



TECHNOLOGY PLATFORM FCC INTERFACE

Version 1.5



FUELING YOUR BUSINESS FOR SUCCESS

Contents

1	General	4
2	Security	5
3	Services	6
3.1	Get Setup	6
3.2	Price Update	7
3.3	Get RFID	8
3.4	Authorize Pump	9
3.5	Stop Pump.....	9
3.6	Get Transaction Status.....	10
3.7	Get Transactions for Order	12
3.8	Login.....	13
3.9	Upload Receipt File	14
3.10	ReportMOP	15
3.11	Get Last Acquired Tag	15
3.12	Get Current Location	16
4	Appendix.....	18
4.1	Error Codes	18
4.2	Transaction Status Codes.....	18
4.3	MOP Codes	19
4.4	Limit / Preset Codes	19

Version History Table

Date	Changed by	Changes	Version
13/10/2020	Ofer Y.	Initial version	1.01
08/01/2021	Ofer Y.	<ul style="list-style-type: none">• Rename SOSTopPump to SOTPStopPump• Added SOTPLLogin• Added SOTPUploadReceiptFile• Change TPGetTransactionStatusResult structure added fields TxnStartDateTime, TxnEndDateTime, OpenTotal, CloseTotal• Added custom fields to transaction data	1.20

		Affect Authorize and Get Transaction for order	
12/01/2021	Ofer Y.	<ul style="list-style-type: none">• Remove signature from GetTransacitonsforOrder and GetStatus• Added ReportMOP WS• Added MOP table• Added limit type table	1.3
12/01/2021	Ofer Y.	<ul style="list-style-type: none">• Use global nozzle id in GetSetup, GetTransactionforOrder and Authorization• Add GetLastAcquiredTag WS• Add additional data for GetStatus and GetTransactionforOrder Pump, Nozzle, PPV, Product name, Preset type, Preset value	1.4
19/01/2021	Ofer Y.	<ul style="list-style-type: none">• Add back GetLocation API	1.5

1 General

This document describes the FCC interface that is available for the technology platform for authorizing fuel transactions on browsers.

Assumptions

- Browsers has one pump – however various API require passing the pump and nozzle number
- Browsers has one product with known code - however various API requires passing product code
- Browser has ROCode that is known to TP server and be sent to validate in various APIs

Notes

- All examples show dummy signatures (SIG)
- Demo, mockup, FCC application (Desktop) is provide
 - Application can be used to get WSDL
 - Application runs as server on given port in address text box (do not change endpoint uri)
 - The application is provided as is for demonstration only
 - No signature check is done – Signed value must be “SIG”
 - Only limit type volume is supported

2 Security

To protect FCC from unauthorized access a security schema is to be used.

- Will use digital signature – for more information see in below links
https://en.wikipedia.org/wiki/Digital_signature
<https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq>
- Note that the communication layer to FCC is using TLS (HTTPS, SSL) protocol. The digital signature is on the application level
- TP server will generate key pairs (private / public)
- TP will use its private key to sign part of critical data passed to FCC
- FCC will use TP public key to validate the signature and thus validate the origin of each API
- In each relevant API there will be SignedData tag and SignValue tag (the signature)
 - Both these tags content is generated by the TP server and pass 'as is' by the mobile application
 - The SignValue will be sent encoded Base64
 - The format of the SignedData is list of commas separated parameters
 - The actual parameters depend per API
- FCC will keep the public key protected and no replacement of public key is allowed (an update version will be required)
- Sign is done using rsa-sha256 algorithm
- Sing is done over the content of the SignedData tag (not including <>)

3 Services

3.1 Get Setup

Purpose: Initial connect of mobile application to get the FCC setup - pumps and products

Usage: Mobile app will call FCC on initial connection to be update with FCC setup

API Name: [TPGetSetup](#)

Security Tag: Not required

In: None

Out: FCC setup

Example:

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPGetSetup>
      </tps:SOTPGetSetup>
    </soapenv:Body>
  </soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetSetupResponse xmlns="http://TPServices/">
      <SOTPGetSetupResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
        <Version>1.01</Version>
        <ROCode>1001</ROCode>
        <Products xmlns:a="http://orpak.com/SiteOmatServices/">
          <a:Product>
            <a:Code>1</a:Code>
            <a:Name>Regular</a:Name>
            <a:Price>0</a:Price>
          </a:Product>
        </Products>
        <Pumps xmlns:a="http://orpak.com/SiteOmatServices/">
          <a:Pump>
            <a:Name>1</a:Name>
            <a:Nozzles>
              <a:Nozzle>
```

```

        <a:Num>1</a:Num>
        <a:ProductCode>1</a:ProductCode>
    </a:Nozzle>
</a:Nozzles>
</a:Pump>
</Pumps>
</SOTPGetSetupResult>
</SOTPGetSetupResponse>
</s:Body>
</s:Envelope>

```

3.2 Price Update

Purpose: To set fuel price

Usage: Mobile app will get price update request from TP server and will call FCC when connected

API Name: [TPUpdatePrice](#)

Security Tag: <ROCode, UniqueID, Expiration Date, ProductCode,PPV>

In: SignedData

Out: None

Errors: OK, AuthorizationFailed, BadProductCode, BadROCode

Example

Request

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPPriceUpdate>
      <tps:SignValue>SIG</tps:SignValue>
      <tps:SignedData>1001,1234,DATE,1,2.3</tps:SignedData>
    </tps:SOTPPriceUpdate>
  </soapenv:Body>
</soapenv:Envelope>

```

Response

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPPriceUpdateResponse xmlns="http://TPServices/">
      <SOTPPriceUpdateResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
      </SOTPPriceUpdateResult>
    </SOTPPriceUpdateResponse>
  </s:Body>
</s:Envelope>

```

```
</s:Body>
</s:Envelope>
```

3.3 Get RFID

Purpose: To get the current RFID detected on specific pump and nozzle

Usage: Mobile application to call periodically to get the RFID

API Name: [TPGetRFID](#)

Security Tag: Not required

In: Pump number, Nozzle number

Out: RFID found on nozzle

Errors:

Example:

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPGetRFID>
      <tps:pump>1</tps:pump>
      <tps:nozzle>1</tps:nozzle>
    </tps:SOTPGetRFID>
  </soapenv:Body>
</soapenv:Envelope>
```

Respond

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetRFIDResponse xmlns="http://TPServices/">
      <SOTPGetRFIDResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
        <RFID>71201234556789000</RFID>
      </SOTPGetRFIDResult>
    </SOTPGetRFIDResponse>
  </s:Body>
</s:Envelope>
```


3.4 Authorize Pump

Purpose: To authorized fueling

Usage: Called by mobile app to start fueling. A global limit is provided by TP server. FCC will validate that total value (volume or amount) of all transactions for the OrderID do not exceed this limit.

API Name: [TPAuthorizePump](#)

Security Tag: <ROCode, OrderID, Expiration Date, Global Limit, Limit Type>

In: Pump, Product code, Limit, CustomFields

Up to optional 20 custom fields can be given.

Out: None

Errors: OK, NoSuchPump, BadProdcutCode, PumpInUse, AuthorizationFailed, AuthorizationExpiered, GlobalLimitReached

Example

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sit="http://orpak.com/SiteOmatServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <sit:SOTPAuthorizePump>
      <sit:SignValue>SIG</sit:SignValue>
      <sit:SignedData>1001,1234,DATE,1000,1</sit:SignedData>
      <sit:pump>1</sit:pump>
      <sit:nozzle>1</sit:nozzle>
      <sit:productCode>1</sit:productCode>
      <sit:limit>100</sit:limit>
      <sit:CustomFields>
        <sit:Field1>A1</sit:Field1>
        <sit:Field2>A2</sit:Field2>
        <sit:Field3>F3</sit:Field3>
      </sit:CustomFields>
    </sit:SOTPAuthorizePump>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPAuthorizePumpResponse xmlns="http://TPServices/">
      <SOTPAuthorizePumpResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
      </SOTPAuthorizePumpResult>
    </SOTPAuthorizePumpResponse>
  </s:Body>
</s:Envelope>
```

3.5 Stop Pump

Purpose: To stop fueling pump

Usage: In case fueling need to be stopped – transaction is terminated (once handle is placed back)

API Name: **TPStopPump**

Security Tag: Not required

In: Pump

Out: None

Errors: OK, NoSuchPump

Example:

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOPump>
      <tps:PumpName>1</tps:PumpName>
    </tps:SOPump>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOPumpResponse xmlns="http://TPServices/">
      <SOPumpResult xmlns:a="http://orpak.com/SiteOmatServices/" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <a:rc>0</a:rc>
        <a:rc_desc>OK</a:rc_desc>
      </SOPumpResult>
    </SOPumpResponse>
  </s:Body>
</s:Envelope>
```

3.6 Get Transaction Status

Purpose: To get running transaction status

Usage: After authorization mobile app will use this API to monitor the transaction status. The running volume and amount are returned. This API should be called at no more than once one second.

Status indicates one of the following (see code values table in Appendix)

- INUSE: Pump in in run mode – on going fueling
- STOPPED: Pump stopped fueling and waits for resume (in case of AVI RFID) or put nozzle back to end transaction
- IDLE, CALL: No transaction in progress – volume and amount are of last transaction
- INACTIVE: Pump in error

API Name: **TPGetTransactionStatus**

Security Tag: None

In: Pump

Out: TxnStartDate, TxnEndDate, Volume, Amount, OpenTotal, CloseTotal, Status, OrderID, TransactionID, Pump, Nozzle, PPV, Product name, Preset type, Preset value, CustomFields

TransactionID will be zero for running transaction

TxnEndTime, OpenTotal, CloseTotal will be zero for running transaction

CustomFields as given in authorization

Error: OK, NoSuchPump, AuthorizationFailed, AuthorizationExpired

Example:

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPGetTransactionStatus>
      <tps:pump>1</tps:pump>
    </tps:SOTPGetTransactionStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetTransactionStatusResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPGetTransactionStatusResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
        <TransactionData>
          <TxnStartDateTime>2020-10-09 12:41</TxnStartDateTime>
          <TxnEndDateTime>2020-10-09 12:52</TxnEndDateTime>
          <TransactionID>0</TransactionID>
          <Volume>1</Volume>
          <Amount>1.1</Amount>
          <OpenTotal>1234.56</OpenTotal>
          <CloseTotal>1235.56</CloseTotal>
          <Status>4</Status>
          <OrderID>1</OrderID>
          <Pump>1</Pump>
          <Nozzle>3</Nozzle>
          <ProductName>MS</ProductName>
          <PPV>50.1</PPV>
          <PresetType>1</PresetType>
          <PresetValue>100</PresetValue>
          <CustomFields>
            <Field1>A1</Field1>
            <Field2>A2</Field2>
          </CustomFields>
        </TransactionData>
      </SOTPGetTransactionStatusResult>
    </SOTPGetTransactionStatusResponse>
  </s:Body>
</s:Envelope>
```

3.7 Get Transactions for Order

Purpose: To retrieve previous transactions done for specific order

Usage: Mobile app can call in case some condition prevented it to get all transaction data (e.g. communication break)

API Name: [TPGetTransactionsForOrder](#)

Security Tag: None

In: OrderID

Out: List of transactions done for that order – per transaction – TxnStartDateTime, TxnEndDateTime, TransactionID, Volume, Amount, OpenTotal, CloseTotal, Pump, Nozzle, PPV, Product name, Preset type, Preset value, Custom Fields

Error: OK, NoSuchOrder, AuthorizationFailed, AuthorizationExpiered

Example

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPGetTransactionsForOrder>
      <tps:OrderID>1234</tps:OrderID>
    </tps:SOTPGetTransactionsForOrder>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetTransactionsForOrderResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPGetTransactionsForOrderResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
        <Transactions>
          <TransactionData>
            <TxnStartDateTime>2020-10-09 12:41</TxnStartDateTime>
            <TxnEndDateTime>2020-10-09 12:52</TxnEndDateTime>
            <TransactionID>20000011</TransactionID>
            <Volume>5.1</Volume>
            <Amount>21.1</Amount>
            <OpenTotal>0</OpenTotal>
            <CloseTotal>0</CloseTotal>
            <Status>0</Status>
            <OrderID>0</OrderID>
            <Pump>1</Pump>
            <Nozzle>3</Nozzle>
            <ProductName>MS</ProductName>
            <PPV>50.1</PPV>
            <PresetType>1</PresetType>
            <PresetValue>100</PresetValue>
            <CustomFields>
              <Field1>A1</Field1>
              <Field2>A2</Field2>
            </CustomFields>
          </TransactionData>
        </Transactions>
      </SOTPGetTransactionsForOrderResult>
    </SOTPGetTransactionsForOrderResponse>
  </s:Body>
</s:Envelope>
```

```

        <Field3>F3</Field3>
    </CustomFields>
</TransactionData>
<TransactionData>
    <TxnStartDateTime>2020-10-09 12:41</TxnStartDateTime>
    <TxnEndDateTime>2020-10-09 12:52</TxnEndDateTime>
    <TransactionID>20000012</TransactionID>
    <Volume>15.1</Volume>
    <Amount>34.1</Amount>
    <OpenTotal>0</OpenTotal>
    <CloseTotal>0</CloseTotal>
    <Status>0</Status>
    <OrderID>0</OrderID>
    <Pump>1</Pump>
    <Nozzle>3</Nozzle>
    <ProductName>MS</ProductName>
    <PPV>50.1</PPV>
    <PresetType>1</PresetType>
    <PresetValue>100</PresetValue>
    <CustomFields>
        <Field1>A1</Field1>
        <Field2>A2</Field2>
        <Field3>F3</Field3>
    </CustomFields>
</TransactionData>
</Transactions>
</SOTPGetTransactionsForOrderResult>
</SOTPGetTransactionsForOrderResponse>
</s:Body>
</s:Envelope>

```

3.8 Login

Purpose: To validate user / password

Usage: Mobile app can call to validate user login credentials. FCC will check existence of username and matched password.

API Name: [TPLogin](#)

Security Tag: None

In: UserName, Passwrod

Out: None

Error: OK, BadUserorPassword

Example

Request

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sit="http://orpak.com/SiteOmatServices/">
    <soapenv:Header/>
    <soapenv:Body>
        <sit:SOTPLLogin>
            <sit:UserName>Name</sit:UserName>

```

```

    <sit:Password>pass</sit:Password>
  </sit:SOTPLLogin>
</soapenv:Body>
</soapenv:Envelope>

```

Response

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPLLoginResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPLLoginResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
      </SOTPLLoginResult>
    </SOTPLLoginResponse>
  </s:Body>
</s:Envelope>

```

3.9 Upload Receipt File

Purpose: To upload a receipt file (image) to FCC

Usage: Mobile app can call to upload an image file to FCC for archive purpose. File size should not exceed 30k Bytes

API Name: [TPUploadReceiptFile](#)

Security Tag: None

In: FileName, Data

Data should be encoded Based 64

Out: None

Error: OK

Example – Given example encode in Base 64 “Test file”

Request

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sit="http://orpak.com/SiteOmatServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <sit:SOTPUploadReceiptFile>
      <sit:FileName>ID_22222_2021_01_12</sit:FileName>
      <sit:Data>VGZzdCBmaWxl</sit:Data>
    </sit:SOTPUploadReceiptFile>
  </soapenv:Body>
</soapenv:Envelope>

```

Response

```

<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPUploadReceiptFileResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPUploadReceiptFileResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
      </SOTPUploadReceiptFileResult>
    </SOTPUploadReceiptFileResponse>
  </s:Body>
</s:Envelope>

```

```
</s:Body>
</s:Envelope>
```

3.10 ReportMOP

Purpose: To notify FCC when receipt process was completed and MOP

Usage: Mobile app should call FCC after is completed the payment / receipt process including the MOP. FCC will record the MOP and that a receipt was printed. If this message is not received in defined time (in FCC) the transaction will be closed with CASH MOP.

API Name: [TPReportMOP](#)

Security Tag: None

In: Transaction ID, MOP

Out: None

Error: OK

Example

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sit="http://orpak.com/SiteOmatServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <sit:SOTPreportMOP>

      <sit:TransactionID>1235</sit:TransactionID>

      <sit:MOP>UPI</sit:MOP>
    </sit:SOTPreportMOP>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPreportMOPResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPreportMOPResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
      </SOTPreportMOPResult>
    </SOTPreportMOPResponse>
  </s:Body>
</s:Envelope>
```

3.11 Get Last Acquired Tag

Purpose: To get the last tag presented in any FCC device

Usage: Mobile app can check if any tag was presented including MC, HHR, RFID.

API Name: [TPGetLastAcquiredTag](#)

Security Tag: None

In: None

Out: Tag

Error: OK

Example

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sit="http://orpak.com/SiteOmatServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <sit:SOTPGetLastAcquiredTag/>
  </soapenv:Body>
</soapenv:Envelope>
```

Response

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetLastAcquiredTagResponse xmlns="http://orpak.com/SiteOmatServices/">
      <SOTPGetLastAcquiredTagResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
        <rc>0</rc>
        <rc_desc>OK</rc_desc>
        <Tag>71205678800078888</Tag>
      </SOTPGetLastAcquiredTagResult>
    </SOTPGetLastAcquiredTagResponse>
  </s:Body>
</s:Envelope>
```

3.12 Get Current Location

Purpose: To get current FCC (browser) location based on GPS

API Name: [SOTPGetLocation](#)

Security Tag: Not required

In: None

Out: Latitude, Longitude

Example

Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tps="http://TPServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <tps:SOTPGetLocation/>
  </soapenv:Body>
</soapenv:Envelope>
```

Respond

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <SOTPGetLocationResponse xmlns="http://TPServices/">
      <SOTPGetLocationResult xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
```



```
<rc>0</rc>
<rc_desc>OK</rc_desc>
<Longitude>34.834720555556</Longitude>
<Latitude>32.104016666667</Latitude>
</SOTPGetLocationResult>
</SOTPGetLocationResponse>
</s:Body>
</s:Envelope>
```

4 Appendix

4.1 Error Codes

Code	Description	Comment
0	OK	Operation successful
1	NoSuchPump	
2	BadROCode	
3	GeneralError	
4	NoSuchOrderID	
5	PumpInUse	
6	OverFuelLimitForOrder	
7	BadProductCode	
8	AuthorizationExpiered	
9	AuthorizationFailed	
10	GlobalLimitReached	
11	BadUserorPassword	

Note that all WS may return standard SOAP fault error in case bad format received.

4.2 Transaction Status Codes

Code	Description	
0	IDEL	
1	CALL	
2	READY	
3	INUSE	
4	STOPPED	

5	PAYABLE	
6		
7	INACTIVE	

4.3 MOP Codes

Code	Description	
CASH	Cash	
CREDIT	Credit	
REDEMP	Redemption	
OWN	Own use	
TEST	Testing	
SAMPLE	Sampling	
LOYLTY	Loyalty	
LOCAL	Local account	
COUPON	Coupon	
UPI	Unified Payments Interface	
WALLET	Wallet	
BQR	Bharat QR	

4.4 Limit / Preset Codes

Code	Type	
0	Money	

1	Volume	
2	None	