ITC SYSTEMS
GLOBAL NIGERIA LIMITED

KAROTA Payment Collection Integration API Documentation

Version:	1.1.0
Release Date:	**1st July, *2021***

Quick Navigation

- 1.0 Introduction
- 2.0 Disclaimer
- 3.0 Authentication
- 4.0 Get Rider Information
- 5.0 Push Payment Notification
- 6.0 Fetch Tickets Price List
- Status Codes
- Legend

1.0 Introduction

The KAROTA Keke riders payment collection API is a RESTful web service which allows KIRS appointed and authorised merchants to integrate their platforms and perform any of the following tasks;

- Getting riders information
- Pushing payments notifications
- Getting items price list

2.0 Disclaimer

- The APIs have predictable resource-oriented URLs, and the system acknowledge only JSON-encoding as the data interchange format.
- It also uses standard HTTP response codes, authentication and verbs. More details on our HTTP codes have been outlined in the Error! Reference source not found. Section.
- Unless otherwise specified, all data types sent as a requests payload are of **String** type.

3.0 Authentication

The authentication API uses Json Web Token (JWT) based authentication system to authenticate all requests. This must be provided on every request.
The token will be generated and issued to all registered merchants using a combination of username and password issued to them.

Test mode secret key have the prefix **mer_demo_** while the live keys have the prefix **mer_prod_**

To generate your token for both the sandbox and live servers, please follow the guides below

URL

```
https://{BOX_NAME}.kigra.gov.ng:{BOX_PORT}/v1/merchant/auth/token/request
```

Request payload

```
{
  "username": "itc",
  "secret": "super-secret"
}
```

The description of each parameter is outlined below

Variable	Description
username	The Username provided to you
secret	The Secret key provided to you

Response payload

```
{
  "access-token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9…. ",
  "expiration": "40000"
}
```

The access-token is a JWT bearer token which must be sent in the authorization header **FOR EACH REQUEST** as follows

```
Authorization: Bearer {access_token}
```

***NOTE*: Please be sure to keep your API keys secure and NEVER use them in publicly accessible areas such as client-side code, html file, URL parameters, etc. And also be aware that ITC will never ask for such credentials.**

4.0 Get Rider Information

This endpoint will be used to fetch the information of the registered Keke rider using his registration number. This is described as below.

Request Method: GET

URL

```
https://{BOX_NAME}.kigra.gov.ng:{BOX_PORT}/merchant/rider/info/request/{REG_NUMBER}
```

The **REG_NUMBER** parameter is **required** and is the registration number of the Keke whose format is generally ***LGA-Year-SerialNum*** e.g **KMC-19-0001**, or **WRA-20-1129** etc

Success response payload

```
{
  "response": {
    "code": 200,
    "data": {
      "mobileNumber": "08037177513",
      "name": "MUKHTAR UMAR",
      "plateNo": "MGR053WH",
      "tricycleNo": "GWL-19-0001"
    },
    "description": "Record found",
    "status": "Success"
  }
}
```

The description of the request parameters is outlined below

Variable	Description
code	The HTTP status code for the request
data	An object containing the rider's information, which includes • **mobileNumber** – The phone number of the rider • **name** : The full name of the Keke driver • **plateNo** – The vehicle plate number registered with the Keke • **regNo** – The registration number of the Keke, issued by KAROTA ***Note: If no Rider was found with the associated REG_NUMBER, the data object will be omitted in the response***

`description`	A textual explanation of the success or failure of the record search. If the record is found, this parameter reads ***"Record Found",*** otherwise, it reads ***"No Record found"***
`status`	A textual explanation of the **code** explained above

When no rider was found with the **REG_NUMBER** sent in the request, a response is returned as below

```
{
  "response": {
    "code": 106,
    "description": "No Record found",
    "status": "Success"
  }
}
```

5.0 Push Payment Notification

This endpoint will be used to post payment information whenever a payment is made and is verified by the merchant

Request Method: POST

URL

```
https://{BOX_NAME}.kigra.gov.ng/merchant/rider/ticket/payment
```

Parameters

```
{
"tricycleNo": "KMC-19-0001",
"code": "1DY",
"transactionDate": "2021-09-23 12:04:23",
"paymentRef": "ABCD1234"
}
```

The description of the request parameters is outlined below

PARAMETER NAME	Description	STATUS
tricycleNo	The registration number of the Keke whose format is generally ***LGA-Year-SerialNum*** e.g **KMC-19-0001**, or **WRA-20-1129** etc	Required
paymentCode	The payment interval code as specified below ● **1DY** for daily ● **1WK** for weekly ● **2WK** for 2 weeks ● **1MM** for monthly ● **2MM** for 2 months ● **3MM** for 3months ● **4MM** for 4 months ● **6MM** for 6 months, and ● **1YY** for Annual	Required
transactionDate	The date when the transaction was carried out and **MUST** be in the format **yyyy-MM-dd hh:mm:ss**.	Required
paymentRef	A unique reference for this payment. If this reference already exists, a duplicate request status is returned .	Required

Sample Success Response

```
{
  "response": {
    "code": 200,
    "data": "OK",
    "description": "Successful",
    "status": "Success"
  }
}
```

Sample Duplicate Response

```
{
  "response": {
    "code": 205,
    "description": "Duplicate Request",
    "status": "DuplicateRequest"
  }
}
```

6.0 Fetch Tickets Price List

This endpoint will be used to get the list of payment items and their respective codes.

Request Method: GET

```
https://{BOX_NAME}.kigra.gov.ng/merchant/ticket/payment/list/request
```

Parameters

No parameters are required for this request

Response Payload

```
{
  "response": {
    "code": 200,
    "data": [
      {
        "amount": 100,
        "code": "DY",
        "name": "Daily"
      },
      {
        "amount": 1400,
        "code": "2WK",
        "name": "2-Weeks"
      },
    ],
    "description": "Ticket Price List",
    "status": "Success"
  }
}
```

The description of each parameters is outlined below

Variable	Description
code	The HTTP status code for the request
data	An array of the list of price list items each with the following fields; • **amount**: The amount to be paid for that item • **code**: The short code of the item • **name**: The name of the item
description	A textual explanation of the success or failure of the record search. If the record is found, this parameter reads ***"Ticket Price List",***
status	A textual explanation of the **code** explained above

7.0 Get Payment Status

This endpoint will be used to get the status of payment made.

Request Method: GET

URL:

```
https://{BOX_NAME}.kigra.gov.ng:{BOX_PORT}/merchant/ticket/payment/verification/request/{paymentReference}
```

Parameters

Variable	Description	Status
paymentReference	The unique reference of the payment	Required

Response Payload

```
{
  "response": {
    "code": 400,
    "data": {
      "paymentCode": "1DY",
      "paymentRef": "123456ACBD4",
      "transactionDate": "2021-09-23T12:33:05",
      "tricycleNo": "KMC-19-0001"
    },
    "description": "record found for 12-123456ACBD4",
    "status": "Success"
  }
}
```

The description of each parameters is outlined below

Variable	Description
code	The HTTP status code for the request
data	An object containing the payment fields as follows • paymentCode: The price code of the payment • paymentRef: The unique payment reference • transactionDate: The date when the transaction occurred • tricycleNo: The registration number of the Keke
description	A textual explanation of the success or failure of the record search. If the record is found, this parameter reads "record found for {transactionRef}"

status	A textual explanation of the **code** explained above

Sample Record Not Found Response

```
{
  "response": {
    "code": 106,
    "description": "No record found for `{paymentRef}`",
    "status": "Success"
  }
}
```

Status Codes

Code	Description
101	Queued
102	Pending Confirmation
103	Pending Validation
104	Validated
105	Booked
106	Not Found
107	Invalid Amount
200	Success
201	Not Supported
203	Application Error
204	Not Allowed
205	Duplicate Request
400	Failed
401	Throttled
402	Expired
500	Reversed

Legend

Variable	Value(s)
BOX_NAME	[sandbox]
BOX_PORT	[7000]