



Introduction to API Integration

Table of Contents

Definition of Terms	2
Introduction.....	3
Redirect	3
setTransaction (RPP)	4

Instant Payment Notification (RPP)	11
<i>getTransaction</i> Request	11
Transaction states (RPP)	16
Transaction result validation (RPP)	16
Enterprise API.....	16
Error Codes.....	17

Definition of Terms

Term	Definition
Handshake	<p>A permanent link created between a Merchant and PayU for an individual customer.</p> <p>A Handshake must exist before any transactions can be carried out for a customer via the API integration.</p>
MiGs	MasterCard Internet Gateway System

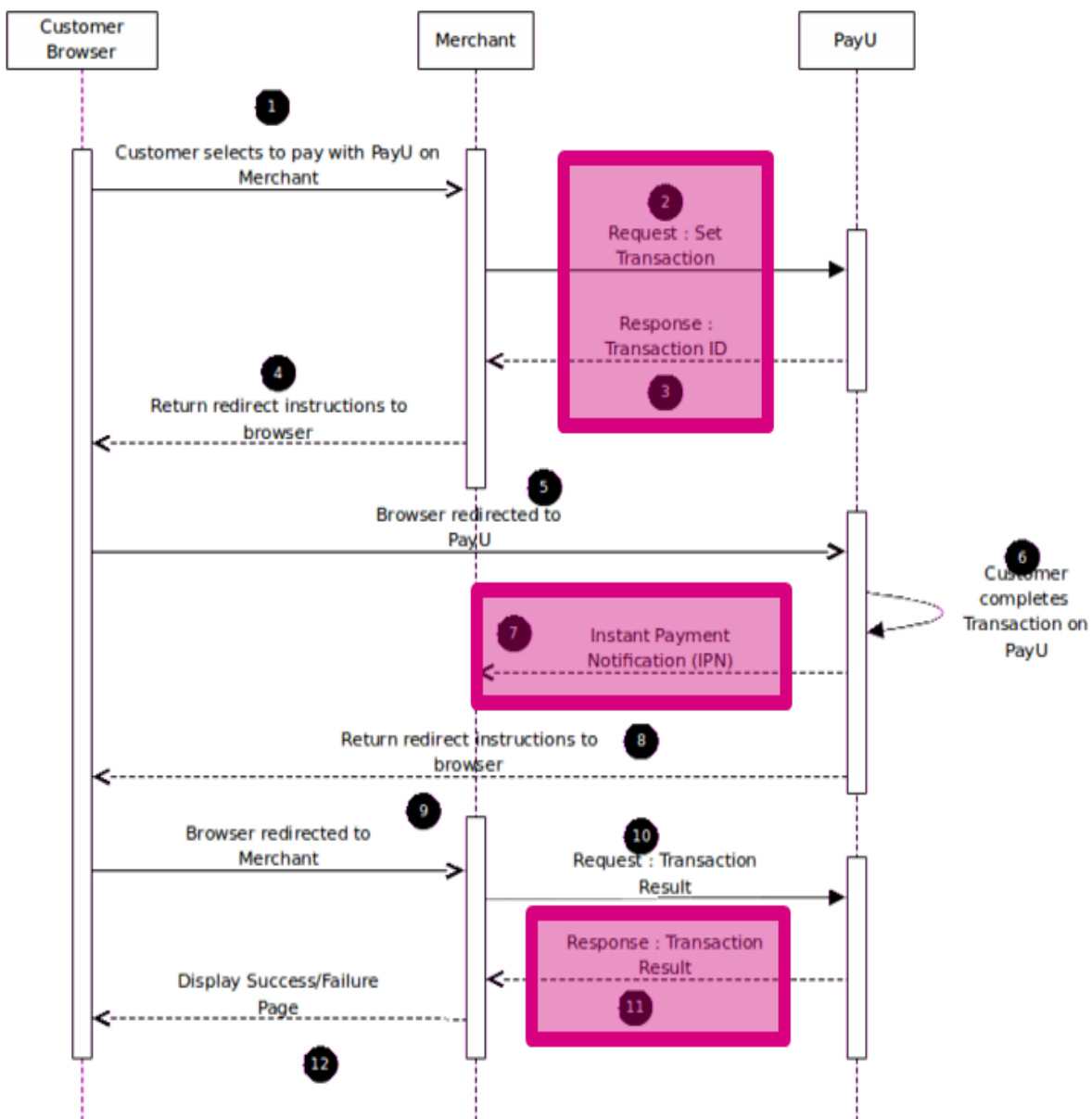
Introduction

The following document provides an introductory guide to integrating with the PayU Platform. The guide covers the basic services to fully process a single, straight payment transaction.

Redirect

The sequence set out in this document (as well as above), is only for a golden path integration and more test cases as well as recommendations are given in the test overview section. This summarized payment sequence can change and is dependent on the transaction type used in the [setTransaction](#) call.

A typical sequence might look like this.



- [SOAP headers \(RPP\)](#)
- [setTransaction \(RPP\)](#)
- [getTransaction \(RPP & 3DS API Redirects\)](#)
- [doTransaction \(RPP\)](#) ○ [Instant Payment Notification \(RPP\)](#)

1. Customer selects to pay with PayU on merchant website.

Merchant can integrate a PayU payment button which will initiate transaction with PayU.

2. Merchant starts the PayU process with a [setTransaction](#) request.

This is the first server-side web service call made by merchant webserver to PayU in order to set up the transaction on PayU.

3. PayU response: Transaction ID.

PayU will respond to a [setTransaction](#) request with a PayUReference which can then be used to redirect the customer to make a payment.

4. Merchant returns redirect [instructions](#) to customer browser.

The merchant has to return the redirect [instructions](#) to the browser in order to redirect the customer to PayU.

5. Customer browser redirected to PayU.

An HTTPS GET will be used to redirect the customer to PayU with the PayUReference. PayU will identify the transaction using the PayUReference and pick up the previous set of values passed in the [setTransaction](#) call.

6. Customer completes [transaction](#) on PayU.

The customer will proceed to [complete](#) the transaction on PayU by using one of the [payment methods](#) provided.

7. Instant [Payment](#) Notification ([IPN](#))

If the notificationURL is populated, an [instant payment notification](#) will be sent to the merchant to immediately inform them of the result of the transaction.

8. PayU returns redirect [instructions](#) to the customer's browser.

After the customer has confirmed their [payment](#), PayU will return redirect instructions to the customer's browser.

9. PayU redirects customer browser back to merchant website.

An HTTPS GET will redirect the customer back to the merchant's website.

10. Merchant requests Transaction Result ([getTransaction](#)).

The merchant issues a server-side web service call to PayU to retrieve the [transaction result](#) and determine [transaction outcome](#).

11. PayU response: [Transaction Result](#)

PayU will return all [payment](#) transaction information in the response. Success or failure and other details.

12. Merchant Displays Success/Failure page on website

The merchant should now redirect the user to the relevant success/failure/invoice page on their own site.

Details of APIs referenced in the above sequence:

setTransaction (RPP)

For the SOAP headers that have to accompany this message, see the section on RPP SOAP headers.

The setTransaction API call is used to setup/initiate transactions between the merchant and PayU using server to server requests. This has to be done before the customer can pay on the PayU payment page. Minimal detail about the transaction is passes, server-to-server, over HTTP POST or GET requests.

After doing a successful setTransaction request, a PayU reference number is issued in response. When the customer's browser redirects to the PayU payment page, this reference number is used to find the correct transaction and effect payment.

Transaction Types

The transaction type parameter in the setTransaction call is used to identify what type of transaction will be done.

You can read more about transaction type parameter values here, but two are typically used - PAYMENT or RESERVE. Each of these transaction types has a different effect on the transaction's authorisation and settles components, which are explained below.

SupportedPaymentMethods

The supportedpaymentmethod can be either CreditCard or Bank Transfer or both. To use the card option, specify CREDITCARD as the value for SupportedPaymentMethod and to use Bank Transfer, specify EFT_BANK_TRANSFER as the value for SupportedPaymentMethod. To have both payment method show on the payment page, specify both method separated by a comma.

RESERVE transaction type

This transaction type is typically used when a merchant issues shippable goods or gets stock from a supplier.

Scenario

Once a customer does a successful payment on the PayU interface, the funds are authorised. The merchant will try and source the goods from its supplier. Once the merchant has received the stock and is ready to ship, the merchant will issue a settle on the funds and ship the goods.

A typical payment sequence for the RESERVE transaction type:

1. Do a setTransaction SOAP call against PayU API on merchant's website
 2. Redirect to the PayU payment interface
 3. Customer submits and gets redirected back to merchant's website
 4. Do a getTransaction SOAP call against PayU API on merchant's website to get transaction payment details e.g. funds successfully reserved
- The following is typically done once the merchant is in possession of the shippable goods
5. Do a doTransaction SOAP call with a FINALIZE transaction type against PayU API on merchant's website
 6. Do a getTransaction SOAP call against PayU API on merchant's website to get transaction payment details e.g. funds successfully reserved

PAYMENT transaction type

The PAYMENT transaction type is typically used in transactions when stock is immediately available e.g. an mp3 download or online tickets.

Scenario

Once a customer does a successful payment on the PayU interface, the funds are authorized and settled. The merchant now gives the customer access to the goods e.g. mp3 download link

A typical payment sequence for the PAYMENT transaction type:

1. Do a setTransaction SOAP call against PayU API on merchant's website
2. Redirect to the PayU payment interface
3. Customer submits and gets redirected back to merchant's website
4. Do a getTransaction SOAP call against PayU API on merchant's website to get transaction payment details e.g. funds successfully reserved

Redirect Channel Types

The redirect channel parameter in the setTransaction call is used to identify what type of payment page should be displayed.

You can read more about redirect channel parameter values below, but two are typically used - web or responsive.

Web *redirect channel* will serve a normal html page with all the support payment methods.

Responsive *redirect channel* will serve a responsive page.

Only CREDITCARD payment method is supported on responsive at the moment.

SetTransaction: Request Parameters

Parameter Name	Description	Data type	Required field for request	
			RESERVE	PAYMENT
Api	Version of the API. <i>Current Version: 1.0</i> (SOAP: ONE_ZERO)	String	Y	Y
Safekey	PayU Merchant Identifier. <i>Provided to merchant upon</i>	String	Y	Y
TransactionType	The type of transaction being performed: <i>RESERVE</i> <i>PAYMENT</i>	String	Y	Y
MerchantReference	Unique merchant identifier for transaction.	String	Y	Y
SupportedPaymentMethods	Comma separated list of: CREDITCARD, EFT_BANK_TRANSFER	String	Y	Y
Secure3d	This will determine if the transaction goes through 3D Secure at the issuing bank. <i>Values: True/False</i>	String	N	N
NotificationURL	URL where the merchant would like to be notified of transaction result via IPN (Instant Payment Notification). (max length is 255 characters)	String	N	N
ReturnUrl	URL to return browser to after a customer has completed transaction. (max length is 255 characters)	String	Y	Y
CancelUrl	URL to return browser to if a customer cancels a transaction. (max length is 255 characters)	String	Y	Y

ShowBudget	Determined whether user should be given option to select budget for credit card. Values: True/False	String	N	N
redirectChannel	Valid values: web responsive mobile	String	N	N
FirstName	Customer first name	String	N	N
LastName	Customer last name	String	N	N
RegionalId	Regional identifier for customer such as Nigerian ID	String	N	N
Mobile	Customer Mobile Number in international format without +(eg 234 018912487)	String	N	N
Email	Customer Email Address	String	N	N
MerchantUserId	Unique Customer ID in merchant system	String	N	N
Description	Basket description that will show on PayU	String	N	N
AmountInCents	Total amount of the basket in cents. <i>Format as 1000 for NGN10.00</i>	String	Y	Y
CurrencyCode	International Currency code. <i>Example: NGN</i>	String	Y	Y
CustomFields				
Key	Identification for custom field	String	N	N
Value	Value for custom field	String	N	N

Y - Mandatory

N - Optional

NA - Not Allowed

setTransaction: Response Parameters

Response Parameters		
Parameter Name	Description	Values
MerchantReference	Merchant identifier for transaction.	String
Successful	Boolean value to determine if the transaction was successful or not.	String

PointOfFailure	Indicated on failed transactions where the point of failure was	String
ResultCode	Result code returned by PayU for transaction	String
ResultMessage	Result message relating to result code for transaction	String
DisplayMessage	Display friendly message for the customer	String
PayUReference	PayU unique reference number for transaction	String

setTransaction: SOAP request EXAMPLE

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://soap.api.controller.web.payjar.com/" xmlns:ns2="http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
```

```
  <SOAP-ENV:Header>
```

```
    <wsse:Security SOAP-ENV:mustUnderstand="1" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis200401-wss-wssecurity-secext-1.0.xsd">
```

```
      <wsse:UsernameToken wsu:Id="UsernameToken-9"
xmlns:wsu="http://docs.oasisopen.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
```

```
        <wsse:Username>Staging Enterprise With Fraud Integration Store 1</wsse:Username>
```

```
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-tokenprofile-1.0#PasswordText">xoV3PFor</wsse:Password>
```

```
      </wsse:UsernameToken>
```

```
    </wsse:Security>
```

```
  </SOAP-ENV:Header>
```

```
  <SOAP-ENV:Body>
```

```
    <ns1:setTransaction>
```

```
      <Api>ONE_ZERO</Api>
```

```
      <Safekey>{CF86C6D5-016C-4E98-9E4F-0F4FE3A0C1BA}</Safekey>
```

```
      <TransactionType>PAYMENT</TransactionType>
```

```
      <AdditionalInformation>
```

```
        <cancelUrl>http://qa.payu.co.za/integration-qa/internal-tools/demos/developer/payu-redirect-paymentpage/cancel-page.php</cancelUrl>
```

```

<demoMode>>false</demoMode>

<merchantReference>mercRef_1395758213</merchantReference>

<notificationUrl>http://qa.payu.co.za/integration-qa/internal-tools/demos/developer/payu-
redirectpayment-page/notification-page.php</notificationUrl>

<redirectChannel>web</redirectChannel>

<returnUrl>http://qa.payu.co.za/integration-qa/internal-tools/demos/developer/payu-redirect-
paymentpage/send-getTransaction-via-soap.php</returnUrl>

<secure3d>>true</secure3d>

<supportedPaymentMethods>CREDITCARD,EFT_BANK_TRANSFER</supportedPaymentMethods>

</AdditionalInformation>

<Customer>

<email>killer@bean.com</email>

<firstName>firstName_1395758213</firstName>

<ip>196.28.165.93</ip>

<lastName>lastName_1395758213</lastName>

<merchantUserId>merchantUserId_1395758213</merchantUserId>

<mobile>2782777777</mobile>

<regionalId>regionalId_1395758213</regionalId>

</Customer>

<Basket>

<amountInCents>6707</amountInCents>

<currencyCode>NGN</currencyCode>

<description>basketDesc_1395758213</description>

</Basket>

</ns1:setTransaction>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

```

setTransaction: SOAP response example

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Body>

    <ns2:setTransactionResponse xmlns:ns2="http://soap.api.controller.web.payjar.com/">

      <return>

        <merchantReference>mercRef_1395758213</merchantReference>

        <payUReference>150238913484</payUReference>

        <successful>true</successful>

      </return>

    </ns2:setTransactionResponse>

  </soap:Body>

</soap:Envelope>
```

+++++

Instant Payment Notification (RPP)

Instant Payment Notifications (IPNs) will be fired when the merchant has provided a *NotificationURL* in the SetTransaction or DoTransaction API call under the following conditions:

1. When a payment on the PayU redirect either fails or succeeds.
2. When a user session times out on the PayU redirect with no chance for the user to finish payment.
3. When a transaction, pending review for fraud, is either approved or rejected by case managers. In the case of approval the IPN will fire after the payment has been attempted and will return that result.

+++++

getTransaction (RPP & 3DS API Redirects)

getTransaction Request

In order to confirm transaction status a transaction lookup is provided by PayU. This transaction lookup is compulsory on any redirect integration and optional on API integration but still recommended to return full Secure 3D information where cards are not enrolled and no redirect happened.

getTransaction: Request Parameters

Parameter Name	Description	Data Type	Mandatory
Api	Version of the API. <i>Current Version: 1.0</i> (SOAP: ONE_ZERO)	String	Y
Safekey	PayU Merchant Identifier. <i>Provided to merchant upon integration</i>	String	Y
AdditionalInformation/PayUReference	PayU unique reference number for transaction	String	See note below *
AdditionalInformation/MerchantReference	MerchantReference supplied for transaction	String	See note below *

Icon

* - Either **PayUReference** or **MerchantReference** should be supplied for the call. **If both are supplied the call will fail**

getTransaction: SOAP request example


See getTransaction Request and Response SOAP API examples.

getTransaction: Response

The response will indicate the transaction state at the time of the request. Should the lookup be done while the transaction is still in progress appropriate responses needs to be interpreted and lookups should be repeated until transaction is in a final state.

getTransaction: Response Parameters

Parameter Name	Description	Data Type
basket/amountInCents	Total amount of the basket in cents. <i>Format as 1000 for NGN10.00</i>	String
basket/currencyCode	International Currency code. <i>Example: NGN</i>	String
basket/description	Basket description that was shown on PayU	String
customFields/key	Identification for custom field. E.g. Secure3D	String
customFields/value	Value for custom field. Multiple values can be included per key. Will be comma separated.	String

customFields/value/ IkpTransactionId	Card enrollment lookup transaction ID issued by 3rd party authentication server	String
customFields/value/ IkpErrorNo	Card enrollment lookup error number if issued by 3rd party authentication server. "0" means successful message flow.	String
customFields/value/ IkpErrorDescription	Card enrollment lookup error description if IkpErrorNo is not equal to "0".	String
customFields/value/ IkpEnrolled	<p>Card enrollment status, possible values "Y", "N", "U"</p> <ul style="list-style-type: none"> "Y" = Card enrolled for Secure 3D and will be redirected for cardholder authentication. "N" = Card not enrolled for Secure 3D and processing conditional as per merchant risk profile configured at the time of transaction. "U" = Card enrollment status unknown. Processing conditional based on PASA guidelines and merchant risk profile configured at the time of transaction 	String
customFields/value/ IkpEciFlag	Card enrollment lookup ECI value. Future use. Possible values that will be returned "1", "2", "5", "6", "7".	String
customFields/value/ authSend	If transaction is subject to Secure 3D authentication and redirected to authentication server from PayU redirect pages. API integrations will have HTML redirect code returned in the doTransaction call and a value of "true" means that redirect code was provided but not an indication  that redirect happened from the merchant checkout page.	String
customFields/value/ authErrorNo	Cardholder authentication error number if issued by 3rd party authentication server. "0" means successful message flow.	String
customFields/value/ authErrorDescription	Card authentication error description if authErrorNo is not equal to "0".	String

customFields/value/ authCavv	Card authentication CAVV value returned from 3rd party authentication server.	String
customFields/value/ authXid	Card authentication transaction ID issued by 3rd party authentication server	String
customFields/value/ authEciFlag	Card authentication ECI value. Possible values that will be returned "1", "2", "5", "6", "7".	String
customFields/value/ authPAREsStatus	<p>Card authentication, possible values "Y", "N", "U", "A"</p> <ul style="list-style-type: none"> "Y" = Authentication successful. Transaction considered fully authenticated. "N" = Authentication failed. Transaction will not be processed. Same as PIN failure on physical POS or ATM. "U" = Authentication status unknown. "A" = Authentication attempted. 	String
* Please refer to Secure 3D guidelines for more information plus charge back risk profiles..		
displayMessage	Customer friendly message to display in a browser to the customer	String
merchantReference	Merchant identifier for transaction.	String
payUReference	PayU unique reference number for transaction	String
paymentMethodsUsed	Subfields can differ per payment method. To highlight Secure 3D credit card as payment method illustrated here.	String
paymentMethodsUsed/amountInCents	Total amount of the basket in cents. <i>Format as 1000 for NGN10.00</i>	String
paymentMethodsUsed/cardExpiry	If Credit Card used, this will return card expiry date.	String
paymentMethodsUsed/cardNumber	If Credit Card used, this will return a masked Credit Card number	String

paymentMethodsUsed/gatewayReference	If external gateway is used the gateway unique reference referenced here.	String
paymentMethodsUsed/information	Payment method information (example: Loyalty program name), if credit card then the issuer schema. E.G. VISA, MasterCard, AMEX, Diners.	String
paymentMethodsUsed/nameOnCard	If Credit Card used, this will return Card holder name of the card	String
resultCode	Result code returned by PayU for transaction	String
resultMessage	Result message relating to result code for transaction	String
successful	Boolean value to determine if the message was successful or not. If the transaction failed the value will be "false" but a value of "true" does not indicate successful financial transaction. Refer to resultCode & resultMessage plus transactionState for transaction state.	
transactionState	The transaction state at the time of lookup. <ul style="list-style-type: none"> • <i>NEW</i> • <i>PROCESSING</i> • <i>SUCCESSFUL</i> • <i>FAILED</i> 	String
transactionType	The type of transaction performed. <ul style="list-style-type: none"> • <i>RESERVE</i> • <i>RESERVE_CANCEL</i> • <i>FINALIZE</i> • <i>CREDIT</i> • <i>PAYMENT</i> 	String
PointOfFailure	Indicated on failed transactions where the point of failure was, blank if successful transaction	String

Transaction states (RPP)

A transaction in PayU can be in one of 4 states: NEW, PROCESSING, SUCCESSFUL and FAILED. SUCCESSFUL and FAILED are both final states that cannot be changed. Once a setTransaction has been completed the transaction will be in a NEW state. When the user has redirected to PayU to complete the transaction the state is changed to PROCESSING. After the payment or stage has been completed the state will change to SUCCESSFUL.

In the case of a doTransaction for a FINALIZE, the transaction will go almost immediately from the NEW state to PROCESSING when PayU go off to the external payment gateway to perform the payment. Upon receiving the response from the external payment gateway the state is changed to SUCCESSFUL or FAILED.

+++++

Transaction result validation (RPP)

Transaction result validation

Interpretation of API responses to validate the result of a transaction should ideally be done on (but not limited to) a combination of the following fields:

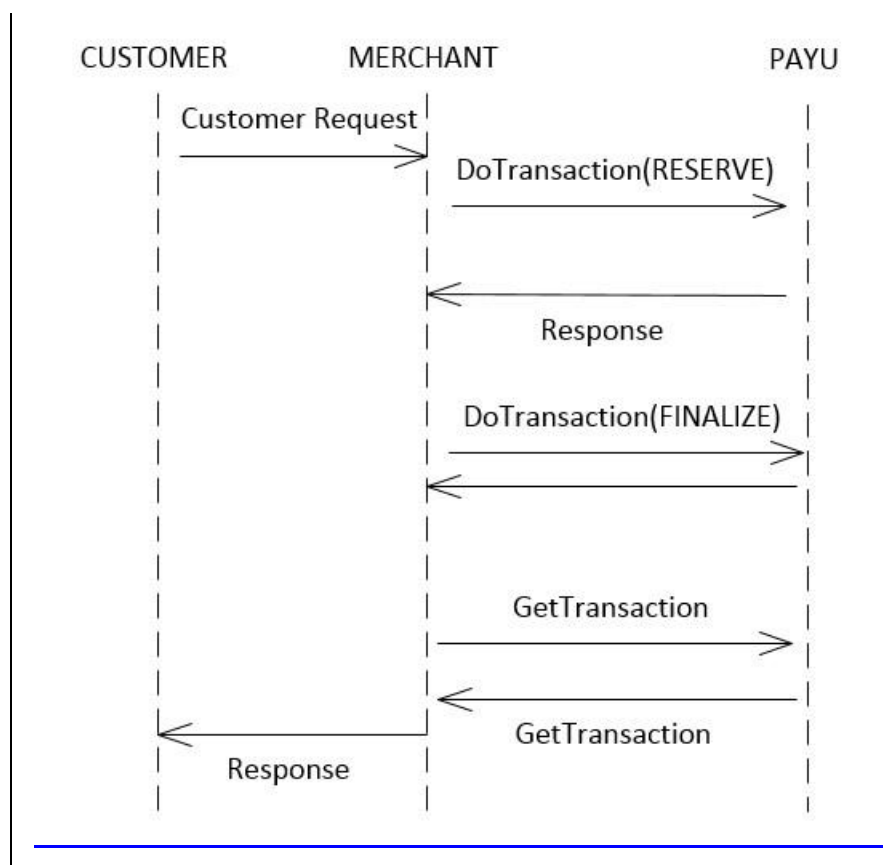
- resultCode
- successful
- transactionState
- transactionType

+++++

Enterprise API

Enterprise APIs are used where a merchant wants the customers to enter or store card details on their system and only want PayU to process the payment without the customer having to redirect to PayU.

Below is a typical flow for an enterprise API approach:



+++++

Error Codes

Point of Failure	Result Code	Result Message
PAYU	P001	Invalid Safekey
	P003	General system error
	P004	IP not allowed
	P005	Invalid PayUReference
	P006	Invalid MerchantReference
	P007	Invalid TransactionType
	P011	PayU request timeout
	P014	Security error
	P015	User cancelled the transaction
	P018	Staged transaction has expired
	P019	Transaction amount must be equal to less than original transaction amount
	P022	Transaction is not in the correct state - last transaction: <TransactionType> state: <TransactionState>

	P023	Transaction in progress
	P025	Lookup Transaction timeout
	P026	Lookup Transaction pm id invalid
	P027	Customer wallet not found
	P028	Requested transaction fails security rules
	P029	User failed to preapprove payments to merchant
	P030	User cancelled
	P031	Registration failed - mobile number already exists
	P032	Registration failed - email already exists
	P033	Registration failed - NationalID/Passport already exists
	P034	Login failed - invalid credentials
	P035	Login failed - wallet locked
	P036	Validation failed - [fieldname]
	P037	Merchant store deactivated
	P038	New PIN does not match format requirements
	P039	New Password does not match format requirements
	P040	Add credit card failed - credit card already exists
	P041	Invalid MerchantID
	P042	Decryption failed
	P043	Invalid VASID
	P044	Rejected for fraud
	P045	Rejected - Fraud system is unavailable
	P046	Payment failed
	P047	Payment method is not configured or allowed
	P048	Payment pending fraud case management
	P049	Payment failed following case management
	P050	Multi-tender not supported with provided payment methods
	P051	Basket amount does not match sum of payment methods
	P052	Wallet topup failed
MIGS	0	Transaction Successful
	1	Transaction could not be processed
	2	Transaction Declined - Contact Issuing Bank
	3	Transaction Declined- No reply from Bank
	4	Transaction Declined - Expired Card
	5	Transaction Declined - Insufficient credit
	6	Transaction Declined - Bank system error
	7	Payment Server Processing Error
	8	Transaction Declined - Transaction Type Not Supported
	9	Bank Declined Transaction (Do not contact Bank)

	A	Transaction Aborted
	B	Transaction Blocked
	C	Transaction Cancelled
	D	Deferred Transaction
	E	Transaction Declined - Refer to card issuer
	F	3D Secure Authentication Failed
	I	Card Security Code Failed
	L	Shopping Transaction Locked
	N	Cardholder is not enrolled in 3D Secure
	P	Transaction is Pending
	R	Retry Limits Exceeded, Transaction Not Processed
	T	Address Verification Failed
	U	Card Security Code Failed
	V	Address Verification and Card Security Code Failed