Mobile Payment SDK Documentation For iOS

Table Of Contents

Chapter 1 Introduction

Introduction

This document is intended to be used as a reference in the planning and building of applications wishing to integrate eSewa SDK. This information should help accelerate the integration efforts of eSewa System with merchant application.

The focus of this document is to detail how partner applications establish connectivity to eSewa and outline the transaction process with verification process. A full and detailed description of the transactions and associated data elements is included.

This document is intended for partner merchant/clients seeking to integrate and transact with the eSewa. It should be used as a reference during the planning, building, and testing of such applications. eSewa enables partner merchant/clients to perform transaction initiated by customers having eSewa account in secure environment. The transaction amount is deposited into their eSewa merchant wallet or bank account.

Overview

General Description

The eSewa Mobile SDK enable native iOS app to accept eSewa payments.
The SDK supports only one use case for making payment – Single Payment (i.e. One payment per one user log in).

Transaction Flow

1. Client Mobile Application initiates eSewa Payment procedure by sending their **client_id** and **client_secret** along with **product/service name**, **product/service price**, **product_ id** and **callback-url** to sdk.

*__client_id__ and __client_secret__ values are provided by eSewa to its merchant/client.

2. SDK verifies merchant/client credentials with eSewa server, proceeds to next step if verification succeeds else freezes with appropriate error message.
3. User login in eSewa.
4. User makes payment if sufficient balance is available .
5. If successful payment is made, SDK receives (from eSewa server) and forwards proof of payment to client mobile application (in the form of transaction details shown in Appendix C) and at the same time, eSewa server also sends the same copy to merchant/client application server in the callback-url. (* This proof of payment is significant for payment verification).
6. Client/merchant verifies the transaction with eSewa through a verification api. This final verification procedure is strongly recommended before product/service delivery.
7. Deliver Product/Service to user.

[Note:
After the completion of login, if the device stays ideal for more than 1 minute before confirming the payment, the particular session will be time out and user have to re-login to confirm the payment.]

System Flow Diagram:

The interactions required to complete a transaction followed by verification process are shown below:

1. Client/Merchant app triggers sdk with required params as mentioned in step one of transaction flow.
2. Sdk (in communication with eSewa server) verifies merchant, proceeds to login eventually landing at payment confirmation.
3. User confirms payment. With payment being successful, eSewa server sends a proof of payment to Client/Merchant Server in their provided callback-url.
4. At the same time, the same copy is also sent to eSewa SDK.
5. eSewa SDK forwards the payment response to client/merchant mobile app.
6. Transaction verification is strongly recommended before the delivery of any product/service.

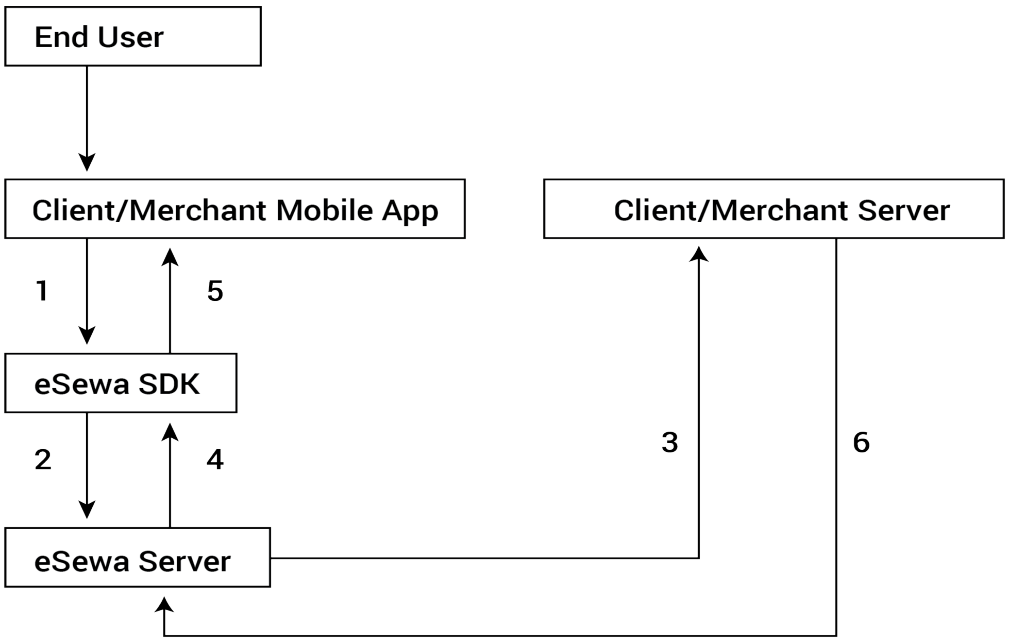

Fig: System Interaction

Call Back Url

Callback-url is an API exposed at merchant/client`server at which eSewa sends a copy of proof of payment after successful payment; the client/merchant must send a **callback-url** while initiating the payment through eSewa. The sent callback-url is later used by eSewa to send a copy of proof of payment after a payment is received. The callback-url must be an API with **Request Method POST**. The pattern in which eSewa sends the proof of payment is described in Chapter 2 (Implementation).

Transaction Verification

Clients using eSewa SDK are requested to go through final verification by comparing all the parameters proof of payment obtained by application through SDK and that obtained by client application server through eSewa. Clients/merchants are **strongly recommended** to send verification request for each successful payment with required parameters to eSewa Server before product/service delivery. This can be done through verification api which is shown in Chapter 2 (Implementation).

The partner merchant will receive transaction details on success or error message on failure response.

Chapter 2 Implementation

Implementation

Overview

eSewa integration on partner merchant's application is the process of implementing eSewa as payment option. The integration process itself is performed in two phases, namely:

1. Staging (Testing) Environment
2. Live (Production) Environment

- **Staging (Testing) Environment**

This phase is intended to test process flow, transactions and other integration results. A test merchant id & secret key along with a test eSewa Id & password will be provided to the partner merchant to test the payments through eSewa.

The primary objectives of this stage are:

Whether merchant requests are getting acknowledged by eSewa system or not.
Whether merchant is getting response of either a successful or failure transaction. Implementation and testing of verification process.

- **Live (Production) Environment**

After successful testing phase, the partner merchant will be provided with production environment merchant id & secret key.

Partner merchants have to modify the eSewa Environment from test to production to use live environment merchant credentials. Then, partner merchant will be able to start accepting payments using eSewa SDK.

Integration

Prerequisite:

Our SDK is built & compiled in Swift. As Swift doesn't have ABI Stability yet, SDK needs to be compiled for each Swift Version. So the client must let eSewa team know their XCode version and respective Swift version that they are using.

Add SDK to your project

Step1.
Drag & Drop provided eSewa SDK (Framework) inside your project. Make sure "copy items if needed" is selected and the framework is copied to your project.

Step 2.

2.1 Click on your Project -> Target -> Embedded Binaries
2.2 Tap + sign under Embedded Binaries, Locate your recently copied framework and press Add.

Step 3.

Now your eSewa SDK has been successfully added to your project. To complete the process, Build your project again.

Step 4.

To use eSewa SDK, you need to import it first in your **UIViewController** or any other classes.

Eg. **Import EsewaSDK**

Note: Implementation Details are explained in Appendix Section.

Calling on callback-url to send proof of payment

The api pattern in which eSewa server send the proof of payment after a successful payment made through eSewa is shown below:

Url: {callback-url sent by merchant while initiating payment }

Request method: **POST**

Headers: {

Content-Type: application/json
merchantId: {merchant_id}
merchantSecret: {merchant_secret_key}

}

Request Body:


```
  {
        "productId":"productId",
        "productName":"productName",
        "totalAmount":"10.0",
        "environment":"live",
        "code":"00",
        "merchantName":"merchantName",
        "message": {
              "successMessage":"Your transaction has been completed.",
              "technicalSuccessMessage":"Your transaction has been completed."
               },


       "transactionDetails": {
       "status":"COMPLETE",
       "referenceId":"00X3XHG",
       "date":"2016-11-17 12:38:18.0"
}
}
```


Transaction Verification API

In case, no request appears in the callback-url, **transaction(s)** can be still authenticated with following two APIs:

1. **API for transaction verification with reference Id:**

Request URI:
Live environment: https://esewa.com.np/mobile/transaction?txnRefId={txnReferenceId}
Test environment: https://ir-user.esewa.com.np/mobile/transaction?txnRefId={txnReferenceId}
Request Method : **GET**
Headers : {

Content-Type: application/json
merchantId: {merchant_id}
merchantSecret: {merchant_secret_key}

}

In **success**, the response is the transaction detail for queried reference Id as in **appendix B**

In **failure**, response is as:

```
{
   "code": 1,
  "message": "Transaction not found."
}
```

2. API for transaction verification with product Id & amount:

Live environment : https://esewa.com.np/mobile/transaction?productId={productId}&amount={amount}
Test environment: https://ir-user.esewa.com.np/mobile/transaction?productId={productId}&amount={amount}
Request Method: GET
Request Headers: {
Content-Type: application/json
merchantId: UEVRTQIcBksdDAoEGw5dCRAKFhYSFg==
merchantSecret: UEVRTQIcBksdDAoEGw5dCRAKFhYSFltHUw==

}

In **success**, the response is the transaction detail for queried reference Id as in **appendix B.**

In **failure**, response is as:

```
{
"code": 1,
"message": "Transaction not found."
}
```

Test merchant ID and merchant secret key and a test eSewa Id with password will be provided to the partner merchant to complete the transaction.

Error Cases and Handling

SDK handles all of the errors and displays appropriate alert to end user. Here are the error cases that can occur in each of the step during implementation of eSewa SDK implementation.

I. During Merchant Verification

a. Invalid Merchant Id or Secret Key
Invalid Merchant Id or Merchant Secret can cause this error. There are different Merchant Id & Secret for Development & Production Environment. Make sure you are providing valid Merchant Id & Secret for Development & Production Environment.

b. No Internet Connection
SDK shows dialog box with message "Internet not available" to end user.

II.During User Login

a. No Internet Connection

b. Invalid user name or password

c. Canceled by user
If end user presses Back Button or Cancel button then user is returned to the screen the sdk is initiated from.

III. During Payment Confirmation

a. No Internet Connection

b. Insufficient Balance
SDK compares the current balance of user with the total amount of the product or service. Payment request is sent if and only if the user has sufficient balance else the following message will be shown to user "Your balance is less than required total amount" and user will be bound to cancel the payment.

And if user has sufficient balance initially but balance is expended via other parties before confirming the payment, appropriate message is shown indicating that transaction is not completed.

c. Canceled by user
If end user presses Back Button or Cancel button then user is returned to the screen the sdk is initiated from.

d. Session Timeout

If the device is kept ideal for at least 2 minute then the particular session for payment expires and user has to re-process the payment again.

During Final Verification

This verification is done after SDK returns proof of payment after a successful transaction. Although payment is already completed, this verification plays a vital role to avoid fraud transactions. The verification can be done by comparing parameters of the response obtained by application from eSewa SDK with the transaction detail obtained at application server. If all the parameters are equivalent then the product/service is ready for delivery else it can be fraud transaction so you are requested to stop the product/service delivery. If mobile application server fails to receive proof of payment from eSewa Server or eSewa server fails to send proof of payment to application server then the request for verification can be sent later too. The process of sending verification request to eSewa is explained above in detail.

Chapter 3	Appendix

I. Appendix A

After importing and adding eSewa SDK in your xcode project (as said in Integration topic). We now move to implementing the payment process.

INTEGRATION

Swift Version 3.0 – 4.1

Step1: Initialize eSewa SDK. **Make sure sdk is an Instance Variable.**

```
// Keep it in class instance
let sdk: EsewaSDK!

// Use it wherever you want
sdk = EsewaSDK(inViewController: self, environment: .development, delegate: self)
```

Step 2: Start payment process by passing required parameters.

```
sdk.initiatePayment(merchantId: “Your Merchant ID”, merchantSecret: “Your Merchant Secret”, productName: "Your Product Name", productAmount: "Your Product Amount", productId: "Your Product Id", callbackUrl: "Your Callback URL")

Note: Provide valid Product Name, Product Amount, Product ID & Callback URL.
```

Step 3: Add two `EsewaSDKPaymentDelegate` methods to observe the success or failure of transaction.

```
func onEsewaSDKPaymentSuccess(info:[String:Any]) {
    // Called when the payment is success. Info contains the detail of transaction.
}

func onEsewaSDKPaymentError(errorDescription: String) {
     // Called when there is error with the description of the error.
}
```

Swift Version 2.3 : Every step is same as before except change in the syntax as follows:

STEP 1:

```
let s = UIStoryboard(name: "esewa", bundle: NSBundle(forClass: RequestViewController.self))
```

STEP 2:

```
let vc = s.instantiateViewControllerWithIdentifier("NetworkSearch") as! RequestViewController
```

STEP 3:

```
vc.sendForAuthentication("{merchant Id/clientId}", "{secret Key}", "{environment}",
"{productName}", "{productAmount}", "{productId}","{callbackUrl}")
```

STEP 4:

```
 presentViewController(vc, animated: true, completion: nil)
```

STEP 5:

```
NSNotificationCenter.defaultCenter().addObserver(self, selector:
#selector(ViewController.getTransactionDetails(_:)), name: "NotificationIdentifier", object: nil)}
```

STEP 6:

```
func getTransactionDetails(notification: NSNotification) {let detailsValue = notification.object as!
[String: AnyObject] print("details Value of transaction Details : \(detailsValue)") }
```

II. Appendix B

Proof of Payment Sample:


```
{
      "code":"00",
      "merchantName":"SDKTESTING ",
      "productName":"SomeProductName",
      "productId":" SomeProductId",
      "totalAmount":"500.0",

      "message":{
            "successMessage":"Your transaction has been completed.",
            "technicalSuccessMessage":"Your transaction has been completed."
                  },
     "transactionDetails":{
                     "status":"COMPLETE",
                  "referenceId":"00CPPAT",
                  "date":"2016-11-23 16:34:44.0"
                        }
}
```


[**Note**: reference Id is also called as transaction id]

Chapter 4 FAQ

For Merchants

i. **What is eSewa Mobile Payment SDK?**
eSewa Mobile payment SDK is the payment gateway for the native android & ios applications. It enables native android & ios application to easily accept eSewa payments.

ii. What are the items I will get for implementing eSewa mobile payment SDK?

You will be provided with .aar file of eSewa SDK & .framework file for iOS. Also you will be provided with documentation for integration, test merchant ID and merchant key and test user ID with password for test transaction.

iii. Can I use same SDK in multiple mobile applications?
No, one merchant Id and key is valid only for one application. However merchant can get other id and key from eSewa for integrating in multiple applications.

iv. Does eSewa provide payment integration in web applications?
Yes, eSewa also provides payment integration in web. You can refer our merchant team for its detail.

v. When Merchant servers must do validation with eSewa Server and Why?
In case, eSewa is unable to send proof of payment to client application server or client application server is unable to receive proof of payment from eSewa then merchant server must validate the payment to avoid fraud transactions and to ensure your account has actually received the expected payment. If you don't verify payments, you open yourself to fraud. Validation request can be sent to eSewa anytime.

vi. What is the settlement mechanism from eSewa to Merchant Bank Account?
On successful transaction the balance from eSewa user is transferred to merchant account and merchant can withdraw balance to respective bank account in free of cost i.e. without any withdrawal charges.

vii. How eSewa provides support up on receiving any trouble because of eSewa mobile payment SDK?

eSewa provides 24 hours of customer support service . You can contact us in case of any inconvenience.

viii. How merchant can start testing eSewa mobile payment SDK?

Test merchant ID and merchant secret key and a test eSewa Id with password will be provided to the partner merchant to complete the transaction. Test transaction can be made by setting the test Environment during setting SDK Configuration.

ix. What if Merchant server received payment confirmation only from Mobile application?

There may be the possibility of two cases:

a. eSewa Server not being able to send the proof of payment in callback-url or merchant server not being able to receive the proof of payment.

b. Fraud case

So, the first case must be cross checked for which the merchant server should call eSewa server with the given product Id and product amount to know if the transaction for the sent id and amount of product/service has been processed or not (as explained in Chapter 2).

If yes, the response is a proof of payment (i.e. transaction detail as show in appendix C) and if no, "no transaction found" error message is returned which may be the case b (Fraud) so delivery of product/service must be stopped and one should contact eSewa Team as soon as possible.

x. What are you supposed to do if the payment amount confirmed by eSewa and transaction amount requested by Mobile application are not equal?

This may be the case of fraud transaction so in this case service/product delivery must be stopped and one should contact eSewa as soon as possible.

xi. Is eSewa SDK mobile platform dependent? Can we implement this on Android and iOS or do we have to take different SDK for different platform?

eSewa SDK is platform dependent and SDK for both android and iOS platform is available. If you want to integrate eSewa SDK in your android application as well, You can consult with our merchant team.

xii. What is the time per session to make successful payment?

User must complete login and payment process within 5 minutes or else will have to re-initiation the payment process from the beginning.

For Users

i. Should I have eSewaID and its credential to make payment by eSewa in Mobile application?

Yes users should be registered in eSewa and must have user ID (i.e. eSewa Id) and password for the completing the transaction.

ii. How can I initiate payment by eSewa in Mobile application if I don't have eSewaID?

User must register to eSewa using **www.esewa.com.np** or bye typing REG and send it to 32121 via SMS.

iii. How can I load balance in eSewa for the payments?

User can load balance by transferring balance from our partner banks or by counter deposit directly from some banks or deposit balance through eSewa Zone/ Point.

iv. Can I ensure that my eSewa ID and credential is safe to payment by eSewa in third party Mobile application?

Your **eSewa ID** and password are safe and secure as it travels in encrypted form and over HTTPS channel.