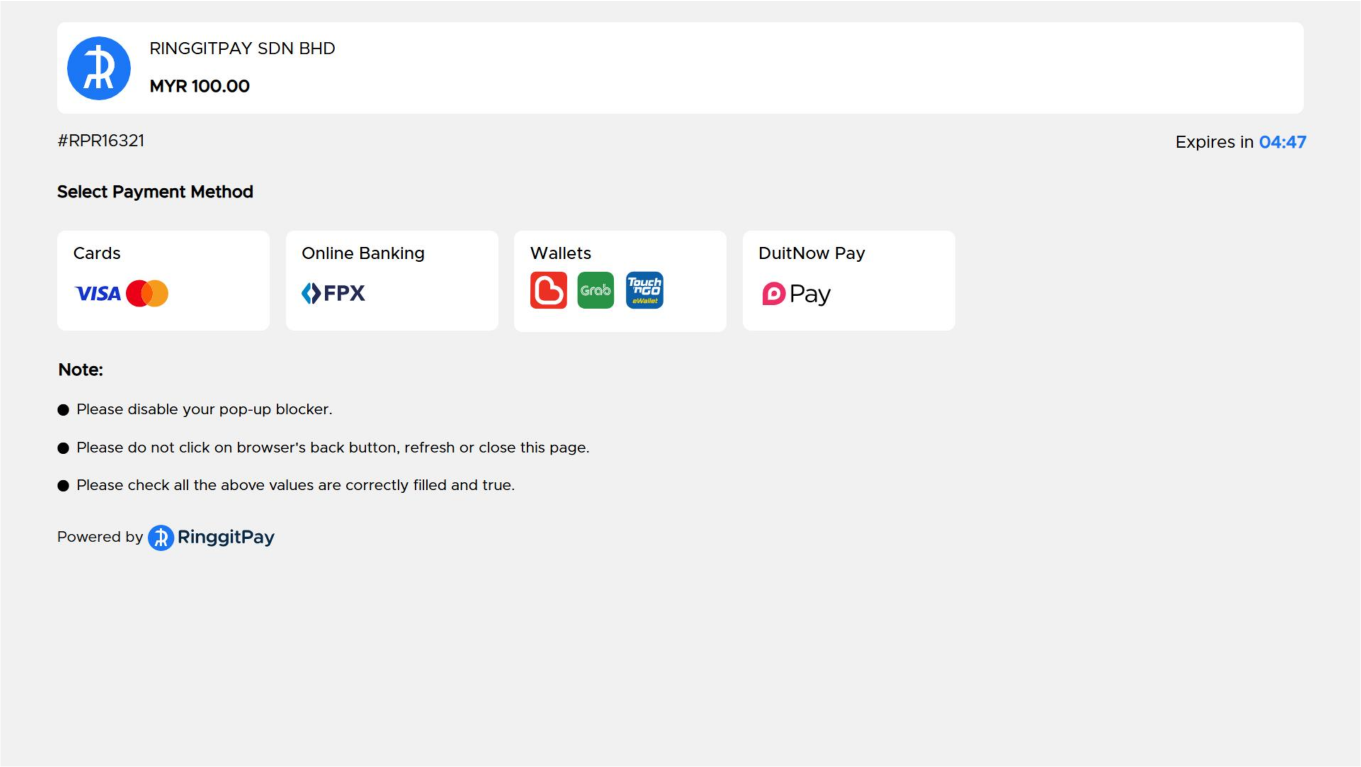
1. **Field Order Matters:** The checksum will be invalid if field order is changed.
2. **Avoid Extra Spaces:** Ensure no leading, trailing, or double spaces in the source string.
3. **Keep Your Keys Secure:** Never expose your REQUESTKEY publicly.
4. **Uppercase Only:** Always convert the final hash to uppercase before submitting.

5.5. UAT Payment Interface Overview

Upon initiating a successful payment request within the UAT (User Acceptance Testing) environment, the end user will be redirected to the RinggitPay payment interface for transaction completion.

Payment Method Presentation Logic

- ❖ **Multiple Payment Channels:** If the merchant is subscribed to more than one payment method (e.g., FPX and card payments), the FPX payment screen will be presented by default. Users may select "Change Payment Method" to switch between available options.
- ❖ **Single Payment Channel:** If only one payment method is enabled for the merchant, the interface will automatically present the corresponding payment screen without requiring manual selection.



RINGGITPAY SDN BHD

MYR 100.00

#RPR16321

Expires in 04:47

Select Payment Method

Cards | Online Banking | Wallets | DuitNow Pay

Note:

- Please disable your pop-up blocker.
- Please do not click on browser's back button, refresh or close this page.
- Please check all the above values are correctly filled and true.

Powered by RinggitPay

5.5.1. FPX (Financial Process Exchange)

FPX (Financial Process Exchange) is a secure and real-time online payment gateway that facilitates interbank fund transfers for e-commerce and online transactions in Malaysia. Operated by Payments Network Malaysia (PayNet), FPX allows customers to make payments using their existing internet banking accounts with participating banks.

FPX supports both Retail (B2C) and Corporate (B2B) banking channels, offering a reliable and seamless payment experience for consumers and businesses. Transactions performed via FPX are authenticated directly by the user’s bank, ensuring high levels of security, integrity, and traceability.
FPX is widely adopted across various industries due to its speed, convenience, and broad accessibility through most major Malaysian banks.

Upon selecting **Online Banking (FPX**) as the payment method, the corresponding FPX payment interface will be displayed.

FPX payment option supports both Retail (B2C) and Corporate (B2B) banking transactions, each with its own transaction limits and behavior.

FPX TRANSACTION LIMITS		
Banking Type	**Minimum Amount**	**Maximum Amount**
Retail Banking	MYR 1.00	MYR 30,000.00
Corporate Banking	MYR 2.00	MYR 10,00,000.00

When testing in the UAT environment, different FPX test banks may simulate varying transaction outcomes

UAT Testing Behavior - Retail Banking	
Test Bank	**Status**
SBI A	Approved/Success
SBI B	Failed

⚠ **Note:** The list of test banks and corresponding behaviors may change periodically. Please consult the RinggitPay Support Team for the most current UAT test credentials and expected outcomes.

5.5.1.1.User Journey: FPX Payment Simulation

To simulate an FPX transaction in the UAT environment, follow the steps below:

1. Access the FPX Payment Interface: Upon initiating the transaction and selecting FPX (Financial Process Exchange) as the payment method, the user is redirected to the FPX payment screen to proceed with the payment.

2. Select Payment Mode and Bank: The user must choose the appropriate payment type:

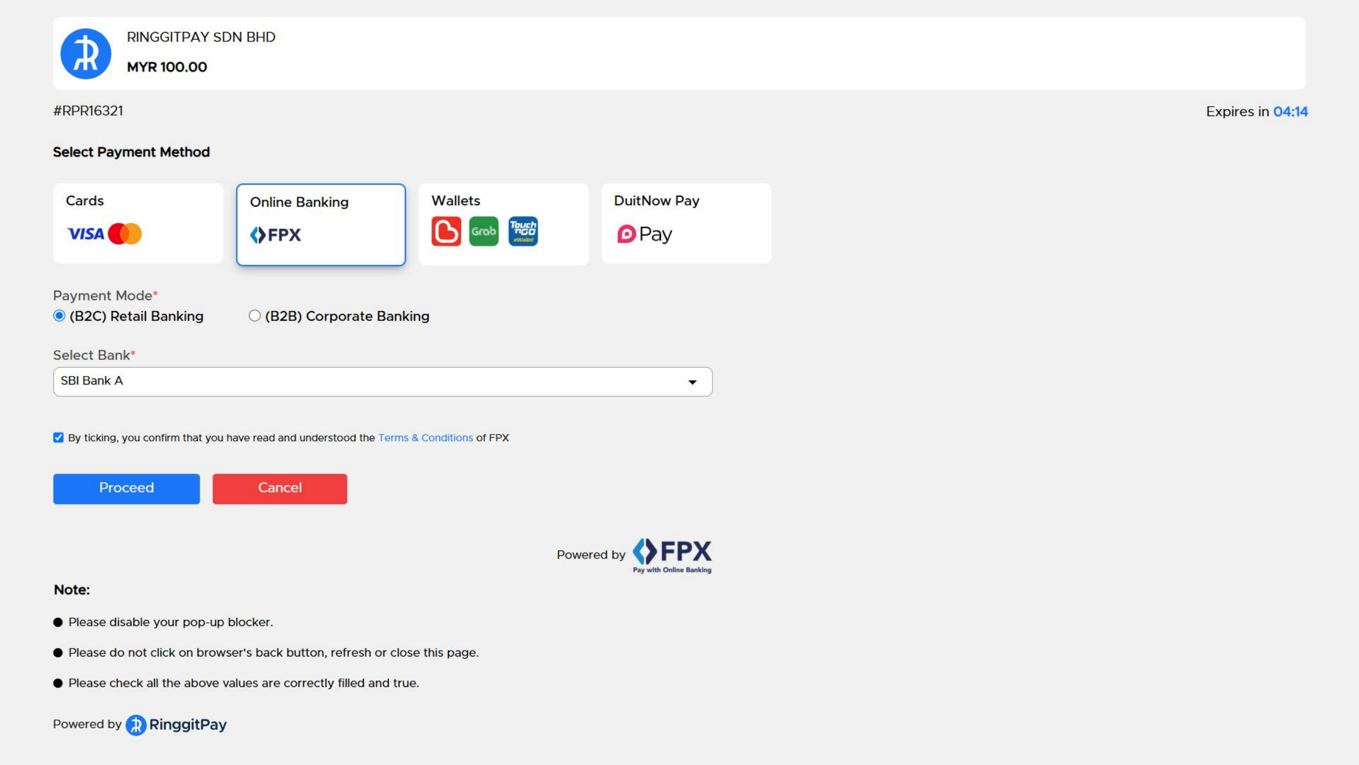
i. Retail (B2C) for individual accounts

ii. Corporate (B2B) for business or enterprise accounts

Then, select the desired test bank from the list.

3. Accept FPX Terms and Conditions: The user is required to review and agree to the FPX Terms and Conditions by checking the consent box.

4. Proceed to Payment: Click the **"Proceed"** button to initiate the payment simulation.



RINGGITPAY SDN BHD

MYR 100.00

#RPR16321

Expires in 04:14

Select Payment Method

Payment Mode*

● (B2C) Retail Banking ○ (B2B) Corporate Banking

Select Bank*

SBI Bank A

☑ By ticking, you confirm that you have read and understood the Terms & Conditions of FPX

Note:

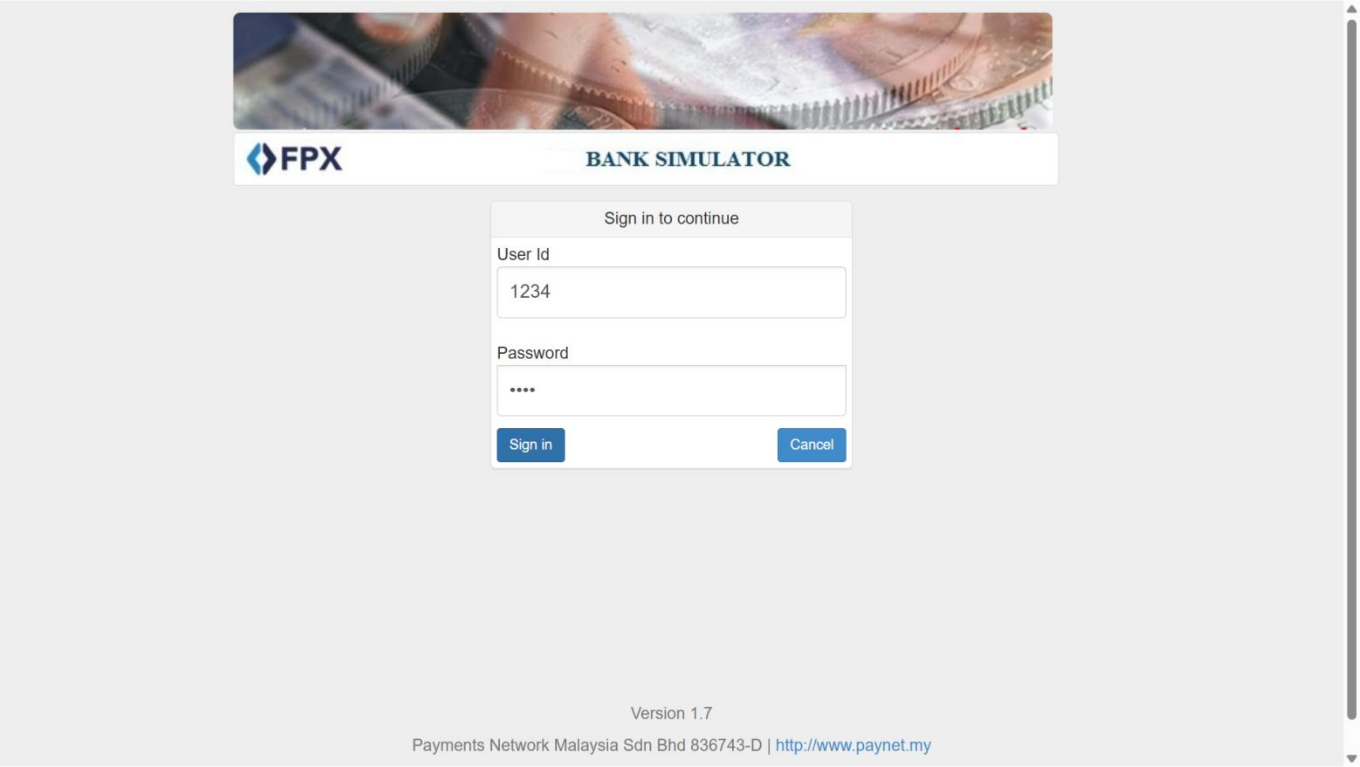
- Please disable your pop-up blocker.
- Please do not click on browser's back button, refresh or close this page.
- Please check all the above values are correctly filled and true.

5. Enter Test Credentials on FPX Simulator Screen

The user will be redirected to the FPX simulator login page. Use the following credentials:

- UserId: 1234
- Password: 1234

⚠ **Note:** These credentials are for UAT testing purposes only and do not represent real banking access.



BANK SIMULATOR

Sign in to continue

User Id: 1234

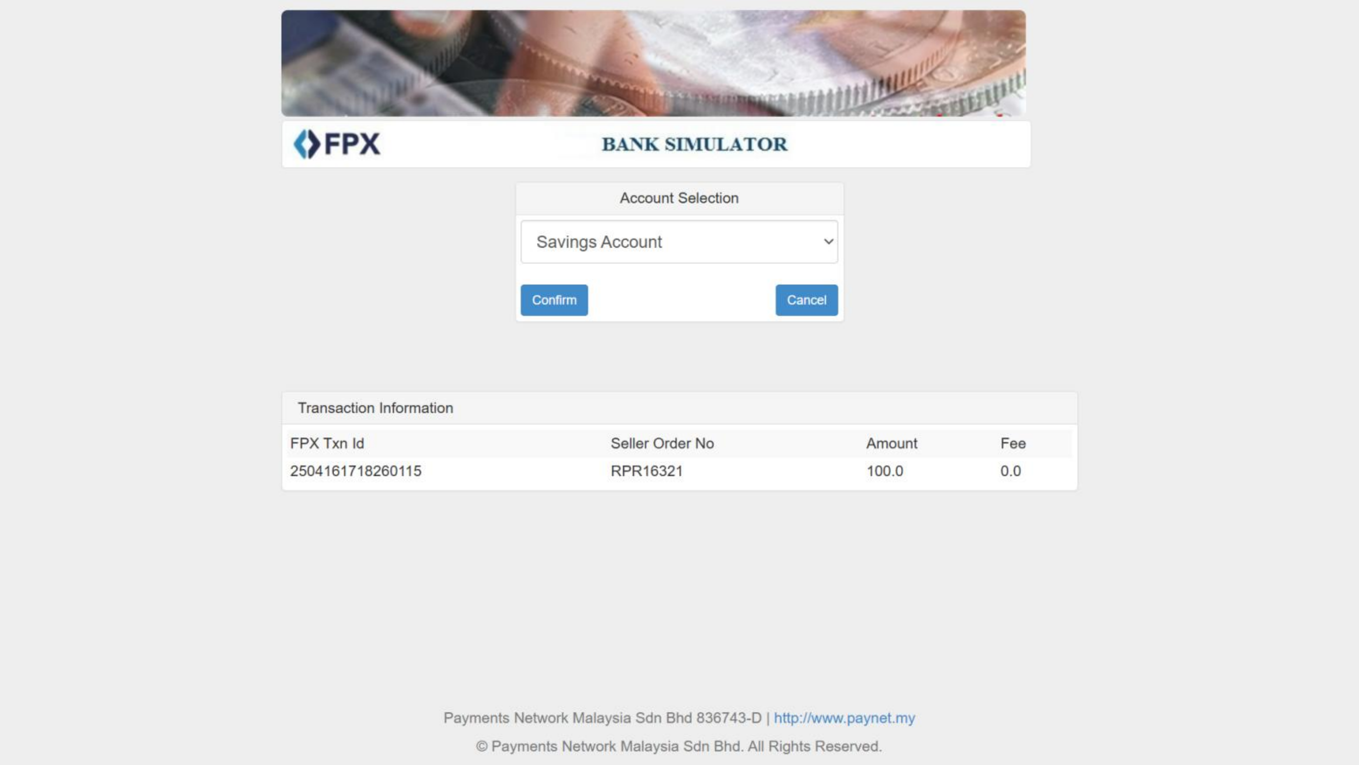
Password: ••••

Version 1.7

Payments Network Malaysia Sdn Bhd 836743-D | http://www.paynet.my

6. Select Account Type:

On the account selection screen, the user must choose the appropriate bank account type - either Savings or Current account to proceed with the transaction.



BANK SIMULATOR

Account Selection

Savings Account

Transaction Information

FPX Txn Id	Seller Order No	Amount	Fee
2504161718260115	RPR16321	100.0	0.0

Payments Network Malaysia Sdn Bhd 836743-D | http://www.paynet.my

© Payments Network Malaysia Sdn Bhd. All Rights Reserved.

7. Transaction Completion and Response Handling

i) For Retail Banking (B2C):

Upon successful completion of the payment, the user will be redirected to the configured **ReturnURL** or **RedirectURL**. Simultaneously, the same transaction response will be sent to the merchant's designated APIURL for back end processing and record keeping.

ii) For Corporate Banking (B2B):

For B2B transactions, an additional authorization step is required. All transactions will initially return with the status **PENDING (RP09)** until an authorizer manually approves or declines the transaction. Simulating Authorizer Action in UAT:

In the UAT environment, merchants must simulate the authorizer's action as follows:

Send an email to the RinggitPay team with the following details:

1. orderId
2. Action to perform: APPROVE or DECLINE

Upon simulation, a direct response message will be sent to the merchant's APIURL with the updated transaction status.

⚠ **Note:** Merchants may also use the Transaction Enquiry API to retrieve the latest status of pending, completed, or failed transactions.

5.5.2. Credit/Debit Card

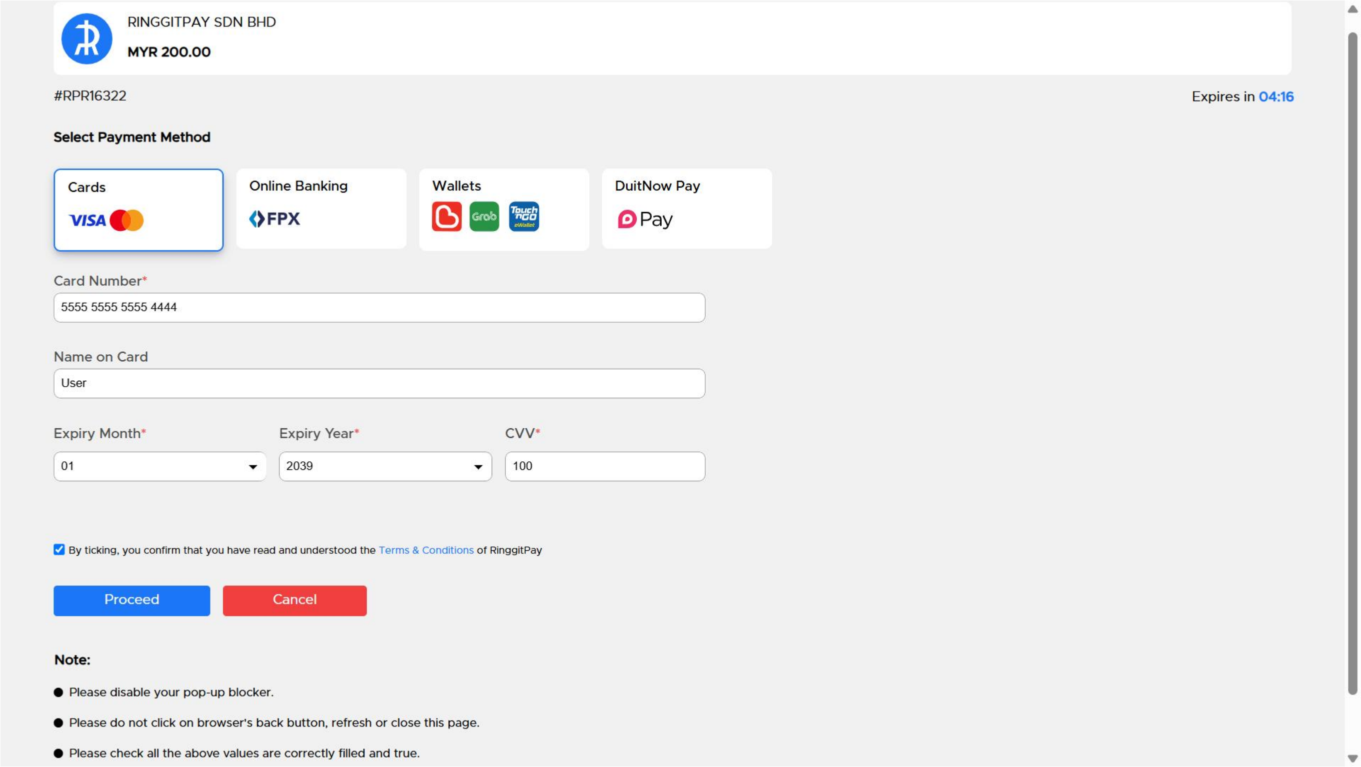
RinggitPay supports credit and debit card payments via a secure, PCI DSS- compliant payment gateway, allowing customers to perform transactions using major card networks such as Visa, MasterCard, and others. Card payments may include 3-D Secure (3DS) authentication, which provides an additional layer of security through customer verification by the issuing bank. When 3DS is enabled, users will be redirected to their bank's authentication page to complete the transaction.

All transactions are processed in real-time, with immediate responses indicating Success, Failure, or Pending, based on the validation from both the payment gateway and the issuing bank.

UAT: Test Card Details

1. Access the CC/DC Payment Interface: Upon initiating the transaction and selecting Cards as the payment method, the user is redirected to the CC/DC payment section to proceed with the payment.
2. Use the following test card details to simulate credit and debit card transactions in the User Acceptance Testing (UAT) environment. These cards are for testing purposes only and should not be used in production.

⚠ **Note:** Cards without 3-D Secure (3DS) support may result in failed transactions or be subject to the gateway's default processing rules.



RINGGITPAY SDN BHD

MYR 200.00

#RPR16322

Expires in 04:16

Select Payment Method

Card Number*: 5555 5555 5555 4444

Name on Card: User

Expiry Month*: 01

Expiry Year*: 2039

CVV*: 100

☑ By ticking, you confirm that you have read and understood the Terms & Conditions of RinggitPay

Note:

- Please disable your pop-up blocker.
- Please do not click on browser's back button, refresh or close this page.
- Please check all the above values are correctly filled and true.

UAT TEST CARD DETAILS

S.No	Payment Network Operators	Card Number	3D Secure Enrolled	Type	Country
1	Mastercard	5288 0499 9999 8964	Yes	CREDIT	MALAYSIA
2	Mastercard	5555 5555 5555 4444	Yes	DEBIT	BRAZIL
3	Mastercard	5111 1111 1111 1118	No (fail scenario)	NA	NA
4	Visa	4508 7500 1574 1019	Yes	DEBIT	MALAYSIA
5	Visa	4180 7799 9999 6392	Yes	CREDIT	MEXICO
6	Visa	4012 0000 3333 0026	No (fail scenario)	NA	NA

CARD EXPIRY MONTH & YEAR

Expiry Date	Simulate Response
Jan-39	APPROVED
May-39	DECLINED
April-27	EXPIRED CARD
August-28	TIMED OUT

CVV NUMBER	
CSV/CVV	**Simulate Response**
100	MATCH
101	NOT PROCESSED
102	NO MATCH

Note: Test card availability may change. Contact the RinggitPay team for the latest test card information.

3. Sample Transaction Simulation

To simulate a successful payment:

- Card Number: 5123 4500 0000 0008
- Expiry month and year: Jan-39
- CVV: 100

After filling in the test card details, agree to the terms and conditions, then click **'Proceed'** to complete the simulation

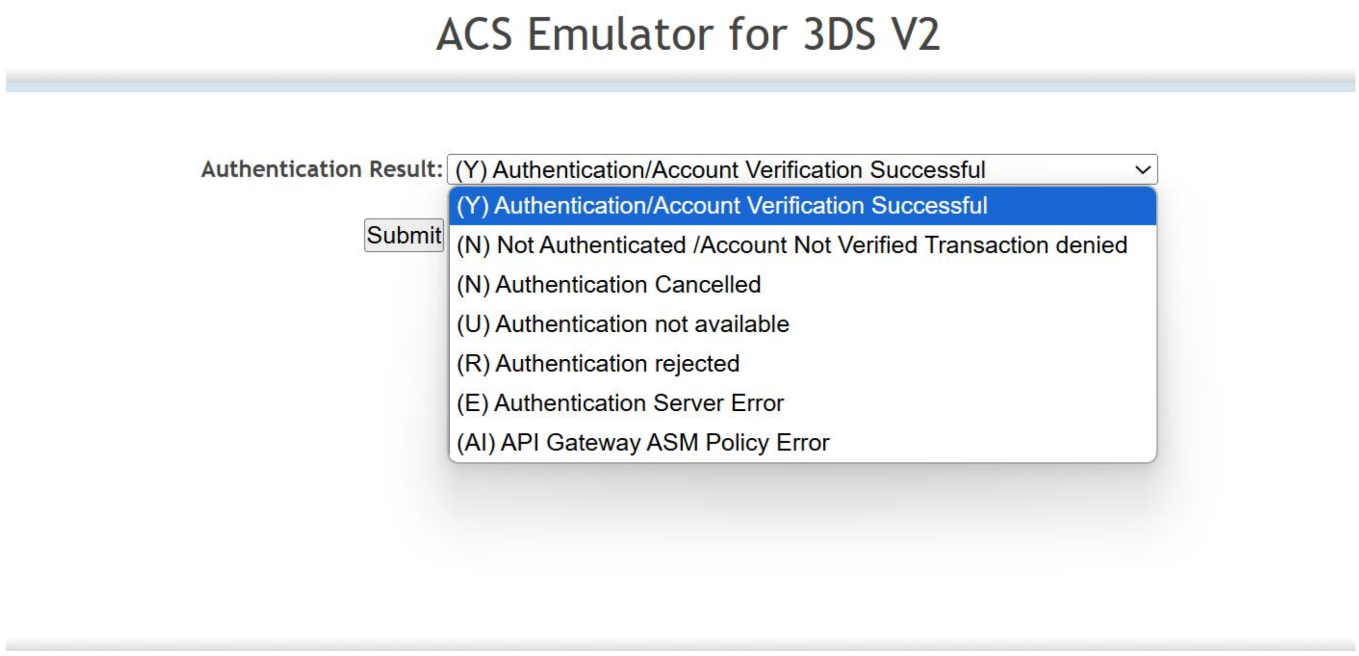
4. 3-D Secure (3DS) v2 Authentication - ACS Emulator

During a simulated 3-D Secure (3DS) v2 transaction, users will be redirected to the ACS (Access Control Server) Emulator screen, which replicates the cardholder authentication step required in real-world scenarios.

ACS is the system operated by the issuing bank to authenticate the cardholder during 3DS transactions
.
Step: Authentication Simulation

i) On the ACS Emulator screen, Leave the 'Authentication Result' as the default selection.
ii) Click the 'Submit' button to proceed.

This action simulates a successful cardholder authentication.



ACS Emulator for 3DS V2

Authentication Result: (Y) Authentication/Account Verification Successful

- (Y) Authentication/Account Verification Successful
- (N) Not Authenticated /Account Not Verified Transaction denied
- (N) Authentication Cancelled
- (U) Authentication not available
- (R) Authentication rejected
- (E) Authentication Server Error
- (AI) API Gateway ASM Policy Error

Upon submission, the user will be redirected back to the merchant’s Return URL with the appropriate response.

⚠ **Note:** No actual authentication or credentials are required in the UAT environment. The ACS Emulator is designed purely for testing and simulates the entire 3DS authentication flow.

5.6. Production Payment Screens

In the Production environment, once the payment request is submitted, the redirection behavior will be as follows:

a. FPX (Online Banking):
The user will be redirected to the selected bank’s login page to complete the transaction via the bank's secure interface.

b. Credit/Debit Card:
The user will be redirected to the appropriate 3-D Secure (3DS) authentication page, as determined by the issuing bank and card details provided.

⚠ **Note:** These redirections ensure secure, real-time transaction processing and cardholder authentication as per industry standards.

6.Payment Response Message

Upon completion of a transaction, RinggitPay communicates the outcome back to the merchant system using two standardized methods: Direct Response (server-to-server) and Indirect Response (client-side browser redirection). These responses ensure accurate transaction tracking, real-time updates, and a seamless user experience.

All response messages are structured in Name/Value Pair (NVP) format and transmitted via HTML form elements. Each response field is prefixed with **“rp”** to clearly identify it as part of the RinggitPay response schema.

1. Direct Response (via APIURL)

The Direct Response is a backend-to-backend message, securely transmitted from RinggitPay to the merchant's designated APIURL. This message is sent once RinggitPay receives the final transaction status from the payment provider or channel (e.g., FPX, credit card gateway).
Key Features:

❖ **Method:** HTTPS POST (server-to-server communication)
❖ **Trigger:** Sent immediately upon transaction status confirmation
❖ **Security:** Can include custom HTTP headers for authentication
❖ **Purpose:** Enables the merchant backend to process transaction results reliably without user interaction

Message Security:

To ensure the integrity and authenticity of messages exchanged between RinggitPay and merchants, a digital Signature is included in applicable responses.

✓ **Signature -** A cryptographic hash generated using HMAC or SHA algorithms. It enables the merchant to verify that the message has not been altered and originates from RinggitPay.

Merchants are strongly encouraged to implement signature verification on their server to prevent tampering and ensure trust in all received responses.

Optional Custom HTTP Headers:

To further strengthen API communication security, RinggitPay supports the inclusion of optional custom HTTP headers in its responses, upon request by the merchant:

✓ **X-API-Key** - A unique key assigned to the merchant, used to identify and authenticate API interactions.

These headers offer an additional layer of verification and are recommended for merchants implementing enhanced security protocols.

⚠ Security Note:

Merchants are advised to implement validation mechanisms for these headers to verify the source and integrity of the response. This is especially important for preventing tampering and ensuring trust in the communication process

2. Indirect Response (via ReturnURL/RedirectURL)

The Indirect Response is a client-side redirect triggered after the transaction process completes. The customer's browser is redirected to the merchant's configured ReturnURL or RedirectURL along with transaction data.

Key Features:

- ❖ **Method:** HTTP POST (preferred) or GET (optional)
- ❖ **Purpose:** To allow the merchant to present the result of the payment to the user
- ❖ **Timing:** Occurs immediately after the payment gateway interaction
- ❖ **Security:** Includes checksum hash (rpCheckSum) for data validation

⚠ **Note:** While indirect responses enhance user experience, they should not be used as the sole method for transaction status confirmation due to the nature of browser redirection.

6.1. Response Format

RinggitPay responds to payment transactions using a Name-Value Pair (NVP) format. The response is sent to both the ReturnURL (browser redirection) and the APIURL (server-to-server), depending on the configuration and the payment channel.

RESPONSE PARAMETERS

S.No	Field Name	Parameter Name	Type	Max Length	Description
1	AppId	rp_appId	C	20	Unique identifier assigned to each merchant by RinggitPay.
2	Currency	rp_currency	C	4	Transaction Currency code (default: MYR)
3	Amount	rp_amount	N	16,2	Transaction Amount
4	Status Code	rp_statusCode	N	4	Status code issued by RinggitPay (prefix RP or IR)
5	Order Id	rp_orderId	C	20	Unique transaction identifier provided by the merchant
6	Transaction Reference	rp_transactionRef	N	15	Unique reference generated by RinggitPay
7	PaymentMode	rp_paymentMode	N	15	Payment channel chosen by the buyer (e.g., FPX, CC, DC, NA).
8	TransactionTime	rp_txnTime	N	14	Date and time when RinggitPay processed the transaction, in UTC, formatted as YYYYMMDDHHMMSS (24- hour clock).
9	CheckSum	rp_checkSum	C	-	SHA-256 checksum
10	Remarks	rp_remarks	C	-	Remarks/comments returned by the bank or gateway

11	Payer Account Verification	rp_xtraInfo	N		Included only for FPX transactions when the Payer Identity Verification feature is enabled; this field is excluded from the checksum calculation. *See Section 6.1.1 for details
12	Payment Country	rp_paymentCountry	C	30	Country of payment: For FPX transactions, “Malaysia” For card transactions, the issuing bank’s country (derived from the card number), provided only on successful transactions
13	Payer Email	rp_buyerEmail	C		Email Id of the payer
14	Message Type	rp_messageType			*Applicable only for settlement response
15	Reference 1	ref1	C	50	Merchant custom reference 1
16	Reference 2	ref2	C	50	Merchant custom reference 2
17	Reference 3	ref3	C	50	Merchant custom reference 3
18	Reference 4	ref4	C	50	Merchant custom reference 4
19	Reference 5	ref5	C	50	Merchant custom reference 5
20	Reference 6	ref6	C	50	Merchant custom reference 6

6.1.1. Payer Account Verification

The **rp_xtraInfo** field is a 6-digit numeric value used only for FPX transactions when the Payer Identity Verification feature is enabled. It encodes three segments, each 2 digits long, representing verification information.

rp_xtraInfo = Account Type (2 digits) + Account Number Verification (2 digits) + Buyer ID Verification (2 digits)

PAYER ACCOUNT VERIFICATION (rp_xtraInfo)				
S.No	**Field Name**	**Type**	**Max Length**	**Description**
1	Account Type	N	2	First 2 digits of rp_xtrainfo. 00 - Undetermined 01 - CASA 02 - LCA
2	Account Number	N	2	Third and fourth digit of rp_xtrainfo. 00 - Undetermined 01 - True 02 - False

3	Buyer ID	N	2	Last 2 digits of rp_xtrainfo. 00 - Undetermined 01 - True 02 - False

6.2. Sample Response

Below is a sample response in Name-Value Pair (NVP) format returned by RinggitPay after processing a transaction:

SAMPLE RESPONSE

```
{
    "rp_appId": "RPA-XXXXXX-XXX",
    "rp_currency": "MYR",
    "rp_amount": "5000.00",
    "rp_statusCode": "RP00",
    "rp_orderId": "RP07022025001",
    "rp_transactionRef": "250X071X056X04",
    "rp_paymentMode": "FPX",
    "rp_txnTime": "20250207145108",
    "rp_checkSum":
"0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436",
    "rp_remarks": "Approved|B2C|RHB",
    "rp_xtraInfo": "010000",
    "rp_paymentCountry": "Malaysia",
    "rp_buyerEmail": "user@ringgitpay.com",
    "rp_messageType": null,
    "rp_ref1": "Agri Motor",
    "rp_ref2": "2025 Model",
    "rp_ref3": "Petrol Engine",
    "rp_ref4": "Energy Efficient",
    "rp_ref5": "1000 KM",
    "rp_ref6": "Free Service"
}
```

6.3. Checksum Verification

To maintain the integrity, confidentiality, and authenticity of each transaction response, RinggitPay mandates the inclusion of a cryptographic checksum **rp_checkSum**. This checksum is a **SHA-256** hash generated from specific response fields combined with a merchant-specific **RESPONSEKEY.**

Purpose of the Checksum

The rp_checkSum field acts as a digital signature to:

✓ Ensure the response has not been tampered with during transmission.
✓ Confirm that the data originates exclusively from RinggitPay.
✓ Safeguard against unauthorized modifications or fraudulent responses.

This verification mechanism provides an essential layer of data integrity assurance between RinggitPay and the merchant's system.

Checksum Validation Process

1. To validate the rp_checkSum received in the response, follow the steps below:
2. Concatenate the required field values in the exact sequence as specified.
3. Use a pipe (|) character as a delimiter between field values.
4. Append the RESPONSEKEY (provided by RinggitPay) at the end of the concatenated string.
5. Apply the SHA-256 hashing algorithm to the entire string.
6. Convert the resulting hash to uppercase hexadecimal format.
7. Compare the generated hash with the rp_checkSum received in the response.

Implementation Notes

- Use only raw field values - do not include field names or labels.
- Exclude fields that are null or not defined, unless explicitly required.
- Ensure there are no leading/trailing white spaces, empty pipes, or formatting inconsistencies.

All fields must appear in the exact order defined below.

SOURCE STRING FORMAT
rp_checkSum = rp_appId|rp_currency|rp_amount|rp_statusCode|rp_orderId|rp_transactionRef |RESPONSEKEY

Example

Input Values:

1. **rp_appId:** RPA-XXXXXX-XXX
2. **rp_currency:** MYR
3. **rp_amount:** 5000.00
4. **rp_statusCode:** RP00
5. **rp_orderId:** RP07022025001
6. **rp_transactionRef:** 250X071X056X04
7. **RESPONSEKEY:** DKFJE34KL9C2HDH5

CONSTRUCTED SOURCE STRING
RPA-XXXXXX-XXX|MYR|5000.00|RP00|RP07022025001|250X071X056X04|DKFJE34KL9C2HDH5

SHA-256 Output (Uppercase)
0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436

After applying SHA-256 and converting to uppercase, the resulting hash must match the value of **rp_checkSum** in the transaction response.

⚠ **Note:**

1. Always perform checksum validation on the server side.
2. If the computed checksum does not match the one received, reject the response and log it for further investigation.
3. Do not hard code the **RESPONSEKEY** in client-side code or expose it publicly.

6.4. Response Handling and Testing

To support a streamlined and uninterrupted payment experience, RinggitPay does not render a dedicated payment response screen to the user upon transaction completion. Instead, all response data is transmitted directly to the merchant's back end system via the configured API endpoint.

Handling Incomplete Transactions

In scenarios where a payment is not immediately finalized—such as user abandonment, gateway delays, or connectivity issues - the transaction status will initially be marked as **PENDING**.

1. If the transaction is delayed due to FPX or credit card (CC) timeout, it will remain in the **PENDING** state until further resolution.
2. Once a final status is determined (e.g., SUCCESS or FAILED), RinggitPay will issue a deferred status update to the merchant's registered API callback URL.

Transaction Status Enquiry (Optional)

To proactively determine the status of a transaction, merchants can initiate a Transaction Enquiry request via the provided RinggitPay API. This allows real-time retrieval of the current transaction status regardless of whether the transaction is still pending or has already been completed.

7.Enquiry Request Message

The Transaction Enquiry API enables merchants to programmatically retrieve the current status of a transaction. This is especially beneficial in scenarios where:

1. The transaction status remains PENDING due to payment delays or interruptions.
2. The response callback from RinggitPay has not yet been received.
3. Confirmation of a transaction's final outcome is required for reconciliation or customer support.

By using this API, merchants can ensure accurate and timely tracking of payment statuses, improving both operational reliability and the end-user experience.

7.1. Endpoints

ENVIRONMENT	URL
UAT	https://ringgitpay.co/transactionenquiry
PRODUCTION	https://ringgitpay.com/transactionenquiry

7.2. Request Format

The **Transaction Enquiry Request** must be submitted using the **HTTP POST** method to the appropriate RinggitPay API endpoint.

The following table defines the parameters required in the request body:

TRANSACTION ENQUIRY REQUEST PARAMETERS						
S.No	**Field Name**	**Parameter Name**	**Mandatory**	**Type**	**Max Length**	**Description**
1	AppId	appId	Yes	C	20	Unique identifier assigned to the merchant by RinggitPay.
2	Order Id	orderId	Yes	C	20	Unique order reference provided by the merchant
3	Transaction Reference	transactionRef	No	N	20	Unique reference number generated by RinggitPay (optional)
4	CheckSum	checkSum	Yes	C	-	SHA-256 hash used to verify the integrity and authenticity of the request.

7.3. Sample Request

Below is a sample Name-Value Pair (NVP) format used to initiate a transaction enquiry request to the RinggitPay:

SAMPLE REQUEST (If Transaction Reference is available)

```
{
"appId":"RPA-Bills2U-380",
"orderId":"RPR16265",
"transactionRef":"250408051730360",
"checkSum":"55FAE21BFCB034B37B353814E7E57C01FF5421E6CDF86BB85FD937324D9DEDA3"
}
```

SAMPLE REQUEST (If Transaction Reference is not available)

```
{
"appId":"RPA-Bills2U-380",
"orderId":"RPR16265",
"checkSum":"55FAE21BFCB034B37B353814E7E57C01FF5421E6CDF86BB85FD937324D9DEDA3"
}
```

7.4. Check Sum Calculation

To ensure the integrity and authenticity of each transaction enquiry request, the RinggitPay system requires the inclusion of a cryptographic checksum (checkSum). This checksum is calculated using a SHA-256 hashing algorithm based on a structured combination of selected data fields and a merchant-specific REQUESTKEY.

Purpose

The checksum serves as a tamper-proof signature that verifies the request has not been altered during transmission. It protects against unauthorized changes and ensures the request originates from a legitimate source.

Checksum Generation:

To generate the checkSum value:

1. Concatenate the required field values in the exact order as documented.
2. Separate each value with a pipe (|) character.
3. Append the REQUESTKEY (provided by RinggitPay) at the end of the string.
4. Apply SHA-256 hashing to the entire string.
5. Convert the resulting hash to uppercase.

⚠ **Note:** The checksum must be generated using the raw values only. Do not include field names or labels. Avoid white space or additional formatting.

SOURCE STRING FORMAT	
If Transaction reference is available	checkSum = appId|orderId|Transaction reference|REQUESTKEY
If Transaction reference is not available	checkSum = appId|orderId||REQUESTKEY

Example: Input Values:-

1. **appId:** RPA-XXXXXX-XXX
2. **orderId:** RP07022025001
3. **transactionRef** : 250X071X056X04
4. **REQUESTKEY:** DKFJE34KL9C2HDH5

CONSTRUCTED SOURCE STRING	
If Transaction reference is available	RPA-XXXXXX-XXX|RP07022025001|250X071X056X04|DKFJE34KL9C2HDH5
If Transaction reference is not available	RPA-XXXXXX-XXX|RP07022025001||DKFJE34KL9C2HDH5

SHA-256 Output (Uppercase)	
If Transaction reference is available	BF131DE021F794863927474816FC7C92DDF10740D16D7CF2C173780CD480368E
If Transaction reference is not available	258BC53EA5D8D3F515D9DDEA531E838027431749619131E9FC1CDBF74677AC48

⚠ **Note:**

1. **Field Order Matters:** The checksum will be invalid if field order is changed.
2. **Avoid Extra Spaces:** Ensure no leading, trailing, or double spaces in the source string.
3. **Keep Your Keys Secure:** Never expose your REQUESTKEY publicly.
4. **Uppercase Only:** Always convert the final hash to uppercase before submitting.

8. Enquiry Response Message

All enquiry response messages are structured using the Name/Value Pair (NVP) format and transmitted via HTML form elements. Each response field is prefixed with rp to clearly identify it as part of the RinggitPay response schema, ensuring consistent parsing and interpretation on the merchant's end.

Message Security

To ensure the integrity and authenticity of messages exchanged between RinggitPay and merchants, a digital Signature is included in applicable responses.

- ✓ **Signature -** A cryptographic hash generated using HMAC or SHA algorithms. It enables the merchant to verify that the message has not been altered and originates from RinggitPay.

Merchants are strongly encouraged to implement signature verification on their server to prevent tampering and ensure trust in all received responses.

Optional Custom HTTP Headers

To further strengthen API communication security, RinggitPay supports the inclusion of optional custom HTTP headers in its responses, upon request by the merchant:

- ✓ **X-API-Key** - A unique key assigned to the merchant, used to identify and authenticate API interactions.

These headers offer an additional layer of verification and are recommended for merchants implementing enhanced security protocols.

⚠ **Security Note:**

Merchants are advised to implement validation mechanisms for these headers to verify the source and integrity of the response. This is especially important for preventing tampering and ensuring trust in the communication process.

8.1. Response Format

RinggitPay responds to enquiry requests using a Name-Value Pair (NVP) format.

ENQUIRY RESPONSE PARAMETERS

S.No	Field Name	Parameter Name	Type	Max Length	Description
1	AppId	rp_appId	C	20	Unique identifier assigned to each merchant by RinggitPay.
2	Currency	rp_currency	C	4	Transaction Currency code (default: MYR)
3	Amount	rp_amount	N	16,2	Transaction Amount

4	Status Code	rp_statusCode	N	4	Status code issued by RinggitPay (prefix RP or IR)
5	Order Id	rp_orderId	C	20	Unique transaction identifier provided by the merchant
6	Transaction Reference	rp_transactionRef	N	15	Unique reference generated by RinggitPay
7	PaymentMode	rp_paymentMode	N	15	Payment channel chosen by the buyer (e.g., FPX, CC, DC, NA).
8	TransactionTime	rp_txnTime	N	14	Date and time when RinggitPay processed the transaction, in UTC, formatted as YYYYMMDDHHMMSS (24- hour clock).
9	CheckSum	rp_checkSum	C	-	SHA-256 checksum
10	Remarks	rp_remarks	C	-	Remarks/comments returned by the bank or gateway
11	Payer Account Verification	rp_xtraInfo	N		Included only for FPX transactions when the Payer Identity Verification feature is enabled; this field is excluded from the checksum calculation. *See Section 8.1.1 for details
12	Payment Country	rp_paymentCountry	C	30	Country of payment: For FPX transactions, “Malaysia” For card transactions, the issuing bank’s country (derived from the card number), provided only on successful transactions
13	Payer Email	rp_buyerEmail	C		Email Id of the payer
14	Message Type	rp_messageType			*Applicable only for settlement response
15	Reference 1	ref1	C	50	Merchant custom reference 1
16	Reference 2	ref2	C	50	Merchant custom reference 2
17	Reference 3	ref3	C	50	Merchant custom reference 3
18	Reference 4	ref4	C	50	Merchant custom reference 4
19	Reference 5	ref5	C	50	Merchant custom reference 5
20	Reference 6	ref6	C	50	Merchant custom reference 6

8.1.1. Payer Account Verification

The **rp_xtraInfo** field is a 6-digit numeric value used only for FPX transactions when the Payer Identity Verification feature is enabled. It encodes three segments, each 2 digits long, representing verification information.

rp_xtraInfo = Account Type (2 digits) + Account Number Verification (2 digits) + Buyer ID Verification (2 digits)

PAYER ACCOUNT VERIFICATION (rp_xtraInfo)				
S.No	**Field Name**	**Type**	**Max Length**	**Description**
1	Account Type	N	2	First 2 digits of rp_xtrainfo. 00 – Undetermined 01 – CASA 02 – LCA
2	Account Number	N	2	Third and fourth digit of rp_xtrainfo. 00 – Undetermined 01 – True 02 – False
3	Buyer ID	N	2	Last 2 digits of rp_xtrainfo. 00 – Undetermined 01 – True 02 – False

8.2. Sample Response

The following is a sample response in Name-Value Pair (NVP) format, as returned by RinggitPay in response to an enquiry request.

SAMPLE RESPONSE

```
{
    "rp_appId": "RPA-XXXXXX-XXX",
    "rp_currency": "MYR",
    "rp_amount": "5000.00",
    "rp_statusCode": "RP00",
    "rp_orderId": "RP07022025001",
    "rp_transactionRef": "250X071X056X04",
    "rp_paymentMode": "FPX",
    "rp_txnTime": "20250207145108",
    "rp_checkSum":
"0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436",
    "rp_remarks": "Approved|B2C|RHB",
    "rp_xtraInfo": "010000",
    "rp_paymentCountry": "Malaysia",
    "rp_buyerEmail": "user@ringgitpay.com",
    "rp_messageType": null,
    "rp_ref1": "Agri Motor",
    "rp_ref2": "2025 Model",
    "rp_ref3": "Petrol Engine",
    "rp_ref4": "Energy Efficient",
    "rp_ref5": "1000 KM",
    "rp_ref6": "Free Service"
}
```

8.3. Checksum Verification

To maintain the integrity, confidentiality, and authenticity of each enquiry response, RinggitPay mandates the inclusion of a cryptographic checksum **rp_checkSum**. This checksum is a **SHA-256** hash generated from specific response fields combined with a merchant-specific **RESPONSEKEY.**

Purpose of the Checksum

The rp_checkSum field acts as a digital signature to:

- ✓ Ensure the response has not been tampered with during transmission.
- ✓ Confirm that the data originates exclusively from RinggitPay.
- ✓ Safeguard against unauthorized modifications or fraudulent responses.

This verification mechanism provides an essential layer of data integrity assurance between RinggitPay and the merchant’s system.

Checksum Validation Process

To validate the rp_checkSum received in the response, follow the steps below:
1. Concatenate the required field values in the exact sequence as specified.
2. Use a pipe (|) character as a delimiter between field values.
3. Append the RESPONSEKEY (provided by RinggitPay) at the end of the concatenated string.
4. Apply the SHA-256 hashing algorithm to the entire string.
5. Convert the resulting hash to uppercase hexadecimal format.
6. Compare the generated hash with the rp_checkSum received in the response.

Implementation Notes

- ❖ Use only raw field values - do not include field names or labels.
- ❖ Exclude fields that are null or not defined, unless explicitly required.
- ❖ Ensure there are no leading/trailing white spaces, empty pipes, or formatting inconsistencies.

All fields must appear in the exact order defined below.

SOURCE STRING FORMAT
rp_checkSum = rp_appId|rp_currency|rp_amount|rp_statusCode|rp_orderId|rp_transactionRef |RESPONSEKEY

Example

Input Values:

1. **rp_appId:** RPA-XXXXXX-XXX
2. **rp_currency:** MYR
3. **rp_amount:** 5000.00
4. **rp_statusCode:** RP00
5. **rp_orderId:** RP07022025001
6. **rp_transactionRef:** 250X071X056X04
7. **RESPONSEKEY:** DKFJE34KL9C2HDH5

<table>
<tr><th>CONSTRUCTED SOURCE STRING</th></tr>
<tr><td>RPA-XXXXXX-XXX|MYR|5000.00|RP00|RP07022025001|250X071X056X04|DKFJE34KL9C2HDH5</td></tr>
</table>

SHA-256 Output (Uppercase)
0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436

After applying SHA-256 and converting to uppercase, the resulting hash must match the value of **rp_checkSum** in the transaction response.

⚠ **Security Note:**

1. Always perform checksum validation on the server side.
2. If the computed checksum does not match the one received, reject the response and log it for further investigation.
3. Do not hard code the **RESPONSEKEY** in client-side code or expose it publicly.

9.Settlement Response Message

RinggitPay issues automated settlement notifications to merchants upon successful completion of transaction settlements. These notifications are sent directly to the merchant’s designated endpoint and conform to the Name/Value Pair (NVP) format. Each data element in the message is prefixed with rp_ to maintain consistency across RinggitPay's response schema.

These notifications provide key details regarding the settlement, including transaction reference, date, amount, and optional remarks. Merchants should validate and log these messages for reconciliation and auditing purposes.

9.1. Response Format

Settlement response includes the following fields, along with parameters that are consistent with those found in the payment/enquiry responses, ensuring uniformity across all transaction work flows.

SETTLEMENT RESPONSE PARAMETERS					
S.No	**Field Name**	**Parameter Name**	**Type**	**Max Length**	**Description**
1	MessageType	rp_messageType	C	2	Identifies the message type. For settlement, this is typically TS
2	SettlementDate	rp_settlementDate	D		The date on which the settlement occurred, in DD-MM-YYYY format.
3	SettlementAmount	rp_settlementamount	N	16,2	The total amount settled for the transaction.
4	Note	rp_settlementnote	C	50	Optional remarks or notes related to the settlement.

9.2. Sample Response

SAMPLE RESPONSE

```
{
    "rp_appId": "RPA-XXXXXX-XXX",
    "rp_currency": "MYR",
    "rp_amount": "5000.00",
    "rp_statusCode": "RP00",
    "rp_orderId": "RP07022025001",
    "rp_transactionRef": "250X071X056X04",
    "rp_paymentMode": "FPX",
    "rp_txnTime": "20250207145108",
    "rp_checkSum":
"0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436",
    "rp_remarks": "Approved|B2C|RHB",
    "rp_xtraInfo": "010000",
    "rp_paymentCountry": "Malaysia",
    "rp_buyerEmail": "user@ringgitpay.com",
    "rp_messageType": null,
    "rp_ref1": "Agri Motor",
    "rp_ref2": "2025 Model",
    "rp_ref3": "Petrol Engine",
    "rp_ref4": "Energy Efficient",
    "rp_ref5": "1000 KM",
    "rp_ref6": "Free Service",
    "rp_messageType":"TS",
    "rp_settlementDate":"08-02-2025",
    "rp_settlementAmount":"4990.00",
    "rp_settlementNote":"Successfully settled to the merchant account"
}
```

9.3. Checksum Verification

To maintain the integrity, confidentiality, and authenticity of each settlement response, RinggitPay mandates the inclusion of a cryptographic checksum **rp_checkSum**. This checksum is a **SHA-256** hash generated from specific response fields combined with a merchant-specific **RESPONSEKEY.**

Purpose of the Checksum

The rp_checkSum field acts as a digital signature to:

- ✓ Ensure the response has not been tampered with during transmission.
- ✓ Confirm that the data originates exclusively from RinggitPay.
- ✓ Safeguard against unauthorized modifications or fraudulent responses.

This verification mechanism provides an essential layer of data integrity assurance between RinggitPay and the merchant's system.

Checksum Validation Process

To validate the rp_checkSum received in the response, follow the steps below:

1. Concatenate the required field values in the exact sequence as specified.
2. Use a pipe **(|)** character as a delimiter between field values.
3. Append the **RESPONSEKEY** (provided by RinggitPay) at the end of the concatenated string.
4. Apply the **SHA-256 hashing algorithm** to the entire string.
5. Convert the resulting hash to uppercase hexadecimal format.
6. Compare the generated hash with the **rp_checkSum** received in the response.

Implementation Notes

- ❖ Use only raw field values - do not include field names or labels.
- ❖ Exclude fields that are null or not defined, unless explicitly required.
- ❖ Ensure there are no leading/trailing white spaces, empty pipes, or formatting inconsistencies.

All fields must appear in the exact order defined below.

SOURCE STRING FORMAT
rp_checkSum = rp_appId|rp_currency|rp_amount|rp_statusCode|rp_orderId|rp_transactionRef |RESPONSEKEY

Example

Input Values:
1. **rp_appId:** RPA-XXXXXX-XXX
2. **rp_currency:** MYR
3. **rp_amount:** 5000.00
4. **rp_statusCode:** RP00
5. **rp_orderId:** RP07022025001
6. **rp_transactionRef:** 250X071X056X04
7. **RESPONSEKEY:** DKFJE34KL9C2HDH5

<table>
<tr><th>CONSTRUCTED SOURCE STRING</th></tr>
<tr><td>RPA-XXXXXX-XXX|MYR|5000.00|RP00|RP07022025001|250X071X056X04|DKFJE34KL9C2HDH5</td></tr>
</table>

<table>
<tr><th>SHA-256 Output (Uppercase)</th></tr>
<tr><td>0B8739AC55F57A90F069585B36FB7638B1525E7B1D7AEDDA8F22B5AA45C90436</td></tr>
</table>

After applying SHA-256 and converting to uppercase, the resulting hash must match the value of **rp_checkSum** in the transaction response.

⚠ **Security Note:**
1. Always perform checksum validation on the server side.
2. If the computed checksum does not match the one received, reject the response and log it for further investigation.
3. Do not hard code the **RESPONSEKEY** in client-side code or expose it publicly.

10.Status Codes

RinggitPay utilizes a standardized set of transaction status codes to indicate the result of payment processing activities. These codes are returned in API responses and are essential for:

1. Determining transaction outcomes
2. Diagnosing errors
3. Reconciling payment records
4. Communicating accurate statuses to end-users

Merchants are expected to capture and store these codes for transaction tracking, audit logging, and troubleshooting purposes.

STATUS CODES	
Response Code	**Description**
IR10	Invalid Order Number
IR11	Invalid Currency
IR12	Invalid Amount
IR13	Invalid Checksum
IR14	Invalid Buyer Email
IR15	Invalid ReturnURL
IR18	Invalid Transaction Reference
IR20	Merchant Not Active
RP00	Transaction Approved or Success
RP01	Transaction Exception
RP02	Merchant app verification pending
RP09	Transaction Pending
RP45	Duplicate Seller Order Number
RP50	Transaction Rejected by Visa/Mastercard
RP80	Insufficient Fund
RP91	Transaction Failed
RP92	Transaction Failed at Channel
RP94	Transaction Amount is Lower Than Minimum
RP95	Transaction Amount is Higher Than Maximum
RP96	Transaction Cancelled by Customer
RP97	Session Timed Out
RP100	User Transaction Limit Exceeded

Integration Guidelines

1. Store status codes with each transaction in your database for audit and reconciliation.
2. Use status codes programmatically to trigger workflows such as refunds, retries, or alerts.
3. Map status codes to appropriate user-facing messages within your frontend or app for better UX.

Handling Pending Transactions (RP09)

Transactions with the status code RP09 (Pending) must not be treated as final. This status indicates that the transaction is still under processing by the acquiring bank or payment channel.

To determine the final outcome of such transactions, merchants are required to:

- ✓ Periodically perform a Transaction Enquiry API call to fetch the latest status.
- ✓ Update their records and notify users once the final result is received from RinggitPay.

Best Practices for Merchants

Validate all transaction responses against the list of supported status codes.

- ❖ Avoid hard-coding descriptions — always refer to the latest documentation to accommodate new or updated codes.
- ❖ Display clear and accurate messages to users based on the transaction status.
- ❖ Monitor and alert on abnormal patterns such as frequent RP91 or RP92 failures.
- ❖ Implement fallback logic for handling ambiguous or pending transaction states (RP09, RP01, etc.).

⚠ **Note:**

1. RinggitPay may introduce new status codes or revise existing ones periodically to reflect new features, channels, or compliance requirements.
2. Always refer to the latest version of the RinggitPay API documentation to maintain compatibility and accuracy in your implementation.

11.Email Notifications

RinggitPay provides automatic email notifications to both buyers and merchants as part of its transaction communication flow. These notifications serve as confirmations and records of transaction status and details.

11.1. Buyer Email Notifications

RinggitPay automatically sends transactional email notifications to buyers, provided that the buyer Email field is included in the payment request. These emails serve as official confirmations of the transaction.

Conditions:

- ❖ The buyerEmail parameter must be correctly provided in the initial payment request.
- ❖ The transaction must reach a final status (e.g., approved, failed, or cancelled).

Email Contents:

1. Merchant Name
2. Order No
3. Order Reference
4. Status
5. RinggitPay Reference
6. Payment Channel
7. Channel Reference

Activation:

- ✓ This feature is optional and must be explicitly requested.
- ✓ To activate buyer email notifications, please contact RinggitPay Careline at the support channel provided in your on-boarding package.

⚠ **Note:**

1. To ensure successful delivery, merchants must validate the accuracy of the email address before submission.
2. To enable this feature, Please contact RinggitPay Careline to activate buyer email notifications.
3. These email notifications provide buyers with an official record of their transaction. However, they are not a replacement for API-based confirmation in the merchant’s system.
4. Instruct buyers to check their spam/junk folder if they do not receive an email within a reasonable time frame.

11.2. Merchant Email Notifications

Merchants may also opt in to receive email notifications for each transaction processed through their account. These notifications can serve as real-time alerts for settlements, or other payment events.

Activation:

1. This feature is optional and must be explicitly requested.
2. To activate merchant email notifications, please contact RinggitPay Careline at the support channel provided in your on-boarding package.

Merchant Email Use Cases:

- ✓ Real-time alerts for successful and failed transactions
- ✓ settlement-related updates (if enabled)

⚠ **Note:**

1. All emails are sent from RinggitPay's official notification server.
2. Merchants are advised to whitelist RinggitPay’s email domain to prevent email filtering or loss.
3. Ensure the buyerEmail field is validated before sending the payment request.

12.Go Live Process

Before going live, merchants are required to complete and submit a UAT (User Acceptance Testing) sign-off checklist provided by RinggitPay. This ensures that all core integration points are tested and verified for production readiness.

12.1. Go Live Checklist Request

Upon successful completion of UAT testing:

1. Contact the RinggitPay team to request the official UAT Checklist.
2. Submit the completed checklist along with required logs, screenshots, and any other supporting documentation for review.
3. Refer to the RinggitPay UAT Sign-Off Document for detailed instructions and formatting requirements.

12.2. UAT Sign-Off Verification Items

Merchants must validate and document the following test cases:

UAT SIGN-OFF VERIFICATION ITEMS
1. Merchant Checkout Page
Verify that the checkout page displays: a. Correct currency (e.g., MYR) b. Amount in two decimal places (e.g., 100.00)
2. FPX Payment Screen
Screen shots required: a. FPX payment screen showing seller name, currency, and amount b. Successful transaction (amount < MYR 100.00): Select Retail → SBI Bank A c. Unsuccessful transaction (amount < MYR 100.00): Select Retail → SBI Bank B d. Simulator confirmation screens for both success and failure

e. If B2B FPX is applicable:

i. Two transactions showing **PENDING** status
ii. Email the RinggitPay Transaction Reference to the RinggitPay team to simulate APPROVAL and DECLINE
iii. Two transactions with FINAL status screen shots (for both approval and decline outcomes)

3. Credit/Debit Card Payment screen

Screen shots required:

a. Credit card payment screen showing seller name, currency, and amount

b. Successful transaction using valid test credit card details (refer to provided test data)

c. Unsuccessful transaction using invalid test credit card details

4. Receipt Screen

Screen shots required:

a. Screenshot of the final receipt screen with transaction/order Id, amount, transaction status, appropriate remarks (if failed)

5. Merchant Response Handling

Redirect URL / Return URL:

a. Screenshot showing the response returned to the merchant

API URL:

b. Log or screenshot showing the raw response received

⚠ **Note:** The Redirect and API response URLs must not include IP addresses or port numbers.

6. Transaction Enquiry

Screen shots required: a. Screenshot or log of a Transaction Enquiry request and its response.
7. Email Notification
Screen shots required: a. Screenshot of the email received at the buyer's email address (provided during test transaction).

12.3. Submission and Review

Once all items are completed:

1. Compile and submit the checklist with all required screen shots/logs to the RinggitPay team.
2. Await review and confirmation.
3. Upon approval, RinggitPay will initiate the Go Live process and provide access to the Production Environment.

13. Notes to the merchant

To ensure reliable and secure transaction handling, merchants are advised to follow the best practices and guidelines outlined below:

1. Acknowledge API Responses

Merchants must return a 200 OK HTTP status code as an acknowledgment when a response is received via the API URL (Server-to-Server callback). Failure to acknowledge may result in repeated callbacks or delayed status updates.

2. Missing Response Handling

If no response is received from the RinggitPay Gateway after a transaction attempt:

- Immediately initiate a Transaction Enquiry request using the available API to check the transaction status.
- Do not assume the transaction has failed or succeeded without verification.

3. Transaction Enquiry Handling

It is the merchant’s responsibility to:

- Correctly handle and process the response from a Transaction Enquiry
- Update the transaction status in your internal system accordingly
- Ensure accurate status reflection for user experience, reporting, and reconciliation

4. Enquiry Time Window

When sending a Transaction Enquiry:

- ❖ RinggitPay forwards the request to the corresponding payment channels (if applicable)
- ❖ The consolidated response is returned to the merchant via the Enquiry API

⚠ **Note:**

1. Most payment channels do not retain transaction records beyond a specific duration.
2. Merchants are strongly advised to limit enquiries to transactions within 21 days from the original transaction date to ensure reliable status retrieval.

5. System Time Synchronization

For security and checksum validation purposes:

- ❖ Ensure that the merchant’s request server/device maintains accurate time synchronization
- ❖ Use NTP (Network Time Protocol) services to sync system clocks regularly
- ❖ Time discrepancies may lead to request rejections or verification failures

*************************** **End of document** ***************************

Thank you for your time and attention in reviewing this document.

If you have any further inquiries or require assistance, please feel free to contact us anytime at **+60 179898253** or email us at careline@ringgitpay.com.

We’re here to help!