[关闭]

@flyouting 2014-06-24 14:21 字数 2096 阅读 562

使用Android:ssp高效过滤Android intents

android intent ssp

我发现在Android4.4中添加了一组还未文档化的XML属性，用于intent过滤的标签中。

android:ssp用于匹配uri属性-"ssp"代表"scheme-specific part"，意思代表东西都在schemme之后出现，例如URI "https://example.com/foo/bar"可以分成scheme部分"https"和scheme-specific part部分"//example.com/foo/bar"。

虽然这不是HTTP地址的常规用法，但是正如android:host和android:path*的存在使得更容易创建intent过滤器一样，ssp使得监控特定类型事件变的更有效。

问题

Android中的广播接收器机制是一个不错的方式来确保我们的app-无论程序进程是否已经运行-可以知晓系统事件，比如当前网络是否接通，电池电量低等等。

然而，当多个程序订阅同一个经常发生的事件，这会导致很多系统进程同时开启，这会导致系统变的很慢。

最常见的情况就是当安装、更新或删除一个Android包。

显然很多包含数据分析sdk的应用喜欢监视和报告系统中当前应用的安装或删除,那就会定义一个广播接收器如下:

```
<receiver android:name=".PackageReceiver">
  <intent-filter>
    <action android:name="android.intent.action.PACKAGE_ADDED" />
    <data android:scheme="package" />
  </intent-filter>
</receiver>
```

注意，我们其实只关注URI类似"package:com.example.someapp"的action，但是这不是一个具有层次性的URI，没有host，port，path，我们不能准确地知道我们希望得到哪些包的通知。因此每个包的安装移除事件都会使监听此事件的应用得到通知并激活。

进入android:ssp及相关

Android4.4允许我们通过android:ssp，android:sspPrefix和android:sspPattern属性来匹配URI的scheme-specific part。

如果是设定包，我们现在可以专门指定一个或多个我们感兴趣的包。例如，我做的一个app有三个不同的包ID用于development,beta,release开发阶段。

我们可以通过过滤器匹配三个app（不包含其他），如下:

```
<receiver android:name=".DataClearedReceiver">
  <intent-filter>
    <action android:name="android.intent.action.PACKAGE_DATA_CLEARED" />
    <data android:scheme="package"
          android:sspPrefix="com.myswitzerland.hotels" />
   </intent-filter>
</receiver>
```

所以，只有包含包URI，切URI以特定文本打头的intent才会导致我妈妈的app进程开启，接收器被触发。如果其他app被清理数据什么的，我们的进程不会被启动，爽吧！

当然，这也使用于其他的非层次性的URI，比如mailto， tel

这有个不太好的例子：我们可以拦截特定的电子邮件地址，让用户选择使用我们的activity，而不是直接使用默认的客户端。

```
<activity android:name=".PremiumSupportActivity">
  <intent-filter>
    <action android:name="android.intent.action.SENDTO" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:scheme="mailto"
          android:sspPattern="support-.*@example.com" />
   </intent-filter>
</activity>
```

ssp属性没有在<data>元素的文档中被提到，但是在对等的java实现的IntentFilter文档中有描述。

所以，如果有需要，我们也可以这样注册：

```
IntentFilter pkgFilter = new IntentFilter(Intent.ACTION_PACKAGE_REMOVED);
pkgFilter.addDataScheme("package");
pkgFilter.addDataSchemeSpecificPart("com.example.someapp",PatternMatcher.PATTERN_LITERAL);
```

例子在4.4系统上是正常的，在XML中也可以使用，因为在4.4之前的版本中会自动过滤掉不存在的ssp属性。

原文地址
翻译：flyouting

- 搜索 flyouting 的文稿标题， *
- 以下【标签】将用于标记这篇文稿：
  - Android

添加新批注

内容目录

9

保存 取消

在作者公开此批注前，只有你和作者可见。

保存 取消

修改 保存 取消 删除

- 私有
- 公开
- 删除

查看更早的 5 条回复

回复批注

×

通知

取消 确认

- ImageSpan 1
  - Span，一个强大的概念
- Span 1
  - Span，一个强大的概念
- UI 1
  - 在Android中创建卡片式UI
- android 3
  - Android应用程序终止的影响
  - 按照预期初始化布局--layout inflation as intended
  - 使用Android:ssp高效过滤Android intents
- card 1
  - 在Android中创建卡片式UI
- inflation 1
  - 按照预期初始化布局--layout inflation as intended
- intent 1
  - 使用Android:ssp高效过滤Android intents
- layout 1
  - 按照预期初始化布局--layout inflation as intended
- ssp 1
  - 使用Android:ssp高效过滤Android intents
- volley 5
  - Volley框架的增强二
  - Volley框架的增强一
  - Volley源码学习笔记三
  - Volley源码学习笔记二
  - Volley源码学习笔记一
- 后台 1
  - Android后台任务最佳实践
- 服务 1
  - Android后台任务最佳实践
- 线程 1
  - Android后台任务最佳实践
- 设计模式 1
  - Android 设计模式--建造者模式
- 未分类 11
  - 优化ListView
  - Android ORM框架之 ORMLite
  - Android ORM 框架之 ActiveAndroid
  - WebView探索
  - Android相机开发指南（五）
  - Android相机开发指南（四）
  - Android相机开发指南（三）
  - Android相机开发指南（二）
  - Android相机开发指南（一）
  - Android transitions 如何工作
  - Android transitions 如何工作

- 下载客户端
- 关注开发者
- 报告问题，建议
- 联系我们