

Contents

- 1. Document Control4
- 2. Introduction to Integration Solutions.....5
 - 2.1 Obtaining integration documentation.....6
 - 2.1.1 Other documentation resources available online 6
- 3. Communication7
 - 3.1 Integration modes and communications supported7
 - 3.2 Preparing your firewall (Wi-Fi and Ethernet connection)7
- 4. Getting Started8
 - 4.1 Development resources from Moneris.....8
- 5. Implementing App to App Integration9
 - 5.1 Summary of the process 10
 - 5.2 Contact Moneris..... 11
 - 5.3 Develop and test your android application 11
 - 5.3.1 Application requirements 11
 - 5.4 Promote your application to production 11
- 6. Implementing Cloud Integration13
 - 6.1 Cloud requirements..... 13
 - 6.2 Hardware and cloud integration device setup 13
- 7. Implementing Direct Integration.....14
 - 7.1 Direct integration pre-requisites..... 14
 - 7.2 Hardware and direct integration device setup 14
- 8. Tokenization on Moneris Go Terminals.....15
 - 8.1 Implementing Tokenization 15
 - 8.2 Tokenization Supported Devices 15
 - 8.3 Tokenization Prerequisites 15
 - 8.4 Tokenization Flow 15
- 9. Dynamic Currency Conversion (DCC) on Moneris Go Terminals.....18
 - 9.1 Implementing Dynamic Currency Conversion (DCC) 18
 - 9.2 Dynamic Currency Conversion (DCC) Supported Devices 18
 - 9.3 Enabling Dynamic Currency Conversion (DCC)..... 18
 - 9.4 Supported Cards and Interface 18
 - 9.5 User experience 19
 - 9.6 What Integrators need to know? 19
- 10. Store and Forward (SAF) on Moneris Go Terminals.....22
 - 10.1 Processing Store and Forward (SAF)..... 22
 - 10.2 Store and Forward (SAF) Supported Devices 23

10.3	Store and Forward Configuration.....	23
10.4	Clearing Stored Transactions from the Terminal.....	24
11.	Installments by Visa on Moneris Go Terminals.....	25
11.1	Supported Terminals	25
11.2	Enabling Installments by Visa	25
11.3	Supported Cards and Interface	25
11.4	User Experience.....	26
11.5	What Integrators need to know?	26
12.	Support.....	29

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 by Visa on Moneris Go Terminals	This is a new section

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 process¹.

Once Moneris has subscribed to the application, it would be made available for download on all Moneris devices².

¹This task may take 1 to 2 weeks

²Moneris 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 Ingenico¹).

Once the application has been signed, you will work with Moneris representative to get the application loaded to the Ingenico app store².

¹This task may take 6 to 8 weeks

²Moneris 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

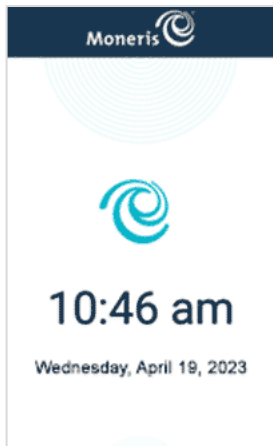
This section describes how to put your Moneris Go terminal into cloud integration mode. Once the terminal connects to the Moneris cloud, you can send transactions from your point of sale (POS) software to the terminal for processing. Your customers can use the terminal to securely enter their payment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

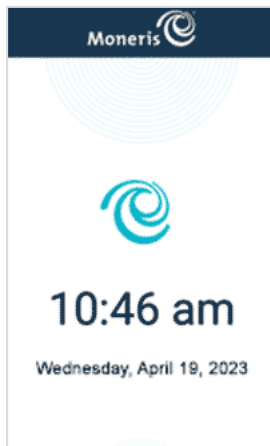
This section describes how to put your Moneris Go terminal into direct integration mode. You can send transactions from your point of sale (POS) software to the terminal for processing over your local area network. Your customers can use the terminal to securely enter their payment cards and leave tips.

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. Tokenization 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 eCommerce-Unified_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 PinPad - 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 eCommerce-Unified_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 eCommerce-Unified_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):

```

1. <?xml version="1.0"?>
2. <request>
3.   <store_id>gateway storeID</store_id>
4.   <api_token>API token</api_token>
5.   <res_tokenize_cc>
6.     <order_id>order ID from Go transaction</order_id>
7.     <txn_number>transaction number from Go transaction (ex.1017-0_140)</txn_number>
8.   </res_tokenize_cc>
9. </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.

Leveraging the eCommerce-Unified_API for future financial transaction with credential on file:

- payment_indicator: U
- payment_information:

Sample Code (Purchase with Token):

```
1. <?xml version="1.0"?>
2. <request>
3. <store_id>gateway storeID</store_id>
4. <api_token>API token</api_token>
5. <status_check>>false</status_check>
6. <res_purchase_cc>
7.     <data_key>Vault token</data_key>
8.     <order_id>example unique order ID</order_id>
9.     <amount>10.00</amount>
10.    <crypt_type>7</crypt_type>
11.    <cof_info>
12.        <payment_indicator>U</payment_indicator>
13.        <payment_information>2</payment_information>
14.    </cof_info>
15. </res_purchase_cc>
16. </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 transactionID must be sourced back to the original pre-auth transaction. Refer to the **eCommerce-Unified_API** or additional information associated with performing financial transactions via a token.

Managing tokens:

You may leverage the eCommerce-Unified_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 **eCommerce-Unified_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 PinPad - 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.

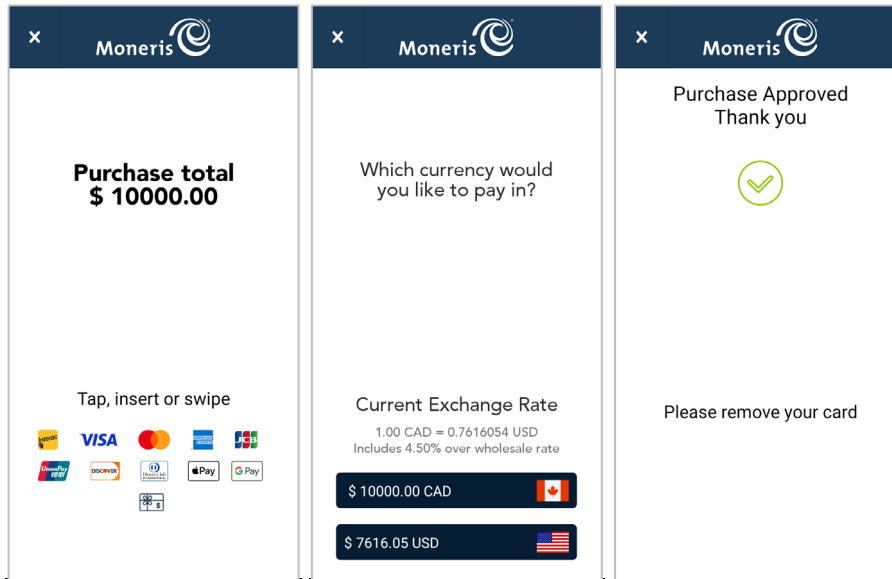
9.4 Supported Cards and Interface

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:



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": "MG01714061241002",
          "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
          3300 BLOOR ST W
          \r\n\r\n
          Purchase
          \r\n\r\n
          12:07:29\r\n\r\n
          VISA
          \r\n\r\n
          Ref#: 047- 0SCIA4H6VUP4FY2
          \r\n\r\n
          COFFEE STORE
          \r\n\r\n
          ETOBICOKE ON
          \r\n\r\n
          Apr 25,2024
          \r\n\r\n
          *****0004\r\n\r\n
          Entry: Chip (C)
          \r\n\r\n
          Auth#: B48739 ....."
        }
      ]
    }
  },
}
```

Sample DCC receipt:

```
----- TRANSACTION RECORD -----
                COFFEE STORE
                3300 BLOOR ST W
                ETOBICOKE    ON

                Purchase
Apr 25,2024                                12:07:29
VISA                                        *****0004
Entry: Chip (C)
Ref#: 047- 0SCIA4H6VUP4FY2
Auth#: B48739                               Response: 01-027
Order:                                       MGO1714061241002

Amount                                     $ 10 000.00

RATE* 1 CAD = 0.7616054 USD
*BASE RATE + 4.5%
SELECTED TRANS CURRENCY
[ ] CAD                                     10 000.00 CAD
[X] USD                                     7 616.05 USD

Total                                       7 616.05 USD

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

A0000000031010 VISA
TVR 80C0008000 TSI 7800

                Approved
                VERIFIED BY PIN

Important: Retain this copy for your record
```

10. Store and Forward (SAF) on Moneris Go Terminals

10.1 Processing Store and Forward (SAF)

This section describes how Store and forward (SAF) processing will allow credit transactions to be accepted at the terminal if connection to the Moneris host is unavailable. The transactions are processed later when a host connection is available.

The terminal will:

- selectively store 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 as deferred authorization transactions and all other cards brands as force post transactions. These transactions will be sent in the background. This will allow new transactions to be accepted while SAF transactions are being forwarded.

Transaction Types:

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

Card Types:

- Visa (Deferred Authorization)
- Mastercard (Deferred Authorization)
- American Express (Force Post)
- JCB (Force Post)
- Discover (Force Post)

Interface Types:

- Insert

Note: Void itself is not SAF. It is available for use while store and forward processing is in effect. While communication is down, Void can be used to delete a SAF 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 Store and Forward (SAF) Supported Devices

Device Model	Store and Forward (SAF)			
	Modes>>>	Standalone	App to App	Cloud
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 PinPad - PAX A35	N/A	N/A	N/A	N/A
Moneris Go Plus - DX8000	N/A	✓	N/A	✓
Moneris Go Slim - EX8000	N/A	✓	N/A	✓

10.3 Store and Forward Configuration

There are some limits associated with processing Store and Forward transactions which can be configured by your Client Consultant. These limits are as follows:

Maximum Transaction Amount: This corresponds to the maximum value of a transaction that can be processed in Store and Forward 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 Store and Forward mode.

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

Risk Limit: This corresponds to the maximum cumulative amount of Store and Forward transactions that are permitted in a single calendar day. If the Risk Limit is reached, the terminal will not allow additional SAF transactions to be performed.

Risk limit is reset provided the terminal has:

1. re-established communication with the host
2. cleared all stored transactions
3. processed a transaction online

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

10.4 Clearing Stored Transactions 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 SAF transactions are being sent in the background.

11. Installments by Visa on Moneris Go Terminals

Installments 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	N/A	N/A	N/A
Moneris Go Unattended - PAX IM30	N/A	N/A	N/A	N/A
Moneris Go PinPad - PAX A35	N/A	N/A	N/A	N/A
Moneris Go Plus - DX8000	✓	✓	✓	✓
Moneris Go Slim - EX8000	N/A	✓	✓	✓

11.2 Enabling Installments by Visa

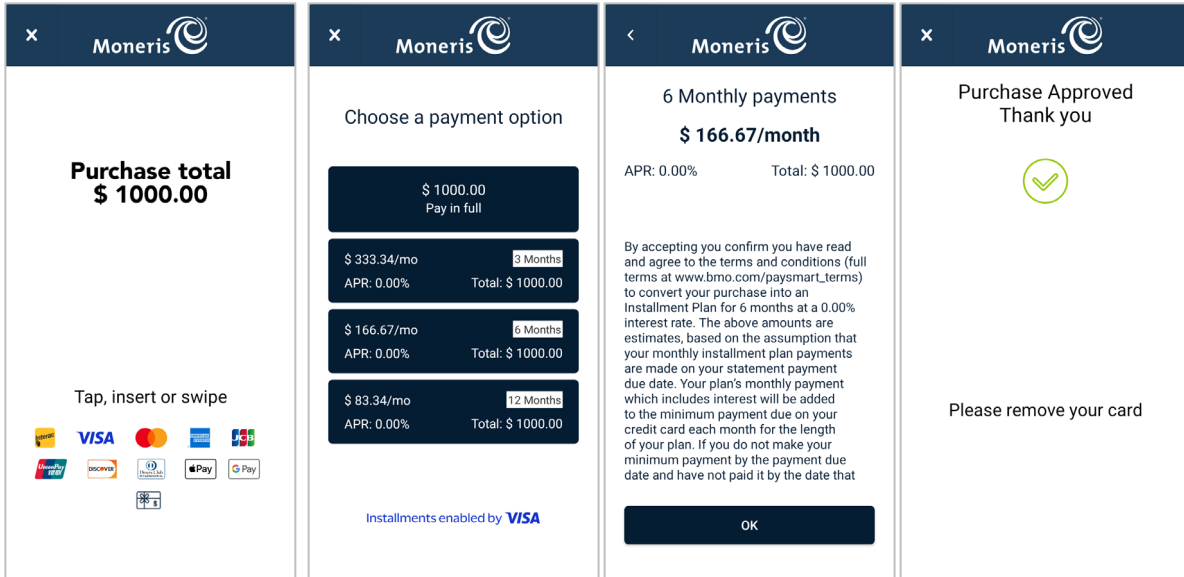
Your Client Consultant can help enable Installments by Visa on your Go Device. You should ensure your device is on the latest software version and sync your terminal with Moneris in order to start using Installments by Visa. To sync your terminal, refer to the Using Your Terminal section of the Moneris 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:



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 -----
      COFFEE SHOP
      3300 BLOOR ST
      ETOBICOKE ON
      M8X2X2

      Purchase
Apr 25,2024                               15:42:24
VISA                                       *****3925
Entry: Chip (C)
Ref#: 010- 0SCIK2O4R8MMDG0
Auth#: B44893                           Response: 01-027
Order:                                   MGO1714074135255

Amount                                     $ 1 000.00

Monthly Payment                           $ 166.67
Number of Installments                     6
APR                                         0.00%
Total Cost                                 $ 1 000.00

A0000000031010 VISA CREDIT
TVR 000020C000 TSI E800

      Approved
      Installments enabled by VISA

By accepting you confirm you
have read and agree to the terms
and conditions (full terms at
www.bmo.com/paysmart_terms) to
convert your purchase into an
Installment Plan for 6 months at
a 0.00% interest rate.

https://www.bmo.com/paysmart_ter
ms

Important: Retain this copy for your record
```

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.



NOTICE OF CONFIDENTIALITY. This document contains information that is the proprietary and confidential property of Moneris. The recipient agrees to maintain this information in confidence and not reproduce or otherwise disclose this information.

MONERIS, MONERIS BE PAYMENT READY & Design and MERCHANT DIRECT are registered trade-marks of Moneris Solutions Corporation. All other marks or registered trade-marks are the property of their respective owners.

© 2024 Moneris Solutions Corporation, 3300 Bloor Street West, Toronto, Ontario, M8X 2X2. All Rights Reserved. This document shall not wholly or in part, in any form or by any means, electronic, mechanical, including photocopying, be reproduced or transmitted without the authorized consent of Moneris Solutions Corporation. 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.

(MGo Int EN April 2024)