

# Cách sử dụng các tệp

1. Tạo dự án Flutter mới: flutter create lab4\_2
2. Thay thế thư mục lib / và tệp pubspec.yaml bằng các tệp đính kèm
3. Có thể thêm hình ảnh, font chữ
4. Chạy lệnh: flutter packages get
5. flutter run (với trình giả lập đang chạy hoặc thiết bị thực được kết nối với máy) để xem ứng dụng

## Nội dung:

### I. Sử dụng gói (packages)

Flutter hỗ trợ sử dụng các gói chia sẻ được đóng góp bởi các nhà phát triển khác cho hệ sinh thái Flutter và Dart. Điều này cho phép nhanh chóng xây dựng một ứng dụng mà không phải phát triển mọi thứ từ đầu.

Các gói hiện có cho phép nhiều trường hợp sử dụng, ví dụ, thực hiện các yêu cầu mạng ( http), điều hướng / xử lý tuyến tùy chỉnh ( fluro), tích hợp với API thiết bị ( url\_launcher và battery) và sử dụng SDK nền tảng của bên thứ ba như Firebase ( FlutterFire ).

Để viết một gói mới, xem các gói đang phát triển . Để thêm tài sản, hình ảnh hoặc phông chữ, cho dù được lưu trữ trong tệp hoặc gói, hãy xem Thêm tài sản và hình ảnh.

### 1. Sử dụng gói

#### a. Tìm kiếm gói

Các gói được xuất bản đến pub.dev

Các trang đích Flutter trên màn pub.dev gói hàng đầu mà tương thích với Flutter (những người thực hiện kê khai phụ thuộc thường tương thích với Flutter), và hỗ trợ tìm kiếm trong tất cả các gói được công bố.

Trang Flutter Favorites trên pub.dev liệt kê các plugin và gói đã được xác định là gói trước tiên bạn nên xem xét sử dụng khi viết ứng dụng của mình. Để biết thêm thông tin về ý nghĩa của việc yêu thích Flutter, hãy xem chương trình Flutter Favorites .

Cũng có thể duyệt các gói trên pub.dev bằng cách lọc trên các plugin Android , plugin iOS , plugin web hoặc bất kỳ sự kết hợp nào của chúng.

#### b. Thêm một phụ thuộc gói vào một ứng dụng

Để thêm gói `css_colors`, vào một ứng dụng:

1. Mở `pubspec.yaml` tệp nằm trong thư mục ứng dụng và thêm vào `css_colors`: bên dưới `dependencies`.

2. Cài đặt nó

- Từ thiết bị đầu cuối: Chạy `flutter pub get`.

HOẶC LÀ

- Từ Android Studio / IntelliJ: Nhấp vào `Packages get` có trong dải băng hành động ở đầu `pubspec.yaml`.

- Từ Mã VS: Nhấp vào `Nhận Packages get` nằm ở bên phải của dải băng hành động ở đầu `pubspec.yaml`.

3. Import: Thêm một `import` tương ứng trong mã Dart.

4. Dừng và khởi động lại ứng dụng, nếu cần

Nếu gói mang mã riêng cho nền tảng (Kotlin / Java cho Android, Swift / Objective-C cho iOS), mã đó phải được tích hợp vào ứng dụng. Tải lại và khởi động lại chỉ cập nhật mã Dart, vì vậy có thể cần phải khởi động lại toàn bộ ứng dụng để tránh các lỗi như `MissingPluginException` khi sử dụng gói.

### c. Giải quyết xung đột

Giả sử bạn muốn sử dụng `some_package` và `another_package` trong một ứng dụng, và cả hai đều phụ thuộc vào `url_launcher`, nhưng trong các phiên bản khác nhau. Điều đó gây ra một cuộc xung đột tiềm năng. Cách tốt nhất để tránh điều này là cho các tác giả gói sử dụng phạm vi phiên bản thay vì các phiên bản cụ thể khi chỉ định phụ thuộc.

```
dependencies:  
  flutter:  
    sdk: flutter  
  css_colors: ^1.0.0  
  url_launcher: ^5.4.0  
  image_picker: '5.4.3'
```

Nếu `some_package` khai báo các phụ thuộc ở trên và `another_package` tuyên bố một `url_launcher` phụ thuộc tương thích như `'5.4.6'` hoặc `^5.5.0`, pub sẽ tự động giải quyết vấn đề. Sự phụ thuộc của nền tảng cụ thể vào các mô-đun Gradle hoặc CocoaPods được giải quyết theo cách tương tự.

Ngay cả khi `some_package` và `another_package` tuyên bố các phiên bản không tương thích cho `url_launcher`, chúng thực sự có thể sử dụng `url_launcher` theo những cách tương

thích. Trong tình huống này, xung đột có thể được giải quyết bằng cách thêm khai báo ghi đè phụ thuộc vào pubspec.yaml tệp của ứng dụng, buộc sử dụng một phiên bản cụ thể.

Ví dụ: để buộc sử dụng url\_launcher phiên bản 5.4.0, hãy thực hiện các thay đổi sau đối với pubspec.yaml tệp của ứng dụng

```
dependencies:
  flutter:
    sdk: flutter
  css_colors: ^1.0.0
  # url_launcher: ^5.4.0
  # image_picker: '5.4.3'
  some_package:
  another_package:
dependency_overrides:
  url_launcher: '5.4.0'
```

## II. Phát triển các gói mới

### 1. Quản lý phụ thuộc gói và phiên bản

#### a. Phiên bản trọn gói

Tất cả các gói có số phiên bản, được chỉ định trong pubspec.yaml tệp của gói. Phiên bản hiện tại của gói được hiển thị bên cạnh tên của nó (ví dụ: xem url\_launcher gói), cũng như danh sách tất cả các phiên bản trước (xem url\_launcher phiên bản).

Khi một gói được thêm vào pubspec.yaml, biểu mẫu tốc ký plugin1: có nghĩa là bất kỳ phiên bản nào của gói plugin1 đều có thể được sử dụng. Để đảm bảo ứng dụng không bị hỏng khi gói được cập nhật, hãy chỉ định phạm vi phiên bản bằng một trong các định dạng sau:

- Giới hạn phạm vi: Chỉ định phiên bản tối thiểu và tối đa. Ví dụ:

```
dependencies:
  url_launcher: '>=5.4.0 <6.0.0'
```

- Các ràng buộc phạm vi với cú pháp dấu mũ tương tự như các ràng buộc phạm vi thông thường:

```
dependencies:
  collection: '^5.4.0'
```

## b. Cập nhật phụ thuộc gói

Khi chạy flutter pub get( Gói nhận được trong IntelliJ hoặc Android Studio) lần đầu tiên sau khi thêm gói, Flutter lưu phiên bản gói cụ thể được tìm thấy trong pubspec.lock tệp khóa . Điều này đảm bảo rằng sẽ có được phiên bản tương tự một lần nữa nếu chạy flutter pub get.

Để nâng cấp lên phiên bản mới của gói, ví dụ: để sử dụng các tính năng mới trong gói đó, hãy chạy flutter pub upgrade ( Nâng cấp phụ thuộc trong IntelliJ hoặc Android Studio) để truy xuất phiên bản có sẵn cao nhất của gói được cho phép bởi ràng buộc phiên bản được chỉ định trong pubspec.yaml.

## c. Phụ thuộc vào các gói chưa được công bố

Các gói có thể được sử dụng ngay cả khi không được công bố trên pub.dev. Đối với các plugin riêng hoặc cho các gói chưa sẵn sàng để xuất bản, các tùy chọn phụ thuộc bổ sung có sẵn:

- Phụ thuộc tuyến

Ứng dụng Flutter có thể phụ thuộc vào plugin thông qua path: phụ thuộc hệ thống tệp . Đường dẫn có thể là tương đối hoặc tuyệt đối. Ví dụ: để phụ thuộc vào plugin plugin1 nằm trong thư mục bên cạnh ứng dụng, hãy sử dụng cú pháp sau:

```
dependencies:  
  plugin1:  
    path: ../plugin1/
```

- Phụ thuộc Git

Cũng có thể phụ thuộc vào gói được lưu trữ trong kho Git. Nếu gói nằm ở gốc của repo, hãy sử dụng cú pháp sau:

```
dependencies:  
  plugin1:  
    git:  
      url: git://github.com/flutter/plugin1.git
```

- Git phụ thuộc vào một gói trong một thư mục

Công cụ pub giả định gói được đặt trong thư mục gốc của kho Git. Nếu đó không phải là trường hợp, chỉ định vị trí với pathđổi số. Ví dụ:

```
dependencies:  
  package1:  
    git:
```

```
url: git://github.com/flutter/packages.git
path: packages/package1
```

### III. Ví dụ

#### 1. Sử dụng gói `css_colors`

Các `css_colors` gói định nghĩa hằng số màu cho màu sắc CSS, vì vậy sử dụng hằng bất cứ nơi nào trong khuôn khổ Flutter hy vọng Color loại

Để sử dụng

B1. Tạo 1 dự án mới

B2. Mở `pubspec.yaml` và thêm `css-colors` phụ thuộc

```
dependencies:
  flutter:
    sdk: flutter
```

B3. Chạy `flutter pub get` trong thiết bị đầu cuối hoặc nhấp vào Gói có trong IntelliJ hoặc Android Studio.

B4. Mở `lib/main.dart` và thay thế nội dung đầy đủ của nó bằng:

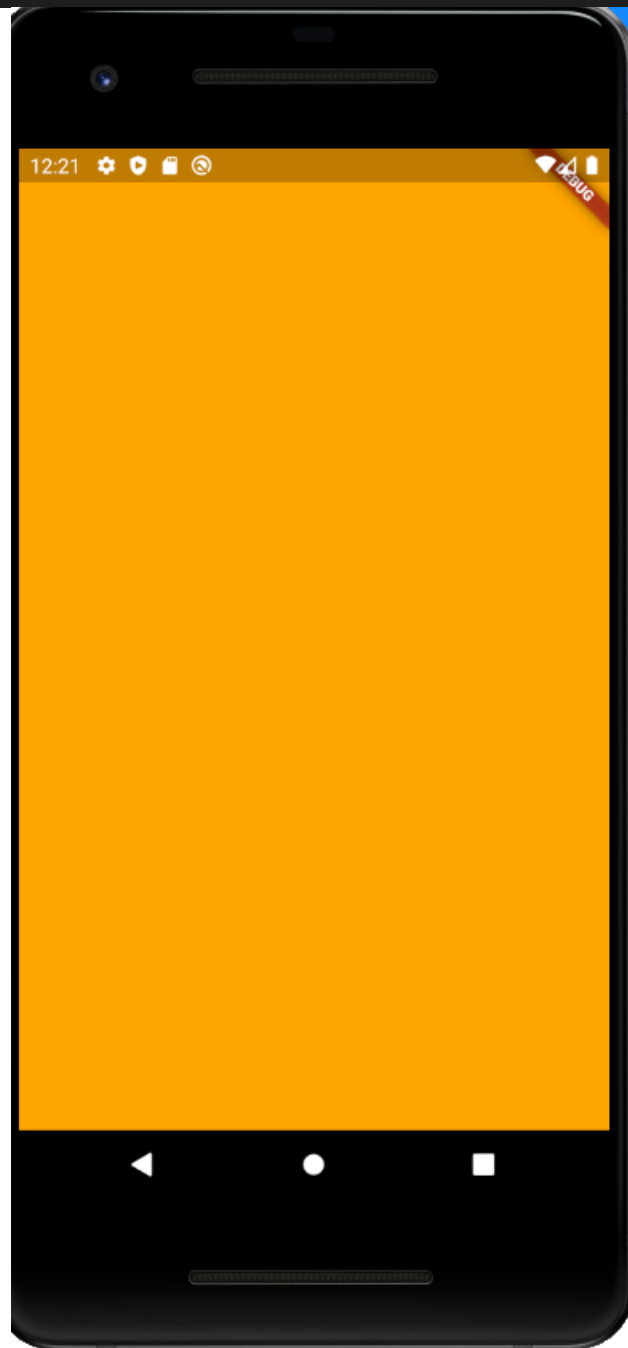
```
import 'package:css_colors/css_colors.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: DemoPage(),
    );
  }
}

class DemoPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(body: Container(color: CSSColors.orange));
  }
}
```

```
}
```



## 2. Sử dụng gói url\_launcher để khởi chạy trình duyệt

Các url\_launcher gói Plugin cho phép mở trình duyệt mặc định trên nền tảng điện thoại di động để hiển thị một URL nào đó, và được hỗ trợ trên Android, iOS, web, và macos. Gói này là gói Dart đặc biệt gọi là gói plugin (hoặc plugin ), bao gồm mã dành riêng cho nền tảng.

Để sử dụng

B1. Tạo dự án mới

B2. Mở pubspec.yaml và thêm url\_launcher phụ thuộc:

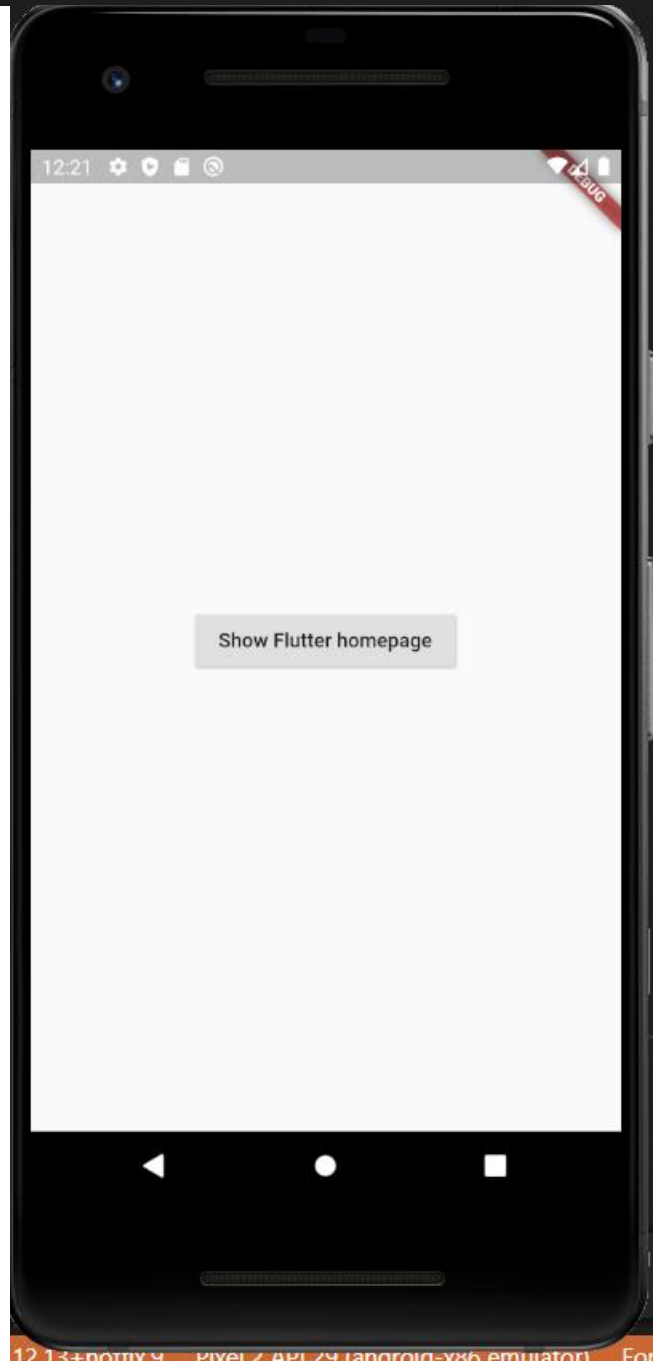
```
dependencies:  
  flutter:  
    sdk: flutter  
  url_launcher: ^5.4.0
```

B3. Chạy flutter pub get trong thiết bị đầu cuối hoặc nhấp vào Gói có trong IntelliJ hoặc Android Studio.

B4. Mở lib/main.dart và thay thế nội dung đầy đủ của nó bằng cách sau:

```
import 'package:flutter/material.dart';  
import 'package:url_launcher/url_launcher.dart';  
  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: DemoPage(),  
    );  
  }  
}  
  
class DemoPage extends StatelessWidget {  
  launchURL() {  
    launch('https://flutter.dev');  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: RaisedButton(  
          onPressed: launchURL,  
          child: Text('Show Flutter homepage'),  
        ),  
      ),  
    );  
  }  
}
```

```
}  
}
```



12:13:40 PM - Pixel 4 API 29 (android-x86\_64 emulator) - For