



BE PAYMENT READY

Moneris Checkout Integration Guide

Version: 1.0.0

Review Draft

Copyright © Moneris Solutions, 2020

All rights reserved. No part of this publication may be reproduced, stored in retrieval systems, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Moneris Solutions Corporation.

Table of Contents

| | |
|---|-----------|
| Getting Help | 4 |
| System and Skills Requirements | 5 |
| 1 About Moneris Checkout | 7 |
| 2 Building Your Moneris Checkout Integration | 8 |
| 2.1 Configuring Moneris Checkout in Merchant Resource Center | 8 |
| 2.1.1 Additional Features to Configure in the MRC | 8 |
| 2.2 Moneris Checkout Transaction Process Flow | 9 |
| 2.3 Preparing Your Client-Side Checkout Page | 9 |
| 2.4 Implementing Preload Server-to-Server Logic | 10 |
| 2.4.1 Preload Request | 10 |
| 2.4.1.1 Optional Preload Request Objects | 13 |
| Recurring Billing Object | 14 |
| Shopping Cart Object | 15 |
| Contact Details Object | 17 |
| Shipping Details Object | 17 |
| Billing Details Object | 18 |
| Shipping Rates Object | 19 |
| 2.4.1.2 Example Preload Request JSON | 20 |
| 2.4.2 Response to Preload Request | 23 |
| 2.4.2.1 Example Preload Response – Successful Preload | 23 |
| 2.4.2.2 Example Preload Response – Failed Preload | 24 |
| 2.5 Displaying the Moneris Checkout Page in the Browser | 24 |
| 2.6 Handling Callbacks | 24 |
| 2.6.1 Callback Types | 24 |
| 2.6.1.1 Callback Response Fields | 25 |
| 2.6.1.2 Page Loaded | 25 |
| 2.6.1.3 Cancel Transaction | 26 |
| 2.6.1.4 Error Event | 26 |
| 2.6.1.5 Payment Receipt | 27 |
| 2.6.1.6 Payment Complete | 27 |
| 2.7 Implementing Receipt Request Server-to-Server Logic | 28 |
| 2.7.1 Receipt Request | 28 |
| 2.7.1.1 Example Receipt Request JSON | 29 |
| 2.7.2 Response to Receipt Request | 29 |
| 2.7.2.1 Definition of Response Fields – Response to Receipt Request | 30 |
| 2.7.2.2 Example JSON Response to Receipt Request | 52 |
| 2.8 Terminating the Moneris Checkout Instance | 56 |
| 3 Additional Features in Moneris Checkout | 57 |
| 3.1 Tokenization of Credentials With Moneris Checkout | 57 |
| 3.2 Fraud Tools in Moneris Checkout | 57 |
| 3.2.1 About Fraud Tools in Moneris Checkout | 57 |
| 3.2.2 Kount as a Fraud Tool in Moneris Checkout | 58 |
| 3.2.3 Fraud Tools and Auto Decision-Making | 58 |
| 3.3 Window Size in Moneris Checkout | 58 |
| 4 Testing Your Moneris Checkout Integration | 59 |

| | |
|---|-----------|
| 5 Moving to Production with Moneris Checkout | 60 |
| 6 Reference | 61 |
| 6.1 Callback Response Codes – Moneris Checkout | 61 |
| 6.2 AVS Response Codes – Moneris Checkout | 61 |
| 6.3 CVD Response Codes – Moneris Checkout | 64 |
| 6.4 CAVV Result Codes | 65 |

Getting Help

Moneris has help for you at every stage of the integration process.

| Getting Started | During Development | Production |
|---|--|--|
| Contact our Client Integration Specialists: clientintegrations@moneris.com | If you are already working with an integration specialist and need technical development assistance, contact our eProducts Technical Consultants: 1-866-319-7450 eproducts@moneris.com | If your application is already live and you need production support, contact Moneris Customer Service: onlinepayments@moneris.com 1-866-319-7450 Available 24/7 |

For additional support resources, you can also make use of our community forums at <http://community.moneris.com/product-forums/>

System and Skills Requirements

In order to integrate with Moneris Checkout as a merchant, you must have:

- An e-commerce website with a back-end server

For development, you should have some understanding of the following:

- JavaScript
- JSON
- Server-side programming

1 About Moneris Checkout

Moneris Checkout gives e-commerce merchants a simple and secure way to process payments by integrating a Moneris-hosted payment module into the merchant checkout page.

2 Building Your Moneris Checkout Integration

- 2.1 Configuring Moneris Checkout in Merchant Resource Center
- 2.2 Moneris Checkout Transaction Process Flow
- 2.3 Preparing Your Client-Side Checkout Page
- 2.4 Implementing Preload Server-to-Server Logic
- 2.5 Displaying the Moneris Checkout Page in the Browser
- 2.6 Handling Callbacks
- 2.7 Implementing Receipt Request Server-to-Server Logic
- 2.8 Terminating the Moneris Checkout Instance

2.1 Configuring Moneris Checkout in Merchant Resource Center

The first step is to configure your Moneris Checkout page in the Moneris Merchant Resource Center (MRC).

In the initial stage of development, you create a test configuration in the testing MRC. Once the solution is ready to be deployed to production, you must create a new, separate configuration for the production environment in the production MRC.

The **checkout ID** is the key value that is generated after the configuration is completed and used within the Preload Request in order to identify the specific Moneris Checkout configuration.

To get the checkout ID and start configuring your page, do the following:

1. Log into the Merchant Resource Center at one of the following URLs (according to your stage of development)
Testing: <https://esqa.moneris.com/mpg>
Production: <https://www3.moneris.com/mpg>
2. In the Admin menu, select **Moneris Checkout Config**
3. Click the **Create Profile** button
4. Follow the on-screen steps to complete the configuration

For more information, see the Merchant Resource Center documentation available for download on the Moneris developer portal at:

developer.moneris.com

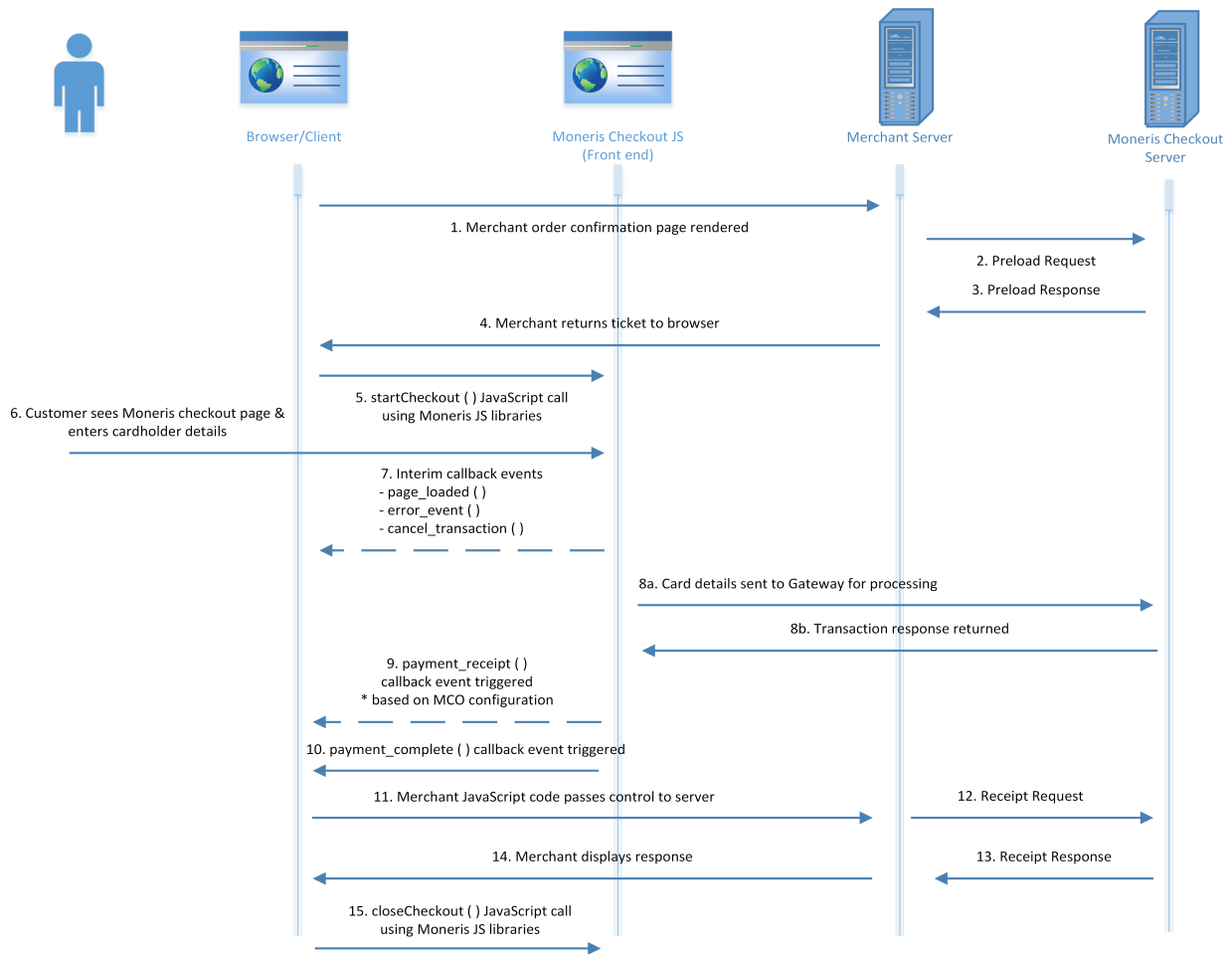
2.1.1 Additional Features to Configure in the MRC

There are other features of the Moneris Checkout page that you can enable using the configurator in the Merchant Resource Center. They include:

- Tokenization of credentials
- Fraud tool behaviour
- Window sizing

For more on configuring these features, see 3 Additional Features in Moneris Checkout.

2.2 Moneris Checkout Transaction Process Flow



2.3 Preparing Your Client-Side Checkout Page

In order to prepare your client-side checkout page for interacting with Moneris Checkout, you need to do a few tasks first:

1. Add a call to the Moneris Checkout JavaScript library in a `<script>` tag:

Testing:

```
<script src="https://gatewayt.moneris.com/chkt/js/chkt_v1.00.js"></script>
```

Production:

```
<script src="https://gateway.moneris.com/chkt/js/chkt_v1.00.js"></script>
```

2. Create a `<div>` in the HTML:

```
<div id="monerisCheckout"></div>
```

3. Instantiate the **monerisCheckout** object and set it up:

```
var myCheckout = new monerisCheckout();

myCheckout.setMode("qa");

myCheckout.setCheckoutDiv("monerisCheckout");
```

4. Set callbacks in JavaScript:

```
myCheckout.setCallback("page_loaded", myPageLoad);

myCheckout.setCallback("cancel_transaction", myCancelTransaction);

myCheckout.setCallback("error_event", myErrorEvent);

myCheckout.setCallback("payment_receipt", myPaymentReceipt);

myCheckout.setCallback("payment_complete", myPaymentComplete);
```

For more information about callbacks in Moneris Checkout, see 2.6 Handling Callbacks.

2.4 Implementing Preload Server-to-Server Logic

The Preload request is the means by which a Moneris Checkout instance is securely generated at transaction time. It involves a server-to-server post using the JSON format documented in 2.4.1 Preload Request.

The response to the Preload request returns a ticket number which uniquely identifies the instance and must be passed in the JavaScript `monerisCheckout.startCheckout (ticket #)` request in order to display the Moneris Checkout page in the browser.

In your server implementation, use the following Moneris Checkout URLs to post to, depending on the development stage:

Testing:

<https://gatewayt.moneris.com/chkt/request/request.php>

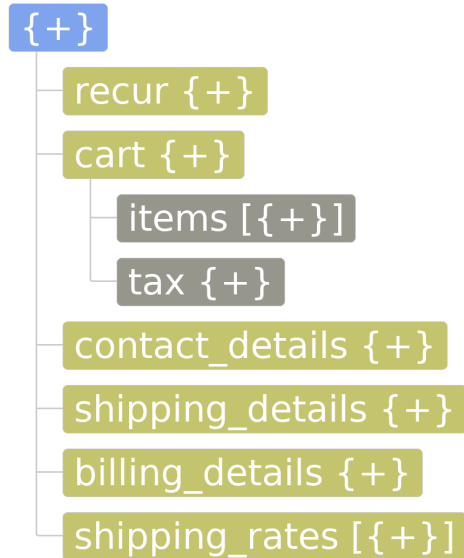
Production:

<https://gateway.moneris.com/chkt/request/request.php>

2.4.1 Preload Request

Transaction requests are sent to the Moneris Checkout server using JSON.

JSON structure overview for Preload request



Request fields for Preload request – Required

| Variable Name | Type and Limits | Description |
|---------------------------------------|---|---|
| store ID store_id | String N/A | Unique identifier provided by Moneris upon merchant account setup |
| API token api_token | String N/A | Unique alphanumeric string assigned upon merchant account activation |
| checkout ID checkout_id | String 30-character alphanumeric | Identifies your Moneris Checkout configuration; this is given to you when you configure your page in the Merchant Resource Center |
| transaction total amount txn_total | String 10-character decimal Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div> | The total dollar amount of the transaction |
| developmental mode | String | Indicates the stage of development |

| Variable Name | Type and Limits | Description |
|---------------|-----------------|---|
| environment | alphabetic | you are sending the request for: testing = qa production = prod |
| request type | <i>String</i> | Type of request being made to Moneris Checkout server |
| action | alphabetic | Allowable values: preload or receipt |

Request fields for Preload request – Optional

| Variable Name | Type and Limits | Description |
|--|---|--|
| order number order_no | <i>String</i> 30-character alphanumeric | The order number is a unique identifier appended to every financial transaction |
| customer ID cust_id | <i>String</i> 30-character alphanumeric | Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant Resource Center |
| dynamic descriptor dynamic_descriptor | <i>String</i> max 20-character alphanumeric total of 22 characters including your merchant name and separator | Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated NOTE: The 22-character maximum limit must take the "/" into account as one of the characters |
| language language | <i>String</i> 2-character alphabetic | Determines which language Moneris Checkout will display information in |

| Variable Name | Type and Limits | Description |
|---------------|-----------------|---|
| | | <p>Allowable values:</p> <p>en – English</p> <p>fr – French</p> |

Additional request objects in Preload request – Optional

| Variable Name | Type and Limits | Description |
|---|----------------------|---|
| Recurring Billing <code>recur</code> | <i>Object</i> N/A | Contains fields related to Recurring Billing |
| Shopping Cart <code>cart</code> | <i>Object</i> N/A | The virtual shopping cart and its contents |
| Contact Details <code>contact_details</code> | <i>Object</i> N/A | Customer contact information This object is returned in the Response to Receipt Request as the Customer Information response object (<code>cust_info</code>) |
| Shipping Details <code>shipping_details</code> | <i>Object</i> N/A | Customer shipping information |
| Billing Details <code>billing_details</code> | <i>Object</i> N/A | Customer billing information |
| Shipping Rates <code>shipping_rates</code> | <i>Object</i> N/A | List of shipping rates to be applied |

2.4.1.1 Optional Preload Request Objects

Moneris Checkout also allows you to send optional objects in the Preload request that reflect additional information entered by the customer at checkout, enable additional features, or meet transaction processing requirements.

If you have configured Moneris Checkout to handle these additional items, you do not send the corresponding object in the Preload request. Only send these optional objects if you are using your own e-commerce page to collect them separately from Moneris Checkout.

Optional objects you can use include:

- Recurring Billing Object
- Shopping Cart Object
- Contact Details Object
- Shipping Details Object
- Billing Details Object
- Shipping Rates Object

The following screenshot shows what you select in the Merchant Resource Center if you are collecting additional items on your own e-commerce page:

Checkout Type

☐ Use Moneris for the complete end to end order process

Use Moneris for the complete order process from the order summary, shipping and payment.

☒ I have my custom order form and want to use Moneris simply for payment processing.

Select this option if you want to embed Moneris payment details within your own order form.

Recurring Billing Object

Optional object

Include this object in Preload request to indicate the start of a series of Recurring Billing transactions that will be managed by Moneris.

Top level object field

recur

Request fields for Recurring Billing object

| Variable Name | Type and Limits | Description |
|--------------------------------------|---|---|
| number of recurs number_of_rekurs | String numeric 1-99 | The number of times that the transaction must recur |
| period recur_period | String numeric 1-999 | Number of recur unit intervals that must pass between recurring billings |
| recurring amount recur_amount | String 10-character decimal, minimum three digits Up to 7 digits (dollars) + decimal point (.) + 2 digits (cents) after the decimal point <div>EXAMPLE: 1234567.89</div> | Dollar amount of the recurring transaction This amount will be billed on the start date, and then billed repeatedly based on the interval defined by period and recur unit |
| recur unit recur_unit | String day, week, month or eom | Unit to be used as a basis for the interval Works in conjunction with the period variable to define the billing frequency |
| start date start_date | String YYMMDD format | Date of the first future recurring billing transaction; this must be a date in the future If an additional charge will be made immediately, the start now variable must be set to true |
| bill now bill_now | String true or false | Set to true if a charge will be made against the card immediately; otherwise set to false |

Shopping Cart Object

Optional object

The shopping cart object can contain multiple items (each item is represented as its own array within the Shopping Cart object).

Top level object field

`cart`

Request fields for Shopping Cart object

| Variable Name | Type and Limits | Description |
|--|--|---|
| shopping cart items <code>items</code> | <i>Object</i> sub-object containing arrays, nested within <code>cart</code> contains following items in blue | Encapsulates the entire array of items in the shopping cart |
| item URL <code>items.url</code> | <i>String</i> alphanumeric | URL that corresponds to the image of the Moneris Checkout shopping cart item |
| item description <code>items.description</code> | <i>String</i> alphanumeric | Describes the item in the shopping cart |
| item product code <code>items.product_code</code> | <i>String</i> 20-character alphanumeric | The SKU for the item |
| item unit cost <code>items.unit_cost</code> | <i>String</i> alphanumeric | Per-unit cost of the item |
| item quantity <code>items.quantity</code> | <i>String</i> numeric 6 characters maximum | Number of individual instances of the given item in the shopping cart |
| subtotal <code>subtotal</code> | <i>String</i> alphanumeric | Total dollar amount of the shopping cart, before taxes |
| tax <code>tax</code> | <i>Object</i> sub-object nested within <code>cart</code> contains following items in blue | Contains information related to taxes charged on the items in the shopping cart |

| Variable Name | Type and Limits | Description |
|---|-------------------------------|-------------------------------------|
| tax amount <code>tax.amount</code> | <i>String</i> alphanumeric | Dollar amount of taxes |
| tax description <code>tax.description</code> | <i>String</i> alphanumeric | Describes type of tax being applied |
| tax rate <code>tax.rate</code> | <i>String</i> alphanumeric | Percentage tax rate charged |

Contact Details Object

Optional object

Top level object field

`contact_details`

Request fields for Contact Details object

| Variable Name | Type and Limits | Description |
|---------------------------------------|---|-----------------------|
| first name <code>first_name</code> | <i>String</i> 30-character alphanumeric | Customer first name |
| last name <code>last_name</code> | <i>String</i> 30-character alphanumeric | Customer last name |
| email <code>email</code> | <i>String</i> 255-character alphanumeric | Customer email |
| phone number <code>phone</code> | <i>String</i> 30-character alphanumeric | Customer phone number |

Shipping Details Object

Optional object

Top level object field

`shipping_details`

Request fields for Shipping Details object

| Variable Name | Type and Limits | Description |
|--------------------------------------|--|---------------------------------------|
| shipping address line 1 address_1 | <i>String</i> 50-character alphanumeric | Customer shipping address |
| shipping address line 2 address_2 | <i>String</i> 50-character alphanumeric | Customer shipping address |
| shipping city city | <i>String</i> 30-character alphanumeric | Customer shipping address city |
| shipping province province | <i>String</i> 30-character alphanumeric | Customer shipping address province |
| shipping city country | <i>String</i> 30-character alphanumeric | Customer shipping address city |
| shipping postal code postal code | <i>String</i> 30-character alphanumeric | Customer shipping address postal code |

Billing Details Object

Optional object

Top level field

billing_details

Request fields for Billing Details object

| Variable Name | Type and Limits | Description |
|-------------------------------------|--|-------------------------------|
| billing address line 1 address_1 | <i>String</i> 50 character alphanumeric | Customer billing address |
| billing address line 2 address_2 | <i>String</i> 50 character alphanumeric | Customer billing address |
| billing city | <i>String</i> | Customer billing address city |

| Variable Name | Type and Limits | Description |
|------------------------------------|-------------------------------------|--------------------------------------|
| city | 30-character alphanumeric | |
| billing province province | String 30-character alphanumeric | Customer billing address province |
| billing country country | String 2-character alphabetic | Customer billing address country |
| billing postal code postal code | String 30-character alphanumeric | Customer billing address postal code |

Shipping Rates Object

Optional array of objects; each shipping option presented is an object

Top level field

shipping_rates

Required Variables for Shipping Rates Object

| Variable Name | Type and Limits | Description |
|--|---|---|
| shipping rate code code | String 20-character alphanumeric | Unique code you create to identify a shipping rate |
| shipping rate description description | String 20-character alphanumeric | Description of the shipping rate category |
| shipping time date | String alphanumeric Suggested form: X days | Describes the turnaround time for shipping, described as number of days (suggested) |
| shipping cost amount | String | Total dollar amount of shipping, as |

| Variable Name | Type and Limits | Description |
|---------------------------------------|----------------------------------|--|
| amount | alphanumeric | per the rate category, on transaction |
| shipping tax amount txn_taxes | String alphanumeric | Total dollar amount of taxes on this transaction |
| transaction total amount txn_total | String free form alphanumeric | This is a snippet |
| default shipping rate default_rate | String true or false | Sets a default rate category |

2.4.1.2 Example Preload Request JSON

This example reflects a Preload request with all optional objects.

```
{
  "store_id": "moneris",
  "api_token": "hurgle",
  "checkout_id": "chkt5BF66neris",
  "txn_total": "452.00",
  "environment": "qa",
  "action": "preload",
  "order_no": "",
  "cust_id": "chkt - cust - 0303",
  "dynamic_descriptor": "dyndesc",
  "language": "en",
  "recur": {
    "bill_now": "true",
    "recur_amount": "1.00",
    "start_date": "2020-1-1",
    "recur_unit": "month",
  }
}
```

```
    "recur_period": "1",
    "number_of_recurs": "10"
  },
  "cart": {
    "items": [
      {
        "url": "ht-tps:\\\\\\\\\\\\esqa.moneris.com\\\\\\\\cr\\\\\\\\checkout\\\\\\\\item1.jpg",
        "description": "One item",
        "product_code": "one_item",
        "unit_cost": "$100.00",
        "quantity": "1"
      },
      {
        "url": "ht-tps:\\\\\\\\\\\\esqa.moneris.com\\\\\\\\cr\\\\\\\\checkout\\\\\\\\item2.jpg",
        "description": "Two item",
        "product_code": "two_item",
        "unit_cost": "$200.00",
        "quantity": "1"
      },
      {
        "url": "ht-tps:\\\\\\\\\\\\esqa.moneris.com\\\\\\\\cr\\\\\\\\checkout\\\\\\\\item3.jpg",
        "description": "Three item",
        "product_code": "three_item",
        "unit_cost": "$100.00",
        "quantity": "1"
      }
    ],
    "subtotal": "$400.00",
    "tax": {
      "amount": "$52.00",
      "description": "Taxes",
      "rate": "13%"
    }
  }
}
```

```
    }
  },
  "contact_details":{
    "first_name":"bill",
    "last_name":"smith",
    "email":"test@moneris.com",
    "phone":"4165551234"
  },
  "shipping_details":{
    "address_1":"1 main st",
    "address_2":"Unit 2012",
    "city":"Toronto",
    "province":"ON",
    "country":"CA",
    "postal_code":"M1M1M1"
  },
  "billing_details":{
    "address_1":"1 main st",
    "address_2":"Unit 2000",
    "city":"Toronto",
    "province":"ON",
    "country":"CA",
    "postal_code":"M1M1M1"
  },
  "shipping_rates":[
    {
      "code":"code01",
      "description":"Standard",
      "date":"10 days",
      "amount":"Free",
      "txn_taxes":"$52.00",
      "txn_total":"452.00",
      "default_rate":"true"
    },
    {
```

```

        "code": "code02",
        "description": "Express",
        "date": "2 days",
        "amount": "$15",
        "txn_taxes": "$53.95",
        "txn_total": "468.95",
        "default_rate": "false"
    }
]
}

```

2.4.2 Response to Preload Request

Response Fields – Response to Preload Request

| Variable Name | Description |
|---|--|
| response <code>"response": {</code> | Top level response object |
| success <code>"success":</code> | Denotes whether the Preload request was successful |
| ticket <code>"ticket":</code> | Identifies the specific Moneris Checkout instance Only returned if success = true |
| error <code>"error": {</code> | Sub-object that encapsulates all errors that occurred as a result of the Preload request Only returned if success = false |
| data <code>"data":</code> | Describes the specific type of error that occurred as a result of some aspect of the Preload request |

2.4.2.1 Example Preload Response – Successful Preload

```

{
  "response": {
    "success": "true",

```

```
    "ticket": "1585G9G9GIKKGGGIGIOG09G9OGKGJFKFJFNjuit8g9"
  }
}
```

2.4.2.2 Example Preload Response – Failed Preload

```
{
  "response": {
    "success": "false",
    "error": {
      "billing_details": {
        "data": "billing address must be set when AVS is enabled"
      }
    }
  }
}
```

2.5 Displaying the Moneris Checkout Page in the Browser

When a customer goes to check out their items for purchase, the Moneris Checkout page is displayed in the <div> tag you created on your web site

To insert the Moneris Checkout instance into the <div>, you call the JavaScript function:

```
monerisCheckout.startCheckout([ticket #])
```

2.6 Handling Callbacks

Callbacks are the means by which Moneris Checkout communicates with your merchant checkout page. All callbacks include a single parameter defined as a JSON-formatted string.

In order to handle callbacks, you need to create JavaScript functions that receive the callbacks being sent by Moneris Checkout when the events occur. These are the functions being referred to as part of the callback set methods, as described in 2.3 Preparing Your Client-Side Checkout Page.

2.6.1 Callback Types

These callbacks are required to be included in the JavaScript of your page:

- Page Loaded
- Cancel Transaction
- Error Event

- Payment Receipt
- Payment Complete

2.6.1.1 Callback Response Fields

| Variable Name | Type and Limits | Description |
|--------------------------------|------------------------|--|
| handler handler | String alphanumeric | Describes the type of callback being used Possible values: cancel_transaction error_event page_loaded payment_complete payment_receipt |
| ticket ticket | String alphanumeric | Identifies the specific Moneris Checkout instance This is also returned in the response to the original Preload |
| response code response_code | String alphanumeric | Identifies the result of the callback For information on response codes, see Callback Response Codes – Moneris Checkout |

2.6.1.2 Page Loaded

Callback Use

To get the page loaded status of the Moneris Checkout page.

This callback is called once the Moneris Checkout is loaded.

JavaScript Set Method for Callback

```
myCheckout.setCallback("page_loaded", myPageLoad);
```

JSON Response Message Format

```
{
  "handler": "page_loaded",
  "ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
  "response_code": "001"
}
```

2.6.1.3 Cancel Transaction

Callback Use

This callback is called in the event the cardholder presses the cancel button in Moneris Checkout.

Standard is to call the `closeCheckout()` method to close the Moneris Checkout `<div>`.

The `closeCheckout()` method will need to be called and a new Preload request will be required in order to initiate a new Moneris Checkout instance.

JavaScript Set Method for Callback

```
myCheckout.setCallback("cancel_transaction", myCancelTransaction);
```

JSON Response Message Format

```
{
  "handler": "cancel_transaction",
  "ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
  "response_code": "001"
}
```

2.6.1.4 Error Event

Callback Use

When an error occurs during the checkout process. This requires the Moneris Checkout session to be closed using the `closeCheckout` function

To attempt the transaction again, a new Preload request must be sent to the Moneris Checkout server in order to get a new transaction ticket number.

JavaScript Set Method for Callback

```
myCheckout.setCallback("error_event", myErrorEvent);
```

JSON Response Message Format

```
{
  "handler": "error_event",
  "ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
  "response_code": "902"
}
```

2.6.1.5 Payment Receipt

Callback Use

Transaction is complete and receipt is ready to be collected.

If you have chosen to have Moneris Checkout generate the receipt, this callback is called once the Moneris Checkout displays the transaction receipt.

JavaScript Set Method for Callback

```
myCheckout.setCallback("payment_receipt", myPaymentReceipt);
```

JSON Response Message Format

```
{
  "handler": "payment_receipt",
  "ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
  "response_code": "001"
}
```

2.6.1.6 Payment Complete

Callback Use

This callback is called once Moneris Checkout has completed payment.

Moneris Checkout should be closed by calling the `closeCheckout()` method

JavaScript Set Method for Callback

```
myCheckout.setCallback("payment_complete", myPaymentComplete);
```

JSON Response Message Format

```
{
  "handler": "payment_complete",
  "ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
}
```

```
"response_code": "001"
}
```

2.7 Implementing Receipt Request Server-to-Server Logic

Once the Payment Complete callback has been called, your merchant website can make the server-to-server Receipt Request call in order to obtain the details of the transaction for the receipt and to determine whether the transaction was approved or declined.

2.7.1 Receipt Request

Once the transaction is finished, you can request the receipt details from the Moneris Checkout server.

Request fields for Receipt Request – Required

| Variable Name | Type and Limits | Description |
|--|--|--|
| store ID <code>store_id</code> | <i>String</i> N/A | Unique identifier provided by Moneris upon merchant account set up |
| API token <code>api_token</code> | <i>String</i> N/A | Unique alphanumeric string assigned upon merchant account activation |
| checkout ID <code>checkout_id</code> | <i>String</i> 30-character alphanumeric (maximum) | Identifies your Moneris Checkout configuration; this is given to you when you configure your page in the Merchant Resource Center |
| ticket number <code>ticket</code> | <i>String</i> maximum 50-character alphanumeric | The unique ticket number that identifies a particular transaction; this returned in the response to the Pre-load request |
| developmental mode <code>environment</code> | <i>String</i> alphabetic | Indicates the stage of development you are sending the request for: testing = qa production = prod |
| request type <code>action</code> | <i>String</i> alphabetic | Type of request being made to Moneris Checkout server Allowable values: <code>preload</code> or <code>receipt</code> |

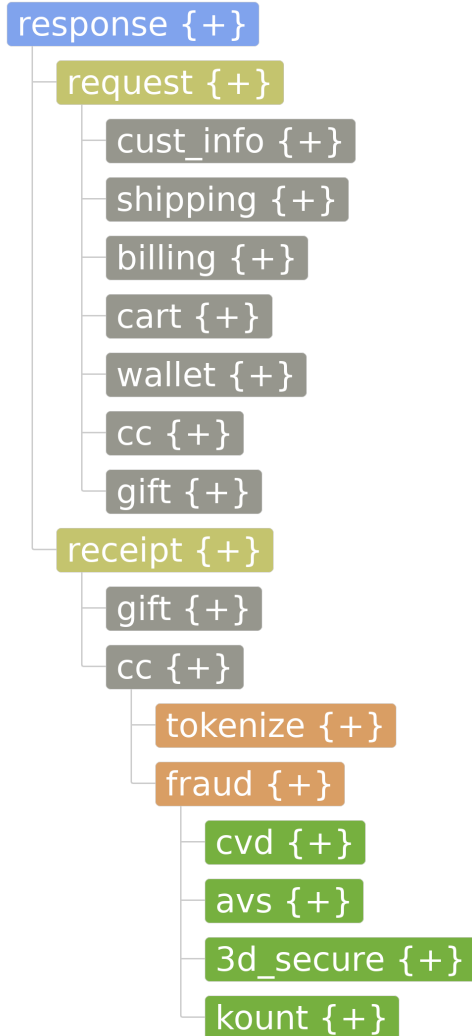
2.7.1.1 Example Receipt Request JSON

```
{  
  "store_id": "example_storeId",  
  "api_token": "example_apiToken",  
  "checkout_id": "example_checkoutId",  
  "ticket": "1539966660vfTyEASfnwNrsQqFE8VkMAOcN169zt",  
  "environment": "qa",  
  "action": "receipt"  
}
```

2.7.2 Response to Receipt Request

Responses to Receipt Requests can contain multiple, nested response objects.

JSON structure for Response to Receipt Request



2.7.2.1 Definition of Response Fields – Response to Receipt Request

The following are fields that may be returned in the Response to Receipt Request, shown with nesting

| Response Field Name and Key | Description |
|-----------------------------|---|
| response {"response": { | Top level response object |
| success "success": | Denotes whether request was successful (i.e., approved, declined) Unsuccessful means error or could not pro- |

| Response Field Name and Key | Description |
|---|--|
| | <p>cess</p> <p>Possible values: <code>true</code> or <code>false</code></p> |
| <p>request</p> <p><code>"request": {</code></p> | Contains information relating to the Preload request |
| <p>transaction total amount</p> <p><code>"txn_total":</code></p> | The total dollar amount of the transaction |
| <p>Customer Information</p> <p><code>"cust_info": {</code></p> | <p>Customer contact information</p> <p>The information presented in this response object will reflect one of three scenarios:</p> <ul style="list-style-type: none"> • If sent in the Preload request, this object will echo the Contact Details object • if Moneris Checkout is set to handle the customer contact information, it will reflect what the customer entered in the web form • If Moneris Checkout was set to not ask for this information, the response object will be empty |
| <p>first name</p> <p><code>"first_name":</code></p> | Customer first name |
| <p>last name</p> <p><code>"last_name":</code></p> | Customer last name |
| <p>phone number</p> <p><code>"phone":</code></p> | Customer phone number |
| <p>email</p> <p><code>"email":</code></p> | Customer email |
| <p>Shipping</p> <p><code>"shipping": {</code></p> | <p>Contains customer shipping information</p> <p>The information presented in this response object will reflect one of three scenarios:</p> |

| Response Field Name and Key | Description |
|---|---|
| | <ul style="list-style-type: none"> If sent in the Preload request, this object will echo the Shipping Details object if Moneris Checkout is set to handle the customer shipping information, it will reflect what the customer entered in the web form If Moneris Checkout was set to not ask for this information, the response object will be empty |
| shipping address line 1 "address_1": | Customer shipping address |
| shipping address line 2 "address_2": | Customer shipping address |
| shipping city "city": | Customer shipping address city |
| shipping country "country": | Customer shipping address country |
| shipping province "province": | Customer shipping address province |
| shipping postal code "postal_code": | Customer shipping address postal code |
| Billing "billing":{ | <p>Contains customer billing information</p> <p>The information presented in this response object will reflect one of three scenarios:</p> <ul style="list-style-type: none"> If sent in the Preload request, this object will echo the Billing Details object if Moneris Checkout is set to handle the customer billing information, it will reflect what the customer entered in the web form |

| Response Field Name and Key | Description |
|---|--|
| | <ul style="list-style-type: none"> If Moneris Checkout was set to not ask for this information, the response object will be empty |
| billing address line 1 <code>"address_1":</code> | Customer billing address |
| billing address line 2 <code>"address_2":</code> | Customer billing address |
| billing city <code>"city":</code> | Customer billing address city |
| billing country <code>"country":</code> | Customer billing address country |
| billing province <code>"province":</code> | Customer billing address province |
| billing postal code <code>"postal_code":</code> | Customer billing address postal code |
| same as shipping <code>"same_as_shipping":</code> | <p>Indicates whether the shipping address is the same as the billing address</p> <p>Possible values: <code>true</code> or <code>false</code></p> |
| Recurring Billing <code>"recur": {</code> | Contains fields related to Recurring Billing |
| number of recurs <code>"number_of_recur":</code> | The number of times that the transaction must recur |
| period <code>"recur_period":</code> | Number of recur unit intervals that must pass between recurring billings |
| recurring amount <code>"recur_amount":</code> | <p>Dollar amount of the recurring transaction</p> <p>This amount will be billed on the start date, and then billed repeatedly based on the interval defined by period and recur unit</p> |

| Response Field Name and Key | Description |
|--|---|
| recur unit <code>"recur_unit":</code> | Unit to be used as a basis for the interval Works in conjunction with the period variable to define the billing frequency |
| start date <code>"start_date":</code> | Date of the first future recurring billing transaction; this must be a date in the future If an additional charge will be made immediately, the start now variable must be set to true |
| bill now <code>"bill_now":</code> | Set to true if a charge will be made against the card immediately; otherwise set to false |
| Shopping Cart <code>"cart":{</code> | The virtual shopping cart and its contents This echos the information contained in the Shopping Cart request object |
| shopping cart items <code>"items": [{</code> | Encapsulates the entire array of items in the shopping cart |
| item URL <code>"url":</code> | URL that corresponds to the image of the Moneris Checkout shopping cart item |
| item description <code>"description":</code> | Describes the item in the shopping cart |
| item product code <code>"product_code":</code> | The SKU for the item |
| item unit cost <code>"unit_cost":</code> | Per-unit cost of the item |
| item quantity <code>"quantity":</code> | Number of individual instances of the given item in the shopping cart |
| subtotal <code>"subtotal":</code> | Total dollar amount of the shopping cart, before taxes |
| tax <code>"tax":</code> | Contains information related to taxes charged on the items in the shopping cart |

| Response Field Name and Key | Description |
|--|---|
| tax amount "amount": | Dollar amount of taxes |
| tax description "description": | Describes type of tax being applied |
| tax rate "rate": | Percentage tax rate charged |
| credit card total "cc_total": | Total amount being charged to the credit card |
| Credit Card (request) "cc": { | Contains cardholder information |
| first 6 last 4 "first6last4": | First 6 and last 4 digits of card number |
| expiry date "expiry": | Card expiry date |
| cardholder "cardholder": | Cardholder name |
| Gift (request) "gift": [{ | Object containing information about a gift card |
| balance remaining "balance_remaining": | The remaining balance on the gift card |
| gift card description "description": | Description of the gift card used for the transaction |
| first 4 last 4 "first4last4": | The first 4 and last 4 digits of the card |
| gift card number "pan": | The account number of the gift card |
| gift card CVD | Card validation digits on the gift card |

| Response Field Name and Key | Description |
|--|---|
| <code>"cvd":</code> | |
| balance used <code>"balance_used":</code> | The amount that was removed from the card's balance as part of the transaction |
| Wallet <code>"wallet":</code> | Contains information from the digital wallet that was used in the transaction |
| wallet type <code>"type":</code> | Type of digital wallet used in this transaction |
| payment data <code>"paymentData":{</code> | Object containing various information related to the payment sent from the digital wallet |
| API version (minor) <code>"apiVersionMinor":</code> | Minor version of the API |
| API version <code>"apiVersion":</code> | Version of the digital wallet's payment API |
| payment method data <code>"paymentMethodData":{</code> | Object containing information about the payment method used in the transaction |
| payment method description <code>"description":</code> | User-facing message to describe the payment method that funds this transaction |
| tokenization data <code>"tokenizationData":{</code> | Object containing information related to tokenization and the digital wallet |
| tokenization type <code>"type":</code> | The type of tokenization to be applied to the selected payment method Possible values: <code>PAYMENT_GATEWAY</code> or <code>DIRECT</code> |
| token <code>"token":</code> | The generated payment method token |
| payment method type <code>"type":</code> | A short identifier for the supported payment method Possible values: |

| Response Field Name and Key | Description |
|--|--|
| | CARD PAYPAL |
| info "info":{ | Object that echoes information about the cardholder, the card and the card network from the digital wallet |
| card network "cardNetwork": | The payment card network |
| card details "cardDetails": | The details about the card; this value is commonly the last four digits of the selected payment account number |
| digital wallet billing address "billingAddress":{ | Object that echoes the cardholder's billing information from the digital wallet |
| address 3 "address3": | Third line of the address |
| sorting code "sortingCode": | The sorting code |
| address 2 "address2": | Second line of the address |
| country code "countryCode": | ISO 3166-1 alpha-2 country code |
| address 1 "address1": | First line of the address |
| postal code "postalCode": | Address postal code or ZIP |
| name "name": | Name of the addressee |
| locality "locality": | City, town, neighbourhood, or suburb |

| Response Field Name and Key | Description |
|---|---|
| administrative area <code>"administrativeArea":</code> | A country subdivision, such as a state or province |
| digital wallet shipping address <code>"shippingAddress": {</code> | Object containing the cardholder's default shipping address information stored in the digital wallet |
| address 3 <code>"address3":</code> | Third line of the address |
| sorting code <code>"sortingCode":</code> | The sorting code |
| address 2 <code>"address2":</code> | Second line of the address |
| country code <code>"countryCode":</code> | ISO 3166-1 alpha-2 country code |
| address 1 <code>"address1":</code> | First line of the address |
| postal code <code>"postalCode":</code> | Address postal code or ZIP |
| name <code>"name":</code> | Name of the addressee |
| locality <code>"locality":</code> | City, town, neighbourhood, or suburb |
| administrative area <code>"administrativeArea":</code> | A country subdivision, such as a state or province |
| ticket number <code>"ticket":</code> | The unique ticket number that identifies a particular transaction; this returned in the response to the Preload request |
| customer ID <code>"cust_id":</code> | Merchant-defined field that can be used as an identifier Searchable from the Moneris Merchant |

| Response Field Name and Key | Description |
|---|---|
| | Resource Center |
| dynamic descriptor "dynamic_descriptor": | <p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> NOTE: The 22-character maximum limit must take the "/" into account as one of the characters </div> |
| order number "order_no": | The order number is a unique identifier appended to every financial transaction |
| electronic commerce indicator "eci": | <p>The e-commerce indicator or crypt type that was used to process the transaction</p> <p>Possible values are:</p> <p>5 - Authenticated e-commerce transaction (3-D Secure)</p> <p>6 - Non-authenticated e-commerce transaction (3-D Secure)</p> <p>7 - SSL-enabled merchant</p> |
| Receipt "receipt":{ | Object containing the receipt information |
| result (financial transaction) "result": | <p>Indicates the result of the financial transaction</p> <p>Possible values are:</p> <p>a = Accepted</p> <p>d = Declined</p> |
| Gift (receipt) "gift":[{ | Contains information related to gift card |
| order number | The order number is a unique identifier |

| Response Field Name and Key | Description |
|--|--|
| <code>"order_no":</code> transaction number <code>"transaction_no":</code> | appended to every financial transaction Moneris Gateway-specific transaction identifier This field is required for any future follow-on transaction requests, such as Refund, Purchase Correction and Pre-Authorization Completion transactions |
| reference number <code>"reference_no":</code> | Terminal used to process the transaction, followed by the shift, batch and sequence number This data is typically used to reference transactions on the host systems, and must be displayed on any receipt presented to the customer This information should be stored by the merchant <div> EXAMPLE Example: 660123450010690030 66012345: Terminal ID 001: Shift number 069: Batch number 003: Transaction number within the batch. </div> |
| response code <code>"response_code":</code> | Transaction response code Possible values are: <50 – transaction approved >=50 – transaction declined NULL – transaction was not sent for authorization For more details on specific response, please see the Response Codes reference topic |
| benefit amount <code>"benefit_amount":</code> | This is the benefit that was generated for the transaction; the amount that was removed from the card as part of the transaction |

| Response Field Name and Key | Description |
|---|--|
| benefit remaining <code>"benefit_remaining":</code> | The remaining balance on the gift card |
| first 6 last 4 <code>"first6last4":</code> | First 6 and last 4 digits of card number |
| Credit Card (receipt) <code>"cc": {</code> | Contains fields describing the response to the credit card transaction |
| order number <code>"order_no":</code> | The order number is a unique identifier appended to every financial transaction |
| customer ID <code>"cust_id":</code> | <p>Merchant-defined field that can be used as an identifier</p> <p>Searchable from the Moneris Merchant Resource Center</p> |
| transaction number <code>"transaction_no":</code> | <p>Used to reference the original transaction when performing a follow-on transaction (i.e., Pre-Authorization Completion, Purchase Correction or Refund)</p> <p>This value is returned in the response of the original transaction</p> <p>Pre-Authorization Completion: references a Pre-Authorization</p> <p>Refund/Purchase Correction: references a Purchase or Pre-Authorization Completion</p> |
| reference number <code>"reference_no":</code> | <p>Terminal used to process the transaction, followed by the shift, batch and sequence number</p> <p>This data is typically used to reference transactions on the host systems, and must be displayed on any receipt presented to the customer</p> <p>This information should be stored by the merchant</p> <div> EXAMPLE Example: 660123450010690030 </div> |

| Response Field Name and Key | Description |
|---|---|
| | <div> 66012345: Terminal ID 001: Shift number 069: Batch number 003: Transaction number within the batch. </div> |
| transaction code <code>"transaction_code":</code> | Type of financial transaction that was performed Possible values: 00 – Purchase 01 – Pre-Authorization |
| transaction type <code>"transaction_type":</code> | ISO transaction code for financial transaction |
| transaction date and time <code>"transaction_date_time":</code> | Processing host date and time stamp Format: YYYY-MM-DD HH:MM:SS |
| corporate card <code>"corporateCard":</code> | Indicates whether the payment card is a corporate card |
| credit card amount <code>"amount":</code> | The total dollar amount that was charged to the credit card |
| response code <code>"response_code":</code> | Transaction response code Possible values are: <50 – transaction approved >=50 – transaction declined NULL – transaction was not sent for authorization For more details on specific response, please see the Response Codes reference topic |
| ISO response code <code>"iso_response_code":</code> | ISO response code returned from issuing institution For more details on specific ISO codes returned, see the Response Codes reference |

| Response Field Name and Key | Description |
|---|--|
| | topic |
| approval code "approval_code": | Authorization code returned from the issuing institution |
| card type "card_type": | <p>Type of payment card used to process the transaction</p> <p>Allowable values:</p> <p>V = Visa</p> <p>M = Mastercard</p> <p>AX = American Express</p> <p>DC = Diner's Card</p> <p>NO = Novus/Discover</p> <p>SE = Sears</p> <p>D = INTERAC® Debit</p> <p>C1 = JCB</p> |
| wallet type "wallet_type": | Type of digital wallet used in this transaction |
| dynamic descriptor "dynamic_descriptor": | <p>Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name</p> <p>Dependent on the card issuer, the statement will typically show the dynamic descriptor appended to the merchant's existing business name separated by the "/" character; additional characters will be truncated</p> <div> <p>NOTE: The 22-character maximum limit must take the "/" into account as one of the characters</p> </div> |
| invoice number "invoice_number": | Identifies an invoice number associated with the transaction |
| customer code | User-defined identifier |

| Response Field Name and Key | Description |
|--|--|
| <p><code>"customer_code":</code></p> <p>electronic commerce indicator</p> <p><code>"eci":</code></p> | <p>The e-commerce indicator or crypt type that was used to process the transaction</p> <p>Possible values:</p> <p>5 - Authenticated e-commerce transaction (3D-Secure)</p> <p>6 - Non-authenticated e-commerce transaction (3D-Secure)</p> <p>7 - SSL-enabled merchant</p> |
| <p>CVD result code</p> <p><code>"cvd_result_code":</code></p> | <p>Indicates the CVD validation result</p> <p>The first byte is the numeric CVD indicator sent in the request; the second byte is the response code</p> <p>Possible response codes are shown in the CVD Response Codes reference</p> |
| <p>AVS result code</p> <p><code>"avs_result_code":</code></p> | <p>Indicates the address verification result</p> <p>For a full list of possible response codes refer to the AVS Response Codes reference</p> |
| <p>first 6 last 4</p> <p><code>"first6last4":</code></p> | <p>First 6 and last 4 digits of card number</p> |
| <p>expiry date</p> <p><code>"expiry_date":</code></p> | <p>Expiry date of the card</p> <p>MMYY format</p> |
| <p>recur success</p> <p><code>"recur_success":</code></p> | <p>Indicates whether the recurring billing transaction has been successfully set up for future billing</p> <p>Possible values: <code>true</code> or <code>false</code></p> |
| <p>issuer ID</p> <p><code>"issuer_id":</code></p> | <p>Unique identifier for the cardholder's stored credentials</p> <p>Sent back in the response from the card brand when processing a Credential on File transaction</p> <p>If the cardholder's credentials are being stored for the first time, and the issuer ID</p> |

| Response Field Name and Key | Description |
|--|---|
| | <p>was returned in the response, you must save the issuer ID on your system to use in subsequent Credential on File transactions (applies to merchant-initiated transactions only)</p> <p>The issuer ID must be saved to your systems when returned from Moneris Gateway in the response data, regardless if the value was received or not</p> <p>As a best practice, if the issuer ID is not returned and you received a value of NULL instead, store that value and send it in the subsequent transaction</p> |
| ECR (electronic cash register) number "ecr_no": | Terminal ID/ECR Number from the request |
| batch number "batch_no": | Batch number; also presented as a component of the reference number |
| sequence number "sequence_no": | Transaction number within the batch; also presented as a component of reference number |
| result (financial transaction) "result": | <p>Indicates the result of the financial transaction</p> <p>Possible values are:</p> <p>a = Accepted</p> <p>d = Declined</p> |
| Tokenize "tokenize":{ | Contains information related to the tokenization of cardholder credentials |
| success (tokenize) "success": | <p>Indicates whether the card was successfully tokenized</p> <p>Possible values: <code>true</code> or <code>false</code></p> |
| first 4 last 4 "first4last4": | The first 4 and last 4 digits of the card |
| data key | Unique identifier for a Vault profile, and used |

| Response Field Name and Key | Description |
|---------------------------------------|--|
| "data_key": | in future Vault financial transactions to associate a transaction with that profile |
| tokenization status "status": | Specifies what type of failure, if any, occurred during the tokenization request Possible values: 001 = approved 940 = Invalid profile id (on tokenization request) 941 = Error generating token 942 = Invalid Profile ID, or source URL 943 = Card data is invalid (not numeric, fails mod10, we will remove spaces) 944 = Invalid expiration date (mmyy, must be current month or in the future) 945 = Invalid CVD data (not 3-4 digits) |
| tokenization message "message": | Provides additional details about the success or failure of the tokenization |
| Fraud "fraud": { | Contains sub-objects that describe information related to fraud tool inquiries |
| CVD "cvd": { | Contains information related to the CVD fraud tool |
| decision origin "decision_origin": | Possible values: Moneris or Merchant |
| CVD result "result": | Possible values: 1 = Success 2 = Failed 3 = Not performed 4 = Card not eligible |
| condition "condition": | Indicates whether this fraud tool was set as a factor for Moneris to use when making an automatic decision on a transaction |

| Response Field Name and Key | Description |
|--|--|
| | <p>Possible values are as follows:</p> <p>0 = Optional</p> <p>1 = Mandatory</p> |
| status "status": | <p>Indicates whether the fraud tool inquiry was performed, and if it was used for auto-decisioning purposes</p> <p>Possible values:</p> <p>success = Fraud tool successful</p> <p>failed = Fraud tool failed (non-auto decision)</p> <p>disabled = Fraud tool not performed</p> <p>ineligible = Fraud tool was selected but card is not a credit card or card not eligible</p> <p>failed_optional = Fraud tool failed and auto decision is optional</p> <p>failed_mandatory = Fraud tool failed auto decision is mandatory</p> |
| CVD code "code": | <p>CVD result code; for a list of possible codes see the CVD Response Codes reference</p> |
| condition "details": | <p>Indicates whether this fraud tool was set as a factor for Moneris to use when making an automatic decision on a transaction</p> <p>Possible values are as follows:</p> <p>0 = Optional</p> <p>1 = Mandatory</p> |
| AVS "avs": { | <p>Contains information related to the AVS fraud tool</p> |
| decision origin "decision_origin": | <p>Possible values: Moneris or Merchant</p> |
| AVS result "result": | <p>Possible values:</p> <p>1 = Success</p> <p>2 = Failed</p> <p>3 = Not performed</p> |

| Response Field Name and Key | Description |
|--|---|
| | 4 = Card not eligible |
| condition "condition": | Indicates whether this fraud tool was set as a factor for Moneris to use when making an automatic decision on a transaction Possible values are as follows: 0 = Optional 1 = Mandatory |
| status "status": | Indicates whether the fraud tool inquiry was performed, and if it was used for auto-decisioning purposes Possible values: success = Fraud tool successful failed = Fraud tool failed (non-auto decision) disabled = Fraud tool not performed ineligible = Fraud tool was selected but card is not a credit card or card not eligible failed_optional = Fraud tool failed and auto decision is optional failed_mandatory = Fraud tool failed auto decision is mandatory |
| AVS code "code": | AVS result code; for a list of potential codes, see the AVS Response Codes reference |
| details "details": | Provides detailed information about the fraud tool query Only populated for Kount and 3-D Secure |
| 3-D Secure "3d_secure":{ | Contains information related to the 3-D Secure fraud tool |
| decision origin "decision_origin": | Possible values: Moneris or Merchant |
| 3-D Secure result "result": | Possible values: 1 = Success 2 = Failed |

| Response Field Name and Key | Description |
|--|--|
| | <p>3 = Not performed</p> <p>4 = Card not eligible</p> |
| <p>condition</p> <p>"condition":</p> | <p>Indicates whether this fraud tool was set as a factor for Moneris to use when making an automatic decision on a transaction</p> <p>Possible values are as follows:</p> <p>0 = Optional</p> <p>1 = Mandatory</p> |
| <p>status</p> <p>"status":</p> | <p>Indicates whether the fraud tool inquiry was performed, and if it was used for auto-decisioning purposes</p> <p>Possible values:</p> <p>success = Fraud tool successful</p> <p>failed = Fraud tool failed (non-auto decision)</p> <p>disabled = Fraud tool not performed</p> <p>ineligible = Fraud tool was selected but card is not a credit card or card not eligible</p> <p>failed_optional = Fraud tool failed and auto decision is optional</p> <p>failed_mandatory = Fraud tool failed auto decision is mandatory</p> |
| <p>3-D Secure code</p> <p>"code":</p> | CAVV result code |
| <p>details</p> <p>"details":{</p> | <p>Provides detailed information about the fraud tool query</p> <p>Only populated for Kount and 3-D Secure</p> |
| <p>Cardholder Authentication Value (CAVV)</p> <p>"cavv":</p> | <p>Value provided by the Moneris MPI or by a third-party MPI</p> <p>Returned by Visa Secure, Mastercard Identity Check or American Express SafeKey transactions</p> |
| <p>3-D Secure message</p> <p>"message":</p> | Describes the reasoning for the outcome of the 3-D Secure inquiry |

| Response Field Name and Key | Description |
|--|--|
| | <p>Possible values:</p> <p>"Authentication Not Available"</p> <p>"Unable to Verify Enrollment"</p> <p>"Successful Payer Authentication"</p> <p>"Cardholder Not Participating"</p> <p>"failed 3-D Secure authentication"</p> <p>"Successful Merchant Attempt"</p> |
| <p>VERes</p> <p>"VERes":</p> | <p>Verification response code</p> <p>Possible values:</p> <p>N = The card/issuer is not enrolled</p> <p>U = The card type is not participating</p> <p>Y = The card is enrolled</p> |
| <p>PARes</p> <p>"PARes":</p> | <p>Payer authentication response code</p> |
| <p>load 3-D Secure</p> <p>"loadvbv":</p> | <p>Only present with value "true" if page was successfully redirected from the 3-D Secure site.</p> |
| <p>Kount</p> <p>"kount": {</p> | <p>Contains information related to the Kount fraud tool</p> |
| <p>decision origin</p> <p>"decision_origin":</p> | <p>Possible values: Moneris or Merchant</p> |
| <p>Kount result</p> <p>"result":</p> | <p>Possible values:</p> <p>001 = Success</p> <p>973 = Unable to locate merchant Kount details</p> <p>984 = Data error</p> <p>987 = Invalid transaction</p> |
| <p>condition</p> <p>"condition":</p> | <p>Indicates whether this fraud tool was set as a factor for Moneris to use when making an automatic decision on a transaction</p> <p>Possible values are as follows:</p> |

| Response Field Name and Key | Description |
|---|--|
| | 0 = Optional 1 = Mandatory |
| status "status": | Indicates whether the fraud tool inquiry was performed, and if it was used for auto-decisioning purposes Possible values: success = Fraud tool successful failed = Fraud tool failed (non-auto decision) disabled = Fraud tool not performed ineligible = Fraud tool was selected but card is not a credit card or card not eligible failed_optional = Fraud tool failed and auto decision is optional failed_mandatory = Fraud tool failed auto decision is mandatory |
| Kount code "code": | Possible values: 001 = Success 973 = Unable to locate merchant Kount details 984 = Data error 987 = Invalid transaction |
| details "details":{ | Provides detailed information about the fraud tool query Only populated for Kount and 3-D Secure |
| Kount response code "responseCode": | Final risk score returned from Kount system |
| message (Kount) "message": | Brief description message about the Kount inquiry |
| receipt ID "receiptID": | The order ID echoed from the original financial transaction |
| Kount result "result": | Possible values are as follows: 1 = Success |

| Response Field Name and Key | Description |
|-----------------------------|--|
| | 2 = Failed 3 = Not performed 4 = Card not eligible |
| Kount score "score": | Final risk score returned from Kount system |
| Kount error "error": | List of errors the Kount request generated |

2.7.2.2 Example JSON Response to Receipt Request

```
{
  "response": {
    "success": "true",
    "request": {
      "txn_total": "452",
      "cust_info": {
        "first_name": "bill",
        "last_name": "smith",
        "phone": "4165551234",
        "email": "test@moneris.com"
      },
      "shipping": {
        "address_1": "1 main st ",
        "address_2": "Unit 2000",
        "city": "Toronto",
        "country": "Canada",
        "province": "Ontario",
        "postal_code": "M1M1M1"
      },
      "billing": {
        "address_1": "1 main st ",
        "address_2": "Unit 2000",
        "city": "Toronto",
```

```
    "country": "Canada",
    "province": "Ontario",
    "postal_code": "M1M1M1"
  },
  "cc_total": "252.00",
  "gift": [
    {
      "balance_remaining": "0.00",
      "Description": "Gift Fixed Reload",
      "first4last4": "*****0214",
      "pan": "0211020000001000214",
      "cvd": "123",
      "balance_used": 200.00
    }
  ],
  "cc": {
    "first6last4": "4761735637",
    "expiry": "0121",
    "cardholder": "bill smith"
  },
  "ticket": "157556871510pAi0PJRFZaOISm4DHQvigFswDnET ",
  "cust_id": " chkt - cust - 0303",
  "dynamic_descriptor": "dyndesc",
  "order_no": null,
  "eci": "7"
},
"receipt": {
  "result": "a",
  "gift": [
    {
      "order_no": "1583250405Ad1BmCSsfHHDeu4_g1",
      "transaction_no": "6198-1583250435590-00157838_15",
      "reference_no": "3276071",
      "response_code": "000",
      "benefit_amount": "200.00",
```

```
        "benefit_remaining":"0.00",
        "first6last4":"0211020214"
    }
],
"cc":{
    "order_no":"1572443908zc9cIHRI6aJTwZU",
    "cust_id":null,
    "transaction_no":"5246-0_14",
    "reference_no":"660187030012750100",
    "transaction_code":"00",
    "transaction_type":"200",
    "transaction_date_time":"2019-10-30 09:59:14",
    "corporateCard":"false",
    "amount":"252.00",
    "response_code":"027",
    "iso_response_code":"01",
    "approval_code":"851268",
    "card_type":"V",
    "dynamic_descriptor":"dyndesc",
    "invoice_number":null,
    "customer_code":null,
    "eci":"7",
    "cvd_result_code":"1M",
    "avs_result_code":"M",
    "first6last4":"4761735637",
    "expiry_date":"0121",
    "recur_success":null,
    "issuer_id":"570106735143835",
    "ecr_no":"66018703",
    "batch_no":"275",
    "sequence_no":"010",
    "result":"a",
    "tokenize":{
        "success":"true",
        "firstlast4":"4761***5637",
```

```
"datakey":"1RNJbQUpcsw4qRkd3LpZ5Lpi7",
"status":"001",
"message":"Successfully registered CC details."
},
"fraud":{
  "cvd":{
    "decision_origin":"Merchant",
    "result":"1",
    "condition":"1",
    "status":"success",
    "code":"1M",
    "details":""
  },
  "avs":{
    "decision_origin":"Merchant",
    "result":"1",
    "condition":"1",
    "status":"success",
    "code":"M",
    "details":""
  },
  "3d_secure":{
    "decision_origin":"Moneris",
    "result":"1",
    "condition":"1",
    "status":"success",
    "code":"5",
    "details":{
      "VERes":"Y",
      "PAREs":"true",
      "message":"Successful Payer Authentication",
      "cavv":"AAACAD1zFBchc3F5NHMUEwAAAAA=",
      "loadvbv":true
    }
  },
}
```

```
"kount":{
  "decision_origin":"Merchant",
  "result":"2",
  "condition":"0",
  "status":"failed_optional",
  "code":"001",
  "details":{
    "responseCode":"001",
    "message":"Success",
    "receiptID":"1567780065NjErgHzsUubOkoIA31gaSUtLkxKseT",
    "result":"D",
    "error":[
      "DECLINE when Device Country is not the same as the
BIN Country for VISA and MC & Data Collector is present",
      "DECLINE when the BIN country is not the same as the
Billing Country for VISA and MC"
    ]
  }
}
```

2.8 Terminating the Moneris Checkout Instance

To terminate the Moneris Checkout instance, call `monerisCheckout.closeCheckout()`, for example:

```
myCheckout.closeCheckout([ticket #]);
```


3 Additional Features in Moneris Checkout

- 3.1 Tokenization of Credentials With Moneris Checkout
- 3.2 Fraud Tools in Moneris Checkout
- 3.3 Window Size in Moneris Checkout

3.1 Tokenization of Credentials With Moneris Checkout

You can configure Moneris Checkout to store a cardholder's credentials in the Moneris Vault and receive a token that represents those credentials for use in future transactions.

If you want to tokenize credentials in Moneris Checkout transactions, you select the **Tokenize Card** option in the Merchant Resource Center

For more information, see the Merchant Resource Center documentation available for download on the Moneris developer portal at:

developer.moneris.com

3.2 Fraud Tools in Moneris Checkout

- 3.2.1 About Fraud Tools in Moneris Checkout
- 3.2.2 Kount as a Fraud Tool in Moneris Checkout
- 3.2.3 Fraud Tools and Auto Decision-Making

3.2.1 About Fraud Tools in Moneris Checkout

Several tools to mitigate the risk of fraud are available for transactions in Moneris Checkout, including:

- AVS
- CVD
- 3-D Secure
- Kount

To select which of these tools to use when performing transactions with Moneris Checkout, go to your Moneris Checkout configurator in the Moneris Merchant Resource Center under the Payment Security section.

For more information, see the Merchant Resource Center documentation available for download on the Moneris developer portal at:

developer.moneris.com

NOTE: CVD is always enabled as a fraud tool and will be performed on each transaction request in Moneris Checkout, but you can choose whether Moneris will treat the CVD result as a mandatory or optional factor to approve or deny the transaction.

3.2.2 Kount as a Fraud Tool in Moneris Checkout

If you select Kount as a fraud tool in Moneris Checkout and your company has its own Enterprise service account from Kount, you will need to include your Kount Merchant ID, Kount API Key and Kount Website ID when you configure your Moneris Checkoutstore in the Merchant Resource Center.

If you are using Moneris's basic fraud service package and do not have your own Kount enterprise account, you do not require this information.

3.2.3 Fraud Tools and Auto Decision-Making

Moneris Checkout can be configured to automatically proceed with or deny transactions as a result of a risk assessment it makes based on the responses it receives from the selected fraud tools.

When you check the box for auto decision-making, you also can choose whether each fraud tool's analysis will be treated by Moneris as an optional or mandatory factor in the decision to approve or deny the transaction.

This information applies to all fraud tools with the following exception:

- 3-D Secure, which is always mandatory *if* enabled

3.3 Window Size in Moneris Checkout

You can customize the appearance of the Moneris Checkout window presented to the customer on their web browser, including how much of the browser window will be taken up by Moneris Checkout.

The default sizing behaviour of the Moneris Checkout window is full-screen, i.e., Moneris Checkout fills the entire web page. You can alter this behaviour to present the customer with a windowed view instead. Opting for a window view requires a few extra steps; for more information on what to do, see [1 Implementing Window View in Moneris Checkout](#).

You configure the sizing along with other aspects of the Moneris Checkout window in the Merchant Resource Center.

4 Testing Your Moneris Checkout Integration

In the testing stage of development:

1. Use the testing Merchant Resource Center at <https://esqa.moneris.com/mpg> to configure your Moneris Checkout page for testing purposes
2. Use the testing URL for server to server requests:
`https://gatewayt.moneris.com/chkt/request/request.php`
3. Reference the testing JavaScript library:
`<script src="https://gatewayt.moneris.com/chkt/js/chkt_v1.00.js"></script>`
4. Set your **monerisCheckout** object to the testing mode:
`myCheckout.setMode("qa");`
5. In all Preload requests use the value "qa" for the **environment** variable
6. In all Preload requests, make sure that you are using the testing version of your credentials for **store ID**, **API token** and **checkout ID**
7. In all Receipt requests use the value "qa" for the **environment** variable
8. In all Receipt requests, make sure that you are using the testing version of your credentials for **store ID**, **API token** and **checkout ID**

5 Moving to Production with Moneris Checkout

Once you have finished testing your Moneris Checkout integration, do the following to move the integration into production:

1. Ensure that you have duplicated your final testing configuration in your Moneris Checkout production configuration in the production Merchant Resource Center at <https://esqa.moneris.com/mpg> to configure your Moneris Checkout page for testing purposes
2. Use the production URL for server to server requests:
`https://gateway.moneris.com/chkt/request/request.php`
3. Reference the production JavaScript library:
`<script src="https://gateway.moneris.com/chkt/js/chkt_v1.00.js"></script>`
4. Set your **monerisCheckout** object to the production mode:
`myCheckout.setMode("prod");`
5. In all Preload requests use the value "prod" for the **environment** variable
6. In all Preload requests, make sure that you are using the production version of your credentials for **store ID**, **API token** and **checkout ID**
7. In all Receipt requests use the value "prod" for the **environment** variable
8. In all Receipt requests, make sure that you are using the production version of your credentials for **store ID**, **API token** and **checkout ID**

6 Reference

- 6.1 Callback Response Codes – Moneris Checkout
- 6.2 AVS Response Codes – Moneris Checkout
- 6.3 CVD Response Codes – Moneris Checkout
- 6.4 CAVV Result Codes

6.1 Callback Response Codes – Moneris Checkout

| Response Code | Reason |
|---------------|-------------------------------|
| 001 | Success |
| 902 | 3-D Secure failed on response |
| 2001 | Invalid ticket/ticket re-use |

6.2 AVS Response Codes – Moneris Checkout

| Code | Visa | Mastercard/Discover | American Express/ JCB |
|------|---|---|---|
| A | Street address matches, zip/postal code does not; acquirer rights not implied | Address matches, zip/postal code does not | Billing address matches, zip/postal code does not |
| B | Street address matches; zip/postal code not verified due to incompatible formats (acquirer sent both street address and zip/postal code) | N/A | N/A |
| C | Street address not verified due to incompatible formats (acquirer sent both street address and zip/postal code) | N/A | N/A |
| D | Street address and zip/- | N/A | Customer name incorrect; |

| Code | Visa | Mastercard/Discover | American Express/ JCB |
|------|--|---|--|
| | postal code match | | zip/postal code matches |
| E | N/A | N/A | Customer name incorrect, billing address and zip/- postal code match |
| F | <i>Applies to UK only:</i> Street address and zip/postal code match | N/A | Customer name incorrect; billing address matches |
| G | <p>Address information not verified for international transaction</p> <p>Any of following may be true:</p> <ul style="list-style-type: none"> • Issuer is not an AVS participant, or • AVS data was present in the request but issuer did not return an AVS result, or • Visa performs AVS on behalf of the issuer and there was no address record on file for this account | N/A | N/A |
| I | Address information not verified | N/A | N/A |
| K | N/A | N/A | Customer name matches |
| L | N/A | N/A | Customer name and zip/- postal code match |
| M | Street address and zip/- postal code match | N/A | Customer name, billing address, and zip/postal code match |
| N | No match; acquirer sent: | Neither address nor zip/- postal code matches | Billing address and zip/- postal code do not match |

| Code | Visa | Mastercard/Discover | American Express/ JCB |
|------|---|---|---|
| | <ul style="list-style-type: none"> postal/ZIP code only, or street address only, or both postal code and street address <p>Also used when acquirer requests AVS but sends no AVS data</p> | | |
| O | N/A | N/A | Customer name and billing address match |
| P | Zip/postal code match; acquirer sent both zip/-postal code and street address, but street address not verified due to incompatible formats | N/A | N/A |
| R | <p>Retry; system unavailable or timed out</p> <p>Issuer ordinarily performs AVS, but was unavailable</p> <div> <p>NOTE: Code R is used by Visa when issuers are unavailable; issuers should refrain from using this code.</p> </div> | Retry; system unable to process | System unavailable; retry |
| S | N/A | AVS currently not supported | AVS currently not supported |
| T | N/A | Nine-digit zip code matches; address does not match | N/A |
| U | <p>Address not verified for domestic transaction, for any of the following reasons:</p> <ul style="list-style-type: none"> Issuer is not an AVS | No data from issuer-/authorization system | Information is unavailable |

| Code | Visa | Mastercard/Discover | American Express/ JCB |
|------|---|---|---|
| | participant, or <ul style="list-style-type: none"> • AVS data was present in the request but issuer did not return an AVS result, or • Visa performs AVS on behalf of the issuer and there was no address record on file for this account | | |
| W | Not applicable; if present, replaced with Z by Visa <i>Available for U.S. issuers only</i> | For U.S. addresses, nine-digit postal code matches, address does not For addresses outside the U.S., postal code matches, address does not | Customer name, billing address, and zip/postal code are all correct matches |
| X | N/A | For U.S. addresses, nine-digit postal code and address match For addresses outside the U.S., postal code and address match | N/A |
| Y | Street address and zip/-postal code match | Billing address and zip/-postal code both match | Billing address and zip/-postal code both match |
| Z | Zip/postal code matches; street address does not match, or street address not included in request | For U.S. addresses, five-digit zip code matches, address does not match | Zip/postal code matches, billing address does not |

6.3 CVD Response Codes – Moneris Checkout

CVD verification is available for Visa, Mastercard, Discover, American Express, JCB and UnionPay transactions.

| Code | Description |
|-------|---|
| M | Match |
| N | No match |
| P | Not processed |
| S | CVD should be on the card, but Merchant has indicated that CVD is not present |
| U | Issuer is not a CVD participant |
| Y | Match for American Express/JCB only |
| D | Invalid security code for American Express or JCB only |
| Other | Invalid response code |

6.4 CAVV Result Codes

The Cardholder Authentication Verification Value (CAVV), the Accountholder Authentication Value (AAV), and the American Express Verification Value (AEVV), are the values that allows Visa, Mastercard and American Express to validate the integrity of the Visa Secure, Mastercard Identity Check and American Express SafeKey transaction data. These values are passed back from the issuer to the merchant after the authentication has taken place.

The merchant then integrates the CAVV/AAV/AEVV value into the authorization request using the Purchase with 3-D Secure or Pre-Authorization with 3-D Secure transaction type, described below:

1. Merchant conducts 3D-Secure authentication request and receives CAVV/AAV/AEVV value in response.
2. Merchant sends the CAVV/AAV/AEVV value to Moneris using the Purchase or Pre-Authorization with 3-D Secure transaction type and receives the CAVV result code in the response.

Visa CAVV result codes

| Result Code | Message | Significance to Merchants |
|-------------|-------------------------------------|--|
| Blank | CAVV not present or not verified | Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks |
| 0 | CAVV authentication results invalid | Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks |
| 1 | CAVV failed validation | Provided that you have implemented the Visa |

| Result Code | Message | Significance to Merchants |
|-------------|---|---|
| | (authentication) | Secureprocess correctly, the liability for this transaction should remain with the Issuer for chargeback reason codes covered by Visa Secure. |
| 2 | CAVV passed validation (authentication) | Fully authenticated transaction. There is a liability shift and the merchant is protected from chargebacks. |
| 3, 8, A | CAVV passed validation (attempt) | Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks. |
| 4, 7, 9 | CAVV failed validation (attempt) | Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks. |
| 6 | CAVV not validated - Issuer not participating | Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks. |
| B | CAVV passed validation; information only | Not a Visa Secure transaction. No liability shift and merchant is not protected from chargebacks |
| C | CAVV was not validated (attempt) | Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks. |
| D | CAVV was not validated (authentication) | Visa Secure has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks. |

Mastercard CAVV result codes

| Result Code | Message | Significance to Merchants |
|-------------|--------------------------|---|
| 0 | Authentication failed | Not a Mastercard Identity Check transaction. No liability shift and merchant is not protected from chargebacks |
| 1 | Authentication attempted | Mastercard Identity Check has been attempted. There is a liability shift and the merchant is protected from certain card fraud-related chargebacks (international commercial cards excluded). |

| Result Code | Message | Significance to Merchants |
|-------------|---------------------------|---|
| 2 | Authentication successful | Fully authenticated transaction. There is a liability shift and the merchant is protected from chargebacks. |

American Express CAVV result codes

NOTE: American Express SafeKey is only available to American Express direct acquired merchants (i.e., not OptBlue merchants). Any questions pertaining to chargebacks, liability and disputes should be addressed to your American Express representative given that American Express is the acquirer of record for these merchants.

| Result Code | Description |
|-------------|--|
| 1 | AEVV Failed - Authentication, Issuer Key |
| 2 | AEVV Passed - Authentication, Issuer Key |
| 3 | AEVV Passed - Attempt, Issuer Key |
| 4 | AEVV Failed - Attempt, Issuer Key |
| 7 | AEVV Failed - Attempt, Issuer not participating, Network Key |
| 8 | AEVV Passed - Attempt, Issuer not participating, Network Key |
| 9 | AEVV Failed - Attempt, Participating, Access Control Server (ACS) not available, Network Key |
| A | AEVV Passed - Attempt, Participating, Access Control Server (ACS) not available, Network Key |
| U | AEVV Unchecked |