

# TÀI LIỆU HƯỚNG DẪN TÍCH HỢP SDK FLUTTER

## EKYC

### I. Tích hợp SDK vào iOS

#### 1. Yêu cầu

- Yêu cầu có sẵn 1 dự án **Flutter**
- Mở thư mục ios trong dự án Flutter và tích hợp SDK vào dự án theo các bước của tài liệu (phần II. Tích hợp SDK vào dự án)

 **iOS VNPT\_eKYC\_Tutorial\_Integrated v3.2.2 [Tiếng Việt]**

#### 2. Thêm cầu nối

- Cầu nối với Flutter được cấu hình trong tệp AppDelegate
- Tham khảo tệp AppDelegate trong dự án tích hợp SDK mẫu
- Một số đoạn code lưu ý:

```
let controller : FlutterViewController = window?.rootViewController as!
FlutterViewController
/// điền tên tương ứng trong dự án flutter
let channel = FlutterMethodChannel(name: "flutter.sdk.ekyc/integrate",
                                   binaryMessenger: controller.binaryMessenger)
```

```
channel.setMethodCallHandler({
    (call: FlutterMethodCall, result: @escaping FlutterResult) -> Void in
    // Note: this method is invoked on the UI thread.
    // Handle battery messages.
    self.methodChannel = result
    if let info = call.arguments as? [String: String] {
        // cấu hình các thông tin về token lấy từ các key tương ứng
        // từ dự án Flutter
        ICEKYCSavedData.shared().tokenId = info["token_id"] ?? ""
        ICEKYCSavedData.shared().tokenKey = info["token_key"] ?? ""
        ICEKYCSavedData.shared().authorization = info["access_token"] ?? ""
    }

    DispatchQueue.main.async {
        // kiểm tra method tương ứng để thực hiện các hàm tương ứng
        // (các hàm này tương tự các hàm tích hợp SDK cho iOS,
        // tham khảo tài liệu đề cập ở phần I)
        // `startEkycFull`: gọi eKYC full luồng
        // `startEkycOcr`: gọi eKYC chỉ OCR giấy tờ
        // `startEkycFace`: gọi eKYC chỉ thực hiện face
        if call.method == "startEkycFull" {
```

```

        self.startEkycFull(controller)
    } else if call.method == "startEkycOcr" {
        self.startEkycOcr(controller)
    } else if call.method == "startEkycFace" {
        self.startEkycFace(controller)
    }
}
})

```

- Các phương thức còn lại tương tự việc tích hợp SDK vào dự án native iOS (tham khảo tài liệu đề cập ở mục 1. Yêu cầu)

## II. Tích hợp SDK vào Android

### 1. Yêu cầu

- Yêu cầu có sẵn 1 dự án Flutter
- Mở thư mục android trong dự án Flutter và tích hợp theo các bước theo tài liệu sau:

 Android VNPT\_eKYC\_Tutorial\_Integrated v3.3.3

### 2. Thêm cầu nối

- Tạo file cầu nối tới **Flutter** thông qua **FlutterChannel** tại **MainActivity.kt** như sau:

```

import android.app.Activity
import com.vnptit.idg.sdk.activity.VnptIdentityActivity
import com.vnptit.idg.sdk.activity.VnptOcrActivity
import com.vnptit.idg.sdk.activity.VnptPortraitActivity
import com.vnptit.idg.sdk.utils.KeyIntentConstants
import com.vnptit.idg.sdk.utils.KeyResultConstants
import com.vnptit.idg.sdk.utils.SDKEnum

import android.content.Intent
import io.flutter.embedding.android.FlutterActivity
import io.flutter.embedding.android.FlutterActivityLaunchConfigs
import io.flutter.embedding.android.TransparencyMode
import io.flutter.embedding.engine.FlutterEngine
import io.flutter.plugin.common.MethodCall
import io.flutter.plugin.common.MethodChannel
import org.json.JSONObject

class MainActivity : FlutterActivity(), MethodChannel.MethodCallHandler {
    companion object {
        private const val CHANNEL = "flutter.sdk.ekyc/integrate"
        private const val EKYC_REQUEST_CODE = 100
    }
}

```

```

}

private lateinit var channel: MethodChannel
private lateinit var result: MethodChannel.Result

override fun getTransparencyMode(): TransparencyMode {
    return TransparencyMode.transparent
}

override fun getBackgroundMode():
FlutterActivityLaunchConfigs.BackgroundMode {
    return FlutterActivityLaunchConfigs.BackgroundMode.transparent
}

override fun configureFlutterEngine(flutterEngine: FlutterEngine) {
    super.configureFlutterEngine(flutterEngine)
    channel = MethodChannel(flutterEngine.dartExecutor.binaryMessenger,
CHANNEL)
    channel.setMethodCallHandler(this)
}

override fun cleanUpFlutterEngine(flutterEngine: FlutterEngine) {
    super.cleanUpFlutterEngine(flutterEngine)
    channel.setMethodCallHandler(null)
}

override fun onMethodCall(call: MethodCall, result: MethodChannel.Result)
{
    this.result = result
    val json = parseJsonFromArgs(call)
    val intent = when (call.method) {
        "startEkycFull" -> activity.getIntentEkycFull(json)
        "startEkycOcr" -> activity.getIntentEkycOcr(json)
        "startEkycFace" -> activity.getIntentEkycFace(json)
        else -> {
            result.notImplemented()
            null
        }
    }
    intent?.let { activity.startActivityForResult(it, EKYC_REQUEST_CODE) }
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data:
Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode == EKYC_REQUEST_CODE) {
        if (resultCode == Activity.RESULT_OK) {
            if (data != null) {

```

```

        val json = JSONObject().apply {
            putSafe(KeyResultConstants.INFO_RESULT,
data.getStringExtra(KeyResultConstants.INFO_RESULT))
            putSafe(KeyResultConstants.LIVENESS_CARD_FRONT_RESULT,
data.getStringExtra(KeyResultConstants.LIVENESS_CARD_FRONT_RESULT))
            putSafe(KeyResultConstants.LIVENESS_CARD_REAR_RESULT,
data.getStringExtra(KeyResultConstants.LIVENESS_CARD_REAR_RESULT))
            putSafe(KeyResultConstants.COMPARE_RESULT,
data.getStringExtra(KeyResultConstants.COMPARE_RESULT))
            putSafe(KeyResultConstants.LIVENESS_FACE_RESULT,
data.getStringExtra(KeyResultConstants.LIVENESS_FACE_RESULT))
            putSafe(KeyResultConstants.MASKED_FACE_RESULT,
data.getStringExtra(KeyResultConstants.MASKED_FACE_RESULT))
        }
        result.success(json.toString())
    }
}
}
}

// Phương thức thực hiện eKYC luồng đầy đủ bao gồm: Chụp ảnh giấy tờ và
chụp ảnh chân dung
// Bước 1 - chụp ảnh chân dung xa gần
// Bước 2 - hiển thị kết quả
private fun Activity.getIntentEkycFace(json: JSONObject): Intent {
    val intent = getBaseIntent(VnptPortraitActivity::class.java, json)

    // Giá trị này xác định phiên bản khi sử dụng Máy ảnh tại bước chụp ảnh
chân dung luồng full. Mặc định là Normal ✓
    // - Normal: chụp ảnh chân dung 1 hướng
    // - ADVANCED: chụp ảnh chân dung xa gần
    intent.putExtra(KeyIntentConstants.VERSION_SDK,
SDKEnum.VersionSDKEnum.ADVANCED.value)

    // Bật/[Tắt] chức năng So sánh ảnh trong thẻ và ảnh chân dung
    intent.putExtra(KeyIntentConstants.IS_COMPARE_FLOW, false)

    // Bật/Tắt chức năng kiểm tra che mặt
    intent.putExtra(KeyIntentConstants.IS_CHECK_MASKED_FACE, true)

    // Lựa chọn chức năng kiểm tra ảnh chân dung chụp trực tiếp (liveness
face)
    // - NoneCheckFace: Không thực hiện kiểm tra ảnh chân dung chụp trực
tiếp hay không
    // - IBeta: Kiểm tra ảnh chân dung chụp trực tiếp hay không iBeta
(phiên bản hiện tại)
    // - Standard: Kiểm tra ảnh chân dung chụp trực tiếp hay không
Standard (phiên bản mới)

```

```

        intent.putExtra(
            KeyIntentConstants.CHECK_LIVENESS_FACE,
            SDKEnum.ModeCheckLiveNessFace.iBETA.value
        )

        return intent
    }

    // Phương thức thực hiện eKYC luồng "Chụp ảnh giấy tờ"
    // Bước 1 - chụp ảnh giấy tờ
    // Bước 2 - hiển thị kết quả
    private fun Activity.getIntentEkycOcr(json: JSONObject): Intent {
        val intent = getBaseIntent(VnptOcrActivity::class.java, json)

        // Giá trị này xác định kiểu giấy tờ để sử dụng:
        // - IdentityCard: Chứng minh thư nhân dân, Căn cước công dân
        // - IDCardChipBased: Căn cước công dân gắn Chip
        // - Passport: Hộ chiếu
        // - DriverLicense: Bằng lái xe
        // - MilitaryIdCard: Chứng minh thư quân đội
        intent.putExtra(
            KeyIntentConstants.DOCUMENT_TYPE,
            SDKEnum.DocumentTypeEnum.IDENTITY_CARD.value
        )

        // Bật/Tắt chức năng kiểm tra ảnh giấy tờ chụp trực tiếp (liveness
        card)
        intent.putExtra(KeyIntentConstants.IS_CHECK_LIVENESS_CARD, true)

        // Lựa chọn chế độ kiểm tra ảnh giấy tờ ngay từ SDK
        // - None: Không thực hiện kiểm tra ảnh khi chụp ảnh giấy tờ
        // - Basic: Kiểm tra sau khi chụp ảnh
        // - MediumFlip: Kiểm tra ảnh hợp lệ trước khi chụp (lật giấy tờ thành
        công → hiển thị nút chụp)
        // - Advance: Kiểm tra ảnh hợp lệ trước khi chụp (hiển thị nút chụp)
        intent.putExtra(
            KeyIntentConstants.TYPE_VALIDATE_DOCUMENT,
            SDKEnum.TypeValidateDocument.Basic.value
        )

        return intent
    }

    // Phương thức thực hiện eKYC luồng đầy đủ bao gồm: Chụp ảnh giấy tờ và
    chụp ảnh chân dung
    // Bước 1 - chụp ảnh giấy tờ

```

```

// Bước 2 - chụp ảnh chân dung xa gần
// Bước 3 - hiển thị kết quả
private fun Activity.getIntentEkycFull(json: JSONObject): Intent {
    val intent = getBaseIntent(VnptIdentityActivity::class.java, json)

    // Giá trị này xác định kiểu giấy tờ để sử dụng:
    // - IDENTITY_CARD: Chứng minh thư nhân dân, Căn cước công dân
    // - IDCardChipBased: Căn cước công dân gắn Chip
    // - Passport: Hộ chiếu
    // - DriverLicense: Bằng lái xe
    // - MilitaryIdCard: Chứng minh thư quân đội
    intent.putExtra(
        KeyIntentConstants.DOCUMENT_TYPE,
        SDKEnum.DocumentTypeEnum.IDENTITY_CARD.value
    )

    // Bật/Tắt chức năng So sánh ảnh trong thẻ và ảnh chân dung
    intent.putExtra(KeyIntentConstants.IS_COMPARE_FLOW, true)

    // Bật/Tắt chức năng kiểm tra ảnh giấy tờ chụp trực tiếp (liveness
card)
    intent.putExtra(KeyIntentConstants.IS_CHECK_LIVENESS_CARD, true)

    // Lựa chọn chức năng kiểm tra ảnh chân dung chụp trực tiếp (liveness
face)
    // - NoneCheckFace: Không thực hiện kiểm tra ảnh chân dung chụp trực
tiếp hay không
    // - iBETA: Kiểm tra ảnh chân dung chụp trực tiếp hay không iBeta
(phiên bản hiện tại)
    // - Standard: Kiểm tra ảnh chân dung chụp trực tiếp hay không
Standard (phiên bản mới)
    intent.putExtra(
        KeyIntentConstants.CHECK_LIVENESS_FACE,
        SDKEnum.ModeCheckLiveNessFace.iBETA.value
    )

    // Bật/Tắt chức năng kiểm tra che mặt
    intent.putExtra(KeyIntentConstants.IS_CHECK_MASKED_FACE, true)

    // Lựa chọn chế độ kiểm tra ảnh giấy tờ ngay từ SDK
    // - None: Không thực hiện kiểm tra ảnh khi chụp ảnh giấy tờ
    // - Basic: Kiểm tra sau khi chụp ảnh
    // - MediumFlip: Kiểm tra ảnh hợp lệ trước khi chụp (lật giấy tờ thành
công → hiển thị nút chụp)
    // - Advance: Kiểm tra ảnh hợp lệ trước khi chụp (hiển thị nút chụp)
    intent.putExtra(
        KeyIntentConstants.TYPE_VALIDATE_DOCUMENT,
        SDKEnum.TypeValidateDocument.Basic.value
    )

```

```

    )

    // Giá trị này xác định việc có xác thực số ID với mã tỉnh thành, quận
    huyện, xã phường tương ứng hay không.
    intent.putExtra(KeyIntentConstants.IS_VALIDATE_POSTCODE, true)

    // Giá trị này xác định phiên bản khi sử dụng Máy ảnh tại bước chụp ảnh
    chân dung luồng full. Mặc định là Normal ✓
    // - Normal: chụp ảnh chân dung 1 hướng
    // - ProOval: chụp ảnh chân dung xa gần
    intent.putExtra(KeyIntentConstants.VERSION_SDK,
        SDKEnum.VersionSDKEnum.ADVANCED.value)

    return intent
}

private fun <T : Activity> Activity.getBaseIntent(clazz: Class<T>, json:
JSONObject): Intent {
    val intent = Intent(this, clazz)

    // Nhập thông tin bộ mã truy cập. Lấy tại mục Quản lý Token
    https://ekyc.vnpt.vn/admin-dashboard/console/project-manager
    intent.putExtra(
        KeyIntentConstants.ACCESS_TOKEN,
        if (json.has("access_token")) json.getString("access_token") else ""
    )
    intent.putExtra(
        KeyIntentConstants.TOKEN_ID,
        if (json.has("token_id")) json.getString("token_id") else ""
    )
    intent.putExtra(
        KeyIntentConstants.TOKEN_KEY,
        if (json.has("token_key")) json.getString("token_key") else ""
    )

    // Giá trị này dùng để đảm bảo mỗi yêu cầu (request) từ phía khách hàng
    sẽ không bị thay đổi.
    intent.putExtra(KeyIntentConstants.CHALLENGE_CODE, "INNOVATIONCENTER")

    // Ngôn ngữ sử dụng trong SDK
    // - VIETNAMESE: Tiếng Việt
    // - ENGLISH: Tiếng Anh
    intent.putExtra(KeyIntentConstants.LANGUAGE_SDK,
        SDKEnum.LanguageEnum.VIETNAMESE.value)

    // Bật/Tắt Hiển thị màn hình hướng dẫn
    intent.putExtra(KeyIntentConstants.IS_SHOW_TUTORIAL, true)

```

// Bật chức năng hiển thị nút bấm "Bỏ qua hướng dẫn" tại các màn hình hướng dẫn bằng video

```
intent.putExtra(KeyIntentConstants.IS_ENABLE_GOT_IT, true)
```

// Sử dụng máy ảnh mặt trước

// - FRONT: Camera trước

// - BACK: Camera trước

```
intent.putExtra(  
    KeyIntentConstants.CAMERA_POSITION_FOR_PORTRAIT,  
    SDKEnum.CameraTypeEnum.FRONT.value  
)
```

```
return intent
```

```
}
```

```
private fun parseJsonFromArgs(call: MethodCall): JSONObject {
```

```
    return try {
```

```
        @Suppress("UNCHECKED_CAST")
```

```
        (JSONObject(call.arguments as Map<String, Any>))
```

```
    } catch (e: Exception) {
```

```
        JSONObject(mapOf<String, Any>())
```

```
    }
```

```
}
```

```
/**
```

```
 * put value to [JSONObject] with null-safety
```

```
 */
```

```
private fun JSONObject.putSafe(key: String, value: String?) {
```

```
    value?.let { put(key, it) }
```

```
}
```

```
}
```



### III. Xây dựng cầu nối

#### 1. Xây dựng file cầu nối

- Tạo file **sdk\_ekyc.dart** để làm cầu nối tương tác giữa Flutter và iOS, Android

```
import 'dart:convert';

import 'package:flutter/foundation.dart';
import 'package:flutter/services.dart';

class SDKEkyc {
  static final SDKEkyc _singleton = SDKEkyc._internal();

  static SDKEkyc get instance {
    return _singleton;
  }

  SDKEkyc._internal();

  Future<Map<String, dynamic>> startEkyc({required MethodEkyc method}) async {
    try {
      final result = await Channels.channel.invokeMethodOnMobile(
        method.name,
        {
          "access_token": "<ACCESS_TOKEN> (including bearer)",
          "token_id": "<TOKEN_ID>",
          "token_key": "<TOKEN_KEY>",
        },
      );

      final Map<String, dynamic> json = jsonDecode(result);

      return json.isEmpty ? {} : json;
    } on PlatformException catch (e) {
      return {"error": e.message ?? ''};
    }
  }
}

extension MethodChannelMobile on MethodChannel {
  Future<T?> invokeMethodOnMobile<T>(String method, [dynamic arguments]) {
    if (kIsWeb) {
      return Future.value(null);
    }

    return invokeMethod(method, arguments);
  }
}
```

```

}

enum MethodEkyc { full, ocr, face }

extension MethodEkycExtension on MethodEkyc {
  String get name {
    switch (this) {
      case MethodEkyc.full:
        return 'startEkycFull';
      case MethodEkyc.ocr:
        return 'startEkycOcr';
      case MethodEkyc.face:
        return 'startEkycFace';
    }
  }
}

/// Native channels.
class Channels {
  static const MethodChannel channel =
    MethodChannel('flutter.sdk.ekyc/integrate');
}

```

## 2. Ví dụ

Ví dụ sử dụng file cầu nối **sdk\_ekyc.dart** như sau:

```

final res = await SDKEkyc.instance.startEkyc(
  method: MethodEkyc.full
);
if (res.containsKey('error')) {
  // xử lý trường hợp lỗi
} else {
  // xử lý trường hợp thành công với dữ liệu trả về `res`
}

```