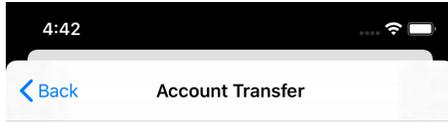
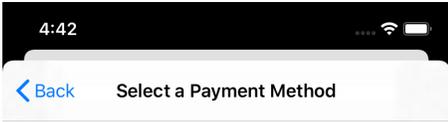
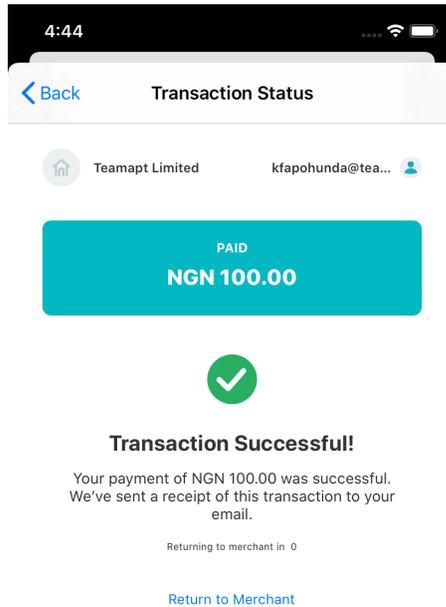
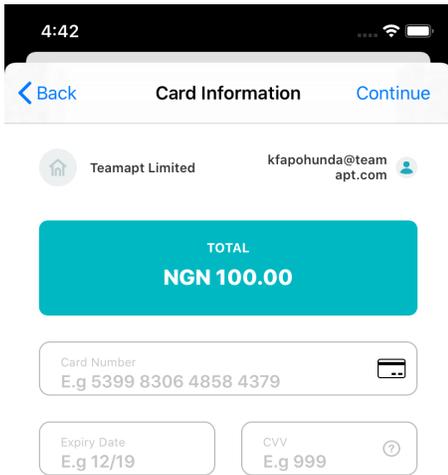


Monnify iOS SDK





Powered by  monnify

Powered by  monnify

Getting started with the Monnify iOS SDK

First off, the Monnify iOS SDK allows you to accept payments from customers in your iOS application via:

1. Card Payment
2. Bank Transfer (this is what is special:)

Adding the iOS SDK to your project via CocoaPods

MonnifyiOSSDK is available through [CocoaPods](<https://cocoapods.org>). To install it, simply add the following line to your Podfile:

```
pod 'MonnifyiOSSDK'
```

Access an instance of the Monnify SDK

```
let monnify = Monnify.shared
```

Set your merchant API key and contract code.

This should be done in your `AppDelegate.swift` file. This should only be done once in the entire application.

This can be a TEST or LIVE. The TEST mode works on a sandbox environment and payment can be simulated [here](<https://websim.sdk.monnify.com/#/bankingapp>).

Remember to switch to ApplicationMode.live when generating live builds for your application on production.

```
let apiKey = "MK_PROD_XXXXXXX"
let contractCode = "1234567890"
let mode = ApplicationMode.test

monnify.setApplicationMode(applicationMode: mode)
monnify.setApiKey(apiKey: apiKey)
monnify.setContractCode(contractCode: contractCode)
```

Specify the transaction parameters as shown below:

```
let amount = Decimal(100)
let paymentRef = "ASDF123454321"

let parameter = TransactionParameters(amount: amount,
                                     currencyCode: "NGN",
                                     paymentReference: paymentRef,
                                     customerEmail: "johndoe@example.com",
                                     customerName: "John Doe" ,
                                     customerMobileNumber: "08000000000",
                                     paymentDescription: "Payment Description.",
                                     incomeSplitConfig: [],
                                     metaData: ["deviceType":"ios",
                                     "userId":"user314285714"],
                                     paymentMethods: [PaymentMethod.card, PaymentMethod.accountTransfer],
                                     tokeniseCard: false)
```

Launch the payment gateway, perhaps when the user clicks on a 'Pay' button.

The initializePayment() method requires the following:

- * A ViewController instance.
- * An instance of TransactionParameters, e.g `parameter` above.
- * A completion/callback to be called with the monnify SDK is completing either after a failure or success.

```
monnify.initializePayment(withTransactionParameters: parameter,
                          presentingViewController: self,
                          onTransactionSuccessful: { result in
    print(" Result \(result) ")
  })
```

Get the outcome of the payment attempt.

The result is available via the callback registered in `initializePayment()` as described above.

For example, you can access the transaction status from the result as shown below.

```
let transactionStatus = result.transactionStatus
```

Other information in the result includes transaction reference, payment reference, payment method, and the amount paid.

Transaction Status Possible Values!

Possible values for transaction status enum returned from the SDK and their meanings below:

Type	Meaning
pending	No payment has been made on this transaction.
paid	Payment has been completed on this transaction.

overpaid	Customer overpaid on this transaction and Merchant has configured that overpayment is accepted. This is only possible for account transfer payment method.
partially Paid	Customer underpaid on this transaction and Merchant has configured that underpayment is accepted. This is only possible for account transfer payment method.
reversed	Payment has been made on this transaction but it cannot be accepted because of the merchant configuration or because of some other reasons. This is only possible for account transfer payment method.
expired	No payment was made before the allowed time elapsed. Hence transaction has expired and no payment can be made on this transaction.
cancelled	User cancelled and closed the SDK before payment was made.
failed	The transaction ended with a failure. This can happen if a transaction has expired or if an incorrect amount was paid and the merchant configured that only the exact amount should be accepted

Please reach out to the support team for any clarification.

Payment Method Possible Values!

Type	Meaning
card	The Payment method used for payment on this transaction is Card.
accountTransfer	The payment method used for payment on this transaction is Account Transfer.