

Checkout by Peach Payments Integration Guide

[Overview](#)

[Experience](#)

[Getting Started](#)

[API Reference](#)

[Hosts](#)

[Security / Authentication](#)

[Signature \(HMAC SHA256\)](#)

[Uniqueness of Request](#)

[Whitelisted Domains](#)

[Checkout API](#)

[Checkout Payment Request](#)

[Payment Response Parameters](#)

[Get the Checkout Status](#)

[Refunds](#)

[Webhooks](#)

[Configuration](#)

[Responding to Notifications](#)

[Result Codes](#)

[HTTP Status Codes](#)

[Testing](#)

[Revision History](#)

Overview

Checkout by Peach Payments is a ready to use payment experience securely hosted and managed by Peach Payments. With Checkout, merchants have the ability to accept a full range of payment methods with only a single developer integration. Support for card, EFT and digital wallets come out the box and can be customized by the merchant through the Peach Payments Console. Checkout is fully PCI compliant and provides both the merchant and customer confidence in the security of web payments.

This document will provide developers with instructions on how to easily integrate Checkout into payment pages for web using the Peach Payments REST API.

Experience

Checkout streamlines the payment experience with a quick way for customers to quickly and confidently complete their purchase.

- 1) Merchant partners will display a payment button on their checkout page when a customer is ready to complete payment for their purchase.
- 2) The merchant will then redirect the customer to the Checkout payment page hosted by Peach Payments. In this redirect, the merchant will securely send a signed POST request with entity credentials along with the purchase details.
- 3) The Checkout page will display to the customer the purchase amount and all of the payment methods available to them to complete the purchase.
- 4) The customer selects their desired payment method to complete the purchase.
- 5) Once successful, the customer will be redirected back to the merchant page with a confirmation of their transaction. If canceled, the customer will be redirected back to provided cancel url.
- 6) If a valid webhook is enabled, the merchant will receive a response to their server endpoint with the details of the completed, canceled or expired transaction.

Getting Started

To get started, merchants will reach out to the Peach Payments team to initially provision test environment access to begin integration and payments testing.

Once a merchant has completed the Checkout integration and has successfully tested purchase transactions, the integration will be reviewed by the Peach Payments team for approval. Email our team when the Checkout integration is ready for review support@peachpayments.com. The review process can take up to two (2) days.

API Reference

Hosts

Live: <https://secure.peachpayments.com/checkout>

Test: <https://testsecure.peachpayments.com/checkout>

Security / Authentication

All requests must be made over SSL.

Signature (HMAC SHA256)

Payment requests to <https://secure.peachpayments.com/checkout> require a signature token to verify the integrity of the payment, ensuring that only the merchant sending the request is accepted. The signature uses the HMAC SHA256 algorithm, using all of the payment parameters in the request and the secret token as the key to generate the signature. This secret token is only known to the merchant and Peach Payments and is available in the Peach Payments Console.

To generate the signature, all of the payment parameters are required to be in alphabetical order, concatenated without any spaces and signed with the secret token as the key. The generated signature itself is added to the payment request for validation by Peach Payments.

The payment responses and transaction webhooks are signed by Peach Payments using the same signing method for the merchant to validate its authenticity.

Example of HMAC Generation:

Concatenated string:

amount2authentication.entityId8ac7a4ca68c22c4d0168c2caab2e0025currencyZARdefaultPaymentMethodCARDmerchantTransactionIdTest1234nonceJHGJSGHDSKJHGJDHGJHpaymentTypeDBshopperResultUrl<https://webhook.site/4e9b63bf-0d99-4d62-bd24-1d36ca866e1b>

Secret Token: 3fcd7cf22f55119eadbe02d14de18c0c

Computed HMAC:

311ed8e11e2da00d98c7479ca390a5396fe643e13629d850243dada877963afd

Verify your HMAC256 algorithm using the tool available [here](#)

Uniqueness of Request

The Checkout payment request additionally requires a unique `nonce` parameter to ensure the authenticity of each payment request. A `nonce` will be unique to each

request made to the [/checkout](#) url and must be managed by the merchant. This unique nonce will generate a unique signature. In doing so, the nonce helps to identify and prevent duplicate requests being made by the merchant or unauthorized third-party.

Whitelisted Domains

For added security, all url domains that initiate the API POST request to the Checkout experience require whitelisting. A list of urls can be whitelisted in the Peach Payments Console under the Checkout section.

Peach Payments Console Url: sandboxconsole.peachpayments.com

Checkout API

Add the Checkout experience to any merchant website by sending a POST request over HTTPS to the [/checkout](#) url to initiate the customer payment flow. The request will need to be signed and include the required parameters below in addition to optional custom parameters. A signature parameter will be included to validate the request. Once the customer has completed their purchase they will be redirected back to the merchant's [shopperResultUrl](#). In the redirect, the merchant will receive a response confirming the outcome of the payment flow.

Checkout Payment Request

Perform a POST request to the [/checkout](#) url. This will redirect the customer to secure.peachpayments.com/checkout to complete their payment. The POST request will contain the entityId, signature parameter, purchase parameters, any custom parameters that a merchant optionally sends.

Parameter	Description	Format	Required
authentication.entityId	The entity for the request. By default this is the channel's ID.	AN32 [a-f0-9]{32}	Required
signature	This signature for the request.	AN64 [a-f0-9]{64}	Required
merchantTransactionId	Merchant-provided reference number, should be unique for your transactions.	AN255 [\s\S]{8,255}	Required
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	A2	Required

	send payment requests with one of the following types: DB		
currency	The currency code of the payment request amount. ZAR and KES supported.	A3 [A-Z]{3}	Required
nonce	Unique value to represent each request.		Required
shopperResultUrl	The customer will be redirected to this URL after completion of Checkout. Valid URL that can be reached through a browser	AN2048 [\s\S]{6,2048}	Required
defaultPaymentMethod	The preferred payment method which will be the active payment method in the Checkout page at the point of redirecting	Possible values: CARD MOBICRED MASTERPASS EFTSECURE OZOW MPESA	Optional
merchantInvoiceId	Merchant-provided invoice number, should be unique for your transactions. This identifier is not sent onwards.	AN255 [\s\S]{8,255}	Optional
customParameters[<i>name</i>]	A name value pair used for sending custom information.	name: AN64 [a-zA-Z0-9\._-]{3,64} value: AN2048 [\s\S]{0,2048}	Optional
customer.merchantCustomerId	An identifier for this customer. Typically this is the ID that identifies the shopper in the shop's system		Optional
customer.givenName	The first name or given name of the customer. Required if you send in any other customer parameters, also required for some risk checks and payment providers. Will be truncated after 48 characters	AN [\s\S]	Optional
customer.surname	The last name or surname of the customer. Required if you send in any other customer parameters, also required for some risk checks and payment providers. Will be truncated after 48 characters	AN [\s\S]	Optional

customer.mobile	The customer's mobile number	AN25 [+0-9][0-9 \./-]{5,24}	Optional
customer.email	The customer's email address	AN128 [\s\S]{6,128}	Optional
customer.status	A status of the customer. Currently two options- NEW, EXISTING	A9 [\s\S]{1,255}	Optional
customer.birthDate	The birth day of the customer in the format yyyy-MM-dd, e.g. 1970-02-17	AN10 {19 20}([0-9]{2})-(0[1-9] 1[0-2])-(0[1-9] 1[0-9] 2[0-9] 3[0-1])	Optional
customer.ip	The customer's IP address.	AN255 [\s\S]{1,255}	Optional
customer.phone	The customer's phone number	AN25 [+0-9][0-9 \./-]{5,24}	Optional
billing.street1	The door number, floor, building number, building name, and/or street name of the billing address	AN100 [\s\S]{1,100}	Optional
billing.street2	The adjoining road or locality (if required) of the billing address	AN100 [\s\S]{1,100}	Optional
billing.city	The town, district or city of the billing address	AN80 [\s\S]{1,80}	Optional
billing.country	The country of the billing address (ISO 3166-1)	A2 [A-Z]{2}	Optional
billing.state	The county, state or region of the billing address	AN50 [a-zA-Z0-9]{1,50}	Optional
billing.postcode	The postal code or zip code of the billing address	AN30 [A-Za-z0-9]{1,30}	Optional
shipping.street1	The door number, floor, building number, building name, and/or street name of the shipping address	AN100 [\s\S]{1,100}	Optional
shipping.street2	The adjoining road or locality (if required) of the shipping address	AN100 [\s\S]{1,100}	Optional
shipping.city	The town, district or city of the shipping address	AN80 [a-zA-Z]{1,80}	Optional
shipping.postcode	The postal code or zip	AN30	Optional

	code of the shipping address	[A-Za-z0-9]{1,30}	
shipping.country	The country of the shipping address (ISO 3166-1)	A2 [A-Za-z]{2}	Optional
shpping.state	The county, state or region of the shipping address	AN50 [a-zA-Z0-9\,]{1,50}	Optional

Example POST

```
<form name="Checkout" action="https://secure.test.peachpayments.com/checkout"
```

```
    method="POST"
    accept-charset="utf-8">
```

```
    <input type="hidden"
        name="amount"
        value="2" />
```

```
    <input type="hidden"
        name="authentication.entityId"
        value="8ac7a4ca68c22c4d0168c2caab2e0025" />
```

```
    <input type="hidden"
        name="currency"
        value="ZAR" />
```

```
    <input type="hidden"
        name="defaultPaymentMethod"
        value="CARD"/>
```

```
    <input type="hidden"
        name="merchantTransactionId"
        value="Test1234" />
```

```
    <input type="hidden"
        name="nonce"
        value="JHGJSGHDSKJHGJDHGJH" />
```

```
    <input type="hidden"
        name="paymentType"
        value="DB" />
```

```
    <input type="hidden"
        name="shopperResultUrl"
value="https://webhook.site/4e9b63bf-0d99-4d62-bd24-1d36ca866e1b"/>
```

```
    <input type="hidden"
        name="signature"
value="311ed8e11e2da00d98c7479ca390a5396fe643e13629d850243dada877963afd"/>
```

```
    <input type="submit" value="Continue to Payment Method"/>
</form>
```

Payment Response Parameters

Parameters passed in the Checkout payment POST request will also be included in the response.

Parameter	Description	Format	Required
authentication.entityId	The entity for the request. By default this is the channel's ID.	AN32 [a-f0-9]{32}	Required
signature	This signature for the request.	AN64 [a-f0-9]{64}	Required
currency	The currency code of the payment request's amount.	A3 [A-Z]{3}	Required
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 for more detailed information.	AN11 [0-9\.]{2,11}	Required
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
checkoutId	The identifier of the checkout request that can be used to reference the payment request later.	AN48 [a-zA-Z0-9\.-]{32,48 }	Required
id	The unique identifier of a successful payment. Will be used for refunds.	AN32 [a-f0-9]{32}	Conditional
paymentType	The payment type for the request. Types: DB 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 Masterpass response details.	name: AN64 [a-zA-Z0-9\.-]{3,64} value: AN2048 [\s\S]{0,2048}	Conditional
customParameters[name]	A name value pair used for sending custom	name: AN64 [a-zA-Z0-9\.-]{3,64}	Optional

	information.	value: AN2048 [\s\S]{0,2048}	
--	--------------	---------------------------------	--

Redirect Response

After the user has completed payment, we will POST the response variables while redirecting the user to the `shopperResultUrl`

Structure of the POST redirect:

```
{
  "amount": "14.99",
  "checkoutId": "0fa125d002044484b0fbfb43c1c5c554",
  "currency": "ZAR",
  "id": "1d6c60ed0dfd4a6a9a26a13922b65766",
  "merchantInvoiceId": "5654-654645-65464",
  "merchantTransactionId": "UAT",
  "paymentBrand": "MOBICRED",
  "paymentType": "DB",
  "result.code": "000.100.110",
  "result.description": "Request successfully processed in 'Merchant in Integrator Test Mode",
  "resultDetails.AcquirerResponse": "Approved",
  "resultDetails.ConnectorTxID1": "20001694247",
  "resultDetails.ExtendedDescription": "Purchase Approved OK",
  "timestamp": "2019-01-25T08:27:46.916519Z",
  "signature": "49d3166063b4d881b50af0b4648c1244bfa9890a53ed6bce6d24b610777"
}
```

Important: All redirect requests are signed with a signature using the secret token as the key

Get the Checkout Status

A request can be made to obtain the status of a Checkout payment request by using the `checkoutId` as a reference to the payment. The `checkoutId` is provided in the webhook sent to the merchant once a Checkout payment is initiated, the status will be Pending. A merchant is also able to provide their `MerchantTransactionId` to query a transaction status, however note that if this identifier is not unique, only the latest Checkout payment status associated with that `MerchantTransactionId` will be returned. (If no `MerchantTransactionId` was provided in the initial Checkout payment request, an error response will be returned.

Payment Status Types

Pending	A Checkout payment has been initiated by the merchant and is awaiting customer completion. Payment can have a future state depending on the customer's action.
Successful	A Checkout payment has been completed and is successful. This is a final state of a payment attempt.
Cancelled	A Checkout payment has been cancelled by the end customer. This is a final state of a payment attempt.

Expired	A Checkout payment has timed-out by the end customer. The customer may not have completed their payment in the allotted time (30 minutes) or could have closed the payment window on their device. This is a final state of a payment attempt.
---------	--

Get Status Request Parameters

api.peachpayments.com/v1/checkout/status

testapi.peachpayments.com/v1/checkout/status

Important: Get status requests are signed with a signature using the secret token as the key.

authentication.entityId	The entity for the request. By default this is the channel's ID.	AN32 [a-f0-9]{32}	Required
signature	This signature for the request.	AN64 [a-f0-9]{64}	Required
id	The unique identifier of a successful payment that can be used to reference the payment at a later stage	AN32 [a-f0-9]{32}	Optional
checkoutId	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}	Optional
merchantTransactionId	Merchant-provided reference number, should be unique for your transactions. Some receivers require this ID. This identifier is often used for reconciliation.	AN48 [a-zA-Z0-9.\-]{32,48}	Conditional
signature	The signature for the request	AN64 [a-f0-9]{64}	Required

This endpoint can be queried to check the payment status during the time which the customer has been redirected to the Checkout payment page. There is a limit to two

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

Example of the GET Request:

```
curl -G https://testapi.peachpayments.com/v1/checkout/status \  
-d authentication.entityId=8ac7a4ca86c2c24d0168c2caab2e0025 \  
-d merchantTransactionId=Test1234 \  
-d signature=79b450ed255e306332f6325c99732cc84fca8cc2cf3fdc2bfd1f3e2465f55'
```

Important: All Get Status requests require a signature and all responses are signed with a signature using the secret token as the key to ensure authenticity.

Refunds

api.peachpayments.com/v1/checkout/refund
testapi.peachpayments.com/v1/checkout/refund

A refund can be performed against a successful payment, referencing the unique payment `id` provided in the Get Status API or successful payment webhook by sending POST request over HTTPS to the api.peachpayments.com/v1/checkout/refund 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.entityId	The entity for the request. By default this is the channel's ID.	AN32 [a-f0-9]{32}	Required
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: <code>RF</code> .	A2	Required
currency	The currency code of the payment request amount.	A3 [A-Z]{3}	Required
id	The unique identifier of the initial payment that is being refunded	AN32 [a-f0-9]{32}	Required
signature	The signature for the request	AN64 [a-f0-9]{64}	Required

Important: All refund requests require a signature and all responses are signed with a signature using the secret token as the key

Webhooks

Webhooks are HTTP callbacks that notify you of payment events from the Checkout flow. A webhook is sent to the merchant provided url for every state change of the Checkout payment request.

Created	A Checkout instance has been launched and the Checkout session has now been initiated.
Pending	A Checkout payment has been initiated by the merchant and is awaiting customer completion. Payment can have a future state depending on the customer's action.
Successful	A Checkout payment has been completed and is successful. This is a final state of a payment attempt.
Cancelled	A Checkout payment has been cancelled by the end customer. This is a final state of a payment attempt.
Expired	A Checkout payment has timed-out by the end customer. The customer may not have completed their payment in the allotted time (30 minutes) or could have closed the payment window on their device. This is a final state of a payment attempt.

Configuration

Transaction webhooks can be set up and managed for both and Live Test environments from within the Peach Payments Console. A valid url, responding with a 200, must be setup for url to register and save.

Webhook notifications are sent for all debit (DB) payments. In order to receive, refund webhooks merchants need to opt-in for refund notifications within the Peach Payments Console.

sandboxconsole.peachpayments.com

Important: All webhooks are signed with a signature using the secret token as the key

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 has failed, and will retry to send the notification later.

Example Debit (DB) Webhook

```
{
  "amount": "14.99",
  "checkoutId": "0fa125d002044484b0fbfb43c1c5c554",
  "currency": "ZAR",
  "id": "1d6c60ed0dfd4a6a9a26a13922b65766",
  "merchantInvoiceId": "5654-654645-65464",
  "merchantTransactionId": "UAT",
  "paymentBrand": "MOBICRED",
  "paymentType": "DB",
  "result.code": "000.100.110",
```

```

"result.description": "Request successfully processed in 'Merchant in Integrator
Test Mode",
"resultDetails.AcquirerResponse": "Approved",
"resultDetails.ConnectorTxID1": "20001694247",
"resultDetails.ExtendedDescription": "Purchase Approved OK",
"timestamp": "2019-01-25T08:27:46.916519Z",
"signature": "49d3166063b4d881b50af0b4648c1244bfa9890a53ed6bce6d24b610777"
}

```

Example Refund (RF) Webhook

```

{
"amount": "2.00",
"currency": "ZAR",
"descriptor": " UAT Checkout 8831.2726.5961",
"id": "8ac7a4a06b4f7618016b50fdaa4305fb",
"merchantTransactionId": "Test1234",
"paymentType": "RF",
"referencedId": "8ac7a4a16b4f6a06016b50fb930431e8",
"result.code": "000.100.110",
"result.description": "Request successfully processed in 'Merchant in Integrator
Test Mode'",
"timestamp": "2019-06-13 13:18:50+0000",
"signature": "49d3166063b4d881b50af0b4648c1244bfa9890a53ed6bce6d2386404b610777"
}

```

Result Codes

The result codes are part of the response containing a code and a description explaining the code.

Result Code	Description
000.200.100	Successfully created checkout (Indicates that the session is now open)
000.200.000	Transaction pending
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)
800.100.152	Transaction declined by authorization system
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/payments instead of payments or wrong IDs like GET /v1/payments/{id} where no payment with {id} exists.

Testing

When a merchant is ready to test and integrate Checkout, send an email to support@peachpayments.com who will then setup the merchant with a Test environment..

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

Revision History

Updated February 12, 2019

Updated April 9, 2019 - Support for default payment method parameter

Updated May 9, 2019 - Addition of POST Redirect

Updated July 23, 2019 - Addition of Refund (RF) webhooks