



MoneyToken API Document

Version 1.0

WadzPay.com

Confidential © 2025 WadzPay

Notice

All information contained in this documentation is proprietary to WadzPay. WadzPay reserves any and all intellectual property rights in respect of this document.

The Customer is given the right to use this document in accordance with the agreement with WadzPay. Unless as permitted by the relevant agreement, no part of this documentation may be reproduced, stored, or transmitted in any form or by electronic, mechanical, recording, or any other means, without the prior written permission of WadzPay.

This document is confidential and proprietary and is not intended for use by any person other than the named Customer.

Version Control

Version	Date	Author	Reviewer	Description
1.0	14-Apr-2025	Vinod L	Ashish K	Baseline

Table of Contents

1. PURPOSE	4
2. JWT TOKEN.....	4
3. WALLET	4
3.1 CUSTOMER / END USER REGISTRATION	4
3.2 BALANCE ENQUIRY	6
3.3 TRANSFER API	8
3.4 TRANSACTIONAL DETAILS.....	11
4. TRANSACTION ACCEPTANCE.....	13
4.1 PAYMENT INFO	13
4.2 GET TRANSACTION STATUS	15
4.3 GET SETTLEMENT REPORT	16
4.4 REFUND	18
4.5 CUSTOMER OFFLINE - POS INITIATED TRANSACTION	19

1. Purpose

This document's purpose is to provide details of Backend APIs (Application programming interfaces) accessible to client.

2. JWT TOKEN

JSON web token (JWT), pronounced "jot", is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. Again, JWT is a standard, meaning that all JWTs are tokens, but not all tokens are JWTs.

Because of its relatively small size, a JWT can be sent through a URL, through a POST parameter, or inside an HTTP header, and it is transmitted quickly. A JWT contains all the required information about an entity to avoid querying a database more than once. The recipient of a JWT also does not need to call a server to validate the token.

There are benefits to using JWTs when compared to simple web tokens (SWTs) and Security Assertion Markup Language (SAML) tokens.

Login Details for JWT Token:

API URL -

Username -

API : /api/v1/getJwtToken

3. Wallet

3.1 Customer / End User Registration

Description:

The API can be used by the financial institution (client) to add a new customer to the institution. When a new user registers with the institution, the institution has to send a request to WadzPay system API to register and activate the end user. The request payload must contain the fields enlisted in "Request Data Definition" section. The response will contain fields given in "Response Data definition" table.

API method and endpoint:

Method: POST

URI user/registration/

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	CustomerName	String	20	Yes	Name of the customer
2.	CustomerID	String	20	Yes	Unique ID of the customer

3.	customerMobile	String	20	Optional	Mobile Number of the customer
4	customerEmail	String	20	Optional	Email of the Customer
5	PartnerInstitutionID	String	20	optional	Unique ID of the Partner Institution ID

Example Request

Actual Request body of API will be place here after implementation of API

```
{
  "customerName": "CUST NAME",
  "customerID": "UNIQUE ID",
  "customerMobile": "+91999999***",
  "customerEmail": "CUST EMAIL",
  "partnerInstitutionID": "Partner Name"
}
```

Possible Errors

#	Error code	Description
1.	400	INVALID_REQUEST
2.	403	Unauthorized
3.	500	User Already Register with same Customer Name and Institute

Response Data Definition

#	Field	Type	Length	Description
1.	CustomerID	String	20	Unique ID of the customer
2.	CustomerName	String	20	Name of the customer
3.	CustomerRegistration Number	String	20	Unique Registration number which will be created by API
4	InstitutionName	String	20	Name of the Institution
5	InstitutionID	String	20	Unique ID of the Institution
6.	CustomerType	String	20	Type of Customer like merchant or individual etc..
7	CustomerwalletID	String	20	Customer WalletID
8.	CreatedDate	Date YYYY:MM:DD HH:MM:SSS		Date along with time when the transaction was created.
9.	Status	String	10	Status of the transaction.

Play Load request and response:

```
{
  "customerName": "CUST Name",
  "customerID": "UNIQUE ID",
  "customerMobile": "+919999999***",
  "customerEmail": "CUST EMAIL",
  "partnerInstitutionID": "PARTNER NAME"
}
```

Response:

```
{
  "customerName ": "CUST NAME",
  InstitutionName: "BANK NAME",
  Institution ID: "ABC1234",
  CustomerID='12345',
  RegistrationID="",
  CustomerRegistration Number="",
  Customer wallet ID='abcd-1234-achs-6789',
  Created Date="2025:01:15 20:18:56.888",
  "Status": Success/failure/Errored
}
```

3.2 Balance Enquiry

Description:

The API can be used by customer(s) (client) to Display available balance at end-user Wallet. When the end user logs into the institution APP (client), the institution sends a request to Wadzpays API to load/Display current available balance. The request payload must contain the fields enlisted in "Request Data Definition" section. The response will contain fields given in "Response Data definition" table.

API method and endpoint:

Method: GET

URI: /user/fetchWalletBalance

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	CustomerID	String	20	Yes	Unique ID of the customer
2.	tokenType	String	10	Optional	Token Name which customer wants to check balance (INRT, SART etc.)

Example Request

Actual Request body of API will be place here after implementation of API

```
/user/fetchWalletBalance?customerID=ID OF THE CUST'
```

Possible Errors

#	Error code	Description
1.	400	INVALID_REQUEST
2.	403	Unauthorized
3.	500	No Data found

Response Data Definition

#	Field	Type	Length	Description
1.	CustomerID	String	10	Unique Customer Number
2.	CustomerName	String	10	Name of the Customer
3.	customerEmail	String	15	Email of the customer
4.	customerPhoneNumber	String	15	Phone number of the customer
5.	TokenType	String	10	Type of token like INRT, SART etc.
6.	NoOfTokens	List	10	10 count of INRT tokens, 20 counts of SART tokens etc....
7.	TransactionDate	Date YYYY:MM:DD HH:MM:SSS	10	Date along with time when the transaction was created.

Play Load request and response:

```
{
  "customerID ": "CUST ID"
}
```

Response:

```
{
  "customerID": "Unique ID",
  "customerEmail": "CUST EMAIL",
  "customerPhoneNumber": "CUST Phone Number",
  "customerName": "CUST NAME",
  "tokenType": {
    INR; 50Units
  },
  "transactionDate": "2025-01-27T06:11:47.965320800Z"
}
```

"Status": success/failure/Errored

3.3 Transfer API

Description:

The API can be used for the scenarios below.

- **Load:** **Load** tokens into Institution's wallet: add additional tokens into circulation
- **Buy:** End user wants to **buy** (load) the tokens: Client sends a request using Wadzpays API to transfer tokens from client's wallet to end user Wallet.
- **Sell:** End user wants to **sell** the tokens: Client sends a request to Wadzpays API to transfer token from end user Wallet to client's wallet.
- **Transfer:** End user wants to **transfer** the tokens to another Customer: Client sends a request to Wadzpays API to transfer token from one end user Wallet to another end user wallet.
- **Withdraw:** Allow customer to **withdraw** tokens, i.e., customer will collect assets equivalent to the tokens and WadzPay will retract those tokens and burn them.

The request payload must contain the fields enlisted in "Request Data Definition" section. The response will contain fields given in "Response Data definition" table.

API method and endpoint:

Method: POST

URI: /user/transferTokens

Transaction Type Data Definition

Transaction Type	Transaction Code
Load	100
Sell	200
Buy	300
Transfer	400
withdraw	500

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	CustomerName	String	20	Optional	Name of the customer
2.	CustomerID	String	20	Yes	Wallet address
3	InstitutionName	String	20	Optional	Name of the Institution
4	InstitutionID	String	20	Yes	Unique ID of the Institution
5	TypeOfTransaction	String	10	Yes	Type of Transaction user wants to perform - Buy, Sell, Withdraw, Transfer.

6.	tokenName	String	10	Yes	Token Name which customer wants to load (INRT, SART etc.)
7.	noOfTokens	Long	10	Yes	How many tokens need to transfer from sender wallet to receiver wallet
8	SenderWallet ID	long	20	Yes	Sender wallet address
9	receiverWalletID	long	20	Yes	Receiver Customer Unique ID.
10	SenderDetails	Object	Object	Optional	Complete details of the sender

Example Request

```
/transferTokens?tokenName='INRT'&noOfTokens=1.01u& CustomerID
&& receiverCustomerID = receiverCustomerId&TypeOfTransaction= "SELL"
```

Possible Errors

#	Error code	Description
1.	400	INVALID_REQUEST
2.	403	Unauthorized
3.	500	receiverWalletId Wallet doesn't exist

Response Data Definition

#	Field	Type	Length	Description
1.	CustomerName	String	20	Name of the customer
2.	CustomerID	String	20	Wallet address
3.	customerUpdatedBalance	Long	10	Customer balance after buy/sell/transfer/withdraw into new tokens
4	InstitutionName	String	20	Name of the Institution
5	InstitutionID	String	20	Unique ID of the Institution
6	institutionUpdatedBalance	Long	10	Institution updated balance after loading new tokens
7	TypeOfTransaction	String	10	Type of Transaction want to user perform - buy,sell,withdraw or transfer

8.	tokenName	String	10	Token Name which customer wants to load (INRT, SART etc.)
9.	noOfTokens	Long	6	How many tokens need to transfer from sender wallet to receiver wallet
	receiverCustomerID	Long	20	Receiver Customer Unique ID.
11	TransactionID	String	40	
12	ReceiverName	String	20	Name of the Receiver
13.	CreatedDate	Date YYYY:MM:DD HH:MM:SSS		Date along with time when the transaction was created.
14.	Status	String	15	Status of the transaction.

```
{
  "customerID ": "ee48bd8b-e4aa-4ddd-ad94-bb4d986c913d",
  "SenderDetails ": [
    {
      "waletaddress": 1,
      "custName ": "firstName LastName",
      "walletBalance":
      "walletTokens":[
        {
          "INRT": available units,
          "SpecialTokens": available units
        }
      ]
    }
  ]
  "customerUpdatedBalance":
  "Status": success/failure/Errored
  "Date": "2025:01:25 14:56:55.123"
}
```

Required fields Vs option fields with respective Transaction Type:

Transaction Type	Required fields	Optional Fields	Description
Buy	Customer ID, Number of tokens, Token type	Rest of filed are optional	Customer ID will be Customer Unique Id
Sell	Customer ID, Number of tokens, Token type	Rest of filed are optional	Receivers' wallet ID will be Institute Wallet ID
Transfer	Customer Idr and Receivers customerID's, Number of tokens, Token type	Rest of filed are optional	PEER to PEER transfer
Withdraw	Customer ID, Number of tokens, Token type	Rest of filed are optional	Receiver wallet ID will be wadzpays Wallet.
Load	Number of tokens, Token type	Rest of filed are optional	Sender wallet ID will be Wadzpays Wallet ID and receiver will be Institute Wallet

3.4 Transactional Details

Description:

The API can be used by customer(s) (bank) to display transactional history of an enduser. When an end user sends request to institution APP(client), institution will send request to Wadzpays API to load/Display historical transactions details. By default, API will return only the last 10 recent transactions. But the end user has the feasibility to see older transactions details as well with respective date and custom fields. The request payload must contain the fields enlisted in "Request Data Definition" section. The response will contain fields given in "Response Data definition" table.

API method and endpoint:

Method: GET

URI: /user/transactionDetails

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	CustomerID	String	40	Yes	Unique customer ID
2.	FromDate	String	10	Optional	Start date of the transaction enquire

3	ToDate	String	10	Optional	End date of the transaction enquire
4	TransactionType	String	20	Optional	Will display the data based on dates
5	assetName	String	10	Optional	Token Name which customer wants to filter (INRT, SART etc.)

Example Request

Actual Request body of API will be place here after implementation of API

```
/user/transactionDetails?customerID=ID OF THE CUSTOMER'&fromDate=""& toDate=""
```

Possible Errors

#	Error code	Description
1.	400	INVALID_REQUEST
2.	403	Unauthorized
3.	500	No Date found

Response Data Definition

#	Field	Type	Length	Description
1.	CustomerID	String	10	Unique Customer Number
2	CustomerName	String	10	Name of the Customer
3.	TransactionID	String	10	Transaction ID
4.	TokenType	String	10	Type of token like INRT, SART etc..
5.	NoOfTokens	String	10	10 count of INRT tokens
6.	TransactionDate	Date YYYY:MM:DD HH:MM:SSS	10	Date along with time when the transaction was created.

Play Load request and response:

```
{
  "customerID ": "CUST ID",
  FromDate:"",
  ToDate:""
}
```

Response:

```
{
```

```
TransactionDeatils{
    "customerName ": "CUST NAME",
    "CustomerID" : "Unique ID",
    "TokenType": INRT,
    NoOfTokens:10,
    Transaction Date:10:10:2025 16:18:23:000
}
"Status": Success/failure/Errored
```

4. Transaction Acceptance

4.1 Payment Info

Description:

The API can be used by Client (POS (point of sale)/E-commerce website/ATM) system to send the transaction details of sale/withdrawal to/ WadzPay system. Once the blockchain based token / currency is chosen (INRT tokens, can be used for value equivalent to India Rupee) at the point of sale, this API is used to obtain the payment parameters for receiving the payment, which is shown to the customer on POS/ATM in the form of QR code. The request payload must contain the fields enlisted in “Request Data Definition” section. The response will contain fields given in “Response Data definition” table. The wallet address given in response fields must be used by Client’s POS/ATM machine for generating the QR code and displaying on POS/ecommerce website/ATM machine for scanning.

API method and endpoint:

Method: POST

URI: /pos/merchant/paymentAlgo

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	digitalCurrencyType	String	10	Yes	Digital currency type for POS/E-COMMERCE/ATM requests.
2.	fiatType	String	10	Yes	Fiat currency type (INR) for transaction requests
3.	fiatAmount	Number	10	Yes	Fiat amount (amount in INR) of order. Supports up to 2 Number places.
4.	posId	String	10	Yes	Terminal ID. Provided by WadzPay. (Example: - 0, 1)
5.	extPosId	String	20	NO	Client Terminal id
6.	extPosSequenceNo	String	20	NO	Client sequence no.
7.	extPosTransactionId	String	50	NO	Client Transaction ID.

8.	description	String	50	Yes	Description of transaction.
9.	extPosLogicalDate	String	20	no	Terminal Logical date provided by Client
10.	extPosShift	String	10	no	Client Terminal Shift No.
11.	extPosActualDate	String	20	no	Client Terminal Actual Date
12.	extPosActualTime	String	20	no	Client Terminal Actual Time

Possible Errors

#	Error code	Description
1.	400	INVALID_AMOUNT_NEGATIVE_OR_ZERO
2.	403	Unauthorized

Response Data Definition

#	Field	Type	Length	Description
1.	uuid	String	20	WadzPay unique transaction Id
2.	totalDigitalCurrency	String	10	Digital amount (amount in INR Tokens) requested
3.	address	String	40	Acquirer's (client's) Wallet Address
4.	totalFiatReceived	number	10	Fiat amount (INR amount) requested
5.	fiatType	String	10	Fiat asset type (INR) requested
6.	digitalCurrencyType	String	10	Digital asset type (INRT Tokens) requested
7.	transactionStatus	String	10	Transaction status SUCCESS, FAIL etc
8.	digitalCurrencyReceived	number	10	Digital amount (Amount in INRT Tokens) received
9.	transactionId	String	20	Transaction Id
10.	blockChainHash	String	20	Block chain hash
11.	merchantName	String	20	Merchant Name
12.	qrEncryptedString	String	180	QR string encrypted (refer below for more details)

Note:

“qrEncryptedString” is created by encrypting the QR Code string that will be formed as below:

Sl. No.	Field	Description
1.	Transaction ID	Transaction Acceptance system assigns a unique Transaction Id to every transaction and records in the transaction database along with the Merchant and POS details.
2.	Separator “ ”	“ ” to separate the fields for easy understanding at Wallet system

3.	Blockchain Address for receiving Tokens	Blockchain address that is assigned to the client by the Transaction Acceptance system.
4.	Separator “ ”	“ ” to separate the fields for easy understanding at Issuance system
5.	Token Id or Type	This is the name of the Closed Loop token that will be accepted at POS (E.g.: “INRT”
6.	Separator “ ”	“ ” to separate the fields for easy understanding at Issuance system
7.	Transaction Amount in tokens	The transaction amount will be in fiat (INR), which maps 1:1 with INRT. Therefore, the transaction amount in INR will be populated as INRT amount.
8.	Separator “ ”	“ ” to separate the fields for easy understanding at Issuance system
9.	Merchant Id	Merchant Id provided by WadzPay
10.	Separator “ ”	“ ” to separate the fields for easy understanding at Issuance system
11.	POS Id	POS Id provided by WadzPay
12.	Separator “ ”	“ ” to separate the fields for easy understanding at Issuance system
13.	Merchant Name Display	This is the name of the merchant where the transaction takes place on POS.

4.2 Get Transaction Status

Description:

This API can be used by Merchant POS (Acquirer’s system) to enquire about the status of transaction done earlier by providing the transaction Id (which was provided in the response of Payment Info API) as the input parameter. The response includes the details of the transaction including status and fiat and digital currency amounts.

API method and endpoint

Method: POST

URI: /pos/merchantDashboard/getTransactionStatusAlgo

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	uuid	String	40	Yes	wadzpay unique transaction id

Response Data Definition

#	Field	Type	Length	Description
1.	uuid	String	20	wadzpay unique id
2.	totalDigitalCurrency	Number	10	Digital currency amount (amount in INRT tokens) of transaction
3.	address	String	20	Blockchain wallet address (acquirer's blockchain address)
4.	totalFiatReceived	Number	10	Transaction amount in Fiat (INR amount)
5.	fiatType	String	10	Fiat type of transaction (INR)
6.	digitalCurrencyType	String	10	Digital currency type (INRT Tokens)
7.	transactionStatus	String	10	Status of transaction where values can be SUCCESSFUL, FAILED, IN_PROGRESS, UNDERPAID, OVERPAID.
8.	digitalCurrencyReceived	Number	10	Digital currency amount (INRT Tokens amount) received at Wallet
9.	transactionId	String	20	WadzPay Unique id for transaction
10.	blockChainHash	String	40	Blockchain transaction hash for tracking on blockchain scan
11.	totalFiatReceived	Number	10	Total Fiat (INR) amount of transaction

Possible Errors

#	Error code	Description
1.	403	Unauthorized
2.	404	Transaction with Transaction Id xxx not found

4.3 Get settlement report

Description:

This API shall be used by the merchant / Client to receive information of transactions that have been successfully completed as well as refunded in a specific date range. The total amount of transactions which have been accepted for Refund will be subtracted from the total amount of sales transactions within same Date range. The request payload must contain the fields enlisted in “Request Data Definition” section. The response will contain fields given in “Response Data definition” table.

API method and endpoint:

Method: GET

URI: /pos/merchant/settlement

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	from	Date	20	Yes	From date and time for Settlement Report (2025-01-20T08:53:05.320Z)
2.	to	Date	20	Yes	To date and time for Settlement Report (2025-01-31T08:53:05.320Z)
3.	posId	String	10	No	Terminal Id provided by WadzPay

NOTE: If Terminal Id is not provided with request: - Settlement report will contain all transactions of all POS (point of sale).

Example Request

Actual Request body of API will be placed here after implementation of API

Possible Errors

#	Error code	Description
1.	400	INVALID_PAGINATION
2.	403	Unauthorized
3.	404	USER_NOT_FOUND, MERCHANT_NOT_FOUND

Response Data Definition

#	Field	Type	Length	Description
1.	uuid	String	20	WadzPay unique id
2.	transactionId	String	20	WadzPay unique transaction id
3.	posid	String	20	WadzPay provided POS Terminal id
4.	date	Date	20	WadzPay transaction date and time
5.	digitalCurrency	String	10	Client digital currency (INRT Tokens) type
6.	digitalAmountRequested	String	10	Digital amount (Amount in INRT tokens) requested by Client
7.	digitalAmountReceived	Number	10	Digital amount (amount in INRT Tokens) received by WadzPay (up to 8 Numbers)
8.	merchantDiscount	Number	10	Merchant Discount applicable for the transaction
9.	payableDigitalAmountToMerchant	Number	10	Digital amount (amount in INRT Tokens) payable to merchant (up to 8 Numbers)
10.	fiatCurrencyName	String	10	Client Fiat currency type (SAR)
12.	status	String	10	Payment status
13.	fiatAmountRequested	String	10	Fiat (INR) Amount Requested
14.	totalFiatAmountReceived	String	10	Fiat (INR) Amount Received

4.4 Refund

Description:

When merchant wants to process refund of a transaction done in digital assets (INRT tokens), they capture the transaction identification from the customer's receipt and obtain the customer's wallet id by scanning their QR code and use this API to process the refund. Once successfully executed, this API will deduct the INRT tokens equivalent to the refund amount from the Merchant's settlement and credit those tokens to the customer's wallet.

API method and endpoint:

Method: POST

URI: /pos/merchantDashboard/submitRefundOnPOS

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1	transactionId	String	20	Yes	'WadzPay's unique transaction Id obtained from Customer's receipt.
2	refundWalletAddress	String	20	Yes	Customer wallet address (merchant will scan customer's QR code from their wallet)
3	refundDigitalType	String	10	Yes	Customer selected digital asset (INRT Tokens)
4	refundFiatType	Number	10	Yes	Customer selected fiat asset (INR)
5	refundAmountFiat	Number	10	Yes	Customer fiat amount to be refunded (Amount in INR)
6	reasonForRefund	String	20	No	Refund reason – Freeform text
7	refundMode	String	10	Yes	Use the value "Wallet" to return the INRT tokens to customer's wallet
8	extPosLogicalDate	String	20	No	Logical Date on POS terminal
9	extPosShift	String	10	No	Shift of POS operator
10	extPosActualDate	String	20	No	Actual Date on POS terminal
11	extPosActualTime	String	20	No	Actual Time on POS terminal

Example Request

Actual Request body of API will be place here after implementation of API

Possible Errors

#	Error code	Description
1.	400	INVALID_AMOUNT_NEGATIVE_OR_ZERO
2.	403	Unauthorized

Response Data Definition

#	Field	Type	Lenght	Description
1	transactionId	String	20	Transaction Id from the request
2	refundWalletAddress	String	20	Customer wallet address to which the refund is done
3	refundDigitalType	String	10	Customer selected digital asset (SAR Tokens)
4	refundFiatType	Number	10	Customer selected fiat asset (SAR)
5	refundAmountFiat	Number	10	Customer fiat amount to be refunded (Amount in SAR)
6	refundAmountDigital	Number	10	Refund amount in digital asset (SAR Token amount)
7	reasonForRefund	String	20	Refund reason – Freeform text
8	refundMode	String	10	Use the value “Wallet” to return the SAR tokens to customer’s wallet
9	refundStatus	String	10	“success”, “failure”
10	refundStatusReason	String	20	Reason for Failure of refund transaction

4.5 Customer Offline - POS Initiated Transaction

Description:

This API can be used by Merchant when Customer is Offline and can’t scan the QR code from the POS. Merchant will need to scan the customer’s QR code (static QR code, available with customer as sticker, key chain, etc.), provide the transaction amount and request the customer to enter their security code on the POS terminal. This API will then be called with Customer’s wallet address (captured from QR code), transaction amount and passcode. The response will provide the confirmation or failure of the transaction back to POS.

API method and endpoint:

Method: POST

URI: /merchantDashboard/admin/merchantReceiveFromWallet

Request Data Definition

#	Field	Type	Length	Mandatory	Description
1.	walletUserAddress	String	40	Yes	Customer's wallet address, obtained by POS scanning customer's QR code.
2.	passCodeHash	String	120	Yes	Security code entered by the customer on POS, encrypted by Customer.
3.	digitalCurrencyType	String	10	Yes	Digital currency type (INRT Tokens).
4.	fiatType	String	10	Yes	Fiat currency type (INR) for transaction requests
5.	fiatAmount	Number	10	Yes	Fiat amount (Amount in INR) of order. Supports up to 2 Number places.
6.	posId	String	10	Yes	Terminal ID. Provided by WadzPay. (Example: - 0, 1)
7.	extPosId	String	20	No	Client Terminal id
8.	extPosSequenceNo	String	20	No	Client sequence no.
9.	extPosTransactionId	String	50	No	Client Transaction ID.
10.	description	String	50	No	Description of transaction.
11.	extPosLogicalDate	String	20	No	Terminal Logical date provided by Client
12.	extPosShift	String	10	No	Client Terminal Shift No.
13.	extPosActualDate	String	20	No	Client Terminal Actual Date
14.	extPosActualTime	String	20	No	Client Terminal Actual Time

Possible Errors

#	Error code	Description
1.	400	INVALID_AMOUNT_NEGATIVE_OR_ZERO
2.	403	Unauthorized

Response Data Definition

#	Field	Type	Length	Description
1.	uuid	String	20	WadzPay unique transaction Id
2.	createdAt	number	10	Created Date and time
3.	amount	String	10	Amount processed
4.	asset	String	10	Currency unit of digital
5.	fiatAmount	String	10	Amount of fiat (amount in INR)
6.	fiatAsset	number	10	Currency unit of fiat (INR)

7.	status	String	20	Status of transaction
8.	statusReason	String	20	Reason for status
9.	Description	number	20	Description of transaction
10.	blockchainTxId	String	180	Transaction Id of blockchain
11.	totalDigitalCurrencyReceived	number	10	Total digital currency received (amount in INRT Tokens)
12.	totalFiatReceived	String	20	Total fiat received (amount in INR)
13.	posId	String	20	Terminal Id provided by WadzPay
14.	transactionId	String	20	WadzPay's unique transaction Id
15.	paymentReceivedDate	String	10	Payment received date
16.	extPosSequenceNo	String	20	Pos sequence number
17.	extPosTransactionId	number	10	Ext pos transaction id
18.	extPosLogicalDate	String	20	Ext pos logical date
19.	extPosShift	number	10	Ext pos shift
20.	extPosActualDate	String	10	Ext pos actual date
21.	extPosActualTime	String	20	Ext pos actual time