Peach Payments Mobicred Integration Guide

- Overview
  - Experience
  - Brand Guidelines
  - Getting Started
- API Reference
  - Hosts
  - Security / Authentication
  - Payments API
    - Send an initial payment
    - Redirect the shopper
    - Get the payment status
    - Refunds
    - Interest Rate
      - Installment Widget
- Webhooks
  - Configuration
  - Integration Guide
    - Example: Payment
  - Encryption
  - Responding to Notifications
  - Decryption
- Result Codes
- HTTP Status Codes
- Rate Limits
- Testing
  - Test Accounts
  - View Test Transactions
  - Manage Test Webhooks
- Upcoming Features
- Revision History

Overview

mobicred is a simple and convenient credit facility that allows customers to safely shop online with participating retailers. When paying with mobicred, customers are able to make their purchase in full with the merchant and then pay their credit facility over time directly with mobicred. This flexibility gives merchants the ability to increase their online customer reach. For customers, they have access to credit that is similar to a credit card or digital wallet experience. Customers will need to apply and be approved for a mobicred account at mobicred.co.za prior to making a purchase using mobicred. In the checkout process, customers simply need to enter their mobicred account username and password on the merchant checkout page to purchase with mobicred payment.

This document will provide merchants with instructions on how to integrate mobicred as a payment option on checkout pages using the Peach Payments REST API.

Experience

Checking out with mobicred provides a simple and secure payment experience without customers needing to enter sensitive card details on a merchant's checkout page.

1) Merchants will display a mobicred payment option on their checkout page for customers to choose when ready to initiate payment of their purchase.
2) After selecting mobicred as a payment option on the merchant's secure checkout page, the customer will need to enter their mobicred username (email format) and password.
3) If the mobicred account is valid, the customer will be redirected to a Peach Payments hosted payment verification page to verify their purchase with a One Time Pin (OTP).
4) The customer will also receive a unique OTP on their mobicred registered mobile number. This OTP will be entered on the verification page. The customer will have thirty minutes to verify the purchase.
5) Once the OTP has been verified, the customer will be redirected back to the merchant's checkout page to complete the purchase with the merchant. (Merchants will receive an API GET request in the redirect to retrieve the verified transaction details).
6) If a valid webhook is enabled, the merchant will receive a response to their server endpoint with the details of the completed transaction.
7) The customer will receive a purchase confirmation from mobicred on their registered mobile device.

Brand Guidelines

Digital assets are available for merchants to include the mobicred payment method to their checkout pages. Marketing and customer messaging assets are also available. (assets coming soon)

Getting Started

To get started, merchant partners will need to add mobicred to their existing Peach Payments account. To get set up, reach out to the Peach Payments team, `support@peachpayments.com`. The Peach Payments team will initially provision the merchant with Test environment access to begin payments integration and testing.

Once a merchant has completed the mobicred integration and has successfully tested purchase transactions, the integration will be reviewed by the Peach Payments team. Email the Peach Payments team when the integration is ready for review `support@peachpayments.com`.

Note: Prior to going live, merchants will need to review and agree to the mobicred terms of service. This will be facilitated by the Peach Payments team.

API Reference

Hosts

Test: `https://testapi.peachpayments.com/v1`

Live: `https://api.peachpayments.com/v1`

Postman Collection:
`https://www.getpostman.com/collections/a311afa7556bbe9ec07f`

Security / Authentication

All requests must be made over SSL.

Test and Live environment credentials are generated through the Peach Payments Console. The `userid` and `password` will authorize each API request. The `entityid` associates the merchant channel entity with the mobicred integration. During account setup the Peach Payments team will assign the desired merchant channel and share the required `entityid` .

`SandboxConsole.peachpayments.com` (for Test environment)
`Console.peachpayments.com` (for Live environment)

Authentication Parameters	**Description**	**Format**	**Required**
authentication.userId	The merchant API userId. Required to authenticate a server-to-server request	AN32 [a-f0-9]{32}	Required
authentication.password	The password for the userId. Required to authenticate a server-to-server request	AN32 [a-zA-Z0-9]{8, 32}	Required
authentication.entityId	The entity for the request. By default this is the channel's ID. It can be the division, merchant or channel identifier.	AN32 [a-f0-9]{32}	Required

Payments API

To accept mobicred as a payment method, add the mobicred payment option to the checkout process on the merchant website. When the mobicred payment option is selected by the customer, the merchant will securely collect the customer's mobicred username and password. These credentials will then be sent over a HTTPS POST request to the `/v1/payments` endpoint to initiate the payment request. The response from this request will include a `redirect.url` which the merchant will redirect the shopper to verify and complete their purchase using the mobicred One Time Pin (OTP). Once the customer has verified and completed their purchase on the verification page, they will be redirected back to the merchant's `shopperResultUrl`. In the redirect, the merchant will receive a GET parameter `resourcePath` that can be used to retrieve the transaction results. Merchants will now be able to complete the customer purchase if successful or determine next steps depending on the result.

The Payments API supports transaction query and transaction refund.

Note: The mobicred password must be hashed on screen entry by the merchant. Both the username and password may not be stored.

Send an initial payment

Perform a server-to server POST request to the `/v1/payments` endpoint to create the payment `id` and to generate a `redirect.url` to redirect the customer to the Peach Payments hosted payment page to verify the purchase. This POST request will contain the authentication parameters, purchase parameters, the customer's mobicred account credentials, and any custom parameters that a merchant can optionally send.

Payment POST Parameters

Parameter	**Description**	**Format**	**Required**
authentication.userId	The userId for the entity. Required to authenticate a server-to-server request	AN32 [a-f0-9]{32}	Required
authentication.password	The password for the userId. Required to authenticate a server-to-server request	AN32 [a-zA-Z0-9]{8,32}	Required
authentication.entityId	The entity for the request. By default this is the channel's ID. It can be the division, merchant or channel identifier.	AN32 [a-f0-9]{32}	Required
merchantTransactionId	Merchant-provided reference number, should be unique for your transactions. Some receivers require this ID.	AN255 [\s\S]{8,255}	Conditional

amount	Indicates the amount of the payment request. The dot is used as decimal separator.	N10.N2 [0-9]{1,10}(\.[0-9]{2})?	Required
paymentBrand	The payment brand of the request. For mobicred this will be set to MOBICRED	AN32 [a-zA-Z0-9_] {1,32}	Required
paymentType	The payment type for the request. The following types are supported: DB, RV or RF.	A2	Required
virtualAccount.accountId	The customer's mobicred username	AN100 [\s\S]{1,100}	Required
virtualAccount.password	The customer's mobicred password.	AN100 [\s\S]{1,100}	Required
currency	The currency code of the payment request's amount.	A3 [A-Z]{3}	Required
shopperResultUrl	This URL will receive the result of an asynchronous payment. Must be sent URL encoded.	AN2048 [\s\S]{6,2048}	Conditional
merchantInvoiceId	Merchant-provided invoice number, should be unique for your transactions. This identifier is not sent onwards.	AN255 [\s\S]{8,255}	Optional

customParameters[*name*]	A name value pair used for sending custom information.	name: AN64
		[a-zA-Z0-9\._]{3,64}
		value: AN2048
		[\s\S]{0,2048}
			Optional

JSON Response

```
{
         "connectorTxID1": "29740001047",
         "customParameters": {},
         "paymentBrand": "MOBICRED",
         "currency": "ZAR",
         "descriptor": "Example Merchant",
         "merchantTransactionId": "8p829494638ef34e0146fcd847001eoe",
         "result": {
                 "code": "000.100.110",
                 "description": "Request successfully processed in 'Merchant in
Integrator Test Mode"
         },
         "redirect": {
                  "url": "https://testapi.ppay.io/v1/verify/",
                  "parameters": [
                       {"name": "customer",
                                  "value": "7064417334"},
                       {"name": "amount",
                                "value": "50.00"},
                       {"name": "connector",
                               "value": "MOBICRED"},
                       {"name": "currency",
                               "value": "ZAR"},
                       {"name": "transaction",
                              "value": "eefeb77a41df481peace7bf6f52e80a49”}
                 ],
                 "method": "POST"
        },
       "amount": "50.00",
       "shopperResultUrl": "http://merchant.example.com",
       "virtualAccount": {
        "accountId": "tarun@atlogys.com",
        "password": "XXXXXXXXXX"
    },
     "id": "eefeb77a41df481peace7bf6f52e80a49",
      "paymentType": "DB",
       "timestamp": "2018-07-30T12:18:19.453777Z",
        "resultDetails": {
              "ExtendedDescription": "Created OK - CellNo=073637411",
              "AcquirerResponse": "Pending"
        }
}
```

Payment Response Parameters

Parameter	**Description**	**Format**	**Required**
connectorTxID1	Mobicred transaction reference identifier.		Required

customParameters[*name*]	A name value pair used for sending custom information.	name: AN64
		[a-zA-Z0-9\._]{3,64}
		value: AN2048
		[\s\S]{0,2048}
			Optional

paymentBrand	The payment brand of the request. For Mobicred this will be set to `MOBICRED`	AN32 [a-zA-Z0-9_] {1,32}	Required
currency	The currency code of the payment request's amount.	A3 [A-Z]{3}	Required
descriptor	Currenlty responds with null.		
merchantTransactionId	Merchant-provided reference number, should be unique for your transactions. Some receivers require this ID. This identifier is often used for reconciliation.	AN255 [\s\S]{8,255}	Conditional
result	The unique code that indicates the result status of the request. See the Result Codes	AN11 [0-9\.]{2,11}	Required

	for more detailed information.		
redirect.url	URL the the shopper must be redirected to in order to proceed.	AN2048 [\s\S]{6,2048}	Conditional
redirect.parameters[n].name	List of parameter names for the redirect.url. The corresponding parameter value is the same parameter number ending with .value like described in the line below.	AN255 [\s\S]{1,255}	Conditional
redirect.parameters[n].value	The parameter values corresponding to the names as described above.	AN255 [\s\S]{1,255}	Conditional
amount	Indicates the amount of the payment request. The dot is used as decimal separator.	N10.N2 [0-9]{1,10}(\.[0-9]{2})?	Required
shopperResultUrl	This URL will receive the result of an asynchronous payment. Must be sent URL encoded.	AN2048 [\s\S]{6,2048}	Conditional
virtualAccount.accountId	The customer’s mobicred username	AN100 [\s\S]{1,100}	Required
virtualAccount.password	The masked password.	AN100 [\s\S]{1,100}	Required
id	The identifier of the checkout request that can be used to reference the payment later. You get this as the field id of a checkout's response and then should use it as the {id} part in subsequent steps.	AN48 [a-zA-Z0-9.\-]{32,48}	Required
paymentType	The payment type for the request. Types: `DB`, `RV` or `RF` are supported.	A2	Required
timestamp	The timestamp the response has generated.	date yyyy-MM-dd hh:mm:ss	Required

resultDetails	A container for name value pair used for enriching the response with mobicred response details.	name: AN64
		[a-zA-Z0-9\._]{3,64}
		value: AN2048
		[\s\S]{0,2048}
			Conditional

AcquirerResponse	Represents the acquirer original response code retrieved from the acquirer directly.	AN2048 [\s\S]{0,2048}	Conditional

Redirect the shopper

The next step is to redirect the customer to the Peach Payments hosted payment page to complete their purchase . To do this you must parse the `redirect` object in the payments response which contains the `redirect.url` and `redirect.parameters`. These parameter will be sent via HTTP POST to the `redirect.url`. During this time the transaction is pending

On the Peach Payments hosted payment page, the customer will enter their One-Time Pin to to complete and verify the purchase. **The customer will have 30 minutes to complete the transaction.**

Sample code for redirect

```
<form name="submitForm" action="https://testapi.ppay.io/v1/verify/ >

<input type="hidden"
        name="customer"
        value="07064417334">
<input type="hidden"
        name="connector"
        value="MOBICRED">
<input type="hidden"
       name="amount"
       value="50.00">
<input type="hidden"
       name="currency"
       value="ZAR">
<input type="hidden"
        name="transaction"
        value="eefeb77a41df481peace7bf6f52e80a49">
<noscript>
<input type='submit' name='submitButton'/>
</noscript>
</form>
```

Get the payment status

Once the customer has entered their OTP on the Peach Payments hosted payments page to verify the purchase, the customer is redirected to the `shopperResultUrl` along with a GET parameter in the `resourcePath`.

IMPORTANT: The baseUrl must end in a "/", e.g. "`https://testapi.ppay.io/`".
To get the status of the payment, you should make GET request to the baseUrl + `resourcePath`, including the authentication parameters.

This endpoint can also be queried to check the payment status during the time which the customer has been redirected to the payment verification page. Note that only two requests per minute is allowed.

```
{
   "paymentType": "DB",
   "amount": 50,
   "id": "eefeb77a41df481peace7bf6f52e80a49",
   "result": {
      "code": "000.000.000",
     "description": "Transaction succeeded"
   },
   "timestamp": "2018-08-07T07:54:57.050884Z",
   "customParameters": {},
   "resultDetails": {
      "ExtendedDescription": "Purchase Approved OK",
      "AcquirerResponse": "Approved",
     "ConnectorTxID1": "29740001047"
    },
  "paymentBrand": "MOBICRED",
  "currency": "ZAR",
  "merchantTransactionId": "8p829494638ef34e0146fcd847001eoe"
}
```

Refunds

A refund is performed against a previous payment, referencing its payment `id` by sending POST request over HTTPS to the `/v1/payments` endpoint. The POST request will contain the authentication parameters, a payment type, refund amount, refund and currency. The `paymentType` parameter will be set to `RF`.

Parameter	**Description**	**Format**	**Required**
authentication.userId	The userId for the entity. Required to authenticate a server-to-server request	AN32 [a-f0-9]{32}	Required
authentication.password	The password for the userId. Required to authenticate a server-to-server request	AN32 [a-zA-Z0-9]{8,32}	Required
authentication.entityId	The entity for the request. By default this is the channel's ID. It can be the division, merchant or channel identifier. Division is for requesting registrations only, merchant only in combination with channel dispatching, i.e. channel is the default	AN32 [a-f0-9]{32}	Required

	for sending payment transactions.		
amount	Indicates the amount of the payment request. The dot is used as decimal separator.	N10.N2 [0-9]{1,10}(\.[0-9]{2})?	Required
paymentType	The payment type for the request. You can send payment requests with one of the following types: RF.	A2	Required
currency	The currency code of the payment request's amount.	A3 [A-Z]{3}	Required

JSON Response

```
{
    "currency": "ZAR",
    "id": "eefeb77a41df481peace7bf6f52e80a49",
    "amount": "100.00",
    "result": {
        "description": "Transaction succeeded",
        "code": "000.000.000"
    },
    "timestamp": "2018-08-01T08:24:36.308493Z",
    "paymentType": "RF"
}
```

Interest Rate

Mobicred provides customers with a credit facility to pay for their purchase over installments with interest. To provide customers with information as to what interest rate they will be charged for their purchase, merchants are able to display the daily interest rate. This rate is updated once a day and should be used to retrieve the current day's interest rate.

The interest rate is available through an API GET request to the `https://testapi.ppay.io/rates/mobicred/interest-rate` endpoint with the merchant authentication parameters.

JSON Response

```
{
          "interestRate": "20.75"
}
```

Installment Widget

The Mobicred Installment Calculator Widget is a simple, yet configurable piece of code that can be added to any HTML-based web page where the need to display installment pricing is needed. The widget accepts multiple parameters allowing for easy configuration and customisation - from the text that displays on the widget to the fonts, colours and even layout.

Widget
`https://app.mobicredwidget.co.za/guides/widgets/instalment`

Webhooks

Peach Payments webhooks
`https://peachpayments.docs.oppwa.com/tutorials/webhooks`

Webhooks are HTTP callbacks that notify you of all events you subscribed to for an entity.

Configuration

Transaction webhooks can be set up and managed for both Test and Live environment, from within the Peach Payments Console.

`Console.peachpayments.com`
(coming soon)

Integration

The decrypted notification body contains the type of notification and its payload

```
{
    "payload": [content],
    "type": [notification_type]
}
```

Parameter	**Description**	**Format**	**Required**
type	Type of the notification (this is always set to PAYMENTS)	PAYMENTS	Required
payload	Content of the notification	JSON	Required

Example: Payment

```
{
'payload': {
   'amount': 50.0,
   'authentication': {'entityId': 'd4b10f6465db11e7b43e0245ffcc0db9'},
   'card': {
        'bin': None,
        'expiryMonth': '',
        'expiryYear': '',
        'holder': '',
        'last4Digits': ''},
 'currency': 'ZAR',
 'customParameters': {
        'SHOPPER_promoCode': 'AT052'},
 'id': 'eefeb77a41df481peace7bf6f52e80a49',
 'paymentBrand': None,
 'paymentType': 'DB',
 'presentationAmount': 50.0,
 'presentationCurrency': 'ZAR',
 'result': {
        'code': '000.000.000',
        'description': 'Transaction succeeded'},
 'timestamp': '2018-08-02 13:47:10.953154+00:00'},
'type': 'PAYMENTS'
}
```

Encryption

The content of the notification are encrypted to protect data from fraud attempts.

Parameter	**Description**
Encryption algorithm	AES
Key	[secret of listener] 64 character-long hexadecimal string in configuration (Provided by Peach Payments)
Key length	256 bits (32 bytes)
Block mode	GCM
Padding	None
Initialization Vector	In HTTP Header (X-Initialization-Vector)
Authentication Tag	In HTTP Header (X-Authentication-Tag)

An example of an encrypted notification:

HEADERS

X-Authentication-Tag: 9CDE76F1E157428D3CFF8CBCB5E5BFED

```
Accept-Encoding: gzip

Cf-Connecting-Ip: 213.131.241.51

X-Dynatrace: FW1;-1;-2107993456;7056;15;-2107993456;7056;3

X-Initialization-Vector: F0B3A54CC4043F28BE7CA326

Content-Type: application/json; charset=UTF-8

X-Request-Id: 67c70736-b4b6-4369-84a2-ecaf00e7f0ec

Content-Encoding: UTF-8

Cf-Ray: 3a86a01909170f75-FRA

Via: 1.1 vegur

Connect-Time: 0

Cf-Ipcountry: DE

User-Agent: Apache-HttpClient

Content-Length: 206

Total-Route-Time: 0

Cf-Visitor: {"scheme":"https"}

Connection: close

Host: requestb.in (This depends on your webhook url)
```

RAW BODY

{"encryptedBody":"12C148F1C9C1AEA110A5E9DAC8EABDF994BA6314387C2B3ABB29CF37A64”}

Responding to Notifications

When your service receives a webhook notification, it must return a 200 HTTP status code. Otherwise, the webhook service considers the notification delivery as failed, and will retry to send the notification later.

Protocol Details:
Protocol: HTTPS (HTTP is allowed in test systems only)
HTTP method: POST
Content Type: text(text/plain)

Decryption

The decryption tool/method is dependent on the programming language you are integrating with. To view your programming language's decryption example, please follow this link:
`https://peachpayments.docs.oppwa.com/tutorials/webhooks/decryption-example`

Result Codes

The result codes are part of the response body's JSON (field result) containing a code and a description explaining the code.

Result Code	Description
000.000.000	Transaction successfully processed in LIVE system
000.100.110	Transaction successfully processed in TEST system
100.396.101	Cancelled by user
100.396.104	Uncertain status - probably cancelled by user
600.200.500	Invalid payment data. You are not configured for this currency of sub type
200.300.404	Parameter in the incorrect format (The returned description will give details on which parameter is in the incorrect format)
600.200.400	Unsupported Payment Type (only DB is allowed as a payment type)
800.900.300	Invalid authentication information (Authentication password is incorrect)
800.900.201	Unknown channel (Entity Id in payment request is incorrect)

HTTP Status Codes

200	Successful
307	Temporary redirect
400	bad request. This might either point to e.g. invalid parameters or values sent. It's also returned if the payment failed e.g. because the acquirer declined.
403	incorrect authentication information, e.g. one of the authentication.* parameters is wrong - please check them or contact us for correct parameters
404	requested resource or endpoint is not found. I.e. endpoint/url doesn't exist. This can also be caused by typos like POST

	/v1/paymnets instead of payments or wrong IDs like GET /v1/payments/{id} where no payment with {id} exists.

Rate Limits

There is a limit two (2) requests per minute on the number of requests that a merchant can make to the purchase status API.

Testing

When a merchant is ready to test and integrate Mobicred, an email will need to be sent to support@peachpayments.com. The Peach Payments team will then setup the merchant with a Test environment for Mobicred.

The Peach Payments team will provide API authentication credentials for the Test environment to the merchant to begin the integration and testing.

Test Accounts

During account setup and Test environment provisioning the Peach Payments team will create test customer account for the merchant to use on the Test environment. Each test account is only able to support 3 transactions per hour before being locked out. If an account is locked out, contact our support team to unlock the account, support@peachpayments.com.

View Test Transactions

To view test transaction
Console.peachpayments.com
(coming soon)

Manage Test Webhooks

Merchants are able to set their transaction webhooks for test transaction from the Test environment within the Peach Payments Console.

Console.peachpayments.com
(coming soon)

Upcoming Features

- Peach Payments Merchant Console
- Asynchronous mobicred account creation

Revision History

Updated Nov 2, 2018
Updated May 15, 2019 - Change to API Endpoint Base URL’s