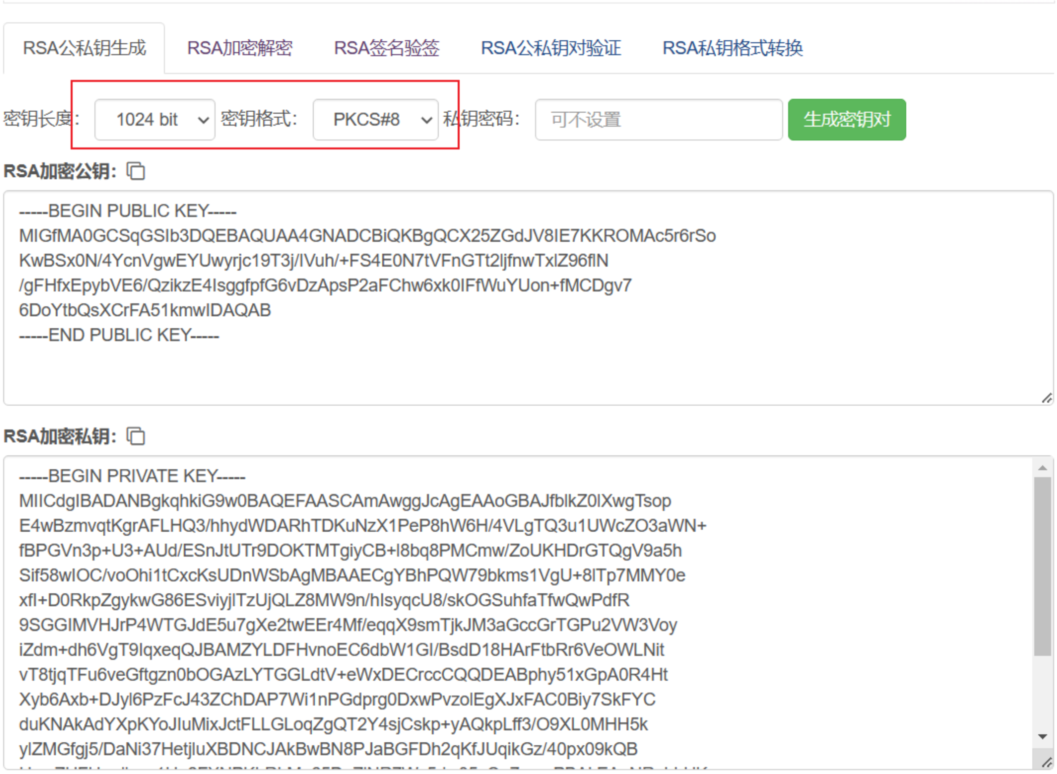
# System docking documents

## docking process

1、Log in to your merchant backend-"Account information-"Set merchant key

2、View the platform public key and upload the merchant public key

3、Generation steps: http://www.metools.info/code/c80.html, generate a pair of PKSC#8 (1024) key pairs. Please save the private key by yourself and upload the public key to the background.



密钥长度: 1024 bit 密钥格式: PKCS#8 私钥密码: 可不设置

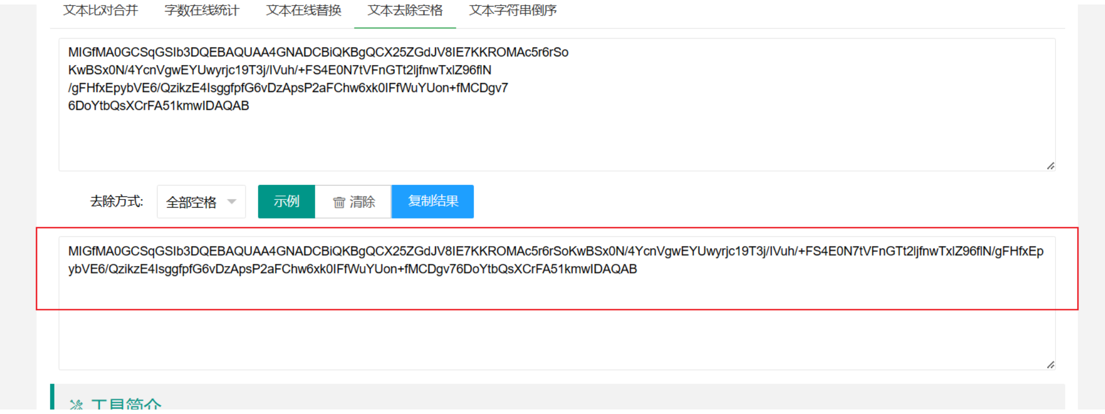
RSA加密公钥:

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCX25ZGdJV8IE7KKROMAc5r6rSo
KwBSx0N/4YcnVgwEYUwyrjc19T3j/IVuh/+FS4E0N7tVFnGTt2ljfnwTxlZ96fIN
/gFHfxEpybVE6/QzikzE4IsggfpfG6vDzApsP2aFChw6xk0IFfWuYUon+fMCDgv7
6DoYtbQsXCrFA51kmwIDAQAB
-----END PUBLIC KEY-----
```

RSA加密私钥:

```
-----BEGIN PRIVATE KEY-----
MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJcAgEAAoGBAJfblkZ0lXwgTsop
E4wBzmvqtKgrAFLHQ3/hhydWDARhTDKuNzX1PeP8hW6H/4VLgTQ3u1UWcZO3aWN+
fBPGVn3p+U3+AUd/ESnJtUTr9DOKTMTgiyCB+I8bq8PMCmw/ZoUKHDrGTQgV9a5h
Sif58wIOC/voOhi1tCxcKsUDnWSbAgMBAAECgYBhPQW79bkms1VgU+8ITp7MMY0e
xfI+D0RkpZgykwG86ESviyjITzUjQLZ8MW9n/hIsyqcU8/skOGSuhfaTfwQwPdfR
9SGGIMVHJrP4WTGJdE5u7gXe2twEEr4Mf/eqqX9smTjkJM3aGccGrTGPu2VW3Voy
iZdm+dh6VgT9IqxeqQJBAMZYLDFHvnoEC6dbW1GI/BsdD18HArFtbRr6VeOWLNit
vT8tjqTFu6veGftgzn0bOGAzLYTGGLdtV+eWxDECrccCQQDEABphy51xGpA0R4Ht
Xyb6Axb+DJyl6PzFcJ43ZChDAP7Wi1nPGdprg0DxwPvzolEgXJxFAC0Biy7SkFYC
duKNAkAdYXpKYoJIuMixJctFLLGLoqZgQT2Y4sjCskp+yAQkpLff3/O9XL0MHH5k
ylZMGfgj5/DaNi37HetjluXBDNCJAkBwBN8PJaBGFDh2qKfJUqikGz/40px09kQB
```

4、Please remove BEGIN/END PRIVATE KEY and BEGIN/END PUBLIC KEY from both public and private keys, and then remove excess spaces (you can use online toolshttp://www.esjson.com/delSpace.html)



```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCX25ZGdJV8IE7KKROMAc5r6rSo
KwBSx0N/4YcnVgwEYUwyrjc19T3j/lVuh/+FS4E0N7tVFnGTt2ljfnwTxlZ96flN
/gFHfxEpybVE6/QzikzE4lsggfpfG6vDzApsP2aFChw6xk0IFfWuYUon+fMCDgv7
6DoYtbQsXCrFA51kmwIDAQAB
```

```
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCX25ZGdJV8IE7KKROMAc5r6rSoKwBSx0N/4YcnVgwEYUwyrjc19T3j/lVuh/+FS4E0N7tVFnGTt2ljfnwTxlZ96flN/gFHfxEp
ybVE6/QzikzE4lsggfpfG6vDzApsP2aFChw6xk0IFfWuYUon+fMCDgv76DoYtbQsXCrFA51kmwIDAQAB
```

5、Bind the IP of the submission service

6、Refer to the interface list below for docking

7、There is a java version of the demo available in the demo.

8、For test merchants, please contact customer service to add

9、Log in to the test backend http://52.74.165.63:81 and operate steps 1-4.

## Interface specification

All interfaces and callbacks use HTTP POST JSON, with the header Content-type: application/json

API address

Test address: 52.74.165.63:8040

Production address: gateway.zippay-in.com

## Interface signature

1、Use the private key to encrypt the parameter json with rsa. Empty and empty strings are not involved (please refer to the RechargeOrderDemo class in the demo).

2、Callback decryption uses the platform public key to exclude the sign parameter and perform rsa decryption. Empty and empty strings are not involved (refer to the RechargeCallbackController class in the demo)

3、The data types must correspond, and the json data needs to be sorted in dictionary order, otherwise the signature may not be possible.

## interface list

### Collection interface

Request address: api address/payment/payin

Encryption rules:

Splice the merchant ID, order ID, amount, nonce, timestamp values and perform rsa encryption

For example
rsa.encryptBase64(payInRequest.getMerchantId() + payInRequest.getMerchantOrderId() + payInRequest.getAmount() + payInRequest.getNonce() + payInRequest.getTimestamp())

| Field name | Is it necessary | type of data | illustrate |
|---|---|---|---|
| merchantId | yes | Long | merchant id |
| merchantOrderId | yes | string | Merchant order id |
| amount | yes | BigDecimal | Amount, such as 10.02 |
| phone | yes | string | User phone number |
| email | yes | string | User email |
| name | yes | string | username |
| uid | no | string | user id |
| ip | no | String | user ip |
| tunnelId | yes | Integer | Channel id (default filled:1203) |
| currency | yes | string | Currency |
| nonce | yes | string | Random string, no more than 32 characters |
| timestamp | yes | Long | The timestamp when the request was initiated |
| sign | yes | string | sign |

Return results synchronously

```
{
"code": 200,
"msg": "Operation successful",
"data": {
"merchantId": 1,
"merchantOrderId": "12356",
"payUrl": "https:\/\/upi.payplus.live\/pay\/insta\/vpa.html?orderId=167549208838755",
"platformOrderId": "1230204142807893191",
"timestamp": 1675492088155
},
"success": true
}
```

Parameter Description

| parameter name | type | illustrate |
|---|---|---|
| code | int | 200 success |
| msg | string | describe |
| success | boolean | whether succeed |
| data | json | Business data, the following data will be returned only when the code is 200 |
| data.merchantId | long | merchant id |
| data.merchantOrderId | string | Merchant order id |
| data.payUrl | string | Payment address |
| data.platformOrderId | string | Platform order number |
| data.timestamp | string | Order timestamp |

## Call back notice on behalf of

After receiving the callback, please return a string no matter what the situation is.successIt means it has been received, otherwise it will be called back multiple times (up to five times). Please do idempotent processing by yourself.(All non-empty fields toJson, and ABCD order sorting encryption)

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 1 is success |
| msg | string | describe |
| merchantOrderId | string | Merchant order id |
| platformFormOrderId | string | Platform order id |
| amount | string | Amount |
| realAmount | string | The actual amount |
| sign | string | sign |

## Collection inquiry interface

Request address: api address/payment/rechargeOrderquery, choose 1 from 3 order numbers, support checking orders of the current month and last month, please bring orderTime for orders outside the range

(All non-empty fields toJson, and ABCD order sorting encryption)

Request fields:

| Field name | Is it necessary | type of data | illustrate |
|---|---|---|---|
| merchantId | yes | Long | merchant id |
| merchantOrderId | no | string | Merchant order id |
| nonce | yes | string | Random string, no more than 32 characters |
| timestamp | yes | Long | The timestamp when the request was initiated |
| platformOrderId | no | string | Platform order id |
| utr | no | String | utr |
| orderTime | no | long | Order timestamp |
| sign | yes | string | sign |

Response fields:

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 200 is successful |
| msg | string | describe |
| data | object | |
| data.merchantId | long | merchant id |
| data.merchantOrderId | string | Merchant order id |
| data.platformOrderId | string | Platform order number |
| data.status | Int | Status, 1 means paid |
| data.orderAmount | string | order amount |
| data.realAmount | string | The amount actually paid |

## Merchant balance inquiry interface

Request address: api address/payment/merchantQuery

(All non-empty fields toJson, and ABCD order sorting encryption)

Request fields:

| Field name | Is it necessary | type of data | illustrate |
|---|---|---|---|
| merchantId | yes | Long | merchant id |
| nonce | yes | string | Random string, no more than 32 characters |
| timestamp | yes | Long | The timestamp when the request was initiated |
| sign | yes | string | sign |

Response fields:

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 1 is success |
| msg | string | describe |
| data | object | |
| data.merchantId | long | merchant id |
| data.payBalance | BigDecimal | Pay balance |

## Payment interface

Request address: api address/payment/payout

Request fields:

Splice the merchant ID, order ID, amount, nonce, timestamp values and perform rsa encryption

For example

rsa.encryptBase64(payInRequest.getMerchantId() + payInRequest.getMerchantOrderId() + payInRequest.getAmount() + payInRequest.getNonce() + payInRequest.getTimestamp())

| Field name | Is it necessary | type of data | illustrate |
|---|---|---|---|
| merchantId | yes | Long | merchant id |
| merchantOrderId | yes | string | Merchant order id |
| amount | yes | BigDecimal | Amount, such as 10.02 |
| phone | yes | string | User phone number |
| email | yes | string | User email |
| uid | no | string | user id |
| ip | no | String | user ip |
| tunnelId | yes | Integer | Channel id (default: 1188) |
| currency | yes | string | Currency |
| nonce | yes | string | Random string, no more than 32 characters |
| timestamp | yes | Long | The timestamp when the request was initiated |
| account | yes | string | Account number (bank card number or vpa) |
| accountName | yes | string | Account name |
| address | yes | string | address |
| subBranch | yes | string | india for ifsc |
| withdrawType | yes | int | 1 is bank card 2 is UPI |
| bankName | yes | string | Bank name |
| remark | yes | string | Remark |
| sign | yes | string | sign |

Response fields:

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 200 is successful |

| | | |
|---|---|---|
| msg | string | describe |
| success | boolean | whether succeed |
| data | object | |
| data.merchantId | long | merchant id |
| data.merchantOrderId | string | Merchant order id |
| data.platformOrderId | string | Platform order id |
| data.timestamp | long | Timestamp |

```
{
        "code": 200,
        "msg": "Operation successful",
        "data": {
                "merchantId": 1,
                "merchantOrderId": "test1234531",
                "platformOrderId": "2230220224108597451",
                "timestamp": 1676904068856
        },
        "success": true
}
```

## Payment callback notification

After receiving the callback, please return a string no matter what the situation is.successIndicates that it has been received, otherwise it will be called back multiple times (up to five times). Please do idempotent processing by yourself.

(All non-empty fields toJson, and ABCD order sorting encryption)

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 1 is success, 2 is failure |
| msg | string | describe |
| merchantOrderId | string | Merchant order id |
| platformFormOrderId | string | Platform order id |
| sign | string | sign |
| utr | string | Utr Bank transaction voucher. If the transaction is successful, most banks will return it. If it is empty, it will not participate in the signature. |

## Payment inquiry interface

Request address: api address/payment/exchangeOrderquery, supports checking orders for the current month and last month, please bring orderTime for orders outside the range

(All non-empty fields toJson, and ABCD order sorting encryption)

Request fields:

| Field name | Is it necessary | type of data | illustrate |
|---|---|---|---|
| merchantId | yes | Long | merchant id |
| merchantOrderId | no | string | Merchant order id |
| nonce | yes | string | Random string, no more than 32 characters |
| timestamp | yes | Long | The timestamp when the request was initiated |
| platformOrderId | no | string | Platform order id |
| utr | no | String | utr |
| orderTime | no | long | Order timestamp |
| sign | yes | string | sign |

Response fields:

| Field name | type of data | illustrate |
|---|---|---|
| code | int | Status code 200 is successful |
| msg | string | describe |
| data | object | |
| data.merchantId | long | merchant id |
| data.merchantOrderId | string | Merchant order id |
| data.platformOrderId | string | Platform order number |
| data.status | Int | Status, 1 means paid |

```
{
    "code": 200,
    "msg": "Operation successful",
    "data": {
        "merchantId": 1,
        "merchantOrderId": "test1234563",
        "platformOrderId": "2230220144722660601",
        "payStatus": 0
    },
    "success": true
}
```

## Error code and description

| error code | illustrate |
|---|---|
| 100000 | Server exception |
| 100001 | Insufficient channel balance |
| 100002 | Signature error |
| 100003 | Merchant channel closed |
| 100004 | Merchant channel is not available |
| 100005 | Merchant public key is not set |
| 100006 | Channel configuration error |
| 100007 | The callback address is not set |
| 100008 | Failed to generate order |
| 100009 | Duplicate order number |
| 100010 | Order number is empty |
| 100011 | Order does not exist |
| 100012 | Merchant query failed |
| 100012 | Order expired |
| 100013 | Order configuration not found |
| 100014 | Wrong amount |
| 100015 | Submission frequency is limited |
| 100016 | Daily quota is limited |