**Integration Steps for iOS**

**Getting Started:**

Follow these simple steps to integrate with EBS Gateway on IOS.

**SDK Installation Prerequisites**

Xcode 5.x. , Xcode 6.x. and Xcode 7.x.

**EBS PG Prerequisites:**

Make sure that you have obtained following parameters from EBS admin

* Merchant SECRET\_KEY
* Merchant ACC\_ID

Following parameters from Mercant Side

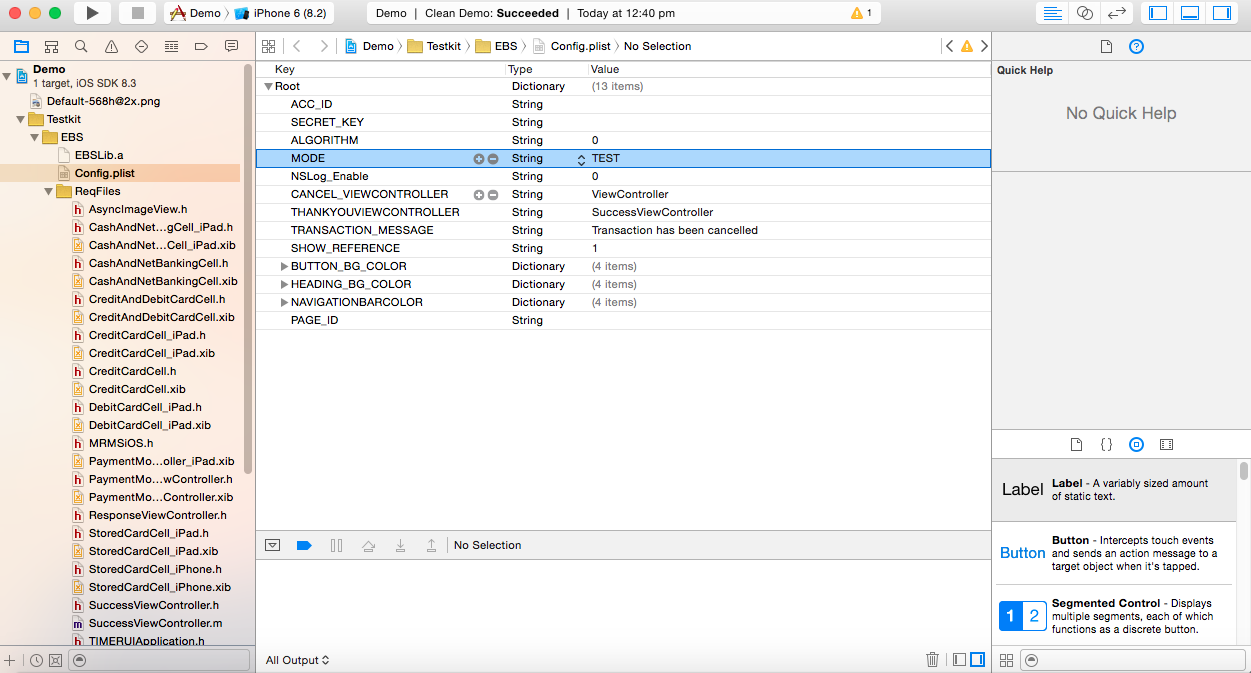
* Merchant REFERENCE\_NO(Generated by Merchant on each payment)

Note: Above three Parameters are Mandatory.

**TEST Mode:**

Please ensure the following values are available or checked in the SDK:

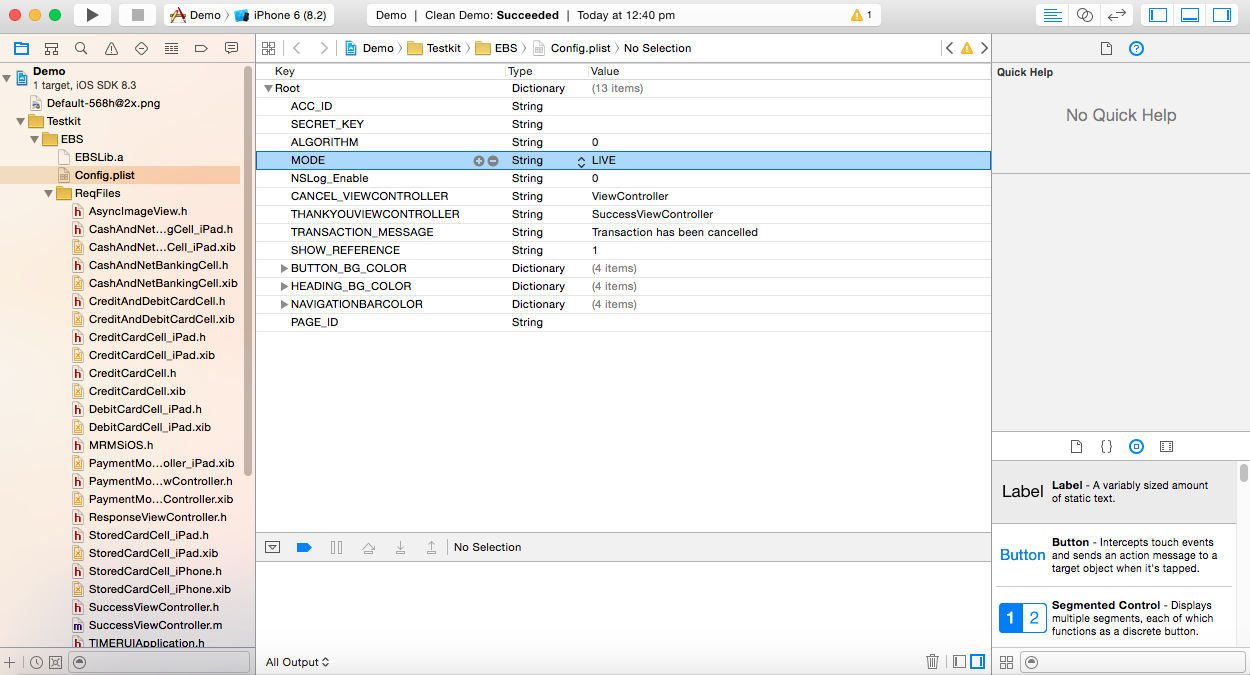
Screenshot



**LIVE Mode:**

Please ensure the following values are available or checked in the SDK:

Screenshot



In **Config.plist** file add the below values.(2015-02-04.zip -> iOS Library ->ReqFiles-> Config.plist)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Parameter** | **Description** | **Type** | **Min** | **Max** | **Mandatory** |
| **ACC\_ID** | Your Acc\_ID has to be obtained from EBS. | Numeric | - | - | YES |
| **SECRET\_KEY** | Your Acc\_ID has to be obtained from EBS. | string | - | - | YES |
| **ALGORITHM** | Algorithm md5,sha1 and sha512 are available, You can configure with 0 or 1 or 2 respectively. | Numeric | - | - | YES |
| **MODE** | TEST and LIVE modes are available | string | - | - | YES |
| **NSLog\_Enable** | NSLog can be enabled or disabled.**1** for Enableand **0 f**or Disable | Numeric | 0 | 1 | NO |
| **CANCEL\_VIEWCONTROLLER** | If the payment is Cancelled or Session expired, You have to configure in which view controller it should be pushed. | string | - | - | YES |
| **THANKYOUVIEWCONTROLLER** | Once the payment is success please configure your own view controller where you can get the response details. | string | - | - | YES |
| **TRANSACTION\_MESSAGE** | You can configure the failed message for cancelled and timed out transactions | string | - | - | NO |
| **SHOW\_REFERENCE** | You can show or hide the amount and Reference number for cancelled and TimedOut Transaction. **1** for Enableand **0 f**or Disable | numeric | 0 | 1 | NO |
| **BUTTON\_BG\_COLOR** | Merchant can customize it using RGB color with alpha value. | numeric | 0 | 255.0 | NO |
| **HEADING\_FONT\_COLOR** | Merchant can customize it using RGB color with alpha value. | numeric | 0 | 255.0 | NO |
| **HEADING\_BG\_COLOR** | Merchant can customize it using RGB color with alpha value. | numeric | 0 | 255.0 | NO |

**Features:**

* Direct credit/debit card (CC, DC) or netbanking payments (NB) and Cash cards.
* Saving Credit/Debit cards into user's account for easier future payments.
* Saved card details for future transaction.

**Installation From source code**

* Get the latest source code from EBS:
* Unzip 2015-02-04.zip
* You could see 2 folders in it

1. Demo
2. Reqfiles

**1.Steps to run EBS Demo Application**

**Demo folder is a test application** provided by EBS to understand the flow of EBS Payment Gateway.You can do a transaction using your Account ID and secret key.

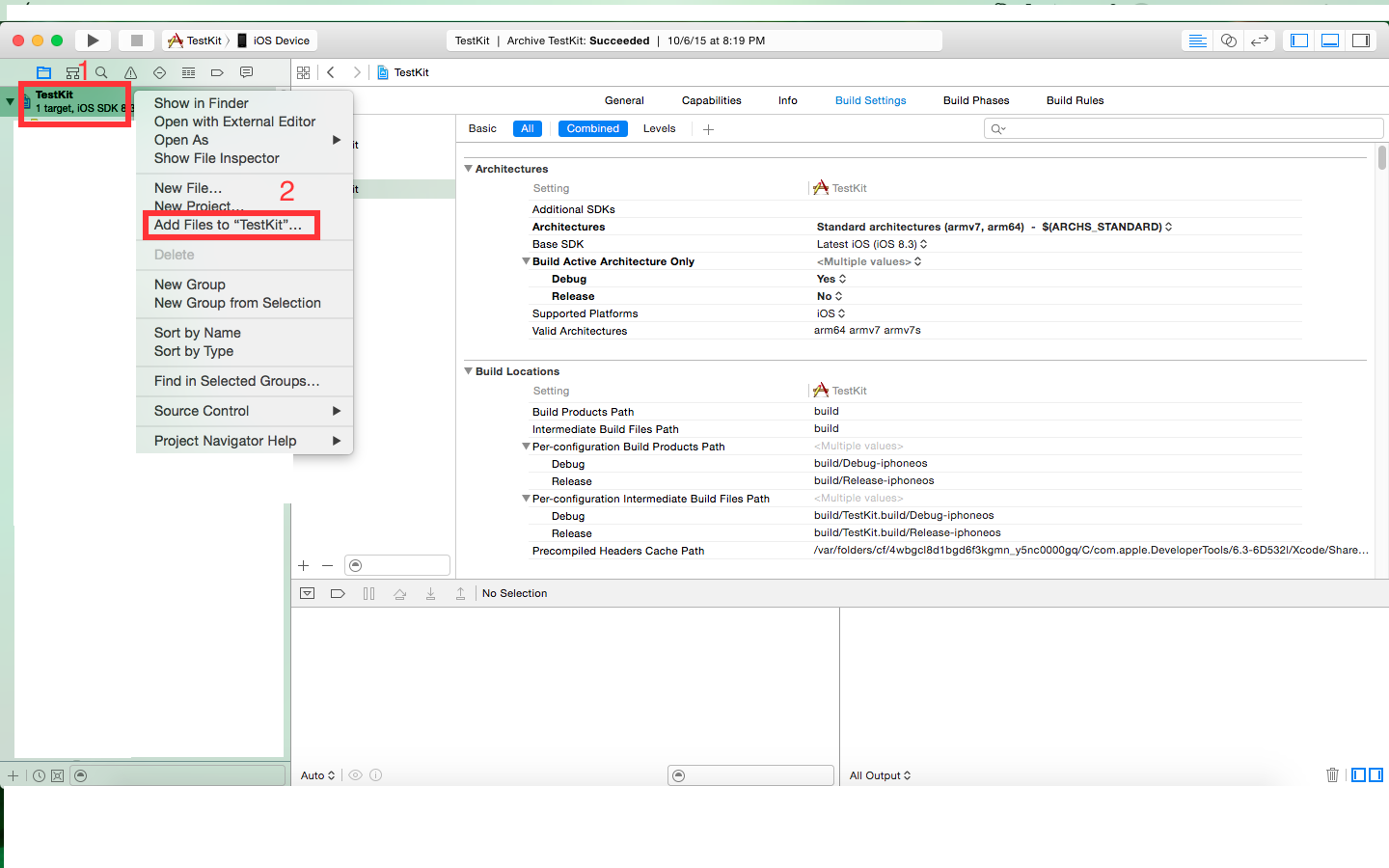
**2.Steps to Integrate in your App**

ReqFiles folder is the library which should be added in the project to make the Transaction threw

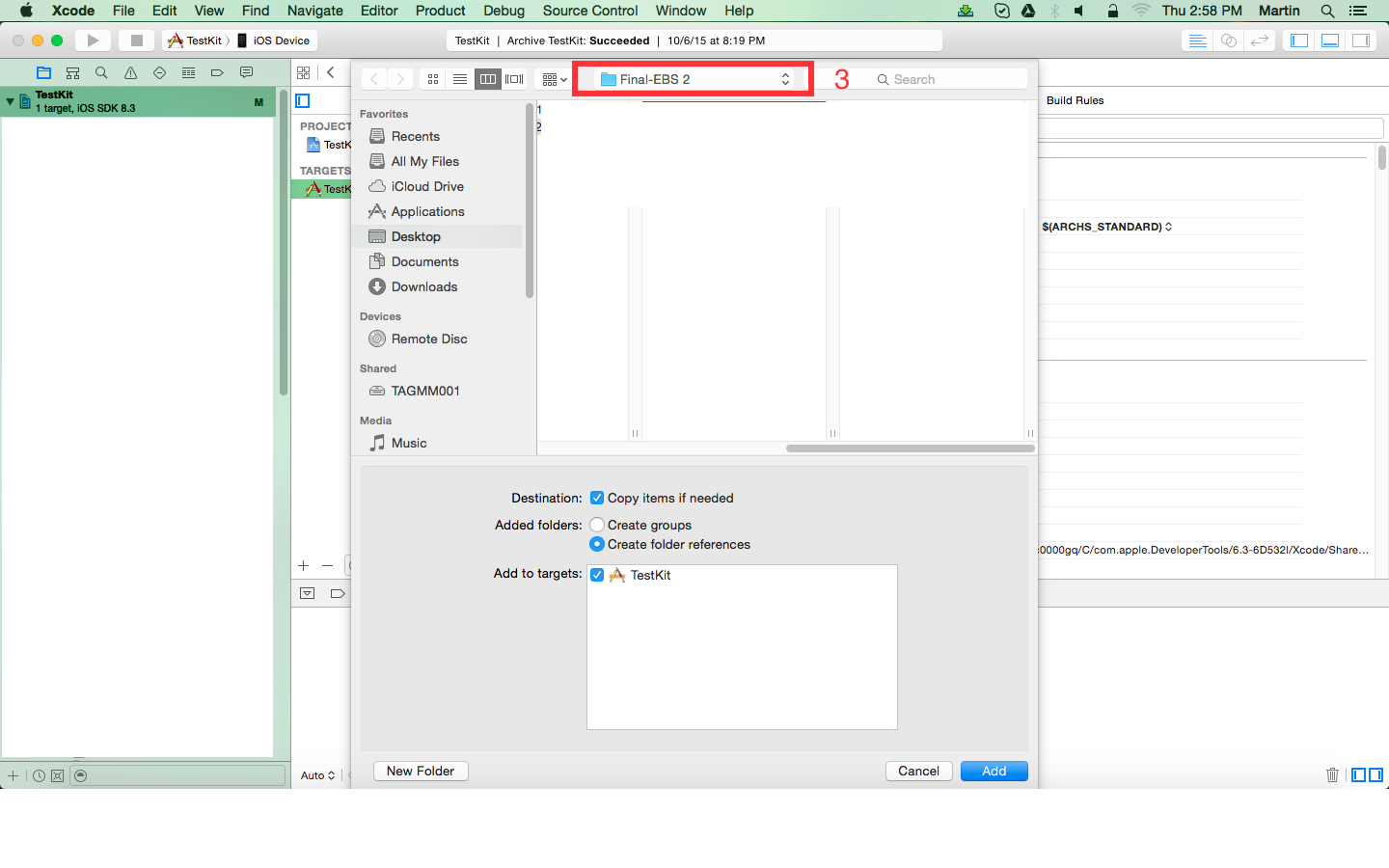
ReqFiles folder has to be added in your project.

**How to Add files in your App**

***Step 1:*** Select your project at top left and right click as shown in screen shot.



***Step 2:*** Browse to the Path where you have downloaded your ReqFiles as shown in screenshot. And select Add(Highlighted in blue).



Files will be added in your project.

**Xcode integration**

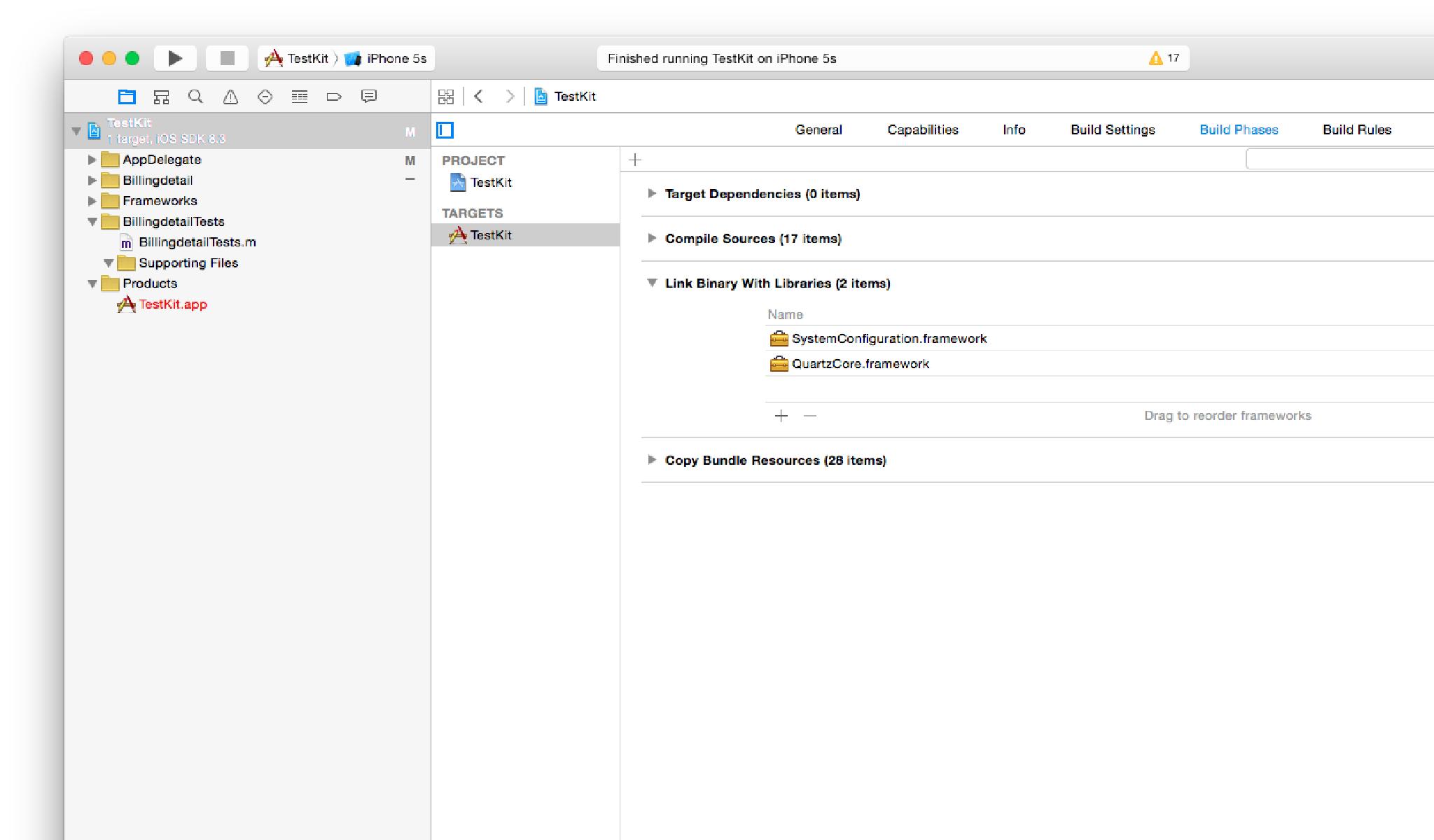
* Project Targets -> Build Settings -> Other Linker Flags -> Click + button and
* type **-ObjC**
* Again click + button and type -> **-all\_load**

**Add below frameworks**

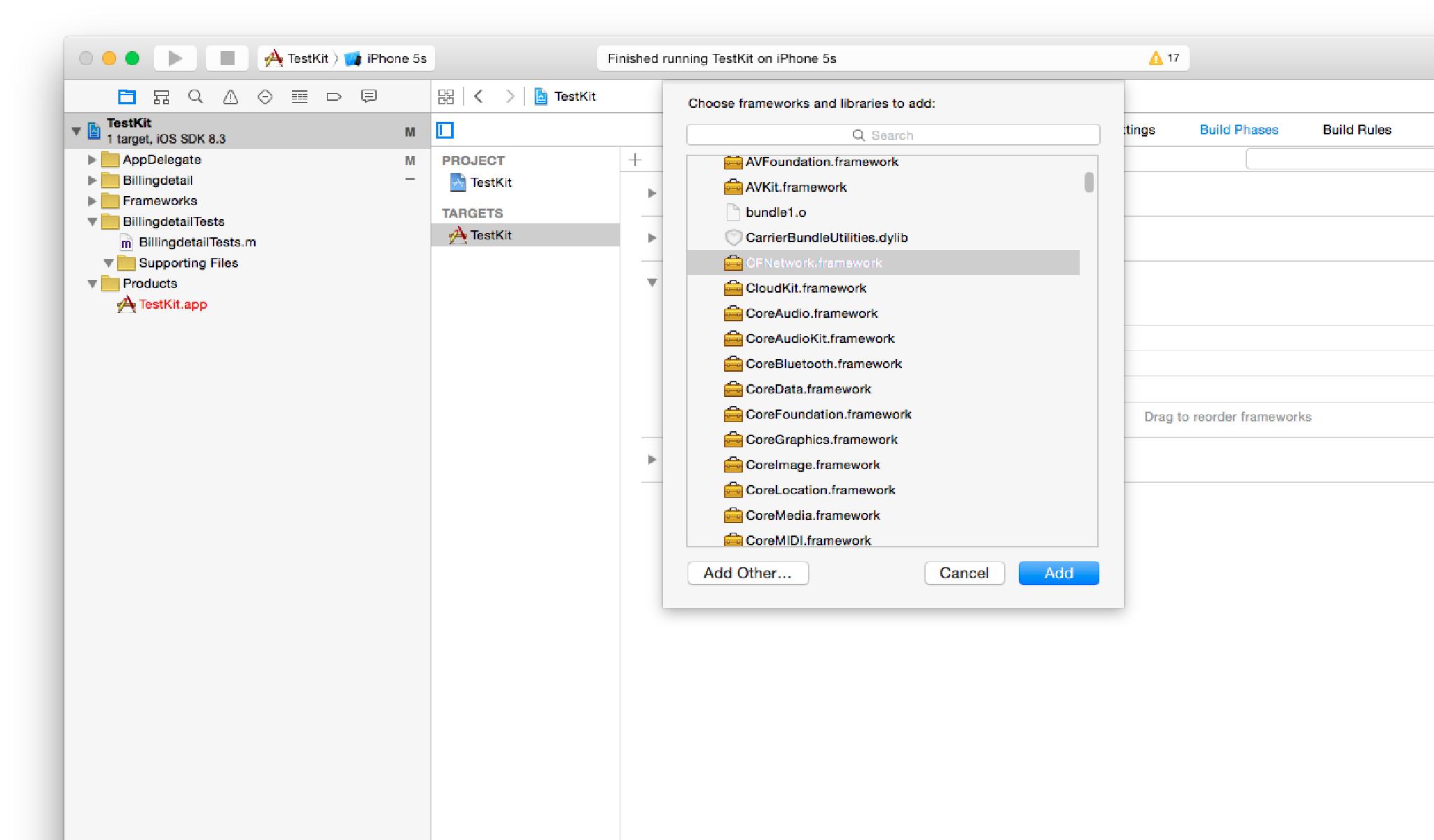
* MobileCoreServices.framework,
* CFNetwork.framework,
* SystemConfiguration.framework

**How to add a framework?**

***Step 1:***Select -> Target -> Build Phases -> Click Link binary with libraries -> And select +



**Step 2:** Type the frameworks need to be integrated and select Add.



In your **Checkout Action** View Controller

Sample code:

#import <SystemConfiguration/SystemConfiguration.h>

#import <netdb.h>

#import "MRMSiOS.h"

#import "PaymentModeViewController.h"

// You can have your own method declaration but paymentAmount and reference\_no are Mandatory.

-(IBAction)BuyAction:(id)sender {

PaymentModeViewController \*paymentView=[[PaymentModeViewController alloc]init];

paymentView.strSaleAmount=[NSString stringWithFormat:@"%.2f",***MERCHANT\_PRICE***];

paymentView.reference\_no= ***MERCHANT\_REFERENCENO***;

*//NOTE:* ***MERCHANT\_PRICE and MERCHANT\_REFERENCENO*** *has to be given by Merchant developer*

NSUserDefaults \*defaults=[NSUserDefaults standardUserDefaults];

[defaults setObject:[NSString stringWithFormat:@"%.2f",***MERCHANT\_PRICE***] forKey:@"strSaleAmount"];

[defaults setObject:***MERCHANT\_REFERENCENO*** forKey:@"reference\_no"];

[defaults synchronize];

paymentView.descriptionString = @"Test Description";

paymentView.strCurrency =@"INR";

paymentView.strDisplayCurrency =@"USD";

paymentView.strDescription = @"Test Description";

paymentView.strBillingName = @"Test";

paymentView.strBillingAddress = @"Bill address";

paymentView.strBillingCity =@"Bill City";

paymentView.strBillingState = @"TN";

paymentView.strBillingPostal =@"625000";

paymentView.strBillingCountry = @"IND";

paymentView.strBillingEmail =@"test@testmail.com";

paymentView.strBillingTelephone =@"9363469999";

paymentView.strDeliveryName = @"";

paymentView.strDeliveryAddress = @"";

paymentView.strDeliveryCity = @"";

paymentView.strDeliveryState = @"";

paymentView.strDeliveryPostal =@"";

paymentView.strDeliveryCountry = @"";

paymentView.strDeliveryTelephone =@"";

//If you want to add any extra parameters dynamically you have to add the Key and value as we //mentioned below

// [dynamicKeyValueDictionary setValue:@"savings" forKey:@"account\_detail"];

// [dynamicKeyValueDictionary setValue:@"gold" forKey:@"merchant\_type"];

// paymentView.dynamicKeyValueDictionary = dynamicKeyValueDictionary;

[self.navigationController pushViewController:paymentView animated:NO];

}

In **main.m**

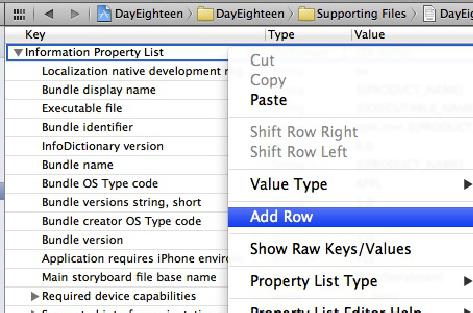
#import "TIMERUIApplication.h"

Replace the below code on return.

**return UIApplicationMain(argc, argv, NSStringFromClass([TIMERUIApplication class]), NSStringFromClass([AppDelegate class]));**

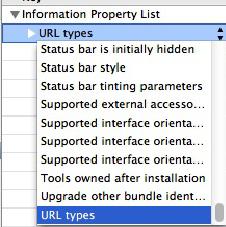
**Configuration of Custom URL Scheme:**

***Step 1:***

Expand the Supporting Files folder and open the App Name-Info.plist file. This file contains some application settings and is where to put our custom URL scheme. Right click on the Information Property List row at the top and go to Add Row:

***Step 2:***

In the drop down that appears, select URL types:



***Step 3:***

Expand the arrows next to URL types and Item 0 and for URL identifier, enter the bundle of your app (for eg com.test.yourAppName)



***Step 4:***

Now, right click on Item 0 and choose Add Row. From the drop down that appears, select URL Schemes:



***Step 5:***

Expand the URL Schemes arrow and for Item 0, enter the text for the URL scheme you want. So if you wanted the URL to open your app to be testkit://myapp you would enter your appname here.



Please add the below methods in Your **AppDelegate**

Sample code:

**#import "ResponseViewController.h"**

- (BOOL)application:(UIApplication \*)application handleOpenURL:(NSURL \*)url

{

if (!url)

{

return NO;

}

NSArray \*parameterArray = [[url absoluteString] componentsSeparatedByString:@"?"];

UIStoryboard \*mainStoryboard = [UIStoryboard storyboardWithName:[[NSBundle mainBundle].infoDictionary objectForKey:@"UIMainStoryboardFile"] bundle:[NSBundle mainBundle]];

ResponseViewController \*controller = (ResponseViewController\*)[mainStoryboard instantiateViewControllerWithIdentifier: @"ResponseViewController"];

controller.transaction\_id=[parameterArray objectAtIndex:1];

UINavigationController \*navigationController = (UINavigationController \*)self.window.rootViewController;

[navigationController pushViewController:controller animated:YES];

return YES;

}

- (BOOL)application:(UIApplication \*)application openURL:(NSURL \*)url sourceApplication:(NSString \*)sourceApplication annotation:(id)annotation

{

if (!url)

{

return NO;

}

NSArray \*parameterArray = [[url absoluteString] componentsSeparatedByString:@"?"];

UIStoryboard \*mainStoryboard = [UIStoryboard storyboardWithName:[[NSBundle mainBundle].infoDictionary objectForKey:@"UIMainStoryboardFile"] bundle:[NSBundle mainBundle]];

ResponseViewController \*controller = (ResponseViewController\*)[mainStoryboard instantiateViewControllerWithIdentifier: @"ResponseViewController"];

controller.transaction\_id = [parameterArray objectAtIndex:1];

UINavigationController \*navigationController = (UINavigationController \*)self.window.rootViewController;

[navigationController pushViewController:controller animated:YES];

return YES;

}

**After Payment Completed:**

**Webview to SuccessViewController :**

The landing page(SuccessViewController) could be configurable by client and neccessary changes to be added as given below

In **SuccessViewController** if you want to navigate to another view controller you have to use below code:

Sample code:

**SuccessVIewController.h**

NSDictionary \*dict;

@property(nonatomic,retain)NSMutableDictionary \*jsondict;

@property(nonatomic,retain)IBOutlet UIScrollView \*scroll;

In **SuccessViewController.m** below methods are mandatory in **viewDidLoad**

**[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(ResponseNew:) name:@"JSON\_NEW" object:nil];**

**[[NSNotificationCenter defaultCenter] postNotificationName:@"JSON\_DICT" object:nil userInfo:nil];**

***//Please add the below method to receive the transaction details***

-(void) ResponseNew:(NSNotification \*)message

{

if ([message.name isEqualToString:@"JSON\_NEW"])

{

NSLog(@"Response = %@",[message object]);

jsondict = [message object];

}

}

-(IBAction)submitAction:(id)sender

{

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//If you want to pop to previous view use below code, if you have your view controller on your view stack

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

NSMutableArray \*allViewControllers = [NSMutableArray arrayWithArray:[self.navigationController viewControllers]];

for (UIViewController \*aViewController in allViewControllers)

{

NSString \*strClass = NSStringFromClass([aViewController class]);

if ([strClass isEqualToString:@"ViewController"])

{

[self.navigationController popToViewController:aViewController animated:NO];

}

}

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//If you want to push to some other view controller using storyboard use below code

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*//This is sample you have to create your own view controller

UIStoryboard \*storyBoard = [UIStoryboard storyboardWithName:[[NSBundle mainBundle].infoDictionary objectForKey:@"UIMainStoryboardFile"] bundle:[NSBundle mainBundle]];

TestKitViewController \*controller = (TestKitViewController\*)[storyBoard instantiateViewControllerWithIdentifier: @"sampleViewController"];

[self.navigationController pushViewController:controller animated:NO];

\*/

}

**Make sure the below thing if you are using XCODE 7.x.**

**In your Target -> Build settings -> Search for Enable BitCode -> YES.**

**To Get Failed Transaction details please follow below steps.**

**Failure Page:**

User will be redirected to failure page incase of the transaction has been timed out or

cancelled. Merchant can Configured(*Once cancel button is pressed it will navigate to the configured view controller*) **CANCEL\_VIEWCONTROLLER** add the below code.

Sample code:

**Step 1:**

In .h file

NSMutableDictionary \*jsondict;

**Step 2:**

In **viewWillAppear**

jsondict = [[NSMutableDictionary alloc]init];

[[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(ResponseNew:) name:@"FAILED\_DICT" object:nil];

[[NSNotificationCenter defaultCenter] postNotificationName:@"FAILED\_DICT\_NEW" object:nil userInfo:nil];

**Step 3:**

-(void) ResponseNew:(NSNotification \*)message

{

if ([message.name isEqualToString:@"FAILED\_DICT"])

{

//You will get the failed transaction details in below log and in jsondict.

NSLog(@"Response json data = %@",[message object]);

jsondict = [message object];

}

}

**Note:**

\* For **CANCEL\_VIEWCONTROLLER** and **THANKYOUVIEWCONTROLLER** configured viewcontrollers please use Storyboard. It works in Storyboard only.

\*Use the Configured view controller name as Storyboard ID.

For eg if **SuccessViewController.h** and **SuccessViewController.m** is the view controller name , Please use **SuccessViewController** as StoryboardID

**Limitations:**

1. EBS iOS sdk works on iPhone 5, iPhone 5c, iPhone 5s, iPhone 6 and iPhone 6 plus and iPad.
2. EBS iOS sdk works on iOS 7 , iOS 8 and iOS 9 SDK.

**Exceptions:**

Finding the error message after a crash, When your app runs and then crashes (whether it crashes as soon as your run it or it crashes after you click something), you need to know where to find the error message so that you can see what’s wrong.

After the crash, go to Xcode in the lower right hand pane, scroll all the way to the top.

