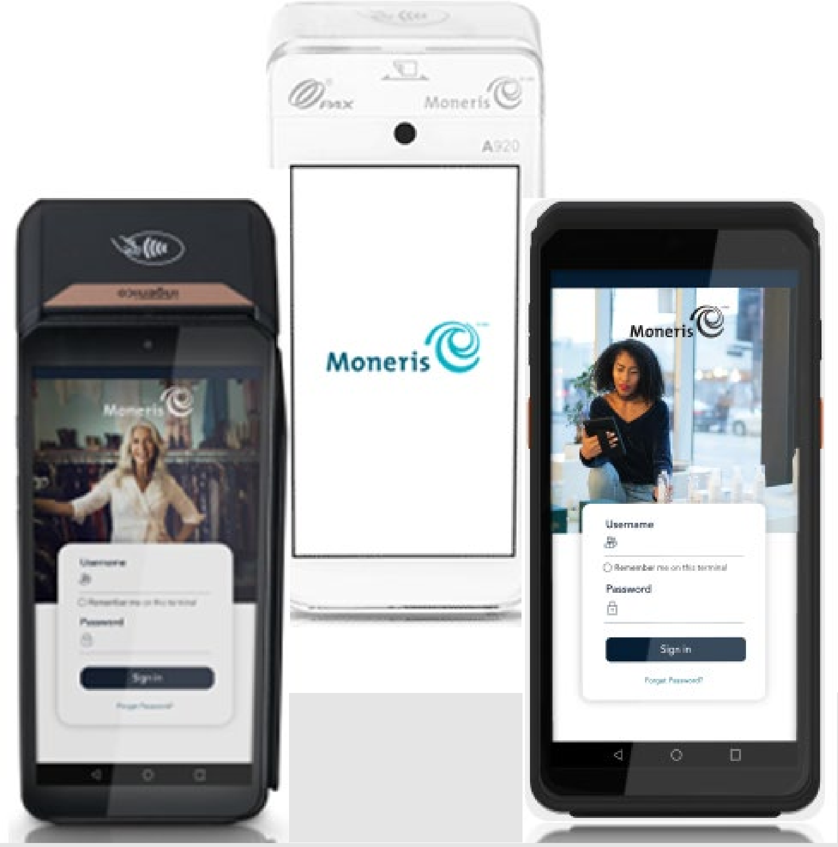
Moneris® Go

Integration Guide



Documentation Version: 1.1.2

Documentation Date: March 6, 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

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
- Moneris Go Cloud 3.0 - API Specifications Cloud API
- Moneris Unified Gateway API Specification
- Moneris Unified eCommerce API Specification

2.1.1 Other documentation resources available online

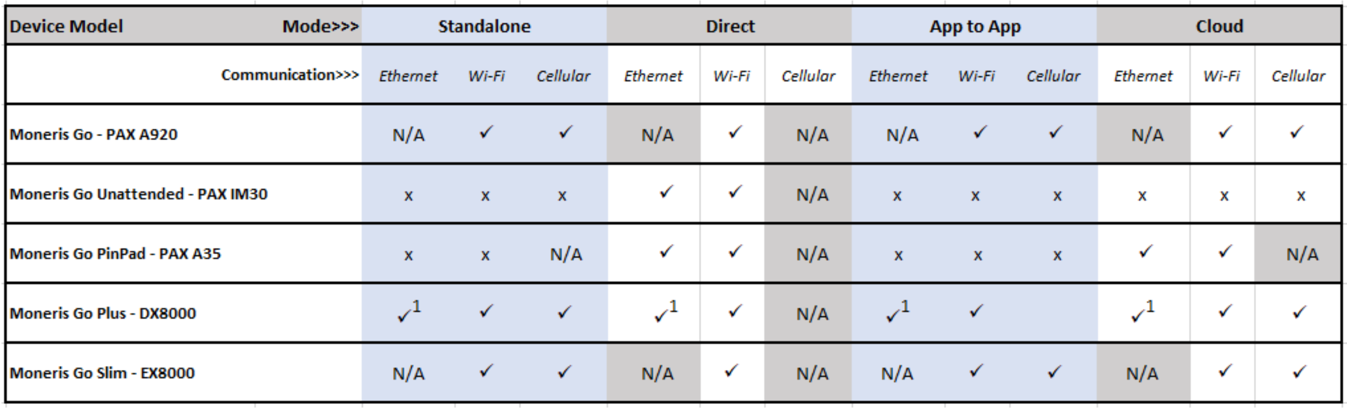
Here are other support resources that are available online:

- 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	x	x	x	✓	✓	N/A	x	x	x	x	x	x
Moneris Go PinPad - PAX A35	x	x	N/A	✓	✓	N/A	x	x	x	✓	✓	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	✓	✓

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 portal:

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***)
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***)

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

5. Implementing App to App Integration

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

3.X 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
- Work with Moneris' teams for the following tasks:

  Whitelist of IPs[1]

  Moneris provided SIM cards route network traffic through a custom Moneris APN. For any third party applications operating on our terminal, it is important to ensure that the IP addresses of any URLs or endpoints required for application operation are provided to Moneris. Communication to IPs not explicitly whitelisted by Moneris will be blocked.

  Whitelist Track[2]

  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).
- 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.

[2] 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.

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
- Work with Moneris' teams for the following tasks:

Whitelist of IPs[1]

Moneris provided SIM cards route network traffic through a custom Moneris APN. For any third party applications operating on our terminal, it is important to ensure that the IP addresses of any URLs or endpoints required for application operation are provided to Moneris. Communication to IPs not explicitly whitelisted by Moneris will be blocked.

Whitelist Track[2]

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).

- 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.

[2] 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.

5.1 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.

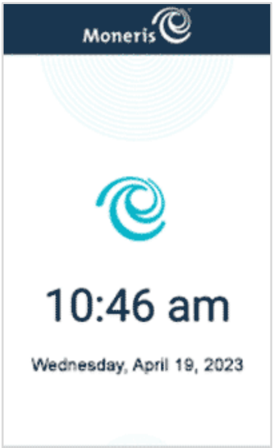
6.1 Cloud requirements

- 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 API documentation (Cloud 3.0 documentation, *optional* Gateway Unified API 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

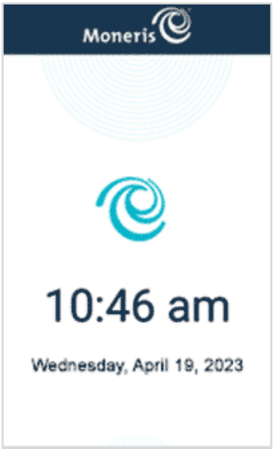
This section describes how to put your Moneris Go terminal into direct integration mode. You can send transactions from your point of sale (POS) software to the terminal for processing over your local area network. Your customers can use the terminal to securely enter their payment cards and leave tips.

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* Gateway Unified API documentation)
  - Usage of the optional Gateway Unified 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. Implementing Additional Features on Moneris Go Terminals

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 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 eCommerce API documentation (refer to section 2.1.1)
- DTD or JAVA, .NET, PHP libraries

8.3 Tokenization Flow

1. 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.*

2. 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 Store and Forward (SaF) and a transaction was approved offline.*

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

3. Leveraging the unified eCommerce API for future financial transaction with credential on file:
   - payment_indicator: U
   - payment_information:

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 completion transaction, however, the orderID and trasnsactionId must be sourced back to the original pre-auth transaction. Refer to the* ***Unified eCommerce API*** *for 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. 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.

This document is for informational purposes only. Neither Moneris Solutions Corporation nor any of its affiliates shall be liable for any direct, indirect, incidental, consequential or punitive damages arising out of use of any of the information contained in this document. Neither Moneris Solutions Corporation or any of its affiliates nor any of our or their respective licensors, licensees, service providers or suppliers warrant or make any representation regarding the use or the results of the use of the information, content and materials contained in this document in terms of their correctness, accuracy, reliability or otherwise. Your credit and/or debit card processing is governed by the terms and conditions of your agreement(s) for merchant credit/debit card processing services with Moneris Solutions Corporation. It is your responsibility to ensure that proper card processing procedures are followed at all times. Please refer to the ***Moneris Merchant Operating Manual*** (available at: **moneris.com/support**) and the terms and conditions of your applicable agreement(s) for credit/debit processing or other services with Moneris Solutions Corporation for details.