Geidea Gateway

iOS SDK EGYPT BNPL Integration Guide

Version 2.0

|  |  |
| --- | --- |
| Document version: | 1.0 |
| Date of version: | 30-05-2002 |
| Author: | Eugen Vidolman |

|  |
| --- |
|  |

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Changes |
| 2.2 | 30.05.2022 | Eugen Vidolman | BNPL valu |

|  |  |
| --- | --- |
|  |  |

Contents

[1. Introduction 3](#_Toc108787388)

[1.1 Document Purpose 3](#_Toc108787389)

[2. iOS Integration 4](#_Toc108787390)

[2.1 How to start 4](#_Toc108787391)

[2.2 ValU Installment Provider: 4](#_Toc108787392)

[2.2.2 Step 2: Get Instalment plan: 5](#_Toc108787393)

[2.2.3 Step 3: Installment Plan selected 8](#_Toc108787394)

[2.2.4 Step 4: Payment Step 10](#_Toc108787395)

[2.2.5 Step 5: OTP Screen 10](#_Toc108787396)

[2.3 Shahry Installment Provider: 12](#_Toc108787397)

[2.3.1 Step 1: Verify customer 13](#_Toc108787398)

[2.3.2 Step 2: Shahry confirm 15](#_Toc108787399)

[2.4 Souhoola Installment Provider: 18](#_Toc108787400)

[2.4.1 Step 1: Verify customer 18](#_Toc108787401)

[2.4.2 Step 2: Retrieve InstallmentPlans 19](#_Toc108787402)

[2.4.3 Step 3: Review Transaction 23](#_Toc108787403)

[2.4.4 Step 4: Down Payment if nextStep param is proceedWithDownPayment 27](#_Toc108787404)

[2.4.5 Step 4 or 5 if needed: OTP and confirm 27](#_Toc108787405)

[3 Geidea Form 29](#_Toc108787406)

[See the main documentation for PayWithGeideaForm including all BNPL providers and payment methods 29](#_Toc108787407)

[Appendix A. Glossary 30](#_Toc108787408)

# Introduction

## Document Purpose

The purpose of the iOS Software Development Kit (iOS SDK) Integration guide is to serve as a technical documentation for merchants that want to integrate with Geidea and want to use the Payment Gateway services with their systems.

This mini document is for BNPL (Buy Now Pay Later) providers on Egypt, available in iOS SDK.

The fully tested implementation it is encapsulated in PayWthGeideaForm available with UI and receipt, ready to use. We recommend to use that function from the main documentation.

# iOS Integration

SDK Requirements:

Minimum iOS version: 11.0

- Swift 4.0, 4.2, 5.X

- Objective C

## How to start

All functions from SDK will be available on GeideaBNPLAPI for how to start with SDK please revise the main documentation

## ValU Installment Provider:

ValU is an Egypt Installment provider that allows your customer to pay with installments

We recommend using our PayWithGeideaForm with a complete flow ready to be integrated. If you want to use you own payment you can configure the payWithGeideaForm with Payment Methods as [valu]

* + 1. Step 1 Verify Customer function

Graphical user interface, application

Description automatically generated

The UI for this function needs to have an input and a confirm button as in PayWithGeideaForm implementation.

On confirm button you need to call the following function from SDK with the customer input PhoneNumber

GeideaBNPLAPI.VALUVeriFyCustomer(with: phoneNumber, completion: { response, error **in**

completion(response,error)

})

### Step 2: Get Instalment plan:

Graphical user interface, application

Description automatically generated

Use this method for getting the valU installment plans object based of GDInstallmentPlanDetails

Please see the parameters below:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | Phone number identifier |
| totalAmount | Double | Total Amount to pe paid |
| adminFees | Double | Can be 0 at this stage. AdminFees will be retrieved from the installment plan response |
| currency | String | Total Amount currency |
| downPayment | Double | Down Payment from the customer |
| giftCardAmount | Double | List of Default payment methods supported by configuration.  Can be restricted using payment Methods in Pay |
| campaignAmount | Double | Default is 1800, used for geideaform |
| completion | ((GDInstallmentPlansResponse?, GDErrorResponse?) -> Void) | Use GDInstallmentPlansResponse |

GDInstallmentPlansResponse properties:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String |  |
| detailedResponseCode | String |  |
| responseMessage | String |  |
| responseCode | String |  |
| language | String |  |
| currency | String | Total Amount currency |
| bnplOrderId | String | Used in the feature steps |
| minimumDownPayment | Double |  |
| totalAmount | Double |  |
| financedAmount | Double | Amount that will be financed on installments |
| giftCardAmount | Double | Amount available in valU |
| campaignAmount | Double | Discount available in valU |
| installmentPlans | [GDInstallmentPlan] | * + - tenm |

Example Swift:

let details = GDInstallmentPlanDetails(customerIdentifier: String, totalAmount: Double, currency: String, downPayment: Double, giftCardAmount: Double, campaignAmount: Double, adminFees: 0)

GeideaBNPLAPI.VALUGetInstallmentPlan(with: details, completion: { [self] response, error in

if let err = error {

if err.detailedResponseCode == "001" || err.detailedResponseCode == "005" || err.detailedResponseCode == "0013" {

// complete the flow, show your failure screen. From this error the user cannot recover

return

}

if err.isError && !err.errors.isEmpty {

//Show in screen error from errors array

} else {

//Show in screen error

}

} else if let plans = response?.installmentPlans {

// SHOW sll the installment plans in a collectionview

}

})

### 2.2.3 Step 3: Installment Plan selected

Use this method after the customer selects an installment plan from table. Order will be retrieved in response. It is mandatory to have the order id for the rest Steps. The next step can be payment if an admin fee or/and down payment must be paid by the customer or OTP Screen

Below you can find the parameters for GDVALUInstallmentPlanSelectedDetails

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | amount: Double – 2 decimals e.g., 2.45  currency: String – 3 letters code e.g., SAR/EGP must be one of your config currencies |
| totalAmount | Double | cardNumber: String – user card number  cardHolderName: String – max 255 chars  cvv: String – 3 or 4 (amex) letters  expiryMonth: Int – digits from 1 to 12  expiryYear: Int – digits from 1 to 99 |
| curency | String |  |
| adminFees | Double | Obtained from GDInstallmentPlan list, selected by the customer at Step 2 |
| downPayment | Double | Down payment from customer input |
| giftCardAmount | Double | From user input |
| campaingAmount | Double | From user input |
| tenure | Int | Obtained from GDInstallmentPlan list, selected by the customer at Step 2 |
| merchantReferenceId \*Optional\* | String | Part of Order details |
| callbakURL \*Optional\* | String | Part of Order details |
| biilingAddress \*Optional\* | GDAddress | Part of Order details |
| shippingAddress \*Optional\* | GDAddress | Part of Order details |
| customerEmail \*Optional\* | String | Part of Order details |
| orderId |  | Optional if you want to use an already created order |
|  |  |  |
| completion\*Required\* | (GDInstallmenrPlanSelectedResponse?, GDErrorResponse?) -? Void | used for returning success **GDInstallmentPlanSelectedResponse**, or failure **GDOrderResponse** to your application |
|  |  |  |

Example parameters Swift:

let installmentPlanDetails = GDVALUInstallmentPlanSelectedDetails(customerIdentifier: String, totalAmount: Double, currency: String, adminFees: Double, downPayment: Double, giftCardAmount: Double, campaignAmount: Double, tenure: Int, merchantReferenceId: String, callbackUrl: String, billingAddress: GDAddress, shippingAddress: GDAddress, customerEmail: String, orderId: String, bnplOrderId: String, otp String, restrictPaymentMethods: Bool, paymentMethods: [String])

Example Swift:

GeideaBNPLAPI.VALUInstallmentPlanSelected(with: installmentPlanDetails, completion: { response, error in

completion(response,error)

})

### 2.2.4 Step 4: Payment Step

Use this method if the customer needs to pay the admin fees or/and down payment before the OTP Screen/ Step. Configure payment amount to be just the AdminFee + Down Payment amount

* + - 1. You can use pay function from GeideaPaymentAPI. See main documentation
      2. You can use payWithGeideaForm function from GeideaPaymentAPI. with restricted payment methods list e.g (visa,mastercard,meeza,meezaDigital), otherwise the customer can select installment providers to pay the down payment.

### 2.2.5 Step 5: OTP Screen

Text, application

Description automatically generated

In the last step you need to use two API calls.

1. GenerateOTP: VALUGenerateOTP function

Use to generate the OTP code sent to the customer. The same API call can be used to resend OTP from the resend code button. OTP code will be 5 alphanumeric characters that must be typed by the customer from the SMS.

|  |  |  |
| --- | --- | --- |
| phoneNumber | String | Phone number identifier of the customer |
| bnplOrderId | String | BNPLOrderId from previous steps response |
| completion | ((GDBNPLResponse?, GDErrorResponse?) -> Void) |  |

GDBNPLResponse params:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String |  |
| detailedResponseCode | String |  |
| responseMessage | ((String |  |
| responseCode | String |  |
| bnplOrderId | String |  |

Example Swift:

GeideaBNPLAPI.VALUGenerateOTP(with: phoneNumber, BNPLOrderId: bnplOrderID, completion: { response, error in

completion(response,error)

})

1. ValuPurchase

You need to use this on the purchase button to finalize the purchase

Parameters for GeideaBNPL.ValuPurchase, GDBNPLPurchaseDetails

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | the customer Intentifier |
| orderId | String | the order Id from Step 2 response. |
| bnplOrderId | String | The bnplOrderId from generateOTP response |
| otp | String | OTP from user input |
| totalAmount | String | totalAmount |
| currency | String | Currency of the total Amount |
| downPayment | Double | From the user in Step2 |
| giftCardAmount | Double | From the user in Step2 |
| campaignAmount | Double | From the user in Step2 |
| tenure | Int | From the user from selected installment plan |
| adminFees | Double | From the user from selected installment plan |

GeideaBNPL.ValuPurchase params

|  |  |  |
| --- | --- | --- |
| purcaseParams | GDBNPLPurchaseDetails |  |
| completion | (GDBNPLPurchaseResponse?, GDErrorResponse?) -? Void | used for returning success **or failure** |

Example Swift:

GeideaBNPLAPI.VALUPurchase(with: purchaseDetails, completion: { response, error in

completion(response,error)

})

## Shahry Installment Provider:

Shahry is an Egypt Installment provider that allows your customer to pay with installments.

Several steps are required if you want to develop the collection of APIs of Shahry.

It is mandatatory for your customers to download Shary app, because installments list and selected installment plan will be in the Shahry App.

Where to find link : <https://shahry.app/my-shahry-id>

Learn More link: https://shahry.app/

We recommend using our PayWithGeideaForm with a complete flow ready to be integrated. If you want to use your own payment you can configure the payWithGeideaForm with Payment Methods as [shahry]

### 2.3.1 Step 1: Verify customer

You need to implement a screen as below containing a textfield for the user

Graphical user interface, text, application, chat or text message

Description automatically generated

## 

When user clicks next button the GeideaBNPLAPI. sharyInstallmentPlanSelected function from the SDK.

|  |  |  |
| --- | --- | --- |
| with | GDShahrySelectInstallmentPlan | Id of the product |
| completion | (GDInstallmentPlanSelectedResponse?, GDErrorResponse?) -? Void | used for returning success **or failure** |

GDInstallmentPlanSelectedResponse:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String |  |
| detailedResponseCode | String |  |
| responseMessage | String |  |
| responseCode | String |  |
| orderId | String |  |
| nextStep | String |  |

Below you can find Parameters for the GDShahrySelectPlanInstallment:

|  |  |  |
| --- | --- | --- |
| customerIdentifer | String | The Shahry Id of the customer |
| totalAmount | Double | Total Amount of your shopping cart |
| currency | String | Currency of your shopping cart |
| merchantReferenceId (Optional) | String | Merchant details |
| callbackURL (Optional) | String | String |
| billingAddress (Optional) | GDAddress | Customer details for envoicing |
| shippingAddress (Optional) | GDAddress | Customer details for envoicing |
| customerEmail (Optional) |  | Customer details for envoicing |
| paymentMethods (Optional) |  |  |
| restrictPaymentMethods (Optional) |  |  |
| Items | [GDBNPItem] | The list of your shopping cart items to sell. It is mandatory |

Below you can find the GDBNPLItem object params. Above you need to send a list of items

|  |  |  |
| --- | --- | --- |
| merchantItemId | String | Id of the product |
| name | String | Product name |
| itemDescription | String | Product description |
| count | Int | Number of items of the product |
| price | Double | Price of the product |
| currency | String | Currency of the product |

Example Swift:

GeideaBNPLAPI.sharyInstallmentPlanSelected(with: installmentPlanDetails, completion: { response, error in

completion(response,error)

})

### 2.3.2 Step 2: Shahry confirm

Graphical user interface, text, application

Description automatically generated

In the previous step, the customer receives a push notification on Shahry App and select an installment Plan. A token order is generated on the app, and the customer needs to input in the textfield for the GeideaBNPLAPI.shahryConfirm API

|  |  |  |
| --- | --- | --- |
| with | GDShahryConfirm | See below |
| completion | (GDOrderResponse?, GDErrorResponse?) -? Void | used for returning success **or failure** |

GDShahryConirm params:

|  |  |  |
| --- | --- | --- |
| orderId | String | See order from previous step |
| orderToken | String | User input |

From GDOrderRespone you’ll find transactions (GDTransactionResponse)

Example calulate fee and down Payment swift:

Let bnplDetails = response?.transactions?.last?.bnplDetails

let amountToCollect = response?.settleAmount ?? 0

let fee = bnplDetails?.adminFees ?? 0 + (bnplDetails?.otherFees ?? 0)

if amountToCollect is higher then 0 you’ll need to Add a new view and then a new step to collect the down payment from user as the below UI: the payment can be done as cash on delivery or using our PaywithGeideaForm with restricted PaymentMethods to e.g [visa, mastercard, meeze, meezadigital] or what payment methods do you need. Do not include the installments providers in payment methods at this step: See the example

Graphical user interface, text

Description automatically generated

Example Swift:

GeideaBNPLAPI.shahryConfirm(with: purchaseDetails, completion: { response, error in

completion(response,error)

})

## Souhoola Installment Provider:

Souhoola is an Egypt Installment provider that allows your customer to pay with installments.

Several steps are required if you want to develop the collection of APIs of Souhoola.

We recommend using our PayWithGeideaForm with a complete flow ready to be integrated. If you want to use your own payment you can configure the payWithGeideaForm with Payment Methods as [souhoola]

### 2.4.1 Step 1: Verify customer

You need to implement a screen as below containing two textfields for the user to collect souhoola customerId and the PIN

Graphical user interface, application

Description automatically generated

When user clicks next button the GeideaBNPLAPI. souhoolaVerifyCustomer function from the SDK with user data collected: phone Number and PIN.

Example Swift:

GeideaBNPLAPI.souhoolaVeriFyCustomer(with: phoneNumber, pin: pin, completion: { response, error **in**

**self**.souhoolaVerifyResponse = response

completion(response,error)

})

You will have in the complection GDSouhoolaVerifyResponse, that should be saved for the next steps and GDErrorResponse in case of error

GDSouhoolaVerifyResponse params

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String | used for reference and status |
| detailedResponseCode | String | used for reference and status |
| responseCode | String | used for reference and status |
| responseMessage | String | used for reference and status |
| language | String | used for reference and status |
| approvedLimit | Double | Souhoola limits. Used in the next steps calculation |
| outstanding | Double | Souhoola limits. Used in the next steps calculation |
| availableLimit | Double | Souhoola limits. Used in the next steps calculation |
| minLoanAmount | Double | Souhoola limits. Used in the next steps calculation |

### 2.4.2 Step 2: Retrieve InstallmentPlans

You need to call the function GeideaBNPLAPI.getInstallmentPlan to retrieve the available installment plans for the user. See below an example UI:

Graphical user interface, text, application

Description automatically generated

GDSouhoolaInstallmentPlans params

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | used for reference and status |
| customerPin | String | used for reference and status |
| totalAmount | Double | Souhoola limits. Used in the next steps calculation |
| currency | Double | Souhoola limits. Used in the next steps calculation |
| downPayment | Double | Souhoola limits. Used in the next steps calculation |

Example Swift:

**let** details = GDSouhoolaRetreiveInstallmentPlans(customerIdentifier: customerId, customerPin: pin, totalAmount: amount, currency: currency, adminFees: 0, downPayment: Double)

GeideaBNPLAPI.souhoolaGetInstallmentPlan(with: details, completion: { response, error **in**

completion(response,error)

})

Response: GDSouhoolaInstallmentPlansResponse

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String | used for reference and status |
| detailedResponseCode | String | used for reference and status |
| responseCode | String | used for reference and status |
| responseMessage | String | used for reference and status |
| language | String | used for reference and status |
| currency | String | currency |
| installmentPlans | [GDInstallmentPlans] | List of installment Plans available for the user |

Response: GDInstallmentPlans

|  |  |  |
| --- | --- | --- |
| tenorMonth | Int | Month of the selected plan |
| installmentAmount | Double | Amount of the selected plan |
| adminFees | Double | Fee of the selected plan |
| downPayment | Double | downPayment for the selected plan |
| minDownPayment | Double | Minimum Down Payment of the selected plan |
| rate | String |  |
| promoCode | String |  |
| downPaymentPromo | Bool |  |
| adminFeesPromo | Bool |  |
| interestPromo | Bool |  |

You need to calculate the range for Souhoola Down Payment and display a range of downPayment min and max:

Example Swift:

* + - availableLimit from GDSouhoolaVerifyResponse
    - minDownPayment from GDSouhoolaInstallmentPlansResponse
    - souhoolaMinimumAmount from GDSouhoolaVerifyResponse or config (sohoolaMinimumAmount)

**func** getMinMax(availableLimit: Double, minDownPayment: Double, souhoolaMinimumAmount: Int) -> (Double,Double) {

**var** minAmount = 0.0

**var** maxAmount = 0.0

//MIN calculation

**if** limit < amount?.amount ?? 0 {

minAmount = (amount?.amount ?? 0) - limit

minAmount = max(minAmount, minDownPayment)

} **else** {

minAmount = minDownPayment

}

maxAmount = (amount?.amount ?? 0) - Double(souhoolaMinimumAmount)

**if** minAmount > maxAmount {

**return** (maxAmount, minAmount)

} **else** {

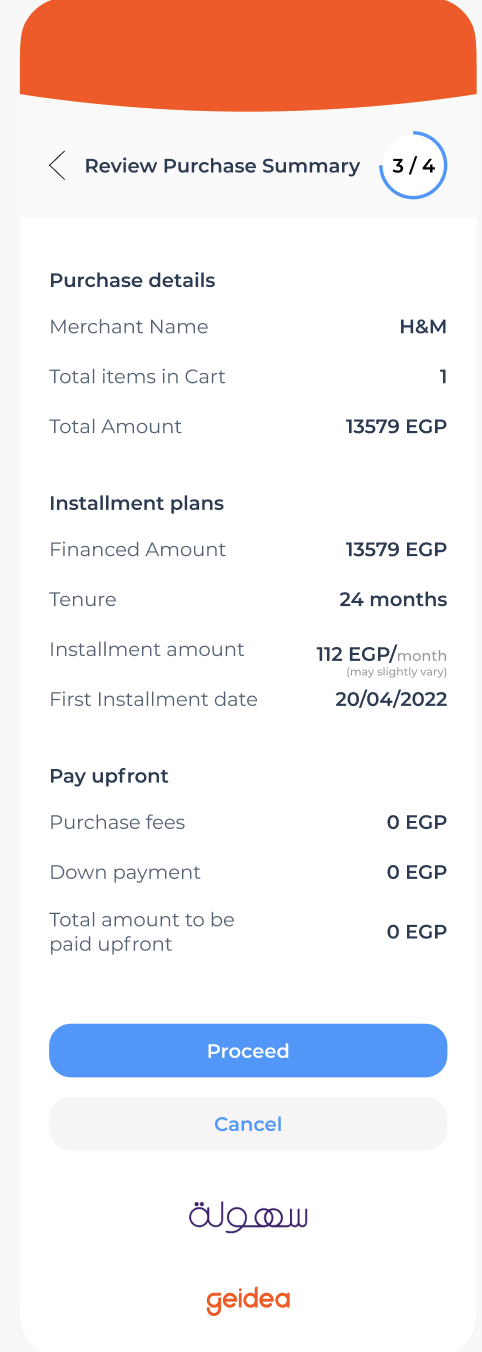
**return** (minAmount, maxAmount)

}

}

### 2.4.3 Step 3: Review Transaction

You need to call the function GeideaBNPLAPI.souhoolaReviewTransaction to retrieve all the details about the Souhoola transaction. A screen will be shown next for the user to see all details about transaction: Image 1 and pay the fee and down payment if needed as our example below Image 2:

Graphical user interface, application

Description automatically generated

GDSouhoolaReviewTransaction params:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | From user input step 1 |
| customerPin | String | From user input step 1 |
| totalAmount | Double | Transaction amount |
| currency | String | Transaction currency |
| downPayment | Double | downPayment from user |
| minimumDownPaymentTenure | Double | From user selected GDSouoolaPlanResponse |
| promoCode | String | From user selected GDSouoolaPlanResponse |
| approvedLimit | Double | From GDSouhoolaVerifyResponse |
| outstanding | Double | GDSouhoolaVerifyResponse |
| availableLimit | Double | GDSouhoolaVerifyResponse |
| minLoanAmount | Double | GDSouhoolaVerifyResponse |
| items | [GDBNPLItems] | From your Shopping cart items |

Example Swift:

GeideaBNPLAPI.souhoolaReviewTransaction(with: details, completion: { response, error **in**

completion(response,error)

})

With response GDSouhoolaReviewResponse:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String | used for reference and status |
| detailedResponseCode | String | used for reference and status |
| responseCode | String | used for reference and status |
| responseMessage | String | used for reference and status |
| language | String | used for reference and status |
| souhoolaTransactionID | String | Important for the next steps |
| Installments | [GDSouhoolaInstallmentPlansResponse] | The list of installmentPlans |
| totalInvoicePrice | Double | TotalInvoicePrice |
| loanAmount | String | Financed Amount |
| downPayment | Double |  |
| aministrativeFees | Double |  |
| netAdministraiveFees | Double |  |
| merchantName | String |  |
| cartCount | Double |  |
| promoCode | String |  |
| mainAdministrativeFees | Double |  |
| annualRate | Double |  |

When the user presses next button you need to call GeideaBNPLAPI.souhoolaInstallmentPlanSelected with GDSouhoolaInstallmentPlanSelected

Example swift:

GeideaBNPLAPI.souhoolaInstallmentPlanSelected(with: details, completion: { response, error **in**

completion(response,error)

})

GDSouhoolaInstallmentPlanSelected:

customerIdentifier: String?

customerPIN: String?

totalAmount = 0.0

currency: String?

merchantReferenceId: String?

callbackUrl: String?

billingAddress: GDAddress?

shippingAddress: GDAddress?

customerEmail: String?

returnUrl: String?

restrictPaymentMethods: Bool

paymentMethods: [String]?

items: [GDBNPLItem]?

bnplDetails: GDSouhoolaBNPLDetails?

cashOnDelivery: Bool = **(true if user selects cash on delivery)**

GDSouhoolaBNPLDetails params:

|  |  |  |
| --- | --- | --- |
| souhoolaTransactionID | String | Important for the next steps |
| totalInvoicePrice | Double | TotalInvoicePrice |
| loanAmount | String | Financed Amount |
| downPayment | Double |  |
| netAmininFees | Double |  |
| tenure | Int |  |
| **mainAdminFees** | Double |  |
| annualRate | Double |  |

Response GDSouhoolaInstallmentPlanSelectedResponse params:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String | used for reference and status |
| detailedResponseCode | String | used for reference and status |
| responseCode | String | used for reference and status |
| responseMessage | String | used for reference and status |
| language | String | used for reference and status |
| souhoolaTransactionID | String | Important for the next steps |
| nextStep | String | Values:  1. proceedWithDownPayment: next screen should be the payment screen  2. proceedWithBNPL: next screen should be the OTP screen step 5. This is the return value if the user doesn’t need to pay anything, or the user selects cashOnDelivery |
| orderId | String | Geidea Order id created. This must be use on the next API calls. |

If the user press backButton from this screen and the souhoolaTransactionID is already created you need to cancel Souhoola transaction for the user, otherwise it will stay active with following **GDSouhoolaCancelDetails**:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | From user input step 1 |
| customerPin | String | From user input step 1 |
| sohoolaTransacationID | String | From Step 2.4.3 |

Example Swift:

GeideaBNPLAPI.souhoolaCancel(with: details, completion: { response, error **in**

completion(response,error)

})

### Step 4: Down Payment if nextStep param is proceedWithDownPayment

* + - You can use either GeideaPaymentAPI.PayWithGeideaForm or GeideaPaymentAPI.Pay with the same OrderId received above in step 2.4.3

### 2.4.5 Step 4 or 5 if needed: OTP and confirm

Graphical user interface, text, application

Description automatically generated

You need to call the function GeideaBNPLAPI.souhoolaGenerateOTP which generates a SMS code sent to the user. See examples below for both functions

* + - You need to display a timer of 1 minute with “resend code” button disabled until the user will be able to press Resend Code. After that you need to call a new function GeideaBNPLAPI.souhoolaResendOTP as action for Resend Code button. User has the possibility to resend the code only once, Attempts Left is always 1, after this counter is 0 you can hide Resend Code button

Example swift:

GeideaBNPLAPI.souhoolaGenerateOTP(with: details, completion: { response, error **in**

completion(response,error)

})

GeideaBNPLAPI.souhoolaResendOTP(with: details, completion: { response, error **in**

completion(response,error)

})

GDSouhoolaOTPDetails params:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | From user input step 1 |
| customerPin | String | From user input step 1 |
| orderId | String | From Step 2.4.3 |
| sohoolaTransacationID | String | From Step 2.4.3 |

GDSouhoolaResendOTPDetails params:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | From user input step 1 |
| customerPin | String | From user input step 1 |

Both functions have the same GDSouhoolaBasic response:

|  |  |  |
| --- | --- | --- |
| detailedResponseMessage | String | used for reference and status |
| detailedResponseCode | String | used for reference and status |
| responseCode | String | used for reference and status |
| responseMessage | String | used for reference and status |
| language | String | used for reference and status |

When the user press finalize purchase you need to call the function GeideaBNPLAPI.souhoolaConfirm with GDSouhoolaConfirmDetails

Example Swift:

BNPLManager().souhoolaConfirm(with: details, completion: { response, error **in**

completion(response, error)

})

GDSouhoolaConfirmDetails params:

|  |  |  |
| --- | --- | --- |
| customerIdentifier | String | From user input step 1 |
| customerPin | String | From user input step 1 |
| orderId | String | From Step 2.4.3 |
| sohoolaTransacationID | String | From Step 2.4.3 |
| otp | String | User input OTP on screen |

Response will be the same response GDOrderResponse and GDErrorRespone both optional from the GeideaPaymentAPI documentation, including BNPL details in transaction

### 3 Geidea Form

### See the main documentation for PayWithGeideaForm including all BNPL providers and payment methods