**Android 6.0 变更**

运行时权限 此版本引入了一种新的权限模式，如今，用户可直接在运行时管理应用权限。这种模式让用户能够更好地了解和控制权限，同时为应用开发者精简了安装和自动更新过程。用户可为所安装的各个应用分别授予或撤销权限。

Android 6.0（API 级别 23）或更高版本为目标平台的应用，请务必在运行时检查和请求权限。要确定您的应用是否已被授予权限，请调用新增的 checkSelfPermission() 方法。要请求权限，请调用新增的requestPermissions() 方法。即使您的应用并不以 Android 6.0（API 级别 23）为目标平台，您也应该在新权限模式下测试您的应用。

**一、说明**

我们都知道，在Android5.0（包括5.0）以下我们的权限是在Manifest文件中申请即可。但是到了Android6.0就不是这样了。google对于权限问题要求更为严格了。将权限进行了如下划分：

**正常权限和危险权限**

normal：这个权限类型并不直接威胁到用户的隐私，可以直接在manifest清单里注册，系统会帮我们默认授权的。

dangerous：这个可以直接给app访问用户一些敏感的数据，不仅需要在manifest清单里注册，同时在使用的时候，需要向系统请求授权。

值得注意一点，这里有特殊权限授权的区别，分别是SYSTEM\_ALERT\_WINDOW 和 WRITE\_SETTINGS，虽然这两个权限也是属于dangerous权限类型，但是这两个授权请求方式和其他dangerous权限是不一样的，需要特殊处理 。

**二、权限管理中实现**

**(1) 声明目标SDK版本**

我们需要在build.gradle中声明targetSdkVersion为23或更高版本

**(2)** [**Android 6.0 SDK 找不到HttpClient**](http://www.cnblogs.com/jxldjsn/p/6001577.html)

原因：android 6.0 （api 23）SDK，不再提供org.apache.http.\*(只保留几个类)。

解决方法：android studio中在相应的module（module .app , module .protocal）下的build.gradle中加入:

android{

    useLibrary 'org.apache.http.legacy'

}

**(3) >=6.0编译下Notification中setLatestEventInfo()方法被弃用**

解决方法：采用Notification.Builder来代替

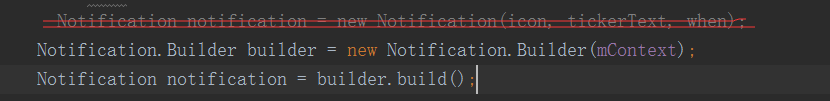
Notification.Builder builder = new Notification.Builder(mContext);

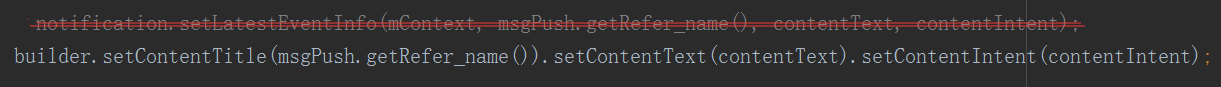
Notification notification = builder.build();

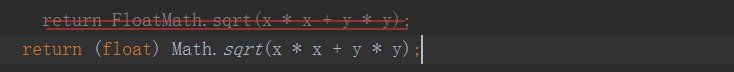
builder.setContentTitle(msgPush.getRefer\_name()).setContentText(contentText).setContentIntent(contentIntent);

FloatMath.sqrt()弃用：

解决:(float) Math.sqrt();







**(4)权限管理实现方式**

**第一步：检查并申请权限**

 要检查是否具有某项权限，请调用 ContextCompat.checkSelfPermission() 方法,如果应用具有此权限，该方法将返回 PackageManager.PERMISSION\_GRANTED，并且应用可以继续操作。如果应用不具有此权限，方法将返回PERMISSION\_DENIED，且应用必须明确向用户要求权限。

     如果应用尚无所需的权限，则应用必须调用一个 requestPermissions() 方法，以请求适当的权限。应用将传递其所需的权限，以及您指定用于识别此权限请求的整型请求代码。此方法异步运行：它会立即返回，并且在用户响应对话框之后，系统会使用结果调用应用的回调方法，将应用传递的相同请求代码传递到 requestPermissions()。

第一个参数是Context；第二个参数是需要申请的权限的字符串数组；第三个参数为requestCode，主要用于回调的时候检测。可以从方法名requestPermissions以及第二个参数看出，是支持一次性申请多个权限的，系统会通过对话框**逐一**询问用户是否授权。

* if (ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE\_EXTERNAL\_STORAGE)
* != PackageManager.PERMISSION\_GRANTED) {
* //申请WRITE\_EXTERNAL\_STORAGE权限
* ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE},
* WRITE\_EXTERNAL\_STORAGE\_REQUEST\_CODE);
* }

**第二步：处理权限请求结果**

请求权限后，系统会弹出请求权限的Dialog, 用户选择允许或需要后，会回调onRequestPermissionsResult方法, 该方法类似于onActivityResult，我们接着需要根据requestCode和grantResults(授权结果)做相应的后续处理，如果你同时申请两个权限，那么grantResults的length就为2，分别记录你两个权限的申请结果。

* @Override
* public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {
* super.onRequestPermissionsResult(requestCode, permissions, grantResults);
* if (requestCode == WRITE\_EXTERNAL\_STORAGE\_REQUEST\_CODE) {
* if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {
* // permission was granted, yay! Do 通过授权
* // contacts-related task you need to do.
* } else {
* // permission denied, boo! Disable the 未通过授权  
   // functionality that depends on this permission.
* }
* }
* }

**总结：**

**1. 首先要知道6.0版本权限模型跟原来版本是不同的，不再是统一在manifest中默认系统授权，而是有需要的时候，向系统请求授权，提高用户体验。**

**2. 了解权限检测流程，一点注意点是如果系统权限弹窗提示框被不再提醒了，需要我们自定义提示弹窗，引导用户去授权。**

**3. 在判断APP是否运行在Android M上，可以用版本号来判断，可以准确点。**

**4. 对于Intent这种方式，很多情况下是不需要授权的甚至权限都不需要的，比如：你是到拨号界面而不是直接拨打电话，就不需要去申请权限；打开系统图库去选择照片；调用系统相机app去拍照等**

**5. 实际上，Android的权限系统中涵盖了两类权限：PROTECTION\_NORMAL、非PROTECTION\_NORMAL。其中，PROTECTION\_NORML类权限在AndroidManifest.xml中声明后，在APP安装时，系统将无需检查权限而直接授予，且用户不可取消，如蓝牙、网络、闪光灯、NFC、WIFI等等；非PROTECTION\_NORMAL类权限在AndroidManifest.xml中声明，在安装APP时，系统将会询问APP每次请求的权限，用户可取消此类权限。系统有时候不会弹出权限授权对话框，需要APP主动检查有没有该类权限，并弹出授权提示对话框，我们处理的就是这类权限。另外，Android权限系统是以组的形式管理权限，比如发送短信、接收短信同属一组权限，一旦其他一个获得授予，其他同组组员都得到了授权。”**

**//健保通项目中使用的dangerous权限列表**

<uses-permission android:name="android.permission.GET\_ACCOUNTS" />// Google 帐户。

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />//手机状态

<uses-permission android:name="android.permission.CALL\_PHONE"/>//拨打电话

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

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />//精确定位

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />//范围性地址

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />//写外部储存

<uses-permission android:name="android.permission.RECORD\_AUDIO" />//音频

**//健保通项目中目前需要主动获取的权限有四类（定位,相机,文件读写,音频）**

**MainActivity 定位权限**

**BasicMessageActivity相机权限，文件读写权限**

**BaseUploadActivity 相机权限，文件读写权限**

**BaiduMapaddmarkerActivity 定位权限**

**RetailShopListActivity 定位**

**GlobalSearchActivity 文件读写权限**

**MessageChatActivity 文件读写权限，音频权限，相机权限**

**UploadReportActivity 相机权限，文件读写权限。**

**FitnessCoachActivity 文件读写权限**

**MedicationReminderAddActivity 相机 ，文件读写权限**

**HospitalListActivity 定位权限**

**FamilyChatActivity 音频权限 ，文件读写权限,相机权限**

**NewAddressActivity 读写权限**

**FamilyPersonInfoActivity 文件读写权限,相机权限**

**MyAccountActivity 文件读写权限, 相机权限**

normal列表：

ACCESS\_LOCATION\_EXTRA\_COMMANDS

ACCESS\_NETWORK\_STATE

ACCESS\_NOTIFICATION\_POLICY

ACCESS\_WIFI\_STATE

BLUETOOTH

BLUETOOTH\_ADMIN

BROADCAST\_STICKY

CHANGE\_NETWORK\_STATE

CHANGE\_WIFI\_MULTICAST\_STATE

CHANGE\_WIFI\_STATE

DISABLE\_KEYGUARD

EXPAND\_STATUS\_BAR

GET\_PACKAGE\_SIZE

INSTALL\_SHORTCUT

INTERNET

KILL\_BACKGROUND\_PROCESSES

MODIFY\_AUDIO\_SETTINGS

NFC

READ\_SYNC\_SETTINGS

READ\_SYNC\_STATS

RECEIVE\_BOOT\_COMPLETED

REORDER\_TASKS

REQUEST\_INSTALL\_PACKAGES

SET\_ALARM

SET\_TIME\_ZONE

SET\_WALLPAPER

SET\_WALLPAPER\_HINTS

TRANSMIT\_IR

UNINSTALL\_SHORTCUT

USE\_FINGERPRINT

VIBRATE

WAKE\_LOCK

WRITE\_SYNC\_SETTINGS

dangerous列表:

group:android.permission-group.CONTACTS

permission:android.permission.WRITE\_CONTACTS

permission:android.permission.GET\_ACCOUNTS

permission:android.permission.READ\_CONTACTS

group:android.permission-group.PHONE

permission:android.permission.READ\_CALL\_LOG

permission:android.permission.READ\_PHONE\_STATE

permission:android.permission.CALL\_PHONE

permission:android.permission.WRITE\_CALL\_LOG

permission:android.permission.USE\_SIP

permission:android.permission.PROCESS\_OUTGOING\_CALLS

permission:com.android.voicemail.permission.ADD\_VOICEMAIL

group:android.permission-group.CALENDAR

permission:android.permission.READ\_CALENDAR

permission:android.permission.WRITE\_CALENDAR

group:android.permission-group.CAMERA

permission:android.permission.CAMERA

group:android.permission-group.SENSORS

permission:android.permission.BODY\_SENSORS

group:android.permission-group.LOCATION

permission:android.permission.ACCESS\_FINE\_LOCATION

permission:android.permission.ACCESS\_COARSE\_LOCATION

group:android.permission-group.STORAGE

permission:android.permission.READ\_EXTERNAL\_STORAGE

permission:android.permission.WRITE\_EXTERNAL\_STORAGE

group:android.permission-group.MICROPHONE

permission:android.permission.RECORD\_AUDIO

group:android.permission-group.SMS

permission:android.permission.READ\_SMS

permission:android.permission.RECEIVE\_WAP\_PUSH

permission:android.permission.RECEIVE\_MMS

permission:android.permission.RECEIVE\_SMS

permission:android.permission.SEND\_SMS

permission:android.permission.READ\_CELL\_BROADCASTS