KingsPay Goods & Services

Updated: 11/02/2025

Merchant integration

1. Merchant GoLive Form

Merchants can submit their GoLive form application at https://admin.kingspay-gs.com/go-live

After successful submission your application will be reviewed. In case of any issues with your details we will contact you and ask for corrections. Once your details are accepted, your account will be fully activated to proceed with production payments. In your Integration page should appear a **production key**.

2. Log in to the web panel

Log in to the web panel at https://admin.kingspay-gs.com/login using the login and password created during the registration. You will see two secret keys needed for payment initialization:

1. **PRODUCTION_SECRET_KEY** - used for real payments in production (available only after account activation, see section 1)
2. **TEST_SECRET_KEY** - used for testing purposes. You can also download the documentation PDF and the recommended KingsPay Goods & Services button image for your website.

Before proceeding with production payments, you must complete your account setup:

1. Set up an African account - This integration allows you to collect donation in NGN, USD, GHS, ZAR, KES. A Nigerian bank account required
2. Set up an International account - This integration allows you to collect donations in CAD, EUR, GBP, USD, ZAR. A US/Canada/Europe/UK bank account is required.
3. Set up an Espees account - for payments in Espees

3. Payment Initialization

To initialize a payment through the Merchant's backend:

Required Header: `'authorization': 'Bearer <PRODUCTION_SECRET_KEY>`

NOTE: For testing, use **TEST_SECRET_KEY** from the web panel (https://admin.kingspay-gs.com/integration).

NOTE: Body must be sent as raw JSON

POST https://api.kingspay-gs.com/api/payment/initialize

Required body parameters:

- `amount` (string) - amount in KOBE/CENTS
- `currency` (string) - currency code: "NGN" for African payments; "CAD", "EUR", "GBP", "USD", "ZAR" for international, “ESP” for Espees
- `description` (string) - payment purpose, shown to customer
- `merchant_callback_url` (string) - redirect URL after payment completion, with query parameter. Example: "[https://merchant-shop.com/?merchantOrderId=1234"](https://merchant-shop.com/?merchantOrderId=1234”)" (merchantOrderId identifies the specific payment)
- `merchant_webhook_url` (string) - merchant’s endpoint, KingsPay G&S will send a webhook there after successful payment. More on webhooks in section webhooks.
- `metadata` (key-value map) - optional merchant data stored in KingsPay G&S. Example: {"product_id": "123"}
- `payment_type` (string, optional) - "international" for CAD, EUR, GBP, USD, ZAR payments; "african" (default) for Naira payments, “espees” for Espees payments
- `email` (string, optional) - email of the customer

EXAMPLES

Successful response for all examples would be as follows:

```
Status: 200 , body: {‘message’: ‘Payment successfully initialized.’,
‘payment_id’: ‘1234’}
```

1. Payment type **“african”**

```
curl --location --request POST '<https://api.kingspay-gs.com/api/payment/initialize>' \\
        --header 'Authorization: Bearer SECRET_KEY' \\
        --header 'Content-Type: application/json' \\
        --data-raw '{
                "amount": 1000,
                "currency": "NGN",
                "description": "Test payment",
                "merchant_callback_url": "[<https://google.com>](<https://google.com/>)",
                "merchant_webhook_url": "<https://webhook.com>",
                "metadata": { "product_id": "1" },
                "payment_type”: "african”
        }'
```

2. Payment type **"international”**

```
curl --location --request POST '<https://api.kingspay-gs.com/api/payment/initialize>' \\
        --header 'Authorization: Bearer SECRET_KEY' \\
        --header 'Content-Type: application/json' \\
        --data-raw '{
                "amount": 1000,
                "currency": "USD",
                "description": "Test payment",
                "merchant_callback_url": "[<https://google.com>](<https://google.com/>)",
                "merchant_webhook_url": "<https://webhook.com>",
                "metadata": { "product_id": "1" },
                "payment_type”: "international”
        }'
```

3. Payment type **“espees”**

```
curl --location --request POST '<https://api.kingspay-gs.com/api/payment/initialize>' \\
        --header 'Authorization: Bearer SECRET_KEY' \\
        --header 'Content-Type: application/json' \\
        --data-raw '{
                "amount": 1000,
                "currency": "ESP",
                "description": "Test payment",
                "merchant_callback_url": "[<https://google.com>](<https://google.com/>)",
                "merchant_webhook_url": "<https://webhook.com>",
                "metadata": { "product_id": "1" },
                "payment_type”: "espees”
        }'
```

4. Next merchant’s backend/website should redirect a Customer to the KingsPay G&S website as follows:

`https://kingspay-gs.com/payment?id=<payment_id>`

The customer must complete the payment form + email. Payment will go through or ask for a PIN or OTP or will be redirected to 3DSecure bank. Then payment will pass or fail. The next step will be a redirection to the merchant.

NOTE: presented demos use different URLs. Please use the URL provided above.

DEMO_1: https://share.getcloudapp.com/GGuoX1z4

DEMO_2: https://share.getcloudapp.com/6qu2RgXE

5. After the customer finishes payment he’s redirected back to the merchant

Merchant's callback url specified in step 2. as `merchant_callback_url`. This allows the merchant’s backend to verify status of the payment by following request

```
GET <https://api.kingspay-gs.com/api/payment/><payment_id>
```

Example response (status: 200)

body:

```
{
  ‘amount’: ‘1000’,
  ‘currency’: ‘USD’,
  ‘description’: ‘Books’,
  ‘environment’: ‘production’,
  ‘merchant_callback_url: “https://merchant-shop.com/?merchantOrderId=1234”,
  ‘metadata’: {‘product_id’: ‘1234’},
  ‘status’: ‘SUCCESS’,
  ‘merchant_name’: ‘Test Merchant’,
  ‘payment_id’: ‘4321’,
  ‘payment_type’: {
    ‘type’: ‘nigerian’
  }
}
```

There are four possible statuses of the payment:

1. `INITIALIZED` - initialized by the merchant, with no payment attempts made,
2. `WAITING` - payment attempt made, waiting for the customer to finish payment,
3. `SUCCESS` - payment successfully processed,
4. `FAILED` - payment failed due to number of reasons like incorrect card data, authorization failure, bank failure, etc.

There are three possible payment types:

1. `african` - supporting only NGN currency,
2. `international` - supporting CAD, EUR, GBP, USD, ZAR currencies.
3. `espees` - supporting payments in Espees

6. Cancel payment

Merchant can cancel payment, which changes status in KingsPay G&S system. It won’t be possible to finalize (pay for) this payment.

```
GET https://api.kingspay-gs.com/api/payment/<payment_id>/cancel
```

Example response (status: 200)

body:

```
{
  "message": "Payment successfully cancelled",
  "payment_id": "1234"
}
```

7. Webhooks

Signature validation

Events sent from KingsPay carry the `x-kingspay-signature` header. The value of this header is a **`HMAC SHA256`** with **`Base16`** encoding. Signature of the event payload signed using your secret key for test or live environments and request body. Verifying the header signature should be done before processing the event.

Events

`“payment.succeeded”`

Example response:

```
{
  "data": {
    "status": STRING -> status of the payment,
    "description": STRING -> description provided by merchant,
    "metadata": JSON -> metadata provided by merchant,
    "currency": STRING -> currency of the payment,
    "amount": INTEGER -> amount in cents, e.g. 100 will be 1.00 in real value,
    "environment": STRING -> "test" or "production",
    "payment_type": STRING -> "african", "international" or "espees",
    "payment_id": STRING -> KP UUID of the payment,
    "payment_code": STRING -> payment code,
    "callback_url": STRING -> redirect to this endpoint after payment,
    "timestamp_utc": STRING -> payment initialization timestamp,
    "webhook_url": STRING -> webhook will be sent to this URL
  },
  "event": STRING -> event name
}
```

```
{
    "data": {
        "status": "SUCCESS",
        "description": "Test payment",
        "metadata": { "kpay_gs": "true", "product_id": "1" },
        "currency": "NGN",
        "amount": 2000,
        "environment": "test",
        "payment_type": "african",
        "payment_id": "d760406b-1988-414e-a0fd-294cad743d30",
        "payment_code": "SqVzOchItk",
        "callback_url": "https://google.com",
        "timestamp_utc": "2024-12-11T12:21:11.773107Z",
        "webhook_url": "https://webhook.site/4a9a8807-2f30-4cc2-927e-310f588ca7d0"
    },
    "event": "payment.succeeded"
}
```

8. URL paths to KingsPay Goods & Services

KPAY_GS_API_HOST - https://api.kingspay-gs.com/

KPAY_GS_WEB - https://kingspay-gs.com/

KPAY_GS_PANEL_WEB - https://admin.kingspay-gs.com/

9. Testing Cards

AFRICAN PAYMENTS

No validation
Card number: 4084 0840 8408 4081
Expiry date: 02/26, CVV: 408

PIN + OTP validation
Card number: 5060 6666 6666 6666 666
Expiry date: 02/26, CVV: 123, PIN: 1234, OTP: 123456

INTERNATIONAL PAYMENTS

VISA (3D Secure Card)
Card number: 4000 0000 0000 3063
Expiry date: future, CVV: ANY

VISA (No authentication)
Card number: 4242 4242 4242 4242
Expiry date: future, CVV: ANY