Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 13: *Permissions & Quản lý Runtime Permissions*

Phòng LT & Mạng

http://csc.edu.vn/lap-trinh-va-csdl

2019

Nội dung

1. Permissions
2. Quản lý Runtime Permissions

Permission là gì?

- Mục đích của permission là để bảo vệ sự riêng tư của người dùng
- Bất kỳ 1 app nào muốn truy cập vào các dữ liệu riêng tư của người dùng (vd: tin nhắn SMS) đều phải xin phép người dùng
- Phân loại:
  - **Normal permissions**: là các permission được tự động cho phép bởi hệ thống, bởi ít nguy hại đến người dùng, hệ thống
  - **Dangerous permissions (Runtime Permissions)**: là các permission cần được người dùng cho phép để app có thể truy cập vào các dữ liệu riêng tư, nhạy cảm của người dung
- Các permission cần được khai báo trong **AndroidManifest.xml**

Normal permissions

- Các **normal permissions** được liệt kê tại https://developer.android.com/guide/topics/permissions/overview#normal_permissions
- Cần được liệt kê trong file **AndroidManifest.xml** như sau:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.csc.permissiondemo">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.SET_ALARM" />

    <application …>

    </application>

</manifest>
```

Runtime Permissions

- Các **runtime permissions** được liệt kê tại: https://developer.android.com/guide/topics/permissions/overview#dangerous_permissions
- Tương tự **normal permissions**, cần được liệt kê trong **AndroidManifest.xml**

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.csc.permissiondemo">

    <uses-permission android:name="android.permission.CALL_PHONE" />

    <application …>


</manifest>
```

- Tiếp theo, cần xin phép người dùng các permission này lúc "runtime"

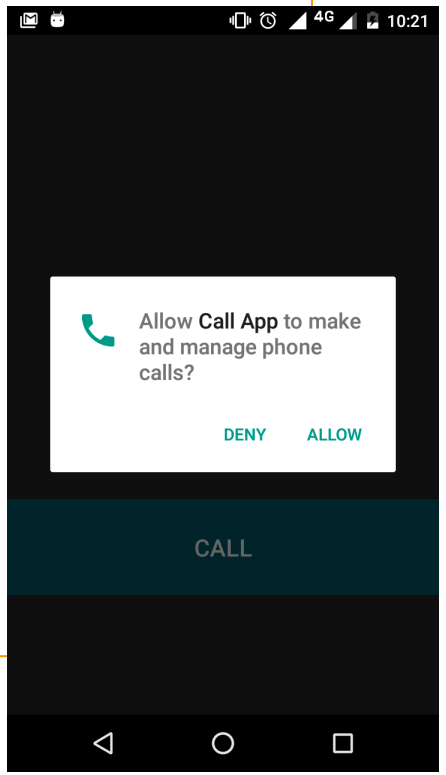
Xin phép người dùng

- **Ví dụ**: để thực hiện một phone call, cần kiểm tra người dùng đã cấp quyền trước đó hay chưa, nếu chưa cần xin phép:

```
private static final int PHONE_CALL_PERMISSIONS_REQUEST = 101;

private void makePhoneCall() {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new String[] {Manifest.permission.CALL_PHONE},
PHONE_CALL_PERMISSIONS_REQUEST);
        }
        return;
    }

    Intent callIntent = new Intent(Intent.ACTION_CALL);
    callIntent.setData(Uri.parse("tel:0288777888"));
    if (callIntent.resolveActivity(getPackageManager()) != null) {
        startActivity(callIntent);
    }
}
```



Xin phép người dùng (tiếp)

- Override phương thức onRequestPermissionResult cho activity/fragment để xử lý kết quả xin phép người dùng (granted hoặc denied)

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    if (requestCode == PHONE_CALL_PERMISSIONS_REQUEST && grantResults.length == 1
            && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        makePhoneCall();
    } else {
        Toast.makeText(this, "Cannot make a call", Toast.LENGTH_SHORT).show();
    }
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
}
```

Các runtime permissions thường gặp

Permission Group	Permissions
CALENDAR	READ_CALENDAR WRITE_CALENDAR
CALL_LOG	READ_CALL_LOG WRITE_CALL_LOG PROCESS_OUTGOING_CALLS
CAMERA	CAMERA
CONTACTS	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO
PHONE	READ_PHONE_STATE READ_PHONE_NUMBERS CALL_PHONE ANSWER_PHONE_CALLS ADD_VOICEMAIL USE_SIP
SENSORS	BODY_SENSORS
SMS	SEND_SMS RECEIVE_SMS READ_SMS RECEIVE_WAP_PUSH RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE

Q&A