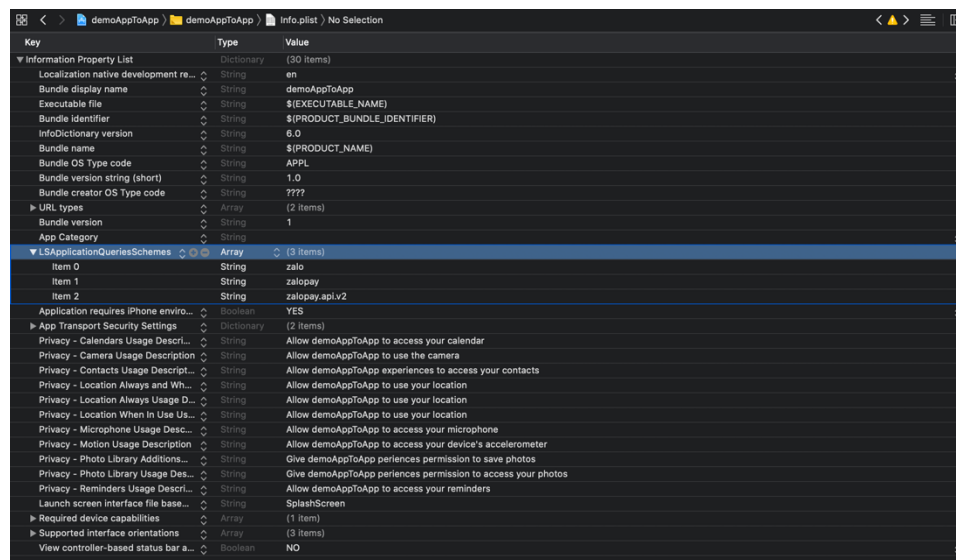




Hướng dẫn chạy thử và tích hợp ZaloPay SDK v5.1.0 trên nền tảng React Native

- Run demo cho iOS
 - Truy vấn đến thư mục demoAppToApp trên Command Line
 - Gọi lệnh : `yarn install` hoặc `npm install`
 - Cài đặt Xcode version 11 và CocoaPods (Bỏ qua nếu đã cài đặt)
 - Truy vấn vào thư mục ios
 - Gọi lệnh: `pod install`
 - Gọi lệnh: `yarn ios` hoặc `npm run ios` ở thư mục demoAppToApp
 - Vì ứng dụng ZaloPay Sandbox không thể cài đặt trên Simulator. Nên để có thể run demo tốt nhất, người dùng vui lòng cài đặt ZaloPay Sandbox và kết nối device thật với Xcode.
- Run demo cho Android
 - Truy vấn đến thư mục demoAppToApp trên Command Line
 - Gọi lệnh : `yarn install` hoặc `npm install` (Bỏ qua nếu đã làm ở các bước trên)
 - Khởi chạy Emulator với Android 8.1 và API Level 27
 - Cài đặt ứng dụng ZaloPay Sandbox trên Emulator
 - Gọi lệnh: `yarn android` hoặc `npm run android` ở thư mục demoAppToApp
- Tích hợp vào iOS

- Thêm framework `zpdk.framework` vào Project
- Thêm 2 file **PayZaloBridge.h** và **PayZaloBridge.m** vào project
- Thêm cấu hình cho phép khởi tạo ZaloPay từ app, bằng cách thêm giá trị **zalo**, **zalopay.api.v2** và **zalopay** vào key **LSApplicationQueriesSchemes**

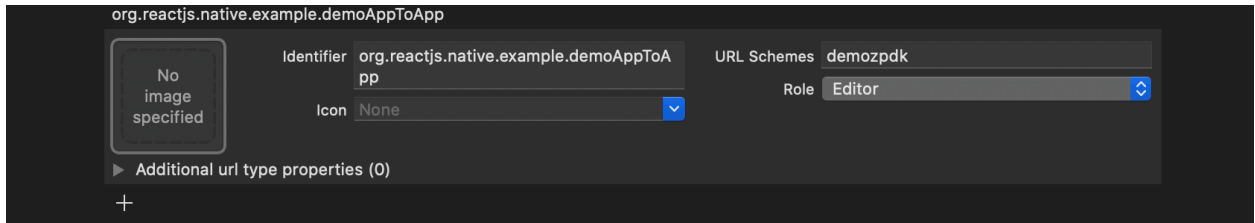


- Trong hàm `didFinishLaunchingWithOptions` của `AppDelegate.m`, thêm dòng code sau:

```
[[ZaloPaySDK sharedInstance] initWithAppId:<yourAppID> uriScheme:@"<yourAppScheme>" environment:<Environment>];
```

Với:

- <yourAppId> là appId do ZaloPay cung cấp. Ví dụ: 553.
- <yourAppScheme> là URI Scheme app của Merchant. Được khởi tạo ở “Project” -> “Info”



```

28 - (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
29 {
30     self.moduleRegistryAdapter = [[UMModuleRegistryAdapter alloc] initWithModuleRegistryProvider:[[UMModuleRegistryProvider alloc] init]];
31     self.launchOptions = launchOptions;
32
33     self.window = [[UIWindow alloc] initWithFrame:[UIScreen mainScreen].bounds];
34     #ifdef DEBUG
35     [self initializeReactNativeApp];
36     #else
37     EXUpdatesAppController *controller = [EXUpdatesAppController sharedInstance];
38     controller.delegate = self;
39     [controller startAndShowLaunchScreen:self.window];
40     #endif
41
42     [super application:application didFinishLaunchingWithOptions:launchOptions];
43     [[ZaloPaySDK sharedInstance] initWithAppId:553 uriScheme:@"demozpdk://app" environment: ZPZPIEnvironment_Sandbox]; // Init ZPDK
44     return YES;
45 }

```

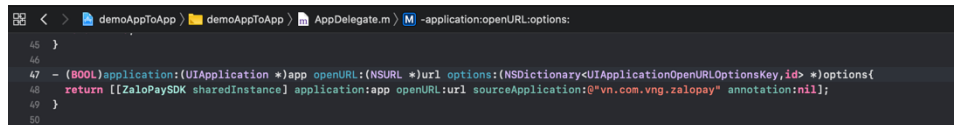
- Thêm vào AppDelegate.m đoạn code bên dưới

```

- (BOOL)application:(UIApplication *)app open
URL:(NSURL *)url options:(NSDictionary<UIApplicationOpenURLOptionsKey,id> *)options{

    return [[ZaloPaySDK sharedInstance] applica
tion:app openURL:url sourceApplication:@"vn.c
om.vng.zalopay" annotation:nil];
}

```



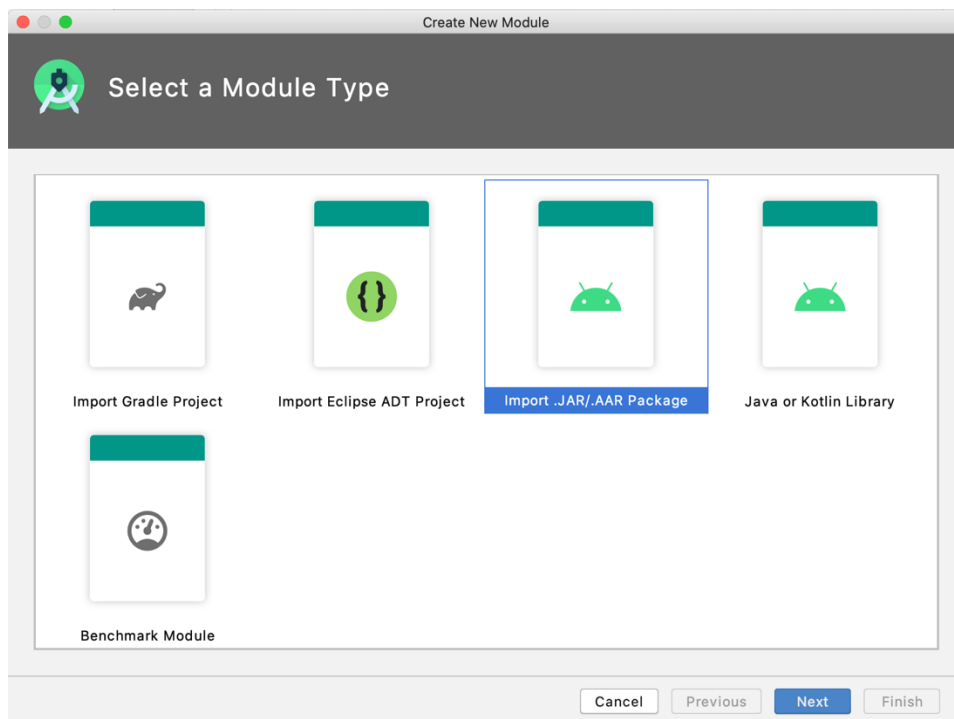
- Merchant cần handle các trường hợp sau khi thanh toán phù hợp với Business của mình. Ví dụ:

```

63 - (void)paymentDidSucceeded:(NSString *)transactionId zpTranstoken:(NSString *)zpTranstoken appTransId:(NSString *)appTransId {
64     //Handle Success
65     [self sendEventWithName:@"EventPayZalo" body:@{@"resultCode": [NSString stringWithFormat:@"%ld", PAYMENTSUCCESS], @"transactionId":transactionId ? transactionId : @"" ,
66     @"zpTranstoken":zpTranstoken ? zpTranstoken : @"" , @"appTransId": appTransId ? appTransId : @""}];
67 }
68 - (void)paymentDidCanceled:(NSString *)zpTranstoken appTransId:(NSString *)appTransId {
69     // Handle Canceled
70     [self sendEventWithName:@"EventPayZalo" body:@{@"resultCode": [NSString stringWithFormat:@"%ld", PAYMENTCANCELED], @"zpTranstoken":zpTranstoken ? zpTranstoken : @"" ,
71     @"appTransId":appTransId ? appTransId : @""}];
72 }
73 - (void)paymentDidError:(ZPaymentErrorCode)errorCode zpTranstoken:(NSString *)zpTranstoken appTransId:(NSString *)appTransId {
74     // Handle Error
75     [self sendEventWithName:@"EventPayZalo" body:@{@"resultCode": [NSString stringWithFormat:@"%ld", PAYMENTFAILED], @"zpTranstoken":zpTranstoken ? zpTranstoken : @"" ,
76     @"appTransId":appTransId ? appTransId : @""}];
77 }
78 @end
79

```

- Tích hợp vào Android
 - Import Lib ZPDK của ZaloPay (file *.aar) vào project.
 - File -> New -> New Module -> Insert Module JAR/AAR file



- Chọn file zpdk-release.aar
- Kiểm tra file build.gradle của module có dạng như sau:

```

183 dependencies {
184     implementation fileTree(dir: "libs", include: ["*.jar"])
185     implementation "com.facebook.react:react-native:+"
186     implementation project(path: ':zpdk-release')
187     // From node_modules
188     addUnimodulesDependencies()
189
190     if (enableHermes) {
191         def hermesPath = "../../node_modules/hermes-engine/android/";
192         debugImplementation files(hermesPath + "hermes-debug.aar")
193         releaseImplementation files(hermesPath + "hermes-release.aar")
194     } else {
195         implementation jscFlavor
196     }
197 }
198

```

- Copy thư mục “**zpmodule**” vào project. Thay đổi package name tương ứng với project hiện tại.
- Import **PayZaloBride** từ “**zpmodule**” và **ZaloPaySDK** vào MainApplication

```

30 import com.demopapptapp.zpmodule.PayZaloBridge;
31 import vn.zalopay.sdk.ZaloPaySDK;
32

```

- Thêm lời gọi hàm khởi tạo ZaloPaySDK trong onCreate của Application với appld là applD của ZaloPay cung cấp cho Merchant

`ZaloPaySDK.init(<yourAppID>, <Environment>);`

Với:

- <yourAppID> là appId do ZaloPay cung cấp
- <Environment> là môi trường cần tích hợp

```

84      @Override
85      public void onCreate() {
86          super.onCreate();
87          SoLoader.init(context: this, /* native_exopackage */ nativeExopackage: false);
88          initializeFlipper(context: this); // Remove this line if you don't want Flipper enabled
89          ZaloPaySDK.init(appId: 553, Environment.SANDBOX);
90          if (!BuildConfig.DEBUG) {
91              UpdatesController.initialize(context: this);
92          }
93      }

```

- Trong hàm getPackages(), thêm dòng code

```
packages.add(new PayZaloBridge());
```

```

@Override
protected List<ReactPackage> getPackages() {
    List<ReactPackage> packages = new PackageList(this).getPackages();
    packages.add(new ModuleRegistryAdapter(mModuleRegistryProvider));
    packages.add(new PayZaloBridge());
    return packages;
}

```

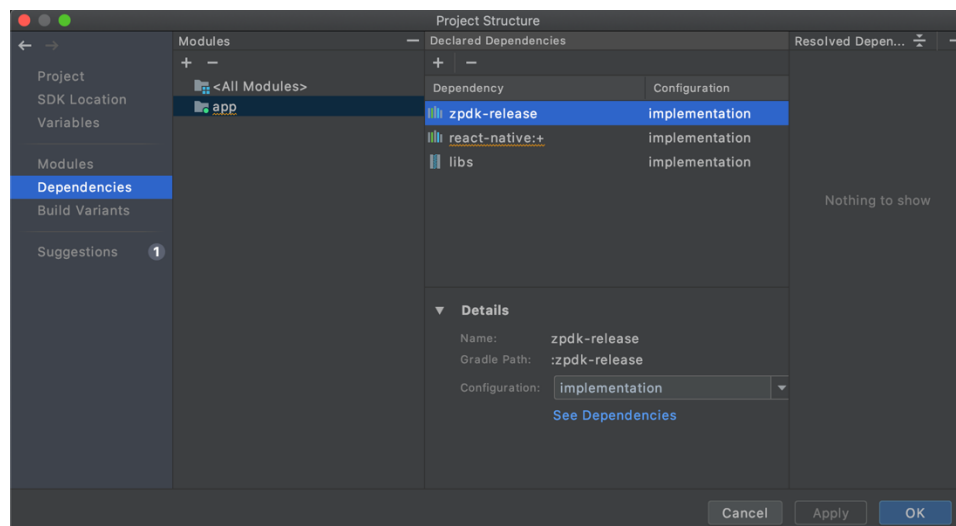
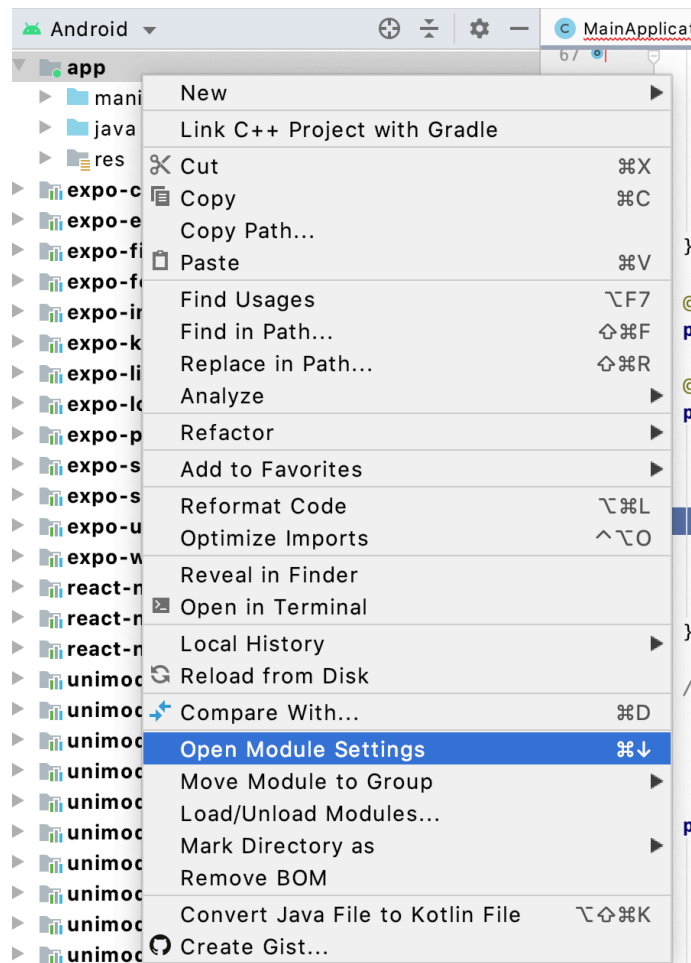
- Merchant cần handle các trường hợp thanh toán phù hợp với Business của mình. Ví dụ:

```

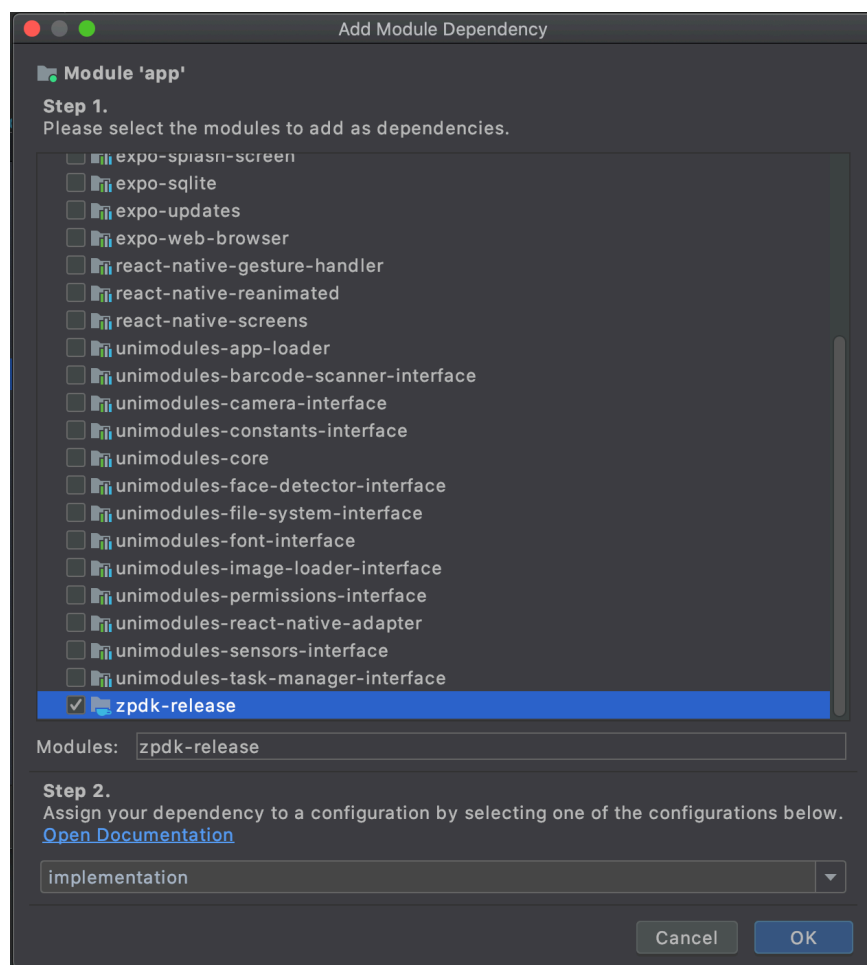
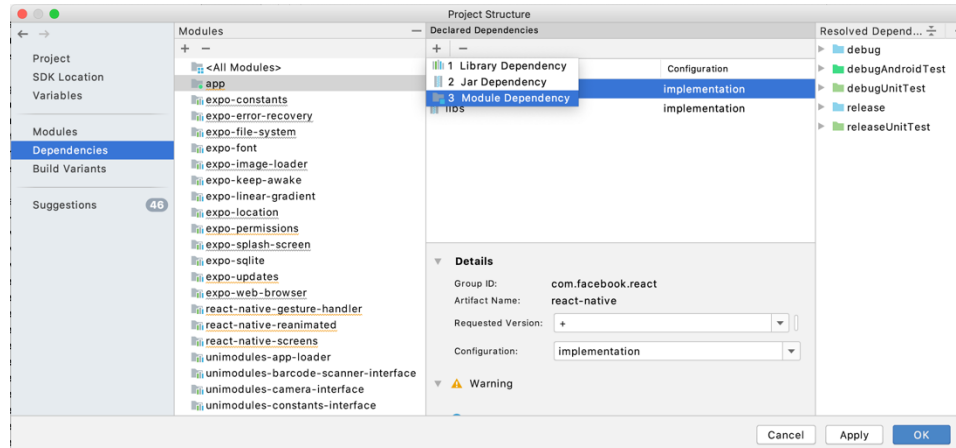
25 PayOrderListener listener = new PayOrderListener() {
26     @Override
27     public void onPaymentSucceeded(String transactionId, String transToken, String appTransID) {
28         // Handle Success
29         WritableMap params = Arguments.createMap();
30         params.putString("transactionId", transactionId);
31         params.putString("transToken", transToken);
32         params.putString("appTransID", appTransID);
33         params.putString("returnCode", PAYMENTSUCCESS);
34         sendEvent(mReactContext, eventName: "EventPayZalo", params);
35     }
36
37     @Override
38     public void onPaymentCanceled(String transToken, String appTransID) {
39         // Handle Cancel
40         WritableMap params = Arguments.createMap();
41         params.putString("returnCode", PAYMENTCANCELED);
42         params.putString("zpTransToken", transToken);
43         params.putString("appTransID", appTransID);
44         sendEvent(mReactContext, eventName: "EventPayZalo", params);
45     }
46
47     @Override
48     public void onPaymentError(ZaloPayError zaloPayError, String transToken, String appTransID) {
49         // Handle Error
50         WritableMap params = Arguments.createMap();
51         params.putString("returnCode", PAYMENTFAILED);
52         params.putString("zpTransToken", transToken);
53         params.putString("appTransID", appTransID);
54         sendEvent(mReactContext, eventName: "EventPayZalo", params);
55     }
56 };

```

- Kiểm tra zpdk-release dependency của app bằng cách chọn Module Setting -> Dependencies



Lưu ý: Nếu không tìm thấy zpdk dependency, người dùng có thể thêm bằng cách chọn dấu “+”
-> Module Dependency -> zpdk-release



- Handle data trả về ở React Native

```
1 import * as React from 'react';
2 import { StyleSheet, Text, ScrollView, KeyboardAvoidingView, NativeModules, NativeEventEmitter } from 'react-native'; //Import NativeModules and NativeEventEmitter
3 import { TextInput } from 'react-native-gesture-handler';
4 import { Button } from 'react-native-elements';
5 import CryptoJS from 'crypto-js';
6
7 const { PayZaloBridge } = NativeModules; //Import PayZaloBridge from NativeModules
8
9 const payZaloBridgeEmitter = new NativeEventEmitter(PayZaloBridge); //Init ZaloPay Bridge Emitter
10
11 const subscription = payZaloBridgeEmitter.addListener( // Add Listener to ZaloPay Bridge Emitter
12   'EventPayZalo',
13   (data) => {
14     console.log('Kết quả giao dịch: ' + data.returnValue);
15     if(data.returnValue == -1){ //ZaloPay not install
16       var payZP = NativeModules.PayZaloBridge;
17       payZP.installApp();
18     }else if(data.returnValue == 1){
19       alert('Pay success!');
20     }else{
21       alert('Pay error! ' + data.returnValue);
22     }
23   }
24 );
```