

KineticPay Doc ^{V1.1.1}

Simple API document for KineticPay

Last Updated on 1 September 2021

Direct Payment

This will direct payers straight from your system to the selected payment gateway (FPX) seamlessly. This is useful for developers who want to customize the full payment experience on their own website or mobile app and direct the payers to their Internet Banking login page.

Direct Payment Flow

1. Merchant send a payment request to an API.
2. Payer directly to selected bank portal.
3. After payment is done, payer will directly to merchant site.
4. In a meantime, callback data will send to the merchant site. Please specify Callback Endpoint at your API request.

API Endpoint URLs

<https://manage.kineticpay.my/payment/create>

Requirement to use the API

Kindly get the following credentials ready in order to start using this feature.

1. Merchant Key

You can retrieve these information from the Collection page, you will need to login to Dashboard.

Kindly ensure the payment request you send to either of the mentioned endpoints comply to the following format using **POST** method. You can set fields with **OPTIONAL** tag to exclude it entirely if it is not required.

Required Arguments

Parameter	Description
merchant_key	Your Merchant Key. Accept alphabets and numbers. Value is pre-defined by system. Do not modify this value to avoid error. TYPE: STRING EXAMPLE: 3ae289227944c922ff33f2d33408645c
invoice	Unique invoice number. Accept only numbers. Length: Max 10 characters. TYPE: INTEGER EXAMPLE: 10001

amount	Final amount to be paid by buyer. Accept ONLY numbers. Must be in 2 decimal places (d.p) format, eg: 192.00 TYPE: FLOAT EXAMPLE: 250.90
description	Purpose of payment and any data related to payment. Accept ONLY alphabets and numbers. Exclamation mark (!) is not accepted. Length: Max 1000 characters. TYPE: STRING EXAMPLE: Payment for Order No 10001, Buyer Name Ahmad Maslan, Email maslan@kineticpay.my, Phone No. +60199800010
bank	Bank code for FPX transaction. You can retrieve all list of bank code from Bank List from Resources at the end of this document. TYPE: STRING EXAMPLE: BCBB0235
callback_success	Callback URL for your front-end successful transaction process. Accepts only valid URL (starts with HTTP / HTTPS) TYPE: STRING EXAMPLE: HTTPS://WWW.EXAMPLE.COM/CALLBACK_SUCCESS
callback_error	Callback URL for your front-end failed transaction process. Accepts only valid URL (starts with HTTP / HTTPS) TYPE: STRING EXAMPLE:HTTPS://WWW.EXAMPLE.COM/CALLBACK_FAILURE

Optional Arguments

Parameter	Description
callback_status	Callback URL for your back-end process. Accepts only valid URL (starts with HTTP / HTTPS). Host name must be the same as the URL or Whitelist Domain specified in Collection page. Send through HTTPS header. TYPE: STRING EXAMPLE: HTTPS://WWW.EXAMPLE.COM/CALLBACK

Code Example

```
<?php
// Get ID from user
$bankid = "BCBB0235";

// This is merchant_key get from Collection page
$secretkey = "3ae289227944c922ff33f2d33408645c";

// This variable should be generated or populated from your system
$name = "Ahmad Maslan";
$phone = "+60199800010";
$email = "maslan@kineticpay.my";
$order_id = 10001;
$amount = 250.90;
```

```

$description = "Payment for Order No " . $order_id . ", Buyer Name " . $name .
", Email " . $email . ", Phone No. " . $phone;

$successURL = "https://www.example.com/success";
$failURL = "https://www.example.com/fail";
$checkStatURL = "https://www.example.com/update?orderid=" . $order_id;

$data = [
    'merchant_key' => $secretkey,
    'invoice' => $order_id,
    'amount' => $amount,
    'description' => $description,
    'bank' => $bankid,
    'callback_success' => $successURL,
    'callback_error' => $failURL,
    'callback_status' => $checkStatURL
];

// API Endpoint URL
$url = "https://manage.kineticpay.my/payment/create";

$ch = curl_init( $url );

// Setup request to send POST request.
$payload = json_encode( $data );
curl_setopt( $ch, CURLOPT_POSTFIELDS, $payload );

curl_setopt( $ch, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
$result = curl_exec($ch);
curl_close($ch);

echo json_decode($result, true)["html"];
?>

```

Common Response

Response will be in form of JSON. You need to echo the "html" from JSON to redirect user to bank page.

Example Response

Content type: application/json.

```

{
  "html": "\n\n<form
action=\"https://www.mepsfpx.com.my/FPXMain/seller2DReceiver.jsp\"
method=\"post\" id=\"ar\">\n\n <input type=\"hidden\" value=\"AR\"
name=\"fpx_msgType\">\n <input type=\"hidden\" value=\"01\"
name=\"fpx_msgToken\">\n <input type=\"hidden\" value=\"979999\"
name=\"id\">\n <input type=\"hidden\" value=\"EX00009999\"
name=\"fpx_sellerExId\">\n <input type=\"hidden\"
value=\"20299901219999-999\" name=\"fpx_sellerExOrderNo\">\n
<input type=\"hidden\" value=\"20219999911945\" name=\"fpx_sellerTxnTime\">\n
<input type=\"hidden\" value=\"10367-99999-099999-SmA\"
name=\"fpx_sellerOrderNo\">\n <input type=\"hidden\"
value=\"SE00039999\" name=\"fpx_sellerId\">\n <input
type=\"hidden\" value=\"01\" name=\"fpx_sellerBankCode\">\n
<input type=\"hidden\" value=\"MYR\" name=\"fpx_txnCurrency\">\n
<input type=\"hidden\" value=\"250.90\" name=\"fpx_txnAmount\">\n
<input type=\"hidden\" value=\"user@example.com\" name=\"fpx_buyerEmail\">\n
<input type=\"hidden\"
value=\"3999B966D5ABEA3CD6E4209B25589AB93135B5A53E1603D3FC456CAF15B6EC782FF132CF

```

```

9BEB89764BE7BEAD89DDFD15D84DB167D9999999906DC99BA0BB37B6E3AE6BD6E7874E5A0D0957DE8
A63B8648D647A5FFA94D2CE22EC085D607A5A2D5F948568D1C34E58FFF4FB6547A73ACC245E4F542
99659878E2973B345A2D09E5A9DACCD9F51AA3F699999999935F1A1E68B28C33EE8C5BE59120257
E8E8F68B956BA788B80F26817DF300FD3C2AB74C75653D13BCA7FFC99B2ABEC1B2BC211738C9C0E7
4799999999926AE8652A79502B6FE6367EB5D9EF16239F308EF999999999C9358CBC9623EFD8B4F49
DF4B8A0C03E4B8829DC00CE33B4A99CAA69DDB482\" name=\"fpx_checkSum\">\n
<input type=\"hidden\" value=\"SOMENAME\" name=\"fpx_buyerName\">\n
<input type=\"hidden\" value=\"BCBB0235\" name=\"fpx_buyerBankId\">\n
<input type=\"hidden\" value=\"\" name=\"fpx_buyerBankBranch\">\n
<input type=\"hidden\" value=\"\" name=\"fpx_buyerAccNo\">\n
<input type=\"hidden\" value=\"\" name=\"fpx_buyerId\">\n
<input type=\"hidden\" value=\"\" name=\"fpx_makerName\">\n
<input type=\"hidden\" value=\"\" name=\"fpx_buyerIban\">\n
<input type=\"hidden\" value=\"7.0\" name=\"fpx_version\">\n
<input type=\"hidden\" value=\"Merchant Site (API)\" name=\"fpx_productDesc\">\n
\n</form>\n<script
src=\"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js\"></
script>\n<script>\n document.getElementById('ar').submit();\n</script>\n\n"
}

```

Get Transactions

In the unlikely event that the payment network does not automatically notify the merchant regarding customer's payment status, use this API to manually query the transaction. This feature enables the merchant to retrieve their transaction status by sending a request direct to the bank or payment provider network. You may use this to send an enquiry request to FPX.

API Endpoint URL

<https://manage.kineticpay.my/payment/status>

Requirement to use the API

Kindly get the following credentials ready in order to start using this feature.

1. Merchant Key

You can retrieve these information from the Collection page, you will need to login to Dashboard.

Kindly ensure the request you send to either of the mentioned endpoints comply to the following format using GET method.

Required Arguments

Parameter	Description
merchant_key	Your Merchant Key Accept alphabets and numbers. Value is pre-defined by system. Do not modify this value to avoid error. TYPE: STRING EXAMPLE: 3ae289227944c922ff33f2d33408645c
invoice	Unique invoice number Accept only numbers. Length: Max 10 characters. TYPE: INTEGER

EXAMPLE: 10001

Code Example

```
<?php
// This is merchant_key get from Collection page
$secretkey = "3ae289227944c922ff33f2d33408645c";

// This variable should be generated or populated from your system
$order_id = 10001;

// API Endpoint URL
$url = "https://manage.kineticpay.my/payment/status";

$ch = curl_init( $url . '?merchant_key=' . $secretkey . '&invoice=' . (string)
$order_id );
curl_setopt( $ch, CURLOPT_HTTPHEADER, array('Content-Type:application/json'));
curl_setopt( $ch, CURLOPT_RETURNTRANSFER, true );
$result = curl_exec($ch);
curl_close($ch);

echo '<pre>' . var_export($result, true) . '</pre>';
?>
```

Common Response

Response will be in JSON. You can refer these variable for your system to update the payment status.

Variable Name	Description
id	Transaction ID on KineticPay system.
code	Payment Status Code. Example : 00
fpx_id	Transaction ID on FPX system.
invoice	Your Invoice Number. Example : 10001
merchant_key	Your Merchant Key

Example Response

Content type: application/json.

```
{
  "id": 176313,
  "code": "00", // Refer to Code References
  "fpx_id": "2199919999450999",
  "invoice": "10001",
  "merchant_key": "3ae289227944c922ff33f2d33408645c"
}
```

Errors Guides (Response Code)

The API will return error code if any of the variables sent is invalid.

Please refer to Response Code in Array in Resources at the end of document.

Resources

Bank List in Array PHP

```
$bankid = array(
'ABMB0212' => 'Alliance Bank Malaysia Berhad',
'ABB0233' => 'Affin Bank Berhad',
'AMBB0209' => 'AmBank (M) Berhad',
'BCBB0235' => 'CIMB Bank Berhad',
'BIMB0340' => 'Bank Islam Malaysia Berhad',
'BKRM0602' => 'Bank Kerjasama Rakyat Malaysia Berhad',
'BMMB0341' => 'Bank Muamalat (Malaysia) Berhad',
'BSN0601' => 'Bank Simpanan Nasional Berhad',
'CIT0219' => 'Citibank Berhad',
'HLB0224' => 'Hong Leong Bank Berhad',
'HSBC0223' => 'HSBC Bank Malaysia Berhad',
'KFH0346' => 'Kuwait Finance House',
'MB2U0227' => 'Maybank2u / Malayan Banking Berhad',,
'MBB0228' => 'Maybank2E / Malayan Banking Berhad E',
'OCBC0229' => 'OCBC Bank (Malaysia) Berhad',
'PBB0233' => 'Public Bank Berhad',,
'RHB0218' => 'RHB Bank Berhad',
'SCB0216' => 'Standard Chartered Bank (Malaysia) Berhad',
'UOB0226' => 'United Overseas Bank (Malaysia) Berhad'
);
```

Response Code in Array

```
$status = [
'B0' => 'Order list format error',
'B1' => 'Invalid seller ID',
'B2' => 'Seller is not allow to refund',
'B3' => 'Seller is not allow to do multiple refund',
'B4' => 'Requested refund amount exceed maximum allowable',
'B5' => 'Original transaction ID is not found',
'B6' => 'Original transaction ID status is still pending debit/credit',
'B7' => 'Original transaction ID status was not successful',
'B8' => 'Previous refund request still pending debit/credit',
'B9' => 'Requested refund amount below minimum allowable',
'C1' => 'Invalid refund transaction model',
'C2' => 'Invalid refund buyer bank',
'C3' => 'Invalid refund seller bank',
'C4' => 'Refund request fail due to no valid order list',
'C5' => 'Order list contain duplicate seller order number',
'1S' => 'Refund Successful Submitted',
'03' => 'Invalid Merchant',
'05' => 'Invalid Seller or Acquiring Bank Code',
'13' => 'Invalid Amount',
'00' => 'Transaction Approved',
'03' => 'Invalid Merchant',
'05' => 'Invalid Seller or Acquiring Bank Code',
'13' => 'Invalid Amount',
'00' => 'Transaction Approved',
'09' => 'Transaction Pending',
'12' => 'Invalid Transaction',
'14' => 'Invalid Buyer Account',
'20' => 'Invalid Response',
'31' => 'Invalid Bank',
```

'39' => 'No Credit Account',
'45' => 'Duplicate Seller Order Number',
'46' => 'Invalid Seller Exchange or Seller',
'47' => 'Invalid Currency',
'48' => 'Maximum Transaction Limit Exceeded RM30,000.00 for B2C',
'49' => 'Merchant Specific Limit Exceeded',
'50' => 'Invalid Seller for Merchant Specific Limit',
'51' => 'Insufficient Funds',
'53' => 'No Buyer Account Number',
'57' => 'Transaction Not Permitted',
'58' => 'Transaction To Merchant Not Permitted',
'70' => 'Invalid Serial Number',
'76' => 'Transaction Not Found',
'77' => 'Invalid Buyer Name or Buyer ID',
'78' => 'Decryption Failed',
'79' => 'Host Decline When Down',
'80' => 'Buyer Cancel Transaction',
'83' => 'Invalid Transaction Model',
'84' => 'Invalid Transaction Type',
'85' => 'Internal Error At Bank System',
'87' => 'Debit Failed Exception Handling',
'88' => 'Credit Failed Exception Handling',
'89' => 'Transaction Not Received Exception Handling',
'90' => 'Bank Internet Banking Unavailable',
'92' => 'Invalid Buyer Bank',
'96' => 'System Malfunction',
'97' => 'Transaction Pending',
'98' => 'MAC Error',
'99' => 'Pending Authorization (Applicable for B2B model)',
'BC' => 'Transaction Cancelled By Customer',
'DA' => 'Invalid Application Type',
'DB' => 'Invalid Email Format',
'DC' => 'Invalid Maximum Frequency',
'DD' => 'Invalid Frequency Mode',
'DE' => 'Invalid Expiry Date',
'DF' => 'Invalid e-Mandate',
'FE' => 'Internal Error',
'OE' => 'Transaction Rejected As Not In FPX Operating Hours',
'OF' => 'Transaction Timeout',
'SB' => 'Invalid Acquiring Bank Code',
'XA' => 'Invalid Source IP Address (Applicable for B2B2 model)',
'XB' => 'Invalid Seller Exchange IP',
'XC' => 'Seller Exchange Encryption Error',
'XE' => 'Invalid Message',
'XF' => 'Invalid Number of Orders',
'XI' => 'Invalid Seller Exchange',
'XM' => 'Invalid FPX Transaction Model',
'XN' => 'Transaction Rejected Due To Duplicate Seller Exchange Order Number',
'XO' => 'Duplicate Exchange Order Number',
'XS' => 'Seller Does Not Belong To Exchange',
'XT' => 'Invalid Transaction Type',
'XW' => 'Seller Exchange Date Difference Exceeded',
'1A' => 'Seller Buyer Session Timeout At Internet Banking Login Page',
'1B' => 'Buyer Failed To Provide The Necessary Info To Login To Internet Banking
Login Page',
'1C' => 'Buyer Choose Cancel At Login Page',
'1D' => 'Buyer Session Timeout At Account Selection Page',
'1E' => 'Buyer Failed To Provide The Necessary Info To Login To Internet Banking
Login Page',
'1F' => 'Buyer Choose Cancel At Account Selection Page',
'1G' => 'Buyer Session Timeout At TAC Request Page',
'1H' => 'Buyer Failed To Provide Necessary Info At TAC Request Page',
'1I' => 'Buyer Choose Cancel At TAC Request Page',
'1J' => 'Buyer Session Timeout At Confirmation Page',

```
'1K' => 'Buyer Failed To Provide Necessary Info At Confirmation Page',  
'1L' => 'Buyer Choose Cancel At Confirmation Page',  
'1M' => 'Internet Banking Session Timeout',  
'2A' => 'Transaction Amount Is Lower Than Minimum Limit RM1.00 for B2C',  
];
```

End of Document