**Giới thiệu về package**

Cách mà Dart tổ chức và chia sẻ các chức năng thông package. Dart Package là một thư viện hay mô hình đơn giản có thể chia sẻ. Nhìn chung, Dart package cũng giống như Dart Application ngoại trừ Dart Package không truy cập vào các điểm chính cùa ứng dung

Cấu trúc chung của Package ( ví dụ về package) dưới đây :

- **lib/src/\* :**tệp Dart ở dạng priavte

- **lib/my\_demo\_package.dart :**phần code chính của Dart, có thể thêm một vài ứng dụng

**import** 'package:my\_demo\_package/my\_demo\_package.dart'

- Một vài tệp ở dạng private có thể được xuất sang tệp chính (my\_demo\_package.dart) :

**export** src/my\_private\_code.dart

- **lib/\***: Ta có thể truy cập vào bất kì tệp nào bên trong thư mục  :

**import** 'package:my\_demo\_package/custom\_folder/custom\_file.dart'

**- pubspec.yaml :**Được hiểu là trình quản lý thư mục của  Package

Để tích hợp được các gói vào dự án thì ta cần phải  có file pubspec.yaml

**Các kiểu Package :**

Kể từ khi Dart package  là một collection có chức năng tương tự , nó có thể được phân loại dựa trên chức năng:

**Dart Package**

Chúng ta có thể sử dụng Dart trên cả 2 môi trường là web và android. Ví dụ , english\_words là một package chứa khoảng 500 từ và có chức năng tiệng ích cơ bản như danh từ ( list các danh từ trong English), âm tiết (liệt kê ra các từ có âm tiết đặc biệt )

**Flutter package**

Phụ thuộc vào Flutter framework và có thể chỉ sử dụng trong môi trường mobile .

**Flutter plugin**

Phụ thuộc vào Flutter framework cũng như nền tảng cơ bản (Android SDK hay iOS SDK). Ví dụ Camera là một plugin (có thể hiểu là một phần mềm hỗ trợ) để tương tác với thiết bị camera. Nó sử dụng SDK để có quyền truy cập vào camera

**Sử dụng Dart Package :**

Dart package được lưu trữ và publish trên các máy chủ, https://pub.dev . Ngoài ra, Flutter cung cấp các tool, pub cơ bản để quản lý các Dart package trong ứng dụng. Các bước cần để sử dụng Package như sau :

-Nhập tên package và phiên bản phù hợp trong file pubspec.yaml như dưới đây :

**dependencies**: **english\_words**: ^3.1.5

-Bản mới nhất sẽ được cập nhật trên server

- Cài đặt package bằng lệnh :

flutter packages **get**

- Khi chúng ta đang dùng Android studio, thì Android studio sẽ phát hiện bất kì thay đổi trong file pubspec.yaml và hiện thông báo để lập trình viên có thể biết

- Dart package có thể được cài đặt hoặc nâng cấp trong Android studio thông qua menu optionsoptions .

- Thêm các file cần thiết sử dụng lệnh dưới đây và bắt đầu làm việc :

**import** 'package:english\_words/english\_words.dart';

- Sử dụng bất kì phương thức có sẵn

**nouns**.take(50).forEach(print);

- Ở trên ta đã dùng hàm nouns để lấy ra 50 từ đầu tiên

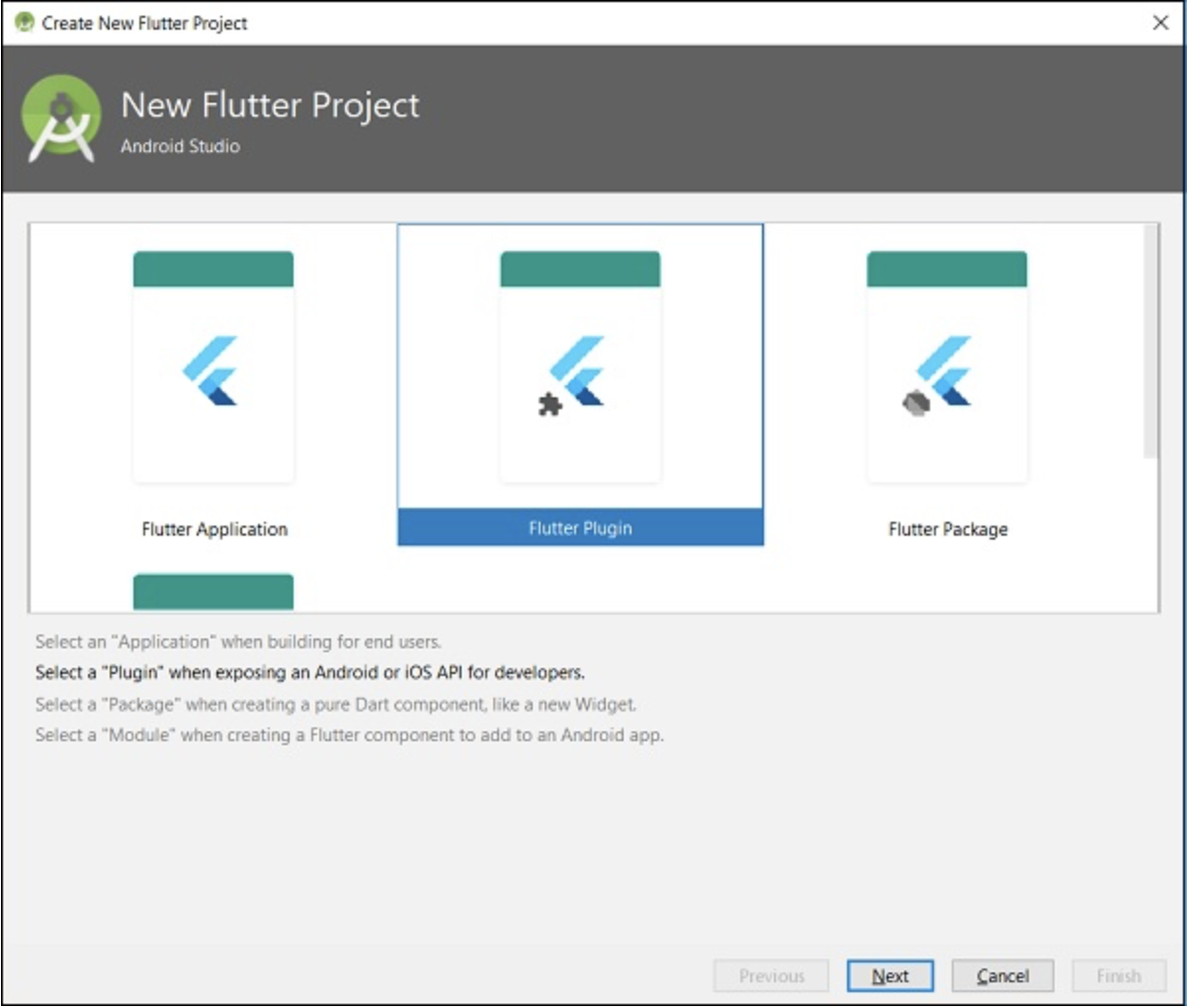
**Làm việc với Flutter plugin package**

Làm việc với Flutter plugin cũng giống như làm việc với Dart package hay Dart application. Chỉ khác ở chỗ Plugin sẽ sử dụng System API(Android hay iOS) để có được những chắc năng cụ thể cần thiết

Như chúng ta đã được tìm hiểu cách truy cập vào các nền tảng riêng của hệ điều hành ở bài trước, hôm nay chúng ta sẽ tự  xây dựng một plugin đơn giản là my\_browser . Chức năng của my\_browser là cho phép ứng dụng mở trình duyện riêng của nền tảng ( IOS hay Android)

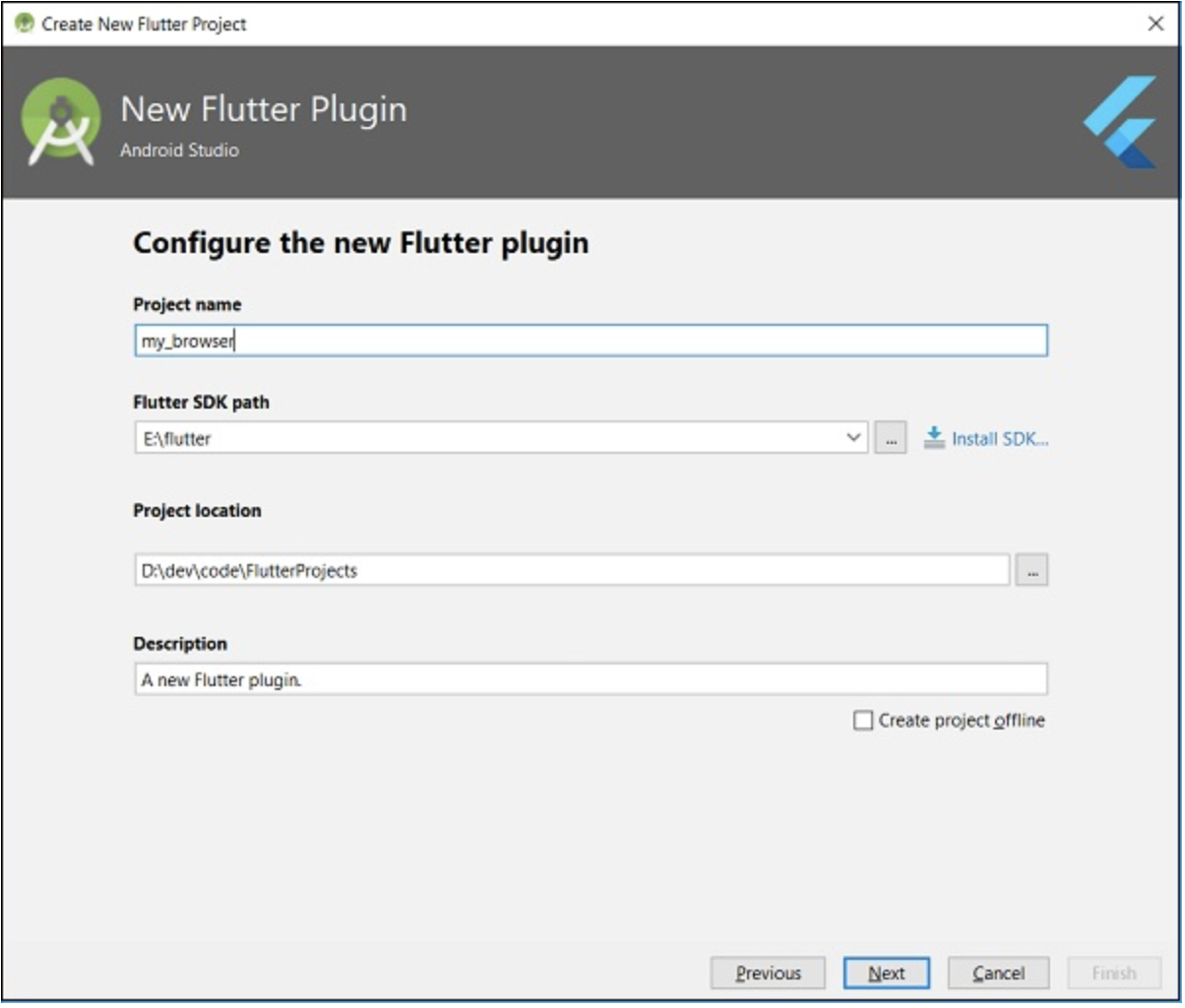
- Đầu tiên mở Android studio

- Nhấn file -> New flutter project và chọn futter plugin

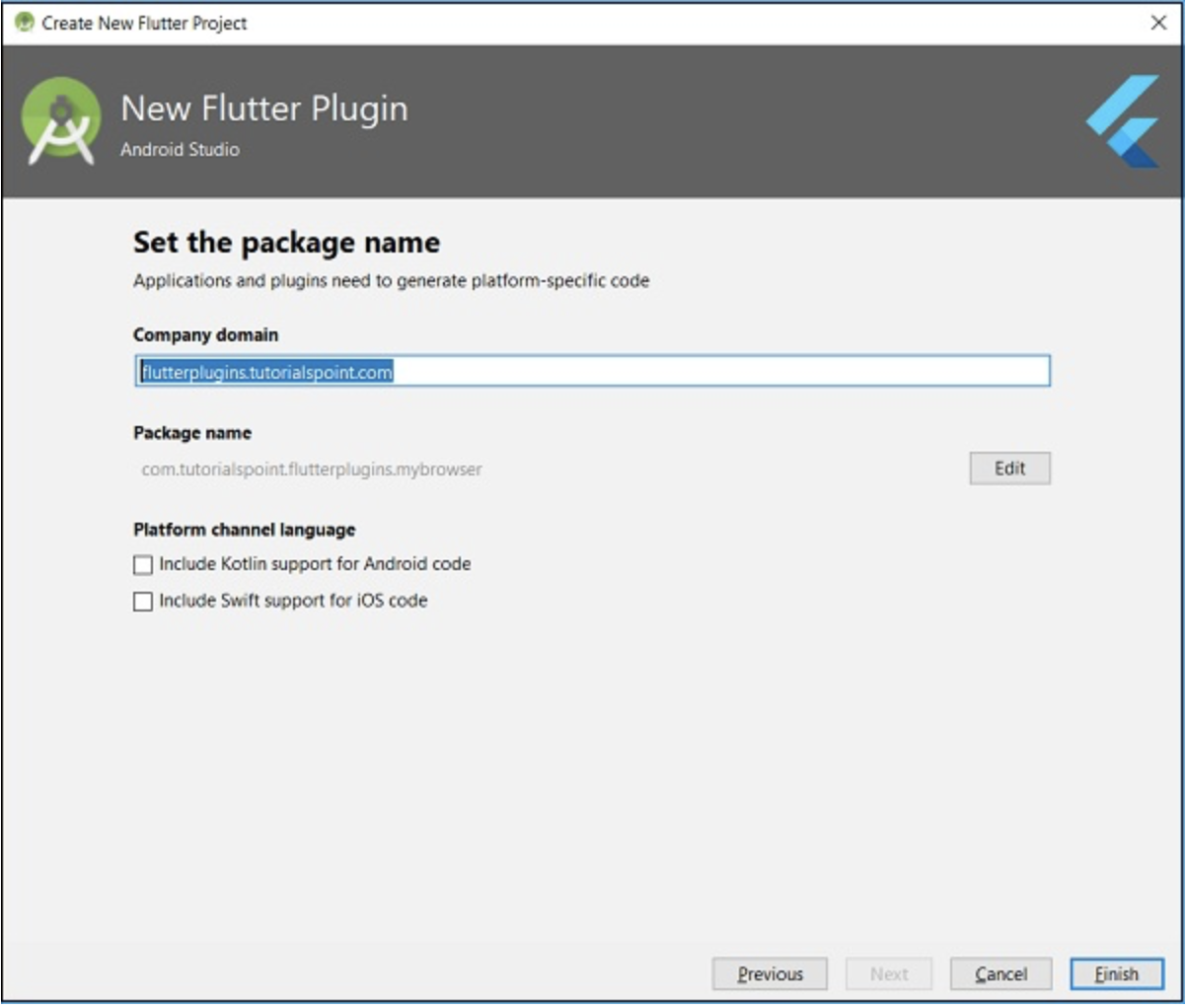


- Nhập tên project "my\_browser" và nhấn Next

- Nhập tên plugin và các thông tin khác như sau :



- Nhập tên miền công ty tên bất kì bạn muốn có dạng tenproject.tendomaincompany.com (không quan trọng)



- Mở file my\_browser.dart và viết các phương thức, openBrowser dùng để gọi phương thức trong nền tảng riêng ( android hay ios )

Future<void> openBrowser(String urlString) **async** {

**try** {

final int result = **await** \_channel.invokeMethod(

'openBrowser', <String, String>{ 'url': urlString }

);

}

on PlatformException **catch** (e) {

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

}

}

- Mở file MyBrowserPlugin.java và import một vài lớp như sau :

**import** android.app.Activity;

**import** android.content.Intent;

**import** android.net.Uri;

**import** android.os.Bundle;

- Ở đây , chúng ta phải thêm thư viện cần thiết để mở browser từ Android

- Thêm các giá trị mRegistrar thuộc kiểu Registrar trong lớp MyBrowserPlugin ở dạng private

**private** **final** Registrar mRegistrar;

- Ở trên Registrar được dùng để lấy thông tin context của code được gọi

- Thêm constructor

**private** **MyBrowserPlugin**(Registrar registrar) {

**this**.mRegistrar = registrar;

}

- Thay đổi registerWith bao gồm constructor mới trong lớp MyBrowserPlugin

**public** **static** **void** **registerWith**(Registrar registrar) {

**final** MethodChannel channel = **new** MethodChannel(registrar.messenger(), "my\_browser");

MyBrowserPlugin instance = **new** MyBrowserPlugin(registrar);

channel.setMethodCallHandler(instance);

}

- Thay đổi onMethodCall gồm hàm openBrowser trong lớp MyBrowserPlugin

@Override

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

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

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

result.success("Android " + android.os.Build.VERSION.RELEASE);

}

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

openBrowser(call, result, url);

} **else** {

result.notImplemented();

}

}

- Sau đó viết hàm openBrowser để truy cập vào browser trong lớp MyBrowserPlugin

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

Activity activity = mRegistrar.activity();

**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);

}

Toàn bộ source code của my\_browser plugin như sau :

**my\_browser.dart**

**import** 'dart:async';

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

**class** **MyBrowser** {

**static** **const** MethodChannel \_channel = **const** MethodChannel('my\_browser');

**static** Future<String> **get** platformVersion async {

final String version = **await** \_channel.invokeMethod('getPlatformVersion'); **return** version;

}

Future<**void**> openBrowser(String urlString) **async** {

**try** {

final int result = **await** \_channel.invokeMethod(

'openBrowser', <String, String>{'url': urlString});

}

on PlatformException **catch** (e) {

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

}

}

}

**MyBrowserPlugin.java**

**package** com.tutorialspoint.flutterplugins.my\_browser;

**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.plugin.common.PluginRegistry.Registrar;

**import** android.app.Activity;

**import** android.content.Intent;

**import** android.net.Uri;

**import** android.os.Bundle;

/\*\* MyBrowserPlugin \*/

**public** **class** **MyBrowserPlugin** **implements** **MethodCallHandler** {

**private** **final** Registrar mRegistrar;

**private** **MyBrowserPlugin**(Registrar registrar) {

**this**.mRegistrar = registrar;

}

/\*\* Plugin registration. \*/

**public** **static** **void** **registerWith**(Registrar registrar) {

**final** MethodChannel channel = **new** MethodChannel(

registrar.messenger(), "my\_browser");

MyBrowserPlugin instance = **new** MyBrowserPlugin(registrar);

channel.setMethodCallHandler(instance);

}

@Override

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

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

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

result.success("Android " + android.os.Build.VERSION.RELEASE);

}

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

openBrowser(call, result, url);

} **else** {

result.notImplemented();

}

}

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

Activity activity = mRegistrar.activity();

**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**);

}

}

- Tạo project mới, lấy tên tuỳ bạn , ở đây mình sẽ dùng *my\_browser\_plugin\_test*để đặt tên cho project

- Trong file pubspec.yaml ta thêm plugin my\_browser trong dependency.

dependencies:

flutter:

sdk: flutter

my\_browser:

path: ../my\_browser

- Android studio sẽ thông báo rằng file pubspec.yam cần được cập nhật và hiển thị bạn chỉ cần click vào get dependency

- Trong file main.dart và my\_browser plugin ta cần thêm thư viện vào :

**import** 'package:my\_browser/my\_browser.dart';

- Gọi hàm openBrowser từ my\_browser plugin như sau :

onPressed: () => MyBrowser().openBrowser("https://flutter.dev"),

Okey, dưới đây là toàn bộ code trong hàm main.dartdart

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

**import** 'package:my\_browser/my\_browser.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: () => MyBrowser().openBrowser("https://flutter.dev"),

),

),

);

}

}

- Sau khi hoàn chỉnh ta sẽ chạy ứng dụng lên và click và nút open browser (hiển thị ở giữa màn hình) như hình dưới đây

A picture containing text, software, computer, web page

Description automatically generated

- Sau khi nhất nút open browser sẻ mở ra trình duyệt như ảnh dưới đây :

