



---

## BE PAYMENT READY

Moneris Checkout Integration Guide

Version: 1.0.0

**Review Draft**

Copyright © Moneris Solutions, 2017

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

---

<b>Getting Help</b>	<b>3</b>
<b>System and Skills Requirements</b>	<b>4</b>
<b>1 About Moneris Checkout</b>	<b>5</b>
<b>2 Building Your Moneris Checkout Integration</b>	<b>6</b>
2.1 Integrating Your Website With Moneris Checkout	6
2.2 Configuring Moneris Checkout in Merchant Resource Center	6
2.3 Moneris Checkout Transaction Process Overview	6
2.4 Moneris Checkout Process Flow	7
2.5 Moneris Checkout Requests	8
2.5.1 Preload Request in Moneris Checkout	8
2.5.1.1 Example Preload JSON Request	10
2.5.2 Receipt Request in Moneris Checkout	13
2.5.2.1 Example Receipt JSON Request	13
2.5.2.2 Example Receipt JSON Response	14
2.5.3 Optional Request Objects	16
2.5.3.1 Shopping Cart Object	16
2.5.3.2 Contact Details Object	17
2.5.3.3 Shipping Details Object	18
2.5.3.4 Billing Details Object	19
2.5.3.5 Shipping Rates Object	19
2.5.4 Callbacks	20
2.5.4.1 Cancel Transaction	20
2.5.4.2 Payment Complete	21
2.5.4.3 Address Change	21
2.5.4.4 Error Event	22
2.5.4.5 Page Loaded	22
2.5.4.6 Payment Receipt	23
<b>3 Testing Your Moneris Checkout Integration</b>	<b>24</b>
<b>4 Moving to Production with Moneris Checkout</b>	<b>25</b>

## Getting Help

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

Getting Started	During Development	Production
<p>Contact our Client Integration Specialists:</p> <p>clientintegrations@moneris.com</p> <p>Hours: Monday – Friday, 8:30am to 8 pm ET</p>	<p>If you are already working with an integration specialist and need technical development assistance, contact our eProducts Technical Consultants:</p> <p>1-866-319-7450</p> <p>eproducts@moneris.com</p> <p>Hours: 8am to 8pm ET</p>	<p>If your application is already live and you need production support, contact Moneris Customer Service:</p> <p>onlinepayments@moneris.com</p> <p>1-866-319-7450</p> <p>Available 24/7</p>

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 separate back-end server with HTTPS certificates for communicating with the Moneris Gateway via JSON

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

- JavaScript
- JSON
- Server-side programming

## 1 About Moneris Checkout

Moneris Checkout gives merchants conducting e-commerce transactions a secure way to process transactions by integrating a Moneris-hosted checkout page into your merchant e-commerce website. Moneris Checkout uses server-to-server communications on the back end with your merchant server that identifies each transaction with a unique ticket number, without exposing this information in the browser.

## 2 Building Your Moneris Checkout Integration

- 2.1 Integrating Your Website With Moneris Checkout
- 2.2 Configuring Moneris Checkout in Merchant Resource Center
- 2.4 Moneris Checkout Process Flow
- 2.5 Moneris Checkout Requests

### 2.1 Integrating Your Website With Moneris Checkout

Integrating your e-commerce website with the Moneris Checkout solution involves a few steps:

1. Configuring the options for your Moneris Checkout page in the Merchant Resource Center
2. Customizing your website's code to work with Moneris Checkout

### 2.2 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 will be using the testing configuration, but later, when you move to production, you will also need to set up a configuration in the production MRC that reflects the one you used for testing.

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

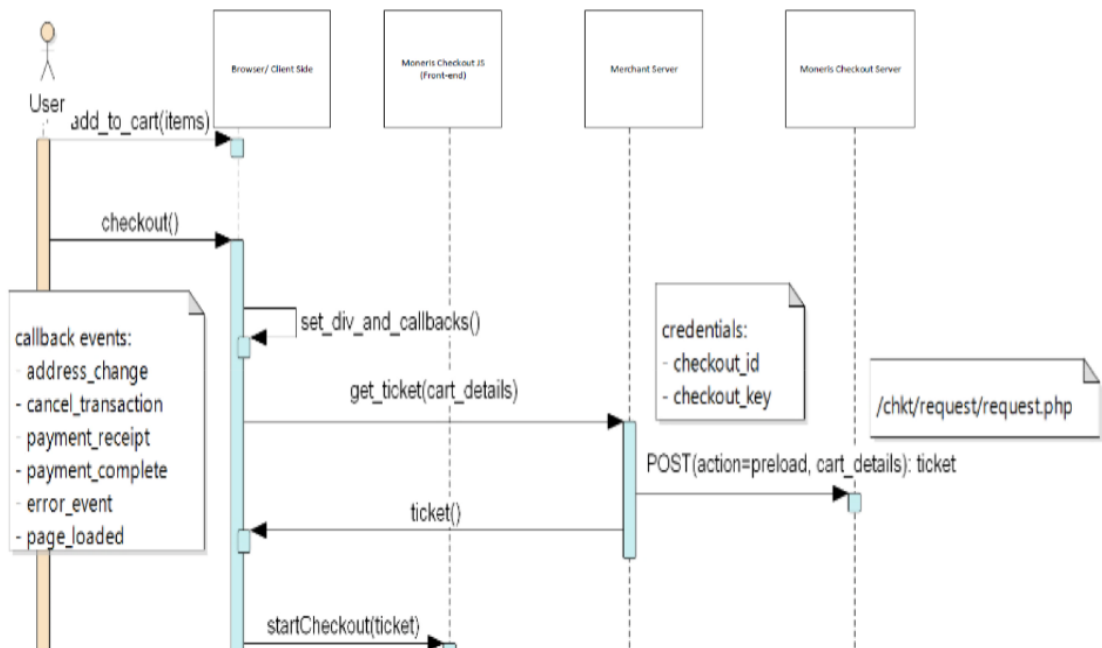
### 2.3 Moneris Checkout Transaction Process Overview

Moneris Checkout transactions follow a general sequence:

1. When the customer clicks to start the checkout process on your merchant e-commerce site, your back-end server sends a preload request to the Moneris Checkout server, containing the Store ID, API Token, Checkout ID and other transaction details
2. Moneris responds with a transaction ticket number and the iFrame form data to be inserted into a div on your website
3. The customer now sees the Moneris Checkout page, where they enter their cardholder details

4. When the customer clicks Checkout on the Moneris Checkout page, the transaction is processed directly by Moneris
5. When transaction is finished, the JavaScript callback `paymentComplete` will be triggered, and depending on how you chose to configure your Moneris Checkout integration:
  - a. If you have chosen to have Moneris Checkout handle the receipt, Moneris will display the receipt in the browser
  - b. If you will handle the receipt, your back-end server sends a Receipt request to the Moneris Checkout server

## 2.4 Moneris Checkout Process Flow



---

## 2.5 Moneris Checkout Requests

There are two types of requests in Moneris Checkout:

- Preload requests, which ask the Moneris Checkout server to respond with a unique transaction ticket number
- Receipt requests, which ask the Moneris Checkout server to respond with the receipt details once a transaction is completed

The transaction ticket number is valid for that transaction only.

### 2.5.1 Preload Request in Moneris Checkout

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

#### Required Preload Request Variables for Moneris Checkout

Attribute or Variable	Size/Type	Description
<code>storeId</code> Store ID	10-character alphanumeric	Unique identifier provided by Moneris upon merchant account set up
<code>apiToken</code> API token	20-character alphanumeric	Unique alphanumeric string assigned upon merchant account activation
<code>checkoutId</code> Checkout ID	alphanumeric	Identifies your Moneris Checkout configuration; this is given to you when you configure your page in the Merchant Resource Center
<code>txn_total</code> Transaction amount	10-character decimal	The total dollar amount of the transaction
<code>integrator</code> Integrator	alphanumeric	Your merchant name
<code>environment</code> Developmental mode	alphabetic <code>qa</code> or <code>prod</code>	Indicates the stage of development you are sending the request for:  testing = <code>qa</code>  production = <code>prod</code>
<code>action</code> Request type	alphabetic	Type of request being made to Moneris Checkout server; is either pre-



Attribute or Variable	Size/Type	Description
	<code>preload</code> or <code>receipt</code>	load or receipt request

### Optional Preload Request Variables

Attribute or Variable	Size/Type	Description
<code>order_no</code> Order number	alphanumeric	
<code>cust_id</code> Customer ID	30-character alphanumeric	Merchant-defined field that can be used for identifying
<code>dynamic_descriptor</code> Dynamic descriptor	maximum 20-character alphanumeric	Merchant-defined description sent on a per-transaction basis that will appear on the credit card statement appended to the merchant's business name

### Optional Objects for Preload Requests in Moneris Checkout

Attribute or Variable	Size/Type	Description
<code>cart</code> Shopping cart	Object	The virtual shopping cart and its contents
<code>contact_details</code> Contact details	Object	Customer contact information
<code>shipping_details</code> Shipping details	Object	Customer shipping information
<code>billing_details</code> Billing details	Object	Customer billing information
<code>shipping_rates</code> Shipping rates	Object	List of shipping rates to be applied

---

### 2.5.1.1 Example Preload JSON Request

This example reflects a preload request with all optional objects.

```
{
  "store_id": "moneris",
  "api_token": "hurgle",
  "checkout_id": "A821neris",
  "integrator": "cr_dev",
  "txn_total": "1000.00",
  "environment": "qa",
  "action": "preload",
  "order_no": "",
  "cust_id": "chkt-cust-1009",
  "dynamic_descriptor": "dyndesc",
  "cart": {
    "items": [
      {
        "url": "https://esqa.moneris.com/cr/checkout/item1.jpg",
        "description": "Oneitem",
        "product_code": "one_item",
        "unit_cost": "$100.00",
        "quantity": "1"
      },
      {
        "url": "https://esqa.moneris.com/cr/checkout/item2.jpg",
        "description": "Twoitem",
        "product_code": "two_item",
        "unit_cost": "$200.00",
        "quantity": "1"
      },
      {
        "url": "https://esqa.moneris.com/cr/checkout/item3.jpg",
        "description": "Three item",
        "product_code": "two_item",

```

```
    "unit_cost": "$100.00",
    "quantity": "1"
  },
  {
    "url": "https://esqa.moneris.com/cr/checkout/item2.jpg",
    "description": "Four item",
    "product_code": "two_item",
    "unit_cost": "$148.00",
    "quantity": "4"
  }
],
"subtotal": "$986.00",
"tax": {
  "amount": "$14.00",
  "description": "Taxes",
  "rate": "15%"
},
"contact_details": {
  "first_name": "firstName",
  "last_name": "lastName",
  "email": "name@email.com",
  "phone": "555-1234",
},
"shipping_details": {
  "address_1": "1 main st",
  "address_2": "Unit 2000",
  "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": "3 days",
      "amount": "$200.00",
      "txn_taxes": "$16.80",
      "txn_total": "1202.80",
      "default_rate": "false"
    },
    {
      "code": "code02",
      "description": "Decline",
      "date": "1 day",
      "amount": "-$999.00",
      "txn_taxes": "$0.04",
      "txn_total": "1.04",
      "default_rate": "true"
    },
    {
      "code": "code03",
      "description": "Pick-up",
      "date": "n/a",
      "amount": "Free",
      "txn_taxes": "$14.00",
      "txn_total": "1000.00",
      "default_rate": "false"
    }
  ]
}
```

## 2.5.2 Receipt Request in Moneris Checkout

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

### Required Receipt Request Variables for Moneris Checkout

Attribute or Variable	Size/Type	Description
<code>storeId</code> Store ID	alphanumeric	Unique identifier provided by Moneris upon merchant account set up
<code>apiToken</code> API token	alphanumeric	Unique alphanumeric string assigned upon merchant account activation
<code>checkoutId</code> Checkout ID	alphanumeric	Identifies your Moneris Checkout configuration; this is given to you when you configure your page in the Merchant Resource Center
<code>ticket</code> Transaction ticket number	alphanumeric	The unique ticket number that identifies a particular transaction; this returned in the response to the pre-load request
<code>environment</code> Developmental mode	alphabetic <code>qa</code> or <code>prod</code>	Indicates the stage of development you are sending the request for:  testing = <code>qa</code>  production = <code>prod</code>
<code>action</code> Request type	alphabetic <code>preload</code> or <code>receipt</code>	Type of request being made to Moneris Checkout server; is either pre-load or receipt request

#### 2.5.2.1 Example Receipt JSON Request

```
{  
  "store_id": "moneris",  
  "api_token": "hurgle",  
  "checkout_id": "A821neris",  
  "ticket": "1539966660vfTyEASfnwNrsQqFE8VkMAOcN169zt",  
  "environment": "qa",  
}
```

---

```
    "action": "receipt"
  }
}
```

### 2.5.2.2 Example Receipt JSON Response

```
{
  "response": {
    "success": "true",
    "request": {
      "txn_total": "7.00",
      "cust_info": {
        "first_name": null,
        "last_name": null,
        "phone": null,
        "email": null
      },
      "shipping": {
        "address_1": "Bloor st",
        "address_2": "3300",
        "city": "Toronot",
        "country": "CA",
        "province": "ON",
        "postal_code": "P0P0P0"
      },
      "billing": {
        "address_1": "3300 Billing Add",
        "address_2": "BAdd2",
        "city": "Toronto",
        "province": "ON",
        "country": "Canada",
        "postal_code": "B0B0B0"
      },
      "cc_total": "7.00",
      "cc": {
        "first6last4": "4012001112",
        "expiry": "1220",

```

```
    "cardholder": "card holder"
  },
  "ticket": "15420486785SOwGk2AOqCNib8VqOuf4a9iDoRlYV",
  "cust_id": "Test Customer ID",
  "dynamic_descriptor": "Test Dynamic Desc",
  "order_no": "TestKP694998976",
  "eci": "7"
},
"receipt": {
  "result": "a",
  "cc": {
    "order_no": "TestKP694998976",
    "cust_id": "Test Customer ID",
    "transaction_no": "15160-0_11",
    "reference_no": "660125050013190080",
    "transaction_code": "00",
    "transaction_type": "200",
    "transaction_date_time": "2018-11-12 13:51:23",
    "corporateCard": "false",
    "amount": "7.00",
    "response_code": "005",
    "iso_response_code": null,
    "approval_code": "078680",
    "card_type": "V",
    "dynamic_descriptor": "Test Dynamic Desc",
    "invoice_number": null,
    "customer_code": null,
    "eci": "7",
    "cvd_result_code": "1P",
    "avs_result_code": null,
    "first4last4": "4012001112",
    "expiry_date": "1220",
    "recur_success": null,
    "ecr_no": "66012505",
    "batch_no": "319",
```

```
        "sequence_no": "008",
        "result": "a"
    }
}
```

### 2.5.3 Optional 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.

To send any optional request object, it must be enabled in your Moneris Checkout configuration in the Merchant Resource Center.

Optional objects you can enable include:

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

#### 2.5.3.1 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 field

`cart`

##### Required Variables for Shopping Cart

Attribute or Variable	Size/Type	Description
<code>items</code> Shopping cart items	sub-object nested within <code>cart</code>	Encapsulates the entire array of items in the shopping cart
<code>url</code> <code>URL</code>	alphanumeric	URL that corresponds to the image Moneris Checkout shopping cart item
<code>description</code>	alphanumeric	Describes the item in the shopping



Attribute or Variable	Size/Type	Description
Description		cart
product_code Product code	20-character alphanumeric	The SKU for the item
unit_cost Unit cost		Per-unit cost of the item
quantity Quantity	numeric 6 characters maximum	Number of individual instances of the given item in the shopping cart
subtotal Subtotal		Total dollar amount of the shopping cart before taxes
tax Tax		
amount Tax amount		
description Tax description		
rate Tax rate		

### 2.5.3.2 Contact Details Object

Optional object

#### Top Level Field

contact\_details

#### Required Variables for Contact Details Object

Attribute or Variable	Size/Type	Description
first_name First name	30-character alphanumeric	Customer first name

Attribute or Variable	Size/Type	Description
last_name Last name	30-character alphanumeric	Customer last name
email e-mail		Customer email
phone Phone number	30-character alphanumeric	Customer phone number

### 2.5.3.3 Shipping Details Object

Optional object

#### Top Level Field

shipping\_details

#### Required Variables for Shipping Details Object

Attribute or Variable	Size/Type	Description
address_1 Shipping address line 1		Customer shipping address
address_2 Shipping address line 2		Customer shipping address
city Shipping city	30-character alphanumeric	Customer shipping address city
province Shipping province	30-character alphanumeric	Customer shipping address province
country Shipping country	30-character alphanumeric	Customer shipping address country
postal code Shipping postal code	30-character alphanumeric	Customer shipping address postal code

### 2.5.3.4 Billing Details Object

Optional object

#### Top level field

`billing_details`

#### Required Variables for Billing Details Object

Attribute or Variable	Size/Type	Description
<code>address_1</code> Billing address line 1		Customer billing address
<code>address_2</code> Billing address line 2		Customer billing address
<code>city</code> Billing city	30-character alphanumeric	Customer billing address city
<code>province</code> Billing province	30-character alphanumeric	Customer billing address province
<code>country</code> Billing country	30-character alphanumeric	Customer billing address country
<code>postal code</code> Billing postal code	30-character alphanumeric	Customer billing address postal code

### 2.5.3.5 Shipping Rates Object

Optional object

#### Top level field

`shipping_rates`

#### Required Variables for Shipping Rates Object

Attribute or Variable	Size/Type	Description
<code>code</code>	numeric	

Attribute or Variable	Size/Type	Description
description	alphanumeric	Description of the shipping rate category
date	alphanumeric X days	Describes the turnaround time for shipping, described as number of days
amount	maximum 10-character decimal	Total dollar amount of shipping, as per the rate category, on transaction
txn_taxes	maximum 10-character decimal	Total dollar amount of taxes on this transaction
txn_total	maximum 10-character decimal	Total dollar amount of transaction
default_rate	true or false	Sets a default rate category

## 2.5.4 Callbacks

Moneris Checkout uses JavaScript callbacks to enable additional actions or events.

These callbacks are required:

- Cancel Transaction
- Payment Complete

The following callbacks are optional:

- Address Change
- Error Event
- Page Loaded
- Payment Receipt

### 2.5.4.1 Cancel Transaction

#### Callback Use

When cardholder cancels transaction.

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

### JavaScript Format

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

### JSON Message Format

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

## 2.5.4.2 Payment Complete

### Callback Use

Payment is complete and Moneris Checkout should be closed by calling the `closeCheckout()` method

### JavaScript Format

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

### JSON Message Format

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

## 2.5.4.3 Address Change

### Callback Use

Used when the shipping address changes, specifically the values for address line 1, postal code and country.

### JavaScript Format

```
myCheckout.setCallback("address_change",myAddressChange);
```

### JSON Message Format

```
{
  "handler":"address_change",
```

---

```
"response": "001",
"ticket": "1539961059DdrvGG3Yj7rxvMAgvRlc4nqKXF7YjT",
"address": {
  "address_1": "1 Main Ave",
  "address_2": "",
  "city": "Toronto",
  "country": "Canada",
  "province": "Ontario",
  "postal_code": "M1M1M1"
}
```

#### 2.5.4.4 Error Event

##### Callback Use

When an error occurs during the checkout process. This requires the Moneris Checkout session to be closed and for a new preload request to be sent to the Moneris Checkout server in order to get a new transaction ticket number.

##### JavaScript Format

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

##### JSON Message Format

#### 2.5.4.5 Page Loaded

##### Callback Use

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

##### JavaScript Format

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

##### JSON Message Format

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

#### 2.5.4.6 Payment Receipt

##### Callback Use

Transaction is complete and receipt is ready to be collected.

If you have chosen to have Moneris Checkout display the receipt, this callback is triggered when the receipt is displayed.

##### JavaScript Format

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

##### JSON Message Format

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

## 3 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. In all preload requests use the value "qa" for the development variable
3. In all preload requests, make sure that you are using the testing version of your credentials for **Store ID**, **API token** and **Checkout ID**



## 4 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:
  - a. Log in to your production MRC store at <https://www3.moneris.com/mpg>
  - b. In the Admin menu, select **Moneris Checkout Config**
  - c. Find your Checkout ID in the list and click **Edit**
2. In all preload requests, use the value **prod** for the `environment` request variable
3. In all preload requests, make sure that you are using the production version of your credentials for **Store ID**, **API token** and **Checkout ID**