LAILAO
PAYMENT SUPPORT

Lailao Payment Gateway

Software Development Kit (SDK)
Web API v1.0.1

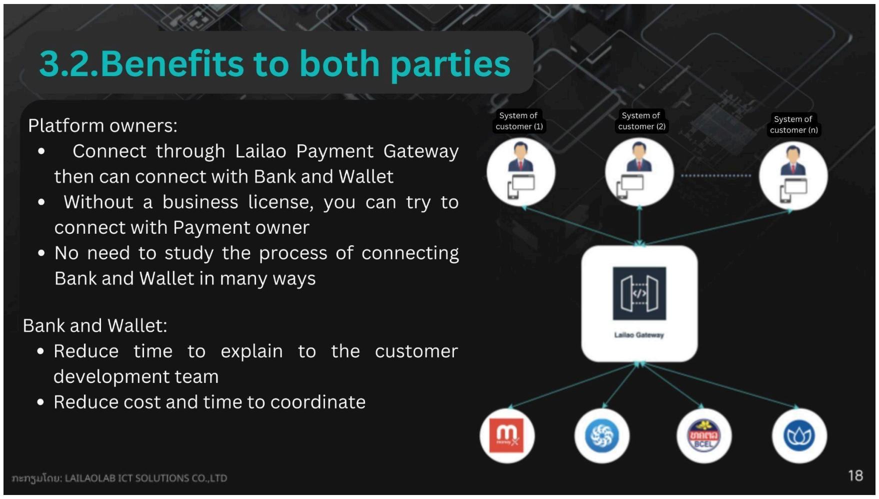
Content

1. Introduction

Welcome to Lailao Payment Gateway! In this section, you can find all the information you need to study and use our service, or Lailao Payments API.
The Lailao Payment Gateway is built to address the challenges faced by startup platforms and various legacy systems. It serves as a bridge, transforming these systems to seamlessly connect with the banking system or wallet systems existing in the Lao PDR.

We, at Lailao Payment Gateway, have developed an intermediary to facilitate and support the connection technology for platform owners who wish to transition into FinTech, enabling them to connect to bank and wallet services easily and quickly.

This guide is intended for developers. If you are not a developer but are interested, you can find an overview of the content of our articles and models provided below.



3.2.Benefits to both parties

Platform owners:
- Connect through Lailao Payment Gateway then can connect with Bank and Wallet
- Without a business license, you can try to connect with Payment owner
- No need to study the process of connecting Bank and Wallet in many ways

Bank and Wallet:
- Reduce time to explain to the customer development team
- Reduce cost and time to coordinate

ກະກຽມໂດຍ: LAILAOLAB ICT SOLUTIONS CO.,LTD

18

1.1 What is Payment Gateway?

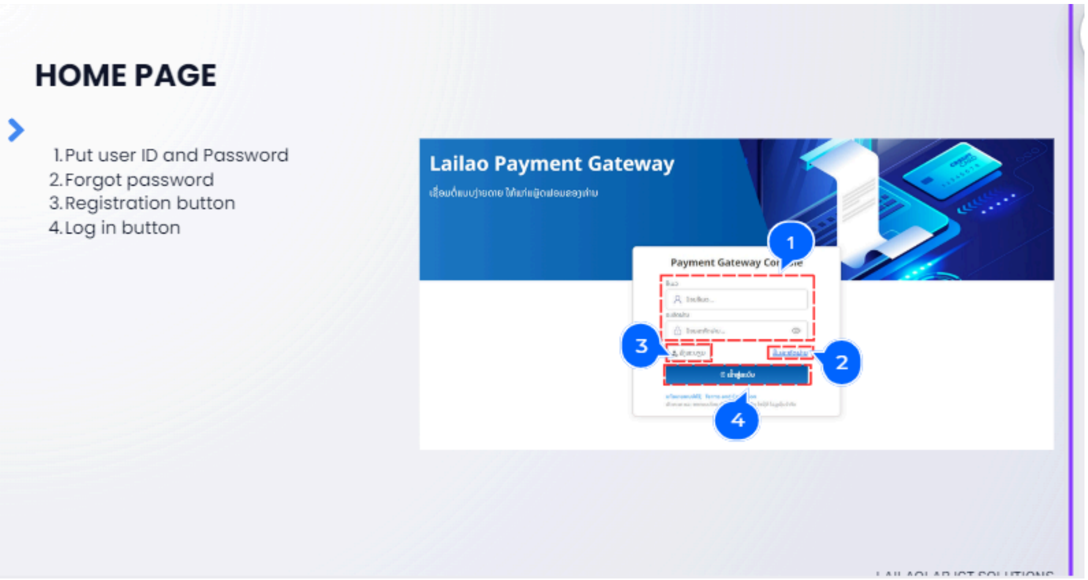
Lailao Payment Gateway is a PCI-certified payment gateway featuring an easy-to-use management dashboard and a straightforward REST API model, preferred by numerous API developers.

We prioritize giving you the flexibility to choose your preferred payment process, ensuring it's secure and direct, whether in the test environment or the production environment.

2 Lailao Payment Connection

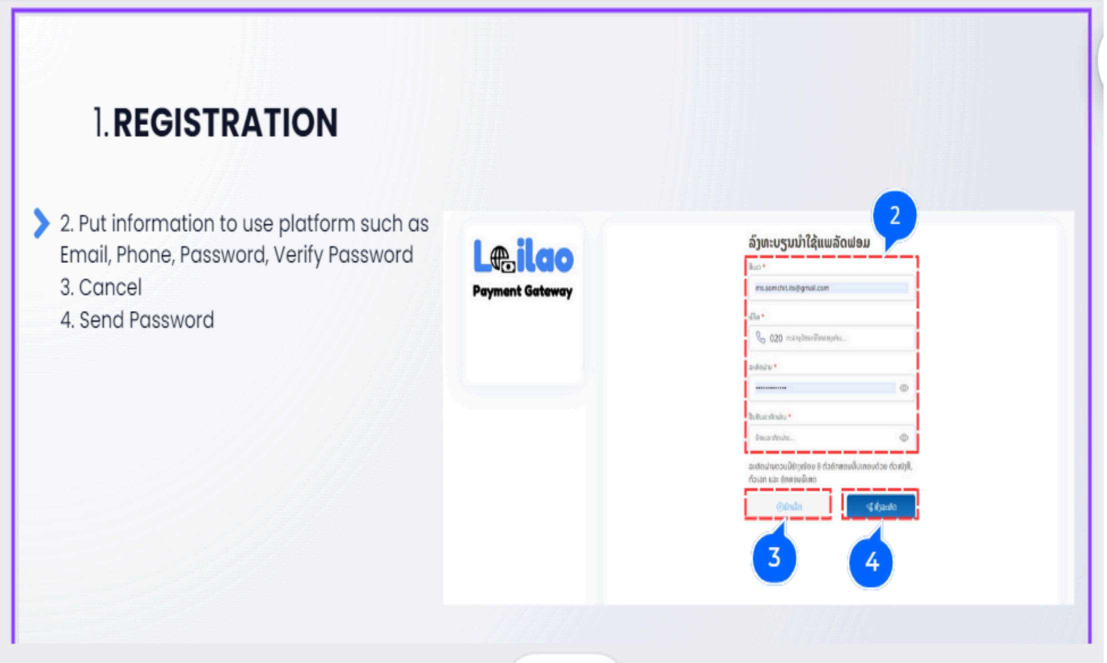
2.1. Getting Started Registration

First, click on the link https://portal.lailaolab.la/ to access Lailao payment gateway portal website it will lead you to the home page, then click the “**3. registration button**” as the steps below:



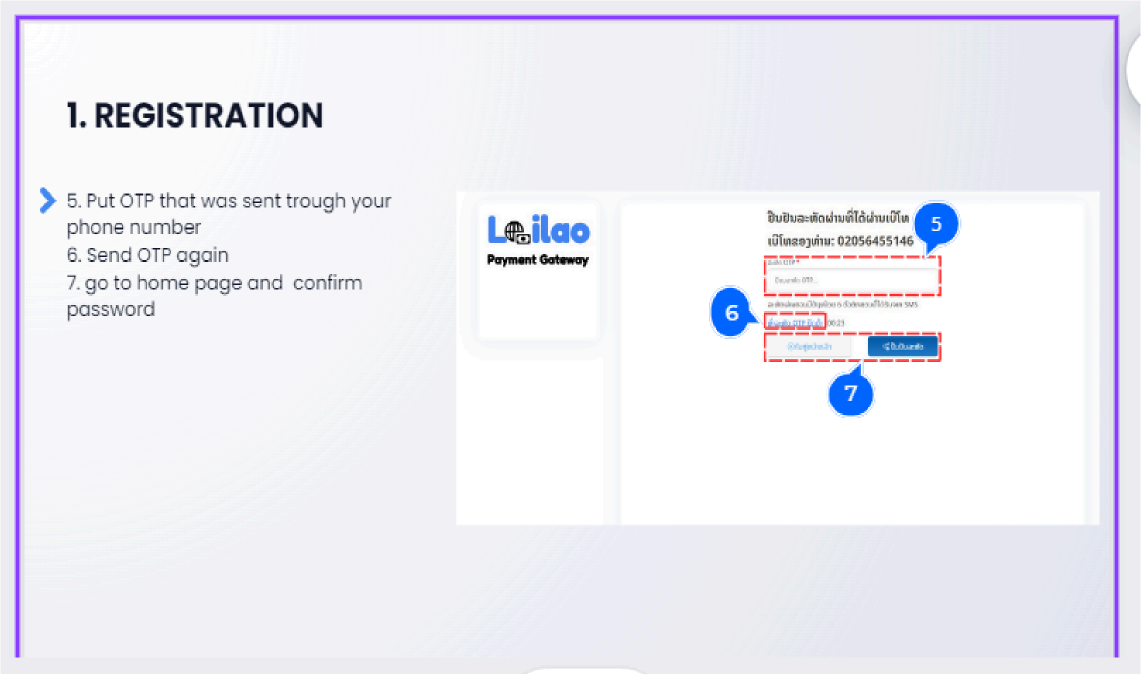
HOME PAGE

1. Put user ID and Password
2. Forgot password
3. Registration button
4. Log in button



1.REGISTRATION

2. Put information to use platform such as Email, Phone, Password, Verify Password
3. Cancel
4. Send Password



1. REGISTRATION

5. Put OTP that was sent trough your phone number
6. Send OTP again
7. go to home page and confirm password

ຢືນຢັນລະຫັດຜ່ານທີ່ໄດ້ຜ່ານເບີໂທ
ເບີໂທຂອງທ່ານ: 02056455146

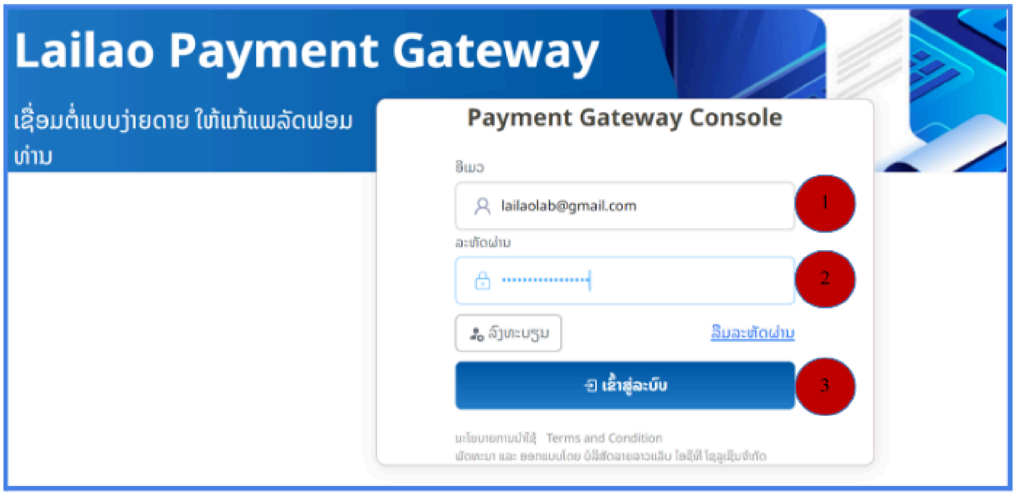
ລະຫັດ OTP *

ປ້ອນລະຫັດ OTP...

ສົ່ງລະຫັດ OTP ອີກຄັ້ງ 00:23

After finishing registration, it will go to the home page.

1. Enter your Email
2. Password
3. Log in



Lailao Payment Gateway

ເຊື່ອມຕໍ່ແບບງ່າຍດາຍ ໃຫ້ແກ່ແພລັດຟອມທ່ານ

Payment Gateway Console

ອີເມວ

lailaolab@gmail.com

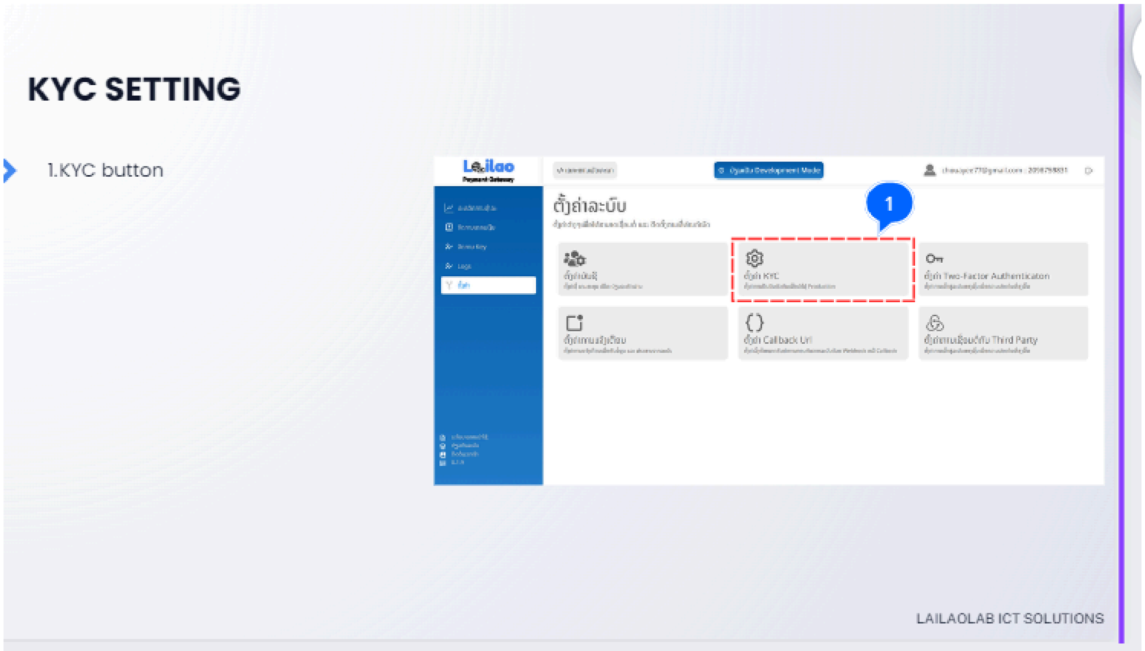
ລະຫັດຜ່ານ

••••••••••••••••

ນະໂຍບາຍການນຳໃຊ້ Terms and Condition
ພັດທະນາ ແລະ ອອກແບບໂດຍ ບໍລິສັດລາຍລາວແລັບ ໄອຊີທີ ໂຊລູເຊີນຈຳກັດ

2.1.1 Getting Started KYC

To use the Lailao Payment Gateway in Production Mode, the platform owner must complete the KYC (Know Your Customer) process by submitting information about the platform's services to the Lailao Payment Gateway first. The KYC procedure is as follows:

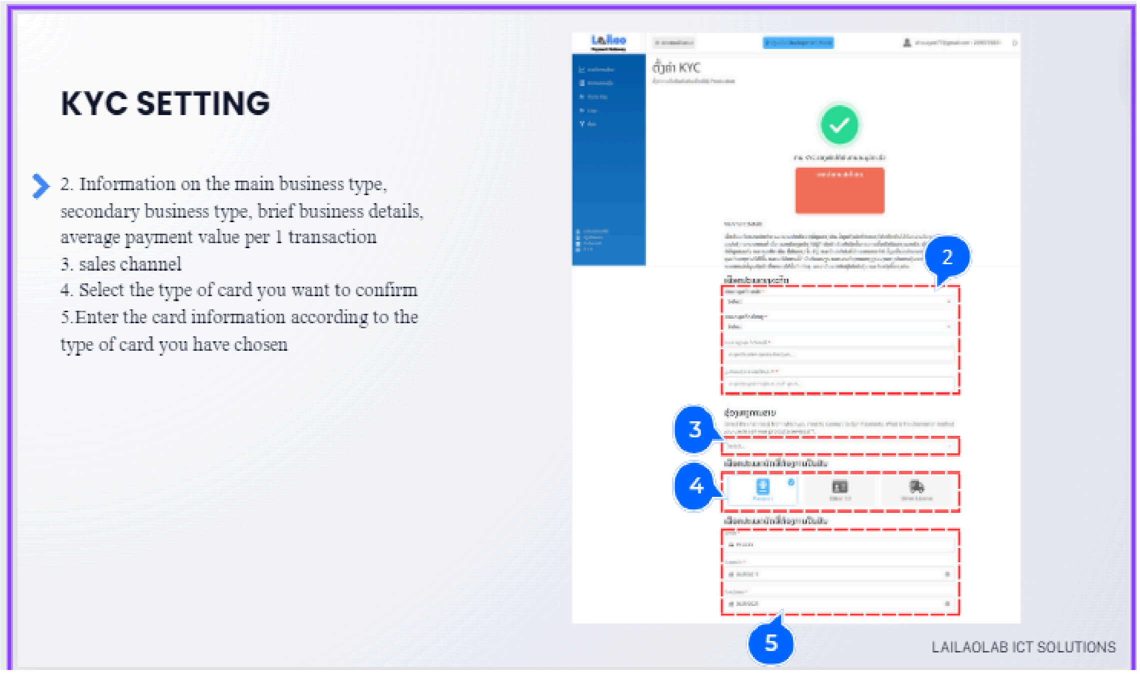


KYC SETTING

1.KYC button

LAILAOLAB ICT SOLUTIONS

Lailao
ຕັ້ງຄ່າລະບົບ
1
ຕັ້ງຄ່າ KYC
ຕັ້ງຄ່າ Two-Factor Authentication
ຕັ້ງຄ່າ Callback Url
ຕັ້ງຄ່າການເຊື່ອມຕໍ່ກັບ Third Party



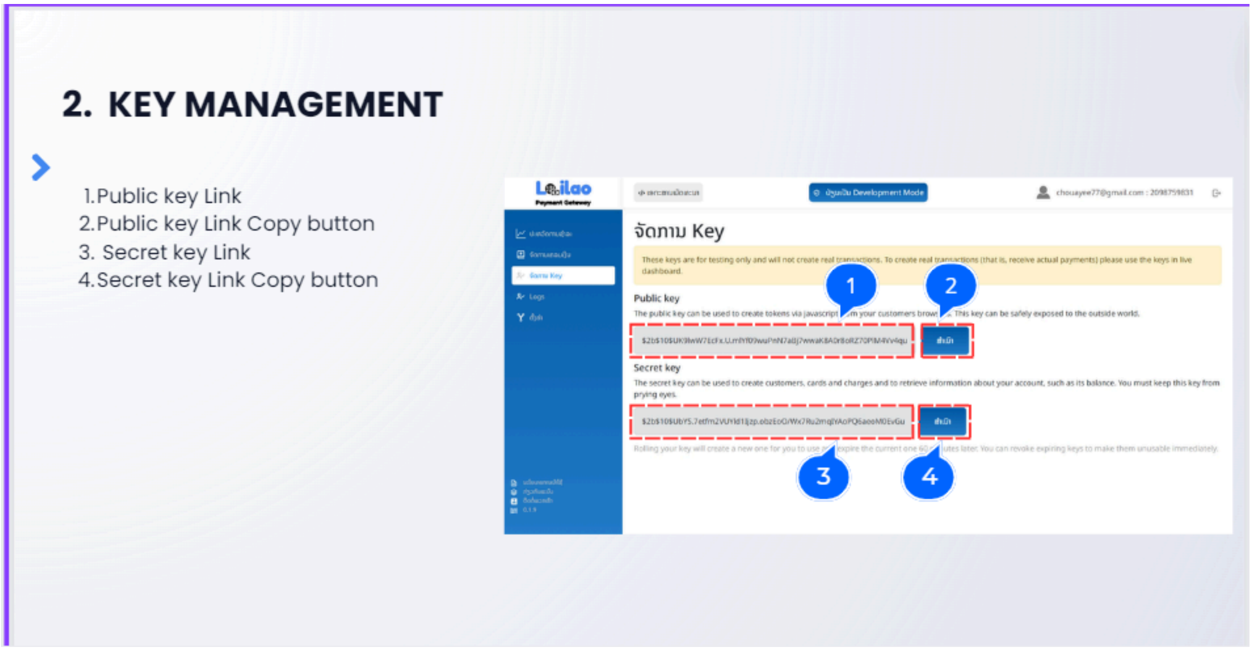
KYC SETTING

2. Information on the main business type, secondary business type, brief business details, average payment value per 1 transaction
3. sales channel
4. Select the type of card you want to confirm
5.Enter the card information according to the type of card you have chosen

LAILAOLAB ICT SOLUTIONS

2.1.2 Getting Started Key Management

After receiving confirmation from Lailao Payment Gateway, the platform owner will receive the Secret Key used for Production Mode and can connect to provide services



2. KEY MANAGEMENT

1. Public key Link
2. Public key Link Copy button
3. Secret key Link
4. Secret key Link Copy button

ຈັດການ Key

These keys are for testing only and will not create real transactions. To create real transactions (that is, receive actual payments) please use the keys in live dashboard.

Public key

The public key can be used to create tokens via javascript m your customers brow . This key can be safely exposed to the outside world.

$2b$10$UK9lwW7EcFx.U.mIYf09wuPnN7aBj7wwaK8A0r8oRZ70PIM4Vv4qu

Secret key

The secret key can be used to create customers, cards and charges and to retrieve information about your account, such as its balance. You must keep this key from prying eyes.

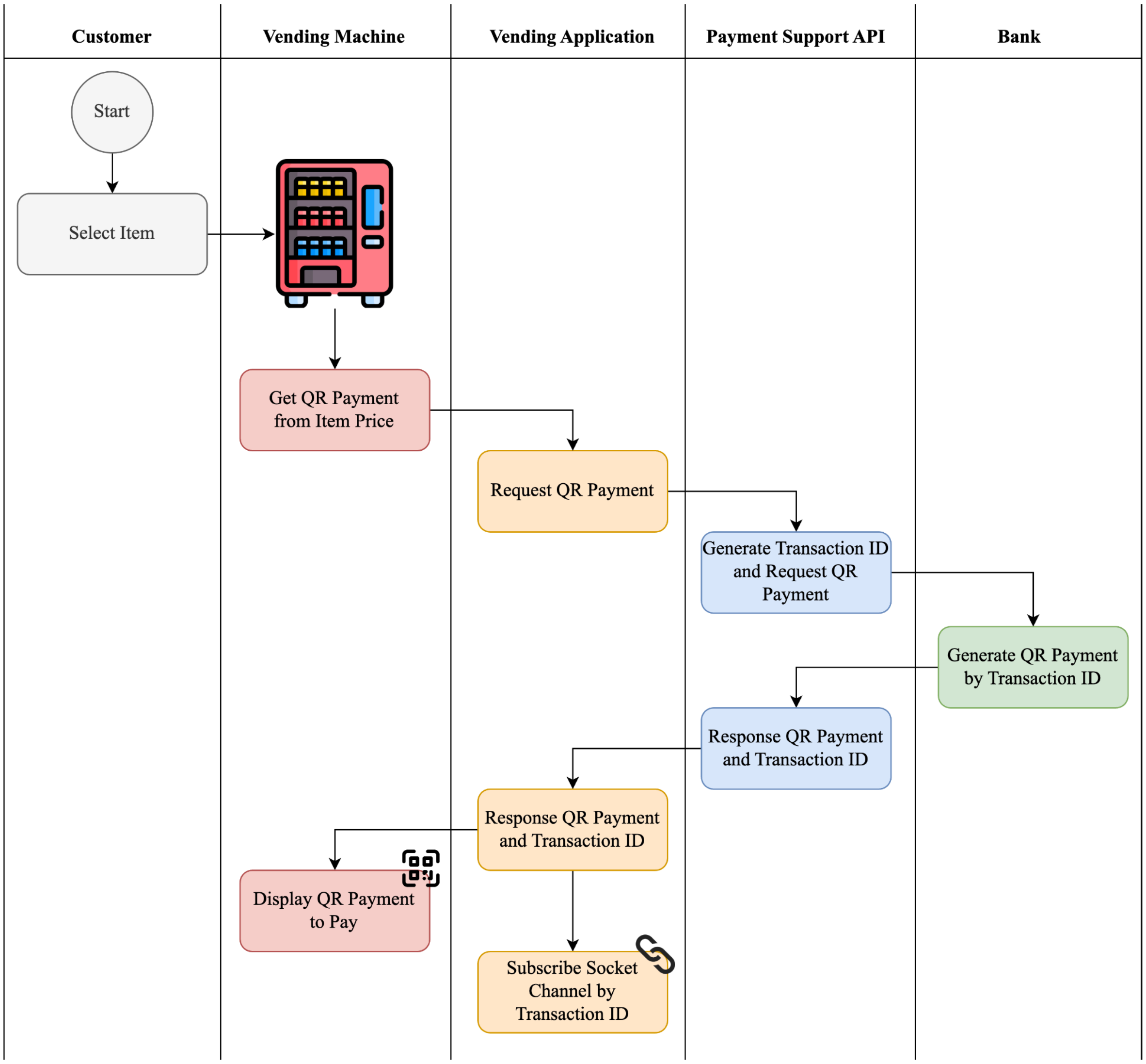
$2b$10$UbY5.7etfm2VUYld1Ijzp.obzEoG/Wx7Ru2mqIYAoPQ6aooM0EvGu

Rolling your key will create a new one for you to use expire the current one 60 utes later. You can revoke expiring keys to make them unusable immediately.

3. Connect to Web API

3.1. Generate QR

To make a payment through the Bank Platform, it is necessary to create a QR String for connecting the bank to use the Mobile Banking app to make the connection. Currently, we can connect to 3 banks in Laos such as: BCEL, Joint Development Bank (JDB), Indochina Bank (IB). The way to connect to create a QR is as follows


Customer
Vending Machine
Vending Application
Payment Support API
Bank
Start
Select Item
Get QR Payment from Item Price
Request QR Payment
Generate Transaction ID and Request QR Payment
Generate QR Payment by Transaction ID
Response QR Payment and Transaction ID
Response QR Payment and Transaction ID
Display QR Payment to Pay
Subscribe Socket Channel by Transaction ID

BCEL_URL: https://payment-gateway.lailaolab.com/v1/api/payment/generate-bcel-qr
JDB_URL: https://payment-gateway.lailaolab.com/v1/api/payment/generate-jdb-qr
IB_URL: https://payment-gateway.lailaolab.com/v1/api/payment/generate-ib-qr
METHOD: POST
HEADER: secretKey: <PAYMENT_GATEWAY_SECRET>

3.1.1. Request Body

Field	Type	Description
amount	Number	Amount to be created for a transaction
description	String	Payment description (Note: should not include “” space)

Example:

```
{
   "amount": 10000,
   "description": "buy products"
}
```

3.1.2 Response Data

Field	Type	Description
message	String	A string indicating the outcome of the API call
transactionId	String	A unique identifier for the transaction
qrCode	String	A string containing the QR code data
link	String	A deep link that can be used to initiate the payment process in a mobile application

Example:

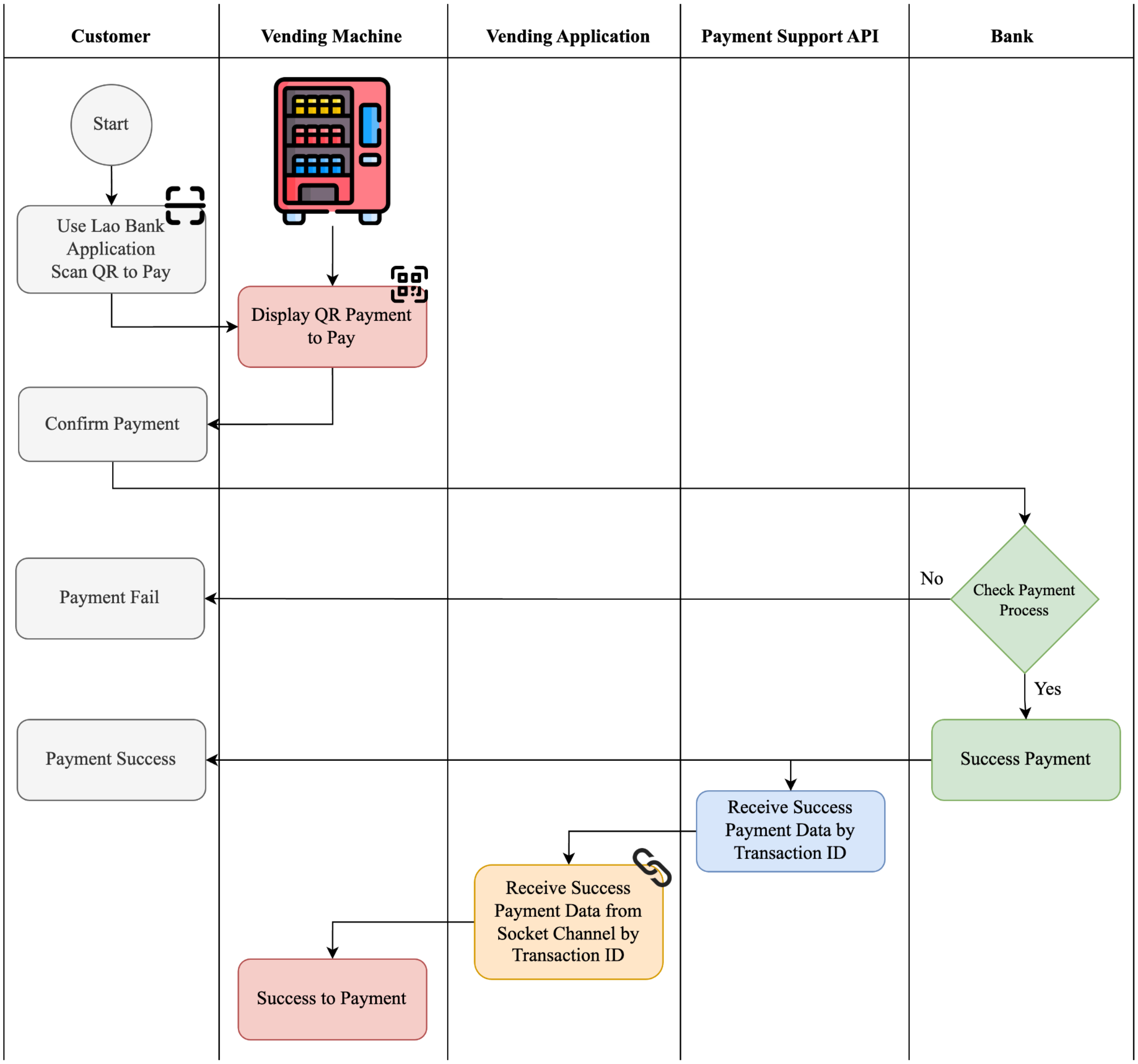
```
{
   "message": "SUCCESSFULLY",
   "transactionId": "8cc876b4-a4af-4886-81f1-3890453eb556",
   "qrCode":
"00020101021133730004BCEL0106ONEPAY0216mch6541c0373ede30314202403271909590513CLOSEWHENDONE53034185405100005803VTE6002LA625305368cc876b4-a4af-4886-81f1-3890453eb5560809Buy Pants630440FF",
   "link":
"onepay://qr/00020101021133730004BCEL0106ONEPAY0216mch6541c0373ede30314202403271909590513CLOSEWHENDONE53034185405100005803VTE6002LA625305368cc876b4-a4af-4886-81f1-3890453eb5560809Buy Pants630440FF"
}
```

3.2. Check Payment Record

We have 2 solutions to check the payment record:

1). Subscription Payment

To receive the payment information when there is a payment transaction, the platform must subscribe to the Payment Gateway. You can receive the Subscription Payment Data in real time within 1 to 2 seconds, which must be connected in the SocketIO format according to the link and must subscribe to the event as below:


Customer
Vending Machine
Vending Application
Payment Support API
Bank
Start
Use Lao Bank Application Scan QR to Pay
Display QR Payment to Pay
Confirm Payment
Check Payment Process
No
Yes
Payment Fail
Payment Success
Success Payment
Receive Success Payment Data by Transaction ID
Receive Success Payment Data from Socket Channel by Transaction ID
Success to Payment

URL: https://payment-gateway.lailaolab.com/?key=<PAYMENT_GATEWAY_SECRET>

Event Name: join::<PAYMENT_GATEWAY_SECRET>

Field	**Type**	**Description**
PAYMENT_GATEWAY_SECRET	String	Production Secret key that retrieved from lailao payment gateway portal https://portal.lailaolab.la/private/key-management

Response Data when socket triggered

When processing payments via banking apps, Socket Server will send the information received from the bank back to the Platform, which will include the following parameters that can be used in the next stage of the platform's work process.

Field	**Type**	**Description**
message	String	A string indicating the status of the API call. For successful calls
refNo	String	A unique reference number assigned to the transaction
exReferenceNo	String	An external reference number associated with the transaction
merchantName	String	The name of the merchant involved in the transaction
memo	String	A brief description of the transaction, e.g., `PAYMENT`
txnDateTime	String	The date and time when the transaction was processed, in the format YYYY-MM-DD HH:MM:SS
txnAmount	Number	The amount that was transacted
billNumber	String	A unique bill number associated with the transaction
sourceAccount	String	The account number from which the funds were sourced
sourceName	String	The name associated with the source account (may be empty)

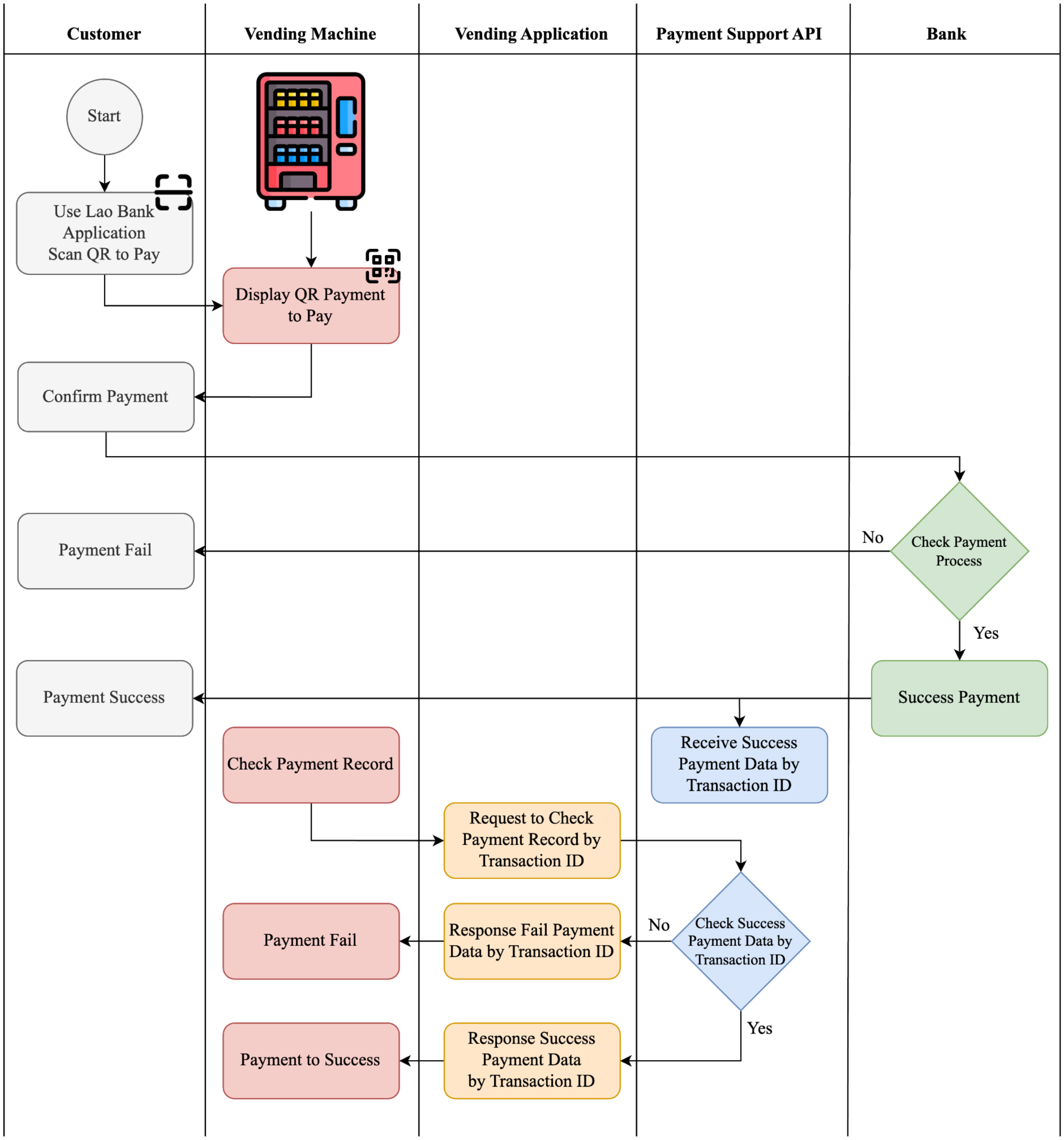
sourceCurrency	String	The currency in which the transaction was made
userId	ObjectId	A unique identifier for the user who initiated the transaction
status	String	The current status of the transaction
transactionId	String	A unique identifier for the transaction, repeated for convenience

Example:

```
{
  "message": 'SUCCESS',
  "refNo": '001LNMI811581649255',
  "exReferenceNo": 'BONEN5JL999C3ZLN',
  "merchantName": 'LAILAOLAB ICT SOLUTIONS CO.,LTD',
  "memo": 'PAYMENT',
  "txnDateTime": '2024-03-28 22:30:50',
  "txnAmount": 100000,
  "billNumber": 'VobAkGl0gVKrVecIbo2ogYrIk',
  "sourceAccount": '1600120000000004810001',
  "sourceName": '',
  "sourceCurrency": 'LAK',
  "userId": new ObjectId("656fd8886b411beab18eca79"),
  "status": 'PAYMENT_COMPLETED',
  "transactionId": 'VobAkGl0gVKrVecIbo2ogYrIk'
}
```

2). Request to Check from API Route

To receive the payment information when there is the payment transaction, the platform can request to check payment record API with using transaction ID for check each events as below:


Customer
Vending Machine
Vending Application
Payment Support API
Bank
Start
Use Lao Bank Application Scan QR to Pay
Display QR Payment to Pay
Confirm Payment
Check Payment Process
No
Yes
Payment Fail
Success Payment
Payment Success
Receive Success Payment Data by Transaction ID
Check Payment Record
Request to Check Payment Record by Transaction ID
Check Success Payment Data by Transaction ID
No
Yes
Response Fail Payment Data by Transaction ID
Payment Fail
Response Success Payment Data by Transaction ID
Payment to Success

Request Data

URL: https://payment-gateway.lailaolab.com/v1/api/payment/transaction/<TRANSACTION_ID>
Method: GET
Headers: secretKey: <your_secret_key>

Field	Type	Description
TRANSACTION_ID	String	A unique identifier for the transaction, repeated for convenience

Response Data

Field	Type	Description
message	String	A string indicating the status of the API call. For successful calls
origin	String	referring to the originating bank
qrString	String	A string representing the QR code data, which can be scanned for payment processing.
appLink	String	A deep link that can be used to initiate the payment process in a mobile application
transactionId	String	A unique identifier for the transaction
amount	Number	The amount that was transacted
description	Number	A string indicating the details of the transaction
status	String	A string indicating transaction status (`PAYMENT_COMPLETED` indicates that the transaction has been successfully paid. `CLOSED` signifies that the transaction remains unpaid.)
user	String	A unique identifier of user in the Payment Support System
createdAt	DateTime	The timestamp indicating when the transaction was created

updatedAt	String	The timestamp indicating the last time the transaction details were updated

Example:

```
{
    "message": "SUCCESSFULLY",
    "data": {
        "origin": "JDB",
        "qrString":
"00020101021238680016A005266284662577010832170418020300203 25GDL8VBRF7
PNYEEQLK1HAVNYUE52045251530341854051200058 02LA5925LAILAOLABICTSOLUTIO
NSCO.,6009Vientiane62920125HGzVkdwHbiDkrmGRZVNVvrl3102010030100401005
0100609LAILAOLAB0716MAKE_JDB_PAYMENT080100901063047AD4",
        "appLink":
"https://jdbbank.com.la/yespay/00020101021238680016A005266284662577 01
0832170418020300203 25GDL8VBRF7PNYEEQLK1HAVNYUE5204525153034185405120 0
05802LA5925LAILAOLABICTSOLUTIONSCO.,6009Vientiane62920125HGzVkdwHbiDk
rmGRZVNVvrl310201003010040100501 00609LAILAOLAB0716MAKE_JDB_PAYMENT080
100901063047AD4",
        "transactionId": "HGzVkdwHbiDkrmGRZVNVvrl31",
        "amount": 12000,
        "description": "JDB Payment",
        "status": "PAYMENT_COMPLETED",
        "user": "65c0a98047e5a1805e626add",
        "createdAt": "2024-05-07T02:41:14.203Z",
        "updatedAt": "2024-05-07T02:49:47.694Z"
    }
}
```

4. Example Code

Here is an example of Lailao Payment Gateway connection for generate QR String by using NodeJS

```
const paymentGateway = async (paymentMethod, amount, description) => {
    try {
        let paymentGatewaySecrete = process.env.PAYMENT_GATEWAY_SECRET;
        let apiUrl =
"https://payment-gateway.lailaolab.com/v1/api/payment/generate-bcel-qr"

        if (paymentMethod === 'JDB') {
            apiUrl =
'https://payment-gateway.lailaolab.com/v1/api/payment/generate-jdb-qr'
        } else if (paymentMethod === 'IB') {
            apiUrl =
'https://payment-gateway.lailaolab.com/v1/api/payment/generate-ib-qr'
        }

        const _result = await axios.post(apiUrl,
            {
                "amount": amount,
                "description": description || 'description'
            },
            {
                headers: {
                    "secretKey": paymentGatewaySecrete,
                },
            }
        )
        return _result
    } catch (error) {
        console.log({ error })
        return null
    }
}
```

Here is an example of socketio connection by using NodeJS

```
 const onSubscribePaymentGateway = (models) => {
  try {
    const _socketPaymentUrl = "https://payment-gateway.lailaolab.com/?key=" +
process.env.PAYMENT_GATEWAY_SECRET
    const socket = io(_socketPaymentUrl);

    if (socket.connected) {
      console.log('Socket is already connected.');
      return; // No need to connect again
    }
    // Connect to the server
    socket.on('connect', () => {
      console.log('Connected to the payment gateway server!');
      // Subscribe to a custom event
      socket.on('join::' + process.env.PAYMENT_GATEWAY_SECRET, async (data) => {
        console.log('Data received:', data);
      });
    });
    // Handle the connection error (optional)
    socket.on('connect_error', (error) => {
      console.error('Connection failed:', error);
    });
    return;
  } catch (error) {
    console.log({ error })
  }
}
```