

Android动态权限申请

 可乐游侠 [关注](#)

 2 2018.01.19 21:02:14 字数 1,058 阅读 51,509

从Android 6.0开始，权限不再是在manifest文件中粘贴一下即可，这时候权限也正式走进大家的视野。项目的6.0适配就是我做的，当时没有仔细总结，最近在另一个项目添加权限的时候发现，同一个功能都没有添加申请权限的代码，一个会挂一个不会，花了几个小时在这个小问题上。因此多花点时间总结一下权限问题。

1. Android系统权限的概念

Android是一个权限分隔的操作系统，每个应用都有独特的系统标识。一般情况下，应用没有权限执行对其它应用、系统、用户可能有不利影响的操作。每个应用都在应用沙盒中运行，因此当应用需要使用沙盒未提供的功能时，需要申请权限，比如读写sd卡、访问网络、访问其它应用的数据、读写联系人、调用摄像头等。

权限在AndroidManifest.xml文件中声明，Android 6.0以前，有的APP一股脑声明了各种各样的权限，用户可能没有细看就安装了，于是这些APP就可以为所欲为，偷鸡摸狗，无法无天。Android 6.0把权限分成正常权限和危险权限，AndroidManifest中声明的正常权限系统会自动授予，而危险权限则需要在使用的时候用户明确授予。

换句话说，就是Android 6.0以上的系统在第一次使用危险权限的时候，需要向用户申请，征得用户的同意。如果还是在没有权限的情况下执行操作，就会获得Crash大礼包，错误日志为 `java.lang.SecurityException: Permission Denial`。因此，应用对危险权限的申请，需要相应的处理。

2. 危险权限和对应的权限组

危险权限都属于权限组，应用在向用户申请危险权限时，系统会弹对话框，描述应用要访问的权限组，这时候用户如果同意授权，则权限组包含的所有权限都会被系统授予。比如，应用被授予READ_EXTERNAL_STORAGE权限之后，如果再申请WRITE_EXTERNAL_STORAGE权限，系统会立即授予该权限。

危险权限表格如下

权限组	权限
CALENDAR	READ_CALENDAR
	WRITE_CALENDAR
CAMERA	CAMERA
CONTACTS	READ_CONTACTS
	WRITE_CONTACTS
	GET_ACCOUNTS
LOCATION	ACCESS_FINE_LOCATION
	ACCESS_COARSE_LOCATION
MICROPHONE	RECORD_AUDIO

权限组	权限
PHONE	READ_PHONE_STATE
	CALL_PHONE
	READ_CALL_LOG
	WRITE_CALL_LOG
	ADD_VOICEMAIL
	USE_SIP
	PROCESS_OUTGOING_CALLS
SENSORS	BODY_SENSORS
SMS	SEND_SMS
	RECEIVE_SMS
	READ_SMS
	RECEIVE_WAP_PUSH
	RECEIVE_MMS
STORAGE	READ_EXTERNAL_STORAGE
	WRITE_EXTERNAL_STORAGE

3. 申请权限的正确姿势

- 需要申请的所有权限在AndroidManifest文件中声明

```
1 | <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
2 | <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

- 使用时检查权限，没有权限则申请

```
//使用兼容库就无需判断系统版本
int hasWriteStoragePermission = ContextCompat.checkSelfPermission(getApplication(), Manifest.permission.WRITE_EXTERNAL_STORAGE);
if (hasWriteStoragePermission == PackageManager.PERMISSION_GRANTED) {
    //拥有权限，执行操作
    initScan();
}else{
    //没有权限，向用户请求权限
    ActivityCompat.requestPermissions(thisActivity, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
}
```

- 覆写onRequestPermissionsResult方法

```
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
    //通过requestCode来识别是否同一个请求
    if (requestCode == CODE_FOR_WRITE_PERMISSION){
        if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            //用户同意，执行操作
            initScan();
        }else{
            //用户不同意，向用户展示该权限作用
            if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
                new AlertDialog.Builder(thisActivity)
                    .setMessage(R.string.storage_permissions_remind)
                    .setPositiveButton("OK", (dialog1, which) -> {
                        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, EventConstConfig.CODE_FOR_CAMERA_PERMISSION);
                    })
                    .setNegativeButton("Cancel", null)
                    .create()
                    .show();
            }
        }
    }
}
```

```

    }
  }
}

```

shouldShowRequestPermissionRationale方法返回值分几种情况，怎么使用看应用的具体交互需求。

1. 第一次请求该权限，返回false。
2. 请求过该权限并被用户拒绝，返回true。
3. 请求过该权限，但用户拒绝的时候勾选不再提醒，返回false。

4. 申请权限的几个小细节

- 使用兼容库

checkSelfPermission、requestPermissions等几个权限相关的方法用v4包里的可以兼容6.0以下版本，否则需要包一层版本判断。

- targetSDKVersion的问题

我遇到的问题就是这个，有个细节没注意到。Android系统触发动态申请权限的条件其实有两个，设备系统版本在Android 6.0以上并且targetSDKVersion>=23。因此其实在targetSDKVersion版本小于23的情况下，即使在6.0以上的设备运行也不会挂，但会在安装时列出所有权限，同6.0以下的设备。官方建议保持targetSDKVersion在最新的版本。

5. 使用第三方库easypermissions

申请权限的第三方库有很多，但没遇到特别简洁的，也就没有特地去研究和使用的，避免导入太多第三方库，easypermissions这个库是其中一个项目用到的，这里简单的提一下。

- 导入easypermissions

```
1 | compile 'pub.devrel:easypermissions:0.1.9'
```

- 检查权限，没有就进行申请

```

1 |         if (!EasyPermissions.hasPermissions(this, Manifest.permission.CAMERA)) {
2 |             EasyPermissions.requestPermissions(this, getString(R.string.camera_permission), 1, this);
3 |         }

```

- 覆写方法

```

1 | @Override
2 | public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
3 |     super.onRequestPermissionsResult(requestCode, permissions, grantResults);
4 |     // Forward results to EasyPermissions
5 |     EasyPermissions.onRequestPermissionsResult(requestCode, permissions, grantResults, this);
6 | }
7 |
8 | @Override
9 | public void onPermissionsGranted(int requestCode, List<String> perms) {
10 | //     light();
11 | }
12 |
13 | @Override
14 | public void onPermissionsDenied(int requestCode, List<String> perms) {
15 |     Dialog dialog = new AlertDialog.Builder(this, R.style.MyAlertDialogStyle)
16 |         .setTitle(R.string.tips)
17 |         .setMessage(R.string.camera_permission_tip)
18 |         .setPositiveButton(R.string.to_open, (dialog2, which) -> {
19 |             startActivity(new Intent(Settings.ACTION_APPLICATION_SETTINGS));
20 |         })
21 |         .setNegativeButton(R.string.cancel, (dialog3, which) -> {
22 |             dialog3.dismiss();

```

```
23         }  
24         .create();  
25         dialog.show();  
26     }
```




28人点赞>



 Android





可乐游侠

Trial and Error.

总资产3

共写了1.1W字

获得45个赞

共6个粉丝

关注