Moneris® Go

Integration Guide

Documentation Version: 1.3.3

Documentation Date: November 29, 2024

Contents

1. Document Control

Version	Date	Section	Summary of changes
1.0	Aug-10-2023	All sections	Initial release
1.0.2	Oct-12-2023	5.0 Implementing App to App Integration	Whitelist of IPs and Track
1.1.2	Mar-6-2024	8.0 Implementing additional features on Go terminals	This is a new section
		2.1 Obtaining integration documentation	This is a new section
		3.1 Integration modes and communications supported	Updated chart
		5.0 Implementing App to App Integration	Updated Ingenico – App to App Integration section
1.2.2	Apr-30-2024	8.0 Implementing additional features on Go terminals	Renamed section to Tokenization on Moneris Go Terminals
		9.0 Dynamic Currency Conversion (DCC) on Moneris Go Terminals	This is a new section
		10.0 Store and Forward (SAF) on Moneris Go Terminals	This is a new section
		11.0 Installments Enabled by Visa on Moneris Go Terminals	This is a new section
1.3.2	Jul-23-2024	Title page	Updated device images to include Moneris Go PIN Pad
		All sections	Updated Unified Gateway/eCommerce API References
		4.1 Getting Started	Wording corrections
		4.2 SYNC	This is a new section
		4.3 BIN Ranges Whitelist	This is a new section
		5.0 Implementing App to App Integration	Moved content around for clarity
		10.0 Store and Forward (SAF) on Moneris Go Terminal	Renamed section to Offline Payments on Moneris Go Terminal and updated refences
		10.3 Store and Forward Configuration	Wording corrections
1.3.3	Nov-29-2024	10.0 Offline Payments on Moneris Go Terminals	Updated reference from Store and Forward (SAF) to Offline Payments
		10.1 and 10.3	Updated content for clarity
		10.4 Offline Payments - API Integration	This is a new section

2. Introduction to Integration Solutions

This guide will help you to integrate your secure, PCI-compliant Moneris Go terminal with your own application(s). This will allow your customers to respond to prompts on screen (including tip prompts), and to safely enter their card information. There are three available modes of integration.

App to App Integration

In this mode, you build your own **Android application** for your customers to use on the Moneris Go terminal. When customers want to make an in-app purchase, your application connects to the Moneris payment application on the terminal using our application programming interface (API). The Moneris payment application takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit card(s).

To integrate Moneris Go with your own Android application, please refer to ***Moneris Go - App to App and Direct Integration - API Specification*** document.

Cloud Integration

In this mode, you build your own **point of sale (POS) application** which runs on a separate device. Your POS application connects to the Moneris terminal through the internet and Moneris' Cloud server. You start transactions on your POS application. Transaction requests are sent through the internet to Moneris' Cloud server which routes the transaction to the appropriate payment terminal. The Moneris payment application on the terminal takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit card(s).

To integrate Moneris Go with your POS application, please refer *to* ***Moneris Go Cloud 3.0 - API Specifications Cloud API*** document.

Direct Integration

In this mode, you build your own **point of sale (POS) application** which runs on a separate device. Your POS application connects to the Moneris terminal over your local network using Wi-Fi or Ethernet. Transactions are started on your POS application which passes the transaction request to the terminal. The Moneris payment application on the terminal takes care of transaction processing by communicating with the Moneris host. Your customers simply respond to the prompts on screen to make purchases quickly and securely using their debit or credit cards(s).

To integrate Moneris Go with your POS application, please refer to ***Moneris Go App to App and Direct Integration - API Specification*** document.

2.1

2.1 Obtaining integration documentation

The Moneris Client Consulting team provides merchants and integrators with the following documentation:

- Moneris Go - App to App and Direct Integration - API Specification v0.xx
- Moneris Go Cloud 3.0 - API Specification v3.xx
- Moneris Unified Gateway API Specification
- Moneris Unified eCommerce API Specification

2.1.1 Other documentation resources available online

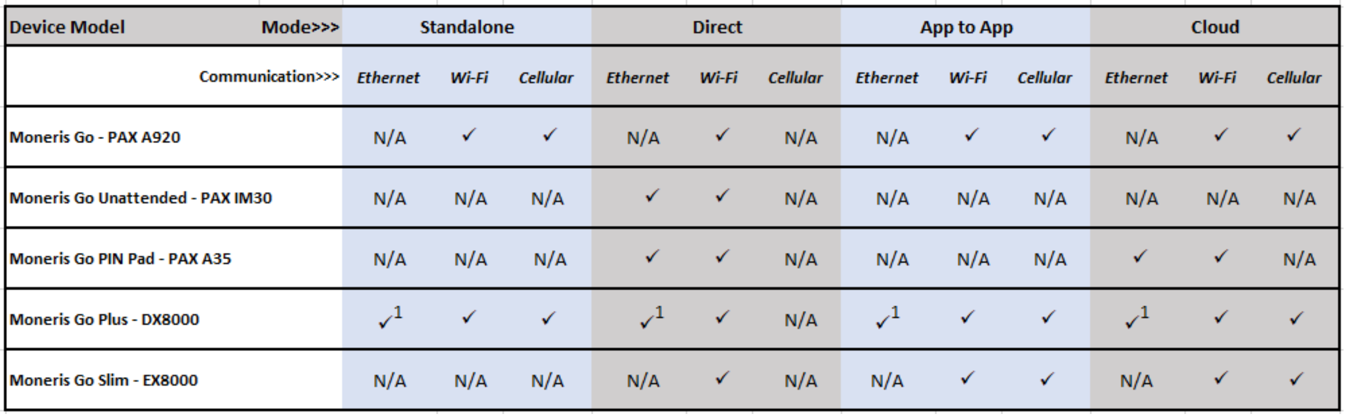
Here are other support resources that are available online:

- Moneris has its own GitHub page, which contains APIs for Moneris eCommerce and Gift & Loyalty products; visit https://github.com/orgs/Moneris/repositories?type=all
- The Moneris Developer Portal contains information on how to integrate, a list of APIs and documentation, DTD or Java .NET & PHP libraries, and FAQs; visit https://developer.moneris.com/
- You can access PAX resources via the PAX developer portal; visit https://developer.pax.us/
- You can access Ingenico resources via the Ingenico developer portal; visit https://ingenico.com/ca-en/developers

3. Communication

Moneris Go terminals can communicate in various ways.

3.1 Integration modes and communications supported



Device Model Mode>>>	Standalone			Direct			App to App			Cloud		
Communication>>>	*Ethernet*	*Wi-Fi*	*Cellular*	*Ethernet*	*Wi-Fi*	*Cellular*	*Ethernet*	*Wi-Fi*	*Cellular*	*Ethernet*	*Wi-Fi*	*Cellular*
Moneris Go - PAX A920	N/A	✓	✓	N/A	✓	N/A	N/A	✓	✓	N/A	✓	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A	✓	✓	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Moneris Go PIN Pad - PAX A35	N/A	N/A	N/A	✓	✓	N/A	N/A	N/A	N/A	✓	✓	N/A
Moneris Go Plus - DX8000	✓[1]	✓	✓	✓[1]	✓	N/A	✓[1]	✓	✓	✓[1]	✓	✓
Moneris Go Slim - EX8000	N/A	N/A	N/A	N/A	✓	N/A	N/A	✓	✓	N/A	✓	✓

1. Functional base required for Ethernet

The terminal can communicate with Moneris, your Electronic Cash Register (ECR) application or manufacturer provided services using any of the communications supported in the above table. When using ethernet or Wi-Fi communication, it would be your responsibility to configure the network so Moneris terminal(s) can communicate with required URL's.

3.2 Preparing your firewall (Wi-Fi and Ethernet connection)

Please refer to *Merchant Network Readiness Checklist*

4. Getting Started

You will need to ensure you have the necessary items for both test and production environments.

4.1 Development resources from Moneris

Check that you have the following items:

- Debug terminal (for App to App only)
- QA Terminal (for Cloud and Direct only)
- API specification for your desired integration mode
- A set of Moneris test cards
- Moneris account credentials for test and production:

Environment	Account credentials
Test	▪ Merchant ID (MID) consisiting of 13 numeric digits beginning with 0030 ▪ Associated Store ID (alphanumeric, typically beginning with mogo- or gwca-) ▪ Terminal ID (TID) consisting of 8 characters ▪ QA Go Portal account (***required prior to device setup***) ▪ API token (for Cloud and to support Tokenization)
Production	▪ Merchant ID (MID) consisiting of 13 numeric digits beginning with 0030 ▪ Associated Store ID (alphanumeric, typically beginning with mogo- or gwca-) ▪ Production Terminal ID (TID) consisting of 8 characters ▪ Production Go Portal account (***required prior to device setup***) ▪ API token (for Cloud and to support Tokenization)

If anything is missing, contact your Moneris Client Consultant for assistance with ordering equipment and setting up the related test account(s).

4.2 SYNC

It is very imporant that ECR and/or applications reside on Moneris devices that are integrated using App to App integration and must support SYNC. SYNC is required in order for the Moneris Go devices to obtain confirmation changes made on the host. These changes may include card plan changes, enabling/disabling various transaction types and/or features (for example: enable/disable TIP). It is recommended to support the “Default” command, but you may build other options in the ECR/APP if required.

4.3 BIN Ranges Whitelist *(Requires Moneris Approval)*

Work with Moneris’ teams for the following tasks:

Whitelist Track[1] (Note: This will be available for future release)

The API command getTrackData can be leveraged to read track2 data. In order to read and accept third party Gift and Loyalty card track2 data using the terminals card reader, you will be required to work with Moneris representative to whitelist the BIN range(s) for those cards. Moneris will require the low and high BIN range along with length of each BIN range. Example: 625173-625173(16) or 62162434626–62162434626(19).

[1] It is your responsibility to inform Moneris should there be any change to the bin-ranges. Moneris is not held responsible should there be any change. Moneris turn-around time to implement bin-ranges is approximately 2 to 4 weeks assuming all accurate data has been provided.

5. Implementing App to App Integration *(Coming Soon)*

You can incorporate App to App integration by following a few steps.

Whitelist of IPs[1]

Moneris provided SIM cards are not exposed to all IP’s on the internet. For any third party applications operating on Moneris terminal utilizing Moneris provided SIM’s will require whitelist of IP addresses (associated to URLs) or endpoints. Communication to IP’s not explicitly whitelisted by Moneris is blocked. Integrators are expected to test their connectivity with SIM cards from all supported Telcos during their integration phase. It is also your responsibility to ask for a SIM Card from each telco Moneris supports.

Duration to Whitelist IP’s: it may take 6 to 8 weeks to whitelist IP’s with the telco’s. It is Integrator’s responsibility to inform Moneris of all IP changes to avoid any impacts to your customers.

Moneris does not support IP’s for individual stores/locations. IP address whitelisted must not exceed more then 10 IP’s per integrator.

- Moneris also doesn’t support URL based whitelisting. While we may ask for URL’s associated to IP’s, they are used for reference purposes.

PAX – App to App Integration

Using app to app integration, you can deploy your own Android application on Moneris terminals allowing your customers to make purchases quickly and securely. Follow this guide to:

- Get access to PAX resources and receive your debug terminal(s) – communication with the Client Consultant.
- It is the integrator’s responsibility to provide list of IP’s associated to URL’s.
- Test your application within your PAX test store.
- Once your application is successfully deployed into the PAX Global marketplace, please work with Moneris representatives to subscribe to the application – adding it to the Moneris PAX test and production marketplace.

[1] Please work with your Moneris representative to whitelist all IPs your application(s) will be communicating with. This includes middleware and APIs. While our telecom partners only have the ability to whitelist by IPs, Moneris will still require the URL. It is your responsibility to inform Moneris should there be any change to IP addresses. Telcos require 8 to 10 weeks to implement change. These timelines may change due to holiday season or any telco’s embargo in affect.

Ingenico – App to App Integration

Using app to app integration, you can deploy your own Android application on Moneris terminals allowing your customers to make purchases quickly and securely. Follow this guide to:

- Get access to Ingenico resources and receive your debug terminal(s) – communication with the Client Consultant.
- Provide your signed test application to Moneris. Moneris will upload the app to the QA Ingenico store. You can then download the application on your QA devices for testing.
- Once tested, please work with Moneris representatives to promote the application to the production store.

[1] Please work with your Moneris representative to whitelist all IPs your application(s) will be communicating with. This includes middleware and APIs. While our telecom partners only have the ability to whitelist by IPs, Moneris will still require the URL. It is your responsibility to inform Moneris should there be any change to IP addresses. Telcos require 8 to 10 weeks to implement change. These timelines may change due to holiday season or any telco's embargo in affect.

5.1 Summary of the process

Below are the high-level steps in the process:

- **Development and testing:**
  1. Develop your own application on a test (debug) terminal from Moneris (refer to section 4.1 Development resources from Moneris for more information).
  2. Upload test application to the PAX test marketplace. Refer to the PAX developer centre for instructions.
- **Production:**
  1. Upload your production application to the PAX production marketplace.
  2. PAX signs your application and makes it available to customers.
  3. Moneris subscribes to your application so that it can be downloaded to Moneris terminals.
  4. Download your application from the Moneris QA and Production PAX marketplace to your production Moneris terminal.

Review each section below to understand how to proceed.

5.2 Contact Moneris

Once you contact Moneris, you will be assigned a Client Consultant who will coordinate your development journey. Moneris will set up a call with PAX to ensure you receive:

- The PAX non-disclosure agreement
- A PAX legal agreement for each development (debug) terminal you request
- Access to the PAX developer portal and test marketplace

Once you sign and return all legal agreements to PAX, Moneris will send you the debug terminal(s) and test cards.

5.3 Develop and test your android application

In addition to the standard Android development tools, you will need the Moneris Go API specification to call the Moneris Go payment application. Please refer to your Moneris Client Consultant on how to obtain the Moneris App to App developer documentation, a PAX A920 debug terminal and test cards.

5.3.1 Application requirements

Your application must conform to the following requirements:

- Printing done using Moneris' receipt API. Receipts must be stored electronically so they can be reprinted if required
- Payment transaction data must be retained for 24 months

Note: *This list is not exhaustive. Refer to the Moneris API specification for complete requirements.*

5.4 Promote your application to production

Follow the steps below to make your application available in the terminal's app store.

PAX

In order for the application to be made available in the PAX store, you are required to upload the application to PAX's production PAX store marketplace. Once PAX has signed the application, it would be your responsibility to inform your Moneris representative. Your Moneris representative will then inform the appropriate group to execute the application subscription process[1].

Once Moneris has subscribed to the application, it would be made available for download on all Moneris devices[2].

[1] This task may take 1 to 2 weeks

[2] Moneris doesn't have the capability to push updates to sets of terminals. It would be your responsibility to inform the merchants so they can update the application.

Ingenico

In order for the application to be made available in the Ingenico app store, you are required to sign the application with the tools provided by Ingenico (If the integrator does not have an Ingenico signing kit, obtain signing card from Ingenico[1]).

Once the application has been signed, you will work with Moneris representative to get the application loaded to the Ingenico app store[2].

[1] This task may take 6 to 8 weeks

[2] Moneris doesn't have the capability to push updates to set of terminals. It would be your responsibility to inform the merchants so they can update the application.

6. Implementing Cloud Integration

This section describes how to put your Moneris Go terminal into cloud integration mode. Once the terminal connects to the Moneris cloud, you can send transactions from your point of sale (POS) software to the terminal for processing. Your customers can use the terminal to securely enter their payment cards and leave tips.

6.1 Cloud requirements

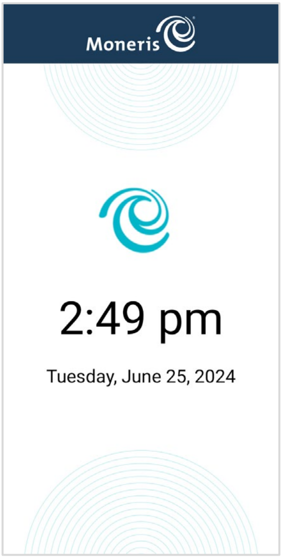
- Gateway store ID and API token
  - For **QA**, refer to your Client Consultant to obtain your store ID and API token
  - For **Production**, your store ID and API token are available through Merchant Direct

  Note: *QA and Production credentials will differ.*

- Moneris Go Cloud 3.0 - API Specification v3.xx
- Moneris Unified Gateway API Specification documentation

6.2 Hardware and cloud integration device setup

Please refer to your device’s specific getting started guide for instructions on how to setup your payment terminal in Cloud operating mode. Once initial setup has been complete, simply power on your device to sync the settings with Moneris.



When you see the idle screen (showing the Moneris logo with the date and time), the device is ready to process transactions sent from your POS software.

7. Implementing Direct Integration

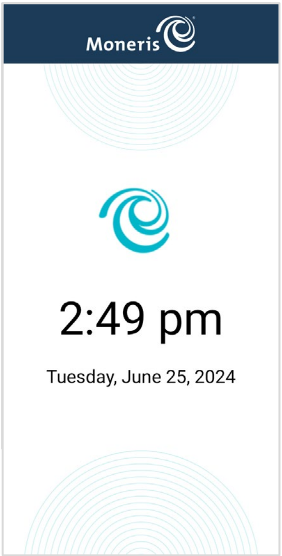
This section describes how to put your Moneris Go terminal into direct integration mode. You can send transactions from your point of sale (POS) software to the terminal for processing over your local area network. Your customers can use the terminal to securely enter their payment cards and leave tips.

7.1 Direct integration pre-requisites

- Local IP address and port of your terminal for communication
- Moneris API documentation (Moneris Go App to App and Direct documentation, *optional* Moneris Unified Gateway API Specification)
  - Usage of the optional Unified Gateway API will require your Gateway store ID and API token

7.2 Hardware and direct integration device setup

Please refer to your device’s specific getting started guide for instructions on how to setup your payment terminal in Direct operating mode. Once initial setup has been complete, simply power on your device to sync the settings with Moneris.



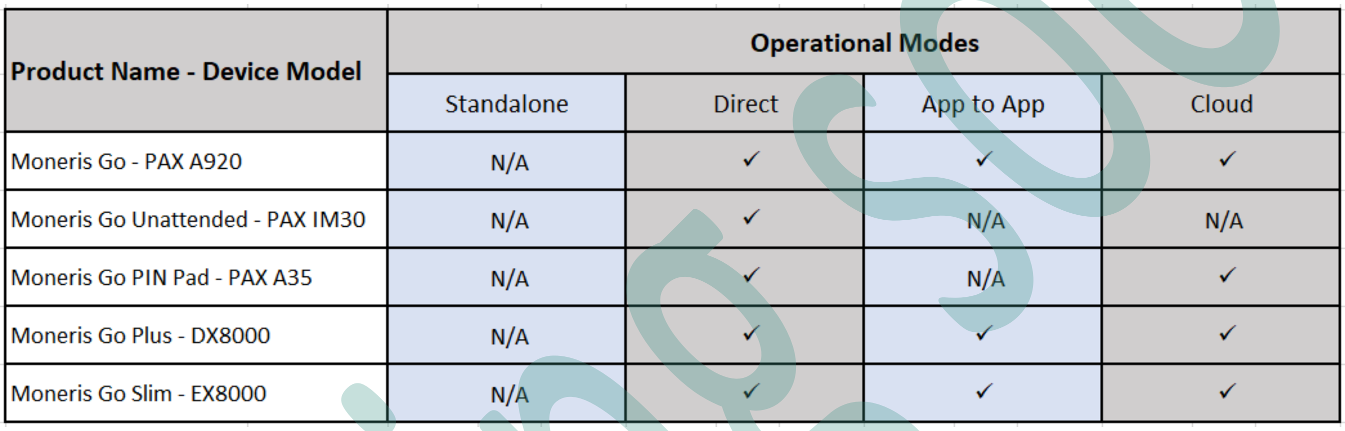
When you see the idle screen (showing the Moneris logo with the date and time), the device is ready to process transactions sent from your POS software.

8. Tokenization on Moneris Go Terminals *(Coming Soon)*

8.1 Implementing Tokenization

This section describes how to implement tokenization features on your Moneris Go terminal for Direct, App to App and Cloud integration mode. Once the integration has been completed and the tokenization feature is enabled, you need to leverage the Unified eCommerce API in order to create, store and manage your tokens via the Moneris Vault.

8.2 Tokenization Supported Devices



Product Name - Device Model	Operational Modes			
	Standalone	Direct	App to App	Cloud
Moneris Go - PAX A920	N/A	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	✓	N/A	N/A
Moneris Go PIN Pad - PAX A35	N/A	✓	N/A	✓
Moneris Go Plus - DX8000	N/A	✓	✓	✓
Moneris Go Slim - EX8000	N/A	✓	✓	✓

8.3 Tokenization Prerequisites

- Gateway store ID and API token
  - For **QA**, refer to your Client Consultant for assistance enabling Tokenization on your existing QA store if not enabled or for assistance obtaining a QA store if one does not exist
  - For **Production**, your store ID and API token are available through Merchant Direct

Note: *QA and Production credentials will differ.*

- Moneris Go API documentation (refer to section 2.1.1)
  - Moneris Go Cloud 3.0 - API specification
  - Moneris Go App to App and Direct Integration - API specification
- Moneris Unified eCommerce API (refer to section 2.1.1)
- DTD or JAVA, .NET, PHP libraries

8.4 Tokenization Flow

On your Moneris Go device, perform a Financial Transaction such as a Card Verification, Purchase or Pre-auth and retain the following information from a successful response:

- orderId,
- transactionId, alternatively known as transaction number

Financial Transactions include the following:

- Purchase
- Refund
- Void (aka Purchase Correction)
- Independent refund
- Pre-Auth
- Completion
- Card verification

*Refer to **Moneris Go API** specifications for additional information.*

Leveraging the Unified eCommerce API, perform a Vault Tokenize Credit Transaction request with the following object values:

- **Mandatory Object Values:**
  - orderId,
  - transactionId, alternatively known as transaction number

  Sample Code (Tokenize Credit Transaction):

```
<?xml version="1.0"?>
<request>
    <store_id>gateway storeID</store_id>
    <api_token>API token</api_token>
    <res_tokenize_cc>
        <order_id>order ID from Go transaction</order_id>
        <txn_number>transaction number from Go transaction (ex.1017-0_140)</txn_number>
    </res_tokenize_cc>
</request>
```

- **Optional Object Values:**
  - Customer ID
  - Data Key
  - Email Address
  - Phone Number
  - Note
  - AVS Information*
  - COF Information

**When passing AVS information the following value are mandatory:*

- AVS street number
- AVE street name
- AVS zip/postal code

Note 1: *A token cannot be generated in the event that a Moneris Go Device has been enabled for Offline Payments and a transaction was approved offline.*

*Refer to **Unified eCommerce API** for additional information associated with adding a token.*

Leveraging the Unified eCommerce API for future financial transaction with credential on file:

- payment_indicator: U
- payment_information: 2

Sample Code (Purchase with Token):

```
<?xml version="1.0"?>
<request>
 <store_id>gateway storeID</store_id>
 <api_token>API token</api_token>
 <status_check>false</status_check>
 <res_purchase_cc>
        <data_key>Vault token</data_key>
        <order_id>example unique order ID</order_id>
        <amount>10.00</amount>
                <crypt_type>7</crypt_type>
                <cof_info>
                        <payment_indicator>U</payment_indicator>
                        <payment_information>2</payment_information>
                </cof_info>
        </res_purchase_cc>
 </request>
```

Note 2: *In the event of a pre-authorization via a Moneris Go Terminal followed by a token request, you have the option to leverage the token to perform a force post transaction, however, the orderID and transactionID must be sourced back to the original pre-auth transaction. Refer to the **Unified eCommerce API** or additional information associated with performing financial transactions via a token.*

Managing tokens:

You may leverage the Unified eCommerce API in order to perform the following token management functions:

- Vault Update Credit Card
- Vault Delete
- Vault Look Up Masked
- Vault Get Expiring
- Vault Add Token

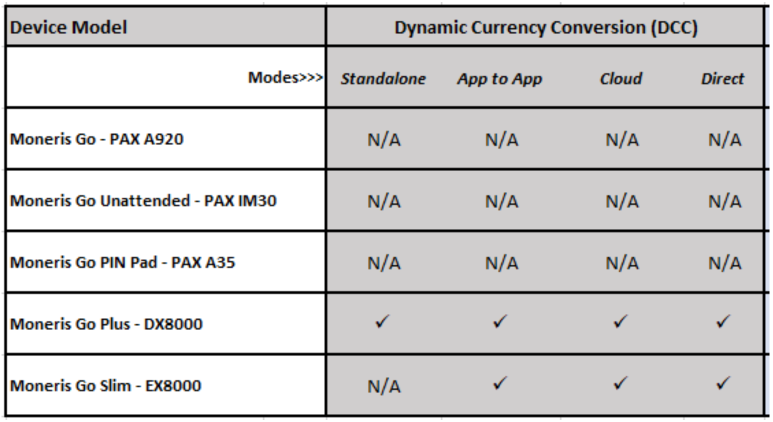
Refer to the ***Unified eCommerce API*** for additional information associated with performing token management functions.

9 Dynamic Currency Conversion (DCC) on Moneris Go Terminals

9.1 Implementing Dynamic Currency Conversion (DCC)

This section describes how Dynamic Currency Conversion (DCC) functions on Moneris Go devices and how integrators can support it. DCC is a feature that allows cardholders the option to pay either in their native currency or in Canadian dollars.

9.2 Dynamic Currency Conversion (DCC) Supported Devices



Device Model	Dynamic Currency Conversion (DCC)			
Modes>>>	*Standalone*	*App to App*	*Cloud*	*Direct*
Moneris Go - PAX A920	N/A	N/A	N/A	N/A
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A	N/A
Moneris Go PIN Pad - PAX A35	N/A	N/A	N/A	N/A
Moneris Go Plus - DX8000	✓	✓	✓	✓
Moneris Go Slim - EX8000	N/A	✓	✓	✓

9.3 Enabling Dynamic Currency Conversion (DCC)

Your client consultant can enable DCC on your Go Devices. You should ensure your device is on the latest software version and sync your terminal with Moneris in order to start using DCC. To reinitialize your terminal, refer to the Using Your Terminal section of the Moneris Reference Guide available at moneris.com/en/support.

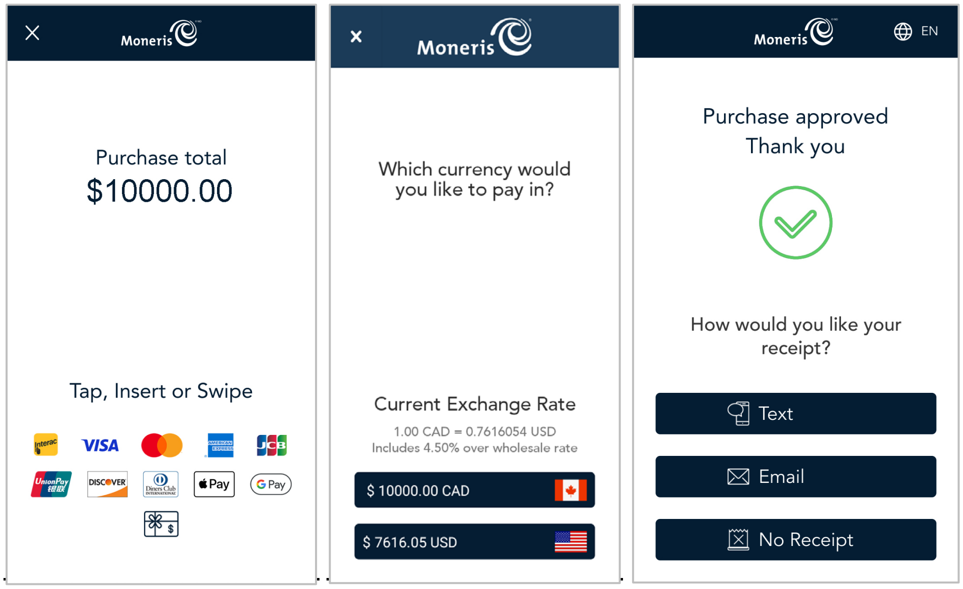
9.4 Supported Cards and Interface

DCC is available for Visa and Mastercard cards issued in a foreign currency for both Contact and Contactless interfaces.

Note: *For Mastercard Contactless transactions, DCC is only offered for mobile wallets.*

9.5 User experience

Example of user screens:



Purchase total

$10000.00

Tap, Insert or Swipe

Which currency would you like to pay in?

Current Exchange Rate

1.00 CAD = 0.7616054 USD

Includes 4.50% over wholesale rate

Purchase approved

Thank you

How would you like your receipt?

9.6 What Integrators need to know?

- There is no development required to support DCC and there are no parameters which need to be sent by the Electronic Cash Register (ECR) to offer DCC on the device.
- An optional DCC opt-in parameter is returned on the response. This is for information purpose only to indicate the cardholder has opt-in to the DCC offer.
- DCC is offered on Purchase, Independent Refund and Pre-Authorization & Completion transaction types. DCC is not offered for Refund transaction.

Sample trace:

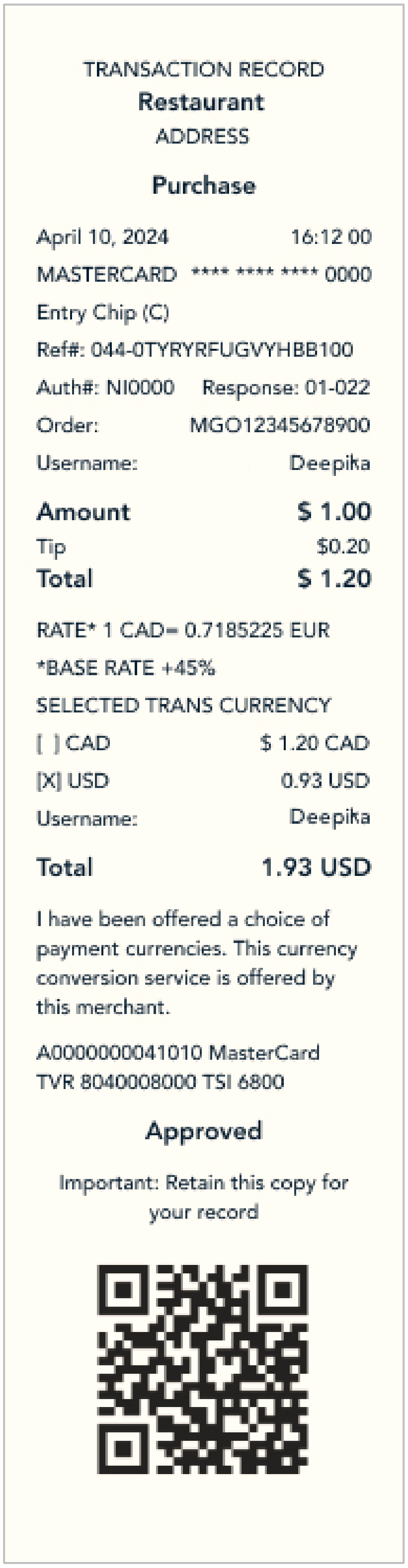
Request:

```
{
  "apiVersion": "3.0",
  "dataId": "1714061235658-001",
  "dataTimestamp": "2024-04-25 12:07:15",
  "data": {
    "request": [
      {
        "idempotencyKey": "4107c774-5d4d-4ab3-bfa6-aaa9e4cc2da6",
        "terminalId": "A2345678",
        "action": "purchase",
        "totalAmount": "1000000"
      }
    ]
  }
}
```

Response:

```
{
  "receipt": {
    "apiVersion": "3.0",
    "dataId": "1714061235658-001",
    "statusCode": "5207",
    "status": "Approved",
    "dataTimestamp": "2024-04-25 12:07:43",
    "data": {
      "response": [
        {
          "statusCode": "5207",
          "status": "Approved",
          "approvedAmount": "1000000",
          "totalAmount": "1000000",
          "cardType": "V ",
          "cardName": "VISA",
          "sequenceNum": "047",
          "realTimeUniqueId": "0SCIA4H6VUP4FY2",
          "responseCode": "027",
          "iso": "01",
          "authCode": "B48739",
          "maskedPan": "************0004",
          "orderId": "MGO1714061241002",
          "transactionId": "1406-0_755",
          "idempotencyKey": "4107c774-5d4d-4ab3-bfa6-aaa9e4cc2da6",
          "action": "purchase",
          "terminalId": "A2345678",
          "saf": "false",
          "dccOptIn": "true",
          "tenderType": "Credit",
          "receiptChoice": "NONE",
          "receipt": "    ------ TRANSACTION RECORD ------     \r\n              COFFEE STORE
                 \r\n             3300 BLOOR ST W              \r\n             ETOBICOKE    ON
                 \r\n\r\n                Purchase                \r\nApr 25,2024
                 12:07:29\r\nVISA                      ************0004\r\nEntry: Chip (C)
                 \r\nRef#: 047- 0SCIA4H6VUP4FY2                \r\nAuth#: B48739 ……}
      ]
    },
  }
```

Sample DCC receipt:



TRANSACTION RECORD

Restaurant

ADDRESS

Purchase

April 10, 2024 16:12 00

MASTERCARD **** **** **** 0000

Entry Chip (C)

Ref#: 044-0TYRYRFUGVYHBB100

Auth#: NI0000 Response: 01-022

Order: MGO12345678900

Username: Deepika

Amount **$ 1.00**

Tip $0.20

Total **$ 1.20**

RATE* 1 CAD= 0.7185225 EUR

*BASE RATE +45%

SELECTED TRANS CURRENCY

[ ] CAD $ 1.20 CAD

[X] USD 0.93 USD

Username: Deepika

Total **1.93 USD**

I have been offered a choice of payment currencies. This currency conversion service is offered by this merchant.

A0000000041010 MasterCard
TVR 8040008000 TSI 6800

Approved

Important: Retain this copy for your record

10. Offline Payments on Moneris Go Terminals

10.1 Processing Offline Payments

This section describes how Offline Payments processing will allow credit transactions to be accepted at the terminal if connection to the Moneris host is unavailable.

The terminal will:

- store eligible financial transactions in a secure fashion should the Moneris host become unreachable
- forward the stored transactions to the host once it becomes available, so that they may be settled by the merchant.

When the host connection resumes, the terminal will send stored Visa and Mastercard transactions to the host.

These transactions will be sent in the background. This will allow new transactions to be accepted while Offline Payments transactions are being forwarded.

Transaction Types:

- Purchase (EMV contact)
- Independent Refund (EMV contact)
- Purchase Void (for purchase transactions stored in the terminal memory)

Card Types:

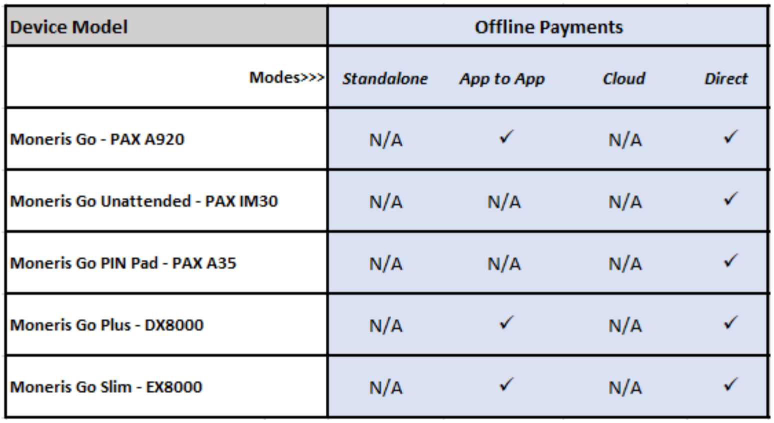
- Visa
- Mastercard
- American Express
- JCB
- Discover

Interface Types:

- Contact chip (contactless not accepted in offline mode)

Note: *Void itself is not Offline Payments. It is available for use while Offline Payments processing is in effect. While communication is down, Void can be used to delete Offline Payments Purchase in the terminal memory. Void cannot be used to nullify a transaction authorized online prior to communication failure. The merchant must wait for communications to be re-established to void online transactions.*

10.2 Offline Payments Supported Devices



Device Model	Offline Payments			
Modes>>>	*Standalone*	*App to App*	*Cloud*	*Direct*
Moneris Go - PAX A920	N/A	✓	N/A	✓
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A	✓
Moneris Go PIN Pad - PAX A35	N/A	N/A	N/A	✓
Moneris Go Plus - DX8000	N/A	✓	N/A	✓
Moneris Go Slim - EX8000	N/A	✓	N/A	✓

10.3 Offline Payments Configuration

There are some limits associated with processing Offline Payments. These limits are as follows:

Maximum Transaction Amount: This corresponds to the maximum value of a transaction that can be processed in Offline Payments mode. For example, if a merchant's Maximum Transaction Amount limit is $100.00, they will not be able to process any transactions that exceed $100.00 while in Offline Payments mode.

Note: *This limit does not apply to Independent Refund Transactions.*

Transaction Count Limit: This corresponds to the maximum total number of Offline Payments transactions that can be stored on a terminal at once. This includes both Purchase and Independent Refund transactions.

Release Delay: Time in minutes defining how long the terminal will stay in offline mode until attempting connection to the Moneris host. During this time period all eligible transactions will be accepted offline. When Release Delay expires the terminal will attempt uploading offline stored transactions to the host. If Release Delay = 0, terminal will always attempt online connection first prior to accepting eligible transaction offline.

10.4 Offline Payments - API Integration

Offline Acceptance Eligibility Flag (safAllowed): Merchant POS system (ECR) must include this parameter in the request message in order to allow offline processing of transaction.

Offline Transaction Indicator (saf): Informs merchant POS system (ECR) whether transaction was accepted offline.

Offline Transaction Counter (safCounter): In every transaction response the terminal will return the number of unprocessed offline transactions stored in the terminal memory.

Refer to "Moneris Go - App to App and Direct Integration - API Specification" for details.

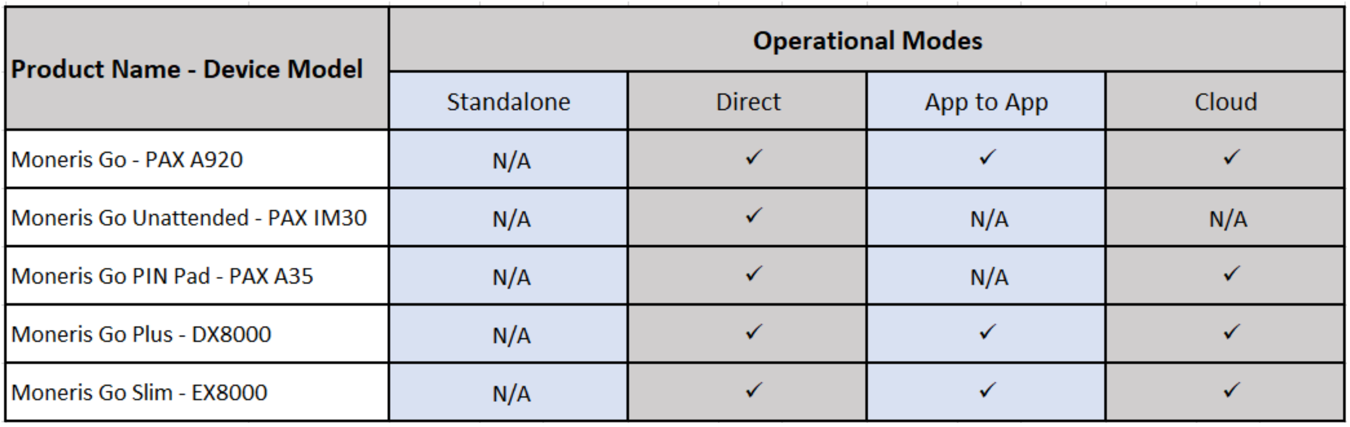
10.5 Clearing Offline Payments from the Terminal

When communication to the host is re-established, the terminal will automatically send any stored transactions in the background. The merchant can continue to use the terminal while Offline Payments transactions are being sent in the background.

11. Installments Enabled by Visa on Moneris Go Terminals

Installments Enabled by Visa offers the cardholder (with the eligible card) the flexibility to pay for merchandise or services in equal payments over a defined period and provides the cardholder with the option: Buy Now Pay Later (BNPL) during purchase.

11.1 Supported Terminals



Product Name - Device Model	Operational Modes			
	Standalone	Direct	App to App	Cloud
Moneris Go - PAX A920	N/A	✓	✓	✓
Moneris Go Unattended - PAX IM30	N/A	✓	N/A	N/A
Moneris Go PIN Pad - PAX A35	N/A	✓	N/A	✓
Moneris Go Plus - DX8000	N/A	✓	✓	✓
Moneris Go Slim - EX8000	N/A	✓	✓	✓

11.2 Enabling Installments by Visa

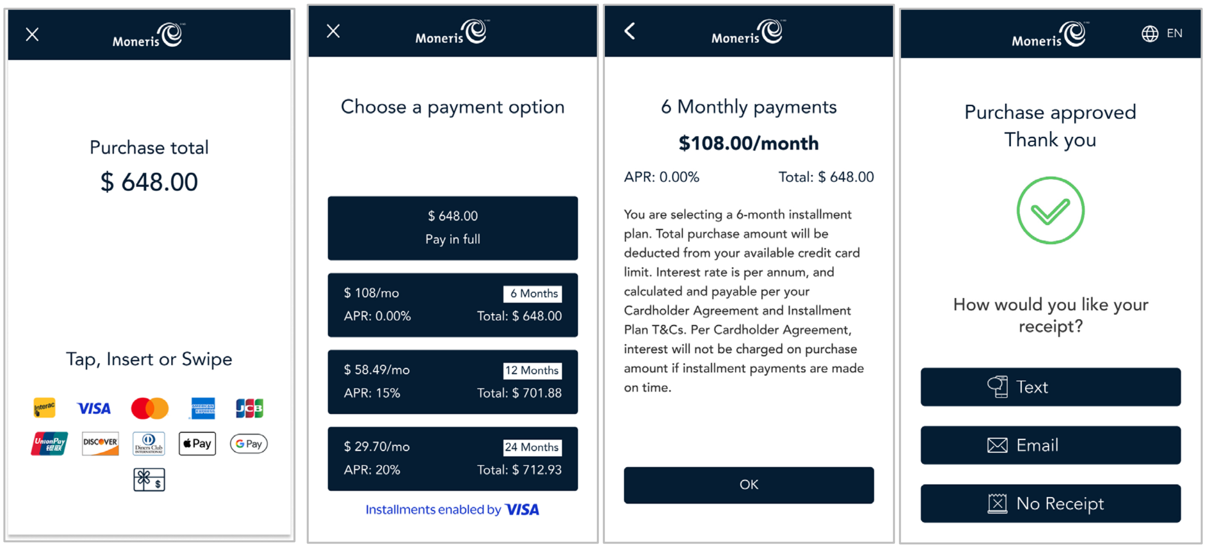
Your Client Consultant can help enable Installments by Visa on your Go Device. You should ensure your device is on the latest software version and sync your terminal with Moneris in order to start using Installments Enabled by Visa. To sync your terminal, refer to the Using Your Terminal section of the Moneris Reference Guide available at moneris.com/en/support.

11.3 Supported Cards and Interface

This feature is available for Visa, Mastercard and American Express cardholders when inserted or tapped.

11.4 User Experience

Example of user screens:



Purchase total

$ 648.00

Tap, Insert or Swipe

Choose a payment option

$ 648.00 Pay in full	
$ 108/mo APR: 0.00%	6 Months Total: $ 648.00
$ 58.49/mo APR: 15%	12 Months Total: $ 701.88
$ 29.70/mo APR: 20%	24 Months Total: $ 712.93

Installments enabled by VISA

6 Monthly payments

$108.00/month

APR: 0.00% Total: $ 648.00

You are selecting a 6-month installment plan. Total purchase amount will be deducted from your available credit card limit. Interest rate is per annum, and calculated and payable per your Cardholder Agreement and Installment Plan T&Cs. Per Cardholder Agreement, interest will not be charged on purchase amount if installment payments are made on time.

Purchase approved
Thank you

How would you like your receipt?

11.5 What Integrators need to know?

- Installment is offered on Purchase and Pre-Authorization transactions. For these transactions there are no parameters which need to be sent by the Electronic Cash Register (ECR) to offer Installments by Visa on the device.
- An installment opt-in parameter is returned on the response to indicate the cardholder has selected an installment plan.
- When initiating Independent Refund transaction, the real time unique identifier parameter must be sent. This information must match the value received from the Purchase response.

Sample Trace:

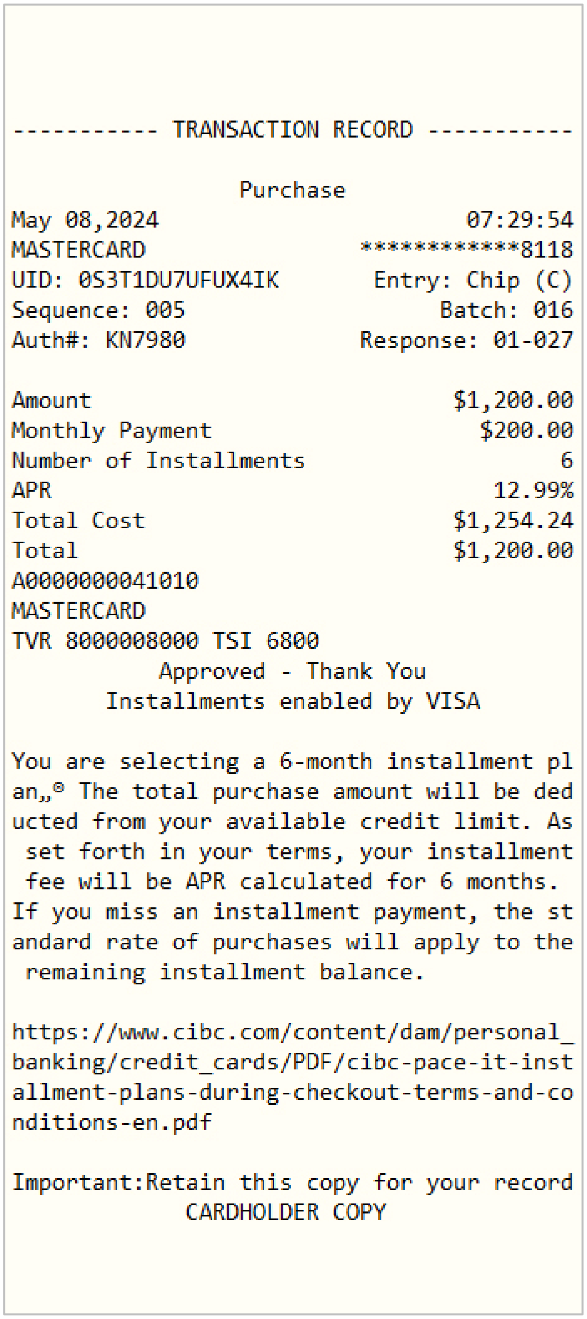
Request

```
{
  "apiVersion": "3.0",
  "dataId": "1714074129526-001",
  "dataTimestamp": "2024-04-25 15:42:09",
  "data": {
    "request": [
      {
        "idempotencyKey": "51283cb9-f551-428a-8ce8-417c40877ffa",
        "terminalId": "A2345678",
        "action": "purchase",
        "totalAmount": "100000"
      }
    ]
  }
}
```

Response

```
{
  "receipt":
  {
    "apiVersion": "3.0",
    "dataId": "1714074129526-001",
    "statusCode": "5207",
    "status": "Approved",
    "dataTimestamp": "2024-04-25 15:42:29",
    "data": {
      "response": [
        {
          "statusCode": "5207",
          "status": "Approved",
          "approvedAmount": "100000",
          "totalAmount": "100000",
          "cardType": "V ",
          "cardName": "VISA",
          "sequenceNum": "010",
          "realTimeUniqueId": "0SCIK2O4R8MMDG0",
          "responseCode": "027",
          "iso": "01",
          "authCode": "B44893",
          "maskedPan": "************3925",
          "orderId": "MGO1714074135255",
          "transactionId": "191-0_755",
          "idempotencyKey": "51283cb9-f551-428a-8ce8-417c40877ffa",
          "action": "purchase",
          "terminalId": "A2345678",
          "saf": "false",
          "installmentOptIn": "true",
          "tenderType": "Credit",
          "receiptChoice": "PRINT",
          "receipt": "     ------ TRANSACTION RECORD ------     \r\n            COFFEE …."
        }
      ]
    },
  }
```

Sample Installment by Visa receipt:



```
----------- TRANSACTION RECORD -----------

                Purchase
May 08,2024                      07:29:54
MASTERCARD               ************8118
UID: 0S3T1DU7UFUX4IK       Entry: Chip (C)
Sequence: 005                  Batch: 016
Auth#: KN7980            Response: 01-027

Amount                          $1,200.00
Monthly Payment                   $200.00
Number of Installments                  6
APR                                12.99%
Total Cost                      $1,254.24
Total                           $1,200.00
A0000000041010
MASTERCARD
TVR 8000008000 TSI 6800
           Approved - Thank You
       Installments enabled by VISA

You are selecting a 6-month installment pl
an„® The total purchase amount will be ded
ucted from your available credit limit. As
 set forth in your terms, your installment
 fee will be APR calculated for 6 months.
If you miss an installment payment, the st
andard rate of purchases will apply to the
 remaining installment balance.

https://www.cibc.com/content/dam/personal_
banking/credit_cards/PDF/cibc-pace-it-inst
allment-plans-during-checkout-terms-and-co
nditions-en.pdf

Important:Retain this copy for your record
             CARDHOLDER COPY
```

12. Support

To obtain support for the terminal solution, follow these directions.

Situation	Directions
Your integration or development provider requires support related to the API specifications and tools.	Contact your Moneris Client Consultant.
You are facing difficulties uploading your application to the PAX marketplace	Contact your PAX representative.
You are facing difficulties uploading your application to the Ingenico marketplace	Contact your Ingenico representative.

MGo Int EN Nov 2024