

Truy xuất các nội dung trong thiết bị



Đọc danh bạ điện thoại

- Sử dụng các Plugin sau đây để đọc/truy cập danh bạ điện thoại
 - flutter_contact: Truy cập và quản lý danh bạ trên OS của thiết bị
 - fluttercontactpicker: Mở danh bạ gốc của OS để chọn một contact
- Ví dụ: Sử dụng fluttercontactpicker để mở danh bạ thiết bị và chọn một contact
 - Định nghĩa lớp Contact

```
class Contact{
  String name, phone;
  Contact({required this.name, required this.phone});
}
```

Nội dung

- Đọc thông tin các contact trong thiết bị
- Gọi điện trong flutter app
- Gửi tin nhắn SMS trong flutter app
- Mở mail client để gửi email trong flutter app
- Lập trình phần cứng trong flutter:
 - Đọc thông tin thiết bị
 - Thông tin kết nối mạng
 - Thông tin các cảm biến
 - Kết nối Bluetooth, Bluetooth Low Energy

Chọn một contact

```
Future<Contact> _pickContact() async{
  bool permission = await FlutterContactPicker.hasPermission();
  if(permission ==false) {
    bool requestsPermission = await FlutterContactPicker.requestPermission();
    if (requestsPermission == true)
      permission = true;
  }
  if(permission == true){
    PhoneContact contact = await FlutterContactPicker.pickPhoneContact();
    String contactName = contact.fullName?? "No name";
    String phoneNumber = contact.phoneNumber?.number?? "No Phone";
    return Contact(name: contactName, phone: phoneNumber);
  }
  else return Future.error("Chưa cấp quyền đọc danh bạ");
}
```

Phone, SMS, Email, Web Url

- Trong một số trường hợp,
 - một ứng dụng Flutter có thể cần phải mở ứng dụng gọi điện/gửi tin nhắn của thiết bị để điền số điện thoại cho người sử dụng quay số/gửi tin nhắn.
 - Mở ứng dụng mail client trên máy để người sử dụng gửi email
 - Mở trình duyệt web để hiển thị một trang web
- Trong các trường hợp như vậy, người sử dụng có thể sử dụng plugin `url_launcher` để mở các ứng dụng đã được cài đặt trên OS của thiết bị để thực hiện các yêu cầu

Email, Web Url

- Mở mail client để gửi email

```
Future<bool> _sendEmail(String to) async{
  final Uri emailLaunchUri = Uri(
    scheme: 'mailto',
    path: to,
  );
  bool check = await canLaunch(emailLaunchUri.toString());
  if(check ==false) return false;
  else return launch(emailLaunchUri.toString());
}
```

- Mở trình duyệt để đi tới địa chỉ website:

```
_gotoWebsite(String url) {
  String httpUrl = url.startsWith("http") ? url : "http://$url";
  launch(httpUrl);
}
```

Phone, SMS

- import: `import 'package:url_launcher/url_launcher.dart';`
- Mở ứng dụng quay số điện thoại và điền số điện thoại


```
Future<bool> openPhoneDial(String phoneNumber) async{
  bool check = await canLaunch('tel:$phoneNumber');
  if(check ==false) return false;
  else return launch('tel:$phoneNumber');
}
```
- Mở ứng dụng SMS để thực hiện gửi tin nhắn cho một contact

```
Future<bool> openSMS(String phoneNumber) async{
  bool check = await canLaunch('sms:$phoneNumber');
  if(check ==false) return false;
  else return launch('sms:$phoneNumber');
}
```

AndroidManifest.xml

Bắt đầu từ API 30, Android yêu cầu cấu hình package rõ ràng trong `AndroidManifest.xml` của bạn, nếu không `canLaunch` sẽ trả về false. Phần tử `<queries>` phải được thêm vào tệp manifest của bạn dưới dạng phần tử con của phần tử gốc.

```
<queries>
  <!-- If your app opens https URLs -->
  <intent>
    <action android:name="android.intent.action.VIEW" />
    <data android:scheme="https" />
  </intent>
  <!-- If your app makes calls -->
  <intent>
    <action android:name="android.intent.action.DIAL" />
    <data android:scheme="tel" />
  </intent>
  <!-- If your app emails -->
  <intent>
    <action android:name="android.intent.action.SEND" />
    <data android:mimeType="*/*" />
  </intent>
</queries>
```

Lập trình phần cứng trong Flutter

- Đọc thông tin thiết bị trong Flutter
 - Đọc thông tin thiết bị
 - Thông tin kết nối mạng
 - Thông tin các cảm biến
 - Kết nối Bluetooth, Bluetooth Low Energy
 - Truy cập camera

Đọc thông tin thiết bị Android


```
Future<Map<String, dynamic>> getAndroidDeviceInfo() async{
  DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
  AndroidBatteryInfo? androidBatteryInfo =
    await BatteryInfoPlugin().androidBatteryInfo;
  AndroidDeviceInfo deviceInfoData = await deviceInfo.androidInfo;
  return {
    "manufacturer": deviceInfoData.manufacturer,
    'display': deviceInfoData.display,
    'hardware': deviceInfoData.hardware,
    'android.version': deviceInfoData.version,
    'system.features': deviceInfoData.systemFeatures,
    'battery.capacity': androidBatteryInfo?.batteryCapacity?.toString()?? "No info",
    'battery.health': androidBatteryInfo?.health?? "No info",
    'battery.temperature': androidBatteryInfo?.temperature?.toString()?? "No info"
  };
}
```

Đọc thông tin thiết bị

- device_info: Đọc thông tin về phần cứng thiết bị: Màn hình, OS, các cảm biến... (Sử dụng device_info_plus để thay thế do device_info không được tiếp tục phát triển)
- battery_plus: Đọc thông tin về pin, trạng thái pin, mức pin...
- battery_info: cung cấp thông tin chi tiết về pin (mức pin, "sức khỏe", trạng thái sạc, v.v.). Hỗ trợ cả iOS và Android.

Đọc thông tin network

- Đọc thông tin của mạng bao gồm:
 - Tên mạng wifi
 - Wifi BSSID (Basic Service Set Identification): **Mã định danh bộ dịch vụ cơ bản** - Một mã định danh duy nhất cho mỗi thiết bị không dây. BSSID là địa chỉ MAC Ethernet của thiết bị.
 - Wifi IPv4
 - Wifi IPv6
 - Wifi Submask
 - Wifi Broadcast: Địa chỉ quảng bá của mạng
 - Địa chỉ cổng mặc định (Default Gateway)
- Sử dụng plugin: network_info_plus 1.1.0

 Kết nối bluetooth

- flutter_blue
- flutter_reactive_ble

Huỳnh Tuấn Anh - ĐHNT13