



ARAKA PAYMENT INTEGRATION MANUAL

HOW TO USE PAYMENT PAGES FOR E-COMMERCE

Abstract

This document describes two ways of using ARAKA payments platform for website e-commerce

Contents

- 1. Create an Araka Business Account2
- 2. Collect Payments3
 - 2.1 One-time Payment Page4
 - 2.2 E-COMMERCE SHOPPING CART5
 - 2.3 ANDROID APP INTEGRATION6
 - 2.4 iOS APP INTEGRATION.....6
 - 2.5 API INTEGRATION6

1. Create an Araka Business Account

As a merchant looking to utilize the Araka Payment platform, you are required to create a merchant/business account on Araka. Go to

<https://www.arakapay.com/auth/register> to create a merchant account

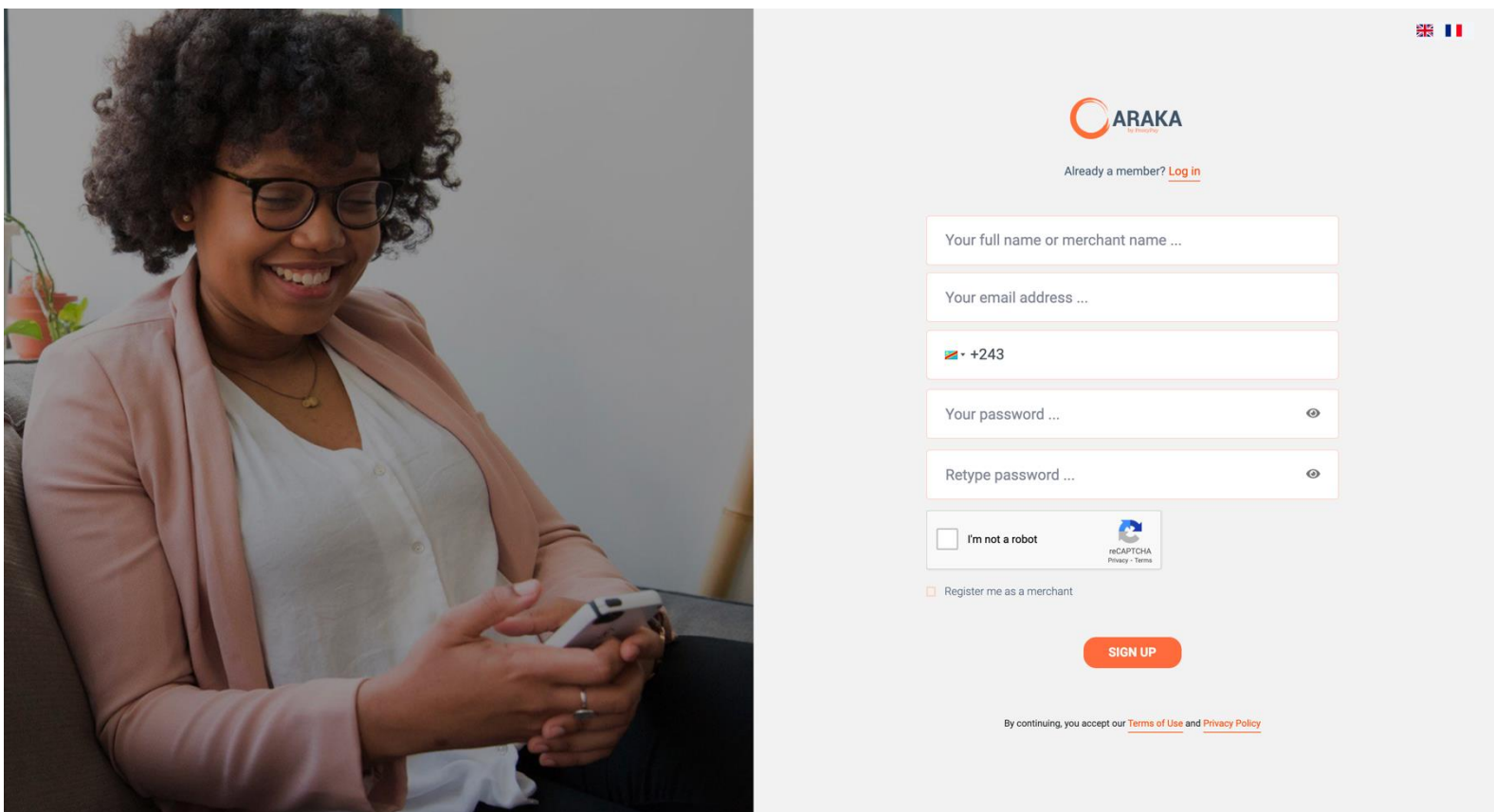


Figure 1. Araka Merchant Signup

Once you visit the URL above, perform the following steps:

1. Enter your **Business Name**
2. Enter your **Email Address**
3. Enter your **Phone Number**
4. Enter your **Password** and re-type to **Confirm**
5. Validate the **re-Captcha**

6. Select, “**Register me as a merchant**” checkbox
7. Click Sign Up

On completing the steps above you will receive a verification email in the email account you specified. This verification email contains a link that allows you to verify your account. Once you have verified your account, you are ready to start accepting payments with ARAKA.

2. Collect Payments

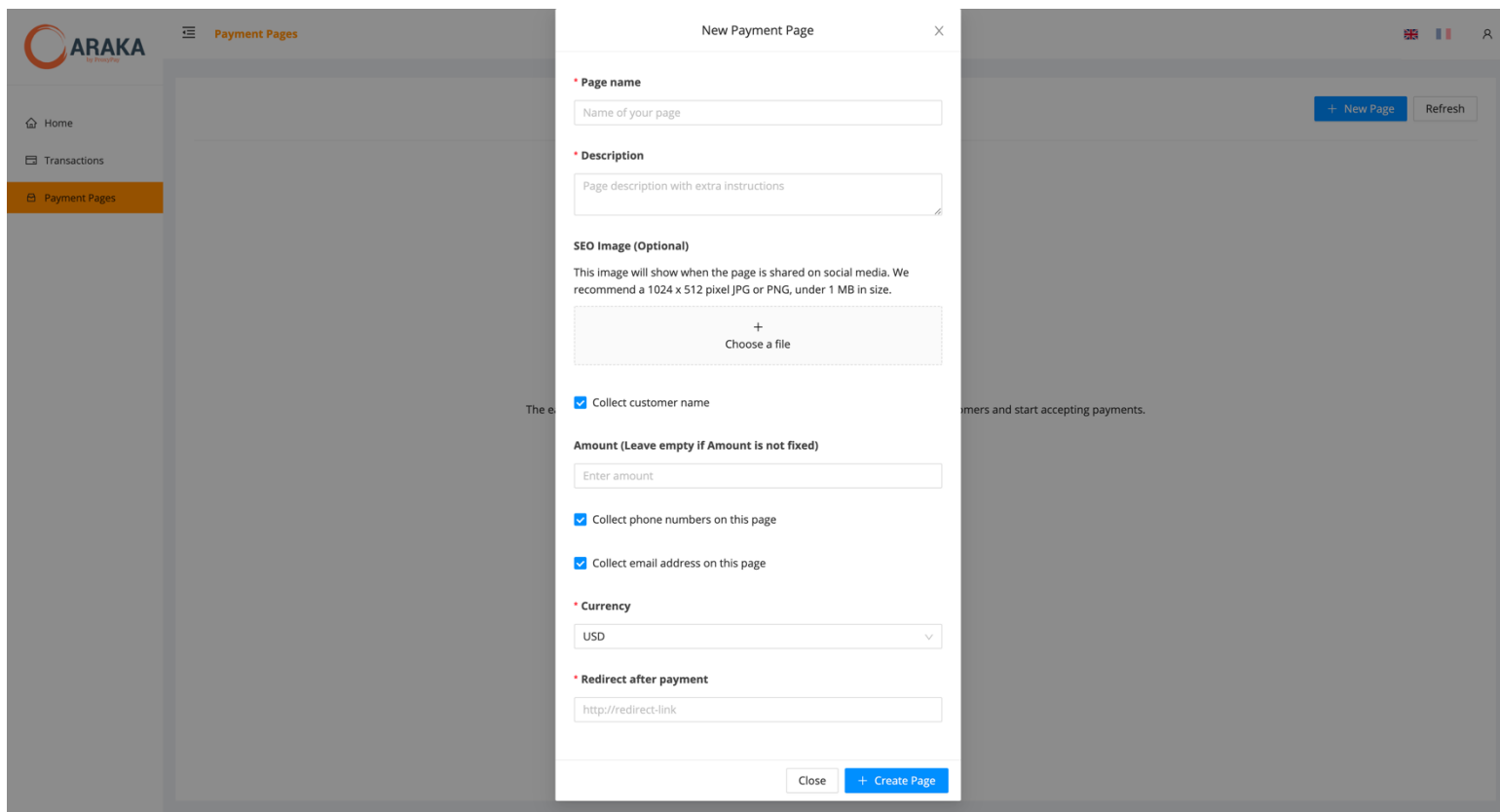
Once you’ve verified your merchant account you will have access to a Merchant Portal where you will see records of transactions made for your goods and services sold via ARAKA and the ability to create and manage payment pages. Go to <https://merchant.arakapay.com/> and login with your account’s credentials.

With an ARAKA Merchant account, you can accept payments on for your goods and services in the following ways:



2.1 One-time Payment Page

A one-time payment page is the easiest way to collect payments prices without performing any code related integrations on your website. The one-time payment page can be accessed by a link which you can share with your customers via email, WhatsApp, Twitter or any medium of your choice. This link is also unique to your

The image shows a screenshot of the ARAKA Merchant Portal. On the left is a sidebar with the ARAKA logo and navigation links: Home, Transactions, and Payment Pages (which is highlighted). The main area displays a modal window titled 'New Payment Page'. The form contains the following fields and options: 'Page name' (text input), 'Description' (text area), 'SEO Image (Optional)' (with a note about image requirements and a 'Choose a file' button), three checked checkboxes for 'Collect customer name', 'Collect phone numbers on this page', and 'Collect email address on this page', 'Currency' (dropdown menu set to USD), and 'Redirect after payment' (text input). At the bottom of the modal are 'Close' and '+ Create Page' buttons.

business, and all your customer has to do is open it and pay you. Take the following steps to create a one-time payment page:

Figure 2 New One-time Payment Page

1. Login to the ARAKA Merchant Portal with the account you created
2. Navigate to Payment Pages located on the sidebar to the left of the screen
3. Click on “New Page”
4. From the option modal that appears, select “One-time Payment”
5. Fill in the details that are required and click “Create Page”
6. On the “Payment Page” screen (see Figure 4) you will see the page you just created.
7. Click on “View” to see the details of the payment page you just created.

8. Copy the Payment Page URL and share to your customers.

2.2 E-COMMERCE SHOPPING CART

For a much more custom feature, where you collect an amount depending on the customers choice of products, for example an e-commerce shopping cart, you can do the following with the one-time payment page:

1. Login to the ARAKA Merchant Portal with the account you created
2. Navigate to Payment Pages located on the sidebar to the left of the screen
3. Click on “New Page”
4. From the option modal that appears, select “One-time Payment”
5. Fill in the details that are required and click “Create Page”
6. Leave the “Amount” field empty
6. On the “Payment Page” you will see the page you just created.
7. Click on “View” to see the details of the payment page you just created.
8. Copy the Payment Page URL.
9. On your checkout page, when you are ready to collect payment, redirect the user with GET request to the “Payment Page” URL you copied.
10. Use the URL <https://merchant.arakapay.com/payment/AAAAA-BBBBBB-CCCCC-DDDD>.
11. Include the following parameters as query strings in the URL:

```
"pageTitle" : "Sample title",  
"pageDescription" : "Sample description",  
"customerFullName" : "Mike Wazowski",  
"customerPhoneNumber" : "+243990998098",  
"customerEmailAddress" : "example@user.com",  
"transactionReference" : "TRW344232134",  
"currency" : "USD",  
"amount" : "40",  
"merchant" : "b255c76a-41cc-4398-bfd2-b2f9e3238c2d"
```

10. Once the transaction is completed or cancelled, the user is redirected to the URL provided in the “Redirect URL” field you specified when creating the Payment Page with the following sample payload:

```
{ "systemReference": "1234", "transactionStatus": "SUCCESS" }
```

The “systemReference” in this payload, can be identified as a “Transaction ID” on the Merchant portal, in the transactions table.

2.3 ANDROID APP INTEGRATION

When developing for Android, use the **WebView** activity to send the transaction payload to the payment page created (see section 2.2)

The transaction payload must include the properties as seen in Section 2.2

After the transaction has been successfully processed, the user will be redirected to the Application's **Webview** using the "**redirectURL**" that was provided when creating the payment page.

2.4 iOS APP INTEGRATION

When developing for iOS, use the **WKWebView** object and programmatically embed into your view hierarchy, or add it using Interface Builder. Send the transaction payload to the payment page created (see section 2.2)

The transaction payload must include the properties as seen in Section 2.2

After the transaction has been successfully processed, the user will be redirected to the Application's **WKWebView** object using the "**redirectURL**" that was provided when creating the payment page.

2.5 API INTEGRATION

For Merchants who prefer to use plain old API requests, there is the possibility of using our API.

The API requires a bearer token that can be obtained by using our login endpoint. Subsequent requests should always include this bearer token in the Authorization header of the request.

A swagger page containing postman collections can always be obtained here:

<https://araka-api-uat.azurewebsites.net/swagger/index.html>

Details of the various requests are as follows:

2.5.1 Login

POST /api/login

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed:



Request Formats

application/json, text/json

Sample:

```
{  
  "emailAddress": "string",  
  "password": "string"  
}
```

Response Information

application/json, text/json

Sample:

```
{  
  "token": "string",  
  "username": "your email"  
}
```

2.5.2 Payment Request

POST /api/ pay/paymentrequest

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

The table below provides information about the properties expected in the payload to be sent to this request:

Property	DataType (maxSize)	Required	Description
order	Object	Yes	Information about the operation to be performed

Order.Paymentpageid	String(20)	Yes	Identifier for payment page. Refer to the section marked 2.1 (One Time Payment Page) the ID is the last part of your payment page URL eg: https://merchanta.arakapay.com/payment/xxxx . Where xxxx is the paymentpageid
Order.customerFullName	String(180)	No	Full Name for customer
Order.customerPhoneNumber	String(180)	No	Phone Number of the customer
Order.transactionReference	String(20)	Yes	Reference transaction ID from source request to be used in reconciliations
order.customerEmailAddress	String(180)	No	Email address of the customer
order.currency	String(3)	Yes	Char 3 ISO currency code
order.amount	Decimal(8,2)	Yes	Amount involved in the transaction

Order.redirectURL	String(400)		URL where result of the transaction is to be sent after completion of the operation. This endpoint should support POST and redirect.
paymentChannel	Object	Yes	Information about the payment channel the client should use in completing the transaction
paymentChannel.channel	String(30)	Yes	The channel to be used for payments. Eg (MOBILEWALLET)
paymentChannel.provider	Object	Yes	The telco to be used to process the payment (MPESA,AIRTEL,ORANGE)
paymentChannel.walletid	String(14)	Yes	phone number in international format (+243XXXX...)

Request Formats

application/json, text/json

Sample:

```
{
  "order": {
    "paymentPageId": "string",
    "customerFullName": "string",
    "customerPhoneNumber": "string",
    "customerEmailAddress": "string",
    "transactionReference": "string",
    "amount": 0,
    "currency": "string", //USD,CDF
    "redirectURL": "string"
  },
  "paymentChannel": {
    "channel": "string", //MOBILEMONEY,CARD
    "provider": "string", //MPESA,AIRTEL,ORANGE
    "walletID": "string" //MSISDN
  }
}
```

Response Information

application/json, text/json

Sample:

```
{
  "transactionId": "string",
  "originatingTransactionId": "string",
  "statusCode": "202",
  "statusDescription": "ACCEPTED"
}
```

2.5.3 Status Inquiry

GET api/reporting/transactionstatus/{transactionid}

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

Request Formats

api/reporting/transactionstatus/xxxx

where xxxx is the transactionId received in the transaction request

Response Information

application/json, text/json

Sample:

```
{
  "transactionId": "string",
  "originatingTransactionId": "string",
  "statusDescription": "string", //APPROVED,DECLINED
  "statusCode": "202"
}
```

2.5.4 Send Mobile Money

POST /api/pay/sendmobilemoney

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

The table below provides information about the properties expected in the payload to be sent to this request:

Property	DataType (maxSize)	Required	Description
order	Object	Yes	Information about the operation to be performed
Order.customerFullName	String(180)	No	Full Name for customer
Order.customerPhoneNumber	String(180)	No	Phone Number of the customer

Order.transactionReference	String(20)	Yes	Reference transaction ID from source request to be used in reconciliations
order.customerEmailAddress	String(180)	No	Email address of the customer
order.currency	String(3)	Yes	Char 3 ISO currency code
order.amount	Decimal(8,2)	Yes	Amount involved in the transaction
Destination	Object	Yes	Information about the payment channel the client should use in completing the transaction
destination.provider	Object	Yes	The telco to be used to process the payment (MPESA,AIRTEL,ORANGE)
destination.walletid	String(14)	Yes	phone number in international format (+243XXXX...)

Request Formats

application/json, text/json

Sample:

```
{
  "order": {
    "paymentPageId": "string",
    "customerFullName": "string",
    "customerPhoneNumber": "string",
    "customerEmailAddress": "string",
    "transactionReference": "string",
    "amount": 0,
    "currency": "string",
    "callbackURL": "string"
  },
  "destination": {
    "provider": "string", //MPESA,AIRTEL,ORANGE
    "walletID": "string" //MSISDN
  }
}
```

Response Information

application/json, text/json

Sample:

```
{
  "transactionId": "string",
  "originatingTransactionId": "string",
  "statusCode": "202",
  "statusDescription": "ACCEPTED"
}
```

2.5.3 Socodee Meter Inquiry

GET api/socodee/getmerterdetails/{transactionid}

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

Request Formats

`api/socodee/getmerterdetails/xxxx`

where xxxx is the meter number

Response Information

`application/json, text/json`

Sample:

```
{ "customerName":"string", "currency":"USD", "amount":3 }
```

2.5.5 Socodee Payment Request

POST `/api/socodee/payment`

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

The table below provides information about the properties expected in the payload to be sent to this request:



Property	DataType (maxSize)	Required	Description
order	Object	Yes	Information about the operation to be performed
Order.Paymentpageid	String(20)	Yes	Identifier for payment page. You can obtain this from your payment page URL eg: https://merchanta.arakapay.com/payment/xxxx . Where xxxx is the paymentpageid
Order.customerFullName	String(180)	No	Full Name for customer
Order.customerPhoneNumber	String(180)	No	Phone Number of the customer
Order.transactionReference	String(20)	Yes	Reference transaction ID from source request to be used in reconciliations
order.customerEmailAddress	String(180)	No	Email address of the customer
order.currency	String(3)	Yes	Char 3 ISO currency code

order.amount	Decimal(8,2)	Yes	Amount involved in the transaction
destination	Object	Yes	Information about the beneficiary
destination.meterNumber	String	Yes	The meter number
destination.phoneNumber	String(14)	Yes	phone number in international format (+243XXXX...)

Request Formats

application/json, text/json

Sample:

```
{
  "order": {
    "customerFullName": "string",
    "customerPhoneNumber": "string",
    "customerEmailAddress": "string",
    "transactionReference": "string",
    "amount": 0,
    "currency": "USD"
  },
  "destination": {
    "meterNumber": "string",
    "phoneNumber": "string" //MSISDN
  }
}
```

Response Information

application/json, text/json

Sample:

```
{  
  "transactionId": "string",  
  "originatingTransactionId": "string",  
  "statusCode": "200",  
  "statusDescription": "SUCCESS"  
}
```



2.5.6 White Label Gateway Request

POST /api/pay/gatewayrequest

This endpoint returns a redirect page for payee to be sent to that is branded with the merchants details and description as designed in the payment page section.

Access to this endpoint requires setup changes on your account. Contact the technical team for this configuration to be completed

The table below provides information about the properties expected in the payload to be sent to this request:

Property	DataType (maxSize)	Required	Description
order	Object	Yes	Information about the operation to be performed
Order.Paymentpageid	String(20)	Yes	Identifier for payment page. Refer to the section marked 2.1 (One Time Payment Page) the ID is the last part of your payment page URL eg: https://merchant.arakapay.com/payment/xxxx . Where xxxx is the paymentpageid
Order.customerFullName	String(180)	No	Full Name for customer
Order.customerPhoneNumber	String(180)	No	Phone Number of the customer

Order.transactionReference	String(20)	Yes	Reference transaction ID from source request to be used in reconciliations
order.customerEmailAddress	String(180)	No	Email address of the customer
order.currency	String(3)	Yes	Char 3 ISO currency code
order.amount	Decimal(8,2)	Yes	Amount involved in the transaction
Order.redirectURL	String(400)		URL where result of the transaction is to be sent after completion of the operation. This endpoint should support POST and redirect.
paymentChannel	Object	Yes	Information about the payment channel the client should use in completing the transaction

paymentChannel.channel	String(30)	Yes	The channel to be used for payments. Eg (MOBILEWALLET)
paymentChannel.provider	Object	Yes	The telco to be used to process the payment (MPESA,AIRTEL,ORANGE)
paymentChannel.walletid	String(14)	Yes	phone number in international format (+243XXXX...)

Request Formats

application/json, text/json

Sample:

```
{
  "order": {
    "paymentPageId": "string",
    "customerFullName": "string",
    "customerPhoneNumber": "string",
    "customerEmailAddress": "string",
    "transactionReference": "string",
    "amount": 0,
    "currency": "string", //USD,CDF
    "redirectURL": "string"
  },
  "paymentChannel": {
    "channel": "string", //MOBILEMONEY,CARD
    "provider": "string", //MPESA,AIRTEL,ORANGE
    "walletID": "string" //MSISDN
  }
}
```

Response Information

application/json, text/json

Sample:

```
{
  "transactionId": "string",
  "originatingTransactionId": "string",
  "statusCode": "202",
  "statusDescription": "ACCEPTED"
}
```

