

MOMO LOAN API Specification Doc

Version	Date	Author	Description
1.0	24-12-2018	Diep Nguyen	Created
1.1	19-03-2019	Diep Nguyen	<ul style="list-style-type: none"> - Change field “loanId” to “momoLoanId” - Add “postcode” and “dueDate” in Get Loan Data API. - Update field “photos” in Get Loan Data API. - Add “reasonId” in Update Status API - Update Object “detail” in Disburse API and Update Status API
1.2	20-03-2019	Diep Nguyen	<ul style="list-style-type: none"> - Change field “fullName” to “firstName”, “middleName”, “lastName” in Get Loan Data API.
1.3	20-03-2019	Diep Nguyen	<ul style="list-style-type: none"> - Add categories for “reasonId” in Update Status API
1.4	21-03-2019	Diep Nguyen	<ul style="list-style-type: none"> - Add “ward” and “productCode” in Get Loan Data API.
1.5	28-05-2019	Diep Nguyen	<ul style="list-style-type: none"> - Edit ErrorCode ,change field “feeDetail” to “feeDetails” in Disburse API, change type LONG to String of fields “momoLoanId”
1.6	03-06-2019	Diep Nguyen	<ul style="list-style-type: none"> - - Add status (3: CUSTOMER CONFIRMED) in Update Status API.
1.7	10-06-2019	Diep Nguyen	<ul style="list-style-type: none"> - Add CheckStatus API.

Introduction.....	4
Flows.....	7
1. Get Loan Data	8
2. UpdateStatus	12
3. Disburse.....	15
4. CheckStatus	19

Introduction

This document describes the API methods that Merchant uses to interact with the Momo.

- Interaction happens over HTTPS using POST requests.
- Each method accepts a single request stream and returns a single response stream.
- Once decrypted, request and response streams are JSON formatted.

Momo needs to provide Partner with:

- Base URL of the endpoints that Momo will hit
- SSL Certificate
- PGP public key for Merchant to encrypt the message body
- Partner-code
- Password (using in some module)

Merchants needs to provide the following data to Momo:

- PGP public key for Momo to encrypt the message body d
- Server IP

Method

All APIs exposed are RESTful API's with request and response payloads formatted as JSON.

Header

Name	value
Content-Type	application/pgp-encrypted
Accept	application/pgp-encrypted
partner-code	Momo will provide when integrate

Payload

Common Request Parameters

Name	Type	Explanation
request_id	string	Unique identifier on Momo's side to track each request.

Common Response Parameters

Name	Type	Explanation
requestId	string	Unique identifier for the request generated by Momo. Should match the request_id from the request. For example: req_12345.

Name	Type	Explanation
referenceId	string	Unique identifier for the request on the Provider's side. For example: ref_12345.
responseCode	int	response_code tells the status of the request
responseMessage	string	response_message returns a human readable string to identify the issue. Momo would only use response_code to distinguish whether this request succeeded or not

HTTP Status Codes and Response Code

HTTP Status Codes

Status Code	Description
200	Success
400	Malformed request
401	Unauthorized
404	Not found
500	Server error

Common Response Code

Response code	Sample Response message	Description
0	Success	No error
2	Decryption Fail	Cannot decrypt the request
4	Invalid wallet Id	
8	Duplicate request Id	
9	Invalid partner code	Fail
10	Unknown result	Check by hand, or call check status to get the final result
11	Other error	Fail

Encryption

1. All requests from partner will be encrypted by the Momo's public key and signed with merchant's private key.

2. All requests from Momo will be encrypted by the partner's public key and signed with Momo's private key.
3. All encrypted messages must be in ASCII-armored format.

Sample Raw message before decrypt:

-----BEGIN PGP MESSAGE-----

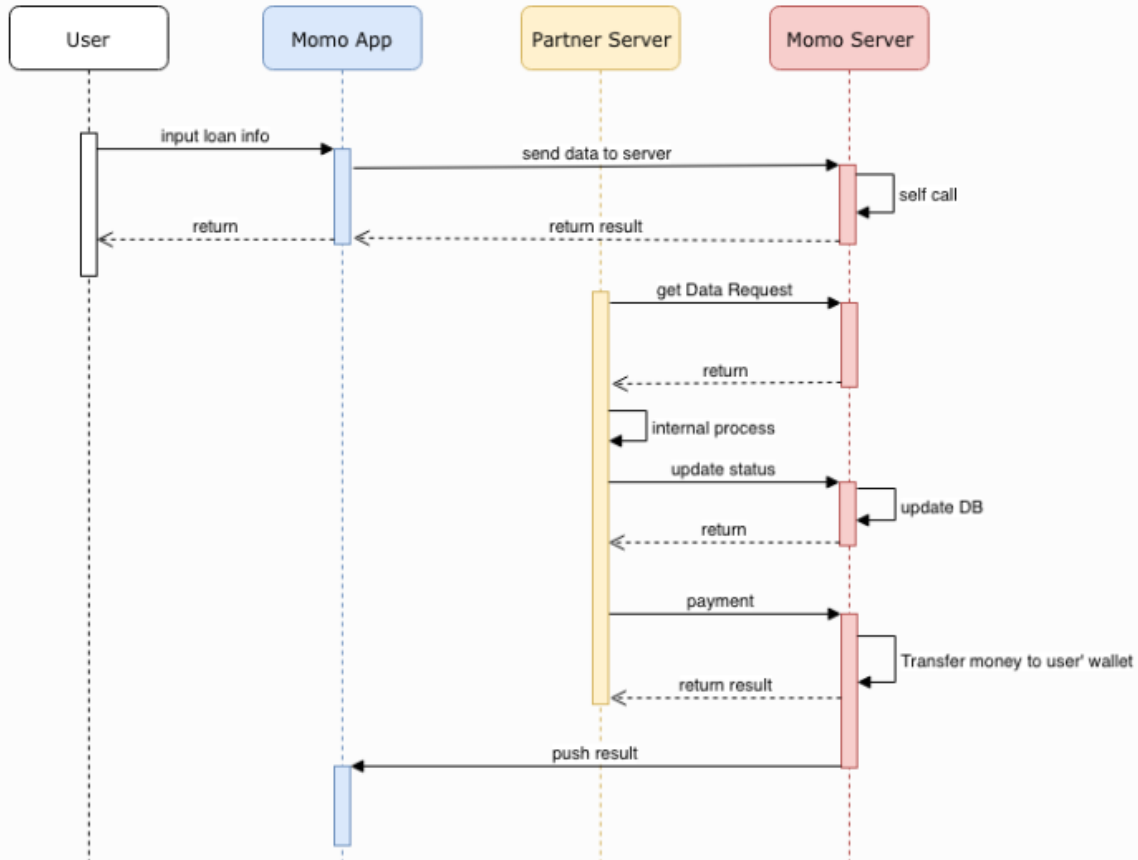
Version: GnuPG v2.0.14 (GNU/Linux)

```
hQEMAzCLjonquyAfAQf/XwqTeJL7z5H7JR55Th8aJhiO0P+AI7dr8abPTTV2j35P
541JnBrIYn+Ee1phRmxWe7BBNPjSpQTd7y+k2eArAmk8TLIZmVKcIr9QfjdtKhjT
HFp2FircA5s3uzHh2jauT8vV3XhQHd2mrdUovlVlMuMdlwhHDXxi+6ZVOf7L1oOM
0duHbQZCV/qcantgN8P6tWTIE/aMyFp+sijoLpCyZ5RTbQG/tnUddgNLBU/Ivwm
TNX8GVdDp1gnOi48gwbllQq0TjLZVhBt3aUN7tK8yONWVfwTFopIhr9Yc5RK3WJv
SAwpHFUe2sWpNFyflKKhgb2du0yzls3BL7iRmVF21sdLpAdQ99m0N7rYBzEzrXp9Y
URXqo+FRjLZbYBoS5XoNMUa9ACHmchPtd1uiwRE3S5n07DEQEoblq1wR6w1mK7e
vh3w+TvdUTJhu9EMIMLxNU37meUJeuso3S6Ly8E356QvLTgCEjS1gQiAwQDeyGbD
NYDBIGvQC2euJMfCUfSOUL9hkqi2UItgNmIF4znXLF1OsdWqK08n44H+9vcMlSh1
Lq8IQWLu56T/ft7+2GNrQ9hAgEMMr1rMYSLP5Go5/JxvFCoaUADY0GE70oXR2+K1
7Hbw+VvYVsrJwALbcBKnmH8kzmHq+2KNkwgTcA30javdMj/9aOQtewP4ZOICQi+W
4KrND/uT3UxRS9DDuF+B+LZPVPE5s8qdT98CKrQ9oeqZ8Kfw5gELCXy6HiixccfK
eCSY0xEP5V16/fmeS5GBUNJWjFUi7Wi8+GXbmvgb7QTm+EFU9UFV9dsUCArRxf8z
8rIz7wAPtTx0Vn0Ux70PKiEpA2CqzqviERQ1uym5kPdbpL5nDT7kMzsnM8Hplym
ovrTTjhOITKzr60d3nZV9WU4r+c8FUYdSvnFAD2Vys3pUpmV7Rokl1el6dt0zBUh
V7nWqUjgJi6/z4vgpNmFxeKHMN0+Hd9VwnL3v+RJbgBx2A+EyRj8Re+xGGIWMGZ+
UpfSfKicPeyt+qVsO/YBsLR6jTn0DASax9wSkP/4VQzfVYRY2Fdf9wLoRNBPw7hk
mtu6zfgxFX/R25iRMbN/xjrdlxzvJG5zlooaSS8rI2xGNgyhBimB7Ha9KwtC2Snr
IIY2sfDa5W/br/oNNEME+c5oEF+soVADZOLQ2wcRzMIPehyCckGCdAAmX14=
=OnzF
```

-----END PGP MESSAGE-----

Flows

Create loan order on Momo App



1. Get Loan Data

Allow partner get loan information.

Example:

Request:

```
{
  "requestId": "1511347353577",
  "from": 1545666902332,
  "to": 1545666902332,
  "eventOffset": 1,
  "numberOfEvents": 4
}
```

Response:

```
{
  "requestId": "wqrwpo2343240xsad213",
  "referenceId": "2313_e3w24_ddqwdq_dasdasda",
  "resultCode": 0,
  "message": "Success",
  "eventOffset": 0,
  "nextEventOffset": 4,
  "totalEvents": 10,
  "datas": [
    {
      "momoLoanId": "4657",
      "productCode": "loan",
      "createDate": 1545666902332,
      "lastName": "Nguyen",
      "middleName": "Ngoc Linh",
      "firstName": "Lan",
      "phoneNumber": "0987654321",
      "personalId": "242354653",
      "issuePlace": "CA.TPHCM",
      "issueDate": "01/01/2018",
      "email": "mail@gmail.com.vn",
      "maritalStatus": "Single",
      "dateOfBirth": "01/01/1980",
      "gender": "MALE",
      "salary": 20000000,
      "address1": "2",
      "address2": "123 Đường XYZ, P.8",
      "ward": "p.9",
      "district": "Q.Phú Nhuận",
      "city": "TP.HCM",
      "postCode": 70000,
      "amount": 5000000,
      "loanTime": 12,
      "dueDate": 3,
      "photos": [
        {
          "type": "National ID Back",
          "link": "https://link.image.1"
        },
        {

```


Name	Type	Required	Level	Description
				events will be returned starting with the third event. If unspecified, eventOffset will be assumed to be zero., Note: If eventOffset exceeds the total number of event(i.e. eventOffset >= totalEvents) then no events will be returned.
numberOfEvents	Number		L1	OPTIONAL: Number of events to show per page. If unspecified or greater than 1000, this will be 1000.

Response:

Name	Type	Required	Level	Description
requestId	String	x	L1	Same with request
resultCode	Integer	x	L1	Result Code of request
referenceId	String	x	L1	Unique for each response
message	String	x	L1	Result message
eventOffset	Number	x	L1	AREQUIRED: The event offset of this response.
nextEventOffset	number		L1	OPTIONAL: The offset of the next event to return. If unspecified there are no more events to retrieve for this statement.
totalEvents	Number	x	L1	
datas	JsonArray	x	L1	Array of JsonObject include these information
momoLoanId	String	x	L2	Loan id
productCode	String	x	L2	Product Code

Name	Type	Required	Level	Description
createDate	Number	x	L2	time in miliseconds. UNIX Time
lastName	String	x	L2	Loan owner lastName
middleName	String	x	L2	Loan owner middleName
firstName	String		L2	Loan owner firsrName
phoneNumber	String		L2	Phone number
personalId	String		L2	Personal Id
issueDate	String		L2	Identification card's created date
issuePlace	String		L2	Issue place of Identification card's
email	String		L2	Email
maritalStatus	String		L2	Marital Status
dueDate	number		L2	MonthlyPaymentAmount
dateOfBirth	String		L2	Format (dd/mm/yyyy)
gender	String		L2	Male/female
salary	number		L2	salary
address1	String		L2	Building address
address2	String		L2	Full address
city	String		L2	City
ward	String		L2	ward
district	String		L2	District
postCode	number		L2	Postal Code
amount	Integer		L2	Loan amount
loanTime	Integer		L2	loan time (month)
photos	JsonArray		L2	ArrayObject of photos url
type	String		L3	Type of photos

Name	Type	Required	Level	Description
link			L3	Photos url
references	JSONArray		L2	Array of JsonObject about reference person
fullName	String		L3	Reference person name
phoneNumber	String		L3	Reference person' phone number
relation	String		L3	Relationship with owner. "husband,wife, workmate, relatives"
personalId	String		L3	

2. UpdateStatus

Allow partner call api to update status loan.

Example:

Request:

```
{
  "requestId": "1511347353577",
  "momoLoanId": "5346",
  "phoneNumber": "098987988",
  "createdTime": 1545666902332,
  "status": 5,
  "description": "Khách hàng bị nợ xấu",
  "reasonId": 1,
  "detail": {
    "tenor": 12,
    "amount":15000000,
    "EMI": 1334562,
    "dueDate": 7,
    "applicationId": "GD6243654"
  }
}
```

Response:

```
{
  "requestId": "wqrwpo2343240xsad213",
  "referenceId": "2313_e3w24_ddqwdq_dasdada",
  "resultCode": 0,
  "message": "Success"
}
```

Request:

- Method: POST
- Url: /api/loan/update
- **Payload:**

Name	Type	Required	Level	Description
requestId	String	x	L1	Unique String for each request
momoLoanId	String	x	L1	Got from get Data
phoneNumber	String	x	L1	Got from Get Data
createdTime	String	x	L1	time in milisecond. UNIX time
status	Number	x	L1	Loan status: 1: PROCESSING 2: APPROVED AND WAITING SMS 3: CUSTOMER CONFIRMED 4: DISBURSED 5: CANCELLED 6: REJECTED 7: LIQUIDATION
description	String		L1	Required in case status equal CANCEL
reasonId	Number	x	L1	Required in case status equal CANCEL or REJECTED. 1 : Passed Eligible 2 : Failed Auto Rejection 3 : Failed Bad debt 4 : Failed Cannot contact 5 : Failed Customer cancel 6 : Failed Document not qualified 7 : Failed FV result 8 : Failed Negative info 9 : Failed Payment Capacity 10 : Failed Product 11 : Failed Ask more checks 12 : Failed Other code 13 : Failed Other (No code) 14 : Note Free text
detail	Json object	x	L1	Loan information if status is APPROVED
tenor	Number	x	L2	Loan duration (month)
amount	Number	x	L2	Total loan amount

Name	Type	Required	Level	Description
EMI	Number	x	L2	MonthlyPaymentAmount
dueDate	Number	x	L2	MonthlyPaymentDate
applicationId	String	x	L2	loanID of TPFICO

Response:

Name	Type	Required	Level	Description
requestId	String	x	L1	Same with request
resultCode	Number	x	L1	Result Code of request
referenceId	String	x	L1	Unique for each response
message	String	x	L1	Result message

Error Code:

Include ErrorCode in **Onboarding/Common Response Code** and **ErrorCode** below.

Code	Description	Status
0	Success	Success
8	Duplicate request Id	Fail
901	Invalid loan id	Fail

3. Disburse

Allow partner transfer money from their wallet to customer wallet

Example:

```

Request:
{
  "requestId": "wqrwpo2343240xsad213",
  "password": "4342543",
  "momoLoanId": "43547",
  "phoneNumber": "098987988",
  "createDate": 1545302933339,
  "totalAmount": 2000000,
  "actualAmount": 1900000,
  "feeDetails": [
    {
      "type": "Insurrance",
      "amount": 50000
    },
    {
      "type": "Service",
      "amount": 50000
    }
  ],
  "detail": {
    "loanId": "GD1235",
    "tenor": 12,
    "rate": 7.9,
    "disbursementDate": "12/12/2018",
    "maturityDate": "11/12/2019",
    "dueDate": 11,
    "EMI": 1200000,
    "paymentBankAccount": "3545567687",
    "firstInstallmentDate": "11/02/2018",
    "firstInstallmentAmount": 1000000
  },
  "description": "Khách hàng được giải ngân"
}
Response:
{
  "requestId": "wqrwpo2343240xsad213",
  "referenceId": "2313_e3w24_ddqwdq_dasdasda",
  "resultCode": 0,
  "message": "Success",
  "paymentRef": "frghdfg434575",
  "actualAmount": 1900000,
}

```

Request:

- **Method:** POST
- **Url:** /api/loan/disburse
- **Payload:**

Name	Type	Required	Level	Description
requestId	String	x	L1	Unique String for each request
password	String	x	L1	Momo will provide when integrate
momoLoanId	String	x	L1	Got from Get Data
phoneNumber	String	x	L1	Got from Get Data
createDate	String	x	L1	created date (milisecond, UNIX Time)
totalAmount	Number	x	L1	total amount of this loan contract
actualAmount	Number	x	L1	actual amount that user can receive. Exclude all fees
feeDetails	JsonArray		L1	Fee information. (JsonArray)
type	Number	x	L2	Type fee
amount	Number	x	L2	Fee amount.
description	String	x	L1	Short description for loan.
detail	Json object	x	L1	loan information
loanId	Number	x	L2	Loan Id of partners
tenor	Number	x	L2	loan duration (month)
rate	Number	x	L2	rate
disbursementDate	String	x	L2	Format: dd/mm/yyyy
maturityDate	String	x	L2	Expired Date Format: dd/mm/yyyy
firstInstallmentDate	Number	x	L2	First Payment Date . Format: dd/mm/yyyy
firstInstallmentAmount	Number		L2	First Payment Amount.
dueDate	Number	x	L2	MonthlyPaymentDate
EMI	Number	x	L2	MonthlyPaymentAmount
paymentBankAccount	String		L2	BankAccount number to payment

Response:

Name	Type	Length	Description
requestId	String	50	Same with request
resultCode	Number	2	Result Code of request
referenceId	String	30	Unique for each response
message	String	200	Result message
acceptedAmount	Number	20	Amount can cashIn.
paymentRef	Number	20	Unique number for each payment request

Error Code:

Include ErrorCode in **Onboarding/Common Response Code** and **ErrorCode** below.

Code	Description	Status
0	Success	Success
5	Invalid password	Fail
8	Duplicate request Id	Fail
9	Invalid partner code	Fail
901	Invalid loan id	Fail
903	Loan Id and phoneNumber does not match	Fail
904	Amount too small	Fail
905	Amount too big	Fail
906	Loan is disbursed	Fail
1004	Wallet cap exceeded	

4. CheckStatus

Allow partner check status of request disbursement from partner.

Request:

```
{
  "requestId": "2300699797",
  "checkRequestId": "2300699797"
}
```

Response:

```
{
  "requestId": "2300699797",
  "referenceId": "MM_13200699797",
  "resultCode": 0,
  "message": "Success",
  "data": {
    "requestId": "wqrwpo2343240xsad213",
    "referenceId": "2313_e3w24_ddqwdq_dasdasda",
    "resultCode": 0,
    "message": "Success",
    "localMessage": "Thành công",
    "paymentRef": "frghdfg434575",
    "actualAmount": 1900000
  }
}
```

Request:

- Method: POST
- Url: /api/loan/checkStatus
- **Payload:**

Name	Type	Required	Description
requestId	String	X	Unique String for each request
checkRequestId	String	X	It depends on request disbursement of partner

Response:

Name	Type	Required	Level	Description
requestId	String	X	L1	Same with request

Name	Type	Required	Level	Description
resultCode	Number	X	L1	Result Code of request
referenceId	String	X	L1	Unique for each response
message	String	X	L1	Result message
data	Number	X	L1	Amount can cashIn.
requestId	String		L2	Same with request disbursement
resultCode	Number		L2	Result Code of request disbursement
referenceId	String		L2	Unique for each response
message	String		L2	Result message
acceptedAmount	Number		L2	Amount can cashIn.
paymentRef	Number		L2	Unique number for each payment request disbursement

Error Code:

Include ErrorCode in **Onboarding/Common Response Code** and **ErrorCode** below.