

FlexiPay Wallet

Payments Interface Specification Document V 4.0

About this document

Purpose

This document explains how to integrate to SBU FlexiPay wallet through the Webservice interface. FlexiPay wallet uses eConnect ESB is the middleware to connect FlexiPay managed channels e.g., POSs, mobile applications, USSD, portals to wallet and to the 3rd party providers.

Intended Audience

- FlexiPay / SBU team.
- FlexiPay wallet integrators and aggregators on the Web Service Interface.

Objective and Scope

- To provide sufficient technical details for a seamless integration to the ESB. This document version will cover the following transactions:

	Transaction
1.	Pay Merchant
2.	Initiate Approval
3.	Transfer to Wallet
4.	Transfer to MNO
5.	Transfer to Bank Account
6.	Bulk payments
7.	Third Party Collections through FlexiPay
8.	Biller Integration through FlexiPay
9.	Aggregator balance check

Revisions in This Version

	Transaction
1.	Updated Transfer to Bank API

Contents

About this document	2
Purpose	2
Intended Audience	2
Objective and Scope.....	2
Revisions in This Version	2
Glossary & Acronyms of Terms.....	4
Communication Protocol.....	4
Message Standard	4
Merchant Payments	5
Request Authentication	5
Sample Code	5
API Requests and Responses	5
1. Merchant Lookup	5
2. Merchant Payment – From Customer Mobile Money Wallet	6
3. Get Merchant Payment Status (Mobile Money Payments).....	7
4. Initiate Approval – From Customer FlexiPay Wallet	8
5. Get Customer Approval Status (FlexiPay Payments)	9
Request and Response Message Fields	10
Bulk Payments to FlexiPay	12
Request Authentication	12
Sample Code	12
API Requests and Responses	13
1. Bulk Payments	13
Funds Transfer	17
Request Authentication	17
Sample Code	17
API Requests and Responses	18
1. Get Wallet Details.....	18
2. Transfer to FlexiPay Wallet	18
3. Transfer to Mobile Money Wallet	19
4. Transfer to Bank	20

5. Get Transaction Status	20
Third Party Collections through FlexiPay.....	21
1. Customer Validation.....	21
2. Customer Payment	21
Bill Payments	22
Request Authentication.....	22
Sample Code	22
API Requests and Responses	23
1. Customer validation at biller.....	23
2. Bill Payment	23
3. Check aggregator wallet balance.....	25
Biller Service IDs	26
Request and Response Message Fields	27
Response Codes	30

Glossary & Acronyms of Terms

Abbreviation	Meaning
ESB	Enterprise Service Bus
ISO	International Standards Organization
MOMO	Mobile Money
C2B	Customer to Business
TCP	Transmission Control Protocol
Dig-Signature	Digital Signature

Communication Protocol

- A RESTFUL web service over TCP/IP

Message Standard

A customized JSON derivation of the ISO 8583 standard.

Merchant Payments

Request Authentication

Field RequestID is mandatory. It is user to authenticate each, and every request sent and should vary from one request to another.

Each request should have Client_ID and Token headers, which are used for authentication.

Client ID is a constant parameter which is shared in another platform.

Token is a result of hashing mandatory fields using HmacSHA256(Client_ID+RequestID+msisdn+Amount) then encoding to base 64

```
Token= SHA256(base64(Client_ID+RequestID+msisdn+Amount))
```

Sample Code

```
private static void generateMerchantPaymentToken(String clientID, String requestID, String msisdn, String amount)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = clientID+requestID+msisdn+amount;

    String secretKey = "Passsword";

    Mac sha256Mac = Mac.getInstance("HmacSHA256");
    SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes(), "HmacSHA256");

    sha256Mac.init(secretKeySpec);

    String token = new String(Base64.getEncoder().encode(sha256Mac.doFinal(data.getBytes())));
}
```

API Requests and Responses

1. Merchant Lookup

Merchant lookup API URL

https://<IP Address>:<port>/merchants/v1.0/validatemerchantdetails

Merchant Lookup Request

```
{  
  "msisdn": "256759889380",  
  "RequestID": "20210706033645",  
  "Merchant_Code": "20400054"  
}
```

Merchant Lookup Response

```
{  
  "StatusCode": "00",  
  "StatusDesc": "Completed successfully",  
  "merchant_name": "Faben",  
  "merchant_location": "Kampala",  
  "FlexiPayRefNum": "EC10000343443"  
}
```

2. Merchant Payment – From Customer Mobile Money Wallet

Merchant payment API URL

https://<IP Address>:<port>/merchants/v1.0/paymerchant

Merchant collection/payment Request

```
{  
  "msisdn": "256759889380",  
  "RequestID": "20210706033645",  
  "Merchant_Code": "20400071",  
  "Reference_no": "10000001",  
  "Amount": "1000",  
  "Source": "AIRTEL",  
  "Narrative": "Joseph wedding pledge"  
}
```

Merchant collection/payment Response

```
{  
  "StatusCode": "00",  
  "StatusDesc": "Completed successfully",  
  "TrfRefNum": "18389303058",  
  "FlexiPayRefNum": "EC10000343443"  
}
```

3. Get Merchant Payment Status (Mobile Money Payments)

Request Authentication

Field RequestID is mandatory. It is user to authenticate each, and every request sent and should vary from one request to another.

Each request should have Client_ID and Token headers, which are used for authentication.

Client_ID is a constant parameter which is shared in another platform.

Token is a result of the encoding to base64 mandatory unique RequestID field and Client Password, then hashing it using SHA256. Sample Java Code shared below)

Token= SHA256(base64(mandatory unique RequestID | Client Password))

Sample Code

```
private static void generateTransactionToken(String requestId, String password)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = requestId+"|"+password;

    String base64Data = new String(Base64.getEncoder().encode(data.getBytes(StandardCharsets.UTF_8)));

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    messageDigest.update(base64Data.getBytes(StandardCharsets.UTF_8));

    String TokenHash = toHexString(messageDigest.digest());

    System.out.println("Token hash : "+TokenHash);
}
```

```
private static String toHexString(byte[] digest) {
    BigInteger number = new BigInteger(1, digest);
    StringBuilder hexString = new StringBuilder(number.toString(16));

    while (hexString.length() < 64){
        hexString.insert(0, '0');
    }

    return hexString.toString();
}
```

Get transaction status API URL

Note – This applies to merchant payments

https://<IP_Address>:<port>/flexipayws/v1.0/getpaymentstatus

Get Transaction Status Request

```
{  
  "RequestID": "20210706033645",  
  "Client_ID": "FLEXUAT"  
}
```

Get Transaction Status Response

```
{  
  "StatusMessage": "Completed successfully",  
  "TrfRefNum": "18389303058",  
  "FlexipayRefNum": "EC10000343443",  
  "Status": "00"  
}
```

4. Initiate Approval – From Customer FlexiPay Wallet

Initiate approval API URL

https://<IP Address>:<port>/merchants/v1.0/paymerchant

Initiate approval Request

```
{  
  "msisdn": "256759889380",  
  "RequestID": "20210706033645",  
  "Merchant_Code": "20400071",  
  "Amount": "10000",  
  "Source": "FLEXIPAY",  
  "Narrative": "Groceries"  
}
```

Initiate approval Response

```
{  
  "StatusCode": "00",  
  "StatusDesc": "Completed successfully",  
  "TrfRefNum": "18389303058",  
  "FlexiPayRefNum": "EC10000343443"  
}
```


5. Get Customer Approval Status (FlexiPay Payments)

Request Authentication

Field RequestID is mandatory. It is user to authenticate each, and every request sent and should vary from one request to another.

Each request should have Client_ID and Token headers, which are used for authentication.

Client_ID is a constant parameter which is shared in another platform.

Token is a result of the encoding to base64 mandatory unique RequestID field and Client Password, then hashing it using SHA256. Sample Java Code shared below)

Token= SHA256(base64(mandatory unique RequestID | Client Password))

Sample Code

```
private static void generateTransactionToken(String requestID, String password)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = requestID+"|"+password;

    String base64Data = new String(Base64.getEncoder().encode(data.getBytes(StandardCharsets.UTF_8)));

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    messageDigest.update(base64Data.getBytes(StandardCharsets.UTF_8));

    String TokenHash = toHexString(messageDigest.digest());

    System.out.println("Token hash : "+TokenHash);
}
```

```
private static String toHexString(byte[] digest) {
    BigInteger number = new BigInteger(1, digest);
    StringBuilder hexString = new StringBuilder(number.toString(16));

    while (hexString.length() < 64){
        hexString.insert(0, '0');
    }

    return hexString.toString();
}
```

Get approval status API URL

Note – This applies to customer approvals

https://<IP_Address>:<port>/merchants/v1.0/getapprovalstatus

Get Transaction Status Request

```
{  
  "RequestID": "20210706033645",  
  "Client_ID": "FLEXUAT"  
}
```

Get Transaction Status Response

```
{  
  "StatusDesc": "Completed successfully",  
  "RequestID": "18389303058",  
  "Client_ID": "00",  
  "FlexipayRefNum": "EC10000343443",  
  "StatusCode": "00"  
}
```

Request and Response Message Fields

Merchant Request Messages Fields:

Parameter name	Description	Required	Format
msisdn	MNO mobile number	Mandatory	String
RequestID	Transaction reference	Mandatory	String
Merchant_Code	Merchant unique code	Mandatory	String
Reference_no	For referenced merchants	Optional	String
Amount	Transactional amount	Mandatory	String
Narrative	Transaction narration/reason for payment	Mandatory	String
Source	Source of value. Can be FLEXIPAY, AIRTEL or MTN. If source is FLEXIPAY, msisdn must be the	Mandatory	String

	number registered on FlexiPay wallet		
--	--------------------------------------	--	--

Merchant Response Messages Fields:

Parameter name	Description	Required	Format
StatusDesc	Transaction Status	Mandatory	String
TrfRefNum	Transaction ID from partner	Mandatory	String
FlexipayRefNum	Transaction ID from partner	Mandatory	String
FlexiPayRefNum	FlexiPay reference number	Mandatory	String
StatusCode	Transaction status code	Mandatory	String

Bulk Payments to FlexiPay

Request Authentication

Field RequestID is mandatory. It is user to authenticate each, and every request sent and should vary from one request to another.

Each request should have Client_ID and Token headers, which are used for authentication.

Client_ID is a constant parameter which is shared in another platform.

Token is a result of the encoding to base64 mandatory unique RequestID field and Client Password, then hashing it using SHA256. Sample Java Code shared below)

Token= SHA256(base64(mandatory unique RequestID | Client Password))

Sample Code

```
private static void generateTransactionToken(String requestId, String password)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = requestId+"|"+password;

    String base64Data = new String(Base64.getEncoder().encode(data.getBytes(StandardCharsets.UTF_8)));

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    messageDigest.update(base64Data.getBytes(StandardCharsets.UTF_8));

    String TokenHash = toHexString(messageDigest.digest());

    System.out.println("Token hash : "+TokenHash);
}
```

```
private static String toHexString(byte[] digest) {
    BigInteger number = new BigInteger(1, digest);
    StringBuilder hexString = new StringBuilder(number.toString(16));

    while (hexString.length() < 64){
        hexString.insert(0, '0');
    }

    return hexString.toString();
}
```

API Requests and Responses

1. Bulk Payments

Bulk payments API URL

https://<IP Address>:<port>/ flexipayws /v1.0/payments/api

Bulk payment Request

Note – For this request you will need to include your callback URL. A callback will be sent from FlexiPay to this URL once the transaction has been posted on the specified wallets to notify you of the payment status

```
{
  "RequestID": "20210706033645",
  "Request_Time": "YYYY/MM/DD HH:MM:SS",
  "no_of_records": "3",
  "total_amount": "300000",
  "Client_ID": "FLEXUAT",
  "Narrative": "Some narrative here"s,
  "callback_URL": "Partner callback URL here",
  "customer_data": [
    {
      "mobile_number": "256773234110",
      "first_name": "Bernard",
      "second_name": "",
      "last_name": "Cheress",
      "amount": "100000",
      "payment_narration": "Weekly dividend"
    }, {
      "mobile_number": "256759556327",
      "first_name": "Kipkirui",
      "second_name": "",
      "last_name": "Bett",
      "amount": "100000",
      "payment_narration": "Weekly dividend"
    }, {
      "mobile_number": "256721411331",
      "first_name": "Kakande",
      "second_name": "MW",
      "last_name": "Sabititi",
      "amount": "100000",
      "payment_narration": "Weekly dividend"
    }
  ]
}
```

Bulk payment Response

```
{
  "Status": "00",
  "callback_url": "Http(S)://domain-name:port/Servlet",
  "StatusMessage": "Bulk payment has been received successfully"
}
```

Bulk payment callback to partner

Note – This callback will be sent from FlexiPay to your specified callback URL once the transaction has been posted on the specified wallets, to notify you of the payment status

```
{
  "RequestID": "20210706033645",
  "Request_Time": "YYYY/MM/DD HH:MM:SS",
  "customer_data": [
    {
      "mobile_number": "256773234110",
      "status": "00",
      "amount": "100000",
      "StatusMessage": "Completed successfully"
    }, {
      "mobile_number": "256759556327",
      "status": "57",
      "amount": "100000",
      "StatusMessage": "Provided name mismatch with registered name on Flexipay"
    }, {
      "mobile_number": "256721411331",
      "status": "00",
      "amount": "100000",
      "StatusMessage": "Completed successfully"
    }
  ]
}
```

Bulk payment callback response from partner

```
{
  "Status": "00",
  "StatusMessage": "Accepted successfully"
}
```

Wallet Request Messages Fields:

Parameter name	Description	Required	Format
Request_Time	Transaction date and time in the format YYYY/MM/DD HH:MM:SS	Mandatory	String
RequestID	Transaction reference	Mandatory	String
Client_ID	Client uniquely assigned code	Mandatory	String
Reference_no	For referenced merchants	Mandatory	String
Amount	Transactional amount	Mandatory	String
Narrative	Transaction narration/reason for batch payment	Mandatory	String
customer_data	An array of json objects containing customer data	Mandatory	String
mobile_number	Customer MSISDN (Phone number) starting with 256 with only numbers	Mandatory	String
first_name	Customer first name as registered at MNO	Mandatory	String
second_name	Customer second(middle) name as registered at MNO	Mandatory, empty string when the customer doesn't have	String
last_name	Customer last name as registered at MNO	Mandatory	String
payment_narration	Reason for payment (Individual payment description)	Mandatory	String
amount	Amount to pay individual customer	Mandatory	String/Integer
no_of_records	Total number of records contained under customer data	Mandatory	String/integer
Total_amount	Sum of all amount under customer data	Mandatory	String/Integer

Wallet Response Messages Fields:

Parameter name	Description	Required	Format
StatusMessage	Description of transaction status	Mandatory	String
TrfRefNum	Original trans reference	Mandatory	String
FlexipayRefNum	FlexiPay reference number	Mandatory	String
Status	Transaction status code	Mandatory	String

Funds Transfer

Request Authentication

Field RequestID is mandatory. It is user to authenticate each, and every request sent and should vary from one request to another.

Each request should have Client_ID and Token headers, which are used for authentication.

Client_ID is a constant parameter which is shared in another platform.

Token is a result of the encoding to base64 mandatory unique RequestID field and Client Password, then hashing it using SHA256. Sample Java Code shared below)

Token = SHA256(base64(mandatory unique RequestID | Client Password))

Sample Code

```
private static void generateTransactionToken(String requestId, String password)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = requestId+"|"+password;

    String base64Data = new String(Base64.getEncoder().encode(data.getBytes(StandardCharsets.UTF_8)));

    MessageDigest messageDigest = MessageDigest.getInstance("SHA-256");
    messageDigest.update(base64Data.getBytes(StandardCharsets.UTF_8));

    String TokenHash = toHexString(messageDigest.digest());

    System.out.println("Token hash : "+TokenHash);
}
```

```
private static String toHexString(byte[] digest) {
    BigInteger number = new BigInteger(1, digest);
    StringBuilder hexString = new StringBuilder(number.toString(16));

    while (hexString.length() < 64){
        hexString.insert(0, '0');
    }

    return hexString.toString();
}
```

API Requests and Responses

1. Get Wallet Details

Get wallet details API URL

https://<IP Address>:<port>/flexipayws/v1.0/payments/getwalletdetails

Get wallet details Request

```
{  
  "MSISDN": "256759556327",  
  "REQUEST_ID": "20210706033645",  
  "CLIENT_ID": "FLEXUAT"  
}
```

Get wallet details Response

```
{  
  "STATUSMESSAGE": "Completed successfully",  
  "WALLET_NAME": "BERNARD CHERESS",  
  "CURRENCY": "UGX",  
  "ALLOW_CR": "Y",  
  "ALLOW_DR": "Y",  
  "STATUS": "00"  
}
```

Funds Transfer API URL

https://<IP Address>:<port>/flexipayws/v1.0/transfers/api

2. Transfer to FlexiPay Wallet

Wallet payment Request

```
{  
  "RequestID": "20210706033666",  
  "Request_Time": "YYYY/MM/DD HH:MM:SS",  
  "Client_ID": "TERRAUAT",  
  "Narrative": "Lunch Money",  
  "Recipient_name": "JULIET STONICK",  
  "Recipient_account": "256778006381",  
  "Mobile_number": "256759556327",  
}
```

```

    "Amount": "100000",
    "Network": "WALLET",
    "Destination": "WALLET"
}

```

Wallet payment response

```

{
    "RequestID": "20210706033666",
    "FlexipayID": "",
    "StatusCode": "00",
    "StatusMessage": "Successful"
}

```

3. Transfer to Mobile Money Wallet

MNO Transfer Request

```

{
    "RequestID": "20210706033645",
    "Request_Time": "YYYY/MM/DD HH:MM:SS",
    "Bank_sort_code": "",
    "Bank_name": "",
    "Client_ID": "FLEXUAT",
    "Narrative": "School Fees Deposit",
    "Recipient_name": "BERNARD CHERESS",
    "Mobile_number": "256759556327",
    "Amount": "5000",
    "Network": "AIRTEL",
    "Destination": "MNO"
}

```

MNO Transfer Response

```

{
    "RequestID": "20210706033645",
    "FlexipayRefNum": "EC1000304333",
    "StatusCode": "00",
    "StatusMessage": "Completed successfully"
}

```

4. Transfer to Bank

Bank Transfer Request

```
{
  "RequestID": "TPSC0006547465",
  "Request_Time": "2023/03/29 16:03:03",
  "Client_ID": "FLEXUAT",
  "Bank_sort_code": "SBICUGKXXXX ",
  "Bank_name": "Stanbic Bank Uganda",
  "Originator_Name": " Matthew",
  "Originator_Surname": "Davidson",
  "Originator_Country_of_residence": "Ireland",
  "Originator_Date_of_Birth": "10/04/1968",
  "Originator_Nationality": "Irish",
  "Mobile_number": "256701221117",
  "Narrative": "Guitar Lessons",
  "Amount": "10000",
  "Beneficiary_account": "9030012345678",
  "Beneficiary_Name": "Alan",
  "Beneficiary_Surname": "Mukisa",
  "Beneficiary_Country_of_resdence": "Uganda",
  "Network": "BANK",
  "Currency": "UGX",
  "Destination": "BANK"
}
```

Bank Transfer Response

```
{
  "RequestID": "20210706033645",
  "FlexipayRefNum": "EC1000304333",
  "StatusCode": "00",
  "StatusMessage": "Your request has been received and is being procssed"
}
```

5. Get Transaction Status

Get Transaction Status Request

```
{
  "RequestID": "20210706033645",
  "Client_ID": "FLEXUAT"
}
```

Get Transaction Status Response

```
{
  "StatusMessage": "Completed successfully",
  "TrfRefNum": "18389303058",
  "FlexipayRefNum": "EC10000343443",
  "Status": "00"
}
```

Third Party Collections through FlexiPay

1. Customer Validation

Validation request to partner

```
{
  "request_reference": "E00112545201",
  "customer_number": " AB123456",
  "transtype": "presentment"
}
```

Validation response from partner

```
{
  "status_code": "00",
  "status_desc": "Completed successfully",
  "customer_name": "John Doe",
  "pending_balance": "5500",
  "payment_mod": "PART"
}
```

2. Customer Payment

Payment request to partner

```
{
  "request_reference": "EC10000343443",
  "customer_number": "0700123456",
  "customer_name": "John Doe",
  "transtype": "payment",
  "narrative": "Payment from customer AB123",
  "reason_for_payment": "loan_repayment",
  "location": "Kampala",
  "amount": "182000"
}
```

Payment response from partner

```
{
  "status_code": "00",
  "status_desc": "Completed successfully",s
  "external_reference": "18389303058",
  "flexipay_ref_num": "EC10000343443"
}
```

Bill Payments

Request Authentication

Field **requestReference** is mandatory. It is used to authenticate each request sent and should vary from one request to another.

Each request should have **ClientId**, **password** and **token** headers, which are used for authentication.

Token is a result of encoding to base64 the fields (ServiceID+ClientId+requestReference+customerId+Amount), then hashing it using HmacSHA256

```
Token=base64(HmacSHA256(ServiceID+ClientId+requestReference+customerId+Amount))
```

Sample Code

```
private static void generateMerchantPaymentToken(String clientID, String requestID, String msisdn, String amount)
    throws NoSuchAlgorithmException, InvalidKeyException {
    String data = clientID+requestID+msisdn+amount;

    String secretKey = "Passsword";

    Mac sha256Mac = Mac.getInstance("HmacSHA256");
    SecretKeySpec secretKeySpec = new SecretKeySpec(secretKey.getBytes(), "HmacSHA256");

    sha256Mac.init(secretKeySpec);

    String token = new String(Base64.getEncoder().encode(sha256Mac.doFinal(data.getBytes())));
}
```

API Requests and Responses

1. Customer validation at biller

Biller payment validation API URL:

https://<IP Address>:<port>/FLEXI_BILLERS_API/Validation

Customer validation request to FlexiPay

```
{
  "ServiceID": "1001",
  "ClientId": "FLEXUAT",
  "requestReference": "000112545201",
  "customerId": "042578934007",
  "Transtype": "presentment",
  "customerMobile": "256700123456",
  "Location": "Kampala"
}
```

Customer validation response from FlexiPay

```
{
  "StatusCode": "00",
  "StatusDesc": "Completed successfully",
  "customer_name": "John Mark",
  "Amount": "32000",
  "FlexipayRefNum": "EC10000343443",
  "processid": "Flexi3434585455650022",
  "defaultPackage": "GOTVSUPA",
  "uedlid": "1202106221106065757",
  "TVcustomernumber": "259883734",
  "schoolCode": "100"
}
```

2. Bill Payment

Biller payment API URL:

https://<IP Address>:<port>/FLEXI_BILLERS_API/POST_PAYMENTS

Bill payment request to FlexiPay

```
{
  "ServiceID": "1001",
  "ClientId": "FLEXUAT",
  "requestReference": "000112545201",
  "Narrative": "Bill payment for DSTV",
  "customerId": "11111",
  "customerName": "Amos Okello",
  "Tin": "58955",
  "uedlid": "1202106221106065757",
  "processid": "Flexi3434585455650022",
  "ProductCode": "GOTVPLS",
  "Location": "Kampala",
  "customerMobile": "254711940525",
  "Amount": "200",
  "schoolCode": "100"
}
```

Bill payment response from FlexiPay

```
{
  "StatusCode": "00",
  "StatusDesc": "Completed successfully",
  "requestReference": "000112545290",
  "uedlid": "1202106221106065757",
  "processid": "Flexi3434585455650022",
  "electricityToken": "62253816533927452555",
  "electricityUnits": "16.5",
  "FlexipayRefNum": "EC10000343487",
  "receiptNumber": "0000000009083138"
}
```

UEDCL amount validation request to Flexipay

```
{
  "ServiceID": "1008",
  "ClientId": "FLEXUAT",
  "customerId": "11111",
  "Transtype": "presentment",
  "Operation": "amountvalidation",
  "requestReference": "18389303058",
  "processid": "Flexi3434585455650022",
  "uedlid": "1202106221106065757"
}
```


UEDCL amount validation response from Flexipay

```
{
  "StatusCode": "00",
  "StatusDesc": "Completed successfully",
  "customer_name": "Test Test",
  "Amount": "5500",
  "FlexipayRefNum": "EC10000343443"
  "uedlid": "1202106221106065757"
}
```

Fetch TV packages request to Flexipay

```
{
  "ServiceID": "1005",
  "ClientId": "FLEXUAT",
  "customerId": "11111",
  "Transtype": "presentment",
  "Operation": "fetchtvPackages",
  "requestReference": "18389303058"
}
```

Fetch TV packages response from Flexipay

```
{
  "StatusCode": "00",
  "StatusDesc": "Completed successfully",
  "customer_name": "Test Test",
  "Amount": "5500",
  "FlexipayRefNum": "EC10000343443"
  "Tvpackages": "GOTV^GOTV VALUE BOUQUET|16000~GOTvPLS^GOTv Plus
Bouquet|26000~GOLITE^GOTV LITE BOUQUET|65000~GOTVMAX^GOTVMAX - Gotv Max
Bouquet|49000~GOTVSUPA^GOTv Supa Bouquet|55000~"
}
```

3. Check aggregator wallet balance

Check aggregator balance API URL:

https://<IP_Address>:<port>/FLEXI_BILLERS_API/BALANCE_CHECK

Balance Check Request

```
{
  "ClientId": "FLEXUAT",
  "requestReference": "18389303058"
}
```

Balance Check Response

```
{
  "StatusCode": "00",
  "StatusDesc": "Completed successfully",
  "AvailableBal": "550000",
  "ActualBal": "550000",
  "FlexipayRefNum": "3000000001236"
}
```

Biller Service IDs

Service ID	Biller	Flow
1001	UMEMEYAKA	This service has 2 set of operations: 1. Presentment/Validation 2. Payment
1002	UMEMEPOST_PAID	This service has 2 set of operations: 1. Presentment/Validation 2. Payment
1003	NWSC	This service has 2 set of operations: 1. Presentment/Validation 2. Payment
1005	DSTV	This service has 3 set of operations: 1. Presentment/Validation 2. Fetch TV packages 3. Payment
1006	GOTV	This service has 3 set of operations: 1. Presentment/Validation 2. Fetch TV packages 3. Payment
1007	SCHOOLPAY	This service has 2 set of operations: 1. Presentment/Validation 2. Payment

1008	UEDCL	This service has 3 set of operations: 1. Presentment/Validation 2. Amount Validation 3. Payment
1009	NSSF	This service has 2 set of operations: 1. Presentment/Validation 2.Payment

Request and Response Message Fields

Header Message Fields

Parameter name	Description	Required	Format
ClientId	Id of Client making the request	M	String
passKey	Shared password Key to be used to encode	M	String
Token	Generated by concatenating the mandatory fields then hashing using HmacSHA256 and encoding using base64.	M/O - (Optional for balance check)	String

Bill payment Request Messages Fields:

Parameter name	Description	Required	Format
ServiceID	Code that identifies the payment item being Paid for.e.g. 1001 for Umemeyaka Bill presentment.	Mandatory	String
ClientId	Id of Client making the request	Mandatory	String
requestReference	The request reference that uniquely Identifies the transaction.	Mandatory	String
Narrative	Transaction narration	Optional	String

Reason for payment	Caters to the different deposit categories a partner may have on their system e.g savings deposit vs loan repayment	Optional	String
Amount	Transactional amount	Mandatory	String
customerId	The customer unique identifier being Validated or paid for e.g. meter number, Account No, PRN, TRN, SmartCardNumber, phone number in the case of airtime top-up etc.	Mandatory	String
customerName	The name of the UMEME customer making the payment N.B – Only for UMEME payments	Optional	String
Tin	Taxpayer Identification Number. This is returned upon doing presentment (Validations) and it's mandatory for URA. NB: Only for URA requests	Optional	String
Location	For NWSC payments indication meter number location e.g. Kampala, Jinja, Entebe etc.	Optional	String
processid	Generated and returned during UEDCL presentment. It should be supplied for both amount validation and payment requests NB: Only for UEDCL requests	Optional	String
uedlid	Returned by UEDCL on both operations for presentment and amount validation. NB: Only for UEDCL	Optional	String

requests

customerMobile	Customer's phone number	Mandatory	String
ProductCode	To be supplied while making payment for DSTV and GOTV only. Returned upon fetching TV packages	Optional	String
Amount	Bill Amount to be paid	Mandatory	String
Operation	Specifies action to be taken e.g fetchtvPackages for DSTV/GOTV, amount validation for UEDCL e.t.c	Optional	String
TVcustomernumber	Returned on GoTV and DSTV Validation response and is to be supplied while fetching TV packages for DSTV or GoTV.	Optional	String
schoolCode	Returned on SchoolPay Validation response and is to be supplied for SchoolPay payments	Optional	String

Bill Payment Response Messages Fields

Parameter name	Description	Required	Format
StatusDesc	Transaction Status	Mandatory	String
requestReference	Original trans reference	Mandatory	String
FlexipayRefNum	Felxipay reference number	Mandatory	String
StatusCode	Transaction status code	Mandatory	String
Amount	Bill amount to be paid.	Optional	String
customer_name	Customer name tied to the customer id being verified	Mandatory	String
pending_balance	Amount due for the customer (0 if missing)	Mandatory	String

Payment_mod	Indicator if the customer can pay any amount (PART), exact amount (EXACT) or anything equal or higher than pending amount (FULL)	Mandatory	String
externalreference	Transaction ID for the consuming third party system	Mandatory	String
Tin	Taxpayer Identification Number. This is returned upon doing presentment (Validations) and it's mandatory for URA only.	Optional	String
processid	Generated and returned during UEDCL presentment. It should be supplied for both amount validation and payment requests	Optional	String
uedlid	Returned by UEDCL on both operations for presentment and amount validation.	Optional	String
defaultPackage	Default customer TV package, Returned by multichoice(DSTV/GOTV) NB: Only for DSTV and GOTV requests	Optional	String
TVcustomernumber	Multichoice customer number. Returned by multichoice(DSTV/GOTV) NB: Only for DSTV and GOTV requests	Optional	String
Tvpackages	All Packages returned under a specific provider, in this case DSTV and GOTV NB: productCode and packagename are separated with symbol ^ then packagename and packagePrice are separated with a pipe() Example: (productCode ^ packagename packagePrice)	Optional	String
AvailableBal	Amount of money in the account that is available and can be utilized.	Optional	String
ActualBal	Amount of money that is in the account at any given time.	Optional	String

Response Codes

Response Code	Description
00	Success
01	Pending
57	Failed
51	Insufficient Funds

58	Invalid Client ID or Authentication failed
59	Invalid or empty payload
94	Duplicate transaction (RequestID)
99	Read timed out
37	Missing Transaction Reference
04	Invalid or amount not supplied
02	Missing customer phone number
16	Single transaction limit exceeded
19	Daily transaction account limit exceeded
20	Daily account balance limit exceeded
17	Amount below set limit
102	Debit account is missing