



---

# Transaction Engine *POS Provider* API

## **v0.7**

The *Transaction Engine* exposes several SOAP webservice endpoints. This document describes the POS Provider API which is used by retailers to process transactions.

## Revision History

<i>Revision History</i>				
<b>Date</b>	<b>Version Number</b>	<b>Description</b>	<b>Authors</b>	<b>Approved By</b>
3 February 2014	0.1	API Creation	Francois du Toit	Francois du Toit
28 March 2014	0.2	API Modification. New formatting applied.	Wesley Jordan	Eugene Smal
31 March 2014	0.3	API Modification. Better fields descriptions and small fixes.	Eugene Smal	Eugene Smal
1 May 2014	0.4	API Modification. Added non-transactional calls and small fixes.	Michelle Schlunz	Michelle Schlunz
19 November 2014	0.5	API updates. Better fields descriptions and small fixes.	Joseph Ou	Joseph Ou
20 January 2015	0.6	API Modification. Corrected transaction history request.	Michelle Schlunz	Michelle Schlunz
26 February 2015	0.7	API Modification. Added refund request.	Michelle Schlunz	Michelle Schlunz

## General Notes

This section will delve into some general API topics.

### API Notation

Code (or content closely-related to code) is formatted as follows:

```
echo 'Hello World';
```

Parameters are formatted as follows:

---

parameterName
Parameter Description.
<i>Parameter Type</i>

---

Required parameters are identified with an asterisk.

A strict ISO 8601 date format is followed for dates:

In Java a date can be formatted in the ISO 8601 format with the following snippet:

```
SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ssZ");
```

### Header Parameters

The following list specifies the header parameters which are required in each web service call.

---

apiCredentials
An API Credentials Object.

---

<b>id</b>
The application id.
<i>String</i>
<b>password</b>
The application password hashed by SHA1.
<i>String</i>

---



---

### Resources

Resources are represented in plural form.

For each resource defined in the API sections the following fields may be defined.

**ENDPOINT**

Displays the resource endpoint.

**METHODS**

Tabulates the available HTTP verbs along with the list of parameters that they accept.

A **bold** field in the Request column indicates that it's mandatory

A ✓ in the Request column of an operation denotes a field available in the request message.

A ✓ in the Response column of an operation denotes its presence in the response payload.

When a parameter is checked under `GET` it entails that the parameter can be added to the URL as a search parameter. E.g. `{root}/{resource}?parameter=value`.

**DESCRIPTION**

Describes the resource.

**PARAMETERS**

Lists all the parameters accepted by the resource endpoint.

---

**WSDL URL**

The QA WSDL can be found at:

`http://rad2.wigroup.co:8080/wigroup-te-za-rad/PosProviderWS?wsdl`

**Monetary Values**

Monetary amounts are both accepted and returned in cents.

# Pos Provider API

## Transaction

### DESCRIPTION

A transaction is logged in the transaction engine and routed to the relevant VSP.

### METHODS

#### Available Methods

	CREATE	
	Request	Response
<b>type</b>	✓	✓
<b>basketAmount</b>	✓	
cashbackAmount	✓	
tipAmount	✓	
<b>totalAmount</b>	✓	
switchTrxId	✓	✓
<b>token</b>	✓	✓
<b>id</b>	✓	✓
payload	✓	✓
<b>type</b>	✓	✓
<b>storeTrxDetails</b>	✓	✓
basketId	✓	✓
trxId	✓	✓
cashierId	✓	✓
storeId*	✓	✓
remoteStoreId*	✓	✓
retailerId*	✓	✓
posId	✓	✓
products	✓	✓
product (repeated)	✓	✓
id	✓	✓
units	✓	✓
pricePerUnit	✓	✓
<b>responseCode</b>		✓
<b>responseDesc</b>		✓
wiTrxId		✓

CREATE	
Request	Response
basketAmountProcessed	✓
cashbackAmountProcessed	✓
tipAmountProcessed	✓
totalAmountProcessed	✓
vsp	✓
id	✓
name	✓
trxId	✓
responseCode	✓
responseDesc	✓
balances	✓
balance (repeated)	✓
name	✓
type	✓
value	✓
coupons	✓
coupon (repeated)	✓
id	✓
name	✓
discount	✓
products	✓
product (repeated)	✓
id	✓
units	✓
discount	✓
discounts	✓
discount (repeated)	✓
name	✓
amount	✓
discountProducts	✓
discountProduct (repeated)	✓
id	✓
units	✓
discount	✓
loyalties	✓
loyalties (repeated)	✓
name	✓
type	✓

**CREATE****Request      Response**

earned	✓
balances	✓
balance (repeated)	✓
name	✓
type	✓
value	✓
* Either <code>remoteStoreId</code> and <code>retailerId</code> is present or just <code>storeId</code> in the request.	

**REQUEST****basketAmount**

This is the total transaction amount (in cents) of all products in the basket. If the transaction type is DEPOSIT, this is the deposit amount. If the transaction type is WITHDRAWAL this is the amount to withdraw.

*Integer*

**cashBackAmount**

This is the amount of cash in cent being withdrawn as part of the transaction. It can either be used in conjunction with a basketed payment or on its own. This is only for use with PAYMENT with cashback and not for WITHDRAWAL.

*Integer*

**tipAmount**

This is the amount of tip on top of the transaction amount. It can only be used with payment transactions.

*Integer*

**products**

A list of products in the basket. Please find information about the Product object here.

*List<Product>*

**storeTrxDetails**

This object contains information about the store in which the transaction took place. Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

**switchTrxId**

Should there be a transaction switch involved in between the POS and the wiPlatform this must be the switch's transaction reference.

*String*

token

This object contains information about the wiPlatform token associated with this transaction. Please find information about the Token object here.

*Token*

totalAmount

This is the total transaction amount in cent. This is the sum of the basketAmount, the cashbackAmount, and tipAmount.

*Integer*

type

This is the type of transaction. It can set to either `PAYMENT`, `DEPOSIT` or `WITHDRAWAL`.

*String*

wiTrxId

This is the wiPlatform unique identifier for a given transaction. The identifier is assigned by TE and will be sent back in the transaction response, so that it can be used to reference a transaction or for use in advises.

*Long*

responseCode

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

responseDesc

Readable description of responseCode.

*String*

totalAmountProcessed

This is the total transaction amount processed in cent. This is the sum of the basketAmountProcessed and the cashbackAmountProcessed.

*Integer*

basketAmountProcessed

This is the total transaction amount processed in cent for all products in the basket. If the transaction type is DEPOSIT, this is the deposit amount processed.

*Integer*

cashbackAmountProcessed



This is the amount of cash in cent being withdrawn as part of the transaction. It can either be used in conjunction with a basketed payment or on its own.

*Integer*

tipAmountProcessed

This is the amount of tip on the transaction. It is processed on top of the basket.

*Integer*

vsp

This object contains information about the VSP associated with the transaction. Please find information about the VSPDetails object here.

*VSPDetails*

discount

This object contains information about the discount associated with this transaction. Please find information about the Discount object here.

*Discount*

loyalty

This object contains information about the loyalty earned on this transaction. Please find information about the Loyalty object here.

*Loyalty*

balance

This object contains information about the VSP balance after this transaction. Please find information about the Balance object here.

*Balance*

---

## Advise Transaction

---

### DESCRIPTION

A transaction must be advised as either finalised or reversed. If TE receives the message successfully, it will always respond that the advise was successful. It will then pass the advise upstream asynchronously if required. Discrepancies will be picked up during settlement process.

### METHODS

---

*Available Methods*

---

CREATE		
	Request	Response
<b>action</b>	✓	
<b>originalTrx</b>	✓	
wiTrxId	✓	✓
type	✓	✓
switchTrxId	✓	✓
vspTrxId	✓	✓
storeTrxDetails	✓	✓
basketId	✓	✓
trxId	✓	✓
cashierId	✓	✓
storeId*	✓	✓
remoteStoreId*	✓	✓
retailerId*	✓	✓
posId	✓	✓
<b>responseCode</b>		✓
<b>responseDesc</b>		✓

\* Either `remoteStoreId` and `retailerId` is present or just `storeId` in the request.

## PARAMETERS

**action**

The advise action to be executed. The options for action are either `FINALISE` or `REVERSE`.

*String*

**originalTrx**

This object contains Advice request sent by the user. Please find information about the OriginalTrx object here.

*OriginalTrx*

**responseCode**

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

**responseDesc**

Readable description of responseCode.

*String*

---

**Consumer Info**


---

## DESCRIPTION

Retrieves information about the consumer linked to a token. This call is currently only relevant for wiTag token types that have a user created in Token Manager (wiProfile)

## METHODS

<i>Available Methods</i>		
CREATE		
	Request	Response
<b>token</b>	✓	
<b>id</b>	✓	
payload	✓	
<b>type</b>	✓	
storeTrxDetails	✓	
basketId	✓	
trxId	✓	
cashierId	✓	
storeId*	✓	
remoteStoreId*	✓	
retailerId*	✓	
posId	✓	
<b>responseCode</b>		✓
<b>responseDesc</b>		✓
vsp		✓
id		✓
name		✓
trxId		✓
responseCode		✓
responseDesc		✓
balances		✓
balance (repeated)		✓
name		✓
type		✓
value		✓
coupons		✓
coupon (repeated)		✓
id		✓
name		✓
discount		✓
products		✓
product (repeated)		✓

**CREATE**

	<b>Request</b>	<b>Response</b>
id		✓
units		✓
discount		✓
consumer		✓
id		✓
avatarURL		✓
birthdayToday		✓
status		✓
mobileNr		✓

---

\* Either `remoteStoreId` and `retailerId` is present or just `storeId` in the request.

**PARAMETERS****token**

This object contains information about the wiPlatform token associated with this transaction. Please find information about the Token object here.

*Token*

**storeTrxDetails**

This object contains information about the store in which the transaction took place. Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

**responseCode**

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

**responseDesc**

Readable description of responseCode.

*String*

**consumer**

This object contains information about the consumer linked to the token. Please find information about the Consumer object here.

*Consumer*

**vsp**

This object contains information about the VSP linked to the token. Please find information about the VSPDetails object here.

*VSPDetails*

## Token Info

### DESCRIPTION

Retrieves information about the token, including the VSP and consumer information related to the token.

### METHODS

#### Available Methods

CREATE		
	Request	Response
<b>token</b>	✓	
<b>id</b>	✓	
payload	✓	
<b>type</b>	✓	
storeTrxDetails	✓	
basketId	✓	
trxId	✓	
cashierId	✓	
storeId*	✓	
remoteStoreId*	✓	
retailerId*	✓	
posId	✓	
vspId	✓	
<b>responseCode</b>		✓
<b>responseDesc</b>		✓
vsp		✓
id		✓
name		✓
trxId		✓
responseCode		✓
responseDesc		✓
balances		✓
balance (repeated)		✓
name		✓
type		✓
value		✓
coupons		✓
coupon (repeated)		✓
id		✓

**CREATE**

	<b>Request</b>	<b>Response</b>
name		✓
discount		✓
products		✓
product (repeated)		✓
id		✓
units		✓
discount		✓
consumer		✓
id		✓
avatarURL		✓
birthdayToday		✓
status		✓
mobileNr		✓

---

\* Either `remoteStoreId` and `retailerId` is present or just `storeId` in the request.

**PARAMETERS****token**

This object contains information about the wiPlatform token associated with this transaction. Please find information about the Token object here.

*Token*

**storeTrxDetails**

This object contains information about the store in which the transaction took place. Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

**vspld**

This field is not currently used and is reserved for future purposes.

*Long*

**responseCode**

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

**responseDesc**

Readable description of responseCode.

*String*

**consumer**

This object contains information about the consumer linked to the token. Please find information about the Consumer object [here](#).

*Consumer*

vsp

This object contains information about the VSP linked to the token. Please find information about the VSPDetails object [here](#).

*VSPDetails*

---

## Transaction History

---

### DESCRIPTION

Retrieves transactions linked to storeTrxDetails.

### METHODS

---

#### *Available Methods*

---

	CREATE	
	Request	Response
dateFrom	✓	✓
dateTo	✓	✓
pageOffset	✓	✓
pageSize	✓	✓
<b>storeTrxDetails</b>	✓	✓
basketId	✓	✓
trxId	✓	✓
cashierId	✓	✓
storeId	✓	✓
posId	✓	✓
<b>responseCode</b>		✓
<b>responseDesc</b>		✓
transactions		✓
mainTransaction (repeated)		✓
id		✓
type		✓
basketAmountProcessed		✓
cashbackAmountProcessed		✓
tipAmountProcessed		✓
totalAmountProcessed		✓
createDate		✓
lastModifiedDate		✓

**CREATE****Request      Response**

state	✓
token	✓
id	✓
payload	✓
type	✓
id	✓

**PARAMETERS****dateFrom**

This is the start date for the transaction history search. The date is exclusive. It should be of the form: `yyyy-MM-dd HH:mm:ss`.

*String*

**dateTo**

This is the end date for the transaction history search. The date is exclusive. It should be of the form: `yyyy-MM-dd HH:mm:ss`.

*String*

**pageSize**

This is the number of transaction results that will be returned per page from the transaction history result set. If this is not provided, TE will set it to a configured default size.

*Integer*

**pageOffset**

This is the page that should be returned from the transaction history result set. The first page starts at 0. TE will, by default, set the pageOffset to 0 if it is not provided in the request.

*Integer*

**storeTrxDetails**

This object contains information about the store in which the request took place. Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

**responseCode**

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

**responseDesc**

Readable description of responseCode.

*String*



## transactions

A list of transactions that meet the criteria for the transaction history request. Each transaction is represented by a MainTransaction object. Please find information about the MainTransaction object here.

*MainTransaction*

## Refund

### DESCRIPTION

A refund transaction is logged in the transaction engine and routed to the relevant VSP. It can only be used with payment transactions.

### METHODS

#### *Available Methods*

	CREATE	
	Request	Response
<b>refundAmount</b>	✓	
switchTrxId	✓	✓
<b>storeTrxDetails</b>	✓	✓
basketId	✓	✓
trxId	✓	✓
cashierId	✓	✓
storeId*	✓	✓
remoteStoreId*	✓	✓
retailerId*	✓	✓
posId	✓	✓
<b>originalTrxDetails</b>	✓	✓
wiTrxId	✓	✓
products	✓	✓
product (repeated)	✓	✓
id	✓	✓
units	✓	✓
pricePerUnit	✓	✓
<b>responseCode</b>		✓
<b>responseDesc</b>		✓
wiTrxId		✓
refundAmountProcessed		✓
vsp		✓
id		✓

**CREATE****Request      Response**

name	✓
trxId	✓
responseCode	✓
responseDesc	✓

\* Either `remoteStoreId` and `retailerId` is present or just `storeId` in the request.

**REQUEST****refundAmount**

This is the refund amount (in cents).

*Integer*

**switchTrxId**

Should there be a transaction switch involved in between the POS and the wiPlatform this must be the switch's transaction reference.

*String*

**storeTrxDetails**

This object contains information about the store in which the transaction took place. Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

**originalTrxDetails**

This object contains information about the original transaction. Please find information about the originalTrx object here.

*originalTrx*

**products**

The products object is used for meta information only. A list of products in the basket. Please find information about the Product object here.

*List<Product>*

**responseCode**

A code indicating the result of the request. -1 means successful. All other codes are treated as failed.

*Integer*

**responseDesc**

Readable description of responseCode.

*String*

**wiTrxId**

This is the wiPlatform unique identifier for a given transaction. The identifier is assigned by TE and will be sent back in the transaction response, so that it can be used to reference a transaction or for use in advises.

*Long*

refundAmountProcessed

This is the total refund amount processed in cent.

*Integer*

vsp

This object contains information about the VSP associated with the transaction. Please find information about the VSPDetails object here.

*VSPDetails*

---

# Objects

## Product

This object contains information about each product in a user's basket. The following list provides information on the attributes of the Product object.

---

### **id**

The SKU/Barcode of product purchased. The id should be consistent in either always sending through the product SKU or product basket.

*String*

### **pricePerUnit**

The price per unit.

*Integer*

### **units**

The number of units of this product purchased.

*Integer*

---

## StoreTrxDetails

This object contains information about the store in which the transaction took place. The following list provides information on the attributes of the StoreTrxDetails object.

---

### **basketId**

This is the identifier for the basket as assigned by the POS. The POS system must ensure that it stays the same across multiple transactions of the same basket so that unique baskets can be tracked and identified.

*String*

### **trxId**

This identifies and groups multiple wiPlatform transactions together into a single POS transaction. The POS system must ensure that it stays the same across multiple transactions of the same basket so that unique baskets can be tracked and identified.

*String*

### **cashierId**

Identifies the cashier that processed the transaction.

*String*

### **storeId\***

This is the unique wiGroup store identifier.

*Long*

#### **remoteStoreId\***

This is the store's own identifier. Can be used instead of the storeId. Must be used in combination with the retailerId.

*Long*

#### **retailerId\***

This is the unique wiGroup retailer identifier. Required if using the remoteStoreId field instead of the storeId field.

*Long*

#### **posId**

Identifies the POS terminal the transaction was processed on.

*String*

#### **products**

This object contains information about the products in a transaction's basket. Please find information about the Product here

*List<Balance>*

---

\* Either `remoteStoreId` and `retailerId` is present or just `storeId` in the request.

## **Token**

This object contains information about the wiPlatform token associated with this transaction. The following list provides information on the attributes of the Token object.

---

#### **id**

The unique token which the wiPlatform uses to identify and route the transaction to the VSP to authorize.

*String*

#### **payload**

This field can be used by the VSP to send through additional transaction details when using a wiTag to further secure the transaction payment token. It will not be used by the wiPlatform.

*String*

#### **type**

Type of token; either `WICODE`, `WITAG`, `WIQR` or `BIN`.

*String*

---

## **VSPDetails**

This object contains information about the VSP associated with this transaction. The following list provides information on the attributes of the VSPDetails object. This object will only be returned if the token is valid.

---

id

This is an unique VSP identifier.

*String*

name

This is the name of the VSP.

*String*

trxId

The VSP's transaction ID for a transaction. This is optionally returned by the VSP in the transaction response.

*String*

responseCode

The responseCode returned by the VSP. This should not be used to determine whether the transaction was approved or not. This should only be used for logging and error reporting to VSP.

*String*

responseDesc

The VSP's response description. Can be displayed addition to the transactionResponse.responseDesc if available to give more information on why the transaction failed.

*String*

message

This is a message that can be printed on the user's till slip. Sent back from the VSP on successful transactions.

*String*

balances

This object contains information about the VSP balances. This is only returned in the Token Info Response. Please find information about the Balance here

*List<Balance>*

coupons

This object contains information about the VSP coupons. This is only returned in the Token Info Response. Please find information about the Coupon here

*List<Coupon>*

---

## Discount

This object contains information about the discount associated with this transaction. The

following list provides information on the attributes of the Discount object.

---

name	Discount name/description.
	<i>String</i>
amount	The total discount amount for all products in discount.
	<i>Integer</i>
products	The products on which the discounts are applied. Please find information about the DiscountProduct here.
	<i>List&lt;DiscountProduct&gt;</i>

---

## DiscountProduct

This object contains information about the products on which discounts are applied in this transaction. The following list provides information on the attributes of the DiscountProduct object.

---

id	The product id used.
	<i>String</i>
units	The number of units processed in the discount.
	<i>Integer</i>
discount	The discount per product.
	<i>Integer</i>

---

## Loyalty

This object contains information about the loyalty earned on this transaction. The following list provides information on the attributes of the Loyalty object.

---

name	The name of the loyalty program earned against.
	<i>String</i>
type	The type of loyalty earned. Valid values are <code>CENT</code> , <code>POINTS</code> or <code>COUNT</code> .
	<i>String</i>
earned	

The loyalty amount earned. This can be points or cent value.

*Integer*

---

## Balance

This object contains information about the VSP balance(s) after this transaction. The following list provides information on the attributes of the Balance object.

---

name

Description of the balance.

*String*

type

The type of balance. Valid values are `CENT`, `POINTS` or `COUNT`.

*String*

value

Value of balance.

*Integer*

---

## OriginalTrx

This object contains Advice request sent by the user.

---

wiTrxId

The wiPlatform unique transaction identifier. It is assigned to the transaction by TE and returned in the transaction response. It is then used, thereafter, in advises to identify the original transaction.

*Long*

type

The transaction type. Value that are allowed: `PAYMENT`, `DEPOSIT`, `WITHDRAWAL`.

*String*

switchTrxId

Should there be a transaction switch involved in between the POS and the wiPlatform this must be the switch's transaction reference.

*String*

vspTrxId

This is the identifier that was assigned to this transaction by VSP and would have been returned by the VSP after processing the transaction.

*String*

basketAmountProcessed

This is the amount in the basket that was processed for this transaction.



*Long*

cashbackAmountProcessed

This is the cashback amount that was processed for this transaction.

*Long*

tipAmountProcessed

This is the tip amount that was processed for this transaction.

*Long*

totalAmountProcessed

This is the total amount of the transaction that was processed.

*Long*

storeTrxDetails

This object contains information about the store in which the transaction took place.

Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

## Consumer

This object contains the consumer (user) details.

name

The name of the user.

*String*

avatarURL

The avatarURL of the user.

*String*

birthdayToday

True if the user's birthday is today, false otherwise.

*Boolean*

status

The status of the user.

*String*

## Coupon

This object contains information about the VSP. The following list provides information on the attributes of the VSP object.

**id**

The identifier for the coupon.

*Long*

**name**

If a voucher or discount is available for the customer this is the name of the promo.

*String*

**value**

Amount in cents of the total discountable value.

*String*

**product**

The products on which the discounts are applied. Please find information about the DiscountProduct here.

*List<DiscountProduct>*

## MainTransaction

This object contains the details of a transaction.

**id**

The unique identifier for the transaction in TE.

*Long*

**type**

The transaction type. This can be set to either `PAYMENT`, `DEPOSIT` or `WITHDRAWAL`.

*String*

**vspId**

This is the identifier of the VSP where this transaction was sent to.

*Long*

**basketAmountProcessed**

This is the amount in the basket that was processed for this transaction.

*Long*

**cashbackAmountProcessed**

This is the cashback amount that was processed for this transaction.

*Long*

**tipAmountProcessed**

This is the tip amount that was processed for this transaction.

*Long*

**totalAmountProcessed**

This is the total amount of the transaction that was processed.

*Long*

**countryId**

This is the country code of the country where this transaction took place.

*String*

vspTrxId

This is the identifier that was assigned to this transaction by VSP and would have been returned by the VSP after processing the transaction.

*String*

vspRef

This is the identifier that TE assigned to this transaction for it to be identified on the VSP.

*String*

adviceRequestCount

The number of times that the advice request for this transaction was sent.

*Long*

createDate

This is the date and time for when TE first received this transaction. The format will be

`yyyy-MM-dd HH:mm:ss`.

*String*

lastModifiedDate

This is the date and time for when TE last did any process related to this transaction.

The format will be `yyyy-MM-dd HH:mm:ss`.

*String*

state

This is the current status of the transaction.

*String*

storeTrxDetails

This object contains information about the store in which the transaction took place.

Please find information about the StoreTrxDetails object here.

*StoreTrxDetails*

token

This object contains the details of the token that was associated with this transaction.

Please find information about the Token object here.

*Token*

---

## Response Codes

Every response contains `responseCode` and `responseDesc` fields. These indicate whether the request was successful or failed. The only successful `responseCode` is denoted by a value of -1. All other response codes indicate failed requests. The table below denotes all possible response codes.

<i>Response Codes</i>	
<b>Response Code</b>	<b>Response Description</b>
-1	Successful
0001	General System Error
0001	General System Error
0001	General System Error
0002	General System Error. System configuration incorrect.
0003	System Error. Request not found.
0004	General Database Error
0005	Duplicate key found
0006	Duplicate entry in db
0007	Could not process your request. We are experiencing connection problems to external service providers.
0021	Email address format invalid
0022	Mobile number invalid"
0023	Required field missing or invalid type used.
0024	General Input Error.
0025	Date format is invalid.
0026	Time format is invalid.
0027	Integer format is invalid.
0100	Invalid request. API credentials missing.
0300	Interface does not exist.
0301	Interface not activated.
0302	Invalid interface credentials.
0303	Interface method does not exist.
1000	VSP Reference invalid. A valid VSP Reference must be specified. Max 255 characters.
1001	VSP invalid. Must specify a value > 0.
1002	The wiCode life is invalid. Must specify a value > 0.
1003	wiCode pool depleted.
1004	wiCode is not reserved.

<b>Response Code</b>	<b>Response Description</b>
1005	wiCode reserved by another VSP.
1006	wiCode vspReference differs. Might be new transaction.
1007	wiCode cannot be refreshed. Either busy being processed or being expired.
1008	wiCode has expired.
1009	wiCode locked. Currently being used in a transaction.
1010	Lock type is invalid. Value must be either 'true' to lock or 'false' to unlock.
1011	The wiCode is invalid. A valid wiCode must be specified.
1012	The transaction type is not allowed for wiCode.
1013	Transaction type invalid.
1501	Invalid state supplied.
1501	Invalid state supplied.
1502	Invalid country id supplied.
1503	Invalid pos id supplied.
1504	Invalid retailer id supplied.
1505	Invalid store owner id supplied.
1506	Invalid store group id supplied.
1507	Invalid subdivision id supplied.
1508	Invalid store id supplied.
1509	Invalid vsp id supplied.
1510	User Authentication failed.
1511	User Authentication failed.
1512	User Authentication failed.
1513	Bank account credentials invalid.
1514	Invalid id supplied.
1515	Invalid batch time format.
1516	Invalid recon time format.
1517	Invalid bank account type.
1518	Invalid Bank account supplied.
2001	Invalid transaction type supplied.
2002	Invalid token type supplied.
2101	Invalid token id.
2102	Could not parse VSP transaction response to JSON.
2103	Could not parse Wiprofile lock response to JSON.
2104	Timeout while waiting for Wiprofile response.
2105	Transaction to VSP timed out.
2106	Timeout while waiting for vsp responset.
2107	Transaction failed.

<b>Response Code</b>	<b>Response Description</b>
2108	VSP not found.
2109	Advice request failed.
2110	Failed to log transaction products.
2111	API POS not found.
2112	Store not found.
2113	Lock failed.
2114	Store does not support Deposits.
2115	Store does not support Payments.
2116	Store does not support Withdrawals.
2117	VSP does not support Deposits.
2118	VSP does not support Payments.
2119	VSP does not support Withdrawals.
2120	Invalid token.
2121	Cashback can only be processed on a Payment.
2122	The totalAmount does not match basketAmount + cashbackAmount.
2123	Connection to VSP could not be established.
2124	POS provider not active.
2125	Store is not active.
2126	Store is not linked to POS provider.
2127	Store does not support Cashbacks.
2128	VSP does not support Cashbacks.
2129	Store does not accept VSP.
2130	Store does not allow Deposits for VSP.
2131	Store does not allow Payments for VSP.
2132	Store does not allow Withdrawals for VSP.
2133	Store does not allow Cashback on Payments for VSP.
2134	Token Info request failed.
3001	Invalid request. User credentials missing.
3002	Invalid user credentials.
3003	User not activated.
3004	Invalid user credentials.
3051	Password invalid. A valid password must be specified.
3052	Invalid state. Available states are {A, D, P, RP}.
3053	Invalid delivery method. Available methods are {sms, email}.
3054	Invalid token type specified.
3055	Country code invalid. A valid country code must be specified.
3056	OTP invalid. A valid OTP must be specified.
3057	General OTP Error.

<b>Response Code</b>	<b>Response Description</b>
3058	OTP is expired.
3059	Id invalid. A valid id must be specified.
3060	Date invalid. Must have format yyyy-MM-dd HH:mm:ss.
3101	Name invalid. Max length allowed is 255 Characters.
3102	Surname invalid. Max length allowed is 255 Characters.
3103	Gender invalid. Available options are {M, F}.
3104	Date of birth invalid. Must have format yyyy-MM-dd.
3105	User registration request failed.
3106	WiTag is already linked to a user.
3107	Username invalid. A valid username must be specified.
3108	User already activated.
3109	User already linked to VSP.
3110	User not linked to wiTag.
3111	User already linked to a wiTag with the same description.
3112	User not linked to VSP.
3113	User and VSP linked state inactive.
3114	User in progress with a transaction.
3115	User already reserved wiCode for VSP with the same VSP reference.
3116	User not reserved by VSP.
3117	Another user with the same email is already registered.
3118	Another user with the same mobile number is already registered.
3151	VSP not activated.
3152	VSP registration request failed. VSP already exists.
3153	VSP description invalid. Max length allowed is 255 Characters.
3154	User cancelled.
3155	VSP is not a discount VSP.
3156	VSP is not a loyalty VSP.
3157	VVSP is not linked to discount VSP.
3158	VSP is not linked to loyalty VSP.
3201	WiTag length invalid.
3202	WiTag description invalid.
3203	The WiTag life is invalid. Must specify a value > 0.
3204	WiTag not linked to a user.
3205	No session active for wiTag.
3230	WiQR invalid.
3231	WiQR is expired.
3251	VSP invalid. Must specify a value > 0.
3252	VSP Reference invalid. Max length allowed is 255 Characters.

<b>Response Code</b>	<b>Response Description</b>
3253	The token life is invalid. Must specify a value > 0.
3254	The WiCode is invalid. A valid WiCode must be specified.
3255	There is no wiCode pool available for your request parameters.
3256	WiCode Pool not activated.
3257	WiCode Pool Connection Error.
3258	Invalid WiCode Pool interface credentials.
3259	General WiCode Pool Error.
3260	WiCode is expired.
3261	WiCode not reserved.
3262	WiCode already locked.
3263	WiCode not linked to a user.
3264	WiCode pool depleted.
3265	WiCode linked to discount/loyalty wicodes.
3300	Invalid BIN.
4001	Invalid date supplied.

---