GAPS INTEGRATION REQUIREMENTS

Creation / Revision History

Version	Summary Of Changes	Created/ Changed By	Date
1.0.1	Document created	Eunice Bolu-Martins	29/08/2024
1.0.2	Bulk Transfers Account statement	Eunice Bolu-Martins	20/02/2025
1.0.3	Get Account in Other Bank	Eunice Bolu-Martins	23/05/2025

Table of Contents

1. Introduction

GTBank Automated Payment System (GAPS) is an internet banking solution which facilitates easy administration of vendor, supplier and staff salary payments in batches or single transactions. It accepts generated payment schedules via either direct integration, SFTP or manual upload directly on the platform. This document provides the details for a direct integration between the customer's ERP and the GAPS platform. Kindly note that customer must be profiled on the GAPS platform to use this service.

GAPS webservice can be accessed as follows:

Test URL: **https://gtweb6.gtbank.com/GSTPS/GAPS_FileUploader/FileUploader.asmx**

Test credentials: Please contact gapssupport@gtbank.com for test credentials.

Live URL:

Live credentials: To be obtained upon being profiled on the GAPS platform

Furthermore, GAPS web application can be accessed as follows:

Test URL: http://gtweb.gtbank.com/GAPSNew/

Live URL: https://ibank.gtbank.com/GAPSNew/Alert.aspx

2. GAPS Transfer Payment Web Service

The GAPS transfer payment web service is used to process payments in single or bulk transactions. It accepts a payment instruction via an xml string and returns a string value indicating the status of the upload. There are several web methods implemented. The customer is to select the web method that best suits their operations. The web methods are described as follows:

2.1 Web Service Description

2.1.1 SingleTransfers_Enc

This method enables the customer initiate single payments by passing an XML string containing the transaction details. This method is designed for straight-through payment of single transfers and will not accommodate multiple/batch transfers.

Function Name: SingleTransfers_Enc

Request Parameters:

Parameter	M/O	Data Type	Description	Encrypted (Y/N)
xmlRequest	M	xmlstring	This contains the transaction details and hash generated in xml format. The transaction details consist of the following tags: • **amount**: This is the payment amount that will be encrypted using the public key given to the customer. • **paymentdate**: This is in yyyy-mm-dd format. • **reference**: This is the unique transaction reference. (***NB: Please use the same reference when retrying a transaction***) • **remarks**: This is the transaction remarks. • **vendorcode**: This is the unique identifier for the beneficiary. Assigned by the customer. • **vendorname**: This is the name of the beneficiary. • **vendoracctnumber**: This is the encrypted NUBAN account number of the beneficiary using the public key that will be given to the customer. • **vendorbankcode**: This is the 9-digit sort code of the beneficiary bank. The banks' HeadOffice code is sufficient to transfer to any branch account (see Appendix for list of bank sort codes) • **customeracctnumber**: This is the encrypted NUBAN account to debit for the transaction using the public key that will be given to the customer.	Y for only the following tags: • Amount • Vendor account number • Customer account number
accesscode	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
username	M	string	The uploader's username. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
password	M	string	The uploader's password. This should be encrypted with RSA algorithm using a	Y

			public key that will be given to the customer.	
channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request:

```
<SingleTransferRequest>
    <transdetails>xmlstring</transdetails>
    <accesscode>encryptedstring</accesscode>
    <username>encryptedstring</username>
    <password>encryptedstring</password>
    <channel>string</channel>
</SingleTransferRequest>
```

Sample xmlstring for the *transdetails*

```
<transaction><amount>kc+MDYquTSuhHTsdlZw9WYck8g87lFowPcHeJoF0p5/J623thap7B
jLY4kxIKZHruCpRBbrp07dOFvWoSpyMi2LfDfGrBezD1yJ+8kLN7QEcpsOXRCIqz/gMdrH9MMG
NraJkNQ/Klye6X7ABhMTRKvaTOPoEnjZvsX5ap0cvN14=</amount>
<paymentdate>2024-08-20</paymentdate><reference>000001</reference>
<remarks>Test20240713</remarks><vendorcode>25437</vendorcode>
<vendorname>FumUdorah</vendorname><vendoracctnumber>e0K2hZUcA3M0sK9DcUrLhv
B/tspPVu5XOf1dLWDFPVapRhV/l4Xti0ntrXifw1gjgKcL9QOSX3oMLUCmaGUipKSTu8s9nFLO
4lGKAJDjFarGUMVZoLkRsO7L2/tYzW8HGHG5jTU4Cu4L36JuYZA
WXXUElBJdXn7cjrSwmZCzR5s=</vendoracctnumber><vendorbankcode>058152052</ven
dorbankcode><customeracctnumber>Xl7yKm/z1UgmjtpyLU0vnrnt7+P76o7zyu9roUlhS8
WGHJBpA5lVJFfY1ulbf+ZjBvyXmKpoh2Cwk/E/O8P/bf/D5WcvQj7BZu7DY/g7vrVZ7nAFcWi5
F1tnyornvJcG1Qd7mzFZ65rkKzRxJA8ByWrD8SPEhr+ap0toahaKtYY=</customeracctnumb
er>
</transaction>
```

To call the webmethod, the xmlstring should be converted to string format before being passed to the *method* as illustrated below:

```
<xmlRequest>
<transdetails>&lt;transaction&gt;&lt;amount&gt;kc
+MDYquTSuhHTsdlZw9WYck8g87lFowPcHeJoF0p5/J623thap7BjLY4kxIKZHruCpRBbrp07dO
FvWoSpyMi2LfDfGrBezD1yJ
+8kLN7QEcpsOXRCIqz/gMdrH9MMGNraJkNQ/Klye6X7ABhMTRKvaTOPoEnjZvsX5ap0cvN14=&
lt;/amount&gt;&lt;paymentdate&gt;2024-08-
20&lt;/paymentdate&gt;&lt;reference&gt;Test20240713&lt;/reference&gt;&lt;r
emarks&gt;TEST&lt;/remarks&gt;&lt;vendorcode&gt;12345&lt;/vendorcode&gt;&l
t;vendorname&gt;Fum
Udorah&lt;/vendorname&gt;&lt;vendoracctnumber&gt;e0K2hZUcA3M0sK9DcUrLhvB/t
spPVu5XOf1dLWDFPVapRhV/l4Xti0ntrXifw1gjgKcL9QOSX3oMLUCmaGUipKSTu8s9nFLO4lG
KAJDjFarGUMVZoLkRsO7L2/tYzW8HGHG5jTU4Cu4L36JuYZA
WXXUElBJdXn7cjrSwmZCzR5s=&lt;/vendoracctnumber&gt;&lt;vendorbankcode&gt;05
8152052&lt;/vendorbankcode&gt;&lt;customeracctnumber&gt;Xl7yKm/z1UgmjtpyLU
```

0vnrnt7+P76o7zyu9roUlhS8WGHJBpA5lVJFfY1ulbf+ZjBvyXmKpoh2Cwk/E/O8P/bf/D5Wcv
Qj7BZu7DY/g7vrVZ7nAFcWi5F1tnyornvJcG1Qd7mzFZ65rkKzRxJA8ByWrD8SPEhr+ap0toah
aKtYY=</customeracctnumber>
</transaction></transdetails>
</xmlRequest>

<username>
AqlSOuvMk9Wd6IRNluVZLCXOHa4KkqonESzjKA+Reon6dSTz7r1cpNTAn49E/ljxMVmXHS0PDE
Ew+yV0aNxGiV0lP2YSiGfbXdqvFfVQweqWi65NnXqCgEdY2AMVawHtHDTfJWvGHlBgm8LJoCK/
XRddiECl+iKf0BizYGzWA68=
<username>

<accesscode>
L3Xo8SafaeTguaSjcnXvcW1Rd1lJ3jauCmYSU5boFtTAovP+a2nYd0nfWyEfv/z8UBiIsNheLq
GY6cpJIQuAt2yBjw5w9g4nnmFefsSC55UWNGRXyNHGgOcqFjtTg854lT5S9+eb1r49h29MLhos
x45m9SwfeiDJdCB5FtPA9oI=
</accesscode>

<password>
bxAozk7zscXznFkGF1pa1BijC8WZ6ozt9bN6hhQB6kjT+kZcpgwlH5Y/J7SBHH/eDVoxw5hMor
8Zmr0yCuO7zJhNomi7yyYFlVgNkJpUKZEhPeU/BLFPi2Usn2nsYEcTTbSPFw6/Bl18px91nL9A
gPUZVswakbsS4Qa38jMaSno=
</password>

<channel>GSTP</channel>

Sample Response:

The result string will contain a response code specifying whether the upload was successful or not.

```
<SingleTransferResponse>
      <Response>xmlstring</Response>
</SingleTransferResponse>
```

Where, xmlstring consists of response code and description.

Response Codes and Descriptions

Response Code	Description	Status	Scenario
1000	Transaction Successful	Success	Customer Paid
Resolution for Error Codes			
Response Code	**Description**	**Resolution**	**Status**	**Scenario**

1100	Transaction is being processed	Requery transaction to check status	Pending	Transaction is still being processed.
1001	Invalid Username / Password	Reset password on GAPS portal	Failed	Customer Not Paid
1002	Access disabled or not activated	Contact your account officer	Failed	Customer Not Paid
1003	Access to the system is locked	Reset password on GAPS portal	Failed	Customer Not Paid
1004	Password expired	Reset password on GAPS portal	Failed	Customer Not Paid
1005	Invalid encrypted value	Reconfirm and encrypt	Failed	Customer Not Paid
1006	Invalid xml format in transaction details	Recheck xml	Failed	Customer Not Paid
1007	Transaction validation error	Please retry	Pending	Customer Not Paid
1008	System error	Please retry	Pending	Customer Not Paid
1010	Failed. Only single transaction allowed	Utilize BulkTransfer method for bulk transactions	Failed	Customer Not Paid

2.1.2 Bulk Transfers

This method enables the customer initiate multiple/batch transfers by passing an XML string containing the transaction details. This method is designed to accommodate payment in bulk or where the transactions can be routed for approvals.

N.B: Processing of these transaction types are not immediate.

Function Name: BulkTransfers_Enc

Request Parameters:

Parameter	M/O	Data Type	Description	Encrypted (Y/N)
xmlRequest	M	xmlstring	This contains the transaction details and hash generated in xml format. The transaction details consist of the following tags: • **amount**: This is the payment amount that will be encrypted using the public key given to the customer. • **paymentdate**: This is in yyyy-mm-dd format. • **reference**: This is the unique transaction reference. (***NB: Please use the same reference when retrying a transaction***) • **remarks**: This is the transaction remarks. • **vendorcode**: This is the unique identifier for the beneficiary. Assigned by the customer. • **vendorname**: This is the name of the beneficiary. • **vendoracctnumber**: This is the encrypted NUBAN account number of the beneficiary using the public key that will be given to the customer. • **vendorbankcode**: This is the 9-digit sort code of the beneficiary bank. The banks' HeadOffice code is sufficient to transfer to any branch account (see Appendix for list of bank sort codes) • **customeracctnumber**: This is the encrypted NUBAN account to debit for the transaction using the public key that will be given to the customer.	Y for only the following tags: • Amount • Vendor account number • Customer account number
accesscode	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
username	M	string	The uploader's username. This should be encrypted with RSA algorithm using a	Y

			public key that will be given to the customer.	
password	M	string	The uploader's password. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request:

```
<BulkTransfers_Enc>
    <transdetails>xmlstring</transdetails>
    <username>encryptedstring</username>
    <accesscode>encryptedstring</accesscode>
    <password>encryptedstring</password>
    <channel>string</channel>
</ BulkTransfers_Enc>
```

Sample xmlstring for the *transdetails*

```
<transactions>
      <transaction>
            <amount>kc+MDYquTSuhHTsdlZw9WYck8g87lFowPcHeJoF0p5/J623thap7Bj
      LY4kxIKZHruCpRBbrp07dOFvWoSpyMi2LfDfGrBezD1yJ+8kLN7QEcpsOXRCIqz/gMdr
      H9MMGNraJkNQ/Klye6X7ABhMTRKvaTOPoEnjZvsX5ap0cvN14=</amount>
            <paymentdate>2024-08-20</paymentdate>
            <reference>000001</reference>
            <remarks>Test20240713</remarks>
            <vendorcode>25437</vendorcode>
            <vendorname>FumUdorah</vendorname>
            <vendoracctnumber>e0K2hZUcA3M0sK9DcUrLhvB/tspPVu5XOf1dLWDFPVap
      RhV/l4Xti0ntrXifw1gjgKcL9QOSX3oMLUCmaGUipKSTu8s9nFLO4lGKAJDjFarGUMVZ
      oLkRsO7L2/tYzW8HGHG5jTU4Cu4L36JuYZA
      WXXUElBJdXn7cjrSwmZCzR5s=</vendoracctnumber>
            <vendorbankcode>058152052</vendorbankcode>
            <customeracctnumber>Xl7yKm/z1UgmjtpyLU0vnrnt7+P76o7zyu9roUlhS8
      WGHJBpA5lVJFfY1ulbf+ZjBvyXmKpoh2Cwk/E/O8P/bf/D5WcvQj7BZu7DY/g7vrVZ7n
      AFcWi5F1tnyornvJcG1Qd7mzFZ65rkKzRxJA8ByWrD8SPEhr+ap0toahaKtYY=</cust
      omeracctnumber>
      </transaction>
      <transaction>
            <amount>kc+MDYquTSuhHTsdlZw9WYck8g87lFowPcHeJoF0p5/J623thap7Bj
      LY4kxIKZHruCpRBbrp07dOFvWoSpyMi2LfDfGrBezD1yJ+8kLN7QEcpsOXRCIqz/gMdr
      H9MMGNraJkNQ/Klye6X7ABhMTRKvaTOPoEnjZvsX5ap0cvN14=</amount>
            <paymentdate>2024-08-20</paymentdate>
            <reference>000001</reference>
            <remarks>Test20240713</remarks>
            <vendorcode>25437</vendorcode>
            <vendorname>FumUdorah</vendorname>
```

```
            <vendoracctnumber>e0K2hZUcA3M0sK9DcUrLhvB/tspPVu5XOf1dLWDFPVap
      RhV/l4Xti0ntrXifw1gjgKcL9QOSX3oMLUCmaGUipKSTu8s9nFLO4lGKAJDjFarGUMVZ
      oLkRsO7L2/tYzW8HGHG5jTU4Cu4L36JuYZA
      WXXUElBJdXn7cjrSwmZCzR5s=</vendoracctnumber>
            <vendorbankcode>058152052</vendorbankcode>
            <customeracctnumber>Xl7yKm/z1UgmjtpyLU0vnrnt7+P76o7zyu9roUlhS8
      WGHJBpA5lVJFfY1ulbf+ZjBvyXmKpoh2Cwk/E/O8P/bf/D5WcvQj7BZu7DY/g7vrVZ7n
      AFcWi5F1tnyornvJcG1Qd7mzFZ65rkKzRxJA8ByWrD8SPEhr+ap0toahaKtYY=</cust
      omeracctnumber>
      </transaction>
<transactions>
```

To call the webmethod, the xmlstring should be converted to string format before being passed to the *method* as illustrated below:

```
<xmlRequest>
<transdetails>&lt;transactions&gt;&lt;transaction&gt;&lt;amount&gt;IIs9kWn
+cv+5u9xntH50/YafVAj8fig8QfCt0zgrLMcsq5FovF66jT81I4UCsSXGWyYTrr96vB3PNRs9x
64B6MecWWPp2ZZ4cXfR8MxiTIrP5bb1zY1al26s8gVh10NruDsFKf11HgWqRdp4eGprhwZWAPJ
LdsfybHz+JGifHC0=&lt;/amount&gt;&lt;paymentdate&gt;2025-02-
26&lt;/paymentdate&gt;&lt;reference&gt;Test20250216&lt;/reference&gt;&lt;r
emarks&gt;TESTS&lt;/remarks&gt;&lt;vendorcode&gt;12345&lt;/vendorcode&gt;&
lt;vendorname&gt;Fum
Udorah&lt;/vendorname&gt;&lt;vendoracctnumber&gt;ITBXHkkyg2O0g/fr/qphReyXc
ukPd88GEsDYr3xSKavwpBVNWJ30RwmdGQbbOUNowKxrGNZOuQmDdt0/yr+2OFbIoLz+f0lwK7Y
Ek15R5pjE7GnUxYwE4aftrwsfwkydRYrXEs5ZD007PIrP2zzrb6CM5hgtg5ICEFUgIF8DR5s=&
lt;/vendoracctnumber&gt;&lt;vendorbankcode&gt;058152052&lt;/vendorbankcode
&gt;&lt;customeracctnumber&gt;bMtBIYrMu0QHmf3P7XHcEAWL/FZXP7sGMu6fOwtTXfC9
cisjo2Y5pBdbQd1hNa0KpcxO5Z/PacY0ZKQBjbv3BqOWSZ9+YdWLFzW/TnwfnpU9a5gta4FYGx
aKu0XhMn0e0K5FOiNYoqbg/gXC1mFsX4xEKZOMcDwZg66NanzzYlY=&lt;/customeracctnum
ber&gt;&lt;/transaction&gt;&lt;transaction&gt;&lt;amount&gt;IBEOdDOVUiy8rv
iGM5OLgNUtFfUiGYzW3xsVyHldYowPPFNiq+MKz4YoJJWLl+FOsEzSY/jRxgqQN2H2tnnTGoZW
i8Yq+9xxmgbAeXxUb5/FXWP8Er4PYoC7cuQv6f42xtzjia6h2L2VpGOqNCsVEPVBFX6N9VizOS
vs9GNl66Y=&lt;/amount&gt;&lt;paymentdate&gt;2025-02-
26&lt;/paymentdate&gt;&lt;reference&gt;Test20250217&lt;/reference&gt;&lt;r
emarks&gt;TESTS&lt;/remarks&gt;&lt;vendorcode&gt;12345&lt;/vendorcode&gt;&
lt;vendorname&gt;Fum
Udorah&lt;/vendorname&gt;&lt;vendoracctnumber&gt;GzKly46eE6NxpdO2AUBuZrbdt
wDeRXe2plqDfcCw9hXSgYwCBiwVqrAxQJj5pzcwK8a7NMBaFlJ/hkE1EG+jDwVooiax+f62XgJ
otyIp+ccySE+xronaZHN3av73vRfcrcdpEJxOmCH/4iJ4bk8ebP+1Nz7yPt7Wtw2bqOSoQCc=&
lt;/vendoracctnumber&gt;&lt;vendorbankcode&gt;058152052&lt;/vendorbankcode
&gt;&lt;customeracctnumber&gt;bMtBIYrMu0QHmf3P7XHcEAWL/FZXP7sGMu6fOwtTXfC9
cisjo2Y5pBdbQd1hNa0KpcxO5Z/PacY0ZKQBjbv3BqOWSZ9+YdWLFzW/TnwfnpU9a5gta4FYGx
aKu0XhMn0e0K5FOiNYoqbg/gXC1mFsX4xEKZOMcDwZg66NanzzYlY=&lt;/customeracctnum
ber&gt;&lt;/transaction&gt;&lt;/transactions&gt;&</transdetails>
</xmlRequest>

<username>
AqlSOuvMk9Wd6IRNluVZLCXOHa4KkqonESzjKA+Reon6dSTz7r1cpNTAn49E/ljxMVmXHS0PDE
Ew+yV0aNxGiV0lP2YSiGfbXdqvFfVQweqWi65NnXqCgEdY2AMVawHtHDTfJWvGHlBgm8LJoCK/
XRddiECl+iKf0BizYGzWA68=
<username>
```

```
<accesscode>
L3Xo8SafaeTguaSjcnXvcW1Rd1lJ3jauCmYSU5boFtTAovP+a2nYd0nfWyEfv/z8UBiIsNheLq
GY6cpJIQuAt2yBjw5w9g4nnmFefsSC55UWNGRXyNHGgOcqFjtTg854lT5S9+eb1r49h29MLhos
x45m9SwfeiDJdCB5FtPA9oI=
</accesscode>

<password>
bxAozk7zscXznFkGF1pa1BijC8WZ6ozt9bN6hhQB6kjT+kZcpgwlH5Y/J7SBHH/eDVoxw5hMor
8Zmr0yCuO7zJhNomi7yyYFlVgNkJpUKZEhPeU/BLFPi2Usn2nsYEcTTbSPFw6/Bl18px91nL9A
gPUZVswakbsS4Qa38jMaSno=
</password>

<channel>GSTP</channel>
```

Sample Response:

The result string will contain a response code specifying whether the upload was successful or not.

```
<BulkTransfersResponse>
      <Response>xmlstring</Response>
</BulkTransfersResponse>
```

Where, xmlstring consists of response code and description.

Response Codes and Descriptions

Response Code	Description	Status	Scenario
1000	Transaction Successful	Success	Customer Paid
Resolution for Error Codes			
Response Code	**Description**	**Resolution**	**Status**	**Scenario**
1001	Invalid Username / Password	Reset password on GAPS portal	Failed	Customer Not Paid
1002	Access disabled or not activated Contact your account officer	Failed	Customer Not Paid	
1003	Access to the system is locked	Reset password on GAPS portal	Failed	Customer Not Paid

1004	Password expired	Reset password on GAPS portal	Failed	Customer Not Paid
1005	Invalid encrypted value	Reconfirm and encrypt	Failed	Customer Not Paid
1006	Invalid xml format in transaction details	Recheck xml	Failed	Customer Not Paid
1007	Transaction validation error	Please retry	Pending	Customer Not Paid
1008	System error	Please retry	Pending	Customer Not Paid

3. Transaction Re-Query Web Service

Users can re-query for the status of a transaction after successfully uploading it to GAPS. There are two web methods implemented for this. The customer is to select the web method that best suits their operations. The web methods are described as follows:

3.1 Web Service Description

3.1.1 TransactionRequery_Enc

This method accepts encrypted request parameters and returns a string value indicating the status (code and message) of the transaction.

The web service is as described below:

Function Name: TransactionRequery_Enc

Request Parameters:

Parameter	M/O	Data Type	Description	Encrypted (Y/N)
transref	M	string	The reference of the transaction to be re-queried.	N
customerid	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y

username	M	string	The uploader's username. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
password	M	string	The uploader's password. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request:

```
<TransactionRequeryRequest>
      <transref>xmlstring</transref>
      <customerid>encrypted string</customerid>
      <username>encryptedstring</username>
      <password>encryptedstring</password>
      <channel>string</string>
</TransactionRequeryRequest>
```

To call the webmethod, the xmlstring should be converted to string format before being passed to the *method* as illustrated below:

```
<xmlstring>&lt;TransactionRequeryRequest&gt;&lt;TransRef&gt;25235954598&lt
;/TransRef&gt;&lt;/TransactionRequeryRequest&gt;</xmlstring>
<customerid>L3Xo8SafaeTguaSjcnXvcW1Rd1lJ3jauCmYSU5boFtTAovP
+a2nYd0nfWyEfv/z8UBiIsNheLqGY6cpJIQuAt2yBjw5w9g4nnmFefsSC55UWNGRXyNHGgOcqF
jtTg854lT5S9+eb1r49h29MLhosx45m9SwfeiDJdCB5FtPA9oI=</customerid>
<username>AqlSOuvMk9Wd6IRNluVZLCXOHa4KkqonESzjKA+Reon6dSTz7r1cpNTAn49E/ljx
MVmXHS0PDEEw+yV0aNxGiV0lP2YSiGfbXdqvFfVQweqWi65NnXqCgEdY2AMVawHtHDTfJWvGHl
Bgm8LJoCK/XRddiECl+iKf0BizYGzWA68=</username>
<password>bxAozk7zscXznFkGF1pa1BijC8WZ6ozt9bN6hhQB6kjT+kZcpgwlH5Y/J7SBHH/e
DVoxw5hMor8Zmr0yCuO7zJhNomi7yyYFlVgNkJpUKZEhPeU/BLFPi2Usn2nsYEcTTbSPFw6/Bl
18px91nL9AgPUZVswakbsS4Qa38jMaSno=</password>

<channel>GSTP</channel>
```

Sample Response:

```
<TransactionRequeryResponse>
      <Response>xmlstring</Response>
</TransactionRequeryResponse>
```

Response Codes and Descriptions

Response Code	Description	Status	Scenario
1000	Transaction Successful	Success	Customer Paid

Resolution for Error Codes				
Response Code	**Description**	**Resolution**	**Status**	**Scenario**
1001	Invalid Username / Password	Reset password on GAPS portal	Failed	Customer Not Paid
1002	Access disabled or not activated	Contact your account officer	Failed	Customer Not Paid
1003	Access to the system is locked	Reset password on GAPS portal	Failed	Customer Not Paid
1004	Password expired	Reset password on GAPS portal	Failed	Customer Not Paid
1005	Transaction not found	Re-initiate payment	Failed	Customer Not Paid
1006	Transaction be waiting to processed	Requery transaction to check status	Pending	Transaction is still being processed. *Kindly utilize resolution action*
1007	Transaction for waiting approval	Requery transaction to check status	Pending	Transaction is still being processed. *Kindly utilize resolution action*

1008	Error requerying transaction	Retry requery	Pending	Transaction is still being processed. *Kindly utilize resolution action*
1010	Transaction failed	Re-initiate payment	Failed	Customer Not Paid

4. Account Validation Web service

The account validation web method would on receipt of request, validate the account number and other input parameters and return a valid account name for valid input parameters passed.

4.1 Account Validation for GTB Web Service Description

Function Name: GetAccountInGTB

Request Parameters:

Field	M/O	Data Type	Description	Encrypted (Y/N)
customerid	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer	Y
username	M	string	The username assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
password	M	string	The password assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
AccountNumber	M	string	The account number to be validated. This should be encrypted with RSA algorithm	Y

			using a public key that will be given to the customer.	
Channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request:

```
<GetAccountInGTB_Enc>
      <accountNo>encrypted string</accountNo>
      <customerid>encrypted string</customerid>
      <username>encrypted string</username>
      <password>encrypted string</password>
      <channel>string</channel>
</GetAccountInGTB_Enc>
```

Sample Response:

```
<GetAccountInGTB_EncResponse>
       <response>
            <CODE>1000</CODE>
            <accountname>COSMIC INTELLIGENT LABS LTD</accountname>
            <CURRENCYCODE>NGN</CURRENCYCODE>
            <CURRENCYDESC>Naira</CURRENCYDESC>
       </response>
</GetAccountInGTB_EncResponse>
```

Response Codes and Descriptions:

Response Code	Description
1000	Successful Validation

Resolution for Error Codes

Response Code	Description	Resolution
1001	Invalid Nuban number	Account number should be NUBAN account number (10 digits length)

4.2 Account Validation for GTB Web Service Description

Function Name: GetAccountInOtherBank

Request Parameters:

Field	M/O	Data Type	Description	Encrypted (Y/N)
customerid	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer	Y
username	M	string	The username assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
password	M	string	The password assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
AccountNumber	M	string	The account number to be validated. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
Bank Code	M	String	The Bank code of the bank where the account resides	N
Channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request:

```
<GetAccountInOtherBank_Enc>
      <accountNo>encrypted string</accountNo>
      <bankcode> string</bankcode>
      <customerid>encrypted string</customerid>
      <username>encrypted string</username>
      <password>encrypted string</password>
      <channel>string</channel>
</GetAccountInOtherBank_Enc>
```

Sample Response:

```
<GetAccountInOtherBank_EncResponse>
       <response>
            <CODE>1000</CODE>
            <accountname>NIGEL INTELLIGENCE LTD</accountname>
       </response>
</GetAccountInOtherBank_EncResponse>
```

Response Codes and Descriptions:

Response Code	Description
1000	Successful Validation
Resolution for Error Codes	

Response Code	Description	Resolution
1001	Invalid Nuban number	Account number should be NUBAN account number (10 digits length)

5. Account Statement Retrieval Web Service

The account statement retrieval web method would on receipt of the request generate and return the customer's statement, based on request parameters.

5.1 Web Service Description

Function Name: AccountStatement_XML_Enc

Request Parameters:

Field	M/O	Data Type	Description	Encrypted (Y/N)
startDate	M	datetime	The account statement start date in format YYYY/MM/DD	N
endDate	M	datetime	The account statement end date in format YYYY/MM/DD	N

pageNumber	O	integer	Page number to spool. This is an optional parameter that should be specified only if customer desires a multipage account statement	N
pageSize	O	integer	Total count of transactions required per page. This is an optional parameter that should be specified only if customer desires a multipage account statement	N
customerid	M	string	GAPS Access Code of the customer. This should be encrypted with RSA algorithm using a public key that will be given to the customer	Y
username	M	string	The username assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
password	M	string	The password assigned to the user. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
AccountNumber	M	string	The account number of the account whose statement should be retrieved. This should be encrypted with RSA algorithm using a public key that will be given to the customer.	Y
Channel	M	string	The "channel" that will be given to the customer. This should *not* be encrypted.	N

Sample Request where pageNumber and pageSize are specified:

```
<AccountStatementRetrievalRequest>
```

```
        <startDate>2023/05/23</startDate>
        <endDate>2023/05/23</endDate>
        <pageNumber>1</pageNumber>
        <pageSize>100</pageSize>
        <Customerid>encrypted string</Customerid>
        <username>encrypted string</username>
        <password>encrypted string</password>
        <accountnumber>encrypted string</accountnumber>
        <channel>string</channel>
</AccountStatementRetrievalRequest>
```

Sample Request where pageNumber and pageSize are not specified:

```
<AccountStatementRetrievalRequest>
        <startDate>2023/05/23</startDate>
        <endDate>2023/05/25</endDate>
        <Customerid>encrypted string</Customerid>
        <username>encrypted string</username>
        <password>encrypted string</password>
        <accountnumber>encrypted string</accountnumber>
        <channel>string</channel >
</AccountStatementRetrievalRequest>
```

Response Parameters:

Parameter	Description
responseCode	Response code
responseDesc	A description of the response code
totalCount	Count of transactions on the current page (for multipage account statements)
pageSize	Maximum count of transactions to be contained in a page (for multipage account statements)
currentPage	Current page number (for multipage account statements)
totalPages	Total number of pages (for multipage account statements)
previousPage	Flag indicating whether there is a previous page (for multipage account statements)
nextPage	Flag indicating whether there is a next page (for multipage account statements)
tra_date	The transaction date i.e. tra_date on the transact table
val_date	The value date i.e. val_date on the transact table

debit	Debit amount
credit	Credit amount
balance	The balance after the transaction
remarks	Transaction narration
reference	Transaction reference
Encryption	encryption of all sensitive parameters

Sample Response:

```
<AccountStatementRetrievalResponse>
	<responseCode>1000</responseCode>
	<responseDesc>Successful</responseDesc>
	<Pagination>
		<totalCount>2</totalCount>
		<pageSize>100</pageSize>
		<currentPage>1</currentPage>
		<totalPages>1</totalPages>
		<previousPage>No</previousPage>
		<nextPage>No</nextPage>
	</Pagination>
	<Transactions>
		<Transaction>
			<tra_date>2023-05-23</tra_date >
			<val_date>2023-06-13</val_date>
			<debit>0</debit>
			<credit>1000</credit>
			<balance>456,000</balance>
			<remarks>Cash Deposit from Austin Osega</remarks>
			<reference>01554666464666</reference>
		</Transaction>
		<Transaction>
			<tra_date>2023-05-23</tra_date>
			<val_date>2023-06-13</val_date>
			<debit>4000</debit >
			<credit>0</credit >
			<balance>400,000</ balance>
			<remarks>transfer via GAPS from 011454973</remarks>
			<reference>01554666464667</reference>
		</Transaction>
	</Transactions>
	<Channel>hdhid87902j20092sjjso2</Channel>
</AccountStatementRetrievalResponse>
```

Sample Response:

```
<AccountStatementRetrievalResponse>
	<responseCode>1000</responseCode>
	<responseDesc>Successful</responseDesc>
	<Transactions>
		<Transaction>
			<tra_date>2023-05-23</tra_date >
```

```
                <val_date>2023-06-13</val_date>
                <debit>0</debit>
                <credit>1000</credit>
                <balance>456,000</balance>
                <remarks>Cash Deposit from Austin Osega</remarks>
                <reference>01554666464666</reference>
            </Transaction>
            <Transaction>
                <tra_date>2023-05-23</tra_date>
                <val_date>2023-06-13</val_date>
                <debit>4000</debit >
                <credit>0</credit >
                <balance>400,000</ balance>
                <remarks>transfer via GAPS from 011454973</remarks>
                <reference>01554666464667</reference>
            </Transaction>
      </Transactions>

      <Channel>hdhid87902j20092sjjso2</Channel >
</AccountStatementRetrievalResponse>
```


Response Codes and Descriptions

Response Code	Description
1000	Successful

Resolution for Error Codes

Response Code	Description	Resolution
1001	Invalid Username or password	Reset password on GAPS portal
1002	Access disabled or not activated	Contact account officer
1003	Access to the system is locked	Reset password on GAPS portal
1004	Password Expired	Reset password on GAPS portal
1005	Account number less than 10 digits	Account number should be Nuban account number (10 digits length)

1006	Account number not numeric	Account number must be numeric, input your account with numeric values only
1007	Account number not profiled	Contact account officer to profile account number
1008	Unable to generate statement	Contact account officer to inform e-support team
1010	Invalid account number	Input valid Nuban account number (10 digits)
1011	Date greater than current date	Input date that is not greater than current date
1012	Start date cannot be greater than current date	Input date that is not greater than current date
1013	No record found based on the parameters supplied	Enter a different date range

Test Public Key:

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCrtPgIUBsQscypy+2A2l6oHKlLRTgD4hlrYKW9
IrAK4l10FPndJ3i57CioPalYKdNMF9+K4mFaGfT3dAMRSgWWWDeaerHx35VLgdX/wFTN5Zf1QYGe
WiKyAmCAXoPwtlfvlLqsr9NMBJ3Ua+fFqSC4/6ThhudMlrxNL/ut/kd+pQIDAQAB

Note: The production public key would be generated by the customer once profiled on the public key generation portal.

Appendix

SAMPLE BANK CODES

S/N	BANK NAME	VENDOR BANK CODE
1.	CENTRAL BANK OF NIGERIA	001
2.	FIRST BANK OF NIGERIA PLC	011
3.	NIGERIA INTERNATINAL BANK (CITIBANK)	023
4.	HERITAGE BANK	030
5.	UNION BANK OF NIGERIA PLC	032
6.	UNITED BANK FOR AFRICA PLC	033
7.	WEMA BANK PLC	035

8.	ACCESS BANK NIGERIA LTD	044
9.	ECOBANK NIGERIA PLC	050
10.	ZENITH INTERNATIONAL BANK LTD	057
11.	GUARANTY TRUST BANK PLC	058
12.	FBNQuest Merchant Bank Limited	060002
13.	DIAMOND BANK LTD	063
14.	STANDARD CHARTERED BANK NIGERIA LTD	068
15.	FIDELITY BANK PLC	070
16.	SKYE BANK PLC	076
17.	KEYSTONE BANK LTD	082
18.	IBILE MFB	090118
19.	HASAL MICROFINANCE BANK	090121
20.	SUNTRUST BANK	100
21.	PROVIDUS BANK	101
22.	FIRST CITY MONUMENT BANK	214
23.	UNITY BANK PLC	215
24.	STANBIC IBTC BANK PLC	221
25.	STERLING BANK PLC	232
26.	JAIZ BANK PLC	301
27.	PAGA	327
28.	RAND MERCHANT BANK	502
29.	PARALLEX MFB	526
30.	NPF Microfinance Bank	552
31.	CORONATION MERCHANT BANK	559

32.	Page MFBank	560
33.	New Prudential Bank	561
34.	FSDH MERCHANT BANK LIMIT	601
35.	FINATRUST MICROFINANCE BANK	608

SAMPLE LIST OF NIGERIAN BANK HEADOFFICE SORTCODES

S/N	BANK NAME	VENDOR BANK SORT CODE
1.	CENTRAL BANK OF NIGERIA	001080032
2.	FIRST BANK OF NIGERIA PLC	011151003
3.	NIGERIA INTERNATINAL BANK (CITIBANK)	023150005
4.	HERITAGE BANK	030150014
5.	UNION BANK OF NIGERIA PLC	032154568
6.	UNITED BANK FOR AFRICA PLC	033152048
7.	WEMA BANK PLC	035150103
8.	ACCESS BANK NIGERIA LTD	044150291
9.	ECOBANK NIGERIA PLC	050150010
10.	ZENITH INTERNATIONAL BANK LTD	057150013
11.	GUARANTY TRUST BANK PLC	058152052
12.	FBNQuest Merchant Bank Limited	060002600
13.	DIAMOND BANK LTD	063150162
14.	STANDARD CHARTERED BANK NIGERIA LTD	068150015
15.	FIDELITY BANK PLC	070150003
16.	SKYE BANK PLC	076151365
17.	KEYSTONE BANK LTD	082150017
18.	IBILE MFB	090185090

19.	HASAL MICROFINANCE BANK	090118509
20.	SUNTRUST BANK	100152049
21.	PROVIDUS BANK	101152019
22.	FIRST CITY MONUMENT BANK	214150018
23.	UNITY BANK PLC	215082334
24.	STANBIC IBTC BANK PLC	221159522
25.	STERLING BANK PLC	232150016
26.	JAIZ BANK PLC	301080020
27.	PAGA	327155327
28.	RAND MERCHANT BANK	502155502
29.	PARALLEX MFB	526155261
30.	NPF Microfinance Bank	552155552
31.	CORONATION MERCHANT BANK	559155591
32.	Page MFBank	560155560
33.	New Prudential Bank	561155561
34.	FSDH MERCHANT BANK LIMIT	601155601
35.	FINATRUST MICROFINANCE BANK	608155608