evopayments.com

SimpleTab QR Integration

Updated: 07/12/2021

Table of Contents

Overview

Accept Touch-free/Contactless payments by using QR Codes.

Here is a video overview:
https://vimeo.com/531389577/2600b99515

Quick video:
https://vimeo.com/531971158/cbde1e14cb

POS integration

Prepare for Integration – Convert Checks to JSON

The first step for the POS developer is to convert and provide a receipt, check, or bill in the JSON format. For anyone unfamiliar with JSON, there are free online developer resources where data in XML, HTML, or even CSV can be converted into JSON (see an example in this document).

Every POS system is different, but our machine learning development method will perform all the integration mapping of the receipt fields.

Required Field Types

Minimally a receipt must have a check number (can also be called order number, ticket number, or receipt number) that is unique to the bill or order as well as a merchant identifier (EVO Merchant number), a subtotal and a tax amount.

Optional Field Types

Optionally, the ticket can also have a customer phone, table number, seat numbers, items and modifiers. These will all be displayed on the payment page to the customer.

Note: Checks cannot be split by the customer using the items or seat numbers. Split function only allows an even 2, 3, or 4 way split.

Option: Providing a customer phone number field within the check JSON will generate an SMS text to the customer with the payment link.

The 1st Step – POST Check to Cloud

Integrate your POS to POST the JSON formatted checks to the SimpleTab QR Cloud server and receive the id and qr code URL in the response.

Note: the JSON of the check or bill must have the EVO assigned merchant MID.

POST

```
POST [/push/{ISV_Number}] HTTP/1.1
Accept: application/json
Content-Length: xxx
Content-Type: application/json
Host: [CLOUD URL]
{  [JSON OF CHECK]  }
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 21:10:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Status: 200 OK
Content-Length: 32
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
{
"id":"[32 character invoice ID]",
"qr":https://[cloudurl]/
}
```

Request Field	Description	Values	Field Type
CLOUD URL	SimpleTab Cloud URL	URL	Static URL
ISV_NUMBER	EVO assigned number indicating the ISV integration.	Specific to each ISV	32 Char ASCII
Response Field	**Description**	**Values**	**Field Type**
id	Invoice Number returned after push function is performed.	Variable	32 Char ASCII
qr	URL value for the POS system to provide to the cardholder as a link or QR code.	Web URL	ASCII

After receiving the response URL… the POS should use a conversion tool or script to generate a QR code representing the URL. This QR code should be prominently printed on the customer check as in the example below…

Sample Printed Check

Date: 2/7/2020	Time: 7:00 pm
Check: 1469	Server: 2006
Seat 1	
Sandwich	$5.99
Soda	$2.49
Seat 2	
Salad	$4.99
Subtotal:	$13.47
Tax:	$1.08
Total:	$14.55

Use your mobile phone camera to scan the QR code and pay.

Camera not reading the QR? Visit scanqr.io on the phones browser to scan.

The 2nd Step – GET Status of Checks

After receiving presenting the check QR code to the cardholder, the POS will need to send a GET pull from SimpleTab QR Cloud server to confirm payment status. The POS would need to automatically perform the GET pull of open checks every 15-20 seconds until the check is Paid or Cancelled.

GET

```
GET [/pull/{ISV_Number}/{id}] HTTP/1.1
Accept: application/x-www-form-urlencoded
Host: [CLOUD URL]
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 21:10:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Status: 200 OK
Content-Length: 4
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
```

{"amount":"25.60","code":"EGUDB4","pan":"XXXXXXXXXXXX1111","id":"42B00FCE7980476B90728FAA55CFF476", "status":"APPROVED","time":"4/20/2021 6:55:16 PM","tip":"3.60","original":"22.0","snap":"Paid", "network":"Visa","split":"false"}

Request Field	Description	Values	Field Type
CLOUD URL	SimpleTab Cloud URL	URL	Static URL
ISV_NUMBER	EVO assigned number indicating the ISV integration.	Specific to each ISV	32 Char ASCII
id	Invoice Number returned after push function is performed.	Variable	32 Char ASCII
Response Field	**Description**	**Values**	**Field Type**
amount	Total amount processed for this transaction	00.01 – 999999.99	Variable numeric with decimal
code	Transaction approval code from issuer	Variable	Variable alphanumeric
pan	Masked card PAN with last 4 digits	X's followed by last 4 of card.	Variable alphanumeric
id	Transaction ID tied to this specific authorization. Can referenced later for reversal.	variable	32 Char ASCII
status	Transaction response status	APPROVED or DECLINED	Variable ASCII
time	Date and Time of transaction	YYYY-MM-DDTHH:MM:SS.SSS	YYYY-MM-DDTHH:MM:SS.SSS
tip	Tip amount added by cardholder	00.01 – 999999.99	Variable numeric with decimal
original	Amount of original bill/check	00.01 – 999999.99	Variable numeric with decimal
snap	Status of entire check/invoice id	Paid, Pending, Cancelled	Variable alphanumeric
network	Card type used for payment	Visa, Mastercard, Discover, American Express	Variable alphanumeric
split	Indicates if check was split	true , false	Variable alphanumeric
PaymentToken	*To be added in Q3 2021*		

Sample response where Split was performed


```
{"amount":"11.00","code":"008340","pan":"527232XXXXXX9929","id":"13605cfcb93648bca0af91ebc227250b","stat
us":"APPROVED","time":"2021-05-
18T14:32:30.153","tip":"0","original":"22.0","snap":"Paid","network":"MasterCard","split":"true"},{"amou
nt":"11.00","code":"008340","pan":"527232XXXXXX9929","id":"13605cfcb93648bca0af91ebc227250b","status":"A
PPROVED","time":"2021-05-
18T14:32:30.153","tip":"0","original":"22.0","snap":"Paid","network":"MasterCard","split":"true"}
```


Other REST API Calls

When working with checks in the cloud, a POS will likely need to either remove and cancel checks or reverse a check for one reason or another.

Remove/Cancel Check

If a check is paid through another method or is no longer valid, the POS should send a remove. The response will be a success/fail. After performing a remove the check will return a snap:cancelled upon performing a /pull status.

PUT

```
PUT [/remove/{ISV_Number}/{INVOICE}] HTTP/1.1
Accept: application/x-www-form-urlencoded
Host: [CLOUD URL]
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 21:10:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Status: 200 OK
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Best practice - After getting the 200 OK response, perform a /pull status to verify that the check is cancelled ("snap":"Cancelled")


```
[ {"amount":"0.00","code":"","pan":"","id":"","status":"NonExistent","time":"","tip":"0.00",
"original":"0.00","snap":"Cancelled","network":"","split":""} ]
```

Reverse – Void and Refund Checks

If a check was paid and processed. The POS can submit a reverse to trigger a reversal of the amount. This can be done on checks that are part of the current open batch (aka void) or on checks that were closed in a prior batch (aka refund). The response will be a success/fail. After performing a reverse the check will return a snap:cancelled upon performing a /pull status, while the status will still be APPROVED.

GET

```
GET [/reverse/{ISV_Number}/{INVOICE}] HTTP/1.1
Accept: application/x-www-form-urlencoded
Host: [CLOUD URL]
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2020 21:10:53 GMT
Server: Apache/2.4.18 (Ubuntu)
Status: 200 OK
Content-Length: 0
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Best practice - After getting the 200 OK response, perform a /pull status to verify that the check is cancelled ("snap":"Cancelled")


```
[ {"amount":"0.05","code":"464415","pan":"XXXXXXXXXXX2090","id":"125B92F32CE7412AA1A5F1717E343F5D",
"status":"APPROVED","time":"2021-07-12 05:44:15.0","tip":"0.00","original":"0.00","snap":"Cancelled",
"network":"AmericanExpress","split":"false"} ]
```

Check Converted from XML to JSON

Sample Printed Check

Date: 2/7/2020 Time: 7:00 pm
Check: 1469 Server: 2006

Seat 1	
Sandwich	$5.99
Soda	$2.49
Seat 2	
Salad	$4.99
Subtotal:	$13.47
Tax:	$1.08
Total:	$14.55

XML Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <localtime>2020-02-07T21:07:00.000</localtime>
  <receipt_id>1469</receipt_id>
  <sale_type/>
  <taxes>
    <name>Tax</name>
    <value>1.08</value>
  </taxes>
  <staff>
    <id>2006</id>
    <has_original_id>true</has_original_id>
  </staff>
  <products>
    <seat>1</seat>
    <quantity>1</quantity>
    <discounts/>
    <price>5.99</price>
    <name>Sandwich</name>
    <modifiers/>
  </products>
  <products>
    <seat>1</seat>
    <quantity>1</quantity>
    <discounts/>
    <price>2.49</price>
    <name>Soda</name>
    <modifiers/>
  </products>
  <products>
    <seat>2</seat>
    <quantity>1</quantity>
    <discounts/>
    <price>4.99</price>
    <name>Salad</name>
    <modifiers/>
  </products>
  <consumer_id>11-1</consumer_id>
  <payment_methods/>
  <shop_id>1535509869033384</shop_id>
  <total>14.55</total>
  <is_void>false</is_void>
  <discounts/>
  <subtotal>13.47</subtotal>
</root>
```

JSON Data

```
{
   "root": {
      "localtime": "2020-02-07T21:07:00.000",
      "receipt_id": "1469",
      "sale_type": "",
      "taxes": {
         "name": "Tax",
         "value": "1.08"
      },
      "staff": {
         "id": "2006",
         "has_original_id": "true"
      },
      "products": [
         {
            "seat": "1",
            "quantity": "1",
            "discounts": "",
            "price": "5.99",
            "name": "Sandwich",
            "modifiers": ""
         },
         {
            "seat": "1",
            "quantity": "1",
            "discounts": "",
            "price": "2.49",
            "name": "Soda",
            "modifiers": ""
         },
         {
            "seat": "2",
            "quantity": "1",
            "discounts": "",
            "price": "4.99",
            "name": "Salad",
            "modifiers": ""
         }
      ],
      "consumer_id": "11-1",
      "payment_methods": "",
      "shop_id": "1535509869033384",
      "total": "14.55",
      "is_void": "false",
      "discounts": "",
      "subtotal": "13.47"
   }
}
```

Testing Endpoints

https://www.mimilieu.net/tomcat

In a future release this will be: https://qr-cert.simpletabcloud.com/tomcat