NEXGO

SmartConnect API Documentation

NEXGO N5 SmartConnect API

IMPORTANT NOTICE

This document contains proprietary information of NEXGO, Inc. The information contained herein is confidential and its use is bound by the conditions of any and all binding Non-Disclosure Agreements. Reproduction or further distribution of any information contained within this document is strictly forbidden unless prior written consent has been obtained from NEXGO, Inc.

NO WARRANTY

No warranty although NEXGO has attempted to ensure the accuracy of the contents of this manual. This manual may contain errors or omissions. This manual is supplied “as-is,” without warranty of any kind, either expressed or implied, including the implied warranties of merchantability and fitness for a particular purpose.

LIMITED LIABILITY

Limited Liability in no event shall NEXGO be liable for any indirect, special, incidental, or consequential damages including damages for loss of business, profits, or the like, even if NEXGO or its representatives have been advised of the possibility of such damages.

Revision History

Date	Version	Author	Revision Description
12/20/2019	1.00	Hayden Morris	-Initial Version
2/12/2020	1.01	Hayden Morris	-Added transaction field tables / details.
3/16/2020	1.02	Hayden Morris	-Added **Auth** and **Capture** transaction types.
5/1/2020	1.03	Hayden Morris	-Added **getLastStatus** transaction type.
11/3/2020	1.04	Hayden Morris	-Add traceNo and transactionID changes -Add signature retrieval example -Add warning for Tip Adjustment (EVO) -Remove TSYS resultCodes
11/10/2020	1.05	Hayden Morris	-Add new transaction types: GiftActivate GiftReload GiftBalance GiftRedemption
11/11/2020	1.06	Hayden Morris	-Add new transaction types: EBTFSPurchase EBTCBPurchase EBTBalance EBTFSReturn EBTFSVoucherReturn EBTFSVoucher
11/17/2020	1.07	Hayden Morris	-Add Token payments (panDataToken) option in request/response messages.
11/20/2020	1.08	Hayden Morris	-Remove baseAmount from EBTBalance transaction. -Add snapAvailableBal and cashAvailableBal response fields to all EBT transactions.
11/23/2020	1.09	Hayden Morris	-Fix typo in **snapAvailableBal** field name.
12/1/2020	1.10	Hayden Morris	-Remove token support from PIN-enabled and unsupported transactions. EBTFSReturn EBTFSPurchase EBTCBPurchase EBTBalance
1/21/2021	1.11	Hayden Morris	Change EBTFSVoucher sample message auth code length to avoid errors on EVO.

Date	Version	Author	Revision Description
4/19/2021	1.12	Hayden Morris	-Make document processor agnostic. -Add TCP SmartConnect Sample Request + Response. -Add Supported Transaction Types section w/ table. -Fix misspelled fieldname (merchandId -> merchantId). -Add miscellaneous section.

Digitally signed
by Hayden Morris
Date: 2021.04.20
10:34:04 -07'00'

Table of Contents

1.0 Background

NEXGO has created various payment solutions for different platforms over the years. Whereas in the past, payment systems had generally been standalone devices with dedicated payment applications – as time has gone on, providing a method for 3rd parties to integrate their application with an already certified payment application has become more attractive.

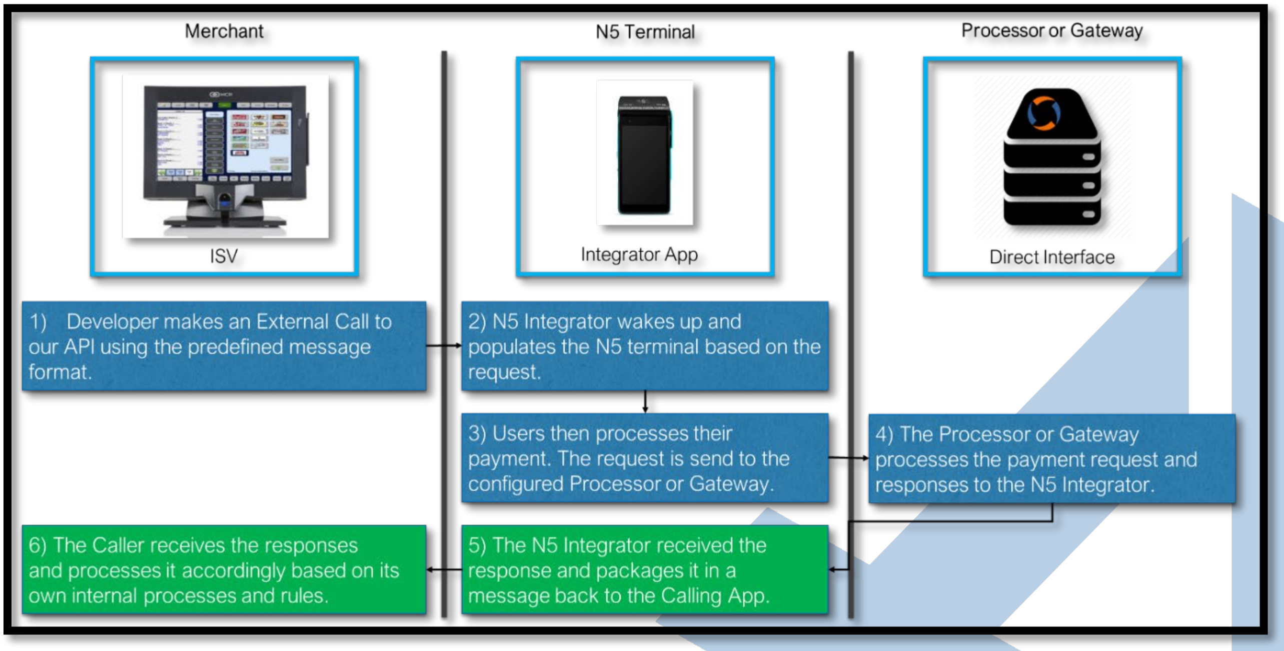
SmartConnect is the name of NEXGO's Semi-Integration API for allowing 3rd party applications to process transactions through our Integrator applications using either Android Intent or TCP communication.

In addition to increasing the ease of which integration can be done, SmartConnect also allows for customers who want to take payments to Integrate with our application without having to perform a certification themselves.

This document will go into detail on the flow, integration mode, message formats, and more for communicating and processing payments through the NEXGO Integrator application using the SmartConnect API.

1.1 Process Flowchart

The Developer creates an External call to the assigned N5 terminal. The Caller passes on instructions on what it is requesting to the N5 Integrator. The Integrator app on the N5 terminal then wakes up and takes over from there. When the payment process is finished it will return the results in JSON format to the Caller app.



1) Developer makes an External Call to our API using the predefined message format.

2) N5 Integrator wakes up and populates the N5 terminal based on the request.

3) Users then processes their payment. The request is send to the configured Processor or Gateway.

4) The Processor or Gateway processes the payment request and responses to the N5 Integrator.

5) The N5 Integrator received the response and packages it in a message back to the Calling App.

6) The Caller receives the responses and processes it accordingly based on its own internal processes and rules.

2.0 Semi-Integration Modes

The NEXGO SmartConnect API supports two methods for sending and receiving messages to and from the NEXGO Integrator application:

1. Android Intent
2. TCP/IP

2.1 Android Intent

The **Android Intent** integration mode is the preferred method for communicating with the Integrator application for a number of reasons.

Firstly, if the Integrator application is not running, sending a request to the Integrator using Android Intent will start the required activity and the Integrator will process the transaction normally. This contrasts with the TCP method, where if the Integrator application is not running, sending a request to the Integrator over TCP will be rejected since the port is not open listening for connections.

Secondly, it is generally more reliable than using TCP since communication takes place between processes instead of between devices on a network.

Lastly, the ease of passing data using an Intent and the extent of the types of data that can be passed (i.e. full objects) are additional reasons for using Android Intent over TCP for sending request messages to the NEXGO Integrator application.

See section ***4.1 Intent Example*** for example on how to use Intent for calling the Integrator application.

2.2 TCP/IP

The **TCP** integration mode is an additional method for communicating with the Integrator application other than using Android Intents.

While the Android Intent integration method is the preferred integration method, the TCP method has some advantages as well.

Firstly, while the Android Intent method only works on the same device on which the caller is located – the TCP method allows for receiving SmartConnect requests from either the same devices, or another device/application that has a routable connection to the device.

See section ***4.2 TCP Example*** for example on how to use TCP for calling the Integrator application.

3.0 Supported Transaction Types

The NEXGO Integrator application supports the following transaction types with SmartConnect:

1. **Sale** - Perform a Credit/Debit sale.
2. **Void** - Void an existing sale.
3. **Auth** (Pre-Auth) - Perform an Auth (Pre-Authorization) transaction.
4. **Capture** (Post-Auth) – Perform a Capture on an existing Auth (Pre-Authorization) transaction.
5. **Tip Adjustment**- Add or adjust a TIP on an existing authorized transaction.
6. **Return** - Refund an amount to a cardholder's card.

7. **Reprint** - Reprint the receipt from a previously approved transaction. (Retrieve JSON)
8. **BatchClose** - Settle the batch currently open on the terminal.
9. **getLastStatus** - Get the raw SmartConnect JSON response from last performed transaction.
10. **GiftActivate** - Load a gift card with an initial balance.
11. **GiftReload** - Reload an existing gift card with additional balance.
12. **GiftBalance** - Query the processor for the balance on an existing gift card.
13. **GiftRedemption** – Redeem value (i.e. Sale) from a gift card.
14. **EBTFSPurchase** – An EBT Food Stamp Purchase transaction.
15. **EBTCBPurchase** – An EBT Cash Benefit Purchase transaction.
16. **EBTBalance** – An EBT Balance Inquiry transaction.
17. **EBTFSReturn** – An EBT Food Stamp Return transaction.
18. **EBTFSVoucherReturn** – An EBT Food Stamp Voucher Clear Return transaction.
19. **EBTFSVoucher** – An EBT Food Stamp Voucher Clear transaction.

3.1 Sale Transaction

A **Sale** transaction is a typical Sale/Authorization transaction for a base amount, and optionally an additional Tip Amount.

Once the Sale request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.1.1 Sale Request

The **Sale** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Sale"
	panDataToken	String	**O**	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”
	amount	String	**M**	The base amount for the transaction.	"1.00"
	tip_amount	String	**O**	If tip amount not included, it will prompt the user to enter TIP on the screen.	"1.00"

	cash_back	String	O		"5.00"

Sample **Sale** request for performing a Sale for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
     "type":"Sale",
     "panDataToken":"03a0106a-d62a-4a47-9301-
     6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538",
    "amount":"1.00",
    "tip_amount":"0.00"
  }
}
```

3.1.2 Sale Response

The **Sale** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host’s ID of the transaction.
	transactionType	String	The type of transaction that was performed.	Credit Sale	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	tipAmount	String	The tip amount for the transaction that was proceessed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.

	cardDataEntry	String	The card entry method.	SWIPED / CHIP / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	emvTags[]	JSON Array	List of EMV tags returned.		Standard EMV tags and their respective values.
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Sale** response from an *APPROVED* Sale transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"Credit Sale",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "emvTags":[
      "9F06=A0000000031010",
      "5F2A=0840",
      "9B=F800",
      "84=A0000000031010",
      "4F=A0000000031010",
      "9F36=0042",
      "9F03=000000000000",
      "9C=00",
      "5F34=01",
      "9F39=05",
      "9F33=E020C8",
      "9F10=06010A03600000",
      "9A=200214",
      "DF78=500W103575",
      "9F26=6206484E62C59D54",
      "9F35=22",
      "9F02=000000000100",
      "9F27=40",
      "82=5C00",
      "9F34=1E0300",
      "9F1A=0840",
      "9F37=B6A9C847",
      "95=42C0008000"
     ],
    "cardHolder":"TEST CARD 01"
    }
}
```

3.2 Auth (Pre-Authorization) Transaction

An **Auth** (Pre-Authorization) transaction attempts to authorize a card for a given amount, and if the user has sufficient funds in their account – the system will place a 'hold' on the funds so the user cannot spend the funds.

The 'hold' on the funds will expire / drop-off in a short period of time. If you wish to actually charge the user for funds held by this **Auth** transaction – you must ***Capture*** the transaction before Settlement.

To ***Capture*** the initial **Auth** transaction – you must perform a ***Capture*** transaction (see section 3.3) by referencing the transactionID returned from the initial **Auth** transaction.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.2.1 Auth (Pre-Authorization) Request

The **Auth** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Auth"

	panDataToken	String	O	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”
	amount	String	M	The base amount for the transaction.	"1.00"
	tip_amount	String	O	If tip amount not included, it will prompt the user to enter TIP on the screen.	"1.00"
	cash_back	String	O		"5.00"

Sample **Auth** request for performing an Auth for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
     "type":"Auth",
     "panDataToken":"03a0106a-d62a-4a47-9301-
     6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538",
    "amount":"1.00",
    "tip_amount":"0.00"
  }
}
```

3.2.2 Auth (Pre-Authorization) Response

The **Auth** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host’s ID of the transaction.
	transactionType	String	The type of transaction that was performed.	Credit Sale	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Auth transactions should return with a base amount.
	tipAmount	String	The tip amount for the transaction that was processed.	0.00	If no tip, will return as 0.00.

	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardDataEntry	String	The card entry method.	SWIPED / CHIP / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the processor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	emvTags[]	JSON Array	List of EMV tags returned.		Standard EMV tags and their respective values.
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Auth** response from an *APPROVED* Auth transaction where an EMV Card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"Pre-Auth",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "emvTags":[
      "9F06=A0000000031010",
      "5F2A=0840",
      "9B=F800",
      "84=A0000000031010",
      "4F=A0000000031010",
      "9F36=0042",
      "9F03=000000000000",
      "9C=00",
      "5F34=01",
      "9F39=05",
      "9F33=E020C8",
      "9F10=06010A03600000",
      "9A=200214",
      "DF78=500W103575",
      "9F26=6206484E62C59D54",
      "9F35=22",
      "9F02=000000000100",
      "9F27=40",
      "82=5C00",
      "9F34=1E0300",
      "9F1A=0840",
      "9F37=B6A9C847",
      "95=42C0008000"
     ],
    "cardHolder":"TEST CARD 01"
    }
}
```

3.3 Capture (Post-Authorization) Transaction

A **Capture** transaction is used to finalize, or 'capture', a previously approved **Auth** (Pre-Authorization) transaction. You can either 'capture' the transaction for the same amount as originally authorized in the 'Auth' transaction – or you can 'capture' the transaction for a greater/lesser amount. You must pass in the transactionID of the original ***Auth*** transaction when performing the ***Capture*** transaction request.

Once the request has been made, the Integrator application will attempt to Capture the transaction for the new amount with the processor.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.3.1 Capture (Post-Authorization) Request

The **Capture** transaction takes the following fields in the request:

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"

	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Capture"
	amount	String	**M**	The base amount for the transaction.	"1.00"
	transaction_id	String	**O / M**	The transactionID of the previously approved 'Auth' transaction. Either transaction_id *or* trace_no must be included in the request. If both transaction_id and trace_no are included in the request – the application will use transaction_id.	"53ABB10B5BFF49C7A20B456AD7040BBD"
	trace_no	String	**O / M**	The traceNo of the previously approved 'Auth' transaction. Either transaction_id *or* trace_no must be included in the request.	"6"
	tip_amount	String	**O**	If tip amount not included, it will prompt the user to enter TIP on the screen.	"0.00"
	cash_back	String	**O**		"0.00"

Sample **Capture** request for capturing a previous *Auth* (traceNo **6**) for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Capture",
    "amount":"1.00",
    "tip_amount":"0.00",
    "transaction_id":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "trace_no":"6"

  }
}
```

3.3.2 Capture (Post-Authorization) Response

The **Capture** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	

	transactionType	String	The type of transaction that was performed.	Credit Sale	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Auth transactions should return with a base amount.
	tipAmount	String	The tip amount for the transaction that was processed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardDataEntry	String	The card entry method.	SWIPED / CHIP / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Capture** response from an *APPROVED* Capture transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "applicationVersion":"N08.011.066_49",
    "cardDataEntry":"CHIP READ  (I)",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardHolder":"TEST CARD 01",
    "batchID":"000001",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"7",
    "baseAmount":"1.00",
    "cardIssuer":"VISA",
    "transactionType":"Post-Auth",
    "referenceNumber":"000007623420",
    "merchantId":"000000001168579",
    "fwVersion":"v4.5.5",
    "cardNumber":"0010"
  }
}
```

3.4 EBTFSPurchase Transaction

An **EBTFSPurchase** (EBT Food Stamp Purchase) transaction is an EBT Food Stamp redemption.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.4.1 EBTFSPurchase Request

The **EBTFSPurchase** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTFSPurchase"
	amount	String	**M**	The base amount for the transaction.	"1.00"

Sample **EBTFSPurchase** request for performing a Purchase for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
     "type":"EBTFSPurchase",
     "amount":"1.00"
  }
}
```

3.4.2 EBTFSPurchase Response

The **EBTFSPurchase** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTFSPurchase	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	cardIssuer	String	The brand of the payment card used to process the transasction.	VISA	
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	"70.00"	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	"50.00"	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.

	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTFSPurchase** response from an *APPROVED* Purchase transaction where an MSR card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTFSPurchase",
    "baseAmount":"1.00"
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.5 EBTCBPurchase Transaction

An **EBTCBPurchase** (EBT Cash Benefit) transaction is an EBT Cash Benefit Purchase.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.5.1 EBTCBPurchase Request

The **EBTCBPurchase** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTCBPurchase"
	amount	String	**M**	The base amount for the transaction.	"1.00"
	cash_back	String	**O**	The amount of cash back to receive.	"5.00"

Sample **EBTCBPurchase** request for performing a Purchase for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"EBTCBPurchase",
    "amount":"1.00",
    "cash_back":"5.00"
  }
}
```

3.5.2 EBTCBPurchase Response

The **EBTCBPurchase** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTCBPurchase	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.

	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	“70.00”	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	“50.00”	
	cardDataEntry	String	The card entry method.	SWIPE / KEYED / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTCBPurchase** response from an *APPROVED* Purchase transaction where a MSR card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTCBPurchase",
    "baseAmount":"1.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.6 EBTBalance Transaction

An **EBTBalance** (EBT Balance) transaction is transaction to query the funds available on an EBT account.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.6.1 EBTBalance Request

The **EBTBalance** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTBalance"

Sample **EBTBalance** request:

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"EBTBalance"
  }
}
```

3.6.2 EBTBalance Response

The **EBTBalance** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTBalance	
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	"70.00"	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	"50.00"	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTBalance** response from an *APPROVED* Balance transaction where a MSR card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTBalance",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.7 EBTFSReturn Transaction

An **EBTFSReturn** (EBT Food Stamp Return) transaction is a transaction to return funds to an EBT account.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.7.1 EBTFSReturn Request

The **EBTFSReturn** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTFSReturn"
	amount	String	**M**	The base amount to return for the transaction.	"1.00"

Sample **EBTFSReturn** request for performing a Return for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"EBTFSReturn",
    "amount":"1.00"
  }
}
```

3.7.2 EBTFSReturn Response

The **EBTFSReturn** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTFSReturn	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	"70.00"	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	"50.00"	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.

	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTFSReturn** response from an *APPROVED* Return transaction where a MSR card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTFSReturn",
    "baseAmount":"1.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.8 EBTFSVoucherReturn Transaction

An **EBTFSVoucherReturn** (EBT Voucher Return) transaction is an EBT Food Stamp Voucher Clear Return.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.8.1 EBTFSVoucherReturn Request

The **EBTFSVoucherReturn** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTFSVoucherReturn"
	amount	String	**M**	The base amount to return for the transaction.	"1.00"
	panDataToken	String	**O**	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”
	voucher_number	String	**M**	The voucher number.	"123456"
	auth_number	String	**M**	The authorization number for transaction.	"123456789"

Sample **EBTFSVoucherReturn** request for performing a Return for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"EBTFSVoucherReturn",
    "amount":"1.00",
    "voucher_number":"123456",
    "auth_number":"123456789"
  }
}
```

3.8.2 EBTFSVoucherReturn Response

The **EBTFSVoucherReturn** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.

	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTFSVoucherReturn	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	"70.00"	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	"50.00"	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTFSVoucherReturn** response from an *APPROVED* Return transaction where a MSR card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTFSVoucherReturn",
    "baseAmount":"1.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.9 EBTFSVoucher Transaction

An **EBTFSVoucher** (EBT Electronic Voucher) transaction is an EBT Food Stamp Voucher Clear.

Once the request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.9.1 EBTFSVoucher Request

The **EBTFSVoucher** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"EBTFSVoucher"
	amount	String	**M**	The base amount for the transaction.	"1.00"
	voucher_number	String	**M**	The voucher number.	"123456"
	auth_number	String	**M**	The authorization number for transaction.	"123456"

Sample **EBTFSVoucherReturn** request for performing a Sale for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"EBTFSVoucher",
    "amount":"1.00",
    "voucher_number":"123456",
    "auth_number":"123456"

  }
}
```

3.9.2 EBTFSVoucher Response

The **EBTFSVoucher** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	EBTFSVoucher	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	snapAvailableBal	String	The remaining SNAP balance on the card after the transaction.	"70.00"	
	cashAvailableBal	String	The remaining cash balance on the card after the transaction.	"50.00"	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.

	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **EBTFSVoucher** response from an *APPROVED* Sale transaction where KEYED entry was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"EBTFSVoucher",
    "baseAmount":"1.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "snapAvailableBal":"70.00",
    "cashAvailableBal":"50.00",
    "cardIssuer":"EBT",
    "cardDataEntry":"KEYED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.10 Void Transaction

A **Void** transaction voids/invalidates an *APPROVED* transaction in the current batch using the original transactionID of the transaction.

Once a Void request has been made, the Integrator will attempt to void the transaction in the batch.

After the Integrator processes the transaction with the processor, the result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

Note: You can only void a transaction that has not been settled yet. Once a transaction is settled (the batch is closed), the transaction can no longer be voided – and instead a **Return** should be used.

3.10.1 Void Request

The **Void** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Void"
	transaction_id	String	**O / M**	The transactionID of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request. If both transaction_id and trace_no are included in the request – the application will use transaction_id.	"53ABB10B5BFF49C7A20B456AD7040BBD"
	trace_no	String	**O / M**	The traceNo of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request.	"6"

Sample **Void** request for Voiding transactionID **6** from the current batch :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Void",
    "transaction_id":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "trace_no":"6"
  }
}
```

3.10.2 Void Response

The **Void** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)

packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionType	String	The type of transaction that was performed.	Credit Refund	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	tipAmount	String	The tip amount for the transaction that was proceessed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	cardIssuer	String	The brand of the payment card used to process the transasction.	VISA	
	cardDataEntry	String	The card entry method.	SWIPED / CHIP READ	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Void** response from a successful Void transaction :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"7",
    "transactionType":"Void",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001560420",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
  }
}
```

3.11 Return Transaction

A **Return** transaction is essentially 'refunding' money to a card. Whereas a Void transaction can only happen while the batch is still open, a **Return** can be performed at any time. Keep in mind that while Voiding a transaction actually drops the transaction from the batch – a return transaction issues money *back to a card*.

Once the Return request has been made, the Integrator application will display the screen to the user indicating the Return amount – and prompting the user to either Swipe / Insert / TAP / Manually enter their payment card. The *Return* will be issued to the card swiped by the user.

After the Integrator processes the transaction with the processor, the result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.11.1 Return Request

The **Return** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Return"
	panDataToken	String	**O**	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”

	amount	String	**M**	The amount to refund to the card.	"1.00"

Sample **Return** request for a $1.00 Return :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Return",
    "amount":"1.00"
  }
}
```

3.11.2 Return Response

The **Return** transaction will return the following fields in the response:

Section	**Field**	**Type**	**Description**	**Example**	**Note(s)**
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	Credit Refund	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	tipAmount	String	The tip amount for the transaction that was proceessed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardIssuer	String	The brand of the payment card used to process the transasction.	VISA	

	cardDataEntry	String	The card entry method.	SWIPED / KEYED / CHIP / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	emvTags[]	JSON Array	List of EMV tags returned.		Standard EMV tags and their respective values.
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Return** response from a Return transaction :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"7",
    "transactionType":"Credit Refund",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001596620",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "emvTags":[
      "9F06=A0000000031010",
      "5F2A=0840",
      "9B=E800",
      "84=A0000000031010",
      "4F=A0000000031010",
      "9F36=0043",
      "9F03=000000000000",
      "9C=20",
      "5F34=01",
      "9F39=05",
      "9F33=E020C8",
      "9F10=06010A03600400",
      "9A=200214",
      "DF78=500W103575",
      "9F26=1DBE67B5E052A204",
      "9F35=22",
      "9F02=000000000100",
      "9F27=40",
      "82=5C00",
      "9F34=1E0300",
      "9F1A=0840",
      "9F37=50535ADA",
      "95=42C0008000"
    ],
   "cardHolder":"TEST CARD 01"
  }
}
```

3.12 Tip Adjustment Transaction

Note: Tip Adjustment is not available for the EVO application as it is only certified for Retail environment. To work around this, you can instead use the Auth + Capture transactions to function as a tip feature. For instance, you can ‘Auth’ the initial sale for $X.XX, and then at a later time ‘Capture’ the transaction for the amount plus the desired tip.

A **Tip Adjustment** transaction is used to adjust the tip of a previously approved transaction in the current open batch. You can use this to either add a tip to a transaction that had previously no tip added, or can be used to adjust an existing tip amount for a previously approved transaction.

Once the **Tip Adjustment** request has been made, the Integrator application will attempt to adjust the tip of the transaction specified in the request’s “transaction_id” field if it exists.

After the Integrator processes the transaction with the processor, the result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

Note: Tip adjustments are performed offline and are uploaded to the host during Settlement. You can *only* adjust a tip while the batch is still open.

3.12.1 Tip Adjustment Request

The **Tip Adjustment** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true

payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Tip Adjustment"
	tip_amount	String	**M**	The new tip amount for the transaction.	"1.00"
	amount	String	**M**	The amount to refund to the card.	"1.00"
	transaction_id	String	**O / M**	The transactionID of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request. If both transaction_id and trace_no are included in the request – the application will use transaction_id.	"53ABB10B5BFF49C7A20B456AD7040BBD"
	trace_no	String	**O / M**	The traceNo of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request.	"6"

Sample **Tip Adjustment** request for adjusting the tip to $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Tip Adjustment",
    "amount":"1.00",
    "tip_amount":"1.00",
    "transaction_id":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "trace_no":"6"
  }
}
```

3.12.2 Tip Adjustment Response

The **Tip Adjustment** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionType	String	The type of transaction that was performed.	Credit Refund	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.

	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	tipAmount	String	The tip amount for the transaction that was processed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	cardIssuer	String	The brand of the payment card used to process the transasction.	VISA	
	cardDataEntry	String	The card entry method.	SWIPED / CHIP / KEYED	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Tip Adjustment** response from a successful Tip Adjustment transaction :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"7",
    "transactionType":"Add Tip",
    "baseAmount":"1.00",
    "tipAmount":"1.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001740120",
    "authorizationCode":"755696",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
  }
}
```

3.13 Gift Redemption Transaction

A **GiftRedemption** transaction is a transaction to redeem previously loaded value from a gift card provided by a customer. A gift card must be activated with a balance in order to be able to be redeemed successfully.

Once the Redemption request has been made, the Integrator application will display the screen to the user indicating the transaction amount – and prompting the user to Swipe / Manually enter their gift card number.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.13.1 GiftRedemption Request

The **GiftRedemption** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"GiftRedemption"
	amount	String	**M**	The base amount for the transaction.	"1.00"
	panDataToken	String	**O**	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”
	tip_amount	String	**O**	If tip amount not included, it will prompt the user to enter TIP on the screen.	"1.00"

Sample **GiftRedemption** request for performing a Sale for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"GiftRedemption",
    "amount":"1.00",
    "tip_amount":"0.00"
  }
}
```

3.13.2 GiftRedemption Response

The **GiftRedemption** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	GiftRedemption	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	tipAmount	String	The tip amount for the transaction that was proceessed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	feeAmount	String	The fee charged for the transaction.	0.00	Fee charged for the transaction on top of the subtotal transaction amount.
	cardNumber	String	The last 4 digits of the PAN used to process the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	newBalance	String	The new balance on the card after the transaction was processed.	0.00	After performing a transaction, the remaining balance available on the card.
	previousBalance	String	The previous balance on the card before this transaction was processed.	0.00	
	isPartialApproval	String	Whether or not the processed transaction was only partially approved.	False	If a transaction is processed for more than was available on the card, it can only approve partially.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardIssuer	String	The brand of the payment card used to process the transasction.	GIFT	

	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **GiftRedemption** response from an *APPROVED* Sale transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"GiftRedemption",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "feeAmount":"0.00",
    "cardNumber":"0010",
    "newBalance":"0.00",
    "previousBalance":"0.00",
    "isPartialApproval":"false",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"GIFT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.14 GiftActivate Transaction

A **GiftActivate** transaction is a transaction to activate a gift card for use. This entails loading a gift card with an initial balance, and paying for the activation with a payment card (i.e. Credit Sale).

Once the Activate request has been made, the Integrator application will display the screen to the user indicating the initial load amount – and prompting the user to Swipe / Manually enter their Credit card to pay for the aforementioned initial loading.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.14.1 GiftActivate Request

The **GiftActivate** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"GiftActivate"
	amount	String	**M**	The base amount for the transaction. This is the amount the customer wishes to 'activate' to the card.	"1.00"

Sample **GiftActivate** request for performing an Activation for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"GiftActivate",
    "amount":"1.00"
  }
}
```

3.14.2 GiftActivate Response

The **GiftActivate** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	

	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7 A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host’s ID of the transaction.
	transactionType	String	The type of transaction that was performed.	GiftActivate	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	feeAmount	String	The fee charged for the transaction.	0.00	Fee charged for the transaction on top of the subtotal transaction amount.
	cardNumber	String	The last 4 digits of the PAN used to process the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	newBalance	String	The new balance on the card after the transaction was processed.	0.00	After performing a transaction, the remaining balance available on the card.
	isPartialApproval	String	Whether or not the processed transaction was only partially approved.	False	If a transaction is processed for more than was available on the card, it can only approve partially.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardIssuer	String	The brand of the payment card used to process the transaction.	VISA	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **GiftActivate** response from an *APPROVED* Sale transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"GiftActivate",
    "baseAmount":"1.00",
    "resultCode":"00",
    "feeAmount":"0.00",
    "cardNumber":"0010",
    "newBalance":"0.00",
    "isPartialApproval":"false",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"VISA",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.15 Gift Reload Transaction

A **GiftReload** transaction is a transaction to reload a gift card for use. This entails reloading a gift card with additional balance, and paying for the reload with a payment card (i.e. Credit Sale).

Once the GiftReload request has been made, the Integrator application will display the screen to the user indicating the reload amount – and prompting the user to Swipe / Manually enter their Credit card to pay for the aforementioned reload.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.15.1 GiftReload Request

The **GiftReload** transaction takes the following fields in the request:

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true

payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"GiftReload"
	panDataToken	String	**O**	The payment token that can be used in lieu of a card swipe.	“03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538”
	amount	String	**M**	The base amount for the transaction.	"1.00"

Sample **Gift Reload** request for performing a Sale for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"GiftReload",
    "amount":"1.00"
  }
}
```

3.15.2 GiftReload Response

The **GiftReload** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host’s ID of the transaction.
	transactionType	String	The type of transaction that was performed.	GiftReload	
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	feeAmount	String	The fee charged for the transaction.	0.00	Fee charged for the transaction on top of the subtotal transaction amount.
	cardNumber	String	The last 4 digits of the PAN used to process the payment.	0010	Last 4 digits of the payment card used to process the transaction.

	newBalance	String	The new balance on the card after the transaction was processed.	0.00	After performing a transaction, the remaining balance available on the card.
	previousBalance	String	The previous balance on the card before this transaction was processed.	0.00	
	isPartialApproval	String	Whether or not the processed transaction was only partially approved.	False	If a transaction is processed for more than was available on the card, it can only approve partially.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardIssuer	String	The brand of the payment card used to process the transasction.	GIFT	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED / TOKEN	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **GiftReload** response from an *APPROVED* Sale transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"GiftReload",
    "baseAmount":"1.00",
    "resultCode":"00",
    "feeAmount":"0.00",
    "cardNumber":"0010",
    "newBalance":"0.00",
    "previousBalance":"0.00",
    "isPartialApproval":"false",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"VISA",
    "cardDataEntry":"SWIPE",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.16 GiftBalance Transaction

A **GiftBalance** transaction is a transaction check the current balance on a gift card.

Once the GiftBalance request has been made, the Integrator application will display the swipe screen – prompting the user to Swipe / Manually enter the gift card they wish to check the balance on.

After the Integrator processes the transaction with the processor, the authorization result will be returned in a JSON packet to the Caller (device/application the initial request originated from).

3.16.1 GiftBalance Request

The **GiftBalance** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"GiftBalance"

Sample **GiftBalance** request for performing a Sale for $1.00 :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"GiftBalance"
  }
}
```

3.16.2 GiftBalance Response

The **GiftBalance** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host's ID of the transaction.
	transactionType	String	The type of transaction that was performed.	GiftBalance	
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	newBalance	String	The new balance on the card after the transaction was processed.	0.00	After performing a transaction, the remaining balance available on the card.
	panDataToken	String	The PaymentAccountDataToken that can be used for future payments.	03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538	Payment tokens are used to perform transactions or adjustments without the card present in the future.
	cardIssuer	String	The brand of the payment card used to process the transasction.	GIFT	
	cardDataEntry	String	The card entry method.	SWIPED / KEYED	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	authorizationCode	String	The authorization code returned by the processor for this transaction.	556475	All approved Sale transactions will have an authorization code.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **GiftBalance** response from an *APPROVED* Sale transaction where an EMV card was used as payment :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"6",
    "transactionType":"GiftBalance",
    "resultCode":"00",
    "cardNumber":"0010",
    "newBalance":"0.00",
    "panDataToken":"03a0106a-d62a-4a47-9301-6060c1774e26200fca8b-a7bd-
4044-by6e-3dab4473d538",
    "cardIssuer":"GIFT",
    "cardDataEntry":"SWIPED",
    "referenceNumber":"000001560420",
    "authorizationCode":"556475",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
    }
}
```

3.17 Reprint Transaction

A **Reprint** transaction is used to either reprint an existing transaction if receipt boolean is true in the request message, or to return a JSON summary of the specified transaction if the receipt Boolean is false in the request message.

After the Integrator processes the request, it will either reprint the receipt if receipt Boolean is true; otherwise will not reprint but will only return the original authorization request in the JSON response.

Note: Reprint can only be performed on transactions that are in the current batch.

3.17.1 Reprint Request

The **Reprint** transaction takes the following fields in the request**:**

Section	**Field**	**Type**	**Required**	**Description**	**Example**
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"Reprint"
	transaction_id	String	**O / M**	The transactionID of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request.	“53ABB10B5BFF49C7A20B456AD7040BBD”

				If both transaction_id and trace_no are included in the request – the application will use transaction_id.	
	trace_no	String	**O / M**	The traceNo of the previously approved transaction. Either transaction_id *or* trace_no must be included in the request.	“6”

Sample **Reprint** request for reprinting the receipt for transactionID **5** :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Reprint",
    "transaction_id":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "trace_no":"6"
  }
}
```

3.17.2 Reprint Response

The **Reprint** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionID	String	The transactionID of this transaction.	53ABB10B5BFF49C7A20B456AD7040BBD	You need to use this transactionID for modification transactions like 'Tip Adjustment' to reference the transaction.
	traceNo	String	The traceNo of this transaction. The traceNo is the local ID of the transaction.	1	The traceNo is the local ID of the transaction. The transactionID is the host’s ID of the transaction.
	baseAmount	String	The base amount for the transaction that was processed (not including any additional fields such as tip).	5.00	All Sale transactions should return with the base amount.
	tipAmount	String	The tip amount for the transaction that was proceessed.	0.00	If no tip, will return as 0.00.
	cashbackAmount	String	The cashback amount for the transaction that was processed.	0.00	If none, will return as 0.00.
	resultCode	String	The response code from the Integrator.		Approved should be **00**; check appendix for full list.
	cardNumber	String	The last 4 digits of the PAN used to processed the payment.	0010	Last 4 digits of the payment card used to process the transaction.
	cardIssuer	String	The brand of the payment card used to process the transasction.	VISA	

	cardDataEntry	String	The card entry method.	SWIPED / CHIP READ	
	referenceNumber	String	The reference number returned for the host that can be used to identify the transaction when troubleshooting with the processor.	000001560420	
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	cardHolder	String	If present, the name returned from the card of the cardholder.	John Doe	

Sample **Reprint** response from a Reprint transaction :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000005",
    "transactionID":"53ABB10B5BFF49C7A20B456AD7040BBD",
    "traceNo":"7",
    "transactionType":"Reprint",
    "baseAmount":"1.00",
    "tipAmount":"0.00",
    "cashbackAmount":"0.00",
    "resultCode":"00",
    "cardNumber":"0010",
    "cardIssuer":"VISA",
    "cardDataEntry":"CHIP READ  (I)",
    "referenceNumber":"000001560420",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "cardHolder":"TEST CARD 01"
  }
}
```

3.18 getLastStatus Transaction

A **getLastStatus** transaction is used to get the original SmartConnect JSON response from the last transaction.

This is useful for the following scenario:

1. When Integrator application performs a transaction, but the calling application either crashes before getting the response – or is unable to parse the response.

2. When Integrator application performs a transaction, but some action like battery dying prevents calling application from getting the Integrator response.

In both cases above, a transaction was processed by Integrator, and the card was charged – but the response want not successfully received/parsed by the calling application. In this scenario, the 'calling' application does *not* know the status of the transaction.

Since the response was either not received/parsed – you would be unable to check the transaction using the transactionId since it would not be know.

A quick way to check the status of the last performed transaction is to call the **getLastStatus** SmartConnect API. This will return the raw response from the last transaction performed.

3.18.1 getLastStatus Request

The **getLastStatus** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"getLastStatus"

Sample **getLastStatus** request for retrieving the response of the last transaction :

```
{
  "action":{
    "processor":"EVO"
  },
  "payment":{
    "type":"getLastStatus"
  }
}
```

3.18.2 getLastStatus Response

The **getLastStatus** transaction will return the same fields as the transaction type it is denoting.

For example, if you performed a ***getLastStatus*** transaction on a terminal where the last transaction was a '**Sale**' – then the response will be the same response as you would expect from the **Sale** transaction.

Alternatively, if you performed a ***getLastStatus*** transaction on a terminal where the last transaction was a '**BatchClose**' – then the response will be the same response format/fields as you would expect from the **BatchClose** transaction.

3.19 BatchClose (Settlement) Transaction

A **BatchClose** transaction is used to Settle the current batch with the processor. If not utilizing the Auto-Batch feature (can be configured in the application configuration either manually on the device or through NEXGO's TAP platform) you must manually settle the batch before the funds will be deposited into your merchant account.

After the Integrator processes the request, it will attempt to settle the batch with the host. If the transactions on the device do not match the host's record, the terminal will re-upload all transactions to the host to bring the batch into balance.

After BatchClose has completed, the terminal will print out a report and return the status to the terminal in a JSON message.

3.19.1 BatchClose (Settlement) Request

The **BatchClose** transaction takes the following fields in the request**:**

Section	Field	Type	Required	Description	Example
action	processor	String	**M**	Processor (i.e. EVO)	"EVO"
	receipt	boolean	**M**	Whether the app should print a receipt or not.	true
payment	type	String	**M**	The transaction type (i.e. Sale / Void / etc.)	"BatchClose"

Sample **BatchClose** request for closing the batch :

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"BatchClose"
  }
}
```

3.19.2 BatchClose (Settlement) Response

The **BatchClose** transaction will return the following fields in the response:

Section	Field	Type	Description	Example	Note(s)
packetData	batchID	String	The batch number this transaction was performed on.	000004	
	transactionType	String	The type of transaction that was performed.	BatchClose	

	resultCode	String	The response code from the Integrator.	00	Approved should be **00**; check appendix for full list.
	merchantId	String	The merchantId of the merchant account being used to process the transaction with the prcocessor.	000000002118579	
	applicationVersion	String	The version of the Integrator application on the machine used to process the payment.	N08.011.066_49	NEXGO Integrator application version.
	fwVersion	String	The firmware version of the device the applications are running on.	v4.5.5	
	settlementTime	String	The date/time settlement completed.	2020/02/14 19:16:01	Local device time.
	creditSaleCount	String	The number of credit sale transactions in the batch that was settled.	6	
	creditSaleAmount	String	The sum of all the sale transaction amounts in the batch that was settled.	20.00	
	creditReturnCount	String	The number of credit return transactions in the batch that was settled.	1	
	creditReturnAmount	String	The sum of all the return transaction amounts in the batch that was settled.	1.00	
	debitSaleCount	String	The number of debit sale transactions in the batch that was settled.	0	
	debitSaleAmount	String	The sum of all the debit sale transaction amounts in the batch that was settled.	0.00	
	debitReturnCount	String	The number of debit return transactions in the batch that was settled.	0	
	debitReturnAmount	String	The sum of all the debit return transaction amounts in the batch that was settled.	0.00	

Sample **BatchClose** response from a BatchClose transaction :

```
{
  "packetID":"XCRT",
  "packetData":{
    "batchID":"000004",
    "transactionType":"BatchClose",
    "resultCode":"00",
    "merchantId":"000000001168579",
    "applicationVersion":"N08.011.066_49",
    "fwVersion":"v4.5.5",
    "settlementTime":"2020/02/14 19:16:01",
    "creditSaleCount":"6",
    "creditSaleAmount":"20.00",
    "creditReturnCount":"1",
    "creditReturnAmount":"1.00",
    "debitSaleCount":"0",
    "debitSaleAmount":"0.00",
    "debitReturnCount":"0",
    "debitReturnAmount":"0.00"
  }
}
```

4.0 Intent Example

The **Android Intent** integration mode is the preferred method for communicating with the Integrator application for a number of reasons.

4.1 Intent Example Request

To send a transaction request message to the Integrator using an ‘Intent’, we need to specify the package to call, include the request message in the intent, and then send the Intent while listening for the response:

```
Intent intent = new Intent();                          //Initialize the ‘intent’ object
intent.setAction("android.intent.action.Integrator");  //Specify the package to call with Intent
intent.putExtra(“Input",message);                      //Put the request message into the Intent
startActivityForResult(intent, REQUEST_CODE)           //Call the Intent, listen for the result
```

Note: ***message*** should be the full JSON request message.

4.2 Intent Example Response

Since the Intent is invoked using the **startActivityForResult(…)** function, when the Integrator action is completed, the **startActivityForResult** callback from the calling application will be executed. We can expect to receive a response back from the Integrator denoting the result of the transaction.

The response will be contained within the ‘data’ Intent object:

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
```

For example, we can get the ‘Transaction Data’ by parsing the ‘transdata’ field in the ‘data’ Intent object:

```
String response = data.getStringExtra("transdata");
```

5.0 TCP Example

The **TCP** integration mode is an alternative method for communicating with the Integrator application. While using Intents are the preferred method for communicating with the Nexgo Integrator application, certain development requirements may require use of the TCP method.

5.1.1 TCP Connection Requirements

Sending a TCP request to the Integrator has certain inherit requirements for successful operation:

1. The device, service, or application *sending* the Request *to* the Integrator must have a routable connection to the Nexgo payment device.
2. The Integrator application must be installed and running on the Nexgo payment device.

If any of the above is not met, communicating with the Integrator may become prone to issues.

5.2 TCP Example Request

To send a transaction request message to the Integrator using a 'TCP' connection, we need to create a socket connection to the Nexgo payment device running the Integrator application. We will use the socket connection to send the SmartConnect JSON request to the Integrator for processing. The default port the applications listen on is **8765**.

Here is a simple example for sending the SmartConnect JSON request to the Integrator:

```
//Create new socket that will connect to the POS endpoint (POS_IP_ADDR:8765)
Socket socket = new Socket();
socket.connect(new InetSocketAddress(POSIPADDR, APP_PORT), CONN_TIMEOUT);

//write data
BufferedOutputStream bos = new
BufferedOutputStream(socket.getOutputStream());
bos.write(request.getBytes("UTF-8"));
bos.flush();

...
```

After sending the TCP request, the Integrator should take control of the device screen – and (if the SmartConnect request was valid) will begin the action for processing the payment for the user.

We do not close the socket yet, because we still need to read the response back from Integrator.

5.3 TCP Example Response

Since we sent the SmartConnect request using a TCP connection, we will listen for the response back from the Integrator on the initial Socket we used to send the request.

Here is an example for receiving the SmartConnect JSON response from the Integrator after the initial request was processed:

```
...
//read response
DataInputStream dis = new DataInputStream(socket.getInputStream());
response += dis.readLine();

//shutdown socket stuff
socket.shutdownInput();
socket.shutdownOutput();
socket.close();
```

Once the response is received, you should parse the JSON to determine the result.

6.0 Miscellaneous

Other items of interest.

6.1 Parse Signature from JSON Response

If you had set **signature:true** in the JSON request, and the transaction performed supports a signature, then the application will prompt the user for their signature after an approved transaction. In this case, the signature will be returned to the calling application inside the **Intent** data object.

```
data.getByteArrayExtra("signature")
```

In such case, you can create a Bitmap object of the signature object for display, storing, or printing using the following sample code:

```
Bitmap signatureBitmap = BitmapFactory.decodeByteArray(data.getByteArrayExtra("signature"), 0,
data.getByteArrayExtra("signature").length);
```

6.2 resultCode Table

The following table represents the possible resultCode values:

resultCode	Description
00	Success / Approved
Non-zero	Unsuccessful; check the processor response list for the error meaning.
-50X -501	General Application Error Transaction Cancelled

6.3 Performing Payments with Payment 'Tokens'

Standard Transactions

Normally, the Integrator application expects a SmartConnect request that, if some type of payment is required, will request the user to Swipe/Insert/Tap their card to use as the payment method for the request to the processor. In such case, no payment information is itself contained in the 'caller apps' SmartConnect request to the integrator.

For example, here is a standard *Credit Sale* SmartConnect request which will prompt the user to swipe their card before processing their transaction with the processor:

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Sale",
    "amount":"1.00",
    "tip_amount":"0.00"
  }
}
```

Token Transactions

Certain processors allow the user to submit a previously obtained ‘payment token’ in lieu of a card swipe (or other physical capture method from the user such as EMV/TAP).

In these cases, the Integrator will allow the ‘caller app’ to submit the ‘payment token’ in the SmartConnect request as an additional field. When this occurs, the Integrator application will use the ‘payment token’ for processing the transaction with the payment host instead of prompting the user to Swipe/Insert/Tap their card. If the payment token is valid, and the processor accepts it – then the transaction should approve normally.

For example, here is a *Credit Sale* SmartConnect request in which the caller application submits a payment token in the request:

```
{
  "action":{
    "processor":"EVO",
    "receipt":true
  },
  "payment":{
    "type":"Sale",
    "panDataToken":"03a0106a-d62a-4a47-9301-
6060c1774e26200fca8b-a7bd-4044-by6e-3dab4473d538",
    "amount":"1.00",
    "tip_amount":"0.00"
  }
}
```

6.3.1 Payment Token Processor Support

Currently, only the following Nexgo Integrator applications support using a Payment Token in lieu of the card number:

EVO

6.4 Supported SmartConnect Transaction Types by Application / Processor

It is important to note which processors support which transaction types through the Integrator application. Not all transactions supported by SmartConnect are supported by every processor / Integrator application.

As an example, GiftActivate and other Gift card transaction types are supported by some processor / applications, but are not supported by others. If you attempt to perform an unsupported transaction type on a processor that does not support the feature – the Integrator will return an unsupported error.

Here is a summarized table of what transaction types are supported by SmartConnect for different processor / Integrator applications:

	API_KEY	EVO	NVL	TSYS
Credit + Debit	Sale	Y	Y	Y
	Auth	Y	Y	Y
	Capture	Y	Y	Y
	Void	Y	Y	Y
	Return	Y	Y	Y
	Reprint	Y	Y	Y
	Tip Adjustment	Y	Y	Y
	getLastStatus	Y	Y	Y
EBT	EBTFSPurchase	Y	N	Y
	EBTCBPurchase	Y	N	Y
	EBTBalance	Y	N	N
	EBTFSVoucher	Y	N	Y
	EBTFSVoucherReturn	Y	N	N
	EBTFSReturn	Y	N	Y
	Batch Close	Y	Y	Y
GIFT	GiftActivate	Y	N	N
	GiftRedemption	Y	N	N
	GiftReload	Y	N	N
	GiftBalance	Y	N	N

Note(s):

- API_KEY is the ***type*** SmartConnect parameter for each transaction type.
- Processor names may be shortened:
    - **EVO** → EVO
    - **NVL** → FD Nashville
    - **TSYS** → TSYS TransIT Multi-Pass