LipaPay payment interface

- Table of contents

Introduce

1 Range

LipaPay aggregated payment platform is an electronic payment platform specially designed for Ugandan merchants, aiming to provide merchants with a comprehensive payment management solution. The platform supports shilling payment methods. Merchants can integrate shilling payment channels with the platform through a simple API interface, making it easier and more efficient to handle the shilling payment process and improve payment success rate and user experience. In addition, the LipaPay aggregated payment platform supports multiple payment channel providers, including mobile payment channel providers such as MTN and Airtel. Merchants can process and manage payment requests from different payment channel providers through an integrated interface. The platform uses advanced security technology to ensure safe and reliable payment transactions, and only supports Ugandan currency settlement and conversion to meet the payment needs of merchants and consumers in the region. In short, the LipaPay aggregated payment platform provides merchants with more comprehensive, efficient, secure and convenient payment management services, making it an ideal choice for Ugandan merchants.

Intended audience: Directly connected merchants

2 Agreement rules

Request method: POST

Request type: application/json

Encoding: UTF-8

Signature algorithm: MD5

API List

1 Unifiedorder

1.1 Application scenarios

Merchant systems can call this API interface to collect and pay users using MTN or Airtel wallets.

1.2 Request URL

Development--http://dev.pay.lipapayug.com/api/pay/unifiedorder

Production--https://pay.lipapayug.com/api/pay/unifiedorder

1.3 Request parameters

Field	Required	Type(length)	Describe
Version	Y	String(10)	Version number, always fill in v1.0
MchID	Y	Int(8)	The merchant number of the directly connected merchant is generated and issued by LipaPay. Example value：100001
TimeStamp	Y	Long(10)	Timestamp (World Time) Example value：1694757261
Channel	Y	Int(1)	Payment channel (enum: MTN = 1, Airtel = 2) Example value：1
OutTradeNo	Y	String(6..36)	Merchant system internal order number，It can only be numbers, uppercase and lowercase letters_-* and is unique under the same merchant number Example value：UG-20230915-16947572610000001
Amount	Y	Int(8)	Transaction amount (cents) Example value：**500 shillings, should be filled in as: 50000** **The transaction amount must be greater than 500 shilling, and since Aritel does not support transaction amounts with decimals, the amount should be a whole number.**

TransactionType	Y	Int(1)	Transaction type (enum: collections = 1, Disbursement = 2) Example value：1
TraderID	Y	String(10..20)	Payee or payer transaction account number, such as MTN’ s phone number Example value：0750000000
TraderFullName	Y	String(50)	Full name of payee or payer
Description	Y	String(200)	Transaction describe Example value：User recharge
NotifyUrl	Y	String(155)	Transaction result notification address Example value：http://zf.com/api/Notify
Sign	Y	String(32)	Concatenate all the fields except the `Sign` Field in the order of the table fields, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Signature field example：Version=v1.0&MchID=2&TimeStamp=1694772667&Channel=1&OutTradeNo=UG-20230915-16947572610000001&Amount=50000&TransactionType=1&TraderID=0750000000&TraderFullName=Sand Box&Description=User recharge&NotifyUrl=http://zf.com/api/Notify&privateKey=db761034110c45058490c6772a99b4ab

Request parameters example：

```
{
    "Version": "v1.0",
    "MchID": 2,
    "TimeStamp": 1694772667,
    "Channel": 1,
    "OutTradeNo": "UG-20230915-16947572610000001",
    "Amount": 50000,
    "TransactionType": 1,
    "TraderID": "0750000000",
    "TraderFullName": "Sand Box",
    "Description": "User recharge",
    "NotifyUrl": "http://zf.com/api/Notify",
    "Sign": "fc92ceeaa10d8efb2783feecc6aae395"
}
```

1.4 Response parameters

Field	Required	Type(length)	Describe
StatusCode	Y	int	Status code
Succeeded	Y	bool	Whether the request was successful
Errors	Y	String Or String[]	Error message or parameter error collection
Extras	N	String	Extended parameters
Timestamp	N	long	Timestamp (World Time)
Data	Y	JSON	Respond to JSON data
OutTradeNo	Y	String	Merchant system internal order number
TransactionId	Y	String	Platform transaction ID
ActualPaymentAmount	Y	decimal	The actual payment amount paid by the payer (UGX)
ActualCollectAmount	Y	decimal	The actual payment amount of the payee (UGX)
PayerCharge	Y	decimal	Payer's fees (UGX)
PayeeCharge	Y	decimal	Payee's fees(UGX)
ChannelCharge	Y	decimal	Channel fees(UGX)
Sign	Y	String	Concatenate all the fields except the `Sign` Field in the order of the fields in the Data part of the table, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Response parameters example：

```
{
    "StatusCode": 200,
    "Data": {
        "OutTradeNo": "100000006",
      "TransactionId": "a53d3c1c-6b73-485e-86fc-f4999eebd1d4",
      "Amount": 80000.0,
      "ActualPaymentAmount": 80250.000000,
      "ActualCollectAmount": 80000.000000,
      "PayerCharge": 250.000000,
      "PayeeCharge": 0.000000,
      "ChannelCharge": 0.0,
      "Sign": "b8076970398f7f3c1b2d1f533d380fa0"
    },
    "Succeeded": true,
    "Errors": null,
    "Extras": null,
    "Timestamp": 1694772671
}
```

2 Payment result query

2.1 Application scenarios

During wallet payment, due to communication failure, server failure and other reasons, the merchant did not receive the payment result notification in the end. If he is not sure about the background processing result of the payment, he can initiate a "query" transaction through the background to query the transaction order in the wallet background. payment result.

2.2 Request URL

Development--http://dev.pay.lipapayug.com/api/pay/orderquery

Production--https://pay.lipapayug.com//api/pay/orderquery

2.3 Request parameters

Field	Required	Type(length)	Describe
Version	Y	String(10)	Version number, always fill in v1.0
MchID	Y	Int(8)	The merchant number of the directly connected merchant is generated and issued by LipaPay. Example value：100001

TimeStamp	Y	Long(10)	Timestamp (World Time) Example value：1694757261
OutTradeNo	Y	String(6..36)	Merchant system internal order number Example value：UG-20230915-16947572610000001
Sign	Y	String(32)	Concatenate all the fields except the `Sign` Field in the order of the table fields, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Signature field example： Version=v1.0&MchID=2&TimeStamp=1694773149&OutTradeNo=UG-20230915-16947572610000001&privateKey=db761034110c45058490c6772a99b4ab

Request parameters example：

```
 {
     "Version": "v1.0",
     "MchID": 2,
     "TimeStamp": 1694773149,
     "OutTradeNo": "UG-20230915-16947572610000001",
     "Sign": "f8f737d0e444693f274f303780b4eacc"
}
```

2.4 Response parameters

Field	Required	Type(length)	Describe
Field	Required	Type(length)	Describe
StatusCode	Y	int	Status code
Succeeded	Y	bool	Whether the request was successful
Errors	Y	String Or String[]	Error message or parameter error collection
Extras	N	String	Extended parameters
Timestamp	N	long	Timestamp
Data	Y	JSON	Respond to JSON data

PayStatus	Y	int	Payment status (enum: processing=0, payment successful=1, payment failed=2)
PayTime	N	String	Payment time (format: yyyy-MM-dd HH:mm:ss)
OutTradeNo	Y	String	Merchant system internal order number
TransactionId	Y	String	Platform transaction ID
Amount	Y	decimal	Transaction amount (UGX)
ActualPaymentAmount	Y	decimal	The actual payment amount paid by the payer (UGX)
ActualCollectAmount	Y	decimal	The actual payment amount of the payee (UGX)
PayerCharge	Y	decimal	Payer's fees (UGX)
PayeeCharge	Y	decimal	Payee's fees(UGX)
PayMessage	Y	decimal	The message content returned by the service provider, this field is not used for signature
Sign	Y	String	Concatenate all the fields except the \`Sign\` Field in the order of the fields in the Data part of the table, add the private key and perform \`MD5\` encryption to obtain the \`Sign\` signature of the request.

Response parameters example：

```
{
    "StatusCode": 200,
    "Data": {
          "PayStatus": 2,
          "PayTime": "2024-03-07 22:30:28",
          "OutTradeNo": "M-2-3-16340028544581",
          "TransactionId": "02ef7c3e-7be3-4e25-a360-234bd16557f2",
          "Amount": 10000.000000,
          "ActualPaymentAmount": 10101.000000,
          "ActualCollectAmount": 9900.000000,
          "PayerCharge": 101.000000,
          "PayeeCharge": 100.000000,
          "PayMessage": "FAIL",
        "Sign": "d489058436ff660d4dfec7d6859c2e25"
    },
    "Succeeded": true,
    "Errors": null,
    "Extras": null,
    "Timestamp": 1694773152
}
```

3 Payment result callback notification

3.1 Application scenarios

After the payment is completed, the channel party will notify the online payment pre-billing system. After receiving the notification, the billing system will organize the result information to be sent to the merchant's notification address. The merchant's notification address is sent when calling the order information. Note: After receiving the notification, the merchant needs to respond to the notification: respond "SUCCESS" when successfully received; respond "FAILED" when failed. To deal with the problem of repeated notifications: If Online Pay does not receive a SUCCESS response from the merchant, or the network is abnormal, the system will try multiple notifications within 24 hours, with an asynchronous notification frequency of 1/30/30/30s. When merchants receive repeated notifications, they can perform the following 1 or 2 for processing: 1. The merchant actively adjusts the online payment query interface, and the query results shall prevail. 2. The notification message contains the merchant's order number. The merchant can determine whether it is a duplicate notification based on the order number and order status, and ignore duplicate notifications for orders that have been processed.

3.2 Request parameters

Field	Required	Type(length)	Describe
PayStatus	Y	int	Payment status (enum: processing=0, payment successful=1, payment failed=2)
PayTime	N	String	Payment time (format: yyyy-MM-dd HH:mm:ss)
OutTradeNo	Y	String	Merchant system internal order number
TransactionId	Y	String	Platform transaction ID

Amount	Y	decimal	Transaction amount (UGX)
ActualPaymentAmount	Y	decimal	Actual payment amount (UGX)
ActualCollectAmount	Y	decimal	Actual collect amount (UGX)
PayerCharge	Y	decimal	Payer charge (UGX)
PayeeCharge	Y	decimal	Payee charge (UGX)
PayMessage	Y	String	The message content returned by the service provider, this field is not used for signature
Sign	Y	String	Concatenate all the fields except the `Sign` Field in the order of the table fields, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Notification parameter example：

```
  {
      "PayStatus": 1,
      "PayTime": "2023-09-15 10:19:11",
      "OutTradeNo": "UG-20230915-16947572610000001",
      "TransactionId": "4a921193-4737-4f0a-81b7-c12460679f6c",
      "Amount": 0,
      "ServiceCharge": 0,
      "Sign": "d489058436ff660d4dfec7d6859c2e25"
 }
```

4 Prepaid bill

4.1 Application scenarios

Before placing a payment order, you can obtain bill information and check payment parameters such as transaction ID, trader name, amount, handling fee, etc. to ensure the accuracy of the payment request.

4.2 Request URL

Development--http://dev.pay.lipapayug.com/api/pay/bill

Production--https://pay.lipapayug.com/api/pay/bill

4.3 Request parameters

Field	Required	Type(length)	Describe
Version	Y	String(10)	Version number, always fill in v1.0
MchID	Y	Int(8)	The merchant number of the directly connected merchant is generated and issued by LipaPay. Example value：100001
TimeStamp	Y	Long(10)	Timestamp (World Time) Example value：1694757261
Channel	Y	Int(1)	Payment channel (enum: MTN = 1, Airtel = 2) Example value：1
TransactionType	Y	Int(1)	Transaction type (enum: collections = 1, Disbursement = 2) Example value：1
TraderID	Y	String(10..20)	Payee or payer transaction account number, such as MTN’ s phone number Example value：0750000001
Amount	Y	Int(8)	Transaction amount (cents) Example value：500 先令，应填写为：50000
Sign	Y	String(32)	Concatenate all the fields except the `Sign` Field in the order of the table fields, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Signature field example：

Version=v1.0&MchID=2&TimeStamp=1694776093&Channel=1&TransactionType=1&TraderID=0750000000&Amount=50000&privateKey=db761034110c45058490c6772a99b4ab

Request parameters example：

```
{
    "Version": "v1.0",
    "MchID": 2,
    "TimeStamp": 1694776093,
    "Channel": 1,
    "TransactionType": 1,
    "TraderID": "0750000000",
    "Amount": 50000,
    "Sign": "f3e19f695ec9fee2fc3059ad94b19a42"
}
```

4.4 Response parameters

Field	Required	Type(length)	Describe
StatusCode	Y	int	Status code
Succeeded	Y	bool	Whether the request was successful
Errors	Y	String Or String[]	Error message or parameter error collection
Extras	N	String	Extended parameters
Timestamp	N	long	Timestamp (World Time)
Data	Y	JSON	Respond to JSON data
TraderID	Y	String	Payee or payer transaction account number, such as MTN’s phone number
GivenName	Y	String	Given name
FamilyName	Y	String	Family name
FullName	Y	String	Full name
Amount	Y	decimal	Transaction amount (UGX)
ServiceCharge	Y	decimal	Service fee (UGX)
ServiceChargeRate	Y	decimal	Service fee rate (%)

Sign	Y	String	Concatenate all the fields except the `Sign` Field in the order of the fields in the Data part of the table, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Response parameters example：

```
{
    "StatusCode": 200,
    "Data": {
        "TraderID": "0750000000",
        "GivenName": "Sand",
        "FamilyName": "Box",
        "FullName": "Sand Box",
        "Amount": 50000,
        "ServiceCharge": 15,
        "ServiceChargeRate": 3,
        "Sign": "f65b6f9dcdbe670399041ee9e65e13bf"
    },
    "Succeeded": true,
    "Errors": null,
    "Extras": null,
    "Timestamp": 1694776100
}
```

5 Get balance

5.1 Application scenarios

Merchants or sellers can use this API to check their account balance on the platform, ensuring they have sufficient funds to cover any refunds, fees, or outgoing transactionMerchants can use this API to check their account balance on the platform, ensuring they have sufficient funds to cover any refunds, fees, or outgoing transactions.

5.2 Request URL

Development--http://dev.pay.lipapayug.com/api/pay/balance

Production--https://pay.lipapayug.com/api/pay/balance

5.3 Request parameters

Field	Required	Type(length)	Describe
Version	Y	String(10)	Version number, always fill in v1.0
MchID	Y	Int(8)	The merchant number of the directly connected merchant is generated and issued by LipaPay. Example value：100001
TimeStamp	Y	Long(10)	Timestamp (World Time) Example value：1694757261
Sign	Y	String(32)	Concatenate all the fields except the \`Sign\` Field in the order of the table fields, add the private key and perform \`MD5\` encryption to obtain the \`Sign\` signature of the request.

Signature field example：

```
Version=v1.0&MchID=2&TimeStamp=1694776093&privateKey=db761034110c45058490c6772a99b4ab
```

Request parameters example：

```
{
    "Version": "v1.0",
    "MchID": 2,
    "TimeStamp": 1694776093,
    "Sign": "f3e19f695ec9fee2fc3059ad94b19a42"
}
```

5.4 Response parameters

Field	Required	Type(length)	Describe
StatusCode	Y	int	Status code
Succeeded	Y	bool	Whether the request was successful
Errors	Y	String Or String[]	Error message or parameter error collection
Extras	N	String	Extended parameters

Timestamp	N	long	Timestamp (World Time)
Data	Y	JSON	Respond to JSON data
Balance	Y	decimal	Transaction amount (UGX)
Sign	Y	String	Concatenate all the fields except the `Sign` Field in the order of the fields in the Data part of the table, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Response parameters example：

```
{
    "StatusCode": 200,
    "Data": {
        "Balance": 18910.000000,
        "Sign": "4c64e2c5d7b13c78f0afd2715bbf096c"
    },
    "Succeeded": true,
    "Errors": null,
    "Extras": null,
    "Timestamp": 1721054291
}
```

6 Get statement of account

6.1 Application scenarios

Merchants can use this API to obtain a detailed reconciliation statement that includes information on all incoming and outgoing transactions, fees, refunds, and other financial activities on the platform. This helps them reconcile their own records with the platform's data, verify the accuracy of their accounts, and identify any discrepancies that may require further investigation.

6.2 Request URL

Development--http://dev.pay.lipapayug.com/api/pay/statement

Production--https://pay.lipapayug.com/api/pay/statement

6.3 Request parameters

Field	Required	Type(length)	Describe
Version	Y	String(10)	Version number, always fill in v1.0
MchID	Y	Int(8)	The merchant number of the directly connected merchant is generated and issued by LipaPay. Example value：100001
TimeStamp	Y	Long(10)	Timestamp (World Time) Example value：1694757261
StartTime	N	String	Query start time (format: yyyyMMdd)
EndTime	N	String	Query end time (format: yyyyMMdd),Note that the maximum query date range is 3 days
Sign	Y	String(32)	Concatenate all the fields except the `Sign` Field in the order of the table fields, add the private key and perform `MD5` encryption to obtain the `Sign` signature of the request.

Signature field example：

Version=v1.0&MchID=2&TimeStamp=1694776093&privateKey=db761034110c45058490c6772a99b4ab

Request parameters example：

```
{
    "Version": "v1.0",
    "MchID": 2,
    "TimeStamp": 1694776093,
    "StartTime": "20240101",
    "EndTime": "20240103"
    "Sign": "f3e19f695ec9fee2fc3059ad94b19a42"
}
```

6.4 Response parameters

Field	Required	Type(length)	Describe
StatusCode	Y	int	Status code

Succeeded	Y	bool	Whether the request was successful
Errors	Y	String Or String[]	Error message or parameter error collection
Extras	N	String	Extended parameters
Timestamp	N	long	Timestamp (World Time)
Data	Y	JSON	Respond to JSON data
PayStatus	Y	int	Payment status (enum: processing=0, payment successful=1, payment failed=2)
PayTime	N	String	Payment time (format: yyyy-MM-dd HH:mm:ss)
OutTradeNo	Y	String	Merchant system internal order number
TransactionId	Y	String	Platform transaction ID
Amount	Y	decimal	Transaction amount (UGX)
ActualPaymentAmount	Y	decimal	The actual payment amount paid by the payer (UGX)
ActualCollectAmount	Y	decimal	The actual payment amount of the payee (UGX)
PayerCharge	Y	decimal	Payer's fees (UGX)
PayeeCharge	Y	decimal	Payee's fees(UGX)
PayMessage	Y	decimal	Payment message

Response parameters example：

```
{
    "StatusCode": 200,
    "Data": {
        "Items": [
            {
                "PayStatus": 2,
                "PayTime": "2024-03-07 22:30:28",
                "OutTradeNo": "M-2-3-16340028544581",
                "TransactionId": "02ef7c3e-7be3-4e25-a360-234bd16557f2",
                "Amount": 10000.000000,
                "ActualPaymentAmount": 10101.000000,
                "ActualCollectAmount": 9900.000000,
                "PayerCharge": 101.000000,
                "PayeeCharge": 100.000000,
                "PayMessage": "FAIL"
            }
        ]
    },
    "Succeeded": true,
    "Errors": null,
    "Extras": null,
    "Timestamp": 1721056501
}
```

7 Account information

Development-Merchant ID: 2

Development-Signing key: db761034110c45058490c6772a99b4ab

8 Status code

Code	Describe
200	The request is successful and data is returned
202	The request is successful and no data is returned
400	Parameter error, please check the request parameters according to the detailed information returned by the interface
401	Signature error, please check whether the signature parameters and methods meet the signature algorithm requirements.
403	The merchant order number is duplicated. Please verify whether the merchant order number has been submitted repeatedly.

Code	Describe
404	The order does not exist, please check whether the order has initiated a transaction
500	System exception, please call again with the same parameters.
502	The channel system is abnormal, please call again with the same paramet ers.