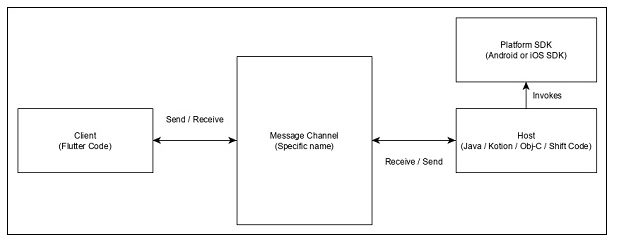
**Code với native Android**

Flutter cung cấp framework chung để truy cập vào các nền tảng có tính năng riêng biệt. Việc này giúp cho các lập trình viên có thể mở rộng các chức năng sử dụng nền tảng lập trình cụ thể như camera, pin, trình duyệt web,...Có thể dễ dàng truy cập thông qua framework

Ý tưởng chung để truy cập vào mã cụ thể của nền tảng thông qua giao thức đơn giản là **messaging.Flutter code, Client , mã nền tảng và Host** liên kết với một thông báo chung gọi là **Message Channel**. *Client*sẽ gửi thông báo đến *Host*thông qua **Message Channel.** *Host* sẽ lắng nghe từ *Message Channel*, nhận thông báo và xử lý các hàm cần thiết và cuối cùng trả kết quả về cho *Clients*thông qua Message Channel.

Dưới đây là kiến trúc platform specific code được hiển thị thông qua sơ đồi khối :



Giao thức thông báo sử dụng mã thông báo tiêu chuẩn( lớp StandardMessageCodec), được hỗ trợ tuần tự nhị phân của *JSON*- như các giá trị kiểu số , chuỗi, boolean,..serialization và de-serialization hoạt động rõ ràng giữa *Clients*và *Host*.

Hôm nay chúng ta sẽ thử viết ứng dụng đơn giản để mở trình duyệt web sử dụng Android SDK

Đầu tiên chúng ta sẽ tạo ứng dụng với tên*"flutter\_browser\_app"*

Sau đó thay thế đoạn code trong hàm **main.dart**thành :

**import** 'package:flutter/material.dart';

**void** **main**() => runApp(MyApp());

**class** **MyApp** **extends** **StatelessWidget** {

@override

Widget **build**(BuildContext context) {

**return** MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: MyHomePage(title: 'Flutter Demo Home Page'),

);

}

}

**class** **MyHomePage** **extends** **StatelessWidget** {

MyHomePage({Key key, **this**.title}) : **super**(key: key);

**final** String title;

@override

Widget **build**(BuildContext context) {

**return** Scaffold(

appBar: AppBar(

title: Text(**this**.title),

),

body: Center(

child: RaisedButton(

child: Text('Open Browser'),

onPressed: **null**,

),

),

);

}

}

Ở đoạn code trên, ta đã tạo một nút có chức năng để mở trình duyệt và tạm thời ta set nó ở trạng thái *NULL*

Bây giờ ta sẽ import đoạn code dưới đây :

**import** 'dart:async';

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

Tiếp theo ta sẽ viết phương thức**\_openBrowser** để gọi nền tảng cụ thể  thông qua  *message channel.*

Future<void> \_openBrowser() **async** {

**try** {

final int result = **await** platform.invokeMethod(

'openBrowser', <String, String>{

'url': "https://flutter.dev"

}

);

}

on PlatformException **catch** (e) {

// Unable to open the browser

print(e);

}

}

Ở đây chúng ta sử dụng *platform.invokeMethod* để gọi *openBrowser*(giải thích ở bước tiếp theo ), *openBrowser*có đối số, url để mở url cụ thể

Giờ chúng ta sẽ thay đổi đối số **null** trong *RaiseButton*thành *\_openBrowser*

onPressed: \_openBrowser,

Tiếp đến bạn hãy mở **MainActivity,java**(bên trong thư mục android) và *import*một số thư viện sau đây:

**import** **android**.app.Activity;

**import** **android**.content.Intent;

**import** **android**.net.Uri;

**import** **android**.os.Bundle;

**import** **io**.flutter.app.FlutterActivity;

**import** **io**.flutter.plugin.common.MethodCall;

**import** **io**.flutter.plugin.common.MethodChannel;

**import** **io**.flutter.plugin.common.MethodChannel.MethodCallHandler;

**import** **io**.flutter.plugin.common.MethodChannel.Result;

**import** **io**.flutter.plugins.GeneratedPluginRegistrant;

Bây giờ chúng ta sẽ viết hàm *openBrowser*để mở trình duyệt nhé :

private void openBrowser(MethodCall call, Result result, String url) {

Activity activity = **this**;

**if** (activity == null) {

result.error("ACTIVITY\_NOT\_AVAILABLE",

"Browser cannot be opened without foreground

activity", null);

**return**;

}

Intent intent = **new** Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse(url));

activity.startActivity(intent);

result.success((Object) true);

}

Trong hàm **MainActivity**ta đặt tên Channel ;

**private** **static** **final** String CHANNEL = "flutterapp.tutorialspoint.com/browser";

Ở hàm *Oncreate*ta sẽ viết mã cụ thể cho *Android*để xử lý message

**new** MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(

**new** MethodCallHandler() {

@Override

**public** void onMethodCall(MethodCall call, Result result) {

String url = call.argument("url");

**if** (call.method.equals("openBrowser")) {

openBrowser(call, result, url);

} **else** {

result.notImplemented();

}

}

})

Nhìn trên, chúng ta đã tạo *message channe*sử dụng lớp **MethodChannel**và lớp **MethodCallHandler**để xử lý thông báo. **onMethodCall** có trách nhiệm gọi đúng mã nền tảng riêng biệt bẳng cách kiếm tra thông báo. Hàm **onMethodCall**đọc url từ thông báo và gọi đến openBrowser khi mà hàm gọi openBrowser. Ngược lại hàm sẽ trả về method *notImplemented*

OKI giờ chúng ta xem toàn bộ code nhé :

**MainActivity.java**

**package** com.tutorialspoint.flutterapp.flutter\_browser\_app;

**import** android.app.Activity;

**import** android.content.Intent;

**import** android.net.Uri;

**import** android.os.Bundle;

**import** io.flutter.app.FlutterActivity;

**import** io.flutter.plugin.common.MethodCall;

**import** io.flutter.plugin.common.MethodChannel.Result;

**import** io.flutter.plugins.GeneratedPluginRegistrant;

**public** **class** **MainActivity** **extends** **FlutterActivity** {

**private** **static** **final** String CHANNEL = "flutterapp.tutorialspoint.com/browser";

@Override

**protected** **void** **onCreate**(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

GeneratedPluginRegistrant.registerWith(**this**);

**new** MethodChannel(getFlutterView(), CHANNEL).setMethodCallHandler(

**new** MethodCallHandler() {

@Override

**public** **void** **onMethodCall**(MethodCall call, Result result) {

String url = call.argument("url");

**if** (call.method.equals("openBrowser")) {

openBrowser(call, result, url);

} **else** {

result.notImplemented();

}

}

}

);

}

**private** **void** **openBrowser**(MethodCall call, Result result, String url) {

Activity activity = **this**; **if** (activity == **null**) {

result.error(

"ACTIVITY\_NOT\_AVAILABLE", "Browser cannot be opened without foreground activity", **null**

);

**return**;

}

Intent intent = **new** Intent(Intent.ACTION\_VIEW);

intent.setData(Uri.parse(url));

activity.startActivity(intent);

result.success((Object) **true**);

}

}

**main.dart**

**import** 'package:flutter/material.dart';

**import** 'dart:async';

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

**void** **main**() => runApp(MyApp());

**class** **MyApp** **extends** **StatelessWidget** {

@override

Widget **build**(BuildContext context) {

**return** MaterialApp(

title: 'Flutter Demo',

theme: ThemeData(

primarySwatch: Colors.blue,

),

home: MyHomePage(

title: 'Flutter Demo Home Page'

),

);

}

}

**class** **MyHomePage** **extends** **StatelessWidget** {

MyHomePage({Key key, **this**.title}) : **super**(key: key);

**final** String title;

**static** **const** platform = **const** **MethodChannel**('flutterapp.tutorialspoint.com/browser');

Future<**void**> **\_openBrowser**() async {

**try** {

**final** **int** result = await platform.invokeMethod('openBrowser', <String, String>{

'url': "https://flutter.dev"

});

}

on PlatformException **catch** (e) {

// Unable to open the browser print(e);

}

}

@override

Widget **build**(BuildContext context) {

**return** Scaffold(

appBar: AppBar(

title: Text(**this**.title),

),

body: Center(

child: RaisedButton(

child: Text('Open Browser'),

onPressed: \_openBrowser,

),

),

);

}

}

Giờ chúng ta chạy thử ứng dụng. Sau khi run ứng dụng, ta sẽ thấy nút Open browser như hình dưới, rồi ấn vào nó

