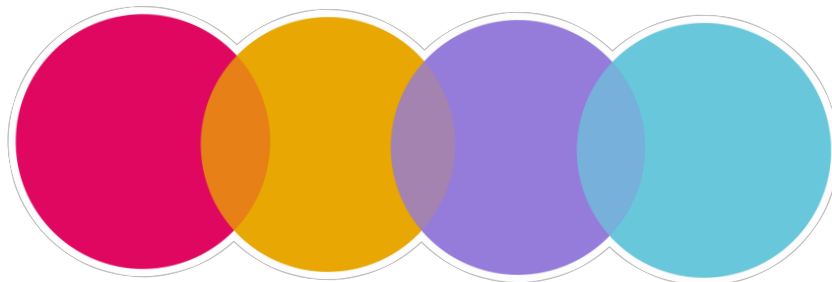


# Introduction to API Integration



**KingsPay Card**





# Introduction to API Integration

Version 1.3.0

January 3rd, 2019

For a Free Merchant Account,  
Signup @ [www.alphaplus.live](http://www.alphaplus.live)



## Table of Contents

Definition of Terms .....	4
Introduction.....	5
One-Off Payment .....	5
Recurrent Payment .....	9
Other APIs for Managing Recurrent Payment .....	12
Sandbox Environment & Test Cards .....	15

# API Integration

## Definition of Terms

Terms	Definitions
<b>Handshake</b>	<p>A permanent link created between a Merchant and KingsPay for an individual customer.</p> <p>A Handshake must exist before any transactions can be carried out for a customer via the API integration.</p>
<b>MiGs</b>	MasterCard Internet Gateway System



## Introduction

The following documentation provides an introductory guide to integrating with the KingsPay Payment Platform. The guide covers the basic services to fully process both single, straight payment transactions and recurring transactions.

## One-off Payment

Integrate KingsPay in these **3 simple steps**:

1. Add KingsPay JavaScript API in the script tag at the bottom of your webpage e.g. –

```
<script type="text/javascript"  
    src="https://api.alphapay.live/apv1/api/alphaPay">  
</script>
```

2. Call the magic '*getpaidSetup()*' function with JavaScript.
3. Verify payment in your callback page using KingsPay's HTTP based RESTful API.

```
'https://api.alphapay.live/apv1/api/verify'
```

## Steps Explained:

### Step #1

```
<script type="text/javascript"  
    src="https://api.alphapay.live/apv1/api/alphaPay">  
</script>
```

This API makes available the KingsPay's payment interface (modal view) on your website. Therefore, there would be no redirect, and all transactions are concluded on your website by just adding the script tag.



## Step #2

### *getpaidSetup()*

Calling this function with JavaScript launches the KingsPay's checkout form in modal view right on your website so your customers can make payment instantly.

All you need do is pass an **object** as the argument into the *getpaidSetup()* function. The following parameters should be included in the **object** passed:

Parameters	Required	Description
<b>PBFPubKey</b>	TRUE	This is the Merchant's PublicKey. This is a unique key generated for the Merchant at Signup. It can be found in the Merchant's Account Settings section. It starts with the prefix ' <b>FLWPUBK</b> ' and ends with the suffix ' <b>X</b> '.
<b>amount</b>	TRUE	This is the charge amount.
<b>customer_email</b>	TRUE	This is the customer's email address.
<b>txref</b>	TRUE	This is the unique reference; unique to the particular transaction being made. It is generated by the merchant for every transaction.
<b>currency</b>	False (Defaults to NGN)	This is the currency code. It represents the currency in which the customer will be charged.
<b>country</b>	False (Defaults to NG)	This is the pair country for the transaction with respect to the currency.

Parameters	Required	Description
<b>custom_title</b>	FALSE	This is a title name that would be displayed at the top of the KingsPay payment interface. It's usually the Merchant's Trade Name.
<b>custom_logo</b>	FALSE	This is the URL of Merchant's logo, to be displayed at the top left corner of the KingsPay checkout interface.
<b>onclose</b>	FALSE	This function contains the URL or the relative path to the intended landing page <b><u>when the customer clicks on the close button</u></b> on the KingsPay checkout interface.
<b>callback</b>	FALSE	This function contains the URL or the relative path to the intended landing page when transaction has been successfully completed.

### Example:

```
getpaidSetup({
  customer_email: 'user@alphapay.com',
  amount: 5000,
  txref: 'alpha-123456',
  PBFPubKey:
  'FLWPUBK-8u5r98f9629531bb0ba1j76fe24typ6e-X',
  currency: 'NGN',
  country: 'NG',
  custom_title: 'alphaPay',
  custom_logo: 'https://alphapay.live/logo.png',
  onclose: function() {
    window.open('index.php', '_self');
  },
  callback: function(response) {
    window.open('callbackPage.php', '_self');
  }
});
```

### Step #3

<https://api.alphapay.live/apv1/api/verify>

This API is used to verify the transaction using the txref value (unique reference) generated by the merchant and passed to the *getpaidSetup()* function.

The API is called using an **HTTP POST REQUEST** and passing the following parameters.

Parameter	Required	Description
<b>SECKEY</b>	TRUE	This is your secret key which is a unique key generated for you when you signup as a merchant. It starts with a prefix 'FLWSECK' and ends with suffix 'X'.
<b>tx_ref</b>	TRUE	This is the unique reference that was generated by the merchant for the transaction that the merchant wants to verify.

- The **Header** should have **Content-Type** = **application/json**.
- This API returns a JSON. The charge response code **chargeResponse** for a successful transaction is **00** or **0**.
- The returned JSON also contained Charged Amount **charged\_amount**, Transaction Currency **transaction\_currency**, Transaction Status **status** etc.
- By using this API, ensure that
  - ♣ **chargeResponse** = **00** or **0**,
  - ♣ **status** = **success**
  - ♣ **transaction\_currency** and **charged\_amount** match your currency and charged amount respectively.





## Recurrent Payment

Make recurrent payment in just **2 simple steps**:

1. Create a **Payment Plan** with KingsPay's API:  
<https://api.alphapay.live/apv1/api/gpx/paymentplans/create> This gives a response that includes **payment plan id**.
2. Add the **payment plan id** as part of the parameters in the magic *getpaidSetup()* function.

### Steps Explained:

#### Step #1

<https://api.alphapay.live/apv1/api/gpx/paymentplans/create>

This API is used to create various payment plan that customers can subscribe to through the merchant's website.

Payment Plans allow you create a subscription for your customers. After the Payment Plan has been created, a customer can be subscribed to it by simply passing the plan ID in the request to charge the customer's card.

Subscription starts from the first time the customer visits and makes payment on the merchant's website. Subsequently, all payments would be automatically charged to the customer's card at the specified intervals.

The API is called using an **HTTP POST REQUEST** by passing the following parameters, and the **Header** should have **Content-Type = application/json**.

Parameter	Required	Description
<b>seckey</b>	TRUE	This is your secret key which is a unique key generated for you when you signup as a merchant. It starts with a prefix 'FLWSECK' and ends with suffix 'X'.

Parameter	Required	Description
<b>interval</b>	False (Defaults to 'daily')	<p>The following are the charge interval possible values:</p> <p>daily; weekly; monthly; yearly; quarterly; bi-annually; every 2 days; every 90 days; every 5 weeks; every 12 months; every 6 years; every x y (where x is a number and y is the period e.g. every 5 months)</p> <p>e.g. <b>interval</b>: "daily"</p>
<b>duration</b>	False (Defaults to 0)	<p>This is the frequency, it is numeric, e.g. if set to 5 and intervals is set to <b>monthly</b> you would be charged <b>5 months</b>, and then the subscription stops.</p>
<b>name</b>	FALSE	<p>This is what would appear on the subscription reminder email</p>

Parameter	Required	Description
<b>amount</b>	False	This is the amount for the plan. Defaults to 0. If <b>amount</b> is 0, amount passed in the <b>getpaidSetup()</b> function would then be used. If amount passed in the <b>getpaidSetup()</b> function is 0, then customer would be allowed to enter amount on the KingsPay's payment interface.

## Handling Amount When Creating A Payment Plan

**Case 1:** If amount is not set, the amount that would be used for the plan is the amount charged from the customer when you start the subscription.

**Case 2:** if amount is set when creating plan, and an amount is passed when doing the subscription i.e. using the **getpaidSetup()** function. Then we charge the customer amount you passed at subscription as initial charge, and for subsequent charges use the amount set when creating the plan.

**Case 3:** if amount is not set when creating plan, and an amount is passed when doing the subscription i.e. using the **getpaidSetup()** function. Then we use the amount you passed as the amount for the plan.

## Retrieving the Payment Plan ID

The parameter you need when you create a payment plan is the payment plan ID, this can be found in the creation response, as **data.id**



## Step #2

### getpaidSetup()

The payment plan ID retrieved from **Step #1** above is then passed as one of the parameters to the getpaidSetup() function e.g. `payment_plan: 13`

Kindly see **Step #2** of **One-off Payment** section above for more details on the getpaidSetup() function.

Once the customer passes this stage, Subscription starts immediately and he won't need to return to the website for subsequent payments to be made during the following payment cycles.

## Other APIs for Recurrent Payment

- 1. List Payment Plan:** This shows how to list all payment plans on an account.

**GET** *'https://api.alphapay.live/apv1/api/gpx/paymentplans/query'*

Query Params: `seckey` (secret key)

Headers: `Content-Type – application/json`

- 2. Fetch a Payment Plan:** This describes how to fetch a single payment plan.

**GET** *'https://api.alphapay.live/apv1/api/gpx/paymentplans/query'*

Query Params: `seckey` (secret key), `id` (payment plan ID), `q` (name of the payment plan)

Headers: `Content-Type – application/json`



3. **Cancel Payment Plan:** This cancels a payment plan.

**POST** '<https://api.alphapay.live/apv1/api/gpx/paymentplans/id/cancel>'

Path Params: **id** (payment plan ID)

Body Params: **seckey** (secret key)

Headers: **Content-Type** – application/json

4. **Edit Payment Plan:** This shows how to edit a payment plan.

**POST** '<https://api.alphapay.live/apv1/api/gpx/paymentplans/id/edit>'

Path Params: **id** (payment plan ID)

Body Params: **seckey** (secret key), **name** (name of the payment plan), **status** (This is the status you would like to update the payment plan to, possible values are **active**, **cancelled**)

Headers: **Content-Type** – application/json

5. **List all subscriptions:** This shows you how to list all subscriptions on a merchant account.

**GET** '<https://api.alphapay.live/apv1/api/gpx/subscriptions/query>'

Query Params: **seckey** (secret key)

Headers: **Content-Type** – application/json

6. **Fetch a subscription:** This describes how to fetch a subscription.

**GET** '<https://api.alphapay.live/apv1/api/gpx/subscriptions/query>'



Query Params: **seckey** (secret key), **id** (This is the subscription id, it can be gotten in the response from a successful subscription charge), **email** (the email of the customer that was charged to retrieve the subscription ID).

Headers: **Content-Type** – **application/json**

- 7. Cancel a subscription:** This describes how to cancel an existing subscription.

**POST** '<https://api.alphapay.live/apv1/api/gpx/subscriptions/id/cancel>'

Path Params: **id** (payment plan ID)

Body Params: **seckey** (secret key)

Headers: **Content-Type** – **application/json**

- 8. Activate a subscription:** This describes how to activate a subscription.

**POST** '<https://api.alphapay.live/apv1/api/gpx/subscriptions/id/activate>'

Path Params: **id** (payment plan ID)

Body Params: **seckey** (secret key)

Headers: **Content-Type** – **application/json**



## Sandbox Environments & Test Cards

### Using Test Cards

To use test cards on any AlphaPlus integration, you would need to sign up on the test environment [here](http://sandbox.kingspay.live) (<http://sandbox.kingspay.live>) and:

1. create test keys and change your public key in your inlineJS or html embed.
2. change your script url to test script url below:

```
<script type="text/javascript"
      src="https://sandboxapi.alphapay.live/apv1/api/alphaPay">
</script>
```

### **Test MasterCard 3DSecure Authentication**

**Card:** 5438 8980 1456 0229  
**cvv:** 789  
**Expiry:** 09/19  
**Pin:** 3310  
**OTP:** 12345

### **Test MasterCard PIN authentication**

**Card:** 5399 8383 8383 8381  
**cvv:** 470  
**Expiry:** 10/22  
**Pin:** 3310  
**OTP:** 12345

### **Test Noauth Visa Card**

**Card:** 4751 7632 3669 9647  
**Expiry:** 09/21

### **Test Noauth VisaCard**

**Card:** 4242 4242 4242 4242  
**cvv:** 812  
**Expiry:** 01/19



### Test Verve Card

Card: 5061 4604 1012 0223 210  
Expiry: 12/21  
cvv: 780  
Pin: 3310  
OTP: 12345

### Test VISA Card (Local)

Card: 4187 4274 1556 4246  
cvv: 828  
Expiry: 09/19  
Pin: 3310  
OTP: 12345

### Test VISA Card (International)

Card: 4556 0527 0417 2643  
cvv: 899  
Expiry: 01/19

### Test American Express Card (International)

Card: 3441 7399 3556 638  
cvv: 828  
Expiry: 01/22

### Test Card Declined

Card: 5143 0105 2233 9965  
cvv: 276  
Expiry: 08/19  
Pin: 3310

### Test Card Fraudulent

Card: 5590131743294314  
cvv: 887  
Expiry: 11/20





Pin: 3310  
OTP: 12345

### **Test Card Insufficient Funds**

Card: 5258 5859 2266 6506  
cvv: 883  
Expiry: 09/19  
Pin: 3310  
OTP: 12345

### **Pre-authorization Test Card**

Card: 5377 2836 4507 7450  
cvv: 789  
Expiry: 09/19  
Pin: 3310

### **Test Bank Accounts**

Here are the test account details on AlphaPlus.

#### **Access Bank**

Account Number: 0690000031  
OTP: 12345

#### **Providus Bank**

Account Number: 5900102340, or 5900002567  
OTP: 12345

#### **Sterling Bank**

Account number: 0061333471  
OTP: 12345



**For a Free Merchant Account,  
Signup @ [www.alphaplus.live](http://www.alphaplus.live)**

**For further enquiries:  
Email: [hello@alphaplus.live](mailto:hello@alphaplus.live)**