

CÔNG TY CỔ PHẦN GIẢI PHÁP THANH TOÁN VIỆT NAM



**HỆ THỐNG CÔNG THANH TOÁN
TÀI LIỆU KẾT NỐI MOBILE SDK**

Phiên bản: 1.1

Hà Nội, tháng 8 năm 2020

I. GIỚI THIỆU CHUNG

1. Mục đích tài liệu

- Tài liệu nhằm mô tả luồng xử lý, code mẫu cũng như cách thức cài đặt SDK call application giữa ứng dụng di động của đối tác và ứng dụng Mobile banking ngân hàng được chấp nhận thanh toán qua hình thức này.

2. Phạm vi tài liệu

- Tài liệu được cung cấp trong phạm vi hợp tác dịch vụ giữa VNPAY và Merchant có nhu cầu tích hợp thanh toán qua cổng thanh toán của VNPAY.

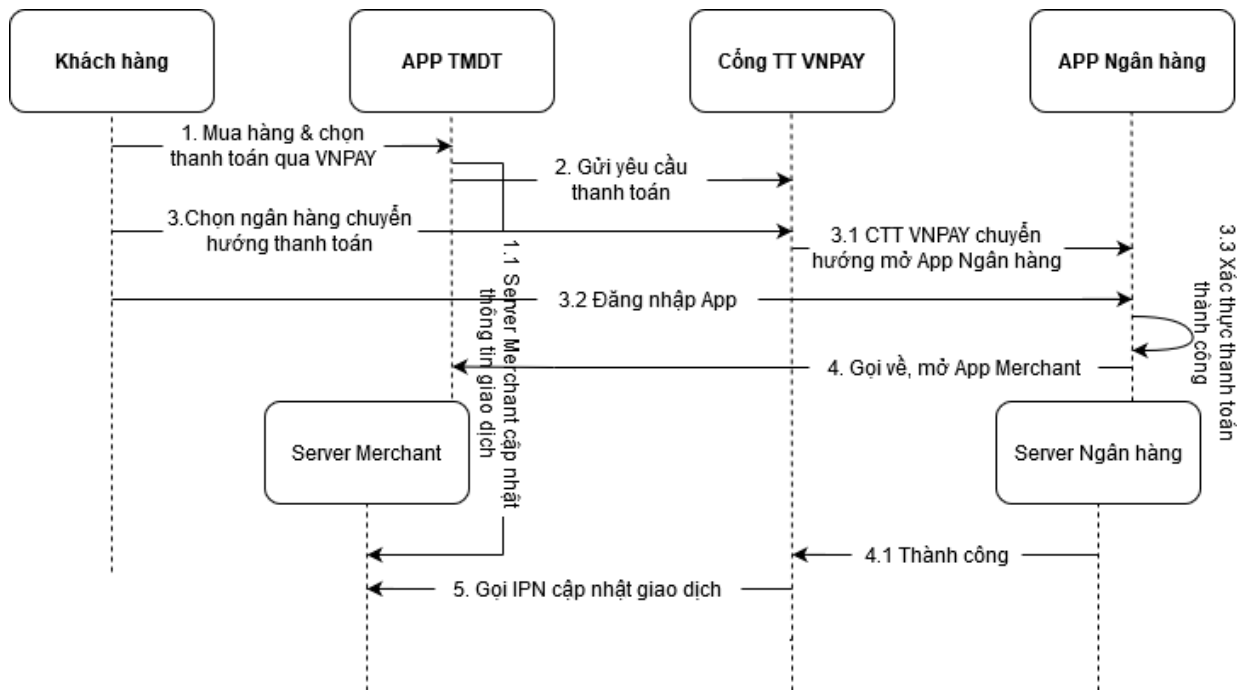
3. Đối tượng sử dụng tài liệu

- Đối tượng sử dụng tài liệu dành cho kỹ thuật tích hợp của Merchant.

II. ĐẶC TẢ KẾT NỐI

1. Sơ đồ tuần tự

1.1 Deep-link apps



Mô tả sơ đồ:

Bước 1: Khách hàng vào APP TMDT thực hiện mua hàng và chọn Thanh toán qua VNPAY.

Hệ thống Merchant lưu thông tin thanh toán của giao dịch khởi tạo.

Bước 2: APP TMDT gửi thông tin thanh toán và sử dụng SDK để chuyển hướng mở URL

thanh toán của VNPAY.

Bước 3: Khách hàng chọn ngân hàng chuyển hướng thanh toán, mở ứng dụng thanh toán của ngân hàng đăng nhập và xác thực thanh toán.

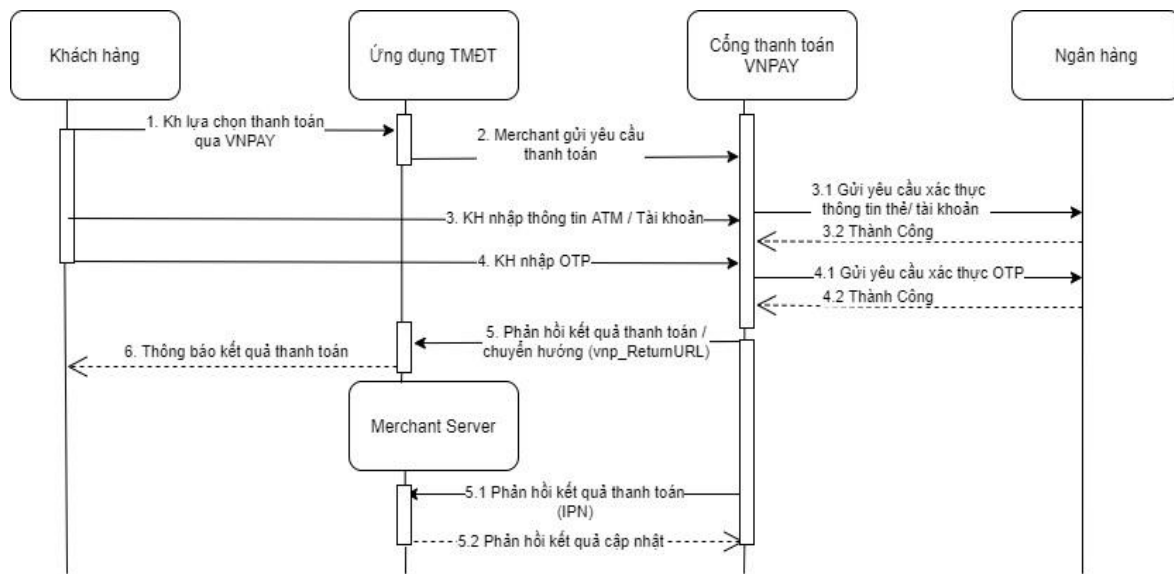
Bước 4: Xác thực thanh toán thành công, ứng dụng của ngân hàng sẽ gọi và mở lại APP TMDT.

Đồng thời hệ thống ngân hàng phản hồi lại kết quả thanh toán cho Cổng thanh toán VNPAY.

Bước 5: Hệ thống Cổng thanh toán VNPAY gọi IPN Merchant trả kết quả thanh toán (Danh sách tham số trả về tại mục 2.5.3.2 trong File tài liệu đặc tả kết nối VNPAY Payment Gateway_Techspec 2.0.1-VN).

Hệ thống Merchant cập nhật trạng thái thanh toán cho giao dịch.

1.2 ATM - Tài khoản - Thẻ quốc tế



Mô tả sơ đồ:

Bước 1: Khách hàng truy cập ứng dụng TMDT đặt hàng hóa - dịch vụ và chọn thanh toán trực tuyến qua VNPAY.

Bước 2: Ứng dụng TMDT gửi yêu cầu thanh toán đến VNPAY. Tham khảo tại mục 2.5.3.1.

Bước 3: Khách hàng nhập thông tin thẻ ATM - Tài khoản tại VNPAY hoặc trang thanh toán của ngân hàng. VNPAY gửi yêu cầu xác thực thông tin ATM - tài khoản đến ngân hàng.

Nếu xác thực tài khoản thành công, ngân hàng sẽ gửi mã OTP cho khách hàng. Khách hàng chuyển sang Bước 4.

Bước 4: Khách hàng nhập mã OTP để hoàn tất giao dịch. VNPAY gửi yêu cầu xác thực OTP đến ngân hàng. Nếu xác thực OTP thành công, hệ thống của ngân hàng trừ tiền tài khoản của khách hàng và phản hồi lại kết quả thành công cho VNPAY. Chuyển sang Bước 5.

Bước 5: VNPAY gửi phản hồi kết quả thanh toán đến ứng dụng TMĐT thông qua 2 URL của merchant:

- **Return URL:** VNPAY phản hồi kết quả thanh toán và chuyển hướng về giao diện thông báo kết quả thanh toán của ứng dụng TMĐT. Tham khảo danh sách tham số VNPAY phản hồi tại mục [2.5.3.2](#).
- **IPN URL (Server - call - server):** Nhận được kết quả thanh toán VNPAY phản hồi, IPN của ứng dụng TMĐT thực hiện cập nhật kết quả và phản hồi tình trạng cập nhật cho VNPAY. Tham khảo danh sách tham số VNPAY phản hồi tại mục [2.5.3.2](#).

2. Setup SDK

2.1 Mô tả:

Thiết lập các thông số đầu vào cho sdk

2.2 Request:

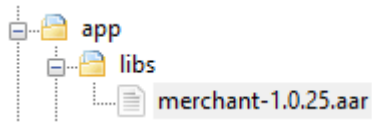
Dữ liệu bao gồm các trường:

STT	Tham số	Mô tả
1	url	URL thanh toán được tạo theo đặc tả kết nối Cổng thanh toán (mục 2.5.3.1 trong file VNPAY Payment Gateway_Techs spec 2.0.1-VN)
2	Scheme	Schemes của APP TMDT: Khi thanh toán thành công ứng dụng Mobile Banking/Ví điện tử gọi mở lại Scheme ứng dụng của bạn.
3	Tmn_Code	vnp_TmnCode: được VNPAY cung cấp khi thực hiện kết nối với hệ thống cổng của VNPAY
4	isSandBox	Cấu hình True or False để chuyển hướng môi trường Sandbox hoặc Production của VNPAY.

III. CÀI ĐẶT

1. Android.

1.1 Cài đặt



- Thêm [merchant-1.0.25.aar](#) vào trong thư mục libs trên Module chính của projects
Cấu hình file build.gradle trong Module chính của project

Thêm gradle config

```
allprojects {
    repositories {
        flatDir {
            dirs 'libs'
        }
    }
}
```

Thêm trong repositories
flatDir { dirs 'libs' }

Thêm trong dependencies

```
implementation 'com.google.code.gson:gson:x.x.x'
implementation 'com.squareup.okhttp3:okhttp:x.x.x'
implementation (name: 'merchant-1.0.25', ext: 'aar')
```

```
dependencies {
    implementation 'com.google.code.gson:gson:2.8.5'
    implementation group: 'com.squareup.okhttp3', name: 'okhttp', version: '3.14.1'

    compile(name: 'merchant-1.0.25', ext: 'aar')
}
```

Thêm config trong manifest

```
<manifest
    .....
    >
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        .....
        >
        <!--Activity để trigger kết quả thanh toán từ app mobile banking
        khi mở lại activity này thì bạn cần kiểm tra trạng thái thanh toán của đơn hàng -->
        <activity
            android:name=".ResultActivity">
            <intent-filter>
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.BROWSABLE" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```

        <data android:scheme="your_scheme_app" />
    </intent-filter>
</activity>
</application>
</manifest>

```

1.2 Code mẫu

JAVA:

Code demo

```

public void openSdk() {
    Intent intent = new Intent(this, VNP_AuthenticationActivity.class);
    intent.putExtra("url", "URL payment redirect"); //bắt buộc, URL thanh toán do
    merchant tạo.
    intent.putExtra("tmn_code", "xxxxxxx"); //bắt buộc, VNPAY cung cấp
    intent.putExtra("scheme", "xxxxxxx"); //bắt buộc, scheme để mở lại app khi có kết quả
    thanh toán từ mobile banking
    intent.putExtra("is_sandbox", false); //bắt buộc, true <=> môi trường test, false <=> môi
    trường live
    VNP_AuthenticationActivity.setSdkCompletedCallback(new
    VNP_SdkCompletedCallback() {
        @Override
        public void sdkAction(String action) {
            Log.wtf("SplashActivity", "action: " + action);

            //Các action trả về từ sdk để client app đối tác nhận diện và điều hướng tiếp

            //action == AppBackAction
            //Người dùng nhấn back từ sdk(ios) hoặc back từ device(android) để quay lại.

            //action == CallMobileBankingApp
            //Người dùng nhấn chọn mở thanh toán qua app thanh toán (Mobile Banking, Ví...).

            //action == WebBackAction
            //user Hủy thanh toán từ Cổng VNPAY-QR
            //Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
            http://cancel.sdk.merchantbackapp

            //action == FaildBackAction
            //kết quả thanh toán không thành công từ phương thức ATM,Tài khoản, thẻ quốc tế
            // Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
            http://fail.sdk.merchantbackapp

            //action == SuccessBackAction
            //kết quả thanh toán thành công từ phương thức ATM,Tài khoản, thẻ quốc tế hoặc scanQR.
            // Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
            http://success.sdk.merchantbackapp

        }
    });
    startActivity(intent);
}

```

```
}
```

2. IOS.

2.1 Cài đặt

- Kéo vào project SKD CallAppSDK.framework
- Để ở chế độ Do Not Embbed trong Tab General của project
- Đặt URL Schemes ở Mục URL Types trong tab Info của Project

2.2 Code mẫu

Mở SDK

```
- (void)dealloc {
    [[NSNotificationCenter defaultCenter] removeObserver:self name:@"SDK_COMPLETED"
    object:nil];
}

- (IBAction)openSDK:(id)sender {

    //các thông số dưới đây là demo
    //vui lòng đọc kỹ comment của từng variable

    UIViewController *fromVC = self; //bắt buộc
    NSString *scheme = @"merchantpaymentresult"; //bắt buộc, tên scheme merchant tự cài đặt
    theo app
    BOOL isSandbox = NO; //bắt buộc, YES <=> môi trường test, NO <=> môi trường live
    NSString *paymentUrl = @"URL payment redirect "; //bắt buộc, URL hệ thống Merchant
    tạo.
    NSString *tmn_code = @"xxxxxxx "; //bắt buộc, VNPAY cung cấp
    BOOL backAction = YES; //bắt buộc, YES <=> bấm back sẽ thoát SDK, NO <=> bấm back
    thì trang web sẽ back lại trang trước đó, nên set là YES, nên set là YES, vì trang thanh toán
    không nên cho người dùng back về trang trước
    NSString *backAlert = @""; //không bắt buộc, thông báo khi người dùng bấm back
    NSString *title = @"VNPAY"; //bắt buộc, title của trang thanh toán
    NSString *titleColor = @"#000000"; //bắt buộc, màu của title
    NSString *beginColor = @"#FFFFFF"; //bắt buộc, màu của background title
    NSString *endColor = @"#FFFFFF"; //bắt buộc, màu của background title
    NSString *iconBackName = @"ic_back"; //bắt buộc, icon back

    //lưu ý, header của sdk sử dụng navigationbar mặc định của ios

    [self showFromVC:fromVC
        scheme:scheme
        isSandbox:isSandbox
        paymentUrl:paymentUrl
        tmn_code:tmn_code
        backAction:backAction
        backAlert:backAlert
```

```

        title:title
        titleColor:titleColor
        beginColor:beginColor
        endColor:endColor
        iconBackName:iconBackName];
    }

- (void)showFromVC:(UIViewController*)fromVC
    scheme:(NSString *)scheme
    isSandbox:(BOOL)isSandbox
    paymentUrl:(NSString *)paymentUrl
    tmn_code:(NSString *)tmn_code
    backAction:(BOOL)backAction
    backAlert:(NSString *)backAlert
    title:(NSString *)title
    titleColor:(NSString *)titleColor
    beginColor:(NSString *)beginColor
    endColor:(NSString *)endColor
    iconBackName:(NSString *)iconBackName {
    [[NSNotificationCenter defaultCenter] removeObserver:self name:@"SDK_COMPLETED"
object:nil];
    [[NSNotificationCenter defaultCenter] addObserver:self selector:@selector(sdkAction:)
name:@"SDK_COMPLETED" object:nil];

    [CallAppInterface setHomeViewController:fromVC];
    [CallAppInterface setSchemes:scheme];
    [CallAppInterface setIsSandbox:isSandbox];
    [CallAppInterface setAppBackAlert:backAlert];
    [CallAppInterface setEnableBackAction:backAction];
    [CallAppInterface showPushPaymentwithPaymentURL:paymentUrl
        withTitle:title
        iconBackName:iconBackName
        beginColor:beginColor
        endColor:endColor
        titleColor:titleColor
        tmn_code:tmn_code];
}

-(void)sdkAction:(NSNotification*)notification {
    if ([notification.name isEqualToString:@"SDK_COMPLETED"]){

        NSString *actionValue=[notification.object valueForKey:@"Action"];
        if ([@"AppBackAction" isEqualToString:actionValue]) {
            //Người dùng nhấn back từ sdk để quay lại

            return;
        }
        if ([@"CallMobileBankingApp" isEqualToString:actionValue]) {
            //Người dùng nhấn chọn mở thanh toán qua app thanh toán (Mobile Banking, Ví...).

```



```

    return;
}
if ([@"WebBackAction" isEqualToString:actionValue]) {
//user Hủy thanh toán từ Cổng VNPAY-QR
//Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://cancel.sdk.merchantbackapp

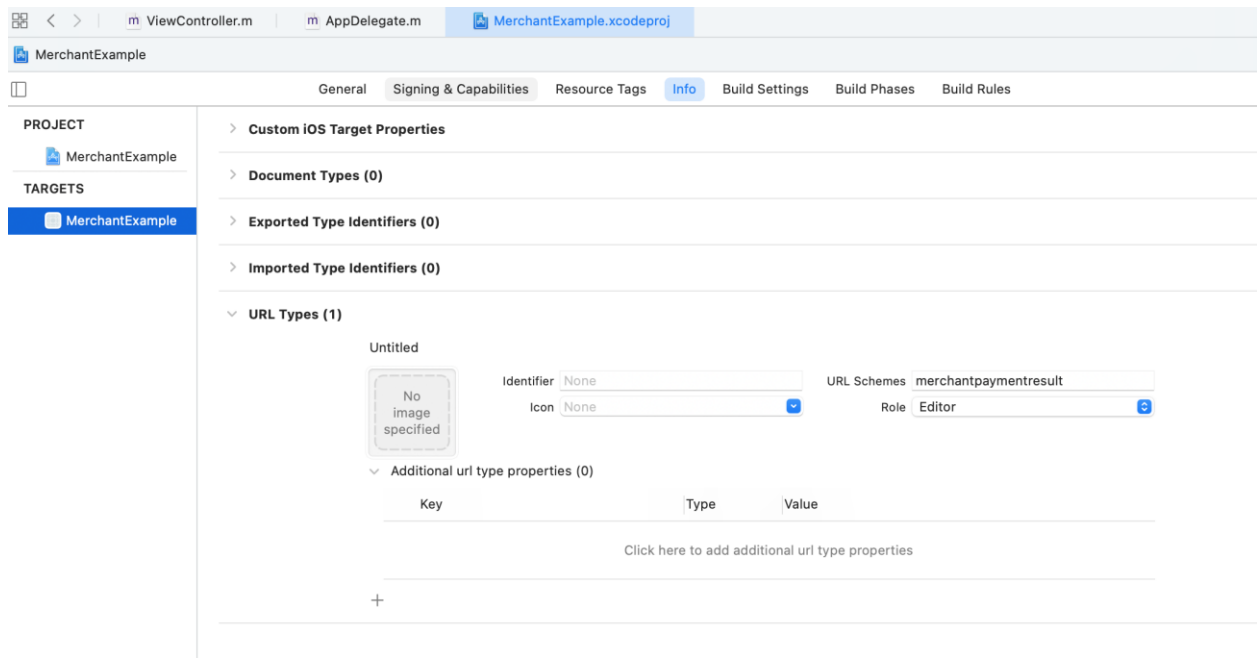
    return;
}
if ([@"FaildBackAction" isEqualToString:actionValue]) {
// kết quả thanh toán không thành công từ phương thức ATM,Tài khoản, thẻ quốc tế
// Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://fail.sdk.merchantbackapp

    return;
}
if ([@"SuccessBackAction" isEqualToString:actionValue]) {
// kết quả thanh toán thành công từ phương thức ATM,Tài khoản, thẻ quốc tế hoặc scanQR.
// Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://success.sdk.merchantbackapp

    return;
}
}
}
}

```

Cài đặt sheme để mở lại app khi thanh toán từ mobile banking



Trigger sự kiện mở lại app khi thanh toán từ mobile banking

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

    if ([url.scheme isEqualToString:@"your_URL_scheme"]){

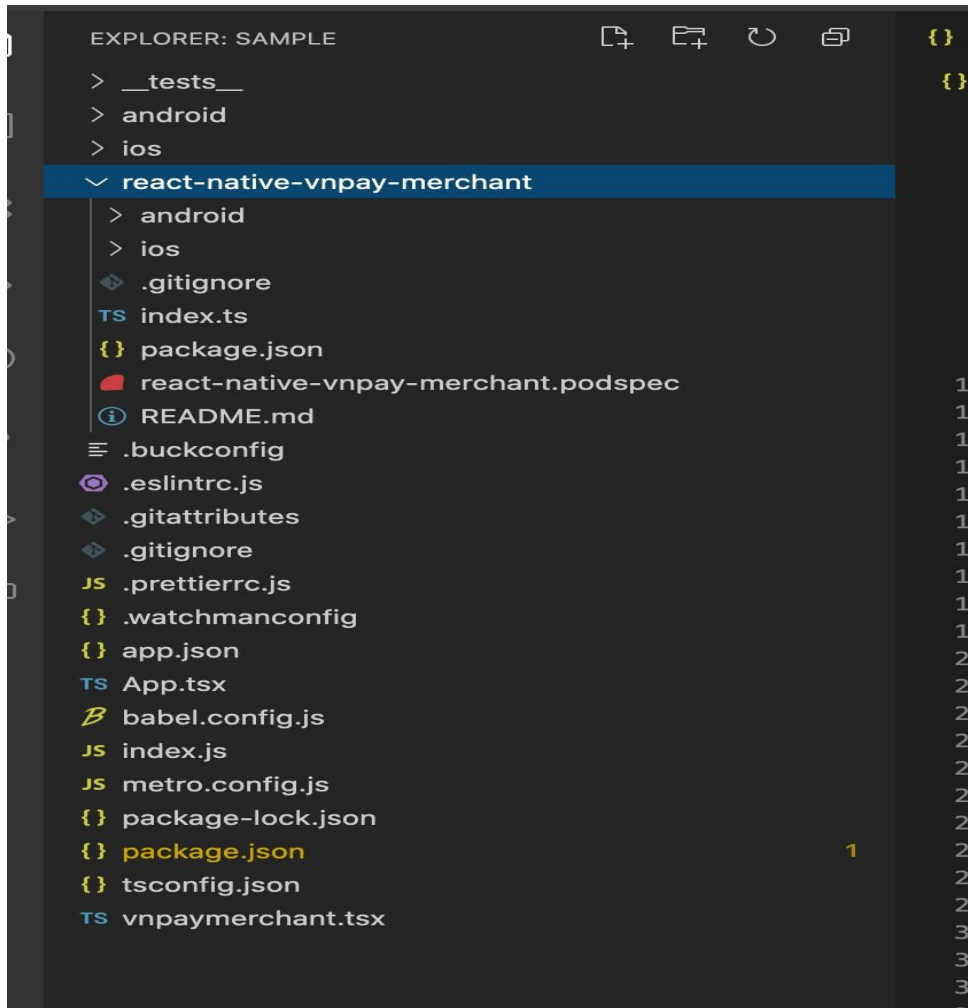
        //payment result, lúc này bạn cần check trạng thái giao dịch

        return YES;
    }
    return YES;
}
```

3. React Native.

Cài đặt:

Step 1: copy folder **react-native-vnpay-merchant** đến project của đơn vị.



Step 2: Thêm module **react-native-vnpay-merchant** từ **package.json**, import as folder

```
"react-native-vnpay-merchant": "file:<path to module>"
```

```
package.json — sample
{} package.json x
{} package.json > {} dependencies > react-native-vnpay-merchant
1 {
2   "name": "SampleVnpayMerchant",
3   "version": "0.0.1",
4   "private": true,
5   "scripts": {
6     "android": "react-native run-android",
7     "ios": "react-native run-ios",
8     "start": "react-native start",
9     "test": "jest",
10    "lint": "eslint . --ext .js,.jsx,.ts,.tsx"
11  },
12  "dependencies": {
13    "react": "16.13.1",
14    "react-native": "0.63.2",
15    "react-native-material-ui": "^1.30.1",
16    "react-native-vector-icons": "^7.0.0",
17    "react-native-vnpay-merchant": "file:react-native-vnpay-merchant"
18  },
19  "devDependencies": {
20    "@babel/core": "^7.8.4",
21    "@babel/runtime": "^7.8.4",
22    "@react-native-community/eslint-config": "^1.1.0",
23    "@types/jest": "^25.2.3",
24    "@types/react-native": "^0.63.2",
25    "@types/react-native-material-ui": "^1.32.2",
```

Step 3: npm install

Step 4: cd ios folder and run script: pod install

Step 5: Setup sdk

Android

Update file `android/build.gradle`

```
allprojects {
  repositories {
    maven {
      url("<path to react-native-vnpay-merchant/android/repo>")
      //ex: $rootDir/../node_modules/react-native-vnpay-merchant/android/repo
    }
  }
}
```

iOS

Update file `AppDelegate.m`

```
#import <CallAppSDK/CallAppInterface.h>

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    //... your code
    [CallAppInterface setHomeViewController:self.window.rootViewController];
    return YES;
}
```

Code demo:

```
import VnpayMerchant, { VnpayMerchantModule } from './react-native-vnpay-merchant'
const eventEmitter = new NativeEventEmitter(VnpayMerchantModule);
```

Step1: Add listener to get sdk result

```
eventEmitter.addListener('PaymentBack', (e) => {
    console.log('Sdk back!')
    if (e) {
        console.log("e.resultCode = " + e.resultCode);
        switch (e.resultCode) {
```

***handle sự kiện trả về từ sdk**

```
eventEmitter.addListener('PaymentBack', (e) => {
    console.log('Sdk back!')
    if (e) {
        console.log("e.resultCode = " + e.resultCode);
        switch (e.resultCode) {
            //resultCode == -1
            //Người dùng nhấn back từ sdk để quay lại
```

```
            //resultCode == 10
            //Người dùng nhấn chọn mở thanh toán qua app thanh toán (Mobile Banking, Ví...).
```

```
//resultCode == 99
//user Hủy thanh toán từ Cổng VNPAY-QR
//Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://cancel.sdk.merchantbackapp

//resultCode == 98
// kết quả thanh toán không thành công từ phương thức ATM, Tài khoản, thẻ quốc tế.
// Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://fail.sdk.merchantbackapp

//resultCode == 97
// kết quả thanh toán thành công từ phương thức ATM, Tài khoản, thẻ quốc tế hoặc scanQR.
// Để có action này trả về từ sdk, từ đầu return url cần redirect về URL:
http://success.sdk.merchantbackapp

    }

    // khi tắt sdk
    eventEmitter.removeAllListeners('PaymentBack')
}
})
```

Remove it

```
eventEmitter.removeAllListeners('PaymentBack')
```

Step 2: Open sdk

```
VnpayMerchant.show({
  isSandbox?: boolean
  scheme: string //to re-open app when payment success (in case, payment call
app: Mobile banking, wallet ...)
  backAlert?: string
  paymentUrl?: string
  title?: string
  titleColor?: string //6 character.
  beginColor?: string //6 character.
  endColor?: string //6 character.
  iconBackName?: string
  tmn_code: string
})
```

Step 3: Cài đặt **scheme** cho kết thành công: Khi giao dịch thành công tại ứng dụng ngân hàng / Ví. ứng dụng của ngân hàng / Ví gọi mở lại schme ứng dụng của đơn vị.

Android:

Update file: AndroidManifest.xml

```
<activity android:name="your activity"
android:screenOrientation="portrait">
  <intent-filter>
    <action android:name="android.intent.action.VIEW" />
    <category android:name="android.intent.category.BROWSABLE" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="xxxxxx" />
  </intent-filter>
</activity>
```

iOS:

Update your project setting

The screenshot shows the Xcode interface for configuring a target. The 'Info' tab is selected, and the 'URL Types (2)' section is expanded. Two URL Type configurations are visible. The first configuration has 'None' for Identifier and Icon, and 'Editor' for Role. The second configuration has 'None' for Identifier and Icon, and 'your scheme' for URL Schemes and 'Editor' for Role. The 'your scheme' text is highlighted with a red box. Below the configurations, there are sections for 'Additional url type properties' with a 'Click here to add additional url type properties' link. A plus sign is visible at the bottom of the list.

In file: AppDelegate.m

Add code

```
- (BOOL)application:(UIApplication *)application openURL:(NSURL *)url
sourceApplication:(NSString *)sourceApplication
annotation:(id)annotation{
    if ([url.scheme isEqualToString:@"your scheme"]){
        //payment result
        return YES;
    }
}
```

```
return YES;
```

```
}
```