

VERSION - 2.0

DATE – 18/09/2025



PRESENTED BY: AIRPAY PAYMENT SERVICES

AIRPAY PAYMENT SERVICES PVT LTD

GROUND FLOOR GYS INFINITY (PRIUS INFINITY), BAJI PRABHU DESHPANDE MARG, VISHNU

PRASAD SOCIETY, NAVPADA, NETAJI SUBHASH NAGAR, VILE PARLE, MUMBAI,

MAHARASHTRA 400057

iOS Kit

## ABSTRACT

This document specifies the technical aspect of integrating SwiftUI iOS Kit Package.

## DISCLAIMER

This documentation shall only be used for evaluating the planned services designated herein and may contain information that is privileged, confidential, Proprietary, or otherwise protected from disclosure. As a result, this document or content thereof shall not be disclosed, used, or duplicated, in whole or in part, for any purpose other than the Scope of Work assigned by airpay Payment Services to your company ("Recipient"). Upon completion of service or termination of service, the Recipient shall return all materials, including, without limiting the generality of the foregoing, all originals, copies, reproductions, and summaries of confidential information. Any unauthorized use or disclosure by the directors, officers, or employees of the Recipient shall be deemed to be unauthorized use or disclosure by the Recipient and the Recipient shall indemnify and hold harmless the airpay Payment Services from and against all damages, losses, costs, and expenses incurred because of such breach. airpay payment services may seek injunctive relief restraining the unauthorized disclosure or use of confidential information in addition to any other legal or equitable remedy otherwise available.

## VERSION HISTORY

VERSION #	IMPLEMENTED BY	REVISION DATE	APPROVED BY	APPROVAL DATE	REASONS
1.0	Tushar Khandekar	[20/05/2025]	-	[15/05/2025]	Initial Draft
2.0	Tushar Khandekar	[18/09/2025]	-	[18/09/2025]	Minor Changes

Table of Contents

ABSTRACT ..... 2

DISCLAIMER..... 3

VERSION HISTORY ..... 4

PURPOSE – ..... 6

INITIAL CONFIGURATION - ..... 6

STEPS – ..... 6

    SUBSCRIPTION TRANSACTION FLOW REQUEST ..... 7

    PRIVATE LOGIC KEY – ..... 8

    CHECKSUM LOGIC ..... 8

    RESPONSE CODE - FOLLOWING THE DELEGATE METHOD TO GET RESPONSE..... 9

CONFIGURING THE PROJECT ..... 10

CLEAN AND RUN..... 10

    DEPLOYMENT ..... 10

    REQUEST PARAMETERS ..... 12

    RESPONSE PARAMETER..... 15

## PURPOSE –

To create a new Single View Project in SwiftUI.

## INITIAL CONFIGURATION -

- Drag and drop the AirpayKit\_Simulator (includes dSYM file) for the simulator, or AirpayKit\_iPhone (includes dSYM file) for real devices, into your Xcode project.
- Locate and click to target settings -> General Tab -> Embedded Binaries. Click on '+' icon and select "Airpay\_Kit\_Swiftui.framework". Please select option to Embed without signing option.
- In AirpayDemoview , add following line at the top import Airpay\_Kit\_Swiftui

## STEPS –

### TRANSACTION FLOW REQUEST

Declare @StateObject var vm = AirpayDemoViewModel()

In the View add a fullScreenCover

```

.fullScreenCover(isPresented: $vm.isPresentingWebView) {
    let calprivateKey = vm.privateKeyNew()
    let calchecksumStr = vm.getCheckSum(privateKey: calprivateKey, currentDate:
vm.DateCalculation())
    if !calprivateKey.isEmpty, !calchecksumStr.isEmpty {
        let airpayViewModel = AirPayWebViewModel(
            envConfigString: "production",
            email: vm.email,
            phoneNumber: vm.phoneNumber,
            orderID: vm.orderID,
            amount: vm.amount,
            secretAPIKey: vm.kAirPaySecretKey,
            successURL: vm.successURL,
            userName: vm.kAirPayUserName,
            password: vm.kAirPayPassword,
            privateKey: calprivateKey,
            checksumStr: calchecksumStr,
            firstName: vm.firstName,
            lastName: vm.lastName,
            address: vm.address,
            city: vm.city,
            state: vm.state,
            country: vm.country,
            pincode: vm.pincode,
            mode: vm.mode,
            merchantID: vm.merchantID,
            customVariable: vm.customVariable,
            transactionSubType: vm.transactionSubType,
            currencyValue: vm.currencyValue,
            isoCurrency: vm.isoCurrency,
            wallet: vm.wallet,

```

```
        token: vm.token,
        delegate: vm.delegate,
        subscriptionType: vm.subscriptionType,
        subscriptionNextRunDate: vm.subscriptionNextRunDate,
        subscriptionPeriod: vm.subscriptionPeriod,
        subscriptionFrequency: vm.subscriptionFrequency,
        subscriptionAmount: vm.subscriptionAmount,
        subscriptionIsRecurring: vm.subscriptionIsRecurring,
        subscriptionRecurringCount: vm.subscriptionRecurringCount,
        subscriptionRetryAttempts: vm.subscriptionRetryAttempts,
        subscriptionMaxAmount: vm.subscriptionMaxAmount,
        client_id: vm.client_id,
        client_secret: vm.client_secret,
        merDom: vm.encodeDomainToBase64(from: vm.successURL),
        appName: vm.appName,
        colorCode: vm.colorCode
    )

    AirPayWebViewContainer(viewModel: airpayViewModel)
        .frame(height: UIScreen.main.bounds.height )
    }
}
```

## SUBSCRIPTION TRANSACTION FLOW REQUEST

For subscription transactions, the transaction request must include the txn sub type parameter with a value of 12 to indicate a subscription-based payment.

Additionally, the following parameters must be provided:

- Subscription Run Date – Specifies the date when the subscription transaction should be executed.
- Subscription Period – Defines the duration of the subscription (e.g., monthly, yearly).
- Subscription Frequency – Determines how often the transaction should occur within the subscription period.
- Subscription Max Amount – Specifies the maximum allowable amount for the subscription.
- Subscription Amount – Defines the fixed transaction amount for each subscription cycle.
- isRecurring – Indicates whether the transaction is recurring.
- Subscription Recurring Count – Specifies the total number of recurring transactions.
- Subscription Retry Attempts - Defines the number of times the transaction should be retried in case of failure.

These parameters must be included in the transaction request to enable subscription-based payments.

**PRIVATE LOGIC KEY –**

```
func privateKeyNew()->String {
    let sTemp = "\(kAirPaySecretKey)\("@")\((kAirPayUserName)\(":|:")\((kAirPayPassword))"
    let hashCode1 = "\(sTemp)"
    let sPrivateKey = hashCode1.sha256Hash()
    print("sPrivateKey: \(sPrivateKey)")
    return sPrivateKey
}
```

**CHECKSUM LOGIC**

Checksum logic is based on the validation of subscription data.

(In case of Subscription subtype value is "12" means it is the subscription transaction. Inside the txosubtype equals 12 condition contains the subscription stringAll elements.)

Code-

```
func getCheckSum(privateKey:String, currentDate:String) -> String {
    var siIndexVar = String()
    if subscriptionType != "12" {
        siIndexVar = ""
    }else{
        if subscriptionPeriod == "A" {
            siIndexVar =
            "\(subscriptionPeriod)\(subscriptionAmount)\(subscriptionIsRecurring)\(subscriptionRetryattempts)"
        }else{
            siIndexVar =
            "\(subscriptionNextRundate)\(subscriptionFrequency)\(subscriptionPeriod)\(subscriptionAmount)\(subscriptionIsRecurring)\(subscriptionRecurringcount)\(subscriptionRetryattempts)"
        }
    }
    let stringAll = "\(email)\(firstName)\(lastName)\(address)\(city)\(state)\(country)\(amount)\(orderId)\(siIndexVar)\(currentDate)"
    let sTemp2 = "\(kAirPayUserName)\("~::~~")\((kAirPayPassword))"
    let hashCode2 = "\(sTemp2)"
}
```



```

let sKey = hashCode2.sha256Hash()
let sAllData = "\(sKey)@\(\stringAll)"
let checksumStr = "\(sAllData.sha256Hash())"
return checksumStr
}

```

## RESPONSE CODE - FOLLOWING THE DELEGATE METHOD TO GET RESPONSE

### Code Snippet

**Note - Delegate method to call when payment finishes**

```

func finishPayment(success: Bool, response: [String: Any]?, error: Error?) {
    guard let data = response?["data"] as? [String: Any] else {
        print("+ Missing 'data' in response")
        return
    }

    let transactionStatus = data["transaction_payment_status"] as? String ?? ""
    let chmod = data["chmod"] as? String ?? ""
    var customerVPA = ""

    if chmod.lowercased() == "upi" {
        customerVPA = ":" + (data["customer_vpa"] as? String ?? "")
    }

    let apSecureHash = String(describing: data["ap_securehash"] ?? "")
    let transID = String(describing: data["orderid"] ?? "")
    let apTransactionID = String(describing: data["ap_transactionid"] ?? "")
    let transactionAmount = String(describing: data["amount"] ?? "")
    let statusMsg = data["bank_response_msg"] as? String ?? ""

    let strParam =
        "\({transID}): \({apTransactionID}): \({transactionAmount}): \({transactionStatus}): \({statusMsg}): \({merchantID}): \({kAirPayUserName}) \({customerVPA})"
    let crc32Str = strParam.data(using: .utf8)

```

```

let calculatedHash = crc32Str?.withUnsafeBytes {
    crc32(0, $0.bindMemory(to: Bytef.self).baseAddress, numericCast(crc32Str?.count ?? 0))
}
let sCRC = "\(calculatedHash ?? 0)"
let upiStatus = (sCRC == apSecureHash) ? "SECURE HASH MATCHED" : "SECURE HASH MIS-MATCHED"
// Update SwiftUI state
DispatchQueue.main.async {
    self.isPresentingWebView = false

    DispatchQueue.main.asyncAfter(deadline: .now() + 0.3) {
        self.alertTitle = transactionStatus
        self.alertMessage = "\(data) \(upiStatus)"
        self.showValidationAlert = true
    }
}
}
}

```

Note: In the above code we mentioned the Secure Hash Code validation and validated calculation secured hash value with server secure hash value.

## CONFIGURING THE PROJECT

Within the Info. List file, incorporate the specified query scheme. Please consult the provided screenshot for reference.

Key	Type	Value
Information Property List	Dictionary	(2 items)
Queried URL Schemes	Array	(8 items)
Item 0	String	phonepe
Item 1	String	gpay
Item 2	String	bhim
Item 3	String	paytmmp
Item 4	String	amazonToAlipay
Item 5	String	whatsapp
Item 6	String	credpay
Item 7	String	mobikwik
App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads	Boolean	YES

## CLEAN AND RUN

### DEPLOYMENT

#### Steps for Deployment

- Select the Project,
- Choose Target → Project Name → Select Build Phases → Press “+” → New Run Script Phase → Name the Script as “Remove Unused Architectures Script”. (only if runScript is required)

**Note-** Add below run script (only if runscript is required)

Here is the Sample Response Structure for reference –

### SAMPLE RESPONSE CODE

```
//Finish Payment = Optional( ["TXN_DATE_TIME": "", "PRI_ACC_NO_START": "", "RRN": "",  
"TRANSACTIONSTATUS": "200",  
"BANKNAME": "State Bank Of India", "CARDISSUER": "", "SURCHARGE": "3.54",  
"TXN_CURRENCY_CODE": "356",  
"STATUS": "200",  
"TRANSACTIONID": "",  
"CONTACTNO": "",  
"CARDTYPE": "",  
"CARDCOUNTRY": "",  
"EMITENURE": "",  
"AP_SECUREHASH": "",  
"CHMOD": "",  
"MERCHANTKEY": "",  
"TRANSACTIONAMT": "1.00",  
"BILLEDAMOUNT": "4.54",  
  
"STATUSMSG": "Success",  
"MERCHANT_NAME": "Test India", "ISRISK": "N", "MERCHANTTRANSACTIONID": "", "EMAIL": "",  
"SETTLEMENT_DATE": "",  
  
"TRANSACTIONVARIANT": "",  
"FULLNAME": "FNAME LNAME", "PRI_ACC_NO_END": "", "CUSTOMERVPA": "", "CUSTOMVAR": "test",  
"TXN_MODE": "LIVE",  
"TRANSACTIONREASON": "Success"]  
)
```

## REQUEST PARAMETERS

Method Name	Input Data Format	Data Length	Example
email	String	6-50	<a href="mailto:abc@gmail.com">abc@gmail.com</a>
phoneNumber	String	8-15	9898989898
firstName	String	1-50	Name
lastName	String	1-50	Name
address	String	4-255	Mumbai
city	String	2-50	Mumbai
state	String	2-50	Maharashtra
country	String	2-50	India
pincode	String	4-8	400001
amount	String	1-7	50
mode	String	0-15	Payment Mode (not required)
			ppc - prepaid card
			pg - payment gateway
			nb - Netbanking
			pgcc - Credit card
			pgdc - Debit card
			cash - Cash
			emi - EMI
			rtgs - RTGS
			upi - UPI
			btqr - Bharat QR
			payltr - Pay later
			va - Virtual account
			enach - eNACH
			remit - Remittance
chmod variable contains Payment Modes available for user. for e.g. If you want to show only Credit Card/Debit Card, then value of the chmod variable will be "pg". If you want Netbanking and Prepaid card then value of the chmod variable will be "nb_ppc".If you want to show all payment options activated for you at airpay, then leave this variable blank.			
checksumStr	String	1-100	Generate Checksum **
privateKey	String	1-100	Generate Private Key **
merchantID	String	1-15	Provide by airpay **
successURL	String	10-255	Success Url

iOS Integration Kit			
customVariable	String	1-100	Alphanumeric, Space, =
			Transaction SubType, type of transaction. (length 1- 100)
			1 - INR-auth-capture
			Allow merchants to first authorize a subscriber's card
			2 - INR-sale auth
			Confirms the cardholder's ability to pay
			3 - INR-Moto
			When a customer makes a card payment over the phone or through mail order
			4 - INR-Moto auth-capture
			When a customer makes a card payment over the phone or through mail order. Allow merchants to first authorize a subscriber's card.
			5 - INR-Sale-dcc
TransactionSubType	String	0-2	Dynamic currency conversion
			6 - INR-Dcc auth-capture
			Auth capture in dynamic currency conversion
			7 - INR-3 Months
			3 Months EMI
			8 - INR-6 Months
			6 Months EMI
			9 - INR-9 Months
			9 Months EMI
			10 - INR-12 Months
			12 Months EMI
			11 - INR-18 Months
			18 Months EMI
			12 - INR-SI
			Subscription in INR
			13 - INR-24 Months
			24 Months EMI
			36 - INR-36 Months
			36 Months EMI
			74 - INR-3 Months Debit
			3 Months EMI
			75 - INR-6 Months Debit

			iOS Integration Kit 6 Months EMI
			76 - INR-9 Months Debit
			9 Months EMI
			77 - INR-12 Months Debit
			12 Months EMI
wallet	String	1	Numeric
currencyValue	Integer	3	Eg. 356
isoCurrency	String	3	Eg. INR
orderId	String	20	Test123
subscriptionNextRunDate	String		Next subscription date (length 103 for enabling subscription) (required)
			mm/dd/yyyy date must be current date+1 (t+1)
subscriptionPeriod	String	1	Subscription period (length 1 for enabling subscription) (required) - D W M Y A
			Day/Week/Month/Year/Adhoc
subscriptionFrequency	String	1-999	Subscription frequency (length 1-999 for enabling subscription) (required)
subscriptionAmount	String	1-7	Subscription amount (length 1-6 .2 for enabling subscription) (required)
subscriptionIsRecurring	String	1	Is subscription recurring (length 1 for enabling subscription) (required)
subscriptionRecurringcount	String	1-999	Subscription Recurring Count (length 1-999 for enabling subscription and Is Subscription Recurring is Yes , if recurring count is 999 than subscription is set as never ending end date its apply only for enach transaction) (required)
subscriptionRetryattempts	String	1	Subscription retry attempts (length 1 for enabling subscription) (required)
subscriptionMaxamount	String	1-7	Maximum amount can charge, and greater than or equal to the amount
client_id	String	1-100	Provide by airpay **
client_secret	String	1-100	Provide by airpay **
appName	String	1-50	Application name eg: AirpayDemoApp
colorCode	String	1-10	Colour code that for top bar background eg: #FF5733

## RESPONSE PARAMETER

Field	Type	Description
transaction_payment_status	Alphanumeric	Transaction Payment Status (required) SUCCESS
		• TRANSACTION IN PROCESS
		• FAILED
		• DROPPED
		• CANCEL
		• INCOMPLETE
		• BOUNCED
		• NO RECORD
merchant_id	Numeric	MerchantID
TRANSACTIONID	Numeric	orderid you have sent to airpay system (required)
ap_transactionid	Numeric	airpay transaction reference number (required)
txn_mode	Alphanumeric	Transaction mode LIVE or Sandbox
chmod	Alphanumeric	Chanel of Payment done
amount	Numeric	Transaction amount (required)
currency_code	Alphanumeric	Payment Currency
transaction_status	Numeric	Transaction Payment Status (required)
		• Success - 200
		• Transaction in Process - 211
		• Failed - 400
		• Dropped - 401
		• Cancel - 402
		• Incomplete - 403
		• Bounced - 405
		• No Records - 503
message	Alphanumeric	Response message received from payment gateway (not required)
		• Success
		• Transaction in Process
		• Failed
		• Dropped
		• Cancel
		• Incomplete
		• Bounced
		• No Records
customer_name	Alphanumeric	Customer Name
customer_phone	Alphanumeric	Customer Phone
transaction_type	Numeric	Transaction Type (length 3, required)
		• Mandate approved, Auth - 310
		• Sale - 320
		• Capture - 330
		• Refund - 340
		• Chargeback - 350

		<ul style="list-style-type: none"> <li>Reversal - 360</li> </ul>
		<ul style="list-style-type: none"> <li>SaleComplete - 370</li> </ul>
		<ul style="list-style-type: none"> <li>SaleAdjust - 380</li> </ul>
		<ul style="list-style-type: none"> <li>TipAdjust - 390</li> </ul>
		<ul style="list-style-type: none"> <li>Sale+Cash - 400</li> </ul>
		<ul style="list-style-type: none"> <li>Cashback - 410</li> </ul>
		<ul style="list-style-type: none"> <li>Void - 420</li> </ul>
		<ul style="list-style-type: none"> <li>Release - 430</li> </ul>
		<ul style="list-style-type: none"> <li>Cashwithdrawal - 440</li> </ul>
risk	Numeric	If the transaction is risk
token	Alphanumeric	Token string(required)
transaction_time	Date	Transaction Time(required)
card_type	Alphanumeric	Type of Card Credit/Debit/Unknown
bank_response_msg	Alphanumeric	Response message from the bank
ap_SecureHash	AlphaNumeric	Secure hash generated by airpay (required)
		<ul style="list-style-type: none"> <li>If Channel is upi,</li> </ul>
		Hash generated by : crc32(TRANSACTIONID. : .APTRANSACTIONID. : .AMOUNT. : .TRANSACTIONSTATUS. : .MESSAGE. :
		.MID. : .USERNAME. : . CUSTOMERVPA);
		<ul style="list-style-type: none"> <li>Otherwise,</li> </ul>
		Hash generated by : crc32(TRANSACTIONID. : .APTRANSACTIONID. : .AMOUNT. : .TRANSACTIONSTATUS. : .MESSAGE. :
		.MID. : .USERNAME);



