



TELKOM API SPECIFICATION

Version 1.4.2

29th September 2020

This document contains confidential information, which is the property of Telkom Kenya Limited.
No part of its contents may be used, copied, disclosed or conveyed to any party in any manner
whatsoever without prior written permission from Telkom Kenya.

Document Versioning

Version	Author	Date
1.0.0	Benson O. Oluga/ Faith N. Wanjala	11 th Sep 2018
1.1.0	Benson O. Oluga/ Faith N. Wanjala	5 th Oct 2018
1.2.0	Benson O. Oluga/ Faith N. Wanjala	22 nd Oct 2018
1.3.0	Benson O. Oluga/ Faith N. Wanjala	31 st Oct 2018
1.4.0	Benson O. Oluga/ Faith N. Wanjala	3 rd Dec 2018
1.4.1	Benson O. Oluga/ Faith N. Wanjala	15 th March 2019

Version	Reviewers	Date
1.0.0	Jimmy K. Bett	13 th September, 2018
1.1.0		
1.4.1	Jimmy K. Bett	18 th March 2019
1.4.2	Edwin Mwendu/ Damaris Musinguzi	29 th September, 2020

Change Log

Version	Date	Change Log	Revision made by
1.0.0	11/Sep/2018	Initial draft	Benson O. Oluga/ Faith N. Wanjala
1.1.0	05/Oct/2018	Addition of: <ul style="list-style-type: none"> - Simulation of C2B notification API - Reversal API - T-kash Code Integration APIs - Update URL - Text formatting and general layout 	Benson O. Oluga/ Faith N. Wanjala

1.2.0	22/Oct/2018	<div>Addition of:<ul style="list-style-type: none">- Additional formatting- API Versioning- Change of all APIs to version 3 (v3)</div>	Benson O. Oluga/ Faith N. Wanjala
-------	-------------	--	-----------------------------------

		<ul style="list-style-type: none"> - Replay Notification - E-jaze synchronous ATP API - Bill validation - Service registration - Send SMS - SMS Delivery status - USSD 	
1.3.0	31/Oct/2018	Addition of: <ul style="list-style-type: none"> - Application password encryption 	Benson O. Oluga/ Faith N. Wanjala
1.4.1	19/Mar/2019	<ul style="list-style-type: none"> - Ejaze API - SMS APIs 	Benson O. Oluga/ Faith N. Wanjala
1.4.2	29/Sept/2020	<ul style="list-style-type: none"> - Till Notification API - C2B Notification API (updated type formatting) 	Tony O. Blair

Glossary

ACRONYM	MEANING
API	Application Programming Interface
ATP	Airtime Purchase
B2C	Business to Customer
C2B	Customer to Business
CC	Country Code (254 for Kenya)
CR	Credit
DR	Debit
HTTP	Hyper Text Transfer Protocol
ISW	Interswitch
JSON	JavaScript Object Notation
L2	Layer 2 Business Accounts
L7	Layer 7 Subscriber Accounts
MFS	Mobile Financial Service(s)
MSISDN	Mobile Subscriber ISDN
NDC	National Destination Code e.g. 077
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
TCS	Telepin Communications System
TKL	Telkom Kenya Limited
URL	Uniform Resource Locator

API Versioning and URL Format

The API version will be in the URL.

The API URLs will take the form:

[{scheme}://{platform}.gw.mfs-tkl.com/tkash/{api_name}/{api_version}/{api_context}](#)

For example, asynchronous airtime purchase API on pre-production platform will be:

`https://preprod.gw.mfs-tkl.com/tkash/airtimerequest/v3/atpAsync`

ACRONYM	MEANING
{scheme}	https
{platform}	sandbox/preprod/uat/dev/"prod"
{api_name}	Specific API being called
{api_version}	v3 (current version as at the time of this document). This is the only component that will change in the URL for any API
{api_context}	Names the specific API with its type i.e. synchronous or asynchronous

Table of Contents

Document Versioning.....	1
Change Log.....	1
Glossary.....	3
API Versioning and URL Format	4
Table of Contents	5
1 Authentication	9
1.1 API Gateway authentication	9
1.2 Application User authentication	9
1.2.1 Sample Java Encryption.....	9
1.2.2 Sample PHP Encryption.....	10
2 URL Registration and Update	14
2.1 Register URL	15
2.2 Register URL Request Object Parameter Definition.....	15
2.3 Update URL	15
2.4 Update URL Request Object Parameter Definition	16
2.5 Service Registration.....	16
2.6 Register URL Request Object Parameter Definition.....	17
3 Airtime Purchase (ATP) API	18
3.1 T-kash Asynchronous ATP	18
3.1.1 Overview	18
3.1.2 Request Object Parameter Definition	19
3.1.3 Response Object Parameter Definition.....	19
3.2 T-kash Synchronous ATP	19
3.2.1 Overview	19
3.2.2 Request Object Parameter Definition	20
3.2.3 Response Object Parameter Definition.....	20
3.3 E-jaze Synchronous ATP	20
3.3.1 Overview	20
3.4 Request Object Parameter Definition	21
3.4.1 Response Object Parameter Definition.....	21

3.5	Ejaze Balance Check	22
3.5.1	Overview	22
3.6	Request Object Parameter Definition	23
3.6.1	Response Object Parameter Definition	23
3.7	Ejaze PIN Change	23
3.7.1	Overview	23
3.8	Request Object Parameter Definition	24
3.8.1	Response Object Parameter Definition	24
4	Transaction Status	25
4.1	Overview	25
4.2	Transaction Status Request Object Parameter Definition	26
4.3	Transaction Status Response Object Parameter Definition	26
5	Replay Notification	26
5.1	Overview	26
5.2	Request Object Parameter Definition	27
5.3	Response Object Parameter Definition	27
6	Business to Customer Transactions	27
6.1	B2C Name Lookup (NLU) API	28
6.1.1	Request Object Parameter Definition	29
6.1.2	Response Object Parameter Definition	29
6.2	Synchronous B2C Payment API	29
6.2.1	Synchronous B2C Payment Request Object Parameter Definition	30
6.2.2	Synchronous B2C Payment Response Object Parameter Definition	30
6.3	Asynchronous B2C Payment API	31
6.3.1	Asynchronous B2C Payment Request Object Parameter Definition	32
6.3.2	Asynchronous B2C Payment Response Object Parameter Definition	32
7	Transaction Notification	33
7.1.1	Simulation API	33
7.1.2	Transaction Notification (SOAP Request)	34
7.1.3	Transaction Notification (SOAP Response)	34
7.1.4	Transaction Notification (REST Request)	35
7.1.5	Transaction Notification (REST Response)	35

7.1.6	Request Object Parameter Definition	35
8	Customer to Business Transactions	36
8.1	C2B1 – Customer Initiated Without Account Validation	37
8.2	C2B2 – Customer Initiated With Account Validation	37
8.2.1	Bill Validation	37
8.2.2	Request Object Parameter Definition	38
8.2.3	Response Object Parameter Definition	38
9	Reversal APIs	39
9.1	Reversal Request	39
9.1.1	Overview	39
9.1.2	Reversal Request Object Parameter Definition	40
9.1.3	Reversal Response Object Parameter Definition – Scenario 1	40
9.1.4	Reversal Response Object Parameter Definition – Scenario 2	41
9.2	Reversal Approval	41
9.2.1	Overview	41
9.2.2	Approval Request Object Parameter Definition	42
9.2.3	Approval Response Object Parameter Definition	42
10	Transfer to Bank APIs	42
10.1	Get List of Pre-registered Bank Accounts	42
10.1.1	Overview	42
10.1.2	List Bank Accounts Request Object Parameter Definition	43
10.1.3	List Bank Accounts Response Object Parameter Definition – Scenario 1	43
10.1.4	Overview	44
10.1.5	List Bank Accounts Request Object Parameter Definition	45
10.1.6	List Bank Accounts Response Object Parameter Definition – Scenario 1	45
11	T-kash Code APIs	46
11.1	Submit T-kash Code	46
11.1.1	Overview	46
11.1.2	Submit T-kash Code Request Object Parameter Definition	47
11.2	T-kash Code Status Inquiry	48
11.2.1	Overview	48
11.2.2	T-kash Code Status Inquiry Request Object Parameter Definition	49

11.2.3	T-kash Code Status Inquiry Response Code Definition	49
12	USSD & Bulk SMS.....	50
12.1.1	On Demand Request Object Parameter Definition.....	50
12.1.2	Subscription Request Object Parameter Definition	50
12.2	Bulk SMS API	51
12.2.1	Overview	51
12.2.2	SMS Delivery Status Request Object Parameter Definition	52
12.2.3	SMS Delivery Status Response Object Parameter Definition.....	52
12.3	SMS Delivery Notification API	52
12.3.1	Overview	52
12.3.2	SMS Delivery Status Request Object Parameter Definition	54
12.3.3	SMS Delivery Status Response Object Parameter Definition.....	54
12.3.4	Delivery Status Definition.....	55
12.4	USSD.....	55
12.4.1	Overview	55
13	Till Notification API	58
13.1.1	Till Notification (REST Request)	58
13.1.2	Till Notification (REST Response).....	59
13.1.3	Request Object Parameter Definition	59
14	ERROR CODES	60

1 Authentication

There are two levels of authentication.

1.1 API Gateway authentication

API Gateway authentication requires the API GW username and password in addition to authentication tokens. The username/password combination is to be used to login to the API store, and also to generate access tokens. At signup on the API store, two keys will be generated and assigned as below:

- Consumer key e.g. 57GyMT7YdM51EV63f62to3WYs94b
- Consumer Secrete e.g. 5fXpOUkNyr6nJAHVPixw_Rl7CrUb

To generate access tokens, you'll have to pass basic authorization header whose value is base64encode of the two keys above separated by a colon with no spaces between e.g. Consumer key:Consumer Secrete.

The following cURL example command shows how to generate an access token from the above keys using the Password Grant type.

```
curl -k -d "grant_type=password&username=Username&password=Password" -H "Authorization: Basic NTdHeU1UN1lkTTUxRVY2M2Y2MnRvMldZczk0Yjo1ZlhwTlVrTnlyNm5KQUhWUGl4d19SMTdDclVi" https://preprod.gw.mfs-tkl.com/token
```

In a similar manner, you can generate an access token using the client credentials grant type with the following cURL command.

```
curl -k -d "grant_type=client_credentials" -H "Authorization: Basic NTdHeU1UN1lkTTUxRVY2M2Y2MnRvMldZczk0Yjo1ZlhwTlVrTnlyNm5KQUhWUGl4d19SMTdDclVi" https://preprod.gw.mfs-tkl.com/token
```

By default, the access token is valid for 1 hour (3600 second)

1.2 ApplicationUser authentication

Application username and password will also be provided in addition to the API username and password. All requests initiated by the 3rd party and are transactional (e.g. disbursement, reversal, airtime purchase) will require application authentication in addition to API GW authentication (token). Non-transactional (e.g. notification replay, transaction status check, etc) will only require API GW authentication.

Sample Java Encryption

The password will need to be RSA encrypted with a public certificate. Sample encryption java code that will take the certificate is given below:

```
import java.io.File;
import java.io.FileInputStream;
import java.security.cert.X509Certificate;
import java.security.cert.CertificateFactory;
import javax.crypto.Cipher;
import java.security.PublicKey;

/*
 * filename: F:\password_ecryption_files\RSAEncryptPassword.java
 * to run this code, you'll need BouncyCastle JAR
 * download link: https://www.bouncycastle.org/download/bcprov-jdk15on-160.jar
 *
 * compile
```

```

* javac -cp "F:\password_ecryption_files\bcprov-jdk15on-160.jar" RSAEncryptPassword.java
*
* execute
* java -cp ".;F:\password_ecryption_files\bcprov-jdk15on-160.jar" RSAEncryptPassword
*
* sample output (should be in one line)
* encrypted: Jw5XVyGZbHaDpzgK/fydIcsjbUvEYwk08o1uJGtLmotYdhzVDL9q+ikbdOJRrFru60gp7S \
* S032LjvBruJwhwtFzQjb2oeM8WkCoRot/0IGA4T5MrNI8CQiWZ+IP9lGy6crPlDqw0ntcsXNxiVUDBRO \
* a2yHhoDRrsE3s7EO0K755Xb121K7XMkFnbU9qVCKl7NDiutsuJ48k9YFU2NepOsi6Gtw/01kwEo4anyD \
* 7RQqNTtES2hmIDMbNxX5i/VjJfT5lKp4ahziB/cUaY4F6ngRuQao0/Q4NzoLjA6/w8c1Xc2Zm8zOG7Iy \
* YEXECXIYCNZNoF5Q4T/G0WA5LswrlEsw==
*/
public class RSAEncryptPassword{
    public static void main(String[] args) throws Exception{
        String password = "2018";
        byte[] input = password.getBytes();
        FileInputStream inStream = new FileInputStream(new
File("F:\\password_ecryption_files\\telepinuatrsa.cer"));
        CertificateFactory cf = CertificateFactory.getInstance("X.509");
        X509Certificate cert = (X509Certificate)cf.generateCertificate(inStream);
        inStream.close();
        PublicKey publickey = (PublicKey)cert.getPublicKey();
        Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.ENCRYPT_MODE, publickey);
        byte[] encryptedBytes = cipher.doFinal(input);
        String encryptedString = new
String(org.bouncycastle.util.encoders.Base64.encode(encryptedBytes)).trim();
        System.out.println("encrypted: " + encryptedString);
    }
}

```

Sample PHP Encryption

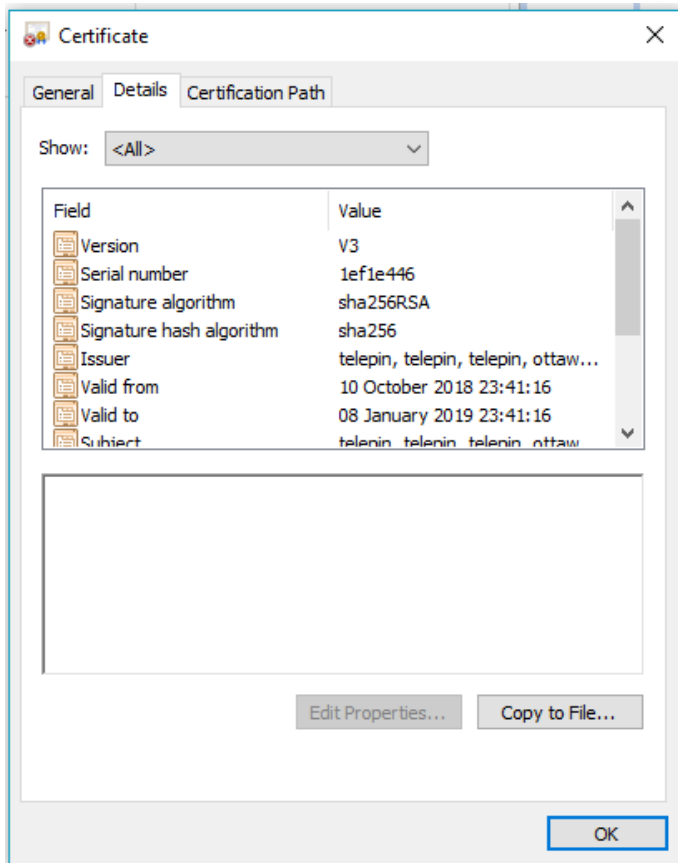
Convert the certificate from DER to PEM format. This can be done using either CLI or GUI.

Using CLI, execute below command while in the directory where `telepinuatrsa.cer` is located (you can define full path).

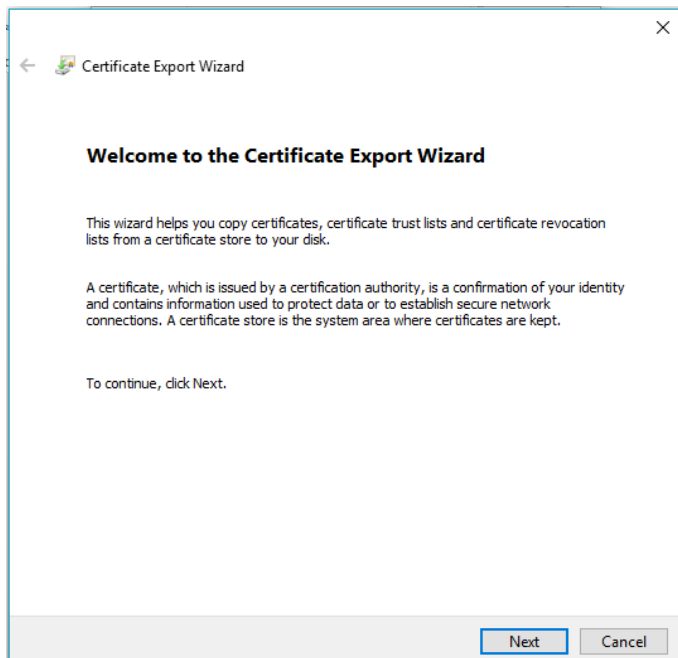
```
openssl x509 -inform DER -in telepinuatrsa.cer > php_telepinuatrsa.cer
```

Using GUI, follow below steps:

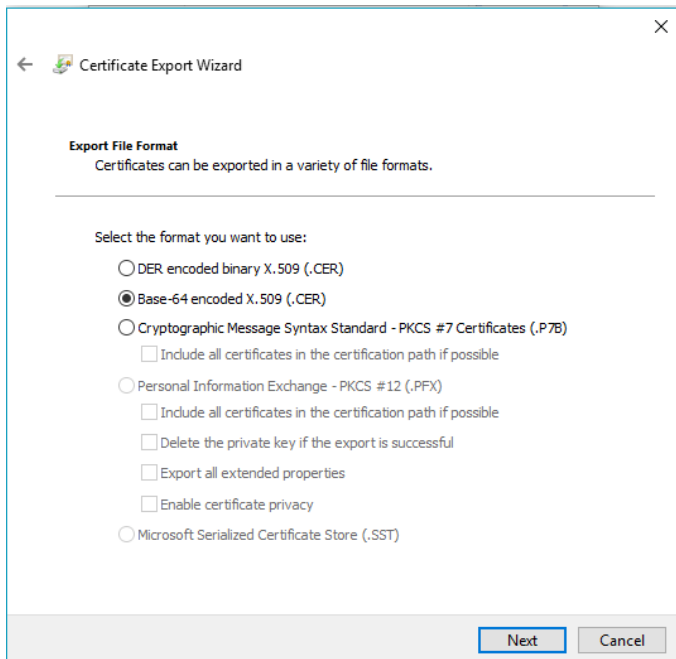
1. Double click on the certificate -> click on "Details" tab, then "Copy To File..." button



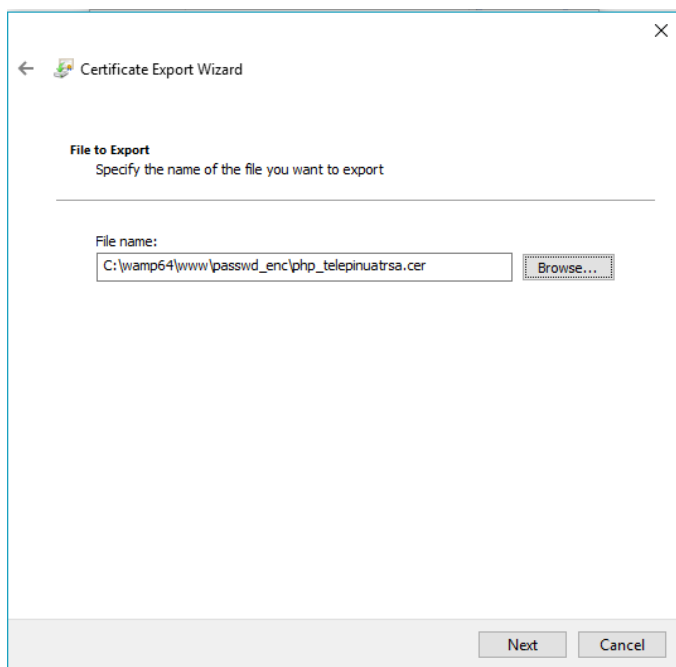
2. Click "Next" button:



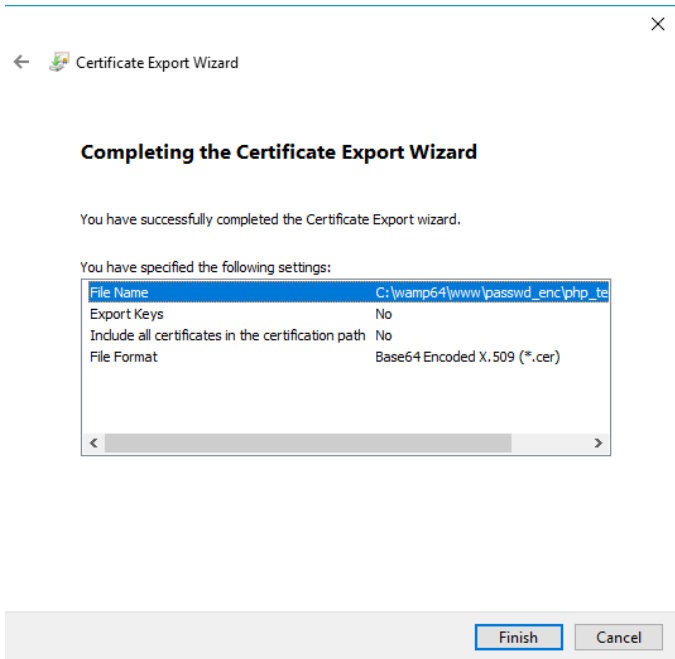
3. Select "Base-64 encoded X.509 (.CER)", then click "Next"



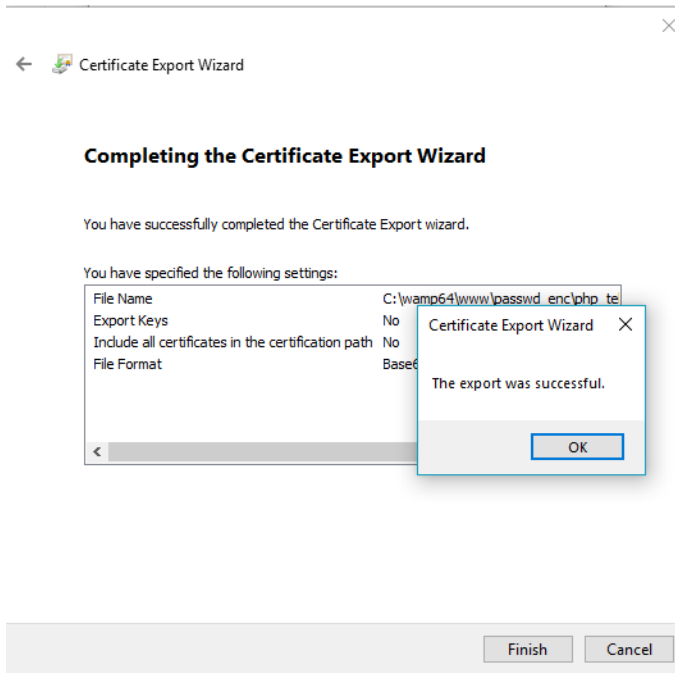
4. Enter new file name and click "Next":



5. Click finish:



6. Click "OK"



Below code can be used to do the encryption:

```
<?php
// RSAEncryptPassword.php
// only works with X509 PEM certs, any DER certs must first be converted
class RSAEncryptPassword {
    public function construct($cert_path, $app_password) {
        // dumb constructor...
        $this->cert_path = $cert_path;
    }
}
```

```

        $this->app_password = $app_password;
    }

    function EncryptPassword() {

        if (!file_exists($this->cert_path)) {
            return("<p>Certificate does not exist in <em>".$this->cert_path."</em>");
        }

        $file_size = filesize($this->cert_path);

        $cert_path_handle = fopen($this->cert_path, "r");
        $cert_data = fread($cert_path_handle, $file_size);
        fclose($cert_path_handle);

        $cert_data = openssl_x509_read($cert_data);

        $public_key = openssl_get_publickey($cert_data);

        openssl_public_encrypt($this->app_password, $encrypted_text, $public_key);

        return(base64_encode($encrypted_text));
        //echo 'php encrypted ' . base64_encode($encrypted_text);
    }
}

/*
 * sample output of an encrypted password, in one line
 *
 * BDl6WkGmpuGZeHBzLcXzcE6Eup8RH6ABOIhUag9EWXI2kffcTd46L3jHfPjL5Lys
 *
 * K06qKjF/GRC4kHw7Zv4D8yrIa75y+vGGfhZFqnELux79RRmPv+BN/193BffphlIc
 * jACEaIqrMLRBfEPFgquwfQ4jilRmlgtB4gurH6905YcR9TN+rI531VqH8rVvfi/Z
 * D+YkSlP4HTixL6d3kHLgff/P8PyEGmSnDh/zkhv3lk/fhRWq2sAgNwYqRtMX9OIF
 * GVEwHyt0zxI9fxXcQUpTOw==
 */

// assumes certificate is in the same directory as this file
$cert_path = dirname(__FILE__) . "\\php_telepinuatrsa.cer";
$app_password = 'siriyangu!';
$pass_enc = new RSAEncryptPassword($cert_path, $app_password);
$encrypted_password = $pass_enc->EncryptPassword();
echo $encrypted_password;
?>

```

2 URL Registration and Update

3rd party application will be required to register different types URL with the API gateway, depending on the services to be consumed.

- The Notification URL (for receiving C2B transactions)
- Account validation URL (for validation of account number entered by subscriber during bill payment, used for [C2B-2](#))
- USSD or ONDEMAND or SUBSCRIPTION URL

2.1 Register URL

API Name	ConsumerAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Consumer
{api_version}	v3
{api_context}	Registerurl
API Location	https://preprod.gw.mfs-tkl.com/consumer/v3/registerurl
Request Method	POST
Payload	<pre>{ "registerUrlRequest": { "consumerId": "string", "notificationUrl": "string", "notificationUrlType": "string", "validationUrl": "string", "validationUrlType": "string", "creationDate": "string" } }</pre>
Content Type	application/json
Responses (Note there's no actual data besides the HTTP codes)	201 – Created
	400 – Bad Request
	409 – Conflict
	500 – Internal Server Error

2.2 Register URL Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
consumerId	String	Biller ID/Paybill number
notificationUrl	String	URL for notification
notificationUrlType	String	REST or SOAP
validationUrl	String	URL for account validation
validationUrlType	String	REST or SOAP
creationDate	String	Date Time Stamp (Format is 13-JUN-2018T12:15:00)

2.3 Update URL

Used to update an existing URL that was previously registered with "[Register URL](#)".

API Name	UpdateConsumerAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	update-consumer
{api_version}	v3
{api_context}	updateUrl
API Location	https://preprod.gw.mfs-tkl.com/update-consumer/v3/updateUrl
Request Method	PUT
Payload	<pre>{ "updateUrlRequest": { "consumerId": "string", "notificationUrl": "string", "notificationUrlType": "string", "validationUrl": "string", "validationUrlType": "string", "creationDate": "string" } }</pre>
Content Type	application/json
Responses	200 – Updated <pre>{ "updatedUrlResponse": { "consumerId": "consumerID passed", "notificationUrl": "notificationUrl", "notificationUrlType": "notificationUrl type regsitered", "validationUrl": "validation url passed", "validationUrlType": "validationUrl type passed", "creationDate": "creationDate" } }</pre>
	400 – Bad Request
	409 – Conflict
	500 – Internal Server Error

2.4 Update URL Request Object Parameter Definition

Similar to [“Register URL”](#) object parameters

2.5 Service Registration

This API shall perform registration 3rd party URL for USSD or sms services which are ONDEMAND, SUBSCRIPTION or BULK.

If type is ONDEMAND or SUBSCRIPTION, the parameter **serviceId** is required. If type is 'USSD' the parameter **ussdLevel** is required. If type is 'ONDEMAND' or 'SUBSCRIPTION' 'serviceId' is required.

API Name	ServiceRegistrationAPI - v3
-----------------	-----------------------------

{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	serviceRegistration
{api_version}	v3
{api_context}	Register
API Location	https://preprod.gw.mfs-tkl.com/serviceRegistration/v3/register
Request Method	POST
Payload	<pre>{ "registrationRequest": { "shortCode": "string", "url": "string", "type": "USSD", "serviceId": "string", "ussdLevel": "Basic", "email": "string" } }</pre>
Content Type	application/json
Responses (Note there's no actual data besides the HTTP codes)	200 – OK
	<pre>{ "result": "string", "referenceId": "string" }</pre>
	400 – Bad Request
	409 – Conflict
	500 – Internal Server Error

2.6 Register URL Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
registrationRequest	Object	
shortCode	String	Content provider shortcode
url	String	Uniform resource locator
type	String	
serviceId	String	
ussdLevel	String	
email	String	

3 Airtime Purchase (ATP) API

Used for airtime purchase and has both asynchronous and synchronous versions. There two platforms for store of value are T-kash and E-jaze

3.1 T-kash Asynchronous ATP

3.1.1 Overview

This API is used to perform an Airtime Purchase by the API consumer and receives an acknowledgement from the API gateway with a `referenceCode` which is used to query back the status of the transaction using the [Transaction Status](#) API.

Note that this is not a call-back, but rather the API consumer polling for transaction status

API Name	AirtimePurchaseAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Airtimerequest
{api_version}	v3
{api_context}	atpAsync
API Location	https://preprod.gw.mfs-tkl.com/tkash/airtimerequest/v3/atpAsync
Request Method	POST
Payload	<pre>{ "atpRequest": { "username": "string", "password": "string", "msisdn": "string", "amount": 0, "brandId": "string" } }</pre>
Content Type	application/json
Responses	202 – Accepted
	<pre>{ "atpAsyncResponse": { "referenceCode": "e11893c5-b033-4e99-9473-a43d66b65fbb", "transactionType": "ATP" } }</pre>
	400 – Bad Request
	500 – Internal Server Error

3.1.2 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
username	String	API application username
password	String	Password of the API application user
msisdn	String	The mobile number for which you are purchasing airtime. The number format can include Country Code without + (254) or National Destination Code (077).
amount	Number	Transaction amount for the initiated transaction KShs. 10 – 10,000
brandId	Number	Internal identifier for type of the transaction, to be shared at integration time. For testing use 933

3.1.3 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
referenceCode	String	Reference code of the of the accepted transaction example: e11893c5-b033-4e99-9473-a43d66b65fbb
transactionType	String	Type of the transaction performed example: ATP

3.2 T-kash Synchronous ATP

3.2.1 Overview

This API is used to perform an Airtime Purchase by the API consumer in a synchronous manner, and waits for the transformed JSON response received from T-kash.

API Name	AirtimePurchaseAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Airtimerequest
{api_version}	v3
{api_context}	Atp
API Location (Pre-prod)	https://preprod.gw.mfs-tkl.com/tkash/airtimerequest/v3/atp
Request Method	POST
Payload	<pre>{ "atpRequest": { "username": "string", "password": "string", "msisdn": "string", "amount": 0, "brandId": "string" } }</pre>

	} }
Content Type	application/json
Responses	200 – OK <pre>{ "atpSyncResponse": { "resultCode": "string", "message": "string", "tklReference": "string", "originalAmount": 0, "finalAmount": 0, "activationDate": 0, "salesOrderNo": "string", "newBalance": 0, "paymentTotal": "string", "trnxFee": 0, "destinationMSISDN": "string" } }</pre>
	400 – Bad Request
	500 – Internal Server Error

3.2.2 Request Object Parameter Definition

The request object parameter definition is similar to those of asynchronous call defined in section [3.1.2 Request Object Parameter Definition](#)

3.2.3 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
resultCode	String	Result code of the Response
message	String	Resulted message of the transaction
tklReference	String	TKL reference number
originalAmount	Number	Original transaction amount
finalAmount	Number	Final transaction amount with taxes (if any)
activationDate		Transaction date
salesOrderNo	Number	Transaction ID
newBalance	Number	New balance of the initiators wallet after the transaction
paymentTotal	Number	Final transaction amount with taxes and other charges (if any)
trnxFee	Number	Transaction fee
destinationMSISDN	String	The MSISDN for which airtime was purchased

3.3 E-jaze Synchronous ATP

3.3.1 Overview

This API is used to perform an Airtime Purchase by the API consumer in a synchronous manner, and waits for a response received from E-jaze.

API Name	EjazeAPI- v1
-----------------	--------------

{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Ejaze
{api_version}	v3
{api_context}	ejaze-atp
API Location	https://preprod.gw.mfs-tkl.com/ejaze/v1/ejazeAtp
Request Method	POST
Payload	<pre>{ "airtimeRequest": { "loginId": "string", "password": "string", "pin": "string", "code": "string", "sourceMsisdn": "string", "destMsisdn": "string", "amount": "string", "extrefnum": "string" } }</pre>
Content Type	application/xml
Responses	200 – OK <pre>{ "airtimeResponse": { "date": "string", "txnStatus": "string", "extrefnum": "string", "message": "string", "transactionId": "string" } }</pre>
	400 – Bad Request
	500 – Internal Server Error

3.4 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
userId	String	Login ID of the Operator user
msisdn	String	Destination subscriber's MSISDN
extrefnum	String	Unique Reference number in the external system

3.4.1 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
---------	------	-------------

date	String	Response Type e.g. OTHERBALANCERESP
txnStatus	String	Status of the Barring request. Transaction Status=200 means Success, Transaction Status Other than 200 means failed
DATE	String	Date and time on which user has been barred in PreTUPS. HH are in 24 Hour format
extrefnum	String	Unique Reference number of the external system.
message	String	Transaction ID
transactionId	String	If successful, ID for the processed transaction

3.5 Ejaze Balance Check

3.5.1 Overview

This API is used to perform an e-jaze balance query.

API Name	EjazeAPI - v1
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Ejaze
{api_version}	v1
{api_context}	ejazeBalance
API Location (Pre-prod)	https://preprod.gw.mfs-tkl.com/ejaze/v1/ejazeBalance
Request Method	POST
Payload	<pre>{ "airtimeBalanceResp": { "txnStatus": "string", "errorKey": "string", "date": "string", "extrefnum": "string", "code": "string", "msisdn": "string", "loginId": "string", "balance": "string" } }</pre>
Content Type	application/json
Responses	<p>200 – OK</p> <pre>{ "airtimeBalanceResp": { "txnStatus": "string", "errorKey": "string", "date": "string", "extrefnum": "string", "code": "string", "msisdn": "string",</pre>

	<pre> "loginId": "string", "balance": "string" } } </pre>
	400 – Bad Request
	500 – Internal Server Error

3.6 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
userId	String	Login ID of the Operator user
msisdn	String	Destination subscriber's MSISDN
extrefnum	String	Unique Reference number in the external system

3.6.1 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
txnStatus	String	Status of the transaction
errorKey	String	Error message if transaction failed
date	String	Date and time on which request generated by external system
extrefnum	Number	Unique Reference number in the external system
code	String	Product code
msisdn	String	Destination subscriber's MSISDN
loginId	String	Login ID of the Operator user
balance	String	Balance of the subscriber account

3.7 Ejaze PIN Change

3.7.1 Overview

This API will perform account pin change for an ejaze account.

API Name	EjazeAPI - v1
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Ejaze
{api_version}	v1
{api_context}	ejazePin

API Location (Pre-prod)	https://preprod.gw.mfs-tkl.com/ejaze/v1/ejazePin
Request Method	POST
Payload	<pre>{ "pinChangeRequest": { "msisdn": "string", "password": "string", "pin": "string", "newPin": "string", "confirmNewPin": "string", "loginId": "string" } }</pre>
Content Type	application/json
Responses	200 – OK <pre>{ "pinChangeResponse": { "date": "string", "txnStatus": "string", "message": "string" } }</pre>
	400 – Bad Request
	500 – Internal Server Error

3.8 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
msisdn	String	Destination subscriber's MSISDN
password	String	Password of the origin
pin	String	Pin of the subscriber
newPin	String	New Pin of the subscriber
confirmNewPin	String	Confirmation on New Pin of the subscriber
loginId	String	LoginId of the origin

3.8.1 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
txnStatus	String	Status of the transaction
errorKey	String	Error message if transaction failed
date	String	Date and time on which request generated by external system
extrefnum	Number	Unique Reference number in the external system
code	String	Product code
msisdn	String	Destination subscriber's MSISDN
loginId	String	Login ID of the Operator user
balance	String	Balance of the subscriber account

4 Transaction Status

4.1 Overview

This is implemented as a generic API that accepts a `referenceCode` and a transaction type, and provides a persistent response when called.

API Name	TransactionStatusAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Transactionstatus
{api_version}	v3
{api_context}	Getstatus
API Location	https://preprod.gw.mfs-tkl.com/tkash/transactionstatus/v3/getstatus?referenceCode={referenceCode}&transactionType={transactionType}
Request Method	GET
Payload	?referenceCode={referenceCode}&transactionType={transactionType}
Responses	<p>200 – OK</p> <p>The actual content of the response will be varied depending on the <code>transactionType</code> value.</p> <p>Sample Response for B2C Pending Response:</p> <pre>{ "Message": " Pending response " }</pre> <p>Sample Response for B2C Success:</p> <pre>{ "b2cResponse": { "statusTrnxId": "737820d7-8ea4-4c9f-9632-07aac9d1d201", "statusTrnxType": "B2C", "resultCode": 0, "message": "TXN ID: 522862 Successful. You have sent KShs. 10.00 bank to wallet transfer to 254773774954 - Faith Wanjala on 11/09/18 12:39. Charges KShs. 11.00. Your balance: KShs. 725,350.00.", "tklReference": 43087, "originalAmount": 10, "finalAmount": 24, "trnxFee": null, "externalRef": "EXT_TXN_ID_011", "salesOrderNo": 522862, "newBalance": 725350, "destinationMSISDN": 254773774954, "networkName": "Benson Oluga", "info": "KCB Bank to Wallet test" } }</pre>

	400 – Bad Request
	404 – <code>referenceCode</code> not found or <code>transactionType</code> not found
	<pre>{ "result": "Transaction ID e11893c5-b033-4e99-9473-a43d66b65fbb and Type B2C not found" }</pre>
	500 – Internal Server Error

4.2 Transaction Status Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
<code>referenceCode</code>	String	Reference code obtained in the transaction request
<code>transactionType</code>	String	Possible values are <ul style="list-style-type: none"> - ATP - B2C - C2B - B2B

4.3 Transaction Status Response Object Parameter Definition

Depends on the `transactionType`, check the actual response parameter definition in the respective synchronous API responses

5 Replay Notification

5.1 Overview

This is implemented as a generic API that accepts a `referenceCode` and a transaction type, and provides a persistent response when called.

API Name	CommonNotificationReplayAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	notificationReplay
{api_version}	v3
{api_context}	replayNotification
API Location	https://preprod.gw.mfs-

	tkl.com/notificationReplay/v3/replayNotification?notificationType={transactionType}&id={consumerID}&limit={notificationCount}
Request Method	GET
Payload	?notificationType={transactionType}&id={consumerID}&limit={limitCount}
Responses	<p>200 – OK</p> <pre>{ "replayNotificationResponse": { "id": "string", "notificationType": "string", "notificationCount": "string" } }</pre> <p>400 – Bad Request</p> <p>404 – id not found</p> <p>500 – Internal Server Error</p>

5.2 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
Id	String	Consumer ID/biller ID/Pay bill number used in the Register URL
notificationType	String	Possible values are <ul style="list-style-type: none"> - ATP - B2C - C2B - B2B
Limit	Number	The number of transactions to be given in the response

5.3 Response Object Parameter Definition

Similar to request parameters

6 Business to Customer Transactions

This is used for transactions where the debit account is a prefunded business disbursement account and the credit account is a subscriber or another business.

B2C transactions have an optional name lookup (NLU) API.

If the business is just disbursing funds, for example, a batch salary payment, then only the payment API call is recommended.

If the business will be performing individual transactions through some external interface e.g. mobile app/USSD, then a name lookup API call is required, followed by the payment API call.

For example, if the business is a bank and has offered its customers to send money from their bank accounts to T-kash wallets through a mobile app, then the bank must make the name lookup API call and present it to the customer with the recipient details, before making the payment API call to complete the transaction.

Note:

NLU is always synchronous. However, post NLU, the actual payment can be synchronous or asynchronous.

6.1 B2C Name Lookup (NLU) API

This API performs a lookup against the MSISDN and returns the name of the intended destination subscriber, among other info related to the transaction.

API Name	B2CPaymentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, <i>"prod"</i>
{api_name}	b2c
{api_version}	v3
{api_context}	namelookup
API Location	https://preprod.gw.mfs-tkl.com/tkash/b2c/v3/namelookup
Content Type	application/json
Request Method	POST
Payload	<pre>{ "nameLookupRequest": { "username": "string", "password": "string", "msisdn": "string", "amount": 0, "brandId": "string", "info1": "string", "info2": "string", "info3": "string", "externalRef": "string" } }</pre>
Response	<p>200 – OK</p> <pre>{ "nameLookupResponse": { "resultCode": "string", "destinationMSISDN": "string", "amount": number, "transactionFee": number, "firstName": " string", "lastName": " string", "info1": "string", "info2": "string", "info3": "string", "externalRef": "string" } }</pre> <p>400 – Bad Request</p>

	404 – account not found
	500 – Internal Server Error

6.1.1 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
Username	String	API username
Password	String	API user password
Msisdn	String	Destination subscriber's MSISDN
Amount	String	Transaction amount for the initiated transaction
brandId	String	Transaction type identifier - a numeric value to be shared at the time of account setup. May be different for test and production environments. For testing use 898 for bank to wallet, 930 every other disbursement payment
info1	String	Sender Institution/Bank/Business Name (Additional information associated with the transaction) - optional
info2	String	Sender Bank Account Name (Additional information associated with the transaction) - optional
info3	String	Sender unique ID/bank account number (Additional information associated with the transaction) - optional
externalRef	String	External reference ID

6.1.2 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
nameLookupResponse	Object	
resultCode	String	Result code of the Response, 0 for success, any other failed
destinationMSISDN	String	Destination subscriber's MSISDN
Amount	Number	Associated fee to be charged by T-kash if the transaction is completed
transactionFee	Number	Fee associated with the transaction
firstName	String	First name of the destination subscriber
lastName	String	Last name of the destination subscriber
info1	String	Sender Institution/Bank/Business Name (Additional information associated with the transaction) - optional
info2	String	Sender Bank Account Name (Additional information associated with the transaction) - optional
info3	String	Sender unique ID/bank account number (Additional information associated with the transaction) - optional
externalRef	String	External reference ID

6.2 Synchronous B2C Payment API

This API shall accept a payment request and provides a response within the request session.

API Name	B2CPaymentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	b2c
{api_version}	v3
{api_context}	Sync
API Location	https://preprod.gw.mfs-tkl.com/tkash/b2c/v3/sync
Content Type	application/json
Request Method	POST
Payload	<pre>{ "creditSyncRequest": { "username": "string", "password": "string", "msisdn": "254772071999", "amount": 10, "brandId": "string", "info1": "Institution/business name", "info2": "string", "info3": "string", "externalRef": "string" } }</pre>
Response	200 – OK
	<pre>{ "creditSyncResponse": { "resultCode": "0", "message": "transaction message ", "tklReference": "45134", "originalAmount": 10, "finalAmount": 10, "trnxFee": 1, "externalRef": "ext001", "salesOrderNo": "549425", "newBalance": 99893, "destinationMSISDN": "254772071999", "networkName": "business name", "info": "001001001" } }</pre>
	400 – Bad Request
	404 – account not found
	500 – Internal Server Error

6.2.1 Synchronous B2C Payment Request Object Parameter Definition

The actual parameter definition is similar to ["name lookup"](#).

6.2.2 Synchronous B2C Payment Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
resultCode	String	Reference code obtained in the transaction request

Message	String	Resulted message of the transaction
tklReference	String	T-kash reference number
originalAmount	Number	Original transaction amount
finalAmount	Number	Final transaction amount with taxes where applicable
trnxFee	String	Transaction fee
externalRef	String	Additional information associated with the transaction as was sent in the request
salesOrderNo	String	Sales order number/transaction ID
newBalance	Number	Source account balance post the transaction
destinationMSISDN	String	Subscriber recipient mobile number
networkName	String	Sender organisation name
Info	String	Additional information associated with the transaction as was sent in the request

6.3 Asynchronous B2C Payment API

This API is implemented in asynchronous nature where the API consumer gets an ACCEPTED (201) status the moment API is invoked with a response that contains a unique `referenceCode`.

It requires the API consumer to ["register a notification UR"](#) with the API Portal.

The `referenceCode` provided to the consumer is used to query the status of that transaction.

Note: this is not a callback

Refer to [Transaction Status](#) query

API Name	B2CPaymentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	b2c
{api_version}	v3
{api_context}	Async
API Location	https://preprod.gw.mfs-tkl.com/tkash/b2c/v3/async
Content Type	application/json

Request Method	POST
Payload	<pre>{ "creditAsyncRequest": { "username": "string", "password": "string", "msisdn": "string", "amount": 0, "brandId": "string", "info1": "string", "info2": "string", "info3": "string", "externalRef": "string" } }</pre>
Responses	<div>200 - OK</div> <pre>{ "creditAsyncResponse": { "referenceCode": "e11893c5-b033-4e99-9473-a43d66b65fbb", "transactionType": "B2C" } }</pre> <div>400 - Bad Request sample1:</div> <pre>{ "Error": "Value you have entered for msisdn is invalid. Enter your mobile number with 12 digits starting with country code. eg:25477xxxxxxx" }</pre> <div>OR</div> <pre>{ "Error": "Could not build full log message: javax.xml.stream.XMLStreamException: Unexpected symbol: STRING" }</pre> <div>Bad Request sample2:</div> <pre>{ "Error": { "code": "-1", "message": "For input string: \"string\"" } }</pre> <div>404 – account not found</div> <div>500 – Internal Server Error</div>

6.3.1 Asynchronous B2C Payment Request Object Parameter Definition

Contains the request object to perform B2C payment API. The definition is similar to [name lookup](#) request parameters.

6.3.2 Asynchronous B2C Payment Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
referenceCode	String	Reference code obtained in the transaction request
transactionType	String	B2C

7 Transaction Notification

This API is triggered to the business when a business account receives funds in any of the below 3 scenarios:

- C2B1
- C2B2
- T-kash code redemption

The notification API is like a “fire-and-forget”. If the notification end point is inaccessible, or the end point responds negatively, the transaction status does not change. However, HTTP 202 Accepted is expected as a response, else, the notification will be queued for a limited duration within which it’ll be available for retrieval by the business in the [replay API](#).

It’s the duty of the 3rd party to filter out duplicates using `transactionId`.

The notification contains two keys: “`notificationType`” and “`notificationBody`”. “`notificationBody`” have child elements which will vary depending on “`notificationType`”, while “`notificationType`” can take any of the three scenarios described above.

Simulation API

This is a GET request on `/c2b/{billerId}`. Gives a static response on the URL that was registered in the “[Register URL](#)”, and is similar to the Notification response below.

API Name	SimulateAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, “ <i>prod</i> ”
{api_name}	Simulate
{api_version}	v3
{api_context}	c2b/{billerID}
API Location	https://preprod.gw.mfs-tkl.com/simulate/v3/c2b/{billerID}
Content Type	application/json
Request Method	GET
Payload	<i>billerID</i> defined at the time of Registering Notification URL
Responses	200 – OK Content payload is similar to C2B notification depending on whether a REST or SOAP defined at the time of Registering Notification URL The actual notification happens post this simulation response on the listening endpoint registered in Register URL
	400 – Bad Request

404 – If the ID is not Registered one. Error Message is „consumer ID not found“
500 – Internal Server Error

7.1.2 Transaction Notification (SOAP Request)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:pay="http://www.telkom.co.ke/PaymentNotification/">
  <soap:Header/>
  <soap:Body>
    <pay:notificationType>RECEIVEPAYMENT</pay:notificationType>
    <pay:notificationBody>
      <transactionAmount>200.50</transactionAmount>
      <transactionDate>2013-02-22T08:22:54</transactionDate>
      <transactionId>4838384</transactionId>
      <!--Optional:-->
      <transactionFee>400</transactionFee>
      <billerMsisdn>25477586949494</billerMsisdn>
      <billerId>4383</billerId>
      <!--Optional:-->
      <accountNumber>A2393</accountNumber>
      <!--Optional:-->
      <accountBalance>444.33</accountBalance>
      <brandId>BillPayment</brandId>
      <brandName>Water Company</brandName>
      <srcAccountName>Water Company</srcAccountName>
      <sourceMSISDN>2547756636363</sourceMSISDN>
      <!--Optional:-->
      <txnTime>05-12-18T 12:13:3</txnTime>
      <accBalance>14500.65</accBalance>
      <destinationFees>0.5</destinationFees>
      <destAccName>Water Company</destAccName>
      <accountId>41231</accountId>
      <customersIDdocumentType>National ID</customersIDdocumentType>
      <customersIDdocumentNumber>41524651</customersIDdocumentNumber>
    </pay:notificationBody>
  </soap:Body>
</soap:Envelope>
```

7.1.3 Transaction Notification (SOAP Response)

Defines a response expected from 3rd party. If this response is not received, then the notification will be queued for [Replay](#)

Notification

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:pay="http://www.telkom.co.ke/PaymentNotification/">
  <soap:Header/>
  <soap:Body>
    <pay:ReceivePaymentResponse>
      <responseCode>10010</responseCode>
      <responseMessage>API was notified</responseMessage>
    </pay:ReceivePaymentResponse>
  </soap:Body>
</soap:Envelope>
```

7.1.4 Transaction Notification (REST Request)

```
{
  "notificationType": "RECEIVEPAYMENT",
  "notificationBody": {
    "transactionAmount": "125",
    "transactionDate": "2018-12-05T12:13:39",
    "transactionId": "248335",
    "transactionFee": "-5",
    "billerMsisdn": "25477xxxxxxx",
    "billerId": "103",
    "accountNumber": "123456",
    "accountBalance": "3126",
    "brandId": "545",
    "brandName": "OfflineAndSemiOnlineBillers",
    "srcAccountName": "Test Name",
    "sourceMSISDN": "254XXXXX",
    "txnTime": "05-12-18T 12:13:3",
    "accBalance": "3126",
    "destinationFees": "-5",
    "destAccName": "Test Collection",
    "accountId": "4513",
    "customersIDdocumentType": "string",
    "customersIDdocumentNumber": "string"
  }
}
```

7.1.5 Transaction Notification (REST Response)

Defines a response expected from 3rd party. If this response is not received, then the notification will be queued for [Replay](#)

Notification

```
{
  "ReceivePaymentResponse": {
    "responseCode": 10010,
    "responseMessage": "API was notified"
  }
}
```

7.1.6 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
notificationType	String	RECEIVEPAYMENT (other types to be implimented)
notificationBody	Object	
transactionAmount	String	Amount as was entered by subscriber at the time of bill payment
transactionDate	String	YYYY-MM-DDTHH24:MI:SS e.g. 2018-12-05T12:13:39
transactionId	String	Transaction identifier

transactionFee	String	Transaction fee. The actual amount credited into the biller's account is <code>transactionAmount less transactionFee</code>
billerMsisdn	String	Account number of the biller in T-kash
billerId	String	Pay bill number/Consumer ID
accountNumber	String	Account on the biller's end to which the payment is being made
accountBalance	String	Biller's account balance post receiving this transaction
brandId	String	Unique identifier of the transaction type
brandName	String	The transaction type associated with <code>brandId</code>
srcAccountName	String	Subscriber's name associated with <code>sourceMSISDN</code>
sourceMSISDN	String	Subscriber's mobile number
txnTime	String	Transaction date, ignore
accBalance	String	Biller's account balance post receiving this transaction, similar to <code>accountBalance</code>
destinationFees	String	Charges on the biller associated with receiving the payment.
destAccName	String	Biller's account name as registered in T-kash and associated with <code>billerMsisdn</code>
accountId	String	A unique identifier associated with <code>billerMsisdn</code>
customersIDdocumentType	String	Registration document type used to register subscriber in T-kash e.g. National ID, Passport
customersIDdocumentNumber	String	Registration document number used to register subscriber in T-kash associated with <code>customersIDdocumentType</code>

8 Customer to Business Transactions

These are transactions in which the subscriber initiates a transaction via subscriber channels such as USSD/APP/DSTK to a 3rd Party. The third party must establish a collections account. Categorised into 3:

1. **C2B0** – Also called **Offline C2B**. The third party collections account receives all transactions. No system integration for API calls required. Reports can be generated at regular interval or on request or downloading from the T-kash web portal <https://partner.tkash.co.ke/>.
2. **C2B1** – Also called **semi-online C2B** or **C2B without account validation**. The third party collections account receives all transactions directed to it and an API notification is sent out at the end of every transaction. System integration for the notification API is required. Reconciliations can still be done off system through other reporting mechanisms.

3. **C2B2** – Also called **online C2B** or **C2B with account validation**. The third party collections account receives all transactions directed to it but before the transaction is completed a validation of the account number entered by the subscriber is done on the 3rd party application. The transaction is only completed if the account is successfully validated. On successful validation, a dr/cr is done then an API notification is also sent out. Requires 2 API calls, one for account validation, and the other for transaction notification.

8.1 C2B1 – Customer Initiated Without Account Validation

The 3rd party is notified of a successful transaction on the URL that was registered in [“Register URL”](#). It utilises both a REST and SOAP [“notification”](#)

8.2 C2B2 – Customer Initiated With Account Validation

The 3rd party is notified of a successful transaction on the URL that was registered in [“Register URL”](#). It utilises both a REST and SOAP [“notification”](#).

However, before a transaction is completed, a [“BillerValidationAPI”](#) call is made to the 3rd party to validate if the account number entered by subscriber exists. If the account exists, the transaction is completed and a notification is sent out to the 3rd party.

The result of the notification does not change the status of the transaction

The C2B-2 can be further grouped into postpaid and prepaid. For postpaid, the 3rd expects a predefined fixed amount of payment e.g. an airline ticket payment. In this case, the bill validation response is expected to contain this amount. For prepaid, the biller is flexible and accepts any amount.

8.2.1 Bill Validation

API Name	BillerValidationAPI - v3
API Location	The API calls the URL provided in the “Register URL”
Content Type	application/json
Request Method	POST
Payload (to the biller)	<pre>validationType:BILLER validationBody{ sourceMsisdn:"string" billerAccountNumber:"string" sourceName:"string" billerId:"string" billerMsisdn:"string" amount:"string" Infol:"string" regDocType:"string" idNumber: }</pre>
Responses (from the biller)	<p>200 – OK</p> <pre>{ "billerValidationRespond": { "resultCode":"0", "userBalance":"1000.00",</pre>

	<pre> "amountOwed": "12000.00", "dueDate": "2013-02-22T08:22:54", "accountNum": "1231123321", "info1": "Extra Reference 1", "info2": "Extra Reference 2", "info3": "Extra Reference 3", "info4": "Extra Reference 4" } </pre>
	400 – Bad Request
	404 – account not found
	500 – Internal Server Error

8.2.2 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
sourceMsisdn	Numeric String	The subscriber's account mobile number including country code.
billerAccountNumber	Numeric String	The subscriber's account number on biller's system
sourceName	String	Subscriber's name
billerId	Numeric String	The billing company ID (biller's alias or paybill number)
billerMsisdn	Numeric String	Biller's account MSISDN
Amount	Float	The Payment Amount as entered by subscriber
Info1	String	The Payment Extra Info 1
regDocType	String	Customer's ID document type (KYC data)
idNumber	String	Customer's ID document number (KYC data)

8.2.3 Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
billerValidationRespond	Object	
resultCode	String	0 for successfully found account number, failed for any other code
userBalance	String	The bill payer's current balance (applicable for prepaid scenario) – describes the

		balance in the customer's balance
amountOwed	String	The payer's amount owed/due (applicable for postpaid scenario) – describes the amount the subscriber is expected to pay
dueDate	String	The due date (applicable for postpaid scenario) format 2013-02-22T08:22:54 (YYYY-MM-DDTHH:MI:SS)
accountNum	String	Account number
info1	String	The payer's account name on the biller's system
info2	String	Extra Reference 2
info3	String	Extra Reference 3
info4	String	Extra Reference 4

9 Reversal APIs

Although maker/checker mechanism exists, the approval is an internal API and is not publicly availed for the details described in the Reversal request below.

9.1 Reversal Request

9.1.1 Overview

This API acts differently on the basis of whether its consumer is source or destination of the transaction being reversed:

Scenario 1:

If the API consumer is the destination of the original transaction being reversed (e.g. in a collection wallet) then the transaction completes in one step. It assumes an approval process has been done on the consumer's ecosystem and so the consumer is just effecting a reversal.

Scenario 2:

If the API consumer is the source of the transaction being reversed (e.g. in a disbursement wallet) then the transaction reserves the amount being reversed from the subscriber account but does not credit the account to which it's being reversed. TK team have a process to approve the transaction post other external validations.

API Name	PaymentReversalAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"

{api_name}	Reversals
{api_version}	v3
{api_context}	Requestrefund
API Location	https://preprod.gw.mfs-tkl.com/reversals/v3/requestrefund
Content Type	application/json
Request Method	POST
Payload	<pre>{ "requestRefundRequest": { "username": "string", "password": "string", "transactionRef": "string", "refundReason": "1" } }</pre>
Response	200 – OK – Scenario 1
	200 – OK – Scenario 2
	<pre>{ "requestRefundResponse": { "resultCode": "string", "message": "string", "requestId": "string" } }</pre>
	400 – Bad Request
	404 – transactionRef not found
	500 – Internal Server Error

9.1.2 Reversal Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
username	String	Application username
password	String	Application password
transactionRef	String	Transaction ID = Sales Order Number of the original transaction being reversed
refundReason	Number	1 – Wrong destination account 2 – Wrong amount 3 – Wrong payment type 4 – Other

9.1.3 Reversal Response Object Parameter Definition – Scenario 1

ELEMENT	Type	DESCRIPTION
resultCode	String	0 for success, failed for any other code
message	String	Status message
requestId	Number	Unique ID that identifies the reversal request. Is used to call the reversal approval API below

9.1.4 Reversal Response Object Parameter Definition – Scenario 2

ELEMENT	Type	DESCRIPTION
resultCode	String	0 for success, failed for any other code
Message	String	Status message
requestId	Number	Unique ID that identifies the reversal request. Is used to call the reversal approval API below

9.2 Reversal Approval

9.2.1 Overview

Take the reversal `refundRequestId` received in the reversal request to complete the reversal.

Given the description in ["Reversal Request"](#), you do not this API and is only here for the documentation purpose.

API Name	PaymentReversalAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	Reversals
{api_version}	v3
{api_context}	Approverefund
API Location	https://preprod.gw.mfs-tkl.com/reversals/v3/approverefund
Content Type	application/json
Request Method	POST
Payload	<pre>{ "approveRefundRequest": { "username": "string", "password": "string", "refundRequestId": "string" } }</pre>
Response	200 – OK
	<pre>{ "approveRefundResponse": { "resultCode": "string", "message": "string", "systemReference": "string", "transactionId": "string" } }</pre>
	400 – Bad Request
	404 – <code>refundRequestId</code> not found
	500 – Internal Server Error

9.2.2 Approval Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
approveRefundRequest	Object	
username	String	Application username
password	String	Application password
refundRequestId	String	A reversal identifier that was received as <code>requestId</code> in the response to the reversal request

9.2.3 Approval Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
resultCode	String	0 for success, failed for any other code
Message	String	Status message
systemReference	Number	Unique ID that identifies the reversal
transactionId	Number	A new transaction ID (Sales Order Number) of the new transaction as a result of reversal

10 Transfer to Bank APIs

This API requires that at least a bank account is pre-registered with the T-kash account. The pre-registration allows a max of 5 bank accounts to be associated with a T-kash account. It involves 2 calls: one, to get the list of preregistered bank accounts and two, to do an actual bank transfer. The 2nd call is a similar in many ways to

10.1 Get List of Pre-registered Bank Accounts

10.1.1 Overview

API Name	PayAgentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	payAgent
{api_version}	v3
{api_context}	listpayees
API Location	https://preprod.gw.mfs-tkl.com/payAgent/v3/listpayees
Content Type	application/json
Request Method	POST
Payload	{ "listPayeeRequest": {

	<pre> "username": "string", "password": "string", "accountMsisdn": "string", "favGroupId": 1 } } </pre>
Response	<p>200 – OK</p> <pre> { "listPayeeResponse": { "resultCode": "string", "message": "string", "payeeList": [{ "accountNumber": "string", "favBankName": "string", "favGroupId": "string", "favAccountNum": "string" }] } } </pre> <p>400 – Bad Request</p> <p>404 –not found</p> <p>500 – Internal Server Error</p>

10.1.2 List Bank Accounts Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
username	String	Application username
password	String	Application password
accountMsisdn	String	MSISDN of the originating account
favGroupId	String	Type of the Payee to list. [1]T-Kash Contacts [27]Favourite Banks

10.1.3 List Bank Accounts Response Object Parameter Definition – Scenario 1

ELEMENT	Type	DESCRIPTION
resultCode	String	0 for success, failed for any other code
message	String	Status message
payeeList		Request object
accountNumber	String	Account Number of the payee

favBankName	String	Name of the bank retrieved
favGroupId	String	Code of the payee group
favAccountNum	String	Account number for the bank to receive the funds

10.1.4 Overview

This API is meant to send money from the third party collection account to the bank account which was created with their account on Tcash.

API Name	PayAgentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, <i>"prod"</i>
{api_name}	payAgent
{api_version}	v3
{api_context}	pay
API Location	https://preprod.gw.mfs-tkl.com/payAgent/v3/listpayees
Content Type	application/json
Request Method	POST
Payload	<pre>{ "payPayeeRequest": { "username": "string", "password": "string", "accountMsisdn": "string", "shortCode": "string", "amount": 0, "accountNumber": "string", "payeeType": "string" } }</pre>
Response	200 – OK <pre>{ "payPayeeResponse": { "resultCode": "string", "message": "string", "customerRefNo": "string", "amount": 0, "finalAmount": 0, } }</pre>

```

"transactionDate": "string",
"transactionCharges": 0,
"transactionId": "string",
"newBalance": 0,
"totalAmount": 0,
"sourceFees1": 0,
"sourceFees2": 0,
"sourceFees3": 0,
"sourceFees4": 0,
"destinationFees1": 0,
"destinationFees2": 0,
"destinationFees3": 0,
"destinationFees4": 0,
"destinationAccount": "string"
}
}

```

400 – Bad Request

404 –not found

500 – Internal Server Error

10.1.5 List Bank Accounts Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
username	String	Application username
password	String	Application password
accountMsisdn	String	MSISDN of the originating account
shortCode	String	Type of the Payee to list. [1]T-Kash Contacts [27]Favourite Banks
amount	String	Send to bank shortcode provided by TKL
accountNumber	String	Account number to receive the funds. (Has to be in the list in the listPayees api)
payeeType	String	Type of the Payee to list. [1]T-Kash Contacts [27]Favourite Banks

10.1.6 List Bank Accounts Response Object Parameter Definition – Scenario 1

ELEMENT	Type	DESCRIPTION
resultCode	String	0 for success, failed for any other code
message	String	Status message
customerRefNo		Customer reference number
amount	String	Payment amount
finalAmount	String	Amount after taxes
transactionDate	String	Code of the payee group

transactionCharges	String	Transaction charges for the payment
transactionId	String	Transaction ID
newBalance	String	New wallet balance after the transaction
totalAmount	String	Payment total amount, after fees and taxes and commission as applicable
sourceFees1	String	Payment source fees1 (fees on source if applicable)
sourceFees2	String	Payment source fees2 (commission to source if applicable)
sourceFees3	String	Payment source fees3 (commission to source parent account if applicable)
sourceFees4	String	Payment source fees4 (tax to source if applicable)
destinationFees1	String	Payment destination fees1 (fees on destination if applicable)
destinationFees2	String	Payment destination fees2 (commission to destination if applicable)
destinationFees3	String	Payment destination fees3 (commission to destination parent account if applicable)
destinationFees4	String	Payment destination fees4 (tax on destination if applicable)
destinationAccount	String	Destination account number

11 T-kash Code APIs

T-kash codes are special types of C2B transactions. They are of three types depending on where it is being redeemed:

- T-kash code for buy goods and services, used at T-kash merchants and other merchant aggregators
- T-kash code for agent withdrawal, used at T-kash agents and other agent aggregators
- T-kash code for ATM withdrawal, used at Verve ATMs

The API only covers T-kash codes for agent withdrawals and merchant payments (buy goods and services) for respective aggregators.

There are 3 related APIs:

- for submitting a code
- for notification
- for checking the code status in case the notification is not sent within the expected timespan

11.1 Submit T-kash Code

11.1.1 Overview

Submits the code as was generated by the subscriber and given to a merchant/agent. The successful synchronous response to this API is just an HTTP Accepted 202. However, a notification is sent on a notification URL with the actual transaction details. The notification URL is part of the request parameters and not the ones in the Register URL.

See [Notification](#) for actual values in [T-kash Code Status Inquiry](#)

API Name	PayCodeAgentAPI - v3
{ scheme }	https

{platform}	preprod, uat, dev, "prod"
{api_name}	paycodeAgent
{api_version}	v3
{api_context}	submitpaycode
API Location	https://preprod.gw.mfs-tkl.com/paycodeAgent/v3/submitpaycode
Content Type	application/json
Request Method	POST
Payload	<pre>{ "submitPaycodeRequest": { "username": "string", "password": "string", "brandId": "string", "amount": 0, "payCode": "string", "otpCode": "string", "merchantName": "string", "merchantCode": "string", "merchantLocation": "string", "info1": "string", "info2": "string", "notificationUrl": "string" } }</pre>
Response	202 – Accepted
	400 – Bad Request
	500 – Internal Server Error

11.1.2 Submit T-kash Code Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
username	String	Application username
password	String	Application password
brandId	String	Unique identifier for transaction type
amount	Number	Transaction amount
payCode	Number	T-kash Code
otpCode	Number	Always pass as 1234 (for merchant and agent redemptions)
merchantName	String	The merchant/agent name
merchantCode	String	The merchant/agent code or other unique ID for the merchant/agent
merchantLocation	String	The merchant/agent physical location
info1	String	Extra info associated with the transaction

info2	String	Extra info associated with the transaction
notificationUrl	String	URL on which you're listening for the notification

11.2 T-kash Code Status Inquiry

11.2.1 Overview

Submits the code as was generated by the subscriber and given to a merchant/agent. The successful synchronous response to this API is just an HTTP Accepted 202. However, a [notification](#) is sent on a notification URL with the actual transaction details. The notification URL is part of the request parameters and is not included in the register URL.

API Name	PayCodeAgentAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, <i>"prod"</i>
{api_name}	paycodeAgent
{api_version}	v3
{api_context}	paycodestatus
API Location	https://preprod.gw.mfs-tkl.com/paycodeAgent/v3/paycodestatus
Content Type	application/json
Request Method	POST
Payload	<pre>{ "paycodeStatusRequest": { "username": "string", "password": "string", "payCode": "string" } }</pre>
Response	<p>200 – OK</p> <pre>{ "paycodeStatusResponse": { "resultCode": "string", "message": "string", "subscriberMobile": "string", "paycode": "string", "userKey": "string", "paycodeStatus": "string", "transactionType": "string", "merchantName": "string", "merchantCode": "string", "merchantLocation": "string", "info1": "string", "info2": "string", "salesOrderNumber": "string", "salesOrderDate": "string", "salesOrderMSISDN": "string", "sourceMSISDN": "string", "sourceAccountName": "string", } }</pre>

	<pre> "destinationMSISDN": "string", "destinationAccountName": "string", "transactionAmount": 0, "notification_time": "string" } </pre>
	400 – Bad Request
	500 – Internal Server Error

11.2.2 T-kash Code Status Inquiry Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
paycodeStatusRequest	Object	
username	String	Application username
Password	String	Application password
payCode	Object	T-kash code submitted in the "Submit T-kash Code"

11.2.3 T-kash Code Status Inquiry Response Code Definition

ELEMENT	Type	DESCRIPTION
paycodeStatusResponse	Object	
resultCode	Number	0 – success, any other code is failed
message	String	Status message
subscriberMobile	String	Subscriber's mobile number
paycode	String	Paycode submitted
userKey	String	Key of the user
paycodeStatus	String	Status of the paycode
transactionType	String	Type of the transaction
merchantName	String	Name of the merchant/agent redeeming the code
merchantCode	String	Code of the merchant (Unique merchant/agent ID) redeeming the code
merchantLocation	String	Location of the merchant/agent redeeming the code
info1	String	Extra information 1
info2	String	Extra information 2
salesOrderNumber	String	Sales order number (transaction ID)
salesOrderDate	String	Sales order date
salesOrderMSISDN	String	MSISDN related to the Sales order
sourceMSISDN	String	Source MSISDN
sourceAccountName	String	Name of the source account holder
destinationMSISDN	String	Destination account MSISDN (aggregator's collection account)
destinationAccountName		Destination account name (aggregator's collection account)
transactionAmount	String	Amount of the transaction
notification_time	String	Time of the notification received

12 USSD & Bulk SMS

12.1.1 On Demand Request Object Parameter Definition

This is the request forwarded to the third party's registered url whenever a customer requests for an their demand service.

Content Type	application/json
Request Method	POST
Payload	<pre>{ "msisdn":"+254771166540", "shortcode":"254xxx", "message":"NEWS", "referenceCode":"436432366684763434522" }</pre>
Response	200 – OK 400 – Bad Request 500 – Internal Server Error

12.1.2 Subscription Request Object Parameter Definition

This is the request forwarded to the third party's registered url whenever a customer requests to either opt in or out of their subscription service.

Content Type	application/json
Request Method	POST
Payload	<pre>{ "msisdn":"+254771166540", "shortcode":"254xxx", "message":"SUBSCRIBE", "referenceCode":"436432366684763434522" }</pre>
Response	200 – OK 400 – Bad Request 500 – Internal Server Error

12.2 Bulk SMS API

12.2.1 Overview

This API shall perform send SMS and provide with a response.

API Name	BSCAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, <i>"prod"</i>
{api_name}	Sendsms
{api_version}	v3
{api_context}	Sendsmsrequest
API Location	https://preprod.gw.mfs-tkl.com/sms/sendsms/v3/sendsmsrequest
Content Type	application/json
Request Method	POST
Payload	<pre>{ "sendSMSRequest": { "CpId": "string", "CpPassword": "string", "timestamp": "string", "serviceId": 0, "senderAddress": "254XXXX", "smsTextMessage": { "message": "string" }, "address": ["string"], "receiptRequest": { "notifyURL": "string", "callbackData": "string" } } }</pre>
Response	200 – OK <pre>{ "errorStatus":false, "message":"it is valid request", "ResponseIdentifier":"436432366684763434535" }</pre>
	400 – Bad Request

500 – Internal Server Error

12.2.2 SMS Delivery Status Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
sendSMSRequest	Object	
CpId	String	Content provider identifier
CpPassword	String	Content provider password
Timestamp	String	Date and time
Serviced	String	Required for ONDEMAND & SUBSCRIPTION SMS registrations
senderAddress	String	Shortcode in the format 254xxxx
smsTextMessage		
Message		Content
Address		MSISDN
receiptRequest	Object	
notifyURL		URL to receive notifications through
callbackData		Executable at a given time

12.2.3 SMS Delivery Status Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
deliveryStatusSMSResponse	object	
errorStatus	String	The type of error
Message	String	Content
ResponseIdentifier	String	Unique response number

12.3 SMS Delivery Notification API

12.3.1 Overview

This API shall perform delivery status check for send SMS and provide with a response.

API Name	BSCDeliveryStatusAPI - v3
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	deliveryreport
{api_version}	v3

{api_context}	status
API Location	https://preprod.gw.mfs-tkl.com/sms/deliveryreport/v3/status
Content Type	application/json
Request Method	POST
Payload	<pre>//For where we query delivery status by MSISDN. { "deliveryStatusSMSRequest": { "CpId": "string", "CpPassword": "string", "timestamp": "string", "serviceId": 0, "address": "254772d4145", "referenceCode": "string" } } //For when we query delivery status without MSISDN { "deliveryStatusSMSRequest": { "CpId": "string", "CpPassword": "string", "timestamp": "string", "serviceId": 0, "referenceCode": "string" } }</pre>
Response	<p>200 – OK</p> <pre>{ "deliveryInfoList": [{ "dateTime": "2019-02-24 16:30:24", "address": "254772d4145", "referenceCode": "1", "senderAddress": "1dsfwfse32rwedfc", "message": "Sample Text", "messageId": "0239842752", "deliveryReceipt": "", "deliveryStatus": "MessageWaiting" }] }</pre>

```

"deliveryInfoList": [
  {
    "dateTime": "2019-02-24 16:30:24",
    "address": "254772d4145",
    "referenceCode": "1",
    "senderAddress": "1dsfwfse32rwedfc",
    "message": "Sample Text",
    "messageId": "0239842752",
    "deliveryReceipt": "",
    "deliveryStatus": "MessageWaiting"
  },
  {
    "dateTime": "2019-02-29 16:30:24",
    "address": "254771241125",
    "referenceCode": "1dsfwfse32rwedfc",
    "senderAddress": "25470025",
    "message": "Sample Message",
    "messageId": "0239842788",
    "deliveryReceipt": "",
    "deliveryStatus": "MessageWaiting"
  }
]
}

```

400 – Bad Request

500 – Internal Server Error

12.3.2 SMS Delivery Status Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
deliveryStatusSMSRequest	Object	
CpId	String	Content Provider Identifier
CpPassword	String	Content Provider Password
Timestamp	String	Date/Time
serviceId	String	Service Identifier
referenceCode	String	

12.3.3 SMS Delivery Status Response Object Parameter Definition

ELEMENT	Type	DESCRIPTION
---------	------	-------------

deliveryStatusSMSResponse	object	
errorStatus	String	
Message	String	
deliveryDetails	String	

12.3.4 Delivery Status Definition

STATUS	DESCRIPTION
DeliveredToTerminal	The message is delivered to terminal successfully.
DeliveryUncertain	The delivery status is unknown. For example, because the message was handed off to another network.
MessageWaiting	The message is still queued for delivery. This status is temporary.
MessageSendToSMSC	The message is accepted by SMSC but not delivered to terminal

12.4 USSD

12.4.1 Overview

This API shall allow consumer to integrate their application with the USSD GW. A separate USSD doc shall be provided, but it's basically a get request with most of the payload in HTTP headers

API Name	
{scheme}	https
{platform}	preprod, uat, dev, "prod"
{api_name}	ussd
{api_version}	v3
{api_context}	ussd-api
API Location	https://dev.gw.mfs-tkl.com/ussd/v3/
Content Type	application/json
Request Method	GET
Payload	Session-ID MSISDN ShortCode <pre>curl -X GET \ 'http://some_url:port/ussd?menu=2' \ -H 'Accept: text/xml' \ -H 'Inputs: xxxx' \ -H 'Session-ID: xxxx'</pre>

	-H 'ShortCode: xxx
Response	200 – OK
	<?xml version="1.0" encoding="UTF-8"?>
	<ussdResponse>
	</ussdResponse>
	400 – Bad Request
	500 – Internal Server Error

13 Till Notification API

The third part will have to register a notification URL with Telkom Kenya Limited to receive the till notification payload from Telkom Kenya Limited.

13.1 Till Notification (REST Request)

```
{
  "notificationType": "RECEIVEPAYMENT",
  "notificationBody": {
    "brandName": "MerchantTillPaymentWithAlias",
    "tillNumber": "624249",
    "customersIDdocumentNumber": "",
    "transactionDate": "Fri Sep 18 22:08:41 EAT 2020",
    "transactionTime": "18-09-20 21:51:07",
    "transactionId": "53775731",
    "destinationAccountName": "Test Merchant",
    "destinationFees": "0.0",
    "accountId": "66896",
    "transactionFee": "0.00000",
    "sourceMSISDN": "25477xxxxxxxx",
    "transactionAmount": "40.0",
    "brandId": "1047",
    "customersIDdocumentType": "String",
    "accBalance": "0.0",
    "accountBalance": "286.50000",
    "billerMsisdn": "25477xxxxxxxx",
    "sourceAccountName": "Test Name"
  }
}
```

13.1.2 Till Notification (REST Response)

Defines a response expected from the 3rd party.

```
{
  "ReceivePaymentResponse": {
    "responseCode": 10010,
    "responseMessage": "API was notified"
  }
}
```

13.1.3 Request Object Parameter Definition

ELEMENT	Type	DESCRIPTION
notificationType	String	RECEIVEPAYMENT
notificationBody	Object	
transactionAmount	String	Amount as was entered by subscriber at the time of bill payment
transactionDate	String	YYYY-MM-DDTHH24:MI:SS e.g. 2018-12-05T12:13:39
transactionId	String	Transaction identifier
transactionFee	String	Transaction fee. The actual amount credited into the biller's account is transactionAmount less transactionFee
billerMsisdn	String	Account number of the biller in T-kash
tillNumber	String	Till number /Consumer ID
accountBalance	String	Biller's account balance post receiving this transaction
brandId	String	Unique identifier of the transaction type
brandName	String	The transaction type associated with brandId
sourceAccountName	String	Subscriber's name associated with sourceMSISDN
sourceMSISDN	String	Subscriber's mobile number
transactionTime	String	Transaction date, ignore
accBalance	String	Biller's account balance post receiving this transaction, similar to accountBalance
destinationFees	String	Charges on the biller associated with receiving the payment.
destinationAccountName	String	Biller's account name as registered in T-kash and associated with billerMsisdn

accountId	String	A unique identifier associated with billerMsisdn
customersIDdocumentType	String	Registration document type used to register subscriber in T-kash e.g. National ID, Passport
customersIDdocumentNumber	String	Registration document number used to register subscriber in T-kash associated with customersIDdocumentType

14 ERROR CODES

Error Code	Description	Example of User Error Message	Corrective Action
-1	General error. Usually a general operation failure or an insufficient privilege.	Please contact customer support	Trace the root cause issue in the logs.
-2	Destination interface is down.	Please retry transaction later	Trace the root cause issue in the logs and try to telnet or ping the interface.
10	New password is empty.	New password is empty.	Supply a non-empty password.
11	New password and new password confirmation do not match.	New password and new password confirmation do not match.	Ensure password and password confirmation match.
13	New password length is less than the length specified in the password policy.	New password length too short.	Supply a longer password.
14	New password contains less alpha characters than the minimum defined in the password policy.	New password less than min alpha length.	Supply a password with more alpha characters.
15	New password length is greater than the length specified in the password policy.	New password length too long.	Supply a shorter password.
16	New password is in your password history.	New password in history.	Pick another password.
17	New password matches the default password.	New password matches default.	Pick another password.
18	New password is in global password dictionary.	New password in dictionary.	Pick another password.

19	New password matches the user's username.	New password matched username	Pick another password.
20	Error decrypting password using the HSM.	New password.	Investigate possible HSM or network issue.
21	Password is all numeric and policy prohibits this	Password all numeric	Provide non-numeric characters in the password
22	Password matches a prohibited pattern.	Password matches prohibited pattern	Pick another password.
901	Rejected execution (access queue is full).	Service is not available. Please try again later.	The TCS license has been reached. Contact the Telkom Technical Team.
100001	Password is incorrect.	Password incorrect, please try again.	Try another password.
100002	Account is locked due to too many invalid password attempts or on purpose by customer care.	Account is locked. Please contact customer care.	Investigate the reason for the lock before activating the account.
100003	User/terminal is locked due to too many invalid password attempts.	User is locked. Please contact customer care.	Investigate the reason for the lock before activating the terminal.
100004	Terminal type supplied does not match the one defined in the password policy.	Unable to log in with the supplied terminal type.	Try a different terminal type.
100005	User/terminal not found	Account not found. Please contact customer care to register.	Register user/terminal.
100006	User/terminal is already in session.	User already in session.	Wait for other user/terminal to complete the transaction.
100051	Invalid OTP	Invalid OTP	Request a new OTP and try again
100098	App version error	You must download new app	Redirect the user to appstore

100099	Common authentication error used for secure environments where the exact nature of the error must be hidden to prevent against security threats.	You've entered an invalid mobile number or PIN. Try again?	Action depends on the exact error. Please see the logs for the exact error code.
100105	User/terminal is already in a web session.	User already in a web session.	Wait for the other user/terminal to complete the web session.
100110	Source plan error. The source account belongs to an unauthorized plan (P2P).	System error. Please contact customer support.	Determine if the source plan is supposed to be prevented from performing this transaction before arranging to have the source plan authorized.
100111	The entered amount is below the minimum amount for the selected brand.	Amount entered is too low.	Enter a higher amount.
100112	Source limit problem for the source velocity control.	Source limit problem	Increase the limit for the source or wait for a new limit cycle (next day)
100113	The entered short code is not authorized for use with this account.	Invalid entry. Please try again or contact customer support.	Account does not have the privilege/access role for this service. This is most likely correct as configured.
100114	Payment to self is not authorized.	You are not allowed to perform this transaction.	Payment to self is prevented. This is most likely correct as configured.
100115	The maximum number of daily transactions (wallet limit) is reached.	You have exceeded the maximum number of daily transactions allowed for your account.	Customer must wait until the following day to try this transaction.
100116	The payment system is unavailable.	The payment system is unavailable. Please try again later.	Check the network connection and inquire about the availability of the 3rd party system in the context of this transaction.

100117	Subscriber is not found in lookup table.	There is a problem looking up your number. Please try again later.	Add the subscriber to the lookup table.
100118	Rejected execution (access queue is full)	Service is not available. Please try again later.	The TCS license limit has been reached.
100119	The source payment system is undefined.	System error. Please contact customer support.	The source for the service class for the transaction type is misconfigured.
100120	Duplicate transaction detected (two consecutive identical transactions).	A duplicate transaction was detected.	Duplicate transaction was prevented as configured.
100121	Bill underpayment is not allowed.	Bill underpayment not permitted. Please try again.	Bill underpayment is not permitted as configured.
100122	Bill overpayment is not allowed.	Bill overpayment not permitted. Please try again.	Bill overpayment is not permitted as configured.
100123	The maximum daily number of transactions (wallet limit) has been reached.	You have exceeded your maximum allowed number of daily transactions.	Customer must wait until the following day to try this transaction.
100124	The account information is not registered.	This account information is not registered. Please complete registration or contact customer care.	Complete account registration.
100125	The account information is not verified.	This account information is not verified. Please contact customer care.	Customer care must verify the account.
100126	The bill information not registered.	You have not registered for this service. Please register your bill first.	Customer must register for this service before using it.

100127	<p>The agent/initiator location is rejected (geolock).</p> <p>Agent is attempting to perform a transaction in a location that is not permitted. This can also occur if the cell tower location cannot be determined.</p>	Transaction rejected due to restriction on agent location.	Agent must move into the permitted location and try again.
100128	<p>The client/recipient location is rejected (geolock).</p> <p>Subscriber is partaking in a transaction in a restricted location. This can also occur if the cell tower location cannot be determined.</p>	Transaction rejected due to restriction on subscriber location.	Subscriber must move into the permitted location and try again.
100129	Supplied billing account number does not match the billing account number criteria.	Incorrect billing length or format.	Customer must enter the correct billing length or format.
100130	<p>Msave cannot be completed.</p> <p>Msave amount is greater than the original amount, or another error has occurred.</p>	Msave operation cannot be completed.	Check logs.

100131	<p>Product does not exist for the specified brand.</p> <p>It is likely that the configuration is correct; however, it is possible that the customer has chosen an amount that is not permitted and there is no associated product with it.</p>	You have entered an invalid amount.	Customer must choose another amount.
100132	The maximum daily sales order amount has been reached for the requested service.	You have reached the maximum daily sales order amount.	Customer should try again the next day.
100133	The maximum monthly sales order amount has been reached for the requested service.	You have reached the maximum monthly sales order amount.	Customer should try again the next month.
100134	The maximum weekly sales order amount had been reached for the requested service.	You have reached the maximum weekly sales order amount.	Customer should try again the next week.
100135	TAC provided by the transaction initiator is invalid or has expired.	Invalid or expired TAC provided by transaction initiator.	Customer should obtain another TAC and try again.
100136	TAC provided by the transaction recipient is invalid or has expired.	Invalid or expired TAC provided by transaction recipient.	Recipient should obtain another TAC and try again.
100137	<p>The account status is “suspended”.</p> <p>For information on account</p>	The recipient account is suspended.	Investigate the reason for the suspended status before activating the account.

	status, see the Customer Service User Guide.		
100138	<p>The account status is “terminated”.</p> <p>For information on account status, see the Customer Service User Guide.</p>	The recipient account is terminated.	Investigate the reason for the termination status before activating the account.
100139	<p>The account status is “non-active”.</p> <p>For information on account status, see the Customer Service User Guide.</p>	The recipient account is not active.	Investigate the reason for the non-active status before activating the account.
100140	Problem with the sender limit velocity check.	The sender has exceeded his maximum allowed amount	Increase the sender limit or wait for the next cycle (next day).
100141	Payment system parameter checks failed.	System error. Please try again.	System is misconfigured. Contact Telepin.
100142	A brand matrix error occurred. There is no brand matrix relationship defined between two parties.	You cannot perform this transaction from this account.	For information on brand matrices contact Telkom Customer Care.
100143	The user entered a bill account number that was too short.	You entered a bill account number that was too short.	Customer must enter the correct bill account number.
100144	The user entered a bill account number that was too long.	You entered a bill account number that was too long.	Customer must enter the correct bill account number.

100145	The user entered a bill account number that did not meet the exact length requirement.	You entered a bill account number that does not meet the exact length requirement.	Customer must enter the correct bill account number.
100146	The user entered a bill account that was not numeric.	You entered a non-numeric bill account number.	Customer must enter the correct bill account format.
100147	The user entered a negative number for bill payment (for example, -25).	You entered a negative amount for a bill payment. Enter a positive amount.	Customer must enter a positive bill account amount.
100148	The bank account is not defined.	No bank account.	Customer must register a bank account.
100149	<p>The payment system learning algorithm is disabled or the payment system is not defined for the subscriber.</p> <p>(When there is more than one node; the first time a subscriber accesses the system, the appropriate node must be found. The learning algorithm stores it to reduce lookup overhead for future requests.)</p>	System error. Please contact customer support.	Consult the logs to determine the 3rd party system. Then contact the 3rd party system to determine why the customer was not found on the 3rd party system.
100150	The account has been automatically marked as dormant. Dormant accounts are not permitted to perform this transaction.	Transaction rejected due to dormant account status.	Investigate reason for the dormant status before activating account.

100151	A default bank account or credit card does not exist.	Default bank account or credit card does not exist.	Customer must register a bank account or credit card (depends on the use case).
100152	The bank account or credit card merchant code does not exist.	Bank account or credit card merchant code does not exist or is entered incorrectly. Contact customer care.	Investigate if the merchant code configured for this user has been entered incorrectly.
100153	The recipient account is not registered at the required level for this transaction.	The recipient account is not properly registered for this transaction.	Recipient must submit the additional information required.
100154	The recipient account is not verified at the required level for this transaction.	The recipient account has not been verified to permit this transaction.	MNO must verify the additional information submitted by the recipient.
100155	The user does not have the appropriate privileges for the action.	You are not authorized to perform this transaction.	Account does not have the privilege/access role for the product associated with the service. This is most likely correct as configured. If not then the user may need to use a different access role
100156	The initiator or recipient accumulator limit was exceeded (e.g. Cash-In/Out, P2P).	The amount exceeds the initiator or recipient's limit. Please try with a lower amount.	The initiator should try with a lesser amount or try again the next day.
100157	The user has not set the secret word, which is marked as mandatory for this transaction.	Unable to perform transaction. Please set your secret word first.	The user needs to set the secret word to proceed.
100158	Unique reference null but must be provided (it is mandatory).	Unique reference is null	API caller must provide a non- null unique reference.
100159	Unique reference is already used.	Unique reference already used	API caller must provide a unique reference that has not been used before.

100161	Cannot change the transaction to POSTED	Cannot change the transaction to POSTED	Certain transactions cannot be posted. Consult the logs
100162	Cannot change the transaction to DECLINED	Cannot change the transaction to DECLINED	Certain transactions cannot be declined. Consult the logs.
100165	Source AML is not WHITE	Cannot execute this transaction	No action
100166	Source AML is BLACK	Cannot execute this transaction	No action
100167	Destination AML is not WHITE	Cannot execute this transaction	No action
100168	Destination AML is BLACK	Cannot execute this transaction	No action
100170	Sales order for refund was not found.	Unable to perform refund, sales order not found.	Customer must specify an existing sales order.
100171	Sales order did not originate with the person attempting to perform a refund and hence is not permitted. A customer who did not originate the sales order is attempting to refund a transaction.	Unable to perform refund, original transaction not performed by you.	If this transaction requires a refund, it must be done by the originator or by customer care.
100172	Sales order is not marked as posted and cannot be refunded.	Unable to perform refund, original transaction not completed successfully.	Customer care must determine why the original transaction is not Posted (PST).
100173	Sales order already refunded.	Unable to perform refund, original transaction already refunded.	No action required.
100174	Refund request not pending	Unable to execute refund request (not pending)	Cannot execute a refund request that is not pending.
100175	Refund request already executed	Refund request already executed	Cannot execute a refund request twice.

100181	Pinless payment unauthorized	Pinless payment unauthorized	User must use payment with pin, or authorize pinless payment
100182	Pinless payment above limit	Pinless payment above limit	User must use payment with pin, or use a lower amount
100201	The transaction for this rollback does not exist. Applies to Cash-In/Out and P2P.	The transaction for this refund does not exist.	Customer must specify an existing transaction.
100202	The transaction for the rollback was not posted. Applies to Cash-In/Out, and P2P.	The transaction for this refund was not completed successfully.	Customer care must determine why the original transaction is not PST.
100203	The transaction was rolled back already. Applies to Cash-In/Out, and P2P.	The transaction was already refunded.	No action required.
100204	Agent is not an operator user. Applies to Cash-In/Out, and P2P.	You are not authorized to perform this refund.	This refund must be performed by customer care.
100205	The account for the rollback does not exist. Applies to Cash-In/Out, and P2P. The customer associated with the original transaction cannot be found, so the refund cannot be issued.	The account that benefits from this rollback does not exist.	No action required.
100206	The customer is not active. Applies to Cash-In/Out, and P2P. The customer associated with the original transaction is not	The account that benefits from this rollback is not active.	No action required.

	active, so the refund cannot be issued.		
100207	Accounts limit error. Applies to Cash- In/Out, and P2P. The rollback initiator or the customer associated with the original transaction has reached the limit.	Either your account or the one that benefits from this rollback has reached their account limit.	Try the rollback on another day.
100208	The transfer ID for the original transaction does not exist. Applies to Cash-In/Out, and P2P.	Transaction cannot be refunded, contact Customer Care.	Investigate logs of the original transaction to see why a transfer ID is missing.
100209	Insufficient funds for rollback. Applies to Cash-In/Out, and P2P.	The rollback cannot be performed due to your lack of necessary funds.	Customer must obtain more funds in his or her wallet before attempting the refund.
100210	The destination account cannot be found. Applies to Cash-In/Out, and P2P. The customer associated with the original transaction cannot be found, so the refund cannot be issued.	The account that benefits from this rollback does not exist.	No action required.
100211	The transaction was not authorized. Applies to Cash-In/Out, and P2P.	You are not authorized to perform this refund.	The refund must be performed by customer care.

	The rollback initiator may not have the proper privilege to perform this refund.		
100213	Transfer to self is not permitted. Applies to Cash-In/Out, and P2P. Initiators cannot transfer cash to themselves.	You are not allowed to perform this transaction.	No action required.
100216	The recipient's cash-in limit was exceeded. Applies to Cash-In/Out, and P2P.	The amount exceeds the recipient's limit. Please try with a lower amount.	The initiator should try the transaction with a lesser amount or try it again the next day.
100217	The destination user does not exist. Applies to Cash-In/Out, and P2P. The destination terminal/user could not be found.	The destination user does not exist.	Customer should verify the input.
100218	The destination user is incorrect. Applies to Cash-In/Out, and P2P. The destination terminal/user could not be found.	The destination user does not exist.	Customer should verify the input.
100301	Agent limit exceeded (mPayment transfer).	You have exceeded your mPayment transfer limit.	Try again when agent limit is renewed.
100302	Not enough credit (mPayment transfer).	You have exceeded your mPayment transfer credit.	Add credit to the account.
100306	Account does not exist (mPayment transfer).	The mPayment transfer account does not exist.	No action required.

100307	The brand does not exist (mPayment transfer).	You have entered an invalid amount.	Try again with a valid brand.
100308	The amount is less than the required minimum (mPayment transfer).	The mPayment transfer amount is less than the required minimum.	Try again with a higher amount.
100309	The amount is more than the permitted maximum (mPayment transfer).	The mPayment transfer amount is more than the allowed minimum.	Try again with a lower amount.
100411	The sales order PIN for the rollback does not exist.	The sales order PIN does not exist.	No action required.
100417	The mobile ID for the rollback does not exist.	The Mobile ID for the sales order rollback does not exist.	No action required.
100418	The mobile number does not exist.	The mobile number for the sales order rollback does not exist.	No action required.
100421	The refund period has expired.	The refund period has expired.	No action required.
100424	The sales order for the rollback does not exist.	The sales order for the rollback does not exist.	No action required.
100425	The refund was already made for the sales order rollback.	The refund was already made for the sales order rollback.	No action required.
100426	The rollback sales order confirmation number does not match.	The refund confirmation number does not match.	Recheck the refund confirmation number and try again.
100428	The brand or amount for the sales order rollback does not exist.	You have entered an invalid amount.	Customer care should determine the cause from the logs.
100429	The sales order rollback cannot be refunded.	The refund has failed. Please contact customer care.	Customer care should determine the cause from the logs

100430	The grace period for the sales order rollback has expired.	The grace period for the sales order rollback has expired.	No action required.
100431	The amount for the sales order rollback was declined.	The amount for the sales order rollback was declined.	No action required.
100432	Rollback mSave must be done first.	The refund has failed. Please contact customer care.	This transaction was part of an mSave transaction, which must be rolled back first.
100433	The payment system for the rollback does not exist.	The refund has failed. Please contact customer care.	Contact Telepin for further action.
100434	The rollback payment system is not supported.	The refund has failed. Please contact customer care.	Contact Telkom Customer Care for further action.
100435	The rollback was declined.	The rollback was declined.	This rollback was declined by another user (checker)
100436	Negative rollback declined.	The rollback value is less than 0 and system does not allow negative rollbacks.	Do not use a negative value for the rollback value.
100437	The rollback has not been completed because it is in the doubt state.	The refund has failed. Please contact customer care.	Contact Telkom Customer Care for further action.
100438	The rollback was declined.	Rollback was declined.	Contact Telkom Customer Care for further action.
100439	The rollback was declined due to limits (account/group/layer).	Rollback was declined due to your limit restrictions.	Reduce rollback amount or try again when limits have been reset.
100440	Unable to rollback.	Unable to rollback	Check the logs for the reason of the rollback failure
100441	Not the destination.	You are not the destination of this transaction	The rollback is configured to be done only by the destination.
100505	Sales request was not defined.	The sales request which you are searching for cannot be	Ensure you are using a valid request ID.

		found	
100506	The maker of the sales request was not from the source account.	You must be the source account.	The maker of the sales request must complete the task.
100507	The maker of the sales request was not from the destination account.	You must be the destination account.	The checker of the sales request must complete the task.
100508	There is no source account defined for the sales request.	You must specify a source account.	Specify a source account.
100510	No source account exists for the supplied phone number.	There is no source account for the phone number.	Use an account that exists
100511	The source account for the sales request does not exist.	The source account does not exist.	Use or create an account that exists
100512	Sales request execution is unauthorized.	Sales request execution unauthorized	This sales request cannot be executed by the calling party. The sales request id being used must be incorrect.
100513	Sales request is in an invalid state.	Sales request cannot be executed	Check the state of the sales request.
100514	Sales request expired	Sales request expired	This sales request cannot be executed as it has expired. A new sales request must be created.
100550	Invalid pin length (off-net redemption).	Invalid PIN length.	Try again with a valid PIN length.
100551	PIN does not exist (off-net redemption).	Invalid PIN.	Try again with a valid PIN.
100552	PIN is not active (off-net redemption).	PIN not active.	No action required.
100553	PIN password is incorrect (off-net redemption).	PIN password is incorrect.	Try again with a different PIN password.
100554	PIN is not for the target	PIN cannot be sold for this	Try again using a different PIN.

	MSISDN (off- net redemption).	MSISDN.	
100555	Networks mismatch (off-net redemption).	The agent and customer are not part of the same network.	Ensure that the source and the destination belong to groups which share the same network ID.
100556	The amount is set to zero (off-net redemption).	0 is an invalid amount for this voucher.	Ensure you are activating a valid Voucher PIN.
100557	Credit problem (off-net redemption).	Problem crediting account. Please contact provider.	Please see TCS logs for a more detailed reason.
100560	Voucher error other (voucher redemption).	No longer valid error code.	N/A
100561	Voucher does not exist (voucher redemption).	The voucher does not exist or could not be found.	No action required.
100562	Voucher expired (voucher redemption).	The voucher has expired.	No action required.
100563	Voucher is not active (voucher redemption).	The voucher is not active.	No action required.
100564	Voucher batch is not active (voucher redemption).	The voucher batch or voucher is no longer active.	Ensure the voucher is valid.
100565	Voucher target error (voucher redemption).	Target specified does not exist OR pin that is being sold does not belong to the user.	Try again using a different PIN.
100566	Voucher brand error.	Voucher brand not found or not active	Make sure the voucher brand is active.
100567	Voucher redemption service error	Voucher is redeemed by an unauthorized service	Make sure the voucher is redeemed by the correct service, which is defined in the voucher brand.
100568	Voucher source MSISDN error. The source MSISDN supplied at redemption is missing or does not match	The source MSISDN for the voucher is incorrect	Supply the correct source MSISDN.

	the source MSISDN of the voucher.		
100570	Voucher is out of stock.	The voucher is out of stock.	No action required.
100571	Voucher QR generation error (invalid terminal ID).	Voucher QR generation error	Provide the correct terminal ID.
10572	Voucher QR generation error (invalid account ID).	Voucher QR generation error	Provide the correct account ID.
100601	Sales not authorized (credit transfer).	You have entered an invalid amount.	Try again with a valid amount.
100602	There is not enough credit for the credit transfer.	You do not have sufficient funds to perform this transaction.	Try again when more credit is available.
100605	The source account for the credit transfer does not exist.	Your account is inactive	Use a valid account.
100606	The destination account for the credit transfer does not exist.	The account you are transferring credit to does not exist	Create a new account or use an existing one.
100608	The daily seller limit has been exceeded for a credit transfer	You have exceeded your daily selling limit.	Try again the next day.
100610	A credit transfer to self is not allowed.	You cannot transfer credit to your own account.	No action required.
100615	The agent account for the mPayment does not exist.	The agent account does not exist.	Create a new account or use an existing one.
100616	The mobile number for the mPayment does not exist.	The mobile number does not exist.	Try again with a valid mobile number.
100617	An invalid amount was entered for an mPayment.	You entered an invalid amount.	Try again with a valid amount.
100700	The wallet for this sales order is disabled.	This wallet is disabled.	Enable the wallet.

100701	The account for the sales order does not exist.	The account does not exist.	Create a new account or use an existing one.
100702	There is not enough credit for the sales order.	You do not have enough credit.	Try again when more credit is available.
100787	This is a duplicate sales transaction sales order.	No longer valid error code.	N/A
100788	The brand or the amount for the sales order does not exist.	You have entered an invalid amount.	Try again with a valid brand.
100789	Other sales order database error.	An error has occurred during this transaction. Please contact Customer Care	Please contact Customer Care.
100790	The amount of the sales order is outside of the flexible limits.	You entered an invalid amount.	Try again with a valid amount.
100791	The amount of the sales order does not align with a flexible step value.	You entered an invalid amount.	Try again with a valid amount.
100792	The amount of the sales order is below the flexible limits.	You entered an amount below valid limit.	Try again with a valid amount.
100793	The amount of the sales order is above the flexible limits.	You entered an amount above valid limit.	Try again with a valid amount.
101001	The beneficiary mobile number is missing.	The beneficiary mobile number is missing.	Add a beneficiary mobile number to the account.
101002	The customer mobile number is missing.	The customer mobile number is missing.	Add the associated customer mobile number when signing up a beneficiary.
101003	The beneficiary first name is missing.	The beneficiary first name is missing.	Add the beneficiary's first name to the account.
101004	The beneficiary last name is	The beneficiary last name is	Add the beneficiary's last name to the

	missing.	missing.	account.
101005	The beneficiary full name is missing.	The beneficiary full name is missing.	Add the beneficiary's full name to the account.
101006	The customer adding the beneficiary does not exist.	Customer not found.	Have the customer use the correct MSISDN.
101007	The maximum number of beneficiaries has been reached for this customer.	Maximum number of beneficiaries has been reached.	Customer must remove an existing beneficiary to add a new one.
101008	The beneficiary already exists.	The beneficiary already exists.	No action required.
101009	The beneficiary date of birth is missing.	The beneficiary date of birth is missing.	Add the beneficiary's date of birth to the account.
101010	The beneficiary's operator name is null or empty.	The beneficiary operator name is missing.	Add the beneficiary's operator name.
101011	The beneficiary's billing method is null or empty.	The beneficiary billing method is missing.	Add the beneficiary's billing method.
101012	The beneficiary's billing method is an invalid value.	The beneficiary billing value is invalid	Must be prepaid (PRE) or postpaid (PST)
101013	The beneficiary's country ID is null or empty.	The beneficiary country Id is missing.	Add beneficiary country ID
101014	The beneficiary's nationality ID is null or empty.	The beneficiary Nationality ID is missing	Add beneficiary nationality ID
101015	Payee number was not found.	Payee number not found	Use a valid payee number (MSISDN).
101016	Payee account is not active.	Payee account not active	Determine why the account is not active.
101017	The payee account and the last 4 digits do not retrieve an account.	Payee record not found	Customer should retry with the correct information.
101018	Payee bank card status is not active	Payee Status is not active	Determine why the payee status is not active

101019	Bank card status is not active	Bank Account is not active	Determine why the bank card/account is not active
101020	Bank account trust is "Uncategorized" or not "White".	Bank Account status error	Check the bank account trust status.
101021	Bank card does not have the proper category value	Bank Account not permitted to be used	Determine why the bank card/account category value.
101022	Service is set to be used only by the owner account.	Service cannot be used by payee only primary account holder	No Action
101023	The service can be used only by the owner of the account.	Service cannot be used by payee only primary account holder	No Action
101024	Service is set up to be used by different accounts.	Service can only be used by Payee	No Action.
101025	Primary account is not active.	Primary account holder is not active, service declined.	Determine why the primary account holder is not active.
101026	Bank card service is not found or is inactive.	Bank card Service status not accepted	Add bank card service or activate bank card service status to 'A'.
101027	Bank card is over daily accumulated limit.	Bank card daily limit has been reached.	Try again when the limit is reset.
101028	Bank card is over the accumulated monthly limit.	Bank card monthly limit has been reached.	Try again next month.
101029	Bank card expiry format is incorrect.	Bank card expiry date is incorrect.	Enter a valid expiry.
101030	Bank card has expired.	The bank card you are trying to use has expired	Use a different bank card.
101031	Problem with the access role for the account.	Your access role could not be found	Use an account with a valid role or assign a valid role to the account.
101032	Adding a clerk error.	User not inserted	Check your data and try again.

101033	Deactivate Account MSISDN parameter is invalid.	The MSISDN you have provided has been deactivated or is invalid.	Use a valid MSISDN.
101034	Deactivated Account MSISDN and Integration parameters do not retrieve a valid account.	The MSISDN you have provided cannot be deactivated because it has already been deactivated.	Use a valid MSISDN and valid account integration values.
101035	Invalid MSISDN for refund.	MSISDN you have provided is invalid for this refund	Use a valid MSISDN.
101036	Invalid transaction ID for refund.	Could not retrieve the transaction with the ID provided.	Use valid transaction ID.
101040	Reset user is not found.	The reset user cannot be found.	Use a valid user name.
101041	The customer was not found when adding or removing a sell relationship.	Couldn't get customer information for customer mobile number supplied.	Use a valid MSISDN and account integration combination.
101042	The agent was not found when adding a sell relationship.	Could not find the account you are trying to set the relationship for.	Use a valid MSISDN and account integration combination.
101043	Update profile cannot find the account that you wish to update.	Account could not be found using the MSISDN you provided.	Use a valid MSISDN.
101044	Update profile could not validate the account you wish to update.	Could not update account, please ensure your update data is valid.	Ensure all details are valid.
101045	The user account that you are trying to disable was not found.	Could not find the account you wish to disable.	Use a valid MSISDN.

101046	Could not find customer with supplied information.	Users can't be deactivated because of the invalid supplied customer information	Use valid MSISDN and account integration values
101047	The customer you are trying to remove was not found using the provided MSISDN.	Could not find the account you are trying to remove.	Use valid MSISDN value.
101048	The customer you are trying to remove was not found using the provided account integration value.	Could not find the account you are trying to remove.	Use valid account integration value.
101049	Error occurred while trying to remove the customer	Error in removing account	Internal function failure. Please contact Telepin.
101050	AML scan failure. Entity ID is null	AML Scan Failed. Entity ID must not be null	Provide the correct entity ID.
101051	AML scan failure. Entity does not exist	AML Scan Failed. Entity ID does not exist	Provide the correct entity ID.
101052	AML scan failure. Full name is null	AML Scan Failed. Full name is null	Provide the correct full name.
101053	AML scan failure. Country name is null	AML Scan Failed. Country name must be supplied	Ensure that the country name is populated.
101054	AML scan failure. Scan type is invalid	The scan type that you are using is invalid.	Use a valid scan type (1 or 0).
101055	AML scan failure. MSISDN is invalid	AML Scan Failed. Invalid MSISDN was supplied	Ensure scan type and customer scan do not match
101056	AML scan failure. Scan could not find a beneficiary.	AML Scan Failed. Beneficiary not found.	Ensure that the scan type and the beneficiary scan are not the same.
101058	AML scan failure. Missing full name	AML Full name not set	Add a full name.
101059	AML scan failure. Missing Country Name	AML Country name not set	Add the country name.
101061	Beneficiary brand is invalid.	Beneficiary Brand was not	Beneficiary brand needs to be

		found	configured. Contact Telepin
101062	Beneficiary scan type is invalid.	Beneficiary scan type is unsupported	Scan type value must be set to 1 (MSISDN) or 2 (SALES ORDER).
101063	Beneficiary MSISDN is invalid.	Beneficiary MSISDN is invalid	Use a valid MSISDN.
101064	Beneficiary sales order is invalid.	Beneficiary Sales order is invalid	Use a valid sales order.
101066	Beneficiary HUB is invalid.	Beneficiary Brand not configured	Please contact Telepin
101067	Beneficiary HUB MSISDN is invalid.	Beneficiary MSISDN is missing	Use a valid MSISDN.
101068	Beneficiary HUB scan type is invalid.	Beneficiary scan type unsupported	Scan type value must be set to 1 (MSISDN) or 2 (SALES ORDER).
101069	Beneficiary HUB sales order is invalid.	Beneficiary Sales order is invalid	Use a valid sales order.
101070	Change Account Status cannot find account.	Account Not Found	Use a valid MSISDN.
101071	Could not find the user.	User not found	Use a valid user name.
101072	Single call voucher failure.	No cards found four this voucher	Use valid voucher values.
101073	Self Transfer failed.	Username has entered invalid number of parameter	Internal logs must be consulted. Please contact Telepin
101074	Account to modify user limits was not found.	User not found	Use a valid user name.
101075	Account is not allowed to be modified because of owner relationships.	Account cannot be modified due to ownership restrictions	Use a different user to modify limits.
101076	AML status already verified.	Account status has already verified	No action is required.

101077	Unable to change the language.	Cannot change language. Please ensure you have the correct privileges and language exists.	Ensure that the language exists and the privileges are present.
101078	Unable to activate the batch.	Batch activation failed	Contact Telepin to refer to internal logs.
101079	Invalid PIN ID.	Your pin was invalid	Use a valid PIN.
101080	PIN cannot be disabled due to and invalid status.	Cannot disable pin. Status is invalid to do this action.	Ensure that the PIN ID and status are correct and try again.
101081	Registering a bill failed because the brand is invalid.	Could not find brand to register this bill.	Use a valid brand for registering a bill.
101082	Failed to register a bill.	Registration from bill has failed	Contact Telepin.
101083	Failed to register a bill. Minimum length was not met.	Bill registration failed. Minimum length not met.	Use a valid length as configured in the web admin.
101084	Failed to register a bill. Maximum length was not met	Bill registration failed. Maximum length was not met.	Use a valid length as configured in the web admin.
101085	Failed to register a bill. Exact length was not met	Bill registration failed. Exact length was not met.	Use a valid length as configured in the web admin.
101086	Failed to register a bill because non- numeric characters were entered.	Bill registration failed. Bill must be numeric.	Use numeric values.
101087	Failed to register a bill. Offline bill bulk was not found.	Bill registration failed. Offline bill bulk was not found.	Add a biller to the offline bill bulk table.
101088	Authentication failure.	Password was invalid or could not be read.	Use a valid password.
101110	Msave account registration failure. Account is not verified.	Account information not verified	Ensure that the account is verified.

101111	Msave account registration failure. Account is already registered.	Account is already registered for msave	No action is required.
101112	Msave account registration failure. Account is not registered for msave.	Account not registered for msave	Register the account for mSave
101113	Msave account registration failure. Msave percent is invalid.	Invalid msave percent.	Enter a valid percent value.
101114	Msave account registration failure. Invalid msave parameters	Invalid msave parameters	Enter valid msave parameter.
101115	Msave account registration failure. Percent value is greater than the maximum allowed percent value.	Msave percent value is larger than the allowed maximum.	Enter a valid msave percent.
101116	Msave account registration failure. Fixed value is less than the minimum allowed fixed value.	Msave value is smaller than the minimum fixed value allowed.	Enter a valid msave fixed value.
101117	Msave account registration failure. Fixed value is greater than the maximum allowed fixed value.	Msave fixed amount is more than maximum fixed amount allowed	Enter a valid msave fixed value
101118	Msave account registration failure (type 1)	Unable to register for msave. Please contact provider.	Internal logs must be consulted. Please contact Customer Care.
101119	Msave account registration failure. (type 2)	Unable to register for msave. Please contact provider	Internal logs must be consulted. Please contact Customer Care.
101120	Sales request failure. The source account could not be found.	MSISDN supplied does not match the request agent.	Enter a valid source account MSISDN.

101121	Sales request failure. Client interface is not set.	Client interface not set	Enter a valid short code, destination, and source for interface.
101122	Registration failure. Insufficient data (type 1).	Information missing. Please ensure all mandatory data is filled in.	Internal logs must be consulted. Please contact Customer Care.
101123	Registration failure. Insufficient data (type 2).	Information missing. Please ensure all mandatory data is filled in.	Internal logs must be consulted. Please contact Customer Care.
101124	Hub transaction failure. Negative value supplied.	Negative amount rejected. System does not allow for negative values	Please contact Customer Care.
101125	Hub transaction failure. Transaction ID not found.	Transaction ID supplied was not found.	Use a valid transaction ID.
101126	Hub transaction failure. External system does not recognize the transaction ID.	The transaction number supplied was invalid.	Use a valid transaction ID.
101127	Secret word failure.	Secret word requested failed. Please try again.	Consult logs for further information.
101128	Authenticate Random Check failed.	User name and MSISDN is invalid combination.	Use a valid MSISDN and random code combination
101129	Username Check failure. Username not found.	User name not found.	Use a valid username.
101130	Register Bank Card failure. Unauthorized bank card.	Unauthorized bank card.	Use a non-black or a grey listed bank card.
101131	Register bank card failure. Invalid prefix.	Invalid prefix.	Ensure the BIN prefix and the bank card PAN have the same prefix
101132	Register bank card failure. BIN length does not match specified length.	BIN length does not match required length	Ensure the PAN length and the card BIN length are equal

101133	Register bank card failure. Maximum number of cards has been reached.	Maximum number of cards that can be unverified has been reached.	Verify the account and ensure that the maximum number of non-verified bank cards has not been reached.
101134	Register bank card failure. Maximum number of cards has been reached.	Maximum cards reached for this account.	No action required.
101135	Register bank card failure. Maximum number of uncategorized cards has been reached.	Max uncategorized cards for this account have been reached.	No action required.
101136	Register bank card failure. Primary ID does not match the user.	Primary ID does not match the current account.	Add primary ID for the account being registered.
101137	Register bank card failure. Maximum number of cards has been reached for the primary ID.	Maximum number of cards reached for this primary ID	No action required.
101138	Register bank card failure. Expiry date does not fit the proper format	Expiry date supplied is not in the valid format.	Ensure the bank card expiry is in the proper format MMY.
101139	Register bank card failure. Expiry date is before the grace period.	Expiry date of the bank card is before the allotted grace period.	Use another card.
101140	Register bank card failure. Last digits do not exist.	Bank card supplied matches another bankcard linked to this account.	Use a valid card.
101142	Register Payee failure. Principal account not found.	Principal account not found	Use valid account credentials.
101143	Register Payee failure. Principal account does not match bank card account.	Principal account not matching bank card account	Ensure that the principal account matches the principal account for the card.

101144	Register Payee failure. Maximum number of allowed payees for the unverified principal account has been reached.	You have reached the maximum number for payees for this unverified account.	Increase the maximum number of payees per card.
101145	Register Payee failure. Maximum number of allowed payees for this principal account has been reached.	Maximum payee per card reached	Increase the maximum number of payees per card.
101146	Register Payee failure. Maximum number of cards per payee has been reached	Maximum card per payee reached	Increase the maximum number of cards per payee.
101147	Password reset failed because the 3 password answers were not populated.	Password answers for reset password challenge have not been set.	Set values for password answers (1, 2, and 3)
101148	Password reset failed due to incorrect attempt.	Password answer is incorrect.	Verify that the information is correct.
101149	Terminate Account failure. Account was not found.	Could not find the account you wish to terminate	Use a valid MSISDN and account integration combination.
101150	Terminate Account failure. Account is already terminated.	Account which you are trying to terminate has already been terminated	No action required.
101151	Verify Account failure. Various reasons will be posted to the logs.	Account data failed to be verified. Please contact your provider	Consult logs for a more detailed error description.
101152	Update Bank Card Expiry failure. Expiry date is in an unknown format.	Expiry date is in an unknown format. Please use valid format	Dates need to be in the correct format.
101153	Update Bank Card Expiry failure. Bank card has expired	The bank card you are trying to us has expired.	No action required.
101154	Bank card registration failure.	Card already exists	Use another card.

	Card already exists		
101158	Signup msisdn below min length	Signup msisdn below min length	Use correct msisdn
101159	Signup msisdn above max length	Signup msisdn above max length	Use correct msisdn
101160	USSD Location check time out	Location check has failed. Please try again.	No action required.
101161	Signup On Demand is not configured for the short code service.	This service is unavailable. Please ensure you are registered to use this service.	Configure the service to use the sign up on demand feature or ensure that you are using the correct short code service.
101162	Signup On Demand failed because the user already exists in the system.	Please select another username to sign up for this service.	Use a different user name or MSISDN.
101163	Signup failed because user name contains an illegal character.	Your user name contains invalid characters. Please ensure all characters are valid.	Ensure that the user name does not include an illegal character ‘.’. For example: my.user
101164	Signup failed because the user is not configured in SUBSCRIBERS_LOOKUP table	Your signup has failed. Please check with your provider to ensure you are eligible for this service.	Ensure that the user exists in the table.
101165	Signup failed because the user does not match the proper signup prefix.	Your MSISDN does not match the prefix that is required. Please ensure you are using the right prefix.	Ensure that the user has a valid prefix in the configuration: ‘Server_Signup_Prefix’
101166	Signup failed because exact length of MSISDN does not match.	Your MSISDN does not meet the length required to sign up for this service.	Ensure that the MSISDN has the exact length specified.
101167	Signup failed. Account creation failed	Failed to create new account. Please contact your provider	Internal function failed. Please contact Telepin

101168	Signup failed. User creation failed	Failed to create new user. Please contact your provider	Internal function failed. Please contact Telepin
101169	Registration failed due to an invalid primary ID.	Could not register your account. Please check your Primary ID to ensure it is correct.	Ensure you are using a valid primary ID.
101170	Signup failed because the user name is invalid.	Please ensure that your user name is numeric	Use a numeric username or change the Server_Signup_Numeric field to false.
101171	TAC was not found.	Password was not set for this user	Ensure that you are using a valid user and TAC combination.
101172	TAC does not match the supplied user.	User name password combination is invalid.	Ensure that you are using a valid TAC for the selected user.
101173	TAC brand does not match the user and password combination supplied.	User name, password, and brand combination is invalid.	Ensure that you are using a valid TAC brand for the selected user and password combination
101174	TAC expiry time is not set.	Password expiry was not set	Ensure you are setting the TAC expiry time.
101175	TAC has expired.	Password has expired	Reset your TAC and try again.
101177	Account creation failed because the account already exists.	Could not create your account because the account you are trying to create already exists.	Ensure that you are using a MSISDN that does not exist or that the MSISDN recycling is set to true
101178	Setting alias failed because the new alias is longer than the maximum allowance.	Alias name is longer than maximum characters allowed	Ensure that the alias meets the required maximum.
101179	Setting alias failed because the new alias is already in use.	Alias name already in use.	Ensure alias is a unique alias
101182	AML Scan Alternate has failed because the sender's full name is null.	Sender full name must be filled in.	Ensure that the sender's full name is entered.

101183	AML Scan Alternate has failed because the sender's county name is null.	Country name must be filled in.	Ensure that the sender's country name is entered.
101185	AML Scan Alternate has failed because the sender's MSISDN was not found.	Sender MSISDN was not found.	Ensure that the sender's MSISDN is entered.
101187	AML Scan Alternate has failed because the AML_CHECK_BRAND was not found.	AML Scan brand was not found.	Ensure that the AML_CHECK_BRAND is configured and exists in the system.
101188	AML Scan Alternate has failed because the beneficiary's full name is null.	Invalid beneficiary full name	Ensure that the beneficiary's full name is entered.
101189	AML Scan Alternate has failed because the beneficiary's country is null.	Invalid beneficiary country name	Ensure beneficiary country name is not null
101200	Authentication failed because the temporary password has expired.	Temporary password expired. Please request a new temporary password.	Reset the password of the user and try again
101201	Authentication failed because the registration level was not met.	You do not have the registration level required for this transaction.	Register the user to have the proper registration level.
101202	Authentication failed because the minimum registration level was not met.	You do not have the minimum registration level required for this transaction.	Register the user to have the proper registration level.
101211	The limit for the source account has been exceeded.	The source account has reached its limit. Please try again at a later date.	Ensure that the source account has not reached its limit.
101212	The limit for the destination account has been exceeded.	The destination account has reached its limit. Please try again at a later date.	Ensure that the destination account has not reached its limit.

101213	Transaction initiator or source wallet is disabled.	Transaction initiator account or initiator source wallet is disabled.	Please check the source account and the wallet status and try again.
101214	Transaction recipient or destination wallet is disabled.	Transaction recipient account or recipient destination wallet is disabled.	Please check the destination account and the wallet status and try again.
101220	Registration failed. Auto-sweep lower limit was invalid during registration.	You have exceeded your auto top-up limit.	Enter a valid value.
101221	Registration failed. An invalid auto top-up amount was entered.	You have entered an invalid auto top-up amount.	Try again with a valid amount.
101222	Registration failed. Auto sweep upper limit was invalid during registration.	Invalid upper auto-sweep out limit.	Enter a valid value.
101223	Registration failed. Auto-sweep amount limit was invalid during registration.	Invalid auto-sweep out amount.	Enter a valid value.
101224	Registration failed. Invalid birth date was entered.	You entered an invalid date of birth.	Enter a valid birth date.
101225	Registration failed. An invalid email address was entered.	You entered an invalid email address.	Enter a valid email address.
101226	Registration failed. An invalid nationality was entered.	You entered an invalid nationality.	Enter a valid nationality.
101227	Registration failed. An invalid country of birth was entered.	You entered an invalid country of birth.	Enter a valid country of birth.
101228	Registration failed. An invalid country of residence was entered.	You entered an invalid country of residence.	Enter a valid country of residence.
101229	Registration failed. An invalid area of interest.	You entered an invalid area of interest.	Enter a valid area of interest.

101230	Register account failed. Invalid auto sweep in flag value.	Invalid auto sweep in flag value.	Enter a valid (0 or 1) value.
101231	Register account failed. Invalid value for auto-sweep.	Invalid auto sweep out flag.	Enter a valid (0 or 1) value.
101232	Register account failed. Invalid value for lower limit.	Value is greater than auto sweep out upper limit.	Use a valid value.
101233	Register account failed. Invalid value for auto-sweep in.	Value is greater than auto sweep in upper limit.	Use a valid value.
101234	Register account failed. Auto-sweep out upper limit is invalid due to wallet limit.	Auto sweep out upper limit is invalid for the account wallet limit.	Use a valid value.
101235	Invalid registration level.	You are trying to register with a level that does not exist.	Correct your desired registration level and try again.
101236	Registration is not permitted.	Registration is not permitted because account id is blacklisted	Please try with an account ID that is not blacklisted.
101237	Invalid registration alert. Threshold limit amount.	Invalid registration alert threshold amount.	Correct the threshold amount and try again.
101238	Registration error. Invalid user type.	Invalid user type	Enter the correct user type.
101239	Registration error, a pending registration exists	Pending registration exists	Resolve the pending registration before proceeding with new one
101240	Invalid wallet number passed during disabling wallet.	Invalid wallet passed during disabling wallet	Correct the wallet ID and try again. Wallet ID should be 1-4 only.
101241	Account does not exist for wallet disabling.	You are trying to disable a wallet for a non-existing account.	Try with an existing account.
101242	Invalid wallet number passed during enabling wallet.	Invalid wallet passed during enabling wallet	Correct the wallet ID and try again. Wallet ID should be 1-4 only.

101243	Account does not exist for wallet enabling.	You are trying to enable a wallet for a non-existing account.	Try with an existing account.
101260	Unable to obtain beneficiaries, missing brand ID.	Unable to obtain beneficiaries, missing Brand ID	Provide the brand ID and try again.
101261	Unable to obtain beneficiaries, missing receiving country ID.	Unable to obtain beneficiaries, missing receiving country id.	Provide the country ID and try again.
101265	Bank relation ID does not match during balance.	Cannot obtain balance as bank relation id doesn't match.	Correct bank relation ID and try again
101270	Encryption error.	Unable to encrypt input	Correct the configuration of the encryption key.
101271	Cannot send notification due to an error. The most likely reason for this is that you are using an invalid notifier.	Cannot send notification due to an error	Make sure you are using the correct notifier and try again
101272	Invalid wallet number in GETMWSTATUS.	Invalid wallet number	Ensure that the wallet number is set to 1 – 4.
101273	Account does not exist in GETMWSTATUS.	Account does not exist	Use a valid account MSISDN.
101274	Account does not exist in VALIDATEID.	Account does not exist	Use a valid account MSISDN.
101275	ID does not match in VALIDATEID.	ID does not match	Use the correct ID.
101276	Account does not exist when calling VALIDATESECRETANSWER.	User does not exist	Use a valid account MSISDN.
101277	Supplied secret answer does not match.	Secret Answer does not match	Customer must provide the correct secret answer.
101284	SBMCheck status is not set.	SBMCheck is not found	Set the status for SBMCheck.

101285	SBMCheck status is not SBM.	SBMCheck is not SBM	Make sure that the transaction is pending.
101286	Admin task name is invalid	Unable to process transaction	Correct the admin task name in the calling service.
101287	Admin task brand ID is invalid.	Unable to process transaction	Correct the admin task brand ID in the calling service.
101390	Invalid action.	Invalid action	Set the action to be 1 or 2.
101391	User with this MSISDN does not exist while attempting to downgrade an account	Invalid MSISDN	Use a valid account MSISDN.
101392	The supplied primary ID and the customer's primary ID do not match while attempting to downgrade an account	Primary ID verification failed	Use the customer's actual primary ID.
101393	Customer is already blacklisted.	Already blacklisted	No corrective action is necessary if the intent is to blacklist the customer.
101394	Invalid registration level was supplied.	Invalid registration level	Use a valid registration level between 1 and 9.
101400	Invalid bank profile ID was supplied.	Invalid bank profile ID	Use a valid bank profile ID
101401	User with this MSISDN does not exist while attempting to add a bank service	Invalid MSISDN	Use a valid account MSISDN
101402	The supplied primary ID and the customer's primary ID do not match while attempting to add a bank service.	Invalid primary ID	Enter the customer's correct primary ID number.
101403	Supplied last 4 digits of the customer's CASA account do not match.	Invalid last 4 digits	Enter the last 4 digits of the customer's CASA account.

101404	Customer does not have a bank card that matches the issuing bank and the last 4 CASA digits that were provided.	Bank card does not exist	Enter the customer's correct bank card number.
101405	Customer does not have a bank card payee for their bank card	Bank card payee does not exist	Enter the customer's bank card payee, or create a new one.
101406	The group that the customer belongs to is not valid to perform the bank card service function.	Invalid group	Change the customer's group.
101500	Relation ID argument was not supplied	Invalid relation ID	Supply relation ID argument
101501	The supplied relation ID and the customer's relation ID do not match.	Relation ID verification failed	Enter the customer's relation ID.
101502	User with this MSISDN does not exist.	Invalid MSISDN	Use a valid account MSISDN.
101503	The supplied primary ID and the customer's primary ID do not match.	Primary ID verification failed	Enter the customer's primary ID.
101504	Primary ID argument was not supplied.	Invalid primary ID	Provide the primary ID argument.
101505	Customer is not registered at level 3.	Not a level 3 customer	Register the customer at level 3.
101600	Cannot update the reference for the submitted transaction.	Submitted Transaction failed to update reference	Make sure that the reference input field is correct.
101601	Cannot update the status for the submitted transaction.	Submitted transaction failed to update status	Make sure that the status input field is correct.

101602	The submitted transaction is missing a return status.	Submitted transaction is missing status	Make sure that the status input field is filled.
101603	Cannot find the sales order number for the submitted transaction.	Submitted transaction sales order is not found	Make sure that the sales order input is valid.
101604	The transaction is not in the submitted state (SBM)	Submitted transaction status is not pending	Make sure that you are checking the correct transaction.
101610	User password status: user name is null.	User name is null	Provide the user name.
101611	User password status: user type is null.	User type is null	Provide the user type.
101612	User password status: user type is invalid.	User type is invalid	Provide the correct user type.
101620	Forward payment: schedule date error	Cannot schedule transaction at given date	Make sure that the schedule date is in the correct format: ddmmyy.
101621	Forward payment: schedule date in the past.	Cannot schedule transaction in the past	Make sure to provide the correct date.
101622	Forward payment: Schedule date too far ahead	Cannot schedule transaction too far ahead	Ensure that the schedule date is less than one year from the current date.
101639	There was no account found with the provided source MSISDN	Source account of transaction not found.	Verify that the source MSISDN is correct. If it is, ensure that the user is registered in the system.
101640	There was no account found with the provided target MSISDN	Target account of transaction not found.	Verify that the target MSISDN is correct. If it is, ensure that the target MSISDN is registered in the system.
101641	Source account is not configured with an account integration ID.	Source account of transaction does not have integration ID.	Update the source account with a proper account integration ID.
101642	Target account is not configured with an account integration ID.	Target account of transaction does not have integration ID.	Update the target account with a proper account integration ID.

101643	Invalid quantity amount was provided.	Invalid payment amount provided for transaction.	Ensure that the payment amount is a number.
101647	Encryption error	Encryption error	Investigate the log files.
101675	Invalid sender account.	Invalid sender account is provided during IMT quote.	Provide the correct sender account and try again
101676	Invalid beneficiary account.	Invalid beneficiary account is provided during IMT quote.	Provide the correct beneficiary account and try again.
101685	Sender account not found.	Sender account not found during IMT transaction.	Provide the correct sender account and try again.
101686	Beneficiary not found.	Beneficiary account not found during IMT transaction.	Provide the correct beneficiary account and try again.
101688	Sender account not active.	Sender account is not active for the IMT transaction.	Contact support.
101690	International remittance transaction declined.	Either sender or Beneficiary AML status is black or grey.	Contact support.
101691	Beneficiary not active.	Beneficiary account is not active for the IMT transaction.	Verify that the beneficiary account information is correct and try again.
101700	Request Not Found	Sales Request ID provided in request is not found	Verify that the request id is correct and try again
101701	Request Status is Invalid	Sales Request you are trying to execute is not in the RQS State	Ensure your request id has a status of RQS
101702	Request Brand Not Found	Sales Request Brand ID in the request is invalid	Ensure there is a brand that
101703	Service Category Not Found	Sales Request Service Category is not found	Ensure the Service exist from the brand id in the sales request
101704	Invalid Number of Params	Number of params need to match function specs	
101705	Invalid Param	Param 2 value must be either APPROVE or REJECT	

99999	International Remittance & Sales Request Exec: transaction is submitted.	Transaction is in submitted state.	The client is to check the transaction status later.
-------	--	---------------------------------------	---