

GCCPAY Payment API Document

1. Introduction

System overview: The GCCPAY payment system is used to complete the payment collection of merchants. The main process of docking is that the merchant creates a payment order through the interface, and guides the payer (buyer) to the GCCPAY payment page. After completing the payment, GCCPAY will guide the payer's browser to redirect to the merchant page, GCCPAY will also notify the merchant through the server.

2. Update History

| Time | Version | Originator | Main changes |
|----------|---------|----------------|--------------------------------------|
| 20221023 | V1.0.0 | Xiongfei.Z hao | First edition |
| 20221110 | V1.0.1 | Xiongfei.Z hao | Added [Notification Callback] module |
| 20221116 | V1.1.0 | Shaopeng. Li | Add iframe docking method |

Release time & Release content

- Version V1.0.0 was released on October 23, 2022
- Version V1.0.1 was released on November 10, 2022
- Version V1.1.0 was released on November 16, 2022

Post content

V1.0.0

- The first draft of the payment system Open API

V1.0.1

- Added [Notification Callback] module

V1.1.0

- Added iframe docking method
 - Added code demo
 - Added [Merchant Information Query] method
-

3. Call Method

3.1 Term Explanation

| Term | Illustration |
|---------------------------------|--|
| Merchant | The merchant is the seller in the payment scenario, and the payment system is helping the seller to realize the collection business. |
| Payer | The buyer pays the seller through the payment page in the payment scene. |
| Payment order | It is created by the seller calling the create order API of the payment gateway by signature, and is used for collection. |
| Transaction | When the payer pays the seller through the payment page, a certain payment behavior is a transaction. One order may generate multiple payment transactions, but only one will succeed. If only one payment transaction succeeds, others will automatically be invalidated. |
| 2FA(2-step verification string) | Using Google Identity to improve account security, users need to import into the Google Identity Authentication installed on their mobile phones, and the verification data needs to be verified. |
| Client | Contains the ID, key and secret used to calculate the signature. |

3.2 Service info

Sandbox environment

Service address : https://sandbox.gcc-pay.com/api_v1/

Communication protocol : HTTP

Encoding : UTF-8

3.3 Public parameters

Public request parameters

HTTP's header :

| Header name | Type | Is required | Illustration |
|---------------------|--------|-------------|---|
| x-auth-signature | String | Yes | The string of auth signature |
| x-auth-key | String | Yes | The key of auth |
| x-auth-timestamp | String | Yes | The request's timestamp, int32, accurate to seconds |
| x-auth-sign-method | String | Yes | The signature calculation method is currently unified as HmacSHA256 |
| x-auth-sign-version | String | Yes | The signature calculation method version is currently unified as 1 |

3.4 Authentication mechanism

1. Construct the header information of the request parameters, and all the information in the signature verification is in the header information
2. Signature calculation method
 - a. Arrange a batch of key-value pairs into the format of key=value. value is processed by urlencode;
 - b. Then sort the batch of key-value pairs in ascending dictionary order, and then use the & symbolic link;
 - c. Then use the secret applied above to use sha256 to calculate the summary information of the spliced string, and the output is base64 encoding.
 - d. The key-value pairs are as follows
 - i. uri: the requested url without the root part, for example:
/users/100000/orders
 - ii. key: the applied key

- iii. `timestamp`: timestamp, int32, accurate to seconds
- iv. `signMethod`: fixed, "HmacSHA256"
- v. `signVersion`: fixed, "1"
- vi. `method`: 'merchant.addOrder', this value is different for different interfaces

3.5 Response Result

Successful response

When the HTTP response status code is `2xx`, the call is successful.

Failure response

When the HTTP response status code is `4xx` or `5xx`, the call fails.

Response Sample

```
HTML
{
  "code": "notAllowed", ### Error code
  "message": "No access", ### Error description
  "data": ### Error detail
  [
    "signature error",
    {
      "uri": "/merchants/M448726",
      "key": "zS83UNCPhVTqBxDHACJ30sImZRKA1zQI",
      "timestamp": 1672991487,
      "signMethod": "HmacSHA256",
      "signVersion": "1",
      "method": "merchant.detail"
    }
  ]
}
```

4. APIS

4.1 Create Payment Order

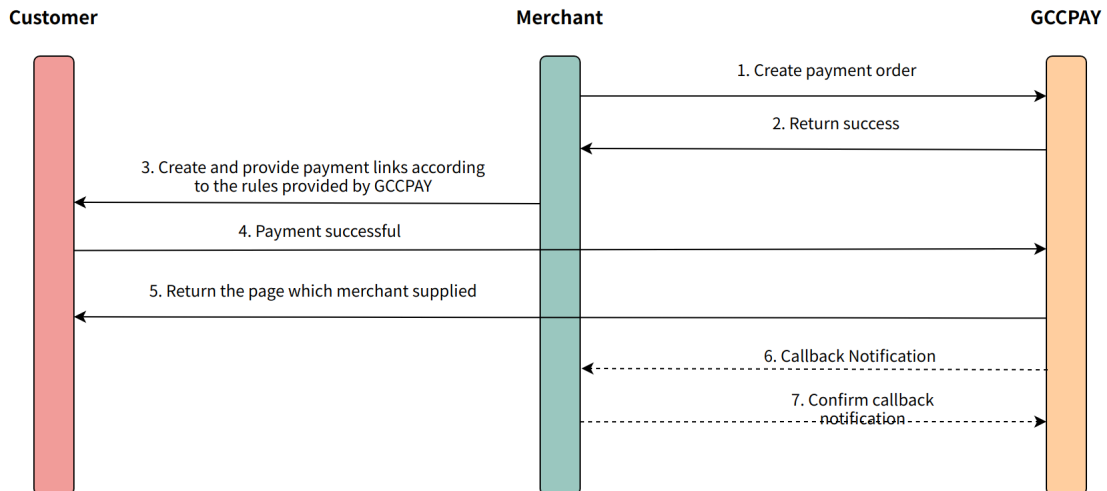
4.1.1 Using GCCPAY Cashier page to create payment

order

Summary

- After creating a payment order through the payment system, use the cashier interface provided by GCCPAY to collect money

Sequence diagram



HTTP request type : POST

URL : /merchants/%YOUR_MERCHANT_ID%/orders

- %YOUR_MERCHANT_ID% : Your merchant ID in system
 - For example: if your merchant ID is 'M00001', then the URL is:
https://sandbox.gcc-pay.com/api_v1/merchants/M00001/orders

Request

The request parameters for [Create Payment Order] are as follows:

- Header**
View public request parameters.
- Request Body**

| Parameters | Type | Required | Describe | Example |
|-----------------|--------|----------|--|------------------------|
| merchantOrderId | string | Yes | The merchant payment order ID, can't repeat. | "2022102218233400123" |
| amount | string | Yes | The order amount, floating | "1.35", "99.10", "127" |

| | | | | |
|-----------------|--------|-----|--|---|
| | | | point number of string, up to 2 digits after the decimal point. | |
| currency | string | Yes | Type of currency, and 3 capital letters. | "SAR", "AED", "BHD", "EGP", "KWD", "OMR", "QAR" |
| name | string | Yes | The order description. | "Buy a game recharge card with a face value of 100" |
| notificationURL | string | No | Address for successful order payment notification. | |
| expiredAt | string | No | The order expires. After the expiration, the order can not be paid. It is recommended to set it to 30 min later, and the default time is 15 days. | "2022-10-23T12:00:21.000Z" |

- **Request demo**

```
HTML
{
  "merchantOrderId": "TEST1672491207",
  "amount": 12.02,
  "currency": "SAR",
  "name": "Order test for desc",
  "notificationURL": "",
  "expiredAt": "2023-01-15T20:53:27.000Z"
}
```

Response

- The response parameters for [Create Payment Order] are as follows:

| Parameters | Type | Description | Sample |
|------------|--------|---|---|
| id | string | Payment order id | "M448726T2022123112531745487632" |
| clientId | string | Client id | "CLT9208307" |
| merchantId | string | Merchant id | "M448726" |
| status | string | Order status <ul style="list-style-type: none"> • pending : GCCPAY payment system has created a payment order, but the channel party has not yet created an order • doing : GCCPAY payment system and channel order have been created, but the user has not paid • paid : User has paid for the order • Closed : The order has been closed • error : The order has some error • locked : The order has been locked • expired : The order has expired | "pending" |
| ticket | string | Bill voucher, used to verify order ID | "QnTkb04gg5RzB6fPKIfKH9AJgTW7PYQKCYnP0Wwdz0uiQQZjSt8UjsHtBJbBBmTMc" |
| name | string | The description of order | "2022-10-23T12:00:21.000Z" |

| | | | |
|-----------------|--------|-------------------------------------|----------------------------|
| merchantOrderId | string | The description of merchant's order | "TEST1672491207" |
| notificationURL | string | Callback address | |
| amount | float | Payment amount | "12.02" |
| currency | string | The payment currency | "SAR" |
| expiredAt | string | The expired time of order | "2023-01-15T20:53:27.000Z" |
| createAt | string | The order created time | "2022-12-31T12:53:17.454Z" |
| refundAmount | float | The order refund amount | "0" |

Response demo

```
HTML
{
  "id": "M448726T2022123112531745487632",          ###
  OrderId
    "clientId": "CLT9208307",          ### ClientID
    "merchantId": "M448726",          ### Merchant ID
    "status": "pending",          ### Order status
    "ticket":
      "QnTkb04gg5RzB6fPK1fKH9AJgTW7PYQKCYNP0Wwdz0uiQZjSt8UjsHtBJbBBmTMc",
      ### 票据凭证，用于验证订单 ID
    "name": "Order test for desc",          ### Order description
    "merchantOrderId": "TEST1672491207",          ### Order ID
  of merchant
    "notificationURL": "",          ### Callback URL
    "amount": 12.02,          ### Payment amount
    "currency": "SAR",          ### Payment currency
    "expiredAt": "2023-01-15T20:53:27.000Z",          ### Order
  expired time
    "createdAt": "2022-12-31T12:53:17.454Z",          ### Order create
  time
    "refundAmount": 0          ### Refund amount
}
```

The rule of payment link

URL : `https://sandbox.gcc-`

`pay.com/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}&language={LANGUAGE}`

- `{ORDER_ID}` : Order ID returned after order created.
- `{ORDER_TICKET}` : Ticket returned after order created.
- `{RETURN_URL}` : The page address after payment, and RETURN_URL needs URLencode.
- `{LANGUAGE}` : The default language of cashier, optional value is EN(English) and AR(Arabic)

You can specify a specific payment method (ensure that your merchant supports this payment mode), and the URL is follows:

Card Pay :

`https://sandbox.gcc-`

`pay.com/mastercard/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}&language={LANGUAGE}`

STCPAY Pay :

`https://sandbox.gcc-`

`pay.com/stcpay/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}&language={LANGUAGE}`

APPLE PAY :

`https://sandbox.gcc-`

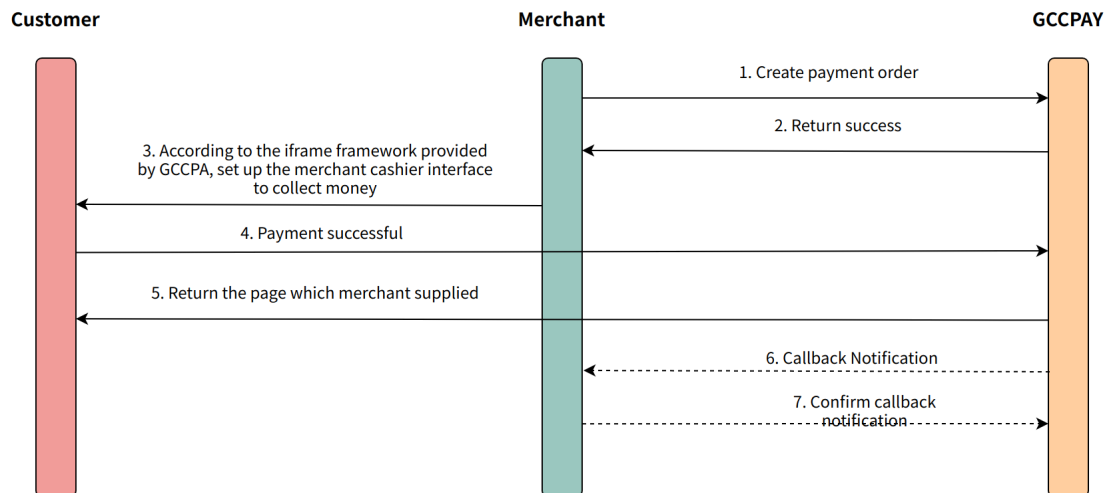
`pay.com/applePay/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}&language={LANGUAGE}`

4.1.2 Using iframe to create payment order

Summery

- After creating a payment order through the payment system, use the iframe framework provided by GCCPAY to build your own cashier interface

Sequence diagram



Create payment order

- Create a payment order in GCCPAY payment system by referring to "Create Payment Order"

Create a cash register using the iframe framework

1. After successfully creating the payment order, construct the cashier URL according to the returned response parameters

`https://sandbox.gcc-pay.com/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}&language={LANGUAGE}`

- `{ORDER_ID}` : The order ID returned by GCCPAY after successfully creating the payment order
- `{ORDER_TICKET}` : The TICKET returned by GCCPAY after successfully creating the payment order
- `{RETURN_URL}` : After the user completes the payment, return to the URL address of the interface, RETURN_URL needs URLencode
- `{LANGUAGE}` : The default language of cashier, optional value is EN(English) and AR(Arabic)

1. Create your own cashier interface. The iframe frame is embedded in the cashier interface. The src of the iframe points to the URL constructed in step 1

- Demo

Python

```

var iframe = document.createElement('iframe');
iframe.src = 'https://sandbox.gcc-pay.com/?orderId={ORDER_ID}&ticket={ORDER_TICKET}&returnURL={RETURN_URL}';
iframe.style = 'width: 1000px; height: 600px;';
  
```

```
document.body.appendChild(iframe);
```

4.2 Query Payment Order

Summary

- Query the order status in the payment system.

HTTP request type : GET

URL : `/orders/%ORDER_ID%`

- `%ORDER_ID%` : The order ID which you want to query

Response

【Query payment order】 response as follows:

- **Response Body**

| Parameters | Type | Describe | Example |
|-----------------|--------|---|--|
| id | string | The payment order ID | "M000001T2022101023455774363043" |
| clientId | string | The client ID which created the payment | "CLT0000001" |
| merchantId | string | The merchant ID to which the payment belong | "M000001" |
| status | string | Payment order status | "pending" |
| ticket | string | This is the payment order operation evidence, the URL must contain this parameter when guiding the payer to pay | "Q6XUD1tntwB0QASXzrZesapHEXBXEUMkiOs5KqLpySxwYtUfr8TkNNNmMyrFAw89" |
| name | string | Payment order describe | "Buy a game recharge card with a face value 100" |
| merchantOrderId | string | The merchant payment order ID | "2022102218233400123" |

| | | | |
|-----------------|--------|--|---|
| notificationURL | string | Address for successful order payment notification | |
| amount | number | The order amount. Up to 2 decimal | 3.23 |
| currency | string | The currency type. 3 capital letters | "SAR", "AED", "BHD", "EGP", "KWD", "OMR", "QAR" |
| expiredAt | string | The order expires. After the expiration, the order can not be paid. It is recommended to set it to 30 min later, and the default time is 15 days. | "2022-10-23T12:00:21.000Z" |
| refundTimes | number | Times of refunds | 0 |
| refundAmount | number | Total refund amount | 0.99 |
| createdAt | string | Payment order creates time | "2022-10-22T12:00:21.000Z" |
| updatedAt | string | Update date | "2022-12-31T12:53:17.000Z" |
| paidAt | string | Payment date | |
| agencyId | string | Agency ID | "A936252" |

- **Response demo**

HTML

```
{
  "amount": 12.02,          ### Order amount
  "refundAmount": 0,        ### Refund amount
  "id": "M448726T2022123112531745487632",  ### Payment order
ID
  "clientId": "CLT9208307",  ### ClientID
  "merchantId": "M448726",   ### Merchant ID
  "agencyId": "A936252",     ### Agency ID
}
```

```

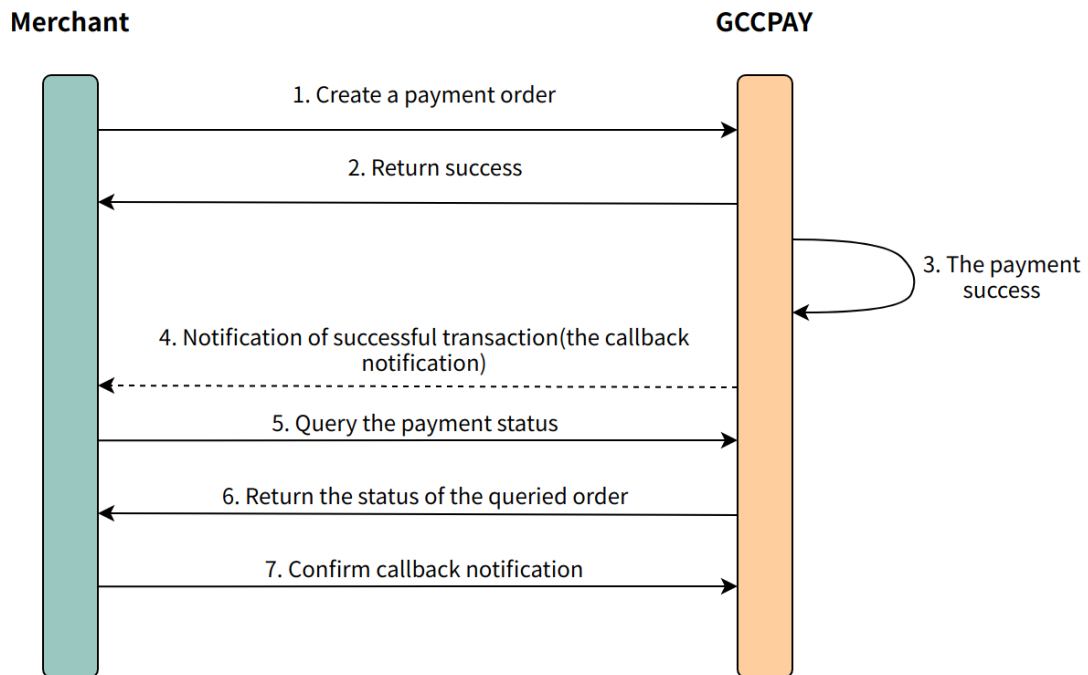
    "status": "pending",          ### Order status
    "ticket":
    "QnTkb04gg5RzB6fPK1fKH9AJgTW7PYQKCYnP0Wwdz0uiQZjSt8UjsHtBJbBBmTmc",
    ### 票据凭证, 用于验证订单 ID
    "name": "Order test for desc",      ### Order description
    "merchantOrderId": "TEST1672491207",    ### Order ID in
merchant
    "notificationURL": " ",          ### Callback address
    "currency": "SAR",              ### Payment currency
    "paidAt": "None",              ### Payment date
    "expiredAt": "2023-01-15T20:53:27.000Z",    ### Expired date
    "refundTimes": "None",          ### Refund date
    "createdAt": "2022-12-31T12:53:17.000Z",    ### Create date
    "updatedAt": "2022-12-31T12:53:17.000Z"      ###
Update date
}

```

4.3 Callback Notification

Summary

- GCCPAY provides notification callback after order payment completion, and will request the corresponding client
- The merchant needs to provide URL which can receive GET request, GCCPAY will send the callback notification after the payment success. And GCCPAY will attach the `_orderId` parameter to the URL, and the parameter value is the order ID.
- **Sequence diagram**



1. Merchant uses GCCPAY API to create a payment order
2. GCCPAY returned success in creating a payment order
3. The payment success
4. GCCPAY sends payment success notification to the URL which is provided by merchant
5. Merchant uses query order API to confirm the order status
6. GCCPAY returns the status of the queried order
7. After merchant confirms the payment status, then returns response to callback notification which body is `COMPLETED: :%ORDER_ID%`
 - `%ORDER_ID%` : The payment order ID notifications.
 - Demo : `COMPLETED: :M448726T2022123112531745487632`

The rules of callback notification

1. If merchants do not support the URL callback notification needed, then the callback notification function will not be triggered.
2. If GCCPAY does not receive callback notification response, then multiple requests will be made. The rules are as follows:
 - The time intervals of the first 10 requests are : 0, 0, 2, 5, 10, 17, 25, 34, 46, 60 second
 - After the 11 times, the interval between each subsequent notification is 2500 seconds, and the max number of notifications is 205

4.4 Query Merchant Information

Summery

- Query the merchant information

HTTP request type : GET

URL : `https://sandbox.gcc-pay.com/api_v1/merchants/{%MERCHANT_ID}`

- {%MERCHANT_ID%} : Merchant ID

Response

【Query Merchant Information】 response as follows :

- **Response Body**

| Parameters | Type | Describe | Example |
|--------------|--------|--|------------------|
| id | string | GCCPAY payment order ID | "M448726" |
| name | string | Merchant name | "GCCPayMerchant" |
| status | string | Merchant status : <ul style="list-style-type: none">• enabled• disabled | "enabled" |
| ownerId | string | Login account | "U771297" |
| name_en | string | User name | |
| CR_file | string | CR file address | |
| CR_number | string | CR number | |
| CR_vatNumber | string | VAT No | |
| CR_expiredAt | string | CR expired date | |
| address | string | Merchant adresss | |
| telephone | string | Merchant telephone | |

| | | | |
|-------------|--------|---|-------------------------------|
| scope | string | Mercahnt business scope | |
| legalPerson | string | Merchant legal person | |
| mcc | string | MCC | "123333" |
| agencyId | string | Agency ID | "A936252" |
| currencys | string | Currency which merchants can use | "SAR,KWD,AED,BHD,EGP,OMR,QAR" |
| createdAt | string | Merchant settlement date | "2022-11-08T23:20:19.000Z" |
| updatedAt | string | Merchant information update date | "2022-12-07T23:26:49.000Z" |
| owner | 数组 | Merchant owner information including: ID, name and mobile | |

- **Response demo**

HTML

```
{
  "id": "M448726",          ### Merchant ID
  "name": "GCCPayMerchant",  ### Merchant name
  "status": "enabled",      ### Merchant status enabled /
disabled
  "ownerId": "U771297",     ### Login account
  "name_en": "null",        ### Merchant name
  "CR_file": None,          ### CR file address
  "CR_number": "null",      ### CR number
  "CR_vatNumber": "null",   ### VAT Number
  "CR_expiredAt": None,     ### CR expired date
  "address": "null",        ### Merchant address
  "telephone": "null",      ### Merchant telephone
  "scope": None,            ### Merchant business scope
  "legalPerson": "null",    ### Merchant legal person
  "mcc": "123333",          ### MCC
  "agencyId": "A936252",    ### Agency ID
  "currencys": "SAR,KWD,AED,BHD,EGP,OMR,QAR",  ### Currency
}
```



```

which merchants can use
  "createdAt": "2022-11-08T23:20:19.000Z",          ### Merchant
settlement date
  "updatedAt": "2022-12-07T23:26:49.000Z",          ### Merchant
information update date
  "owner":          ### Merchant owner infomation
  {
    "id": "U771297",          ### login ID
    "name": "GCCPayMerchant",    ### login name
    "mobile": "17610908585"      ### login telephone
  }

```

Appendix

Test Card Information

| Card NO | Cardholder Name | Expired Date (MM/YY) | CVV | 3D Secure Enrolled |
|------------------------|---------------------------|-----------------------------|-----|-----------------------|
| 5123450000000000 08 | GCCPay_%Merchant Name% | 07 / 24 | 100 | Y |

- %MerchantName% needs to be replaced with the English name of merchant

Order status

- **pending** : Merchant's payment order is pending
- **paid** : The payer has paid the payment order
- **closed** : The payment order is closed
- **expired** : The payment order is expired

Demo Link

Git : <https://github.com/SGATE-IT/gccpay-client-demo>

Payment Transaction limit

| Country | Currency | Minimm payment amount | Maximum payment amount |
|--------------------------|-----------------|------------------------------|-------------------------------|
| Saudi Arabia | SAR | 0.1 | unlimited |
| Kuwait | KWD | 0.25 | unlimited |
| Bahrain | BHD | 0.1 | 300 |
| The United Arab Emirates | AED | 0.1 | 2900 |
| Oman | OMR | 0.1 | 300 |
| Qatar | QAR | 0.1 | 2900 |
| Egypt | EGP | 0.1 | 23800 |

PS: All limits are gateway transaction limits. If the user's bank card has a limit and is less than the gateway transaction limit, the user's bank card limit shall prevail