# Android自定义URL使用Scheme方式唤起Activity或App

2016年05月13日 20:12:58

* 标签：
* [android](http://so.csdn.net/so/search/s.do?q=android&t=blog) /
* [app](http://so.csdn.net/so/search/s.do?q=app&t=blog) /
* [scheme](http://so.csdn.net/so/search/s.do?q=scheme&t=blog) /
* [url](http://so.csdn.net/so/search/s.do?q=url&t=blog)
* 23492

最近由于工作需要，使用Scheme方式自定义URL来跨应用间调用，踩了一些坑，现在记录一下。

1、Scheme方式配置

manifest中配置能接受Scheme方式启动的activity

**[html]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. **<intent-filter>**
2. **<action** android:name="android.intent.action.VIEW"**></action>**
3. **<category** android:name="android.intent.category.DEFAULT"**></category>**
4. **<category** android:name="android.intent.category.BROWSABLE"**></category>**
5. **<data**
6. android:scheme="app"
7. android:host="test"**>**
8. **</data>**
9. **</intent-filter>**

如果需要配置能被js调起，一定要配置下面这句

**[html]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. **<category** android:name="android.intent.category.BROWSABLE"**></category>**

data配置的属性：

**[html]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. android:scheme="app"表示配置接受的协议为app，相当于http://www.baidu.com中的http或者https

**[html]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. android:host="test"表示配置接受的域名为test，相当于http://www.baidu.com中的www.baidu.com

2、使用URL调起Activity

**[java]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. Uri uri=Uri.parse("app://test");
2. Intent intent=**new** Intent(Intent.ACTION\_VIEW,uri);
3. startActivity(intent);

3、使用js调起Activity

**[javascript]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. window.location = "app://test";

4、实际使用中的一些小细节

当自定义的URL配置在LAUNCHER对应的Activity上时，上述配置就足够了。

但是当自定义的URL配置在非LAUNCHER对应的Activity时，还需要增加额外几步操作。

问题一：使用自定义的URL启动Activity时，默认是已FLAG\_ACTIVITY\_NEW\_TASK的方式启动的，所以可能存在URL启动的Activity跟应用已启动的Activity不再同一个堆栈的现象。

解决方式：这种情况下，需要在manifest中将Activity多配置一个taskAffinity属性，约束URL启动的Activity与应用自身的启动的Activity在同一个堆栈中。

问题二：应用A使用url的方式唤起应用B的Activity时，可能存在应用B的Activity启动了，但是堆栈仍然在后台的现象，即应用B的Activity没有聚焦的问题。

解决方式：这种情况下，应用B的Activity收到启动的请求后，可以主动将Activity对应的堆栈移动到最前端。

**[java]** [view plain](https://blog.csdn.net/xc765926174/article/details/51397847) [copy](https://blog.csdn.net/xc765926174/article/details/51397847)

1. ActivityManager activityManager = (ActivityManager) getSystemService(Context.ACTIVITY\_SERVICE);
2. activityManager.moveTaskToFront(getTaskId(), ActivityManager.MOVE\_TASK\_WITH\_HOME);

使用这种方式需要注意的是该api是在Android 3.0（api 11）以后才提供的，现在基本上手机rom版本都是Android4.4以上了，就不太需要关注3.0一下怎么处理了，且使用这个需要在manifest中申请android.permission.REORDER\_TASKS权限。