



VIRTUALIŲ PASLAUGŲ OPERATORIUS

Ateities g. 10, LT-08303, Vilnius, Lietuva
PVM mokėtojo kodas: LT100001611012
A/s LT242140030002025351 Nordea Bank Finland PLC
Įm. kodas: 300093064
Tel.: 8 5 2461706, Faks.: 8 5 2412150, Info@vpo.lt



VPO.GateWay Transactions. Creating V 1.2

© 2019 UAB Virtualių paslaugų operatorius

Table of Contents

1. Document summary	2
1.1. Purpose.....	2
1.2. Revision history.....	2
1.3. Definitions	2
1.4. Related documents	2
2. Getting started.....	3
3. Products information querying commands.....	4
3.1. Products.Info command	4
3.2. Products.List command	6
4. Transaction creating commands.....	8
4.1. Transactions. Create command.....	8
4.2. Transactions. Commit command	10
4.3. Transactions. Rollback. command.....	12

1. Document summary

1.1. Purpose

GateWay is used for external systems, partners to interact with VPO internal system. There are three types of interaction through GateWay: products querying, transaction creating and transaction querying.

This document describes transaction creating (together with products information querying) of GateWay interface. It is assumed, that you are familiarized with general part document of GateWay (see related documents).

1.2. Revision history

Version	Date	Description	Author
1.0	2015-10-08	Initial version	Simonas Fergelis
1.1	2016-02-09	Added Terminal attribute	Simonas Fergelis
1.2	2019-03-25	Kind clarification	Andrius Vosylius

1.3. Definitions

Definition	Description
GateWay	VPO transaction creating and querying solution
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable

1.4. Related documents

Document name	File name with extension
Vpo. GateWay. General	VPO.GateWay.0.General.vX.En.docx
Vpo. GateWay. Transactions. Querying	VPO.GateWay.2.Transactions.Querying.vX.En.docx

2. Getting started

GateWay interface use cases for transaction creating are shown below:

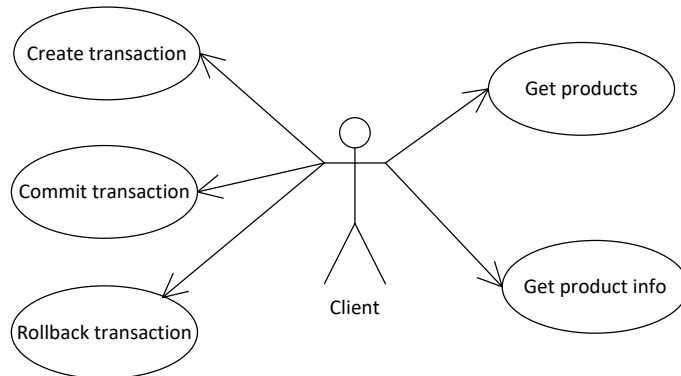


Figure 1: GateWay. Transactions creating

Transaction process consists of three use cases:

- Create transaction – transaction reservation
- Commit transaction – transaction confirmation
- Rollback transaction – transaction cancelation

Transaction has its own status. Following table shows transaction status changes:

From	To	Description
	Reserved	Transaction creation
Reserved	Committed	Transaction confirmation
Reserved	Rollbacked	Removing from receipt or canceling transaction reservation
Committed	Commit Approved	Transaction approval. Final status
Committed	Commit Forced	Status for manual finalization of successful transaction. Final status
Rollbacked	Rollback Approved	Transaction cancelation approval. Final status
Rollbacked	Rollback Forced	Status for manual finalization of reservation canceling. Final status

Graphical representation of status changes:

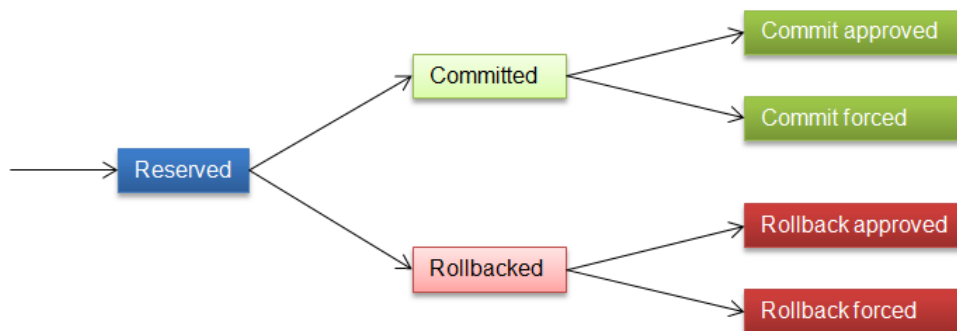


Figure 2: Transaction status changes

3. Products information querying commands

Main command XML structure for product quering is defined below:

```
<Products>
  <!-- Command -->
</Products>
```

The following sections describe each of the commands in more detail.

3.1. Products.Info command

Command “Info” returns detail information about concrete product. XML structure is defined below:

```
<Products>
  <Info Product="" Barcode="" />
</Products>
```

Info attributes description:

Name	Type	Mandatory	Description
Product	String(14)	No (Yes – if Barcode is empty)	Product identificator
Barcode	String(50)	No (Yes – if Product is empty)	Identifier (e.g. Card number, payee identificator and etc.)

One of attribute is required for identifying product. If both attributes are set Product is used for getting product information by default.

Request example:

```
<Request>
  <Data RequestId="1" Source="Source">
    <Products>
      <Info Product="456789" Barcode="CG42752525" />
    </Products>
  </Data>
  <Signature>
    NrDuBzPPZVCuvFbt8sbCYj0XlevvN9mor7IAQWm3SXnmCpkV2lcLdG7fudzWP5b9lwkfFriINdEKV+qbg
    CFx/1VeaNPS3BUVM9B/+Tiyss1wqtxSMABkhzdzxuqL0IZgSrAMd6fVVeex8WaN5A/qUDUJuZ6fK+rEIM
    5Mf8DyYlgXtJ/s17thJCYmYR10u7Ju4yh1HuLNzDB1rc34Mkk6577z018N3Cnt5CEUGa5H/2L1o+/w+5J
    f069fuLfY5peTyJuU/onMarcbk6aU7oI9TsEOMBm2rd4ehNlvOATbQMjaiWMQtDV0OfM34sg1Ng0mMGY
    wLvFHb17/ij+lw9Og==
  </Signature>
</Request>
```

After successful product information collection GateWay returns following XML response:

```
<Products>
  <Info>
    <Product Product="" Name="" ShortName="" Price="" MinPrice="" MaxPrice="" FloatPrice="" Tax=""
    Kind="">
      <Message></Message>
      <Barcode>
        <SemanticRules>
          <Rule Priority="" Type=" " Begin="" Length="" Constant="" ErrorText="" />
          <Rule Priority="" Type=" " Begin="" Length="" Constant="" ErrorText="" />
          ...
        </SemanticRules>
        <Aliases>
          <RegExp PriceDecimalPart="" PriceFractionalPart="" Price="" Tax="" Barcode="">
            <!-- RegExp example - ^(?#Bill payment)(?<Barcode>(CG4274\d{6})(?<Price>(\d{6})))$-->
          </RegExp>
        </Aliases>
      </Product>
    </Info>
  </Products>
```

```

<RegExp PriceDecimalPart="" PriceFractionalPart="" Price="" Tax="" Barcode="">
  <!-- RegExp example - ^(?#Bill payment)(?<Barcode>(CG4274\d{6})(?<Price>(\d{6})))$-->
</RegExp>
...
</Aliases>
</Barcode>
<Rows>
  <Row>
    No="" Name="" FromRequired="" TillRequired="" AmountRequired="" FromValue="" TillValue=""
    TariffValue="" AmountValue="" IntegerLength="" FractionalLength="" />
  <Row>
    No="" Name="" FromRequired="" TillRequired="" AmountRequired="" FromValue="" TillValue=""
    TariffValue="" AmountValue="" IntegerLength="" FractionalLength="" />
...
</Rows>
</Product>
</Info>
</Products>

```

Product attributes description:

Name	Type	Mandatory	Description
Product	String(14)	Yes	Product identifier
Name	String(200)	Yes	Product name
ShortName	String(38)	Yes	Product short name
Price	Decimal(12,2)	No	Product price
MinPrice	Decimal(12,2)	No	Product minimal allowed price
MaxPrice	Decimal(12,2)	No	Product maximal allowed price
FloatPrice	Bool (True, False)	Yes	If "True" product price must be set by third party. It could be parsed with Barcode or attribute Price with transaction command "Create"
Tax	Decimal(12,2)	No	Product tax
Kind	String(50)	No	If product is cash in, attribute is missed, else if cash out, returns "CashOut"

Message node value represents text that needs to be showed to cashier.

Response example:

```

<Response>
  <Products>
    <Info>
      <Product>
        Product="Bill" Name="Lesto" Price="50.00" MinPrice="1.00" MaxPrice="100.00"
        FloatPrice="True" Tax="2.00">
        <Message>Please enter bill info</Message>
        <Barcode>
          <SemanticRules>
            <Rule>
              Priority="1"
              Type="1"
              Begin=""
              Length=""
              Constant="&gt;1"
              ErrorText="Neteisingas kodas" />
            </Rule>

```

```

        Priority="2"
        Type="1"
        Begin=""
        Length=""
        Constant="&gt;1"
        ErrorText="Neteisingas kodas" />
</SemanticRules>
<Aliases>
  <RegExp Price="25" Barcode="CG427425">
    ^(?#Bill payment)(?<Barcode>(CG4274\d{6})(?<Price>(\d{6})))$
  </RegExp>
  <RegExp Price="25" Barcode="CG427525">
    ^(?#Bill payment)(?<Barcode>(CG4275\d{6})(?<Price>(\d{6})))$
  </RegExp>
</Aliases >
</Barcode>
<Rows>
  <Row
    No="1"
    Name="Eilutė 1"
    FromRequired="true"
    TillRequired="false"
    AmountRequired="true"
    FromEdit="true"
    TillEdit="true"
    TariffEdit="true"
    AmountEdit="true"
    FromValue="24"
    TillValue="24"
    TariffValue="51"
    AmountValue="51"
    IntegerLength="3"
    FractionalLength="2" />
  <Row
    No="2"
    Name="Eilutė 2"
    FromRequired="true"
    TillRequired="false"
    AmountRequired="false"
    FromValue="24"
    TillValue="24"
    TariffValue="545"
    AmountValue="51"
    IntegerLength="3"
    FractionalLength="2" />
</Rows>
</Info >
</Products>
</Response>

```

3.2. Products.List command

Product context command “List” returns all products. XML structure is defined below:

```

<Products>
  <List />
</Products>

```

Request example:

```

<Request>

```

```

<Data RequestId="1" Source="Source">
  <Products>
    <List />
  </Products>
</Data>
<Signature>
  NrDuBzPPZVCuvFbt8sbCYj0XlevvN9mor7IAQWm3SXnmCpkV2lcLdG7fudzWP5b9lwkfFriINdEKV+qbg
  CFx/1VeaNPS3BUVM9B/+Tiyss1wqtxSMABkhzdzuqL0IZgSrAMd6fVWeex8WaN5A/qUDUJuZ6fK+rEIM
  5Mf8DyYIgtJ/s17thJCYmYR10u7Ju4yh1HuLNzDB1rlc34MKK6577z018N3Cnt5CEUGa5H/2Lt1o+/w+5J
  f069fuLfY5peTyJuU/onMarcbk6aU7oI9TsEOMBm2rd4ehNlvOATbQMjaiWMQtdV0OfM34sg1Ng0mMGY
  wLvFHb17/ij+lw9Og==
</Signature>
</Request>

```

After successful product information collection GateWay returns following XML response:

```

<Products>
  <Info>
    <Product Product="" Name="" ShortName="" Price="" MinPrice="" MaxPrice="" FloatPrice="" Tax=""
    Kind="" />
    <Product Product="" Name="" ShortName="" Price="" MinPrice="" MaxPrice="" FloatPrice="" Tax=""
    Kind="" />
    <Product Product="" Name="" ShortName="" Price="" MinPrice="" MaxPrice="" FloatPrice="" Tax=""
    Kind="" />
    ...
  </Info>
</Products>

```

Product attributes description:

Name	Type	Mandatory	Description
Product	String(14)	Yes	Product identifier
Name	String(200)	Yes	Product name
ShortName	String(38)	Yes	Product short name
Price	Decimal(12,2)	No	Product price
MinPrice	Decimal(12,2)	No	Product minimal allowed price
MaxPrice	Decimal(12,2)	No	Product maximal allowed price
FloatPrice	Bool (True, False)	Yes	If "True" product price must be set by third party. It could be parsed with Barcode or attribute Price with transaction command "Create"
Tax	Decimal(12,2)	No	Product tax
Kind	String(50)	No	If product is cash in, attribute is missed, else if cash out, returns "CashOut"

Response example:

```

<Response>
  <Products>
    <List>
      <Product
        Product="235456" Name="Lesto" Price="50.00" MinPrice="0.01" MaxPrice="3000.00"
        FloatPrice="True" Tax="2.00" />
      <Product
        Product="26787" Name="Name" Price="2.00" MinPrice="0.01" MaxPrice="3000.00"
        FloatPrice="True" Tax="2.00" Kind="CashOut" />
    </List>
  </Products>
</Response>

```


4. Transaction creating commands

Main command XML structure for transaction creating is defined below:

```
<Transactions>
  <!-- Command -->
</Transactions>
```

The following sections describe each of the commands in more detail.

4.1. Transactions. Create command

Transaction reservation is used with command “Create”. This command has following XML structure:

```
<Transactions>
  <Create
    Product="" Barcode="" Price="" ReceiptNo="" PartnerTransactionId="" PrinterWidth=""
    SalesPlace="" Terminal="" Kind="" >
    <Rows><!-- Not mandatory. Used only with transactions that accept rows -->
      <Row No="" From="" Till="" Diff="" Amount="" />
      <Row No="" Till="" Diff="" Amount="" />
      ...
    </Rows>
  </Create>
</Transactions>
```

Create attributes description:

Name	Type	Mandatory	Description
Product	String(14)	No (Yes - if Barcode is empty)	Product identifier
Barcode	String(50)	No (Yes - if Product is empty)	Identifier (e.g. Card number, payee identifier and etc.)
Price	Decimal(12,2)	No	Transaction amount
ReceiptNo	Int	No	Receipt number
PartnerTransactionId	String(64)	Yes	Client (external system) transaction id
PrinterWidth	Int	No	Printer width describes width of receipt text that used in “Create” command response
SalesPlace	String(50)	No	Place name (store or kiosk no) where transaction was made
Terminal	String(50)	No	Terminal Id (cashregister no.) where transaction was made
Kind	String(50)	No	If transaction is cash in, attribute can be missed or declared as “CashIn”, else if cash out, must specify “CashOut”

One of Product or Barcode is mandatory. If Product and Barcode are set then Product is used for identification. Barcode can hold additional information like Price, Tax.

Row attributes description:

Name	Type	Mandatory	Description
No	Int	Yes	Row number
From	Int or Decimal(x,y)	No	Counter from

Till	Int or Decimal(x,y)	No	Counter till
Diff	Int or Decimal(x,y)	No	Counter difference
Amount	Int or Decimal(x,y)	No	Counter amount

Request example:

```

<Request>
  <Data RequestId="1" Source="Source">
    <Transactions>
      <Create
        Product="Bill"
        Barcode="20707000010719"
        Price="50.00"
        ReceiptNo="45848"
        PartnerTransactionId="223"
        PrinterWidth="28"
        CashRegister="CashRegister">
      </Create>
    </Transactions>
  </Data>
  <Signature>
    DK2zqL32ZKiW/AF7X3laBy4TuUcrAitcp7HEv3COtr40QaxBpMpwjrxVYeXnBcBMcePu2f14aT/RwR6EUg
    agEuuvbcsSapz3cDXPVhzx1MH65o/O1mNjweqtVC6q+mZvRVlcFZ8wcshGsmMMbns+p1wlkYvBZtRdd
    J8vSvT0plHHGnkPjq24zQ3aroRBPmD4adSu6geTQsYujdh4qTppjixOk8caJSm2VZApJ8bZ4ODgu1rWG
    3+0kEOUh8RWSkoTilUyfcISYFDGwFRvcetOMlpBTazDGdjEMeZXFg29j3g6mRZjAq+VMJIAYdpLFzT6y
    b7hngyg2AQZMxjyDLQAw==
  </Signature>
</Request>

```

After successful transaction reservation GateWay returns following XML response:

```

<Transactions>
  <Create>
    <Transaction
      TransactionId=""
      PartnerTransactionId=""
      Product=""
      Barcode=""
      Created=""
      Price=""
      Tax=""
      Source=""
      ReceiptNo=""
      Status=""
      SalesPlace=""
      Terminal=""
      Kind="" >
    <ReceiptText></ReceiptText>
  </Transaction>
</Create>
</Transactions>

```

Transaction attributes description:

Name	Type	Mandatory	Description
TransactionId	BigInt	Yes	Id of transaction created in GateWay
PartnerTransactionId	String(64)	Yes	Client (external system) transaction id
Product	String(14)	Yes	Product identifier

Barcode	String(50)	No	Identifier (e.g. Card number, payee identifier and etc.)
Created	DateTime	Yes	Transaction creation date and time (2015-10-01T08:25:18)
Price	Decimal(12,2)	Yes	Transaction amount (50.00)
Tax	Decimal(12,2)	Yes	Transaction tax (3.00)
Source	String(200)	Yes	Transaction source
ReceiptNo	Int	Yes	Receipt number
Status	String(200)	Yes	Transaction status (Reserved, Committed, Rollbacked, Commit approved, Commit forced, Rollback approved, Rollback forced)
SalesPlace	String(50)	No	Place name (store or kiosk no) where transaction was made
Terminal	String(50)	No	Terminal Id (cashregister no.) where transaction was made
Kind	String(50)	No	If transaction is cash in, attribute is missed, else if cash out, returns "CashOut"

ReceiptText node value stores text that needs to be printed on receipt.

Response example:

```

<Response>
  <Transactions>
    <Create>
      <Transaction
        TransactionId="1"
        PartnerTransactionId="1"
        Product="Bill"
        Barcode="20707000010719"
        Created="2015-10-01T08:05:18"
        Price="50.00"
        Tax="3.00"
        Source="Source"
        ReceiptNo="1"
        Status="Reserved"
        SalesPlace="SalesPlace"
        Terminal="Terminal" >
      <ReceiptText>Receipt text</ReceiptText>
    </Transaction>
  </Create>
</Transactions>
</Response>

```

4.2. Transactions. Commit command

Transaction must be committed by using command "Commit". XML structure is defined below:

```

<Transactions>
  <Commit TransactionId="" PartnerTransactionId="" />
</Transactions>

```

Commit attributes description:

Name	Type	Mandatory	Description
------	------	-----------	-------------

TransactionId	BigInt	No (Yes – if PartnerTransactionId is empty)	Id of transaction created in GateWay
PartnerTransactionId	String(64)	No (Yes – if TransactionId is empty)	Client (external system) transaction id

One of attribute is required for committing transaction. If both attributes are set TransactionId is used for committing transaction by default.

Request example:

```
<Request>
  <Data RequestId="1" Source="Source">
    <Transactions>
      <Commit TransactionId="1" />
    </Transactions>
  </Data>
  <Signature>
    H7YjXb6bSPOTGUni+Vqvo8zFFuOKNw7WfWrg0FG+E9UITwg+RY6pYD/DuLCnxpWSeQBONQumFKx
    KOV4nbwvSQHjclyo36+0GygH1n4d9G9e5FAyOqLZA6Wt26LJ6fud0kGWY9VK5DQjydPAjT6mBnixYP
    /oDPuS6GhUgKSgj+c6ONsrHTppUnKCjV0s0jyP+ptmhU4UFdWZakkf/KtTAPkVfcEmllWJdV/C3BaFidLf
    8vBBAFoNTb1XVW4YQ1yMHkR834Z+8XI2hWKPzlwSjz2yxLyCbi1plbD9T0bWl53Vvdg7SBZLGkjTbeK
    GWPFrHxFnQGkS60hCXY91i36g==
  </Signature>
</Request>
```

After successful transaction commitment GateWay returns following XML response:

```
<Transactions>
  <Commit>
    <Transaction
      TransactionId=""
      PartnerTransactionId=""
      Product=""
      Barcode=""
      Created=""
      Price=""
      Tax=""
      Source=""
      ReceiptNo=""
      Status="Committed"
      SalesPlace=""
      Terminal=""
      Kind="" />
    </Commit>
  </Transactions>
```

Transaction attributes description:

Name	Type	Mandatory	Description
TransactionId	BigInt	Yes	Id of transaction created in GateWay
PartnerTransactionId	String(64)	Yes	Client (external system) transaction id
Product	String(14)	Yes	Product identifier
Barcode	String(50)	No	Identifier (e.g. Card number, payee identifier and etc.)
Created	DateTime	Yes	Transaction creation date and time (2015-10-01T08:25:18)

Price	Decimal(12,2)	Yes	Transaction amount (50.00)
Tax	Decimal(12,2)	Yes	Transaction tax (3.00)
Source	String(200)	Yes	Transaction source
ReceiptNo	Int	Yes	Receipt number
Status	String(200)	Yes	Transaction status (Reserved, Committed, Rollbacked, Commit approved, Commit forced, Rollback approved, Rollback forced)
SalesPlace	String(50)	No	Place name (store or kiosk no) where transaction was made
Terminal	String(50)	No	Terminal Id (cashregister no.) where transaction was made
Kind	String(50)	No	If transaction is cash in, attribute is missed, else if cash out, returns "CashOut"

Response example:

```

<Response>
  <Transactions>
    <Commit>
      <Transaction
        TransactionId="1"
        PartnerTransactionId="1"
        Product="Bill"
        Barcode="20707000010719"
        Created="2015-10-01T08:25:18"
        Price="50.00"
        Tax="3.00"
        Source="Source"
        ReceiptNo="1"
        Status="Committed"
        SalesPlace="SalesPlace"
        Terminal="Terminal" />
      </Commit>
    </Transactions>
  </Response>

```

4.3. Transactions. Rollback. command

Transaction can be roll backed by using command "Rollback". XML structure is defined below:

```

<Transactions>
  <Rollback TransactionId="" PartnerTransactionId="" />
</Transactions>

```

Rollback attributes description:

Name	Type	Mandatory	Description
TransactionId	BigInt	No (Yes – if PartnerTransactionId is empty)	Id of transaction created in GateWay
PartnerTransactionId	String(64)	No (Yes – if TransactionId is empty)	Client (external system) transaction id

One of attribute is required for roll backing transaction. If both attributes are set TransactionId is used for roll backing transaction by default.

Request example:

```

<Request>
  <Data RequestId="1" Source="Source">
    <Transactions>
      <Rollback PartnerTransactionId="1" />
    </Transactions>
  </Data>
  <Signature>
    JByO1x9esbUAF9eB7cmZf3Gbg3IN3sLAeXc4jHbMaoXGZdTgsE64FftetGmKdwts1ZXd9MWVm5y6AXt
    kG6iRyci9ERwKCZ/biFx0UtHS3RpzOEFwXyown7zTB6HK9FITHFVpd1zuKZbUgT4LOFr+KoRRMRSLYp
    OTcZrMLNTc76FI0zmjP8iZLzUeKt9DgsvYjqW+VI7LQg4yvKlpUq45OeOkSI4uqwdQtdb2M1GZ9hMfgEC
    w5AcU3UdDGVcqsRqDEhiqrdHErduivo6fnpgBsuEQFq8q5vOnFRRLgUVQ7a9ACKDpbusm7dWzmT65t
    Ob/d46R63rsc/bAbXZ8BxOMA==
  </Signature>
</Request>

```

After successful transaction rollback GateWay returns following XML response:

```

<Transactions>
  <Rollback>
    <Transaction
      TransactionId=""
      PartnerTransactionId=""
      Product=""
      Barcode=""
      Created=""
      Price=""
      Tax=""
      Source=""
      ReceiptNo=""
      Status="Rollbacked"
      SalesPlace=""
      Terminal=""
      Kind=""
    />
  </Rollback>
</Transactions>

```

Transaction attributes description:

Name	Type	Mandatory	Description
TransactionId	BigInt	Yes	Id of transaction created in GateWay
PartnerTransactionId	String(64)	Yes	Client (external system) transaction id
Product	String(14)	Yes	Product identifier
Barcode	String(50)	No	Identifier (e.g. Card number, payee identifier and etc.)
Created	DateTime	Yes	Transaction creation date and time (2015-10-01T08:25:18)
Price	Decimal(12,2)	Yes	Transaction amount (50.00)
Tax	Decimal(12,2)	Yes	Transaction tax (3.00)
Source	String(200)	Yes	Transaction source
ReceiptNo	Int	Yes	Receipt number

Status	String(200)	Yes	Transaction status (Reserved, Committed, Rollbacked, Commit approved, Commit forced, Rollback approved, Rollback forced)
SalesPlace	String(50)	No	Place name (store or kiosk no) where transaction was made
Terminal	String(50)	No	Terminal Id (cashregister no.) where transaction was made
Kind	String(50)	No	If transaction is cash in, attribute is missed, else if cash out, returns "CashOut"

Response example:

```

<Response>
  <Transactions>
    <Rollback>
      <Transaction
        TransactionId="1"
        PartnerTransactionId="1"
        Product="Bill"
        Barcode="20707000010719"
        Created="2015-10-01T08:25:18"
        Price="50.00"
        Tax="3.00"
        Source="Source"
        ReceiptNo="1"
        Status="Rollbacked"
        SalesPlace="SalesPlace"
        Terminal="Terminal" />
      </Rollback>
    </Transactions>
  </Response >

```