



kingsley1212

码龄3年 暂无认证

11 17万+ 30万+ 1万+  
原创 周排名 总排名 访问 等级

255 1 3 12 11  
积分 粉丝 获赞 评论 收藏



TA的主页

私信

关注

搜博主文章



广告 关闭

科大讯飞全球开发者大会  
AI焕新 更美好  
10.22-10.24 相约云端 共享A.I.

预约参会

热门文章

Android 中RxPermissions 的使用 6490

Butterknife Unbinder 在fragment和  
viewPager使用闪退问题 639

## Android 中RxPermissions 的使用

翻译

kingsley1212

2019-07-12 16:05:28

6532 收藏 10

分类专栏:

android

文章标签:

Rxpermission

android

权限

## Android 中RxPermissions 的使用方法详解,

以请求拍照、读取位置权限为例

第一步, 在module的build.gradle中的

```
1 dependencies {  
2     // RxPermissions 的使用方法详解  
3     implementation 'com.github.tbruyelle:rxpermissions:0.10.2'  
4     implementation 'io.reactivex.rxjava2:rxandroid:2.1.0'  
5     implementation 'io.reactivex.rxjava2:rxjava:2.2.4'  
6 }
```

还有在最外面那个文件夹下面的build.gradle 加入

```
1 allprojects {  
2     repositories {  
3         ...  
4         maven { url 'https://jitpack.io' }  
5     }  
6 }
```

第二步, 在AndroidManifest.xml中添加自己需要用到的权限:

```
1 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>  
2 <uses-permission android:name="android.permission.CAMERA"/>
```

Android 防止按钮重复点击 547

android EditText输入金额，限制位数和显示小数点后几位 532

android获取屏幕宽高 493

#### 分类专栏



popupwindow

1篇



android

16篇

#### 最新评论

Butterknife Unbinder 在fragment和viewP...  
qq\_36975897: 好的，谢了

Butterknife Unbinder 在fragment和viewP...  
kingsley1212 回复 qq\_36975897: [code=java] @Override @CallSuper public void un...

Butterknife Unbinder 在fragment和viewP...  
qq\_36975897: 如果不解绑会不会内存泄漏啊

Butterknife Unbinder 在fragment和viewP...  
kingsley1212 回复 景二倩: 都去掉了呀，删除就好

Butterknife Unbinder 在fragment和viewP...  
景二倩: 都去去除掉private Unbinder unbinder了，那onDetach方法中的unbinder哪里来...

#### 最新文章

PopupWindow工具类

二分查找

## 第三步，activity中使用以及注意事项

### 1,创建 RxPermissions 实例

```
1 RxPermissions rxPermissions = new RxPermissions(this);
```

注意：this 参数可以是 FragmentActivity 或 Fragment。如果你在 fragment 中使用 RxPermissions，你应该传递 fragment 实例作为构造函数参数而不是 fragment.getActivity()，否则你可能抛出异常 java.lang.IllegalStateException: FragmentManager is already executing[执行]

### 2、请求权限，必须在初始化阶段比如 onCreate 中调用

如上所述，由于您的应用程序可能在权限请求期间重新启动，因此请求必须在初始化阶段完成。这可能是 Activity.onCreate 或 View.onFinishInflate，但不能在 onResume 等 pausing 方法中，因为由于您请求的 activity 在权限请求期间被框架暂停了，因此您可能会创建一个无限的请求循环。

如果没有，并且如果您的应用程序在权限请求期间重新启动（例如，由于配置更改），则用户的响应将永远不会发送给订阅者。

### 3、RxPermissions 里面的方法总结，

request	不支持返回权限名; 返回的权限结果:全部同意时值true,否则值为false	申请结果:false
requestEach	把每一个权限的名称和申请结果都列出来	权限名称:android.permission.READ_CONTACTS,申请结果:false权限名称:android.permission.WRITE_CONTACTS,申请结果:false权限名称:android.permission.READ_SMS,申请结果:false权限名称:android.permission.ACCESS_FINE_LOCATION,申请结果:true权限名

一个好用的tablayout

2020年 1篇

2019年 15篇

目录

[Android 中RxPermissions 的使用方法详解,](#)

个人demo

github地址:

		称:android.permission.ACCESS_COARSE_LOCATION, 申请结果:true	
requestEachCombined	返回的权限名称:将多个权限名合并成一个; 返回的权限结果:全部同意时值true,否则值为false	权限名称:android.permission.READ_CONTACTS, android.permission.WRITE_CONTACTS, android.permission.READ_SMS, android.permission.ACCESS_FINE_LOCATION, android.permission.ACCESS_COARSE_LOCATION,申请结果:false	
ensureEach	必须配合rxjava,回调结果跟requestEach一样		
ensureEachCombined	必须配合rxjava,回调结果跟requestEachCombined一样		

#### 4、RxPermissions 里面的方法具体实现，

```
1 public class RxPermissionsActivity extends AppCompatActivity {  
2     /**  
3      * 权限组  
4      */  
5     private static final String[] permissionsGroup = new String[]{  
6         Manifest.permission.ACCESS_FINE_LOCATION,  
7         Manifest.permission.CAMERA};  
8  
9     @Override  
10    protected void onCreate(Bundle savedInstanceState) {  
11        super.onCreate(savedInstanceState);  
12        setContentView(R.layout.activity_rx_permissions);  
13        findViewById(R.id.onclick_tv).setOnClickListener(new View.OnClickListener()  
14            @Override  
15            public void onClick(View v) {  
16                testRequestEach(v);  
17            }  
18    })  
19 }
```

```

18     });
19 }
20
21 /*
22     private void 方法一() {
23     //         创建 RxPermissions 实例
24         RxPermissions rxPermissions = new RxPermissions(RxPermissionsActivit
25         rxPermissions.request(permissionsGroup)
26             .subscribe(new Observer<Boolean>() {
27                 @Override
28                 public void onSubscribe(Disposable d) {
29
30                 }
31
32                 @Override
33                 public void onNext(Boolean aBoolean) {
34                     if (aBoolean) {
35                         Toast.makeText(RxPermissionsActivity.this, "已获取权限
36                             .show();
37                     } else {
38 //只有用户拒绝开启权限，且选了不再提示时，才会走这里，否则会一直请求开启
39                         Toast.makeText(RxPermissionsActivity.this, "主人，
40                             .show();
41                     }
42                 }
43                 @Override
44                 public void onError(Throwable e) {
45
46                 }
47
48                 @Override
49                 public void onComplete() {
50
51                 }
52             });
53 }*/
54

```

```

55  /**
56   * request例子:
57   * 不支持返回权限名;
58   * 返回的权限结果:全部同意时值true,否则值为false
59   */
60  public void testRequest(View view) {
61      RxPermissions rxPermissions = new RxPermissions(this);
62      rxPermissions.request(permissionsGroup)
63          .subscribe(new Consumer<Boolean>() {
64              @Override
65              public void accept(Boolean aBoolean) throws Exception {
66
67              }
68          });
69  }
70
71  /**
72   * requestEach例子:
73   * 把每一个权限的名称和申请结果都列出来
74   */
75  public void testRequestEach(View view) {
76      RxPermissions rxPermissions = new RxPermissions(this);
77      rxPermissions.requestEach(permissionsGroup)
78          .subscribe(new Consumer<Permission>() {
79              @Override
80              public void accept(Permission permission) throws Exception {
81                  if (permission.granted) {
82                      // 用户已经同意该权限
83                  } else if (permission.shouldShowRequestPermissionRationale) {
84                      // 用户拒绝了该权限, 没有选中『不再询问』 (Never ask again)
85                  } else {
86                      // 用户拒绝了该权限, 而且选中『不再询问』那么下次启动时, 就不
87                  }
88              }
89          });
90  }
91  }

```

```

92
93 /**
94  * requestEachCombined例子:
95  * 返回的权限名称:将多个权限名合并成一个
96  * 返回的权限结果:全部同意时值true,否则值为false
97  */
98 public void testRequestEachCombined(View view) {
99     RxPermissions rxPermissions = new RxPermissions(this);
100     rxPermissions.requestEachCombined(permissionsGroup)
101         .subscribe(new Consumer<Permission>() {
102             @Override
103             public void accept(Permission permission) throws Exception {
104
105             }
106         });
107 }
108
109 /**
110  * ensure例子:
111  * 必须配合rxjava,回调结果与request一样
112  */
113 public void testEnsure(View view) {
114     RxPermissions rxPermissions = new RxPermissions(this);
115     Observable.timer(10, TimeUnit.MILLISECONDS)
116         .compose(rxPermissions.ensure(permissionsGroup))
117         .observeOn(AndroidSchedulers.mainThread())
118         .subscribe(new Consumer<Boolean>() {
119             @Override
120             public void accept(Boolean aBoolean) throws Exception {
121                 Log.i("cxw", "申请结果:" + aBoolean);
122             }
123         });
124 }
125
126 /**
127  * ensureEach例子:
128  * 必须配合rxjava,回调结果跟requestEach一样

```

```

129 */
130 public void testEnsureEach(View view) {
131     RxPermissions rxPermissions = new RxPermissions(this);
132     //
133     Observable.timer(10, TimeUnit.MILLISECONDS)
134         .compose(rxPermissions.ensureEach(permissionsGroup))
135         .observeOn(AndroidSchedulers.mainThread())
136         .subscribe(new Consumer<Permission>() {
137             @Override
138             public void accept(Permission permission) throws Exception {
139                 Log.i("cxw", "权限名称:" + permission.name + ",申请结果:" +
140                     );
141             });
142     }
143
144 /**
145  * ensureEachCombined例子:
146  * 必须配合rxjava,回调结果跟requestEachCombined一样
147  */
148 public void testEnsureEachCombined(View view) {
149     RxPermissions rxPermissions = new RxPermissions(this);
150     Observable.timer(10, TimeUnit.MILLISECONDS)
151         .compose(rxPermissions.ensureEachCombined(permissionsGroup))
152         .observeOn(AndroidSchedulers.mainThread())
153         .subscribe(new Consumer<Permission>() {
154             @Override
155             public void accept(Permission permission) throws Exception {
156                 Log.i("cxw", "权限名称:" + permission.name + ",申请结果:" +
157                     );
158             });
159     }
160 }

```

个人demo

<https://github.com/kingsley1212/RxPermissions>

**github地址:**

<https://github.com/tbruyelle/RxPermissions>

感谢 作者: 崔兴旺

链接: <https://www.jianshu.com/p/9c98348c67be>

**使用RxPermissions** ( 基于RxJava2 )

[Abracadabra的专栏](#) 4万+

**使用RxPermissions** ( 基于RxJava2 ) 0. 背景Android 6.0 ( API level 23 ) 中, 将权限分成了两类。一类是Install...

**RxPermissions** 权限的**使用**(非常精简,简单实用)

[wolfking0608的博客](#) 5321

一直以来,Android 6.0以上动态权限申请是一件挺麻烦的事情,网上也有各种方法,和第三方框架提供,但是用起来,有...



优质评论可以帮助作者获得更高权重

抢沙发

评论

**Android6.0运行时权限解析,RxPermissions的使用**, 自己封装一套权限... [weixin\\_33877092的博客](#) 432

**Android6.0运行时权限解析,RxPermissions的使用**, 自己封装一套权限框架 在Android6.0中, 新增加了一个运行...

**RxPermissions** 动态权限管理

[Theprogramleaders的博客](#) 384

1、介绍: This library allows the usage of RxJava with the new Android M permission model. 即: 这个库支持R...

**Android动态权限使用RxPermissions实现**(基于RxJava2...\_CSDN博客

9-1

假设执行在Android 6.0的环境中,该app在执行时必须主动申请这些dangerous权限(调用requestPermissions()). ...

**Android使用RxPermissions动态申请权限\_济沧海 x 远沧溟-CSDN博客**

10-7

**Android使用RxPermissions动态申请权限一、目标1. 效果图2. 下载地址二、功能设计三、准备工作四、组合起来...**

**winfrom中嵌套html, 跟html的交互**

08-20

winfrom中嵌套html, 跟html的交互,源码就在里面一看就懂,很简单

**Android RxPermissions 使用教程**

[anymyna的专栏](#) 216

github: <https://github.com/tbruyelle/RxPermissions> demo: [https://github.com/anymyna/android\\_laboratory](https://github.com/anymyna/android_laboratory) 步骤...

**android6.0新权限申请以及RxPermissions库的使用\_懒猪之旅-CSDN博客**

9-11



RxPermission: 用于适配Android 6.0新的权限模型的开源框架。 github地址:https://github.com/tbruyelle/RxPermis...

Android6.0运行时权限解析,RxPermissions的使用,自己封...\_CSDN博客

Android6.0运行时权限解析,RxPermissions的使用,自己封装一套权限框架在Android6.0中,新增加了一个运行时的...

10-11

RxJava2 中多种取消订阅 dispose 的方法梳理( 源码分析 )

Github 相关代码: Github地址 一直感觉 RxJava2 的取消订阅有点混乱, 这样也能取消, 那样也能取消, 没能系统起...

嘎啦嘎啦 1万+

Android最火的框架系列 ( 十一 ) RxPermissions

做Android开发的同学应该都知道, Android6.0以后, 权限系统有了很大的变化。危险权限, 单纯在AndroidManife...

zhaojun的博客 1万+

...6.0运行时权限第三方库的使用---RxPermissions\_星光...\_CSDN博客

由于Rxjava的巨大优势和RxPermissions的优秀封装使用起来非常方便 RxPermissions rxPermissions =newRxPer...

9-26

android 权限动态申请 RxPermissions使用\_大粪球-CSDN博客

// 如果是Android 6.0 (Api 23)之前,则权限被允许使用。 return lisMarshmallow() || mRxPermissionsFragment.isGr...

8-29

android开发--RxPermissions的简单使用

1.RxPermissions的好处 RxPermissions是帮助开发者简化requestPermissions()相关的处理。 ( 1 ) 开发者不用...

zztt113的博客 245

RxPermissions使用

RxPermissions使用 基于RxJava2, 使用包名为com.tbruyelle.rxpermissions2的RxPermissions。 5.1 App module...

ZJQ的博客 139

Android6.0动态权限申请及RxPermissions权限库使用\_csd...\_CSDN博客

(2)危险权限,共分为9组需要动态申请,每组只要有一个权限申请成功,默认整组权限都可以使用了。(3)普通权限,只...

9-11

Android rxpermissions二次封装,打造高大上的权限请求

.observeOn(AndroidSchedulers.mainThread()) .subscribe(permission1 -> { }); 二、 RxPermissions二次封装。 兄...

8-4

©2020 CSDN 皮肤主题: 大白 设计师:CSDN官方博客 返回首页

关于我们 招聘 广告服务 网站地图 kefu@csdn.net 客服论坛 400-660-0108 QQ客服 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2020北京创新乐知网络技术有限公司 版权与免责声明 版权申诉

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

PDFCROWD