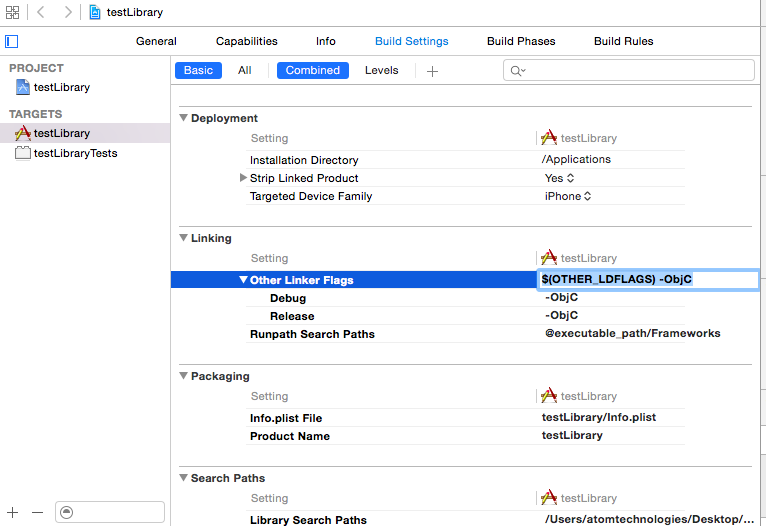
**ATOM Mobile SDK Integration**

The Atom mobile integration is designed to enable you to process payments through mobile applications.

**Integration Type:-** Non-Seamless:

**Setup**

* Create new Group in your project hierarch & add all the files from “payAtom” in it.
* Select your Project from Left Panel
  1. Go to targets tab & select the application
  2. Go to Build Setting & select Basic & Combined Tabs
  3. Add the following property as shown below



If you cannot find “Other Linker Flags” then you can do these steps below

1. Select the project file from the project navigator on the far left side of the window.

2. Select the target for where you want to add the linker flag.

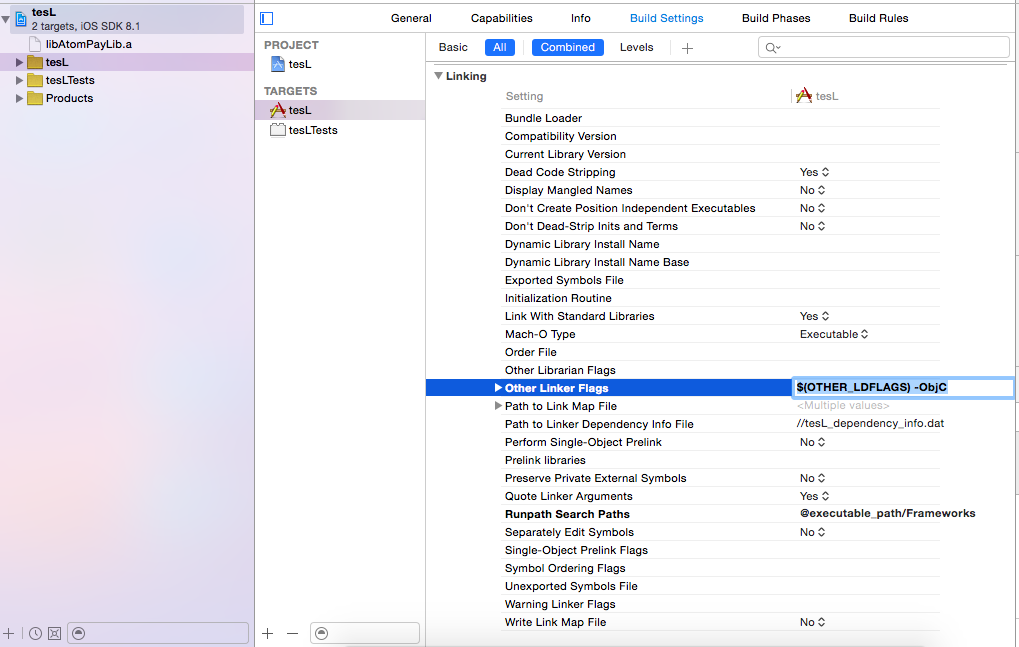
3. Select the "Build Settings" tab

4. Choose "All" to show all Build Settings.

5. Scroll down to the "Linking" section, and double-click to the right of where it says "Other Linking Flags".

6. A box will appear, Click on the "+" button to add a new linker flag.

7. Type "$(OTHER\_LDFLAGS) -ObjC" (no quotes) and press enter.



8. Add Bundle ”resourcesLib” in Build Phases selecting your project target(Copy Bundle Resources).

**Integration:**

* Merchant will design a screen wherein he will accept all the required fields including the bank detail, payment options and card details.
* Pass the data to Library as follows in the same format:

ViewController.h File

#import <UIKit/UIKit.h>

#import "nb.h"

#import "card.h"

#import "NSString+DES.h"

#import "NSData+DES.h"

@interface ViewController : UIViewController <NSXMLParserDelegate,nbDelegate,cardDelegate>

{

NSXMLParser \*parser;

}

@property (weak, nonatomic) IBOutlet UIButton \*InitiateRequest;

@property (nonatomic,retain) NSXMLParser \*parser;

-(IBAction)callVC:(id)sender;//Call for NetBanking Transaction

-(IBAction)callVCard:(id)sender;//Call For Card Transaction

@end

ViewController.m File

#import "ViewController.h"

#import "nb.h"

#import "card.h"

@interface ViewController ()

@end

@implementation ViewController

@synthesize parser;

- (void)viewDidLoad {

[super viewDidLoad];

}

- (void)didReceiveMemoryWarning {

[super didReceiveMemoryWarning];

// Dispose of any resources that can be recreated.

}

-(IBAction)callVC:(id)sender

{

nb \*netB = [[nb alloc] init];

netB.myDelegate = self;

netB.loginid=@"459";

netB.txnscamt=@"0";

netB.loginid=@"459";

netB.password=@"Test@123";

netB.prodid=@"NSE";

netB.txncurr=@"INR";

netB.clientcode=@"001";

netB.custacc=@"100000036600";

netB.amt=@"100.000";

netB.txnid=@"9Q6";//unique each time

netB.date=@"23/08/2010%2011:57:00";//unique each time

netB.bankid=@"2001";

netB.ru=@"https://paynetzuat.atomtech.in/paynetz/param";

[self presentViewController:netB animated:YES completion:nil];

}

-(IBAction)callVCard:(id)sender

{

card \*cc = [[card alloc] initWithNibName:@"CView" bundle:nil];

cc.myDelegate = self;

cc.loginid=@"459";

cc.txnscamt=@"0";

cc.loginid=@"459";

cc.password=@"Test@123";

cc.prodid=@"NSE";

cc.txncurr=@"INR";

cc.clientcode=@"001";

cc.custacc=@"100000036600";

cc.amt=@"200.000";

cc.txnid=@"124";

cc.date=@"24/09/2015%2012:57:00";

cc.channelid=@"INT";

cc.ru=@"https://paynetzuat.atomtech.in/paynetz/param";

cc.cardtype=@"CC";

cc.cardAssociate=@"VISA";

[self presentViewController:cc animated:YES completion:nil];

}

-(void)secondviewcontrollerDissmissed:(NSString \*)stringToFirst

{

NSString \*getResult;

getResult = stringToFirst;

NSLog(@"received---->%@",getResult); //This will return status success or Fail of Transaction

}

@end

**Net Banking**

* After filling the required details he will be redirected to the bank page wherein he will authorize himself.
* After authorization he will be redirected to a response i.e. Success or failure will be sent to the merchant page which was configured as return url.

**Card Payment**

* On passing all the parameters, enter card details on next screen
* Payment will be processed with encrypted card details & transaction status will be displayed on return url.
* This page will be closed in 5 seconds & control will be back to the calling application

.