

Merchant SDK Documentation-iOS

MVisaMerchantSDK

What's in MerchantSDK- $\{x.y.z\}$.zip :

Inside MerchantSDK- $\{x.y.z\}$.zip, there will be MVisaMerchantSDK- $\{x.y.z\}$ -DEBUG-SWIFT{swift version}.zip and MVisaMerchantSDK- $\{x.y.z\}$ -PRODUCTION-SWIFT{swift version}.zip zip files:

MVisaMerchantSDK- $\{x.y.z\}$ -DEBUG-SWIFT{swift version}.zip

The content of package are to be used during the development phase of the issuer application. This package contain binaries for universal (debug/development) architecture supporting simulator and device, but not eligible for app store submission.

This folder will have following data :

Name	Description
MVisaMerchantSDK.framework	Contains the public interface for integrating with MVisaMerchantSDK.framework
SampleApp	Sample App which use MVisaMerchantSDK.framework
CHANGE_LOG	Change log file to show all changes on MVisaMerchantSDK.framework for each version

MVisaMerchantSDK- $\{x.y.z\}$ -PRODUCTION-SWIFT{swift version}.zip

The content of package are to be used in the release version of the issuers application. The binaries in this package supports device only architectures and eligible for app store submission.

This folder will have following data :

Name	Description
MVisaMerchantSDK.framework	Contains the public interface for integrating with MVisaMerchantSDK.framework
CHANGE_LOG	Change log file to show all changes on MVisaMerchantSDK.framework for each version

How to integrate MVisaMerchantSDK.framework:

1. Xcode -> New -> Project -> Single View Application -> language Swift
2. Select Project -> Target -> General -> Embedded Binaries -> click plus (+) -> Add Another -> Add following frameworks (if needed select copy checkbox) -> MVisaMerchantSDK.framework
3. Open ViewController.swift and import MVisaSDK

```
import MVisaMerchantSDK
```

4. **Request Payment Flow:** Implement protocol MVisaRequestPaymentDelegate on ViewController

```
class ViewController: UIViewController, MVisaRequestPaymentDelegate
```

5. Add code to launch Request Payment Flow (Create QR screen)

```

// Launch Request Payment Flow
func launchRequestPaymentFlow() {
    // Create and configure dictionary data as per EMVco spec
    // This dictionary can hold all EMVco data tags
    var qrData = [String: AnyObject]()
    qrData[QRCodeTag.jcb1] = "77777777" as AnyObject
    qrData[QRCodeTag.currencyCode] = "356" as AnyObject
    qrData[QRCodeTag.pointOfInitiation] = "11" as AnyObject
    qrData[QRCodeTag.postalCode] = "401104" as AnyObject
    qrData[QRCodeTag.amex1] = "99999999" as AnyObject
    qrData[QRCodeTag.countryCode] = "IN" as AnyObject
    qrData[QRCodeTag.cityName] = "Mumbai" as AnyObject
    qrData[QRCodeTag.payloadFormatIndicator] = "01" as AnyObject
    qrData[QRCodeTag.merchantCategoryCode] = "6010" as AnyObject
    qrData[QRCodeTag.discover1] = "72222222" as AnyObject
    qrData[QRCodeTag.npciid1] = "91111111" as AnyObject
    qrData[QRCodeTag.mVisaMerchantID] = "47613617" as AnyObject
    qrData[QRCodeTag.unionPay1] = "66666666" as AnyObject
    qrData[QRCodeTag.tag17] = "33333333" as AnyObject
    qrData[QRCodeTag.masterCardPAN1] = "51111111" as AnyObject
    qrData[QRCodeTag.merchantName] = "Corner Store" as AnyObject
    qrData[QRCodeTag.additionalDataFieldTemplate] = [QRCodeTag.billID: "5001", QRCodeTag.mobileNumber: "741
1566609"] as AnyObject
    // Create a request with dictionary as parameter
    let request = MVisaPaymentRequest(withTagValueDictionary: qrData)
    // Launch request payment flow with the new request created
    MVisaMerchantSDK.launchRequestPaymentFlow(request: request, delegate: self)
}

```

6. Implement delegate method of MVisaPayMerchantDelegate to get the response :

```

// Callback method once request payment flow is completed

func requestPaymentFlowDidFinish(_ successful: Bool, withResponse response: MVisaRequestPaymentResponse?, withError error: MVisaError?) {
    MVisaMerchantSDK.endMVisaFlow()

    if successful {
        print("Request flow finished successfully")
    } else {
        print("Request flow finished with error")
        if let errDesc = error?.description {
            print(errDesc)
        }
        print(error?.errorCodes ?? "")
    }
}
}
}

```

7. Refund Flow: Implement protocol MVisaRefundPaymentDelegate on ViewController

```

class ViewController: UIViewController, MVisaRefundPaymentDelegate

```

8. Add code to launch Refund Flow

```

//Refund Payment SDK Launch

func launchRefundPaymentFlow() {
    let request = MVisaRefundPaymentRequest(last4:"1234", cardHolderName:"James Gawn", currencyCode:"840", refundAmount:"123456789.99")

    MVisaMerchantSDK.launchRefundPaymentFlow(request: request, delegate: self)
}

```

9. Implement delegate method of MVisaRefundPaymentDelegate to get the response :

```
//MARK: Refund Payment Flow Delegate

func refundPaymentFlowDidFinish(_ successful: Bool, withResponse response: MVisaRefundPaymentResponse?, withError error: MVisaError?) {
    MVisaMerchantSDK.endMVisaFlow()

    if successful {
        print("Refund flow finished successfully with refund amount")

        if let amount = response?.refundAmount {
            print(amount)
        }
    }
    else {
        print("Refund flow finished with error")

        if let errDesc = error?.description {
            print(errDesc)
        }
    }
}
```

MVisaMerchantSDK.framework API Detail:

MVisaMerchantSDK	
Classes	
	MVisaMerchantSDK Class that have all the public interfaces for integrating with the Refund, Request Payment flow.
	MVisaRefundPaymentDelegate Delegate that needs to be implemented for handling the detail returned by the SDK from the Refund Payment flow
	MVisaRequestPaymentDelegate Delegate that needs to be implemented for handling the detail returned by the SDK from the Request Payment flow
	MVisaCardDetails Designated initializer for storing the Card details of the customer
	MVisaConfig One time setup values are set in this class for SDK integration
	MVisaError If SDK had some issue or failed validation, this object will be returned to the delegate functions

	<p>MVisaRefundPaymentRequest</p> <p>A request class which will have refund related detail like last 4 digit of card number, card holder name, refund amount, currency code.</p> <p>Object of this class will be passed as part of refund flow.</p>
	<p>MVisaRefundPaymentResponse</p> <p>A response class which will have refund amount.</p>
	<p>MVisaPaymentRequest</p> <p>A request class which will have QR related detail like mVisa Merchant id, Merchant Name, City etc. Please check MVisaPaymentRequest class for other variables.</p> <p>Object of this class will be passed as part of Request Payment flow.</p>
MVisaMerchantSDK	
Invoke Refund flow	<p>func launchRefundPaymentFlow(request: MVisaRefundPaymentRequest, delegate: MVisaRefundPaymentDelegate)</p> <ul style="list-style-type: none"> parameter request: object of MVisaRefundPaymentRequest that captures data for invoking the refund flow like last 4 digit of card number, card holder name, refund amount, currency code. parameter delegate: An object must conform to the MVisaRefundPaymentDelegate protocol to ensure that results are properly passed back after completion of the flow. <p>func launchRefundPaymentFlow(request: MVisaRefundPaymentRequest, completionHandler: @escaping (_ successful: Bool, _ refundPaymentResponse: MVisaRefundPaymentResponse?, _ error: MVisaError?) -> Void)</p> <ul style="list-style-type: none"> parameter request: object of MVisaRefundPaymentRequest that captures data for invoking the refund flow like last 4 digit of card number, card holder name, refund amount, currency code. parameter completionHandler: This block handler will return <ul style="list-style-type: none"> -successful: a bool flag to identify if flow is successful or failure. -refundPaymentResponse: object of MVisaRefundPaymentResponse which will have refund amount -error: object of MVisaError, which will have error detail if any.
Invoke Request Payment flow	<p>func launchRequestPaymentFlow(request: MVisaPaymentRequest, delegate: MVisaRequestPaymentDelegate)</p> <ul style="list-style-type: none"> parameter request: object of MVisaPaymentRequest that captures data for invoking the Request Payment flow like mVisa Merchant id, Merchant Name, City etc. Please check MVisaPaymentRequest class for other variables. parameter delegate: An object must conform to the MVisaRequestPaymentDelegate protocol to ensure that results are properly passed back after completion of the flow. <p>func launchRequestPaymentFlow(request: MVisaPaymentRequest, completionHandler: @escaping (_ successful: Bool, _ error: MVisaError?) -> Void)</p> <ul style="list-style-type: none"> parameter request: object of MVisaPaymentRequest that captures data for invoking the Request Payment flow like mVisa Merchant id, Merchant Name, City etc. Please check MVisaPaymentRequest class for other variables. parameter completionHandler: This block handler will return <ul style="list-style-type: none"> -successful: a bool flag to identify if flow is successful or failure. -error: object of MVisaError, which will have error detail if any.

<p>Exit MVisaMerchantSDK calls and return control back to issuer app</p>	<pre>func endMVisaFlow(completionHandler: (() -> ())? = default)</pre> <ul style="list-style-type: none"> • parameter completionHandler: A completion handler that is invoked when control is successfully returned back to the host app. <p>Note: This should only be used after you have successfully handled a callback from the SDK. Calling this at any other time can cause unexpected behavior such as dismissing the SDK mid flow, forcing the user to restart the flow.</p>
<p>Configure and customize SDK during integration</p>	<pre>func setupOneTimeConfiguration(config: MVisaConfig)</pre> <ul style="list-style-type: none"> • parameter config: One Time Configuration details of mVisa merchant flow