Protocol Rules

The following specifies the rules for calling the API when a third-party merchant accesses KBZPay payment

	description
Transfer Mode	Use HTTPS for secure transactions
Submit Mode	Use POST method
Data Format	Data Format Data submitted and returned is in json format
Char Encoding	Use UTF-8 character encoding
Signature Algorithm	SHA256
Signature Requirement	Signature-checking is required for requesting and receiving data. For more information, see Section 1.3.1 Signature Algorithm.
Certificate Requirement	A certificate is required for calling the Submit Refund API or Reverse API.
Logic Judgment	Determine 'result' field in response first, then 'code' filed for business result.

Parameter Specifications

1. Payment Amount The currency type for transaction is MMK (Myanmar Kyat) by default. The unit used in payment amount is Ks. The amount can be without decimals, and at most two decimal places. Temporarily does not support foreign currency transactions
2. Payment Method Payment Method is determined by trade_type field in Unified Order Placing Interface. The options are APP, APPH5, PAY_BY_QRCODE, QRCODE_H5, MINIAPP、 PWAAPP and MICROPAY. APP (In-App Payment): In-App payment also refers to a mobile-based payment in which the Merchant calls the KBZPay payment module by using the open SDK integrated in their mobile-based app to pay for transactions. APPH5/QRCODE_H5 (In-App Web-based Payment): The Payer opens the Merchant's HTML5 pages on their KBZPay and calls the KBZPay payment module via the JSAPI interface to pay their transaction. This mode is applicable in the following scenarios: a. The Payer enters the Merchant's official website via a integrated button in KBZPay APP; b. The Payer enters the Merchant's official website in KBZPay APP by scanning a QR Code. PAY_BY_QRCODE (QR Code Payment): The Merchant generates a transaction QR Code according to the KBZPay Payment Protocol and the Payer goes to "Scan QR Code" in their KBZPay in order to complete payment. This mode is applicable to payments made on websites, physical stores, media advertising, or other scenarios. PWAAPP (PWA Payment): The Payer goes to Merchant's mobile website, and redirect to KBZPay APP to complete payment. MICROPAY (Customer Present QR Code Payment): The Payer shows their

bar code or QR Code on KBZPay 'Scan and Pay' page to the Vendor to scan in order to pay directly. This mode applies to offline payment scenarios

3. Currency Type The list of available currency types: MMK Myanmar Kyat.
4. Timestamp A timestamp is calculated as seconds since 1970/01/01 00:00:00 UTC. Note: Milliseconds should be rounded to seconds (10-digits).
5. Merchant's Order Number The order number for a payment is defined by the merchant and must be unique. We suggest adding a random sequence to the end of the current time in order to create a unique and sequenced number. The original order number is used when initiating a payment again in order to avoid duplicate transactions. However, orders that are paid, closed or revoked cannot be paid again.

Security Specifications

Signature Algorithm

General steps to create a signature:

Step 1: Presume all data sent and received is the set M. Sort non-empty values in M in ascending alphabetical order (i.e. lexicographical sequence), and join them into string A via the corresponding URL key-value format (e.g. key1=value1&key2=value2…). Notes:

- Sort parameter names in ascending alphabetical order based on their ASCII encoded names (e.g. lexicographical sequence);
- All parameters except sign and `sign_type` participate in the signature (the JSONArray format field does not participate in the signature, such as the `refund_info` in the queryorder API response);
- Empty parameter values are excluded in the signature;
- Parameter names are case-sensitive;
- The API interface may add fields, and the extended extension field must be supported when verifying the signature.

Step 2: Add "key= (APP key value) to the end of stringA to get stringToSign, perform SHA256 arithmetic on stringToSign, convert all result chars to upper case, thus get sign's value (signValue). Example:

For the following transferred parameters:

```
{
  "Request": {
    "timestamp": 1536637503,
    "notify_url": "http://test.com/payment/notify",
    "nonce_str": "84525591030856448 1",
    "sign_type": "SHA256",
    "method": "kbz.payment.precreate",
    "sign": "wait_to_generate",
    "version": "1.0",
    "biz_content": {
      "merch_order_id": "201811212009001",
      "merch_code": "100001",
      "appid": "kp12345678998765432 1abcdefghijkl",
      "trade_type": "APPH5",
      "total_amount": "1000",
      "trans_currency": "MMK"
    }
  }
}
```


1. Sort ASCII code of parameter names by lexicographical sequence based on the format of "key=value".

    a. stringA = "appid=kp12345678998765432 1abcdefghijkl&merch_code=100001&merch_order_id=201811212009001&method=kbz.payment.precreate&nonce_str=84525591030856448 1&notify_url=http://test.com/payment/notify&timestamp=1536637503&total_amount=1000&trade_type=APPH5&trans_currency=MMK&version=1.0";

2. Join API keys
   stringToSign = "stringA&key=********b4c****ec**edce**f*a*d";
   sign = SHA256(stringToSign).toUpperCase() =
   "9772777A4AA42FA8B4B4106BEC3348EB7BA2AB2C70BEEA6F17C8051B91F38A1D ";

3. Obtain data to be transferred below:
   a.
```
{
 "Request": {
  "timestamp": 1536637503,
  "notify_url": "http://test.com/payment/notify",
  "nonce_str": "845255910308564481",
  "sign_type": "SHA256",
  "method": "kbz.payment.precreate",
  "sign":
"9772777A4AA42FA8B4B4106BEC3348EB7BA2AB2C70BEEA6F17C8051B91F38A1D
",
  "version": "1.0",
  "biz_content": {
   "merch_order_id": "201811212009001",
   "merch_code": "100001",
   "appid": "kp12345678998765432 1abcdefghijkl",
   "trade_type": "APPH5",
   "total_amount": "1000",
   "trans_currency": "MMK"
  }
 }
}
```

4. KBZPay provides online signature tools for this API. The appkey is used as the signature key. Click the signature address to sign.

Random String Algorithm

nonce_str is included in KBZPay payment API protocols to ensure unpredictability for signatures. We suggest calling the random() function to create a signature and convert its value into a string.

Use Vendor Certificate

A certificate is required for calling the Submit Refund API or Reverse API.

Callback API Security

The callback message of the payment gateway is a confirmation that the payment success. Therefore, we suggest that the notify_url provided by merchants use HTTPS protocol and verify the signature in the callback request.

OrderInfo Signature Example

1. The data of orderinfo is as follows:
    a. prepay_id: KBZ00c25d94271b4d950ec748fdaf20c81d2b154042384
    b. merch_code: 200001
    c. appid: kp419a753459284f72aa76d2ae9d6057
    d. timestamp: 1535165303
    e. nonce_str: 5K8264ILTKCH16CQ2502SI8ZNMTM67VS
2. The character string stringA concatenated by valid parameters in lexicographic order is as follows:
3. stringA="appid=kp419a753459284f72aa76d2ae9d6057&merch_code=200001&nonce_str=5K8264ILTKCH16CQ2502SI8ZNMTM67VS&prepay_id=KBZ00c25d94271b4d950ec748fdaf20c81d2b154042384&timestamp=1535165303"

4. The key value negotiated with the app key and added to the end of the character string stringA is as follows:

stringSign = stringA + "&key=******"

5. The signature generated using the SHA256 algorithm. Fields whose parameter type is list are not involved in signing.

sign = SHA256(stringSign)

6. finally, orderinfo string is as follow:

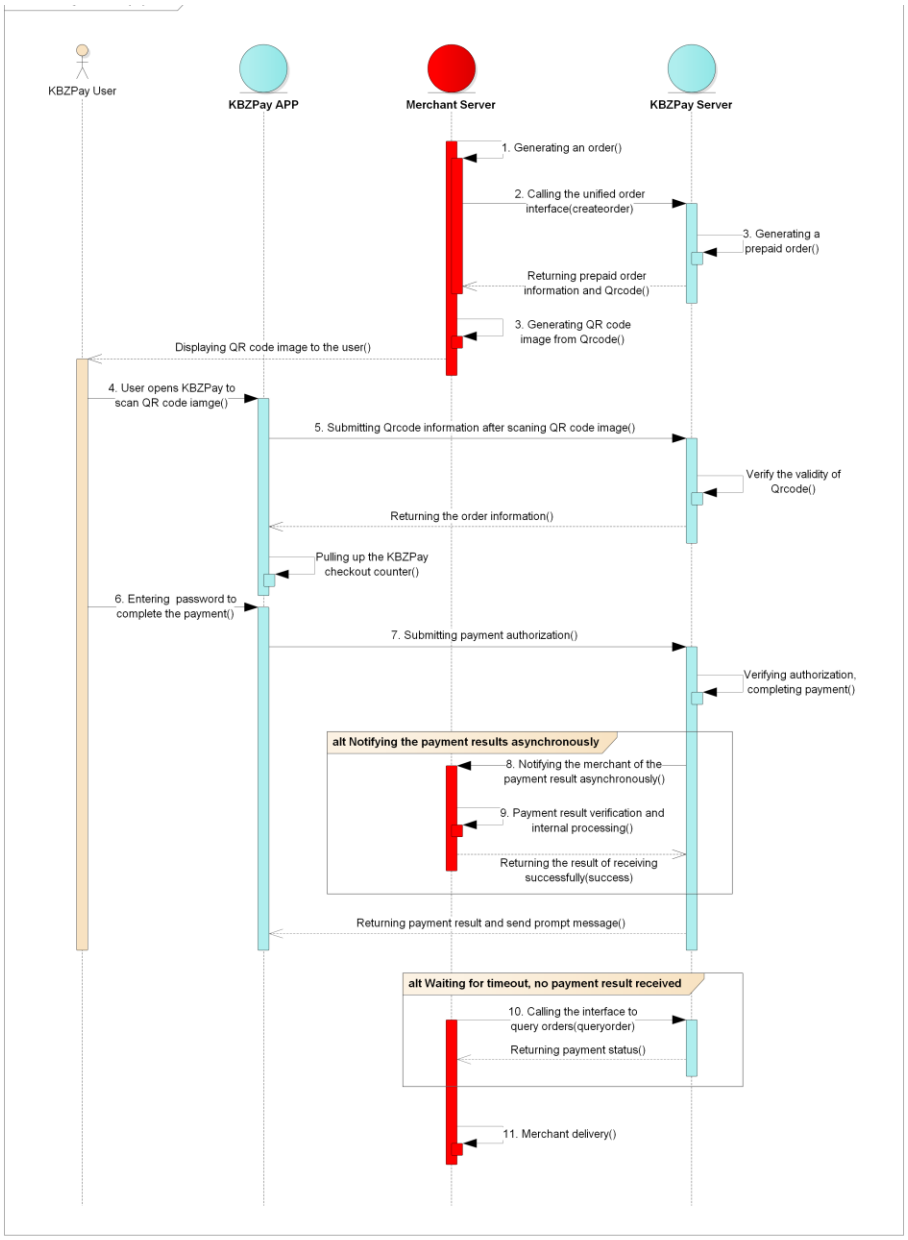
orderinfo="appid=kp419a753459284f72aa76d2ae9d6057&merch_code=200001&nonce_str=5K8264ILTKCH16CQ2502SI8ZNMTM67VS&prepay_id=KBZ00c25d94271b4d950ec748fdaf20c81d2b154042384&timestamp=1535165303&sign=1609E56F220E4D2040F8457F60A6CB3D56861280CDB3434F545469CBAB493573"

QR PAYMENT

QR scan Payment

Scenario Introduction

The merchants display unique QR code for each transaction, the customer scans the QR code by KBZPay app and enter the PIN to complete payment.



Interaction between merchant system and KBZPay system

1. A merchant submits an order based on the offering selected by a customer in the merchant system.
2. The merchant server invokes the order placing interface to place an order.
3. After receiving the request, the KBZPay Server creates a prepaid order and returns the prepaid order information as well as the QR code.

4. The merchant server receives the QR code in response and display the QR code as an image.
5. The customer scan the QR code by KBZPay and send it to KBZPay server.
6. KBZPay server verifies the QR code and KBZPay app start the cash register if the verification passes.
7. The customer enter the PIN to complete the payment.
8. The merchant server receives a callback message (detailed in callback rule) from KBZPay.
9. After receiving the callback message, the merchant server needs to return a success message. Otherwise, the message sending is regarded as failed. In this case, the resending mechanism is available.
10. The merchant can invoke the kbz.payment.queryorder interface to query the payment result.
11. The merchant delive commodity and displays the final payment information to the user.

Unified Order Placing Interface

Use Case

In the payment scenario except Micropay, the merchant system invokes this interface to generate a prepayment transaction order in the payment service background, returns the correct prepayment transaction session ID, and then invokes the KBZPay app for payment based on the specific business scenario.

Certificate Requirement

No certificate is required.

URL

- URL(TEST): `http://api.kbzpay.com/payment/gateway/uat/precreate`
- URL(Prod): `https://api.kbzpay.com/payment/gateway/precreate`

METHOD NAME: `kbz.payment.precreate`

Request Parameters

Common request parameters

Parameter	Type	Required	Example	Description
timestamp	String(13)	Yes	1563161657	Time when a request is sent. UTC timestamp. The unit is second.
notify_url	String(512)	Yes	http://test.payment.com/notify	Specifies the callback address for receiving KBZPay payment notifications if payment is successful.
method	String	Yes	kbz.payment.precreate	Set to 'kbz.payment.precreate', fixed for this interface
nonce_str	String(32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Random character string containing a maximum of 32 characters, including uppercase letters, lowercase letters, digits, but not special characters.
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported. The field is not involved in the signature.
sign	String	Yes	F957B724D1E3704E90BA801BC6CB6DF4FA0898EBF94C67710CB71ABF40BFF8A1	Request signature. For more information, see Signature Algorithm. The field is not involved in the signature.
version	String(64)	Yes	1.0	Interface version number. Only support 1.0 now
biz_content	String			Service parameter set. For detail, see below table.

Service request parameters

Parameter	Type	Required	Example	Description
appid	String(32)	Yes	kp12312b84664d44679693855d82a291	Application ID allocated to a merchant by KBZPay.

merch_code	String	Yes	200001	Specifies an Official Account Code assigned by KBZPay.
merch_order_id	String	Yes	201907152017001	The order number generated by the merchant side. It must be in the form of letters, numbers, and underscores. Other special characters are not allowed.
trade_type	String	Yes	PAY_BY_QRCODE	Trade type. The option are PAY_BY_QRCODE
title	String	No	GameRecharge	Offering name.
total_amount	String	Yes	20000	Total order amount, in Ks(MMK). The value can contain two decimal places at most.
trans_currency	String	Yes	MMK	Three-letter code complying with international standards, for example, MMK (Kyat).
timeout_express	String	No	100m	Latest payment time allowed for an order. The transaction will be closed after the deadline. The value ranges from 1 minute to 120 minutes. The value of this parameter cannot contain dots. For example, the value 1.5 hours must be converted to 90 minutes. If this parameter is not set, 120 minutes is used by default.
callback_info	String(512)	No	title%3diphonex	After the payment is successful, KBZPay will notify the merchant server of the callback request to

				return this field as it is, and the merchant can pass the business parameters according to its own needs. Must be in URL encoding format.
operator_id	String(32)	No	00001	Specifies a store operator ID (such as staff number as defined by the merchant)
sub_merch_code	String	No	200002	Code of a sub-merchant (mandatory in the service provider mode)
sub_appid	String(40)	No	Kp12312b84664d44679693855d82a564	AppId of a sub-merchant (mandatory in service provider mode)
store_id	String(32)	No	00001	Specifies a Store ID (such as store number as defined by the merchant)
terminal_id	String(32)	No	00001	Specifies a Terminal device ID
business_param	String(512)	No	type%3da	Allow merchants to pass in additional business information, the specific value should be agreed with KBZPay, and applied to security, marketing and other parameters direct transmission scenarios. Must be URL encoded format

Request example


```
{
 "Request": {
  "timestamp": "1535166225",
  "notify_url": "https://xxxxxx",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "method": "kbz.payment.precreate",
```

```
  "sign_type": "SHA256",
  "sign": "35E5D95F89DE5525E3DAC50B41C8664B816A7FE4D3AA908C3CEDEB87FE6B2689",
  "version": "1.0",
  "biz_content": {
   "appid": "kp65ad48c26a4c4b84b486dab3835112",
   "merch_code": "09991234567",
   "merch_order_id": "0101234123456789012",
   "trade_type": "PAY_BY_QRCODE",
   "title": "iPhone X",
   "total_amount": "5000000",
   "trans_currency": "MMK",
   "timeout_express": "100m",
   "callback_info": "title%3diphonex"
  }
 }
}
```


Response Parameters

Common response parameters

Parameter	Type	Required	Example	Description
result	String	Yes	SUCCESS	SUCCESS or FAIL. When this field is FAIL, the code field is a specific business error code.
code	String	Yes	0	Return code. 0 is successful, the rest is the business error code
msg	String	Yes	Success	Return information, simple error description.
nonce_str	String (32)	Yes	274E40E9388047778768B67068B9C8AF	Random character string. 32 characters or fewer.
sign	String	Yes	F315809E95194034C997C9B15C907C7AC36D59142AF207D93D17883CF8D034F0	Specifies a signature. For more information, see Section 1.3.1Signature Algorithm
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.

Response parameters (When return_code is SUCCESS, return data will also include the following fields)

Parameter	Type	Required	Example	Description
prepay_id	String (64)	Yes	KBZ007a6bd3175cdb3c658545a4f3f85fac23143239021	ID of the customer payment process.
merch_order_id	String (40)	Yes	201907151435001	Order ID on the merchant side.
qrCode	String	No	000201010212020211105000346KBZ007506e47a617bef22e48635f996ea8ba7144157120294600062000010732kp65ad48c26a4c4b84b486dab383511250200006KBZPay0106KBZPay5303MMK5802MM62170813PAY_BY_QRCODE64060002my630444BA	This parameter is available only in the PAY_BY_QRCODE scenario. QR code character string, which is used in dynamic payment scenarios. The merchant needs to display the received character string as a QR code for the customer to scan the QR code for payment.

Success response example


```
{
 "Response": {
  "result": "SUCCESS",
  "code": "0",
  "msg": "success",
  "prepay_id": "KBZ00FDKSBAKLBGFD4364564GF6",
  "merch_order_id": "0101234123456789012",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "qrCode":
"000201010212020211105000346KBZ007506e47a617bef22e48635f996ea8ba7144157120294600062000010732kp65ad48c26a4c4b84b486dab383511250200006KBZPay0106KBZPay5303MMK5802MM62170813PAY_BY_QRCODE64060002my630444BA",
  "sign_type": "SHA256",
  "sign": "B283D131E94FE097C1ACD280A7011571A4F1694CA9C070F869720AF6A676B221"
 }
}
```

Failure response example


```
{
 "Response": {
  "result": "FAIL",
  "code": "ATHENTICATION_FAIL",
  "msg": "merchant authentication fail."
 }
}
```


Order Query Interface

Use Case

This API allows inquiry of all payment orders made from[Unified Order Placing Interface]. After receiving a status code using this API, merchants can proceed with the next step in service logic. The following are situations when to use the Query Order API:

1. The Merchant doesn't receive any payment due to an exception in the Merchant's backend, network or server;
2. In MICROPAY scenario, a system error or unknown transaction status is returned after calling the [Micropay Interface];
3. To confirm payment status before calling the Close Order API or Revoke Order API.
4. A system error, refund status 'Refunding' or timeout is returned after calling[Refund Interface].

Certificate Requirement

No certificate is required.

URL

- URL(UAT): `http://api.kbzpay.com/payment/gateway/uat/queryorder`
- URL(Prod): `https://api.kbzpay.com/payment/gateway/queryorder`

METHOD NAME: `kbz.payment.queryorder`

Request Parameters

Common request parameters

Parameter	Type	Required	Example	Description

timestamp	String (13)	Yes	1563161657	Time when a request is sent. UTC timestamp. The unit is second.
method	String	Yes	kbz.payment.queryorder	Set to ‘kbz.payment.queryorder, fixed for this interface
nonce_str	String (32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Random character string containing a maximum of 32 characters, including uppercase letters, lowercase letters, digits, but not special characters.
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.
sign	String	Yes	F957B724D1E3704E90BA801BC6CB6DF4FA0898EBF94C67710CB71ABF40BFF8A1	Request signature. For more information, see Section 1.3.1 Signature Algorithm
version	String (64)	Yes	3.0	Interface version number. The latest version is 3.0
biz_content	String			Service parameter set. For detail, see below table.

Request parameters

Parameter	Type	Required	Example	Description
appid	String (32)	Yes	kp12312b84664d44679693855d82a291	Specifies an Official Account ID assigned by KBZPay.
merch_code	String	Yes	200001	Specifies an Official Account Code assigned by KBZPay.
merch_order_id	String (40)	Yes	201907152017001	The order number generated by the merchant side. It must be in the form of letters, numbers, and underscores. Other special characters are not allowed.
refund_request_no	String	No	201907152017001	Unique refund request ID. If the merchant has used the

				refund interface, can check the associated refund status in this interface.

Request example


```
{
 "Request": {
  "timestamp": "1535166225",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "method": "kbz.payment.queryorder",
  "sign_type": "SHA256",
  "sign": "BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE",
  "version": "3.0",
  "biz_content": {
   "appid": "kp65ad48c26a4c4b84b486dab3835112",
   "merch_code": "200001",
   "merch_order_id": "201907151443001"
  }
 }
}
```


Response Parameters

Common response parameters

Parameter	Type	Required	Example	Description
result	String	Yes	SUCCESS	SUCCESS or FAIL. When this field is FAIL, the code field is a specific business error code.
code	String	Yes	0	Return code. 0 is successful, the rest is the business error code
msg	String	Yes	Success	Return information, simple error description.
nonce_str	String (32)	Yes	274E40E9388047778768B67068B9C8AF	Random character string. 32 characters or fewer.
sign	String	Yes	F315809E95194034C997C9B15C907C7AC36D59142AF207D93D17883CF8D034F0	Specifies a signature. For more information, see Section 1.3.1Signature Algorithm

sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.

Response parameters (When return_code is SUCCESS, return data will also include the following fields)

Parameter	Type	Required	Example	Description
merch_order_id	String(40)	Yes	20190715144300 1	Order ID on the merchant side.
total_amount	String	Yes	2000	Total order amount.
trans_currency	String	Yes	MMK	Currency type used in order placing
trade_status	String	Yes	WAIT_PAY	Transaction status. The options are PAY_SUCCESS, PAY_FAILED, WAIT_PAY, PAYING, ORDER_EXPIRED and ORDER_CLOSED
mm_order_id	String	No	1535166225	KBZPay side transaction order id. (Only have a value when the trade status is PAY_SUCCESS)
pay_success_time	String	No	01012341234567 89012	Payment success time. (Only have a value when the trade status is PAY_SUCCESS)
refund_info	JSONArray	No		If the original order has a associated refund request, there will be a list here. If you initiate an excessive refund request, all information will be listed here, in order of creation time. This parameter does not participate in the signature

Description of "trade_status"

trad_status	Description
PAY_SUCCESS	The order has been paid successfully
PAY_FAILED	Order payment failure
WAIT_PAY	Wait for payment
PAYING	During the payment
ORDER_EXPIRED	Order Expired
ORDER_CLOSED	Order Closed

Refund info array (When original order has associated refund receipt, return data will also include the following fields)

Parameter	Type	Required	Example	Description
refund_amount	String	Yes	20000	Refund amount.
refund_currency	String	Yes	MMK	Refund currency.
refund_status	String	Yes	REFUND_SUCCESS	Refund Status, REFUND_SUCCESS, REFUNDING, REFUND_FAILED and REFUND_DUPLICATED
refund_reason	String	No	Recharge failed	Refund reason text.
create_time	String	Yes	1563267587	The timestamp is the UTC format. The unit is second.
refund_time	String	No	1563267587	Refund success timestamp. This parameter is valid only when the refund is successful.
refund_request_no	String	Yes	201907161706001	Refund unique request number generated by merchant when calling Refund Interface
trans_order_id	String	No	01012341234567896 66	Specifies the KBZPay refund order id number
error_code	String	No	FIND_REQUEST_NO_ FAIL	This parameter is valid only when the refund information fails to be queried.
error_msg	String	No	can not find refund receipt by this refund_request_no.	This parameter is valid only when the refund information fails to be queried.

Success response example (paid)


```
{
 "Response": {
  "result": "SUCCESS",
  "code": "0",
  "msg": "sucess",
  "merch_order_id": "01012341234567890 12",
  "total_amount": "500000",
  "trans_currency": "MMK",
```

```
   "trade_status": " PAY_SUCCESS",
   "pay_success_time": "1535166225",
   "sign_type": "SHA256",
   "sign": "BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE "
  }
}
```


Success response example (unpaid)


```
{
  "Response": {
   "result": "SUCCESS",
   "code": "0",
   "msg": "sucess",
   "merch_order_id": "0101234123456789012",
   "trade_status": "WAIT_PAY",
   "sign_type": "SHA256",
   "sign": "BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE "
  }
}
```


Success response example (refund)


```
{
  "Response": {
   "result": "SUCCESS",
   "code": "0",
   "msg": "sucess",
   "merch_order_id": "0101234123456789012",
   "total_amount": "500000",
   "trans_currency": "MMK",
   "trade_status": " PAY_SUCCESS",
   "pay_success_time": "1535166225",
   "refund_info": [
    {
     "refund_amount": "500000",
     "refund_currency": "MMK",
     "refund_status": "REFUND_DUPLICATED",
     "refund_reason": "REFUND",
     "create_time": "1535166225",
     "refund_request_no": "123456789999"
    },
    {
     "refund_amount": "500000",
     "refund_currency": "MMK",
```

```
      "refund_status": "REFUND_SUCCESS",
      "refund_reason": "REFUND",
      "create_time": "1535166225",
      "refund_time": "1535166225",
      "refund_request_no": "123456789999"
     }
   ],
   "sign_type": "SHA256",
   "sign": " BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE "
 }
}
```


Success response,refund fail example


```
{
 "Response": {
   "merch_order_id": "201810111530",
   "total_amount": "3.00",
   "trans_currency": "MMK",
   "trade_status": "PAY_SUCCESS",
   "pay_success_time": 1539243010,
   "refund_info": [
    {
     "error_code": "FIND_REQUEST_NO_FAIL",
     "error_msg": "can not find refund receipt by this refund_request_no."
    }
   ],
   "result": "SUCCESS",
   "code": "0",
   "msg": "success",
   "nonce_str": "1D6DE7BF008049FA89F45374147E8D56",
   "sign_type": "SHA256",
   "sign": "14A8222CEBA10B10D6B2EC64A898644421FB8F70EB2CE31D2B6607BB8C8A1799"
 }
}
```


Failure response example


```
{
 "Response": {
   "result": "FAIL",
   "code": "AUTHENTICATION_FAIL",
   "msg": "merchant authentication fail."
 }
```

}

Failure response example(auth_code is expired /Customer close the pin pad)


```
{
 "Response": {
  "result": "FAIL",
  "code": "AOP14505",
  "msg": "Could not find the order.",
  "nonce_str": "3C6LIUTG1KNSTL019QVJUR6PUTL5241P",
  "sign_type": "SHA256",
  "sign": "D3DCC29A4BF0ADB35641F0AAF1C6F3B5BF95F61000DDB3E2E7D28AD148CCA4F5"
 }
}
```


Error Code Description

error_code	error_msg
FIND_REQUEST_NO_FAIL	can not find refund receipt by this refund_request_no.
QUERY_REFUND_RECEIPT_FAIL	query refund receipt fail.
AOP18034	The statement does not exist, please check whether the order is correct.
REQUEST_FAIL	The Request is invalid

Querying Refunds Interface

Use Case

This interface is invoked to query the refund status after a refund application is submitted.

Certificate Requirement

No certificate is required.

URL

- URL(TEST) : http://api.kbzpay.com/payment/gateway/uat/queryrefund
- URL(Prod) : https://api.kbzpay.com/payment/gateway/queryrefund

METHOD NAME : kbz.payment.queryrefund

Request Parameters

Common request parameters

Parameter	Type	Required	Example	Description
timestamp	String (13)	Yes	1563161657	Time when a request is sent. UTC timestamp. The unit is second.
method	String	Yes	kbz.payment.queryrefund	Set to 'kbz.payment.queryrefund, fixed for this interface
nonce_str	String (32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Random character string containing a maximum of 32 characters, including uppercase letters, lowercase letters, digits, but not special characters.
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.
sign	String (64)	Yes	F957B724D1E3704E90BA801BC6CB6DF4FA0898EBF94C67710CB71ABF40BFF8A1	Request signature. For more information, see Section 1.3.1 Signature Algorithm
version	String	Yes	1.0	Interface version number. The latest version is 1.0
biz_content	String	Yes		Service parameter set. For detail, see below table.

Request parameters

Parameter	Type	Required	Example	Description
appid	String (32)	Yes	Kp12312b84664d44679693855d82a291	Specifies an Official Account ID assigned by KBZPay.
merch_code	String	Yes	200001	Specifies an Official Account Code assigned by KBZPay.
merch_order_id	String (40)	Yes	201907152017001	The order number generated by the merchant

				side. It must be in the form of letters, numbers, and underscores. Other special characters are not allowed.
sub_type	String	No	32	Sub-merchant Type
sub_identifier_type	String	No	32	Sub-merchant Identifier Type
sub_identifier	String	No	32	Sub-Merchant Identifier
refund_request_no	String	No	201907152017001	Unique refund request ID. If the merchant has used the refund interface, can check the associated refund status in this interface.

Request example


```
{
 "Request": {
  "timestamp": "1535166225",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "method": "kbz.payment.queryrefund",
  "sign_type": "SHA256",
  "sign": "90293E22BA22A2E4FE03860DC0FD362C0CF15CC7AFC0334A80DB82FDFCAC8CA9",
  "version": "1.0",
  "biz_content": {
   "appid": "kp65ad48c26a4c4b84b486dab3835112",
   "merch_code": "100973500",
   "merch_order_id": "111111111111111008"
  }
 }
}
```


Response Parameters

Common response parameters

Parameter	Type	Required	Example	Description
result	String	Yes	SUCCESS	SUCCESS or FAIL. When this field is FAIL, the code field is a specific business error code.

code	String	Yes	0	Return code. 0 is successful, the rest is the business error code
msg	String	Yes	Success	Return information, simple error description.
nonce_str	String (32)	Yes	274E40E9388047778768B67068B9C8AF	Random character string. 32 characters or fewer.
sign	String	Yes	F315809E95194034C997C9B15C907C7AC36D59142AF207D93D17883CF8D034F0	Specifies a signature. For more information, see Section 1.3.1Signature Algorithm
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.

Response parameters (When return_code is SUCCESS, return data will also include the following fields)

Parameter	Type	Required	Example	Description
merch_code	String	Yes	32	Specifies an Official Account Code assigned by KBZPay.
sub_type	String	No	32	Sub-merchant Type
sub_identifier_type	String	No	32	Sub-merchant Identifier Type
sub_identifier	String	No	32	Sub-Merchant Identifier
merch_order_id	String	Yes	32	Merchant order ID.
trans_order_id	String	Yes	32	Original transaction order ID on the KBZPay side.
refund_finished	String	Yes	32	Indicates whether the refund is completed. The options are Y (yes) and N (no). The default value is N.
total_refund_amount	String	No	32	Amount Refunded (returned when refund is not completed)
remain_refund_amount	String	No	100	Remaining refundable amount (returned only in partial refund response)
remain_refund_times	String	No	2	Remaining number of refunds (returned only in partial refund responses)
refund_info	Jsonobject	No		Refund Information

- refund_order_id	String	No	32	Refund transaction ID on the KBZPay.
- refund_amount	String	No	32	Merchant Refund Amount
- refund_currency	String	No	32	Merchant Refund Currency
- refund_time	String	No	32	Refund Time
- refund_status	String	No	32	Refund status: REFUNDING, REFUND_FAILED, REFUND_SUCCESS
- amount_detail	Jsonobject	No	32	Refund Amount Details
* payer_refund	String	No	32	Amount refunded to a subscriber, excluding all coupon amounts.
* discount_refund	String	No	32	Coupon refund amount. The default value is 0.
* currency	String	No	32	Refund Currency

Success response example


```
{
  "Response": {
    "msg": "success",
    "nonce_str": "L762DZZWX17BU9SXPK9GIWZM5OFQGWAE",
    "trans_order_id": "31074T15091748000151",
    "code": "0",
    "sign": "209A8D64002853CEADA083C05514A672183C4DA283127439195766BCBDB03277",
    "refund_info": [
      {
        "refund_order_id": "31075C16485247000104",
        "refund_request_no": "PassportAPIOffus",
        "refund_amount": "350000",
        "refund_currency": "MMK",
        "refund_time": "1747045134",
        "refund_status": "REFUND_SUCCESS",
        "amount_detail": {
          "payer_refund": "350000",
          "discount_refund": "0",
          "currency": "MMK"
        }
```

```
        }
      ],
      "reference_number": "MMQR1074T14112935000001",
      "result": "SUCCESS",
      "refund_finished": "Y",
      "merch_order_id": "PassportTest6",
      "Wallet_identifier": "uab",
      "merch_code": "70123401002",
      "mmqr_ref": "E2E22376720250429150913875004180 1",
      "sign_type": "SHA256",
      "invoice_number": "PassportTest6"
    }
}
```


Failure response example


```
{
  "Response": {
    "result": "FAIL",
    "code": "ATHENTICATION_FAIL",
    "msg": "Merchant authentication fail."
  }
}
```


Error Code Description

code	msg	场景
REQUEST_FAIL	The Request is invalid	请求格式或参数内容不合法
ATHENTICATION_FAIL	Merchant authentication fail.	商户鉴权失败
QUERYORDER_FAIL	The order does not exist.	查询的订单不存在
FLOW_CONTROL	Your requests is too frequent,please try again later.	请求过于频繁，请稍后再试。
SYSTEM_ERROR	System is unable to process your request now, please try later.	系统错误

QUERY_CUSTINFO_FAIL	Fail to query customer info in H5 page.	H5 页面内无法获取用户信息

Close Order Interface

Use Case

This API needs to be called before a merchant wants to create a new order due to the Payer failing to pay for an order. The original order will be closed in order to avoid repeat payment. After an order is created in the KBZPay payment system, if the Payer doesn't pay within the required time, no further operation can be performed in the system. To prevent the Payer from proceeding with their order, this API is called to close the order.

Certificate Requirement

No certificate is required.

URL

- URL(UAT): https://api.kbzpay.com/payment/gateway/uat/closeorder
- URL(Prod): https://api.kbzpay.com/payment/gateway/closeorder

METHOD NAME: `kbz.payment.closeorder`

Request Parameters

Common request parameters

Parameter	Type	Required	Example	Description
timestamp	String (13)	Yes	1563161657	Time when a request is sent. UTC timestamp. The unit is second.
method	String	Yes	kbz.payment.closeorder	Set to 'kbz.payment.closeorder, fixed for this interface
nonce_str	String (32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Random character string containing a maximum of 32 characters, including uppercase letters, lowercase letters, digits, but not special characters.

sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.
sign	String	Yes	F957B724D1E3704E90BA801BC6CB6DF4FA0898EBF94C67710CB71ABF40BFF8A1	Request signature. For more information, see Section 1.3.1 Signature Algorithm
version	String (64)	Yes	3.0	Interface version number. The latest version is 3.0
biz_content	String			Service parameter set. For detail, see below table.

Service request parameters

Parameter	Type	Required	Example	Description
appid	String (40)	Yes	kp12312b84664d44679693855d82a291	Application ID allocated to a merchant by KBZPay.
merch_code	String	Yes	200001	Specifies an Official Account Code assigned by KBZPay.
merch_order_id	String	Yes	201907152017001	The order number generated by the merchant side. It must be in the form of letters, numbers, and underscores. Other special characters are not allowed.

Request example


```
{
 "Request": {
  "timestamp": "1535166225",
  "method": "kbz.payment.closeorder",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "sign_type": "SHA256",
  "sign": "768E0C18F7FF0450B6A652000068980335E5DD1067FD276994116E6799EE9FCC",
  "version": "3.0",
  "biz_content": {
   "merch_order_id": "0101234123456789012",
   "merch_code": "200001",
   "appid": "kp1234567890987654321aabbccddeef "
  }
 }
```

}

Response Parameters

Common response parameters

Parameter	Type	Required	Example	Description
result	String	Yes	SUCCESS	SUCCESS or FAIL. When this field is FAIL, the code field is a specific business error code.
code	String	Yes	0	Return code. 0 is successful, the rest is the business error code
msg	String	Yes	Success	Return information, simple error description.
nonce_str	String (32)	Yes	274E40E9388047778768B67068B9C8AF	Random character string. 32 characters or fewer.
sign	String	Yes	F315809E95194034C997C9B15C907C7AC36D59142AF207D93D17883CF8D034F0	Specifies a signature. For more information, see Section 1.3.1Signature Algorithm
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.

Response parameters (When return_code is SUCCESS, return data will also include the following fields)

Parameter	Type	Required	Example	Description
merch_order_id	String(40)	Yes	201907151435001	Order ID on the merchant side.

Success response example


```
{
 "Response": {
  "result": "SUCCESS",
  "code": "0",
  "msg": "sucess",
  "merch_order_id": "0101234123456789012",
  "nonce_str": "5K8264ILTKCH16CQ2502SI8ZNMTM67VS",
  "sign_type": "SHA256",
  "sign": "768E0C18F7FF0450B6A652000068980335E5DD1067FD276994116E6799EE9FCC"
 }
```

```
}
```

Failure response example

```
{
 "Response": {
  "result": "FAIL",
  "code": "AUTHENTICATION_FAIL",
  "msg": "merchant authentication fail."
 }
}
```

Error Code Description

code	msg	场景
REQUEST_FAIL	The Request is invalid	请求格式或参数内容不合法
ATHENTICATION_FAIL	Merchant authentication fail.	商户鉴权失败
QUERYORDER_FAIL	The order does not exist.	查询的订单不存在
FLOW_CONTROL	Your requests is too frequent,please try again later.	请求过于频繁，请稍后再试。
SYSTEM_ERROR	System is unable to process your request now, please try later.	系统错误
ORDER_ALREADY_CLOSED	Order already closed.	订单已关闭
AOP03028	Close order failed.	关单失败

Refund Interface

Use Case

If a refund is required due to the buyer or seller's reasons after a transaction occurs, the seller can return the payment to the buyer through the refund interface. After receiving the refund request and verifying it, KBZPay refunds the payment to the buyer's account

according to the refund rules.A maximum of three refunds can be applied for each payment order.

Certificate Requirement

Certificate is required.

URL

- URL(TEST): `https://api.kbzpay.com:18008/payment/gateway/uat/refund`
- URL(Prod): `https://api.kbzpay.com:8008/payment/gateway/refund`

METHOD NAME: `kbz.payment.refund`

Request Parameters

Common request parameters

Parameter	Type	Required	Example	Description
timestamp	String (13)	Yes	1563161985	Time when a request is sent. UTC timestamp. The unit is second.
method	String	Yes	kbz.payment.refund	Set to 'kbz.payment.refund, fixed for this interface
nonce_str	String (32)	Yes	5K8264ILTKCH16CQ2502SI8ZNMTM67VS	Random character string containing a maximum of 32 characters, including uppercase letters, lowercase letters, digits, but not special characters.
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.
sign	String	Yes	F957B724D1E3704E90BA801BC6CB6DF4FA0898EBF94C67710CB71ABF40BFF8A1	Request signature. For more information, see Section 1.3.1 Signature Algorithm
version	String (64)	Yes	1.0	Interface version number. The latest version is 1.0
biz_content	String			Service parameter set. For detail, see below table.

Request parameters

Parameter	Type	Required	Example	Description
appid	String(32)	Yes	kp12312b84664d44679693855d82a291	Specifies an Official Account ID assigned by KBZPay.
merch_code	String	Yes	200001	Specifies an Official Account Code assigned by KBZPay.
merch_order_id	String(40)	Yes	201907152017001	The original order number generated by the merchant side.
sub_type	String	Yes	50000	Sub-merchant Type
sub_identifier_type	String	Yes	04	Sub-merchant Identifier Type
sub_identifier	String	Yes	20006	Sub-Merchant Identifier
Refund_reason	String(256)	No	Recharge failed.	Refund reason.
refund_request_no	String(32)	Yes	2019071523017001	Unique refund request ID generated by the merchant side.
refund_amount	String(32)	No	100	Merchant Refund Amount(This field is left blank when all refunds are performed.)
is_last_refund	String(2)	No	Y	If part of the refund amount has been refunded, if this field is set to Y, the remaining amount is refunded.

Request example


```
{
 "Request": {
  "timestamp": "1536637503",
  "nonce_str": "8452559103085644 81",
  "method": "kbz.payment.refund",
  "sign_type": "SHA256",
```

```
  "sign": "90293E22BA22A2E4FE03860DC0FD362C0CF15CC7AFC0334A80DB82FDFCAC8CA9",
  "version": "1.0",
  "biz_content": {
   "appid": "kp12345678998765432 1abcdefghijkl",
   "merch_order_id": "111111111111111008",
   "merch_code": "100973500",
   "sub_type": "50000",
   "sub_identifier_type": "04",
   "sub_identifier": "200006",
   "refund_request_no": "201809111536632503",
   "refund_reason": "testtestetstestes",
   "refund_amount":"100"
  }
 }
}
```


Response Parameters

Common response parameters

Parameter	Type	Required	Example	Description
result	String	Yes	SUCCESS	SUCCESS or FAIL. When this field is FAIL, the code field is a specific business error code.
code	String	Yes	0	Return code. 0 is successful, the rest is the business error code
msg	String	Yes	Success	Return information, simple error description.
nonce_str	String (32)	Yes	274E40E9388047778768B67068B9C8AF	Random character string. 32 characters or fewer.
sign	String	Yes	F315809E95194034C997C9B15C907C7AC36D59142AF207D93D17883CF8D034F0	Specifies a signature. For more information, see Section 1.3.1Signature Algorithm
sign_type	String	Yes	SHA256	Signature type. Currently, only SHA256 is supported.

Response parameters (When return_code is SUCCESS, return data will also include the following fields)

Parameter	Type	Required	Example	Description

merch_code	String	Yes	20001	Short code registered by a merchant with the Mobile Money.
merch_order_id	String (40)	Yes	201810151651001	Merchant order ID.
trans_order_id	String	Yes	01001383050000036415	Original transaction order ID on the KBZPay side.
refund_order_id	String	Yes	01001383050000036420	Refund transaction order ID on the KBZPay side.
refund_amount	String	Yes	20000	Refund amount.
refund_currency	String	Yes	MMK	Refund currency.
refund_status	String	Yes	REFUND_SUCCESS	Refund Status: REFUND_SUCCESS, REFUNDING, REFUND_FAILED
refund_time	String	No	1563267587	Refund success timestamp. This parameter is valid only when the refund is successful.
remain_refund_amount	String	No	100	Remaining refundable amount (returned only in partial refund response).

Refund Successful


```
{
 "Response": {
  "merch_code": "100973500",
  "merch_order_id": "201810151651001",
  "trans_order_id": "01001383050000036415",
  "refund_order_id": "01001383000000036420",
  "refund_amount": "100",
  "refund_currency": "MMK",
  "refund_status": "REFUND_SUCCESS",
  "refund_time": "1539593508",
  "result": "SUCCESS",
  "code": "0",
  "msg": "success",
  "nonce_str": "14D56DB4926B403F8F8875DEE882FEB7",
  "sign_type": "SHA256",
  "sign": "BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE"
 }
}
```

Refund request accept, but wait for retrying


```
{
 "Response": {
  "merch_code": "100973500",
  "merch_order_id": "201810151651001",
  "trans_order_id": "01001383050000036415",
  "refund_status": "REFUNDING",
  "result": "SUCCESS",
  "code": "0",
  "msg": "success",
  "nonce_str": "14D56DB4926B403F8F8875DEE882FEB7",
  "sign_type": "SHA256",
  "sign": "BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE"
 }
}
```


Refund Information Query Error Code Definition

Error Code	Message	Scenario
FIND_REQUEST_NO_FAIL	The request contains a unique refund request number, but no refund record corresponding to the refund request is found.	The request contains a unique refund request number, but no refund record corresponding to the refund request is found.
QUERY_REFUND_RECEIPT_FAIL	Failed to query the refund receipt.	Failed to query the refund receipt. This may be caused by an internal system error.
REQUEST_FAIL	The Request is invalid	The Request is invalid
ATHENTICATION_FAIL	Merchant authentication fail.	Merchant authentication fail.
FLOW_CONTROL	Your requests is too frequent,please try again later.	Your requests is too frequent,please try again later.
SYSTEM_ERROR	System is unable to process your request now, please try later.	System is unable to process your request now, please try later.
QUERY_REFUND_SUMMARY_FAILED	Query refund info failed.	Query refund info failed.

AOP07012	Partial amount invalid.	The partial refund amount cannot be greater than the original amount.
UPDATE_REFUND_APPLICATION_FAILED	Update refund application failed.	Update refund application failed.

Note: If the refund information fails to be queried, the basic queryorder process is not affected.

6 Callback Interface

Use Case

After completing a payment, the KBZPay payment system will send the relevant payment result and user information to the Merchant. When this happens, the Merchant's backend will need to receive the result and return a reply to the KBZPay payment system.

When interacting with this API, if the KBZPay payment system does not receive a response from the Merchant backend indicating success or timeout, the KBZPay payment system will consider it as an unreceived notification and initiate further payment result notifications at a regular interval, so as to ensure successful receipt. However, the KBZPay payment system cannot ensure successful receipt of payment notifications in every case. (Notification frequency: 2 times 60s、600s)

As payment result notifications may be sent from the KBZPay payment system to the Merchant's backend multiple times, a single payment result might be notified to the Merchant's backend multiple times. For this reason, the Vendor's system must be able to handle duplicate notifications properly.

In the case that the order status is unknown or the payment result notification is not received, **it is recommended that the merchant actively call the Query Order Interface to confirm the order status.**

The callback request will be sent to the url address provided by the merchant using POST method. Please ensure that the provided url supports the POST method.

Note: To avoid receiving fake notifications, merchants must verify the signature of the notification and check whether the order amount in the notification is the same with the one in merchant's system.

Certificate Requirement

No certificate is required.

URL

This URL may be configured via `notify_url`, a parameter submitted via the Unified Order Interface. The notify URL must be a URL that can be directly accessed and cannot contain parameters. **Ensure that the IP address is accessible from the public network.** If merchants can't open this URL, they will not be able to receive any notifications sent from the KBZPay payment system.

Callback Notification Parameters

Parameter	Type	Required	Example	Description
appid	String (40)	Yes	kp12312b84664d44679693855d82a291	Application ID allocated to a merchant by KBZPay.
notify_time	String	Yes	1563267587	Notification sending time. The timestamp is of the long type. The time is the UTC format. The unit is second.
merch_code	String	Yes	200001	Short code registered by a merchant with the Mobile Money.
merch_order_id	String	Yes	201907161732001	Order ID on the merchant side.
mm_order_id	String	Yes	0101234123456745641	Order ID on the KBZPay side.
total_amount	String	Yes	2000	Order payment amount in a transaction.
trans_currency	String	Yes	MMK	Transaction currency type.
trade_status	String	Yes	PAY_SUCCESS	Transaction status.
trans_end_time	String	Yes	1563267587	Time when a transaction ends. The unit is second.
callback_info	String	No		Callback information. If this parameter is transferred when a merchant places an order,

				the parameter is returned to the merchant during asynchronous notification.
nonce_str	String	Yes	FD546GFD46GFDA768R7DER	Random character string.
sign	String	Yes	BC4EE8D710BAC6A7E33DE4511A1CE7723024615EEF491B80DEF7DC743D4DADBE	Response signature.
sign_type	String	Yes	SHA256	Signature type of the response.

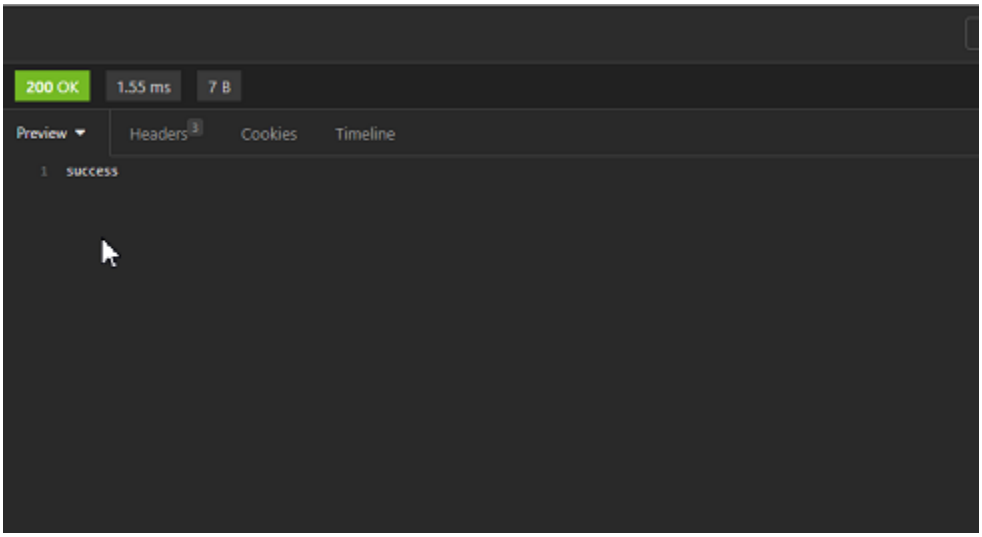
Callback request example


```
{
  "Request":{
    "notify_time":1576842150,
    "merch_code":"200001",
    "merch_order_id":"0101234123456789012",
    "mm_order_id":"01001814070006560257",
    "trans_currency":"MMK",
    "total_amount":"300",
    "trade_status":"PAY_SUCCESS",
    "trans_end_time":1576834704,
    "callback_info":"urlendcode",
    "nonce_str":"513ba55344ad44c8b69465aae66f7703",
    "sign_type":"SHA256",
    "appid":"kp1234567890987654321aabbccddeef",
    "sign":"A570F3625DE55BD1893491677BD85D904044E4742296F66E62D08B3BFB6C3B29"
  }
}
```


Callback Response

The merchant receives the successful callback information. After the program is executed, the program must response the "success" message (Do not include quotation marks and are not case sensitive.). If the message returned by the merchant is not "success", the server attempts to resend the notification.

Callback response example



rror Code Description

Error Codes

Error Code	Message	Scenario
REQUEST_FAIL	The request format or parameter content is invalid.	The request format or parameter content is invalid.
ATHENTICATION_FAIL	Merchant authentication fails.	Merchant authentication fails.
PRECREATE_FAIL	An order fails to be placed.	An order fails to be placed.
ORDER_ALREADY_PAID	The order has been paid and fails to be placed again.	The order has been paid and fails to be placed again.
ORDER_ID_USED	The order fails to be placed because the order ID has been used.	The order fails to be placed because the order ID has been used.
QUERYORDER_FAIL	The queried order does not exist.	The queried order does not exist.
FLOW_CONTROL	You tried to submit it too may times. Try again later.	You tried to submit it too may times. Try again later.
SYSTEM_ERROR	System error. Try again later.	System error. Try again later.

CUSTOMER_CLOSED	The original customer has been closed and cannot be refunded.	The original customer has been closed and cannot be refunded.
EXCEED_REFUND_LIMIT	The number of refund requests exceeded the upper limit.	The number of refund requests exceeded the upper limit.
REFUND_ALREADY_SUCCESS	The original transaction has a successful refund. The refund cannot be initiated repeatedly.	The original transaction has a successful refund. The refund cannot be initiated repeatedly.
BALANCE_INSUFFICIENT	The merchant account balance is insufficient and the refund cannot be initiated.	The merchant account balance is insufficient and the refund cannot be initiated.
QUERY_CUSTINFO_FAIL	Failed to obtain subscriber information on the H5 page.	Failed to obtain subscriber information on the H5 page.