

Bakong QR Pay Integration



DOCUMENT HISTORY

Version	Date	Description
1.0	28.12.2020	
1.0.1	28.01.2021	<ul style="list-style-type: none">- Adding SDK system interaction flow.- Update product flow diagram.- Adding header parameter in api document.
1.0.2	16.07.2021	<ul style="list-style-type: none">- Add the notice part in the system integration section.
1.0.3	06.08.2021	<ul style="list-style-type: none">- Update content on EmvCo QR generation section.

Table of Contents

Abbreviation	2
Introduction	3
Overview	3
Purpose	3
Scope	3
Product Flow	3
How it works	4
System Interaction Flow	5
Integration Process	9
Prerequisites:	9
EMVCo QR Generation:	9
Implementing API:	11

Abbreviation

Abbreviation	Meaning
API	Application Programming Interface
QR Code	Refer to the Quick Response code for Cambodia which compiled with standard QR code called KHQR.
Customer	Refer to individuals who use bakong/bank apps to make payment via Bakong.

Introduction

Overview

QR code is the best option of payment for the quick response. Consumers can pay for their goods/services by using any mobile apps to scan QR codes that are displayed by merchants in a faster and easier way. So, let's start integration with Bakong dynamic QR code payments which offer more security than credit cards, as several high-profile data breaches have illustrated due to all the transferred data being encrypted.

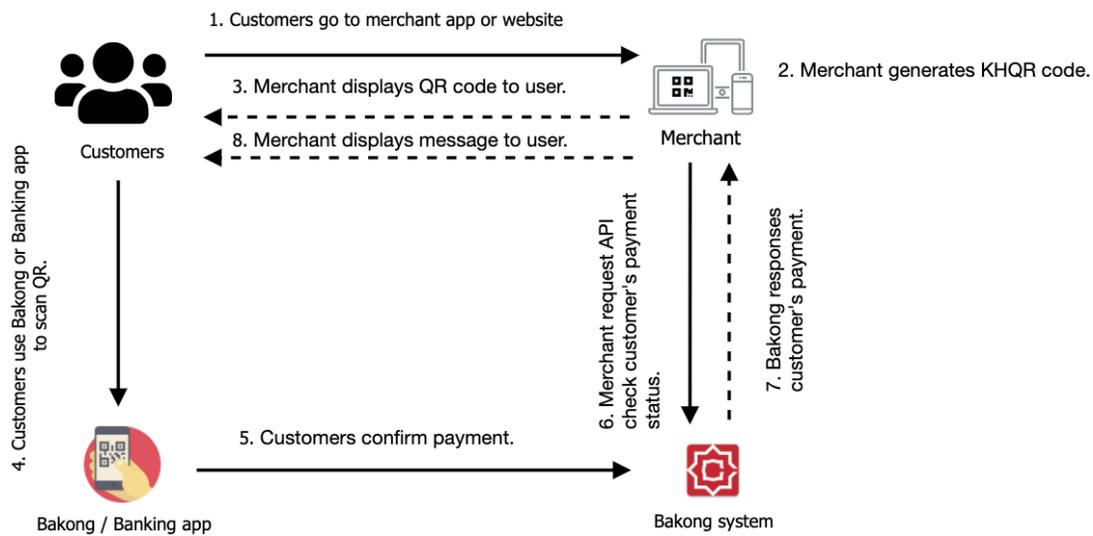
Purpose

This document describes the detailed specification of how to integrate Bakong dynamic QR code payment which is offered by National Bank of Cambodia. The expected readers are NBC technical team, third-party technical, POS service provider, software developers, etc. This can be used as reference for any interest related to the KHQR Integration.

Scope

This document contains the complete description of the QR pay integration specification including: Product Flow, System interaction flow, Integration Process and API List.

Product Flow



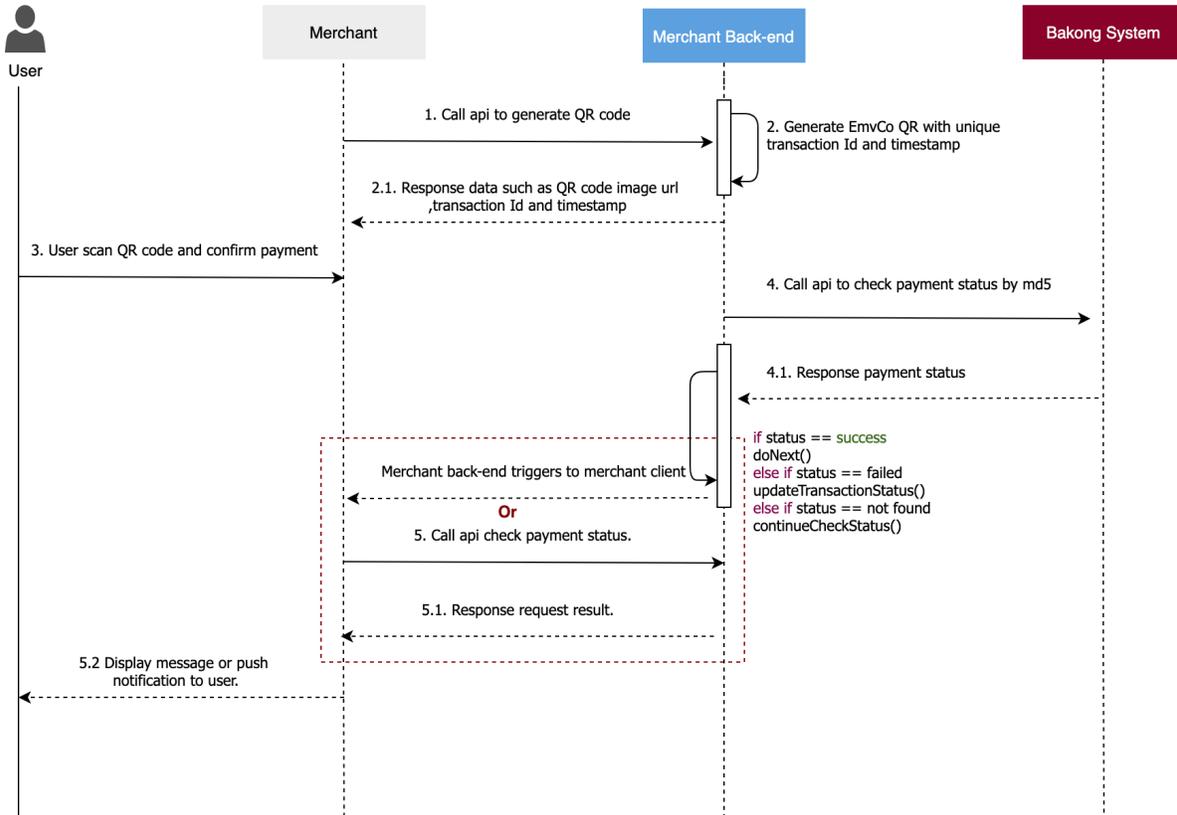
How it works

1. Customers make payment via a merchant app, website, or dynamic invoice.
2. Merchant generates KHQR code by using KHQR SDK which is provided by the **National bank of Cambodia** or they can generate QR by themselves following the QR EmvCo standard at their back-end then respond to the client side via API.
3. Merchant displays QR to the user along with expiration time.
4. Users scan QR code via the Bakong/Bank app that supports KHQR (In general all Bakong members support it).
5. Users confirm payment.
6. Merchant sets a time interval to request API to check transaction of customer's payment at Bakong side in specific duration by following QR expiration time (it shall complete within 10s at most or the merchant can go back to manual check with their customer by asking to show the receipt within Bakong/Bank app).
7. Bakong response payment status to merchant such as success, failed.
8. Merchant displays a message or push notification of the payment to the customer.

System Interaction Flow

This section provides high-level information about the system interaction by 2 options as below

➤ Generate QR at merchant back-end side:



1. Users go to the merchant app/website/POS to make payment then merchants call their own API to create QR code.
2. Merchants back-end generate EMVCo QR code with unique transaction id, timestamp and any additional data if needed. Then response data back to the merchant client such as QR code image URL, transaction id, QR expiration datetime , Channel name...

Note:

- Expiration datetime can be defined based on integrator business.
- Transaction Id must be unique for every time you generate QR code.

2.1 The merchant client displays QR code.

3. Users use the Bakong/Banking app to scan QR and make payments.
4. Merchant back-end sets a time interval in a specific second to call API [Get transaction with MD Hash](#) to the Bakong system to check payment results.

Note:

- MD5 hash generated from QR string by using md5 encryption method.
- The QR code time-out shall not exceed 10 mins. Or you may need to go back to the manual process by checking if the customer money is deducted or not? by asking the customer to show the transaction on their mobile app.

4.1 Bakong Response payment result to system integrator.

- If status is **success** or **failed** then the integrator back-end updates transaction status and can perform any additional action at their side such as print-out the bill or display a message to the user.
- else if status is **not found** and QR **doesn't expire** then still keep the process checking payment status to Bakong with api mentioned in step 4 above.
- else if status is **not found** and QR **expired** then integrator back-end updates transaction status to time-out.

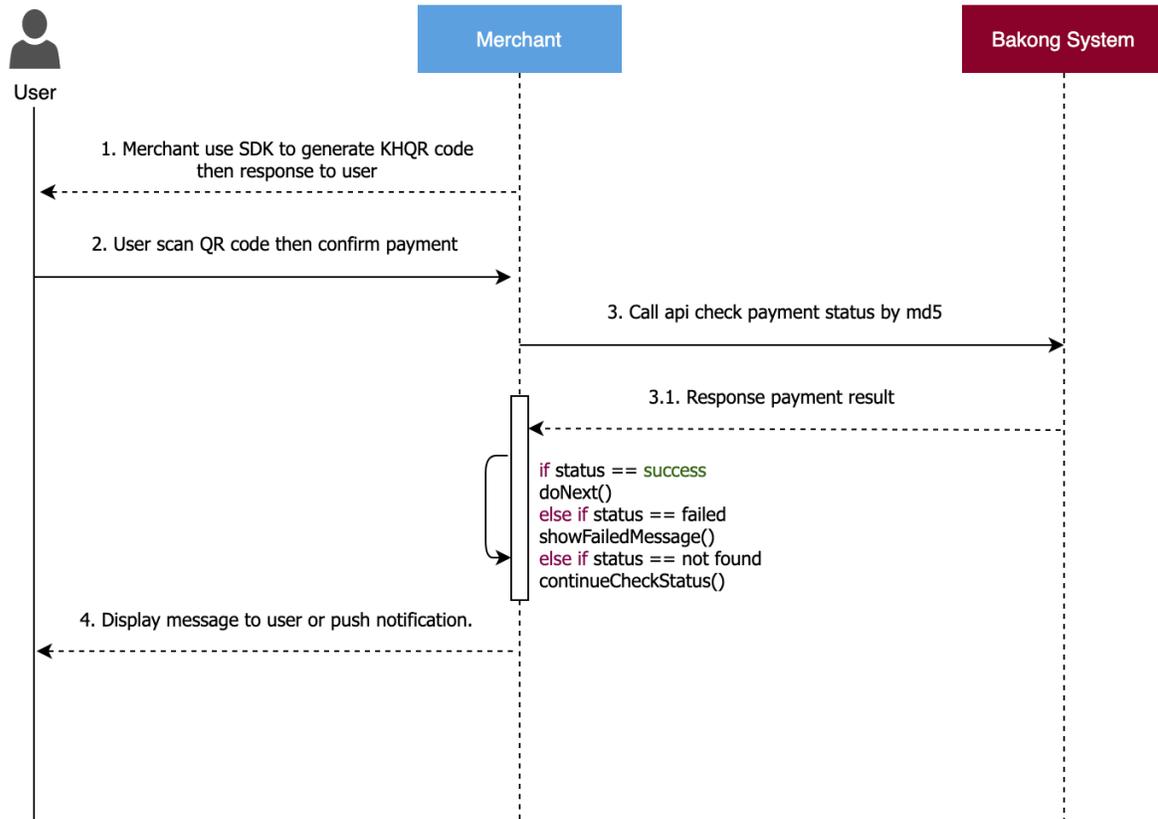
After getting a response from the Bakong side, the merchant back-end can trigger their merchant clients.

5. If there is no trigger function, then merchant clients have to call API to check the payment status to their own back-end in every specific second.

5.1 Merchant back-end responses request results to their merchant client.

5.2 Merchant client displays messages or push notifications to users based on request results.

➤ **Generate QR at the client side by using KHQR SDK:**



1. Users go to the merchant store/app/website/POS to make payment then merchants use KHQR sdk which is provided by **National Bank of Cambodia** to generate and display QR code to users.

Note:

- For display QR code should have an expiration time for making payment.
- QR expiration time can be defined based on integrator business.

2. Users use the Bakong/Banking app to scan QR and make payments.
3. Merchant sets a time interval in a specific second to call API [Get transaction with MD Hash](#) to Bakong system to check payment result.

Note:

- The request timer for calling API check payment status can be defined by your own.
- The QR code time-out which shall not exceed 10 mins. Or you may need to go back to the manual process by checking if the customer money is deducted or not? by viewing the screen of the customer mobile app.

- 3.1. Bakong response payment results to merchants.
- If status is **success** or **failed** then the merchant updates transaction status then can perform any additional action at their side such as display message to user.
 - else if status is **not found** and QR **doesn't expire** then still keep the process checking payment status to Bakong with api mentioned in step 4 above.
 - else if status is **not found** and QR **expired** then merchant updates transaction status and shows a message to users to retry payment again with new QR code.
4. The merchant displays messages or push notifications to users based on request results.

Notice:

- The API check transaction by md5 applies for dynamic KHQR only. For the static KHQR to check transaction status, the bank can check directly to db PG.
- The transaction will go direct to the Bakong account Id (user bakong account or FIs KHQR settlement account) only.
- For credit to the merchant/customer bank account (CASA) is on the bank side implementation (realtime or T+1, T+2,...)
- For sending notification to the user is on the third party side.

Integration Process

Prerequisites:

Before starting integration with Bakong QR code payment function, Please ready for some follow preparations as below:

- Bakong account which is registered under a commercial bank.
- Read API specification and integration flow. Be cleared about the process.

KHQR Generation:

For generating KHQR without using SDK please refer to this reference [KHQR Guideline](#).

For generating KHQR using SDK please refer to [KHQR SDK document](#).

Implementing API:

How to guide:

- **Generate Token** (New user) :

To generate the token to request API check transaction status, Please go to Bakong Open API website via:

- SIT: <https://sit-api-bakong.nbc.org.kh/>
- Production: <https://api-bakong.nbc.org.kh/>

then follow steps as below:

1. Go to the register page.
2. Fill in all information such as:
 - Project name: The project name which is going to use this API.
 - Organization name: Your organization name. example: ABA, Acleda,...
 - Valid email: The valid email that NBC will send a token to this address.
3. Click the "Register" button.

4. NBC will send a confirmation email to your registered email address.
5. Copy code and use it to verify steps.
6. After verification success, you will get another mail of access token.

★ **Token Life Cycle:** Token will expire in **90 days**.

 **Note:** Before **3 days** of token expiration, NBC will send email alerts to users to generate new token and replace them at their side.

- **Renew Token**

To Renew token when the previous token expired, Please go to Bakong website via this link

- SIT: <https://sit-api-bakong.nbc.org.kh/>
- Production: <https://api-bakong.nbc.org.kh/>

then follow steps as below:

1. Go to the register page.
2. Click on the renew button.
3. Enter the email which you use to register.
4. NBC will send a new token to your email address.
5. Replace new token when request to API check transaction status.

 **Note:** For renewing token can be generated via API as well. Please find the document API in the Bakong Open API website in the link above.

Request Check Transaction Status API

Name	Method	URL	Description
Check transaction by md5	POST	{{url}}/v1/check_transaction_by_md5	Checking the transaction using 32 length string hash.
Check transaction by hash	POST	{{url}}/v1/check_transaction_by_hash	Checking the transaction by hash

HTTP code status

Code	Status
200	Accepted
400	Bad Request
401	Unauthorized
404	Not Found
500	Internal Server Error

Custom code status

Response Code	Status
0	Success
1	Failed

Check transaction by md5

API Specifications

HTTP Method	POST
Name	Check transaction by md5
Endpoint	{{url}}/v1/check_transaction_by_md5
Description	

Request

- Request Parameters

Parameter name	Type	Mandatory	Description
Header parameters			
Authorization	String	1	Must be "Bearer <access token>"
Content-Type	String	1	Must be "application/json"
Body Parameters			
md5	String	1	md5 hash got from QR string encryption. Validation <ul style="list-style-type: none">- Min: 1- Max: 255

Response

- Response Parameter

Parameter	Data Type	Mandatory	Description
responseCode	int	1	
responseMessage	string	1	
data	object	1	

- Data object

Parameter	Data Type	Mandatory	Description
hash	string	1	
fromAccountId	string	1	
toAccountId	string	1	
currency	string	1	
amount	float	1	
description	string	0	

- Sample Response data

Success

```
{
  "responseCode": 0,
  "responseMessage": "Getting transaction successfully.",
  "data": {
    "hash": "8465d722d7d5065f2886f0a474a4d34dc6a7855355b611836f7b6111228893e9",
    "fromAccountId": "rieu_dhqj_1984@devb",
    "toAccountId": "bridge_account@devb",
    "currency": "USD",
    "amount": 1.0,
    "description": "testing bakong generator"
  }
}
```

Failed

```
{
  "data": null,
  "errorCode": 1,
  "responseCode": 1,
  "responseMessage": "Transaction failed."
}
```

Not Found

```
{
  "data": null,
  "errorCode": 1,
  "responseCode": 1,
  "responseMessage": "Transaction could not be found. Please check and try again."
}
```

Check transaction by hash

API Specifications

HTTP Method	POST
Name	Check transaction by hash
Endpoint	{{url}}/v1/check_transaction_by_hash
Description	

Request

- Request Parameters

Parameter name	Type	Mandatory	Description
Header parameters			
Authorization	String	1	Must be "Bearer <access token>"
Content-Type	String	1	Must be "application/json"
Body Parameters			
hash	String	1	Validation <ul style="list-style-type: none">- Min: 1- Max: 255

Response

- Response Parameter

Parameter	Data Type	Mandatory	Description
responseCode	int	1	
responseMessage	string	1	
data	object	1	

- Data object

Parameter	Data Type	Mandatory	Description
hash	string	1	
fromAccountId	string	1	
toAccountId	string	1	
currency	string	1	
amount	float	1	
description	string	0	

- Sample Response sample

```

Success
{
  "responseCode": 0,
  "responseMessage": "Getting transaction successfully.",
  "data": {
    "hash": "8465d722d7d5065f2886f0a474a4d34dc6a7855355b611836f7b6111228893e9",
    "fromAccountId": "rieu_dhqj_1984@devb",
    "toAccountId": "bridge_account@devb",
    "currency": "USD",
    "amount": 1.0,
    "description": "testing bakong generator"
  }
}

```

Failed

```
{  
  "data": null,  
  "errorCode": 1,  
  "responseCode": 1,  
  "responseMessage": "Transaction failed."  
}
```

Not Found

```
{  
  "data": null,  
  "errorCode": 1,  
  "responseCode": 1,  
  "responseMessage": "Transaction could not be found. Please check and try again."  
}
```