



API – EPXPay Semi-Integration

Public - Security Level 0

July 2021

REVISION HISTORY

Date	Version	Author(s)	Comments
5/5/16	2.0	C. Meaney	Reformatting
5/25/16	2.1	C. Meaney	Revised tip adjust description, included new code examples
9/7/16	2.2	C. Meaney	Updated Figure 1.
7/7/17	2.3	C. Meaney	Updated list of supported transaction types. Added remote load functionality.
6/25/20	2.4	M. Billips	Bring doc up to date with new features / functionality

CONFIDENTIALITY STATEMENT

This document contains confidential and proprietary information that belongs exclusively to Electronic Payment Exchange (EPX). Receipt of this document imposes the obligation on the recipient to protect the information from loss or disclosure to other parties.

This publication may not be reproduced or distributed for any purpose without the written permission of EPX.

© 2021 Electronic Payment Exchange. All rights reserved.

Contents

EPXPay overview	1
Process flow	1
EPXPay request	3
Tip functionality	3
Transaction types	3
Example request	4
EPXPay cardholder interaction	5
EPX payment gateway response	6
Batch settlement	7
Refunds / Credits	8
Semi-Integrated Response Codes	9
Additional Features	10
EPXPay Device to Prompt for Credit or PIN Debit Sale	10
Credit Card Details	10
Request:	10
Response	10
PIN Debit Details	11
Request:	11
Response	11
Cancel / Abort Request	12
Timeout Reversal	12
HTTPS Support	12
Semi-Integrated receipt response tags	13
Signature required	13
Quick service	13
EMV tags	13
Remote load functionality	14
Overview	14
Configuring download parameters	14
Invoking the remote load	14

List of Figures

Figure 1: EPXPay processing flow	1
Figure 2: Example Request	4
Figure 3: Example EPX Gateway Response	6

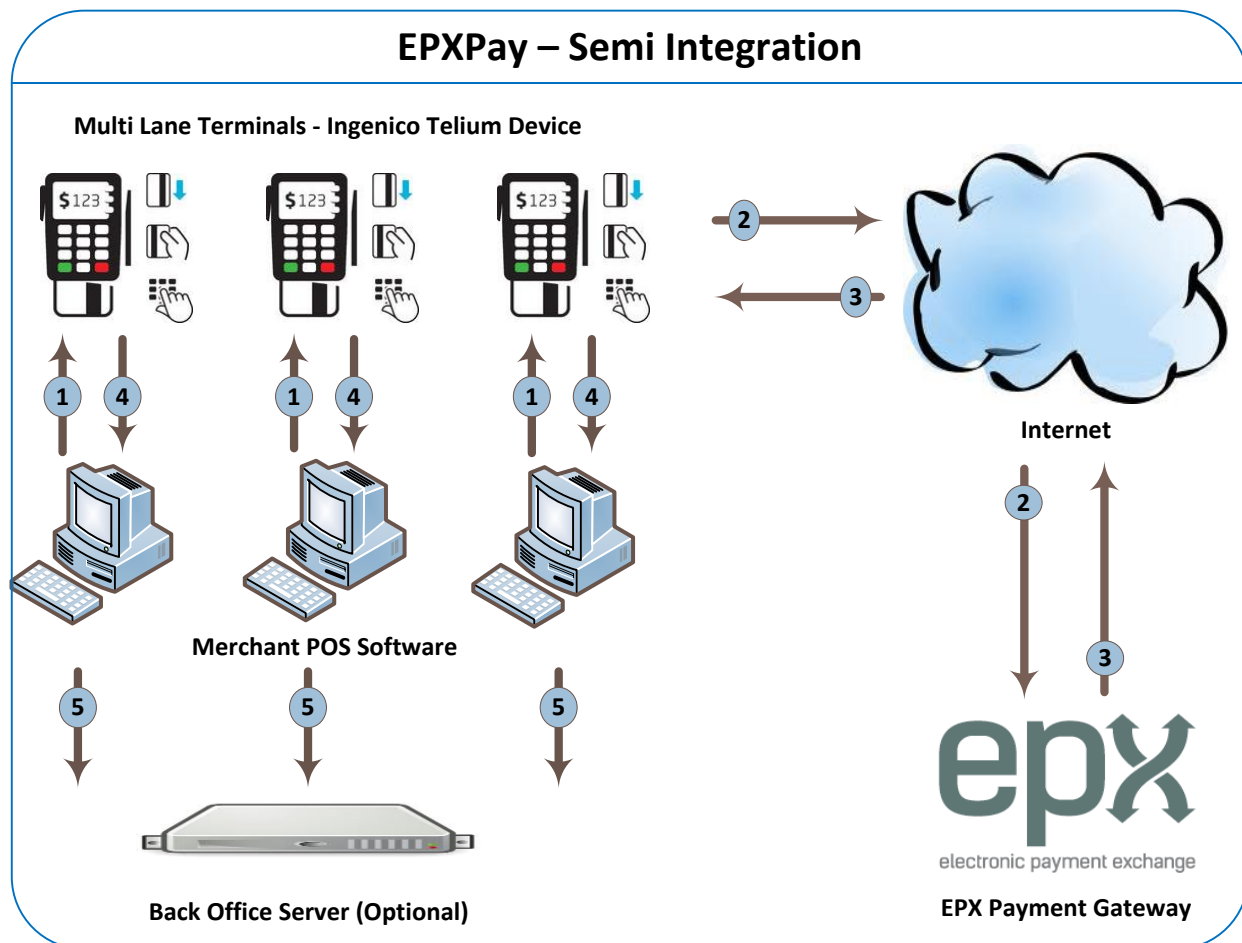
EPXPay overview

The EPXPay Terminal Application is a payment solution built to run on Ingenico Telium and Tetra branded terminal devices. When fully implemented, merchants can send various transaction types securely from the cardholder to EPX for real-time financial processing. EPXPay employs end-to-end encryption and EPX tokenization technology that assists the merchant in meeting all of their processing and PCI needs.

Process flow

Figure 1 shows an example of the EPXPay processing flow.

Figure 1: EPXPay processing flow



The following sequence describes Figure 1:

1. The merchant POS software interface sends the transaction request and amount via an HTTP POST to the terminal running EPXPay.

2. The cardholder completes the payment transaction using EPXPay on the terminal. EPXPay completes the transaction by communicating with the EMV Chip card and the EPX Gateway.
3. (3, 4, 5) The data returned from the HTTP/HTTPS POST contains the transaction response.

EPXPay request

The request from the merchant POS software interface needs to be sent as an HTTP POST with an XML payload. Required fields are the transaction type and requested amount. Optional fields can be passed including up to 10 user data fields. Please refer to the *EPX Retail Transaction Specs Document* and the *EPX Data Dictionary* for additional information.

Tip functionality

When the Tip functionality is enabled in the terminal profile, the device will prompt for the Tip Amount at the time of the sale. The EPXPay application will add the keyed Tip Amount to the total amount sent in the transaction authorization request, and the TIP_AMT is used as a reference field only. The response will contain the TIP_AMT tag with the associated tip amount that can be used to update the application records.

Example:

```
<TIP_AMT>1.00</TIP_AMT>
```

Alternatively, if the POS application will manage the tip, then this feature must be disabled within the device's terminal profile. For information about using the POS to manage the tip process, refer to *EPX Transaction Specs –Restaurant*.

NOTE: The Tip functionality is enabled in the terminal profile only upon client request.

Transaction types

Following are the supported transaction types for EPXPay Semi-Integration:

- CCR0 - Account Verification
- CCR1 - Retail Purchase Authorization & Capture transaction
- CCR2 - Retail Purchase Authorization Only transaction
- CCR4 - Retail Purchase Capture Only transaction
- CCR7 - Retail Purchase Authorization Reversal
- CCR8 - BRIC Storage
- CCR9 - Retail Return Capture Transaction
- CCRX - Retail Void transaction
- CCRZ - Retail Close Batch transaction
- DB00 - PIN Debit Sale

- DB01 - PIN Debit Return
- DB0V - PIN Debit Void / Reversal

Example request

Figure 2 shows an example request. In the figure, **bold** text denotes required fields.

Figure 2: Example Request

```
POST /HTTP/1.1
Content-Type: application/xml
Host: localhost:Port
Connection: close
Content-Length: 213

<DETAIL>
<TRAN_TYPE>CCR1</TRAN_TYPE>
<AMOUNT>1577.00</AMOUNT>
<BATCH_ID>20160523</BATCH_ID>
<TRAN_NBR>17</TRAN_NBR>
<INVOICE_NBR>11767</INVOICE_NBR >
<USER_DATA_1>Zipp 303 wheelset</USER_DATA_1>
</DETAIL>
```

NOTE: The syntax of the HTTP POST must be properly formatted and valid. The **Content-Length** must be the actual number of bytes of the payload. In the example above there are CR LF bytes after each line and count as separate bytes in the total content-length. Also, the payload (**<DETAIL>...</DETAIL>**) must be properly formatted XML.

The 4 part key (CUST_NBR , MERCH_NBR , DBA_NBR , TERMINAL_NBR) is not required in the request POST from the SI Client as the EPXPay SI device is programmed with that information and will include it with the transaction request to the Host.

EPXPay cardholder interaction

EPXPay displays the transaction amount on the terminal display. If the terminal supports multiple display lines, they are displayed as well.

EPXPay verifies with the cardholder that the authorization amount is correct, prompts for the desired method of payment*, and completes the authorization request via the EPX processing platform.

NOTE: EPXPay is designed to bypass the "method of payment" prompt, if the information is supplied from the POS Software in the HTTP Post.

EPX payment gateway response

The complete transaction response data from the EPX gateway is returned in the payload (<RESPONSE>...</RESPONSE>) of the HTTP/HTTPS POST response. If configured for verbose response, EPX includes all information in the original request, excluding security data and card data. The merchant can utilize the desired response data to update their software and database records. EPX offers the ability for enhanced data responses if needed (ENHANCED response data option). Please refer to the *EPX Data Dictionary* for additional information.

Figure 3 shows an example response.

Figure 3: Example EPX Gateway Response

```

HTTP/1.1 200 OK
Connection: close
Content-Type: application/xml
Content-Length: 1168

<RESPONSE>
<INVOICE_NBR>11767</INVOICE_NBR>
<USER_DATA_1>Zipp 303 wheelset</USER_DATA_1>
<CARD_ENT_METH>H</CARD_ENT_METH>
<SI_SIGNATURE_REQUIRED>Y</SI_SIGNATURE_REQUIRED>
<SI_QUICK_SERVICE>N</SI_QUICK_SERVICE>
<CURRENCY_CODE>840</CURRENCY_CODE>
<MSG_VERSION>003</MSG_VERSION>
<CUST_NBR>1234</CUST_NBR>
<MERCH_NBR>123456</MERCH_NBR>
<DBA_NBR>1</DBA_NBR>
<TERMINAL_NBR>1</TERMINAL_NBR>
<TRAN_TYPE>CCR1</TRAN_TYPE>
<BATCH_ID>20160523</BATCH_ID>
<TRAN_NBR>17</TRAN_NBR>
<LOCAL_DATE>052616</LOCAL_DATE>
<LOCAL_TIME>155837</LOCAL_TIME>
<AUTH_GUID>03QE4JFZV72VR03M0E9</AUTH_GUID>
<AUTH_RESP>00</AUTH_RESP>
<AUTH_CODE>000315</AUTH_CODE>
<AUTH_AVS> </AUTH_AVS>
<AUTH_RESP_TEXT>APPROVAL 000315</AUTH_RESP_TEXT>
<AUTH_CARD_TYPE>V</AUTH_CARD_TYPE>
<AUTH_TRAN_DATE_GMT>05/26/2016 07:58:36 PM</AUTH_TRAN_DATE_GMT>
<AUTH_AMOUNT_REQUESTED>1577.00</AUTH_AMOUNT_REQUESTED>
<AUTH_AMOUNT>1577.00</AUTH_AMOUNT>
<AUTH_CURRENCY_CODE>840</AUTH_CURRENCY_CODE>
<NETWORK_RESPONSE>00</NETWORK_RESPONSE>
<AUTH_CARD_F>Y</AUTH_CARD_F>
<AUTH_CARD_I>N</AUTH_CARD_I>
<AUTH_MASKED_ACCOUNT_NBR>*****1111</AUTH_MASKED_ACCOUNT_NBR>
<AMOUNT>1577.00</AMOUNT>
</RESPONSE>

```

Batch settlement

The following methods are available for batch settlement:

- **POS Software Interface:** The POS application is to initiate a batch close (CCRZ) command to the EPXPay SI device.
- **EPX / Host:** Settlement is driven by the EPX Host at a predefined time. The specific time is configured within the EPX 4 part key profile.
- **Back Office Server:** The back office server closes out all open batches by sending an XML batch file via SFTP to capture all transactions processed for the day. Please note that the original transaction will process as an authorization only (CCR2) and the XML batch file will capture (CCR4) the authorization transaction to complete the Sale with the very last transaction in the batch file as a batch close (CCRZ) command to close all of the capture records in the XML batch.

Refunds / Credits

Refunds can be processed with a BRIC/GUID, via chip read, or key-entered at the terminal. In all cases the card number is then encrypted.

Semi-Integrated Response Codes

The table below provides a list of the possible EPXPay semi-integration response codes that takes place between the integrated application and the EPXPay semi-integration device. These response codes are returned from the EPXPay semi-integration device and not the EPX Host / Platform.

Response Code (AUTH_RESP)	Response Message (AUTH_RESP_TEXT)	Response Description
S0	SI ERROR	HTTP Error, or any other error not listed below
	CHIP ERROR	Processing error during an EMV transaction
	CHIP DECLINED	Card decline resulted in an EMV or contactless transaction
	CANCELLED	Cardholder pressed cancelled during an EMV transaction
S1	SI TIMEOUT	User input timeout due to inactivity
S2	SI USER CANCEL	User cancel
S3	SI NOT SUPPORTED	Requested transaction type is not supported

Additional Features

EPXPay Device to Prompt for Credit or PIN Debit Sale

EPXPay SI offers the ability for the SI Client to send a Sale or Authorization command to the EPXPay SI device and prompt for the customer to select "Credit" for Credit Card or "Debit" for PIN Debit. This is only offered for Sale and Authorization Only transactions, in both cases the PIN Debit Sale DB00 would be sent as Debit since there is not an authorization only transaction available for PIN Debit. Since the customer is ultimately selecting the method of transaction, the integrated application will need to validate the values returned in TRAN_TYPE and AUTH_CARD_TYPE tags in the response to determine the type of transaction and card type that was processed.

Credit Card Details

The values of "SALE" or "AUTH" can be sent with the TRAN_TYPE tag in the request to the EPXPay SI device and the respective transaction type will be sent to the Host for processing and returned in the TRAN_TYPE tag in the response for the SI client to parse as needed. The below provides specifics on the relevant request and response tags surrounding this functionality.

Request:

- TRAN_TYPE
 - SALE = CCR1 Credit Card Sale (Authorization and Capture)
 - AUTH = CCR2 Credit Card Authorization Only

Example:

```
<TRAN_TYPE>SALE</TRAN_TYPE>
```

Response

- TRAN_TYPE
 - SALE = CCR1 Credit Card Sale (Authorization and Capture)
 - AUTH = CCR2 Credit Card Authorization Only

Example:

```
<TRAN_TYPE>CCR1</TRAN_TYPE>
```

- AUTH_CARD_TYPE

- Any value other than "O" (Other Debit) should be viewed as a credit card. Please reference AUTH_CARD_TYPE in the Data Dictionary for all possible values.

Example:

```
<AUTH_CARD_TYPE>V</AUTH_CARD_TYPE>
```

PIN Debit Details

The values of "SALE" or "AUTH" can be sent in the TRAN_TYPE tag in the request to the EPXPay SI device. Regardless of the type, a DB00 PIN Debit Sale will be sent to the Host for processing and returned in the TRAN_TYPE tag in the response for the SI client to parse as needed. The below provides specifics on the relevant request and response tags surrounding this functionality.

Request:

- TRAN_TYPE
 - SALE = DB00 PIN Debit Sale
 - AUTH = DB00 PIN Debit Sale

Example:

```
<TRAN_TYPE>AUTH</TRAN_TYPE>
```

Response

- TRAN_TYPE
 - SALE = DB00 PIN Debit Sale
 - AUTH = DB00 PIN Debit Sale

Example:

```
<TRAN_TYPE>DB00</TRAN_TYPE>
```

- AUTH_CARD_TYPE
 - The value of "O" (Other Debit) is viewed as a PIN debit card type. Please reference AUTH_CARD_TYPE in the Data Dictionary for all possible values.

Example:

```
<AUTH_CARD_TYPE>O</AUTH_CARD_TYPE>
```

Cancel / Abort Request

EPXPay SI offers the ability for the SI Client to send a cancel command to the device when prompting for payment causing the terminal to abort the previous transaction request. This eliminates the need for any manual interaction with the device to cancel the status of prompting for payment.

This feature is not intended to cancel a transaction if the customer has already provided a means of payment and the EPXPay SI device has sent the transaction to the Host for processing. In that scenario, once the response is received from the Host the SI Client is responsible for invoking a reversal or void to offset the transaction sent in error.

NOTE: The Cancel / Abort request will be ignored by the EPXPay SI device once card data input from the customer has started.

Example of Request Body:

```
<DETAIL>
<ABORT>1</ABORT>
</DETAIL>
```

Timeout Reversal

HTTPS Support

EPXPay SI offers the ability for the SI Client to send request POSTs over HTTPS. The EPXPay profile must be configured appropriately. If the integrator is interested in this feature, please contact the integration team directly.

Semi-Integrated receipt response tags

Signature required

The signature required tag is used for receipt printing. A value of "Y" indicates that the cardholder's signature is required for CVM (Cardholder Verification Method).

```
<SI_SIGNATURE_REQUIRED>Y</SI_SIGNATURE_REQUIRED>
```

Quick service

The quick service tag allows the client to recognize that the transaction has met the quick service qualification. A value of "Y" indicates the transaction meets quick service rules, where neither a customer signature nor a printed receipt is required.

```
<SI_QUICK_SERVICE>N</SI_QUICK_SERVICE>
```

EMV tags

The EMV tags shown in the example that follows must be printed on the customer receipt for *all* EMV-based transactions.

```
<SI_EMV_APP_LABEL>Visa Credit</SI_EMV_APP_LABEL>  
<SI_EMV_TVR>0800008000</SI_EMV_TVR>  
<SI_EMV_AID>A0000000031010</SI_EMV_AID>  
<SI_EMV_TSI>E800</SI_EMV_TSI>
```

Remote load functionality

Overview

For unattended kiosks, EPXPay provides the ability to invoke remote loads to update the EPXPay application. The remote load requires two steps, as follows:

- Configuring the download parameters
- Invoking the remote load

Configuring download parameters

The client application can configure EPXPay download parameters for an unattended EPXPay Semi-Integration device by sending the following commands to the device:

```
<DETAIL>
<TMS_CONFIG>1</TMS_CONFIG>
<TMS_DOMAIN_NAME>***.***.com</TMS_DOMAIN_NAME>
<TMS_PORT>****</TMS_PORT>
<TMS_IDENTIFIER>123456789</TMS_IDENTIFIER>
</DETAIL>
```

NOTE: The highlighted values will vary. Replace these values with the configuration data provided by EPX.

Invoking the remote load

Once the download parameters are configured, the client application must send the following command to the device to invoke the remote download:

```
<DETAIL>
<TMS_DOWNLOAD>1</TMS_DOWNLOAD>
</DETAIL>
```