

CCAvenue MOBILE SDK FOR IOS

INTEGRATION GUIDE

VERSION 2.0

AVENUES WORLD FZ LLC

Dubai, United Arab Emirates

DATE: 2ND MAY 2018



Table of Contents

Introduction	2
Steps to Integrate	2
Download & Unzip.....	2
Upload GetRSA.jsp & ResponseHandler.jsp on Merchant Server	2
Include the SDK in Xcode Project.....	2
Installation	3
Finish SDK and get Response from SDK.....	5

INTRODUCTION

The purpose of this integration guide is to provide a step-by-step guide to integration the CCAvenue Mobile SDK for iOS.

STEPS TO INTEGRATE

Download & Unzip

1. Download CCAvenueSDKiOS.zip file for iOS. This contains the framework file, bundle resources file, response handler file (ResponseHandler.jsp), RSA file (GetRSA.jsp) and a demo app (Swift & Objective-C).
2. Unzip the downloaded CCAvenueSDKiOS.zip file anywhere on the file system.

Upload GetRSA.jsp & ResponseHandler.jsp on Merchant Server

1. GetRSA.jsp kept on the merchant server for fetching the RSA Public Key and send the file path URL with request.
2. Also ResponseHandler.jsp kept on the merchant server for the response will be sent by the CCAvenue server to the merchant server on the return URL that was configured at the URL that was sent in the request. Merchant should then decrypt the response, which can then be parsed to get the all transaction details and will send back to the merchant app.

Include the SDK in Xcode Project

1. Drag the CCAvenueSDK.framework to Frameworks & CCAvenueSDK.bundle under project in project navigator.
2. Create a new group Frameworks if it does not exist.
3. Choose Create groups for any added folders. Make sure to select Copy files if needed.
4. For Swift Projects add the following:
#import <CCAvenueSDK/InitialViewController.h> to the Bridging-Header.h

Note:

Ensure linked once in the Linked Framework and Libraries or just drag the CCAvenueSDK.framework to Embedded Binaries in the general tab in the project settings.

In Xcode, secondary click your project's '.plist' file and select Open As -> Source Code. Insert the following XML source code snippet into the body of your file just before the final.

```
</dict>element
    <key>NSAppTransportSecurity</key>
    <dict>
        <key>NSAllowsArbitraryLoads</key>
        <true/>
    </dict>
```

Installation

1. Import the CCAvenue Library

```
#import <CCAvenueSDK/InitialViewController>
```

2. Initialize InitialViewController and pass values in Initializer for SDK.

ObjectiveC:

```
InitialViewController *initVC = [InitialViewController alloc] initWithOrderId:@\"1234\"
                                merchantId:@\"79562\"
                                accessCode:@\"ABHGH784GFDE\"
                                amount:@\"10.00\"
                                currency:@\"AED\"
                                rsaKeyUrl:@\"www.abc.ae/RSA.jsp\"
                                redirectUrl:@\"www.abc.ae/Redirect.jsp\"
                                cancelUrl:@\"www.abc.ae/Cancel.jsp\"
                                showAddress:@\"Y\"
                                billingName:@\"ABC\"
                                billingAddress:@\"Star City\"
                                billingCity:@\"Dubai\"
                                billingState:@\"Dubai\"
                                billingCountry:@\"United Arab Emirates\"
                                billingTel:@\"+971 1234567890\"
                                billingEmail:@\"test@gmail.com\"
                                deliveryName:@\"XYZ\"
                                deliveryAddress:@\"Internet City\"
                                deliveryCity:@\"Dubai\"
                                deliveryState:@\"Dubai\"
                                deliveryCountry:@\"United Arab Emirates\"
                                deliveryTel:@\"+971 9876543210\"
                                promoCode:@\"PROMO1270\"
                                merchant_param1:@\"Param 1\"
```

```

merchant_param2:@"Param 2"
merchant_param3:@"Param 3"
merchant_param4:@"Param 4"
merchant_param5:@"Param 5"
useCCPromo:@"Y"];

```

Swift:

```

let initialVC = InitialViewController.init(OrderId:"1234" ,
merchantId:"79562" ,
accessCode:"ABHGH784GFDE",
amount:"10.00",
currency:"AED",
rsaKeyUrl:"www.abc.ae/RSA.jsp",
redirectUrl:"www.abc.ae/Redirect.jsp",
cancelUrl:"www.abc.ae/Cancel.jsp",
showAddress:"Y",
billingName:"ABC",
billingAddress:"Star City",
billingCity:"Dubai",
billingState:"Dubai",
billingCountry:"United Arab Emirates",
billingTel:"+971 1234567890",
billingEmail:"test@gmail.com"
deliveryName:"XYZ",
deliveryAddress:"Internet City",
deliveryCity:"Dubai",
deliveryState:"Dubai",
deliveryCountry:"United Arab Emirates",
deliveryTel:"+971 9876543210"
promoCode:"PROMO1270"
merchant_param1:"Param 1"
merchant_param2:"Param 2"
merchant_param3:"Param 3"
merchant_param4:"Param 4"
merchant_param5:"Param 5"
useCCPromo:"Y")

```

3. Initialize InitViewContollerDelegate for last class where you start SDK.

ObjectiveC:

```

@interface ViewController () <InitialViewControllerDelegate>

```

Swift:

```

class ViewController: UIViewController, InitialViewControllerDelegate

```

4. Assign an instance of the class that's adopted the delegate protocol to its delegate property.

ObjectiveC:

```
initVC.delegate = self;
```

Swift:

```
initialVC?.delegate = self;
```

5. Present Initial controller of SDK to process payment.

ObjectiveC:

```
[self presentViewController:initVC animated:true completion:nil];
```

Swift:

```
self.present(initialVC!, animated: true, completion: nil);
```

Finish SDK and get Response from SDK

1. Call delegate method of InitialViewConrollerDelegate.

ObjectiveC:

```
-(void)getResponse:(NSMutableDictionary *)responseDict_  
{  
    NSLog(@"Response in merchant app :- %@",responseDict_);  
  
    // here SDK will closed and you get response in Dictionary and use for  
    further process  
}
```

Swift:

```
self.present(initialVC!, animated: true, completion: nil);  
  
func getResponse(_ responseDict_: NSMutableDictionary!)  
{  
    print("Response in merchant app :- \(responseDict_)");  
    // here SDK will closed and you get response in Dictionary and use for  
    further process  
}
```

Note: Currently only Portrait orientation mode is supported within the CCAvenue Mobile SDK.