TÀI LIỆU HƯỚNG DẪN XÂY DỰNG CỔNG KẾT NỐI

1 Tổng quan

1.1 Mục đích của tài liệu

Tài liệu mô tả kịch bản MoMo kết nối với đối tác để sử dụng dịch vụ thanh toán, kiểm tra thông tin hóa đơn trực tiếp của người dùng thông qua hệ thống cổng kết nối API.

1.2 Phạm vi sử dụng

Tài liệu này được sử dụng cho kỹ thuật 2 bên trong quá trình tích hợp. Tài liệu chỉ mang tính chất tham khảo, có thể sẽ khác biệt trong quá trình triển khai.

1.3 Quy trình kết nối

- Bước 1: Đối tác xây dựng API:
    - Xây dựng API kiểm tra thông tin nợ (checkInfo)
    - Xây dựng API thanh toán nợ (payment)
    - Xây dựng API kiểm tra trạng thái giáo dịch (checkTrans)
- Bước 2: Đối tác gửi thông tin tài khoản cho MoMo gồm:
    - Đường dẫn (URL) (Môi trường TEST)
    - Thông tin mã hóa của đối tác (*): Public Key RSA (1024bit) v.v… sẽ được sử dụng trong quá trình gọi API ở môi trường TEST
- Bước 3: MoMo gửi thông tin cho đối tác gồm:
    - Thông tin mã hóa của MoMo (**) : Public Key RSA (1024bit) (Môi trường TEST)
    - IP của MoMo. Đối tác có trách nhiệm cho phép IP này gọi vào hệ thống của đối tác.
- Bước 4: Sau khi kết nối và nghiệm thu thành công trên môi trường TEST
    - Thực hiện lại Quy trình kết nối. Cung cấp lại thông tin kết nối và thông tin bảo mật trên môi trường thật (PRODUCTION)

2 Các hàm API

2.1 Check Info

Địa chỉ nhận Request:

https://doitac.com/api/v2/service/checkInfo

Method: POST

Max request time-out: 30 giây

Định dạng: JSON

Tham số	**Mô tả**
username (String, required)	Tên tài khoản của MoMo do đối tác cung cấp
password (String, required)	Mật khẩu của MoMo do đối tác cung cấp
requestTime (Long, required)	Thời gian MoMo gọi sang đối tác (Vd : 1567462678781)
billId (String, required)	Mã truy vấn nợ của khách hàng

dataSign (String, required)	Chữ ký điện tử trên giao dịch tương ứng của MoMo. Chữ ký điện tử theo thuật toán RSA 1024 bit bằng private key của MoMo tạo trong cặp key và cung cấp cho Đối tác public key (**) để xác nhận chữ ký. Nếu xác nhận đúng chữ ký sẽ thực hiện theo thông tin yêu cầu:
	`plainText = username + "|" + password + "|" + requestTime + "|" + billId`
	
	`dataSign = createSign(plainText , "privateKeyMoMo")`

Sau khi xử lý, hệ thống sẽ trả về nội dung ở dạng json bao gồm các tham số như sau:

```
{
     "value": "Message",
     "code": Mã lỗi – Xem phụ lục 3.1,
     "requestTime": "Thời gian MoMo gọi sang đối tác",
     "dataSign": Chữ ký số,
     "transaction": {
          "totalAmount": 30000,
          "accountName": "Nguyễn Van A",
          "accountAddress": "Địa chỉ 123/a1 đường, quận, tp",
          "billDetail": [{
               "paymentId": 11,
               "amount": 10000,
               "cycle": "9/2019",
               "desc": "Phí dịch vụ tháng 9"
               },
               {
               "paymentId": 22,
               "amount": 20000,
               "cycle": "10/2019",
               "desc": "Phí dịch vụ tháng 10"
               }]
     }
}
```

dataSign được mã hóa bằng thuật toán RSA 1024:

```
plainText = code + "|" + requestTime;
dataSign = createSign(plainText , "privateKey Đối tác");
```

2.2. Payment

Địa chỉ nhận Request:

https://doitac.com/api/v2/service/payment

Method: POST

Max request time-out: 30 giây

Định dạng: JSON

Tham số	**Mô tả**
username (String, required)	Tên tài khoản của MoMo do đối tác cung cấp
password (String, required)	Mật khẩu của MoMo do đối tác cung cấp
requestTime (String, required)	Thời gian MoMo gọi sang đối tác (Vd : 1567462678781)
requestId (String, required)	Mã giao dịch phát sinh từ phía MoMo. Trong tất cả các giao dịch phát sinh từ phía MoMo thì mã này không trùng nhau. Mã giao dịch chỉ gồm: [a-z][A-Z][0-9]{_,-}, tối đa 36 ký tự
billId (string, required)	Mã truy vấn nợ của khách hàng
totalAmount (Long, required)	Tổng tiền của các kỳ cần thanh toán
paymentIds(String, required)	Mã thanh toán của các kỳ cần thanh toán (VD: `"11,22"`)
dataSign (String, required)	Chữ ký điện tử trên giao dịch tương ứng của MoMo. Chữ ký điện tử theo thuật toán RSA 1024 bit bằng private key của MoMo tạo

	trong cặp key và cung cấp cho Đối tác public key (**) để xác nhận chữ ký. Nếu xác nhận đúng chữ ký sẽ thực hiện theo thông tin yêu cầu: **plainText** = username + "|" + password + "|" + requestTime + "|" + requestId + "|" + billId + "|" + totalAmount + "|" + paymentIds **dataSign** = createSign(plainText , "privateKey MoMo")

Sau khi xử lý, hệ thống sẽ trả về nội dung ở dạng json bao gồm các tham số như sau:

```
{
    "value": "Message",
    "code": Mã lỗi – Xem phụ lục 3.1,
    "transactionId": Mã giao dịch của đối tác (duy nhất không trùng – MoMo sẽ lưu mã giao dịch này lại để đối soát) ,
    "dataSign": "Chữ ký số"
}
```

dataSign được mã hóa bằng thuật toán RSA 1024:

```
plainText = code + "|" + transactionId;
dataSign = createSign(plainText , "privateKey Đối tác");
```

2.3. checkTrans (Truy vấn, kiểm tra thông tin giao dịch)

Địa chỉ nhận Request:

https://doitac.com/api/v2/service/checkTrans

Method: POST

Max request time-out: 30 giây

Định dạng: JSON

Tham số	Mô tả
username (String, required)	Tên tài khoản đăng ký với FiviPay
password (String, required)	Mã tài khoản API, được cung cấp khi đăng ký tài khoản merchant
requestId (String, required)	Mã giao dịch phát sinh từ phía đối tác. Trong tất cả các giao dịch phát sinh từ phía đối tác thì mã này không được trùng nhau. Mã giao dịch chỉ gồm: [a-z][A-Z][0-9]{_, -}, tối đa 36 ký tự
dataSign (String, required)	Chữ ký điện tử trên giao dịch tương ứng của MoMo. Chữ ký điện tử theo thuật toán RSA 1024 bit bằng private key của MoMo tạo trong cặp key và cung cấp cho Đối tác public key (**) để xác nhận chữ ký. Nếu xác nhận đúng chữ ký sẽ thực hiện theo thông tin yêu cầu: **plainText** = username + "|" + password + "|"+ requestId **dataSign** = createSign(plainText , "privateKey MoMo")

- Sau khi xử lý, hệ thống sẽ trả về nội dung ở dạng json bao gồm các tham số như sau:

```
{
    "value": "Message",
    "code": Mã lỗi – Xem phụ lục 3.1,
    "requestId": "Mã giao dịch của MoMo",
    "dataSign": "Chữ ký số"
}
```

dataSign được mã hóa bằng thuật toán RSA 1024:

```
plainText = code + "|" + requestId

dataSign = createSign(plainText , "privateKey Đối tác")
```

3. Phụ lục

3.1. Mã lỗi

Mã lỗi	Miêu tả
0	Thành công
1	Giao dịch thất bại
2	Hóa đơn này đã được thanh toán
3	Giao dịch không tồn tại
4	Số tiền thanh toán không đúng
5	Thông tin username không đúng
6	Thông tin password không đúng
2000	Tham số đầu vào không đúng
2005	Giao dịch bị trùng lặp
2702	Trùng requestId
4001	Tài khoản không tồn tại
4002	Tài khoản bị tạm khóa
4003	Thông tin kết nối không chính xác
4004	Địa chỉ IP không hợp lệ
4200	Lỗi dịch vụ
4300	Lỗi tham số đầu vào

4301	Ký dữ liệu không đúng (Dữ liệu bị sửa đổi hoặc bị mất)
4302	Ngày yêu cầu không hợp lệ
4303	Mã giao dịch đã tồn tại
4400	Vi phạm cấu hình hệ thống, Quá hạn mức giao dịch cho phép trong ngày
4401	Quá số lần tra cứu giao dịch cho phép
4402	Quá nhiều giao dịch trên một đơn vị thời gian cho phép
4403	Quá hạn mức giao dịch cho phép trong ngày
9000	Giao dịch nghi vấn (timeout)
9999	Lỗi không xác định

4. Thông tin tham khảo

4.1. Tạo chữ ký (create signature)

Java code:

```
public static String createSign(String data, String filePath){
    try {
        final File privKeyFile = new File(filePath);
        final byte[] privKeyBytes = readFile(privKeyFile);
        final KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        final PKCS8EncodedKeySpec privSpec = new PKCS8EncodedKeySpec(privKeyBytes);
        final PrivateKey pk = (PrivateKey) keyFactory.generatePrivate(privSpec);
        final Signature sg = Signature.getInstance("SHA1withRSA");
```

```
            sg.initSign(pk);

            sg.update(data.getBytes());

            final byte[] bDS = sg.sign();

            return new
            String(org.apache.commons.codec.binary.Base64.encode
            Base64( bDS));}

        catch (Exception ex) {

            ex.printStackTrace();
        }

        return "";

}
```

4.2. Verify chữ ký(verify signature)

```
public static boolean checkSign(String sign, String data,
String publicKeyFile) {

        try {

            File pubKeyFile = new File(publicKeyFile);

            byte[] pubKeyBytes = readFile(pubKeyFile);

            X509EncodedKeySpec pubSpec = new

            X509EncodedKeySpec(pubKeyBytes);

            KeyFactory keyFactory =
        KeyFactory.getInstance("RSA");


            PublicKey k =
        (RSAPublicKey)keyFactory.generatePublic(pubSpec);

            Signature signature =

            Signature.getInstance("SHA1withRSA");

            signature.initVerify(k);

            signature.update(data.getBytes());
```

```
            return
        signature.verify(org.apache.commons.codec.binary.Base64.
        decodeBase64(sign.getBytes()));
        }
        catch (Exception ex) {
            ex.printStackTrace();
            System.out.println(ex.getMessage());
        }
        return false;
}
```

4.3. Quản lý tài liệu

Version	Ngày	Nội dung	PIC
1.0.0	08/03/2019	Khởi tạo	Đặng Công Toàn
2.0.0	18/06/2019	Thêm mã hóa dữ liệu và xác nhận chữ ký	Đặng Công Toàn
2.1.0	03/09/2019	Cập nhật tài liệu	Đặng Công Toàn