**MATRIX DESINGS / CIS**

**INTERPAY API TEST ENVIRONMENT FOR ECOBANK**

The purpose of this document is to provide a detailed documentation of the API that Ecobank will use to access the Interpay App. This document is directed to Ecobank IT staff only.

This this documentation is only for the testing environment only, if everything works out as expected during testing then we shall provide the live documentation API for Ecobank.

**Requirements:**

Before using this API there are some important settings and changes that need to be done. A web interface has been created to be used in setting up these parameters.

Go to <http://107.22.240.148/interpay_v2/bank/>

Login: ecobank\_admin

Password: admin

Note: Please note that only one account has been created for use, multiple accounts can be created after login. You must change the username and password after first login.

After you login, details of the menus on the left are as follows:

**Bank Branches:**

This page is used for setting up all the bank’s branches that Interpay will require. In creating a branch, always make sure you provide the branch code (branch\_code) as this will is what you will pass as an input parameter when calling the API methods.

**Users (Tellers):**

This page is used for setting up all the bank’s tellers per branch that Interpay will require. In creating a teller, always make sure you provide the username (teller\_id) as this will is what you will pass as an input parameter when calling the API methods.

For Ecobank even though you will be using an App to call the API, it is important to create users (tellers) for each branch. This will help in reporting purposes of knowing which teller made a credit transaction at which branch. By so doing passing the teller\_id and branch\_code to any API methods becomes easier.

**Configure API:**

This page is used for setting up Ecobank’s API credentials. Here you can change your APP ID, APP KEY and SALT value to any value you want. You can periodically be changing these settings as many times as you want. But as often as you change these settings, so must you update your third party app API credentials settings.

**API:**

The URL for accessing the API is <http://107.22.240.148/interpay_v2/ecobank>. To access a method (action) you construct the following URL <http://107.22.240.148/interpay_v2/ecobank>/the\_method\_name. You then POST the required input parameters and value to the methods URL. When you call a method and pass some parameters you will always expect a json format in the form:

{

"response\_code": value,

"message": value,

“data”: array

}

The “response\_code” is either 0 or 1 indicating success or failure respectfully. The “message” is the message for the success or failure. The message can be a single string or an array of strings. The “data” may or may not be present depending on the method called. Some method will return data others will not. The “data” is returned in an array format.

Details of the API methods and parameters are as follows: Parameters with \* are required to pass.

|  |  |  |  |
| --- | --- | --- | --- |
| **Method Name** | **Description** | **URL** | **Parameters and explanation** |
| search | This will pull a student record (Basic info, banks credit balances and last 5 credit transactions) with the last 5 credit transactions. | /search | \*APP\_ID = ‘APP\_ID value’  \*APP\_KEY =’APP\_KEY value’  \* index\_number =’Student Index number’  \* branch\_code =’ Any branch code’ |
| credit\_account | This is used in crediting the student interpay portal account when a deposit is made for a student. If success, It will return the deposit info which contains ‘trans\_ref\_no which may be of need in getting details if the transaction’ with an updated student wallet balance. | /credit\_account | \*APP\_ID = ‘APP\_ID value’  \*APP\_KEY =’APP\_KEY value’  \* index\_number =’Student Index number’  \* branch\_code =’ The branch code of the branch where the deposit was made’  \* teller\_id = ‘The username of the teller that made the deposit transaction’  \* amount = ‘Amount tendered. Acceptable: 2000 or 2000.00. Invalid : 2,000 or 2,000.00’  \* payee\_name = ‘Name of depositor’  \* payee\_phone =’Telephone of depositor’  payee\_address = ‘Address of depositor’  \* payment\_method = ‘Mode of payment. Pass either one of these (cash, cheque)’  payin\_cheque\_number = ‘Cheque Number’  payin\_cheque\_bank = ‘Issuing Bank for the cheque’  If the payment\_method is ‘cheque’, then payin\_cheque\_number and payin\_cheque\_bank will be required |
| get\_credit\_transactions | This is used to get all credit transactions. This can be filtered by the branch, teller. To get a specific transaction you pass trans\_ref\_no. You obtain trans\_ref\_no whenever you credit a student account.  You can also find transactions by a specific time period, by passing the start\_dt (Start date) and end\_dt (End date). The date format is yyyy-mm-dd. | /get\_credit\_transactions | \*APP\_ID = ‘APP\_ID value’  \*APP\_KEY =’APP\_KEY value’  filter\_teller\_id=’Filter for this teller. E.g I want transactions for this teller only’  filter\_branch\_code=’ Filter for this branch\_code. E.g. I want transactions for this branch\_code only ’  start\_dt= ‘Start date for querying transactions. Format : yyyy-mm-dd’  end\_dt= ‘End date for querying transactions. Format : yyyy-mm-dd’  Note: start\_dt and end\_dt will be required if trans\_ref\_no is not passed.  trans\_ref\_no = ‘This is a unique number that interpay generates to differentiate each transaction. If you want to get a particular transaction then pass this. You obtain trans\_ref\_no whenever you credit a student account’ |
| get\_fee\_payment\_transactions | When a student wallet has been credited, they go online to make payments (distribute the funds). Depending on where a fee type account is located a transaction will be created for that bank. E.g. If the student pays fees on the portal for tuition say GHC 80, and the tuition’s account number is at Ecobank, a transaction will be created for Ecobank stating that move an amount of GHC 60 from the transit account (account number of where all student wallet deposit are paid into) to the tuition’s account number.  Note that if the students have not made fee payments (funds distributions on the portal) you will not get any transactions to process. | /get\_fee\_payment\_transactions | \*APP\_ID = ‘APP\_ID value’  \*APP\_KEY =’APP\_KEY value’  \* branch\_code =’ The branch code of the branch where the fee payment transactions will be processed.’  \* teller\_id = ‘The username of the teller who will authorize this processing’ |

NOTE: In all API method calls, you have to always pass the APP\_ID and APP\_KEY as this will be used in the authentication process.

NOTE:

The method “/get\_fee\_payment\_transactions” may or may not return transactions students pay fees on the portal or does not pay any fee. In order to test this method you can use this student profile: login name: 05DBW145 password: 05DBW145.

First credit the student account using the “/credit\_account” method.

Go to <http://107.22.240.148/interpay_v2/student>.

Log in with the username and password provided above for the student profile, from the left menu click on Payment of Other Services.

From the right window, select any of the items from the dropdown and click on the “Add to Cart” button.

You can add as many items as you want. When done click on the “Bill Me Now” button (you will be prompted twice, just click on ok). You will then be redirected to the fee payment screen.

From the right window scroll to the bottom you should see bill record(s) for the item(s) you just purchased.

Select (Click on the select button next to the fee item) the item(s) that applies and click on the “Next” button. Click on “Complete Payment” button (You will be prompted, just click ok).

If you followed the above steps correctly then accessing “/get\_fee\_payment\_transactions” should now give you transactions to process. You can repeat the above process several times in order to get a large number of fee payment transactions to download.

To see examples of the response format, attached to the mail is a sample.rar file which contains basic setups and API access examples. You can use this guide.