

QoS INTEGRATED CENTER INTERFACE SPECIFICATION FOR MOBILE MONEY

Component Name	QoS-IC Mobile Payment Integration Documentation
Document Number	SSL-QS-100
General Description	Detailing exposed interfaces between QoS-IC and Integrating partner applications
Target Audience	Integrating Partners

Created By: Toluwase Ashaolu
Approved By:

Document Scope

This document describes how a third-party service provider will call MobilePayment© Interface from QoS-IC. This implementation addresses connection to a Mobile payment platform through a secure and lighter Interface.

Modification Log

V	Author	Date	Summary of Changes
1.0	Toluwase Ashaolu	07 th July,2019	Initial Draft

References

S/N	Document Ref	V	Title

Contents

1. Introduction	1
1.1 Overview	1
2. Design	1
2.1 Overview	1
3. Request Payment	1
3.1 RequestPayment Request (Sent from Third-party)	1
3.2 RequestPayment Response	2
4. Get Transaction Status	3
4.1 Get Transaction Status Request (Sent from ThirdParty).....	3
4.2 Get Transaction Status Response.....	4
6. Communication	7
7. Security	7
8. Response Code	7
9. Connecting to QoS Test Platform	7
APPENDIX A	8

1. Introduction

1.1 Overview

This document describes how a third party will connect to a MobilePayment© Interface from QoS-IC. Note that all data formats and response definitions are in conformance with the REST standard.

2. Design

2.1 Description

A MobilePayment© transaction provides Mobile Money capability application programming interfaces (APIs) for third-party applications (App for short) to connect to it and use its Mobile Money capability for Making Payment and Querying Subscriber's account in a third-party system.

The subsequent sections show the message structure for Mobile Payment

3. Request Payment

The RequestPayment is a request to debit mobile money from subscribers account.

Partners must code the App based on the API field requirements so that the App can send correct requests accordingly.

3.1 RequestPayment Request (Sent from Third-party)

The following describes the important element required to be sent for RequestPayment request.

S/N	Data Element (DE)	M/O	Description
1.	msisdn	MANDATORY	Subscriber's account i.e. 22967307747
2.	amount	MANDATORY	Transaction amount i.e. 5000
3.	firstname	OPTIONAL	Firstname of the subscriber
4.	lastname	OPTIONAL	Lastname of the subscriber
5.	transref	MANDATORY	Unique transaction reference. To be used as unique identifier in call to request for a payment.
6.	narration	OPTIONAL	Comment in English alphabet and number. Special characters not allowed

7.	clientid	MANDATORY	A unique generated id for each third-party application. This id is generated by qos and should be included in all payment requests.
----	----------	-----------	---

Sample Request

```
POST {context}/QosicBridge/user/requestpaymentmv
```

```
Content-Type: application/json
```

```
[Basic Authentication headers]...
```

```
{
  "msisdn": "22967307747",
  "amount": "1",
  "firstname": "David",
  "lastname": "Ashaolu",
  "transref": "12345",
  "narration": "payment",
  "clientid": "MOOVTEST"
}
```

3.2 RequestPayment Response

The following describes the important data element required for a PaymentRequest Response.

S/N	Data Element(DE)	Description
1.	responsecode	This is the response code for the transaction i.e 0
2.	responsmsg	This describes the response message
3.	transref	This is the transref sent in the request message. Each transref must be unique for every request

Successful Response:

```
HTTP Status Code: 202
```

```
{  
  "responsecode": "0",  
  "responsemsg": "SUCCESS",  
  "transref": "12345",  
  "comment": "payment"  
}
```

Un-Successful Response (Failed authentication):

```
HTTP Status Code: 401
```

```
HTTP Status 401: Bad credentials
```

4. Get Transaction Status

A Get Transaction Status message is a request to get the details of an already processed transaction.

4.1 Get Transaction Status Request (Sent from ThirdParty)

The following describes the important element required to be sent for Get Transaction Status request.

S/N	Data Element(DE)	M/O	Description
1.	clientid	MANDATORY	The Unique generated id provided by QoS
2.	transref	MANDATORY	The transaction reference of the payment transaction

Sample Request

```
POST {context}/QosicBridge/user/gettransactionstatus
```

```
Content-Type: application/json
```

```
[Basic Authentication headers]...
```

```
{  
  "transref" : "12345",  
  "clientid": "QOS3P001"  
}
```

4.2 Get Transaction Status Response

The following describes the important data element required for a Transaction Status Response.

S/N	Data Element(DE)	Description
1.	responsecode	This is the response code for the transaction i.e 00
2.	responsemsg	This describes the response message
3.	transref	This is the transref sent in the request message

Successful Response:

```
HTTP Status Code: 200
{
  "responsecode": "0",
  "responsemsg": "SUCCESS",
  "transref": "12345",
  "comment": "payment"
}
```

Un-Successful Response (Failed authentication):

```
HTTP Status Code: 401
HTTP Status 401: Bad credentials
```

5. Communication

Requests will be sent over the REST protocol.

6. Security

Requests will be sent over HTTPS only. Each request is also sent using a Basic Authentication. With Basic Authentication, clients send its Base64 encoded credentials **with each request**, using HTTP [Authorization] header. That means each request is independent of other request and server may/does not maintain any state information for the client.

7. Response Code

Please refer to the standard HTTP response codes. Where anything starting with 2XX signifies approved, 4XX means client error and 5XX indicates server error. See Appendix A for response code descriptions.

8. Connecting to QoS Test Platform

Connection to QoS test platform requires login. To request for login credentials, please send email to support@qosic.com and adebayo.lakatan@qosic.com

APPENDIX A

Sample response Codes obtainable are below. Note that this list is not exhaustive.

Response Code	Description	Possible Cause
0	Success	Transaction Completed
92	Exception on USSD PUSH	Timeout in USSD PUSH/ Cancel in USSD PUSH
94	Transaction not exists	Transaction not Exists
95	Transaction Failed	Transactions was failed due to <error>
91	Parameters Incomplete	Parameters not complete
98	Invalid Token	Invalid Credentials
99/-1	System Busy	Interface Internal error Database connection error