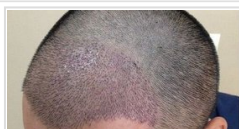
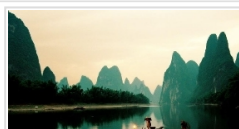




带牌出售二手车



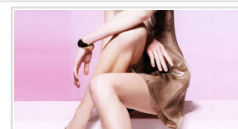
种头发原理



桂林必去景点



唇部怎么脱毛



冰点脱毛

## 揭密微信自动抢红包原理——AccessibilityService

MrLee 6月前

微信红包出来有几年了,其中自动抢红包的软件也是越来越多,那么他们是怎么实现的呢?

今天给大家揭密这个自动抢红包的原理!

在Android设备上呢?话说也偶然,Google为了让Android系统更实用,

为用户提供了无障碍辅助服务(AccessibilityService).

AccessibilityService运行在后台,并且能够收到由系统发出的一些事件(AccessibilityEvent, 这些事件表示用户界面一系列的状态变化),比如焦点改变,输入内容变化,按钮被点击了等等, 该种服务能够请求获取当前活动窗口并查找其中的内容.换言之,

界面中产生的任何变化都会产生一个时间,并由系统通知给AccessibilityService.

这就像监视器监视着界面的一举一动,一旦界面发生变化,立刻发出警报.

是不是感觉很棒?接下来,让我们来看看如何AccessibilityService的基本使用,在不同的阶段, 对其中的一些点做深入的说明,之后我们从实际应用出发,探讨其中的一些使用场景.

## 深入AccessibilityService使用

### 1. 创建服务类

编写自己的服务类,需要继承AccessibilityService类.

其中要实现onAccessibilityEvent(AccessibilityEvent event)及onInterrupt()两个重要的方法:

```
1 public class RobService extends AccessibilityService {  
2     @Override  
3     public void onAccessibilityEvent(AccessibilityEvent event) { }  
4     @Override  
5     public void onInterrupt() { }
```

这里我们对该类常用的方法做下说明,更详细的内容参见官方文档

### 方法

### 作用

disableSelf()

禁用当前服务,也就是在服务可以通过该方法停止运行

findFoucs(int falg)

查找拥有特定焦点类型的控件

getRootInActiveWindow()

如果配置能够获取窗口内容,  
则会返回当前活动窗口的根结点

getServiceInfo()

获取当前服务的配置信息

onAccessibilityEvent(AccessibilityEvent  
event)

有关AccessibilityEvent事件的回调函数.  
系统通过sendAccessibiliyEvent()  
不断的发送AccessibilityEvent到此处

performGlobalAction(int action)

执行全局操作,比如返回,回到主页,打开最近等操作

setServiceInfo(AccessibilityServiceInfo  
info)

设置当前服务的配置信息

getSystemService(String name)

获取系统服务

onKeyEvent(KeyEvent event)

如果允许服务监听按键操作,该方法是按键事件的回调,  
需要注意,这个过程发生了系统处理按键事件之前  
系统成功绑定该服务时被触发,

onServiceConnected()

也就是当你在设置中开启相应的服务,  
系统成功的绑定了该服务时会触发,  
通常我们可以在这里做一些初始化操作

## 2. 声明服务

像其他Service服务一样,需要在AndroidManifest.xml中声明该服务.除此之外,  
该服务还必须配置以下两项:



配置,其name为固定的: android.accessibilityservice.AccessibilityService

声明BIND\_ACCESSIBILITY\_SERVICE权限,以便系统能够绑定该服务(4.0版本后要求)

注意:任何一点配置错误,系统都无反应,因此其固定配置如下:

```
1 <service android:name=".RobService" android:enabled="true"
2   android:exported="true"
3   android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
4     <intent-filter>
5       <action
        android:name="android.accessibilityservice.AccessibilityService"/>
      </intent-filter>
    </service>
```

### 3. 服务参数设置

在AndroidManifest.xml声明了该服务之后,接下来就是需要对该服务进行一些参数设置了.

该服务能够被配置用来接受指定类型的事件,监听指定package,检索窗口内容,

获取事件类型的时间等等.目前有两种配置方法:

方法一:4.0之后提供了可以通过标签进行配置

方法二:通过setServiceInfo()进行配置

#### 3.1. 通过进行配置

在AndroidManifest.xml生命的的service中提供一个meta-data标签,

然后通过android:resource指定相应的配置文件(在res目录下创建xml文件,

并在其中创建配置文件accessibility.xml):

```
1 <service android:name=".RobService" android:enabled="true"
2   android:exported="true"
3   android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
4     <intent-filter>
5       <action
        android:name="android.accessibilityservice.AccessibilityService"/>
      </intent-filter>
6     <meta-data android:name="android.accessibilityservice"
      android:resource="@xml/accessibility"/>
    </service>
```



接下来我们来看accessibility.xml的相关配置:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"
  android:accessibilityEventTypes="typeNotificationStateChanged|typeWindowStateChanged|typeWindowContentChanged"
  android:accessibilityFeedbackType="feedbackGeneric"
  android:accessibilityFlags="flagDefault" android:canRetrieveWindowContent="true"
  android:notificationTimeout="100" android:packageNames="com.tencent.mm" />
```

后面,我们在只需要仿照该配置文件根据自己的需求进行修改即可.

下面我们会对这些属性进行介绍.

## 3.2 通过setServiceInfo(AccessibilityServiceInfo info)

也可以通过setServiceInfo(AccessibilityServiceInfo)为其配置信息,除此之外,

通过该方法可以在运行期间动态修改服务配置.需要注意,

该方法只能用来配置动态属性:eventTypes,feedbackType,flags,notificationTimeout及packageNames.

通常是在onServiceConnected()进行配置,如下代码:

```
1 @Override
2     protected void onServiceConnected() {
3         AccessibilityServiceInfo serviceInfo = new
4         AccessibilityServiceInfo();
5         serviceInfo.eventTypes = AccessibilityEvent.TYPES_ALL_MASK;
6         serviceInfo.feedbackType = AccessibilityServiceInfo.FEEDBACK_GENERIC;
7         serviceInfo.packageNames = new String[]{"com.tencent.mm"};
8         serviceInfo.notificationTimeout=100;
9         setServiceInfo(serviceInfo);
10    }
```

在这里涉及到了AccessibilityServiceInfo类,做个说明:

AccessibilityServiceInfo该类被用于配置AccessibilityService信息,

该类中包含了大量用于配置的常量字段及用来xml 属性,比如常见的:

android:accessibilityEventTypes,android:canRequestFilterKeyEvents,android:packageNames  
等等,更多信息参见官方文档

现在我们对配置中的重要属性进行说明: accessibilityEventTypes:

表示该服务对界面中的哪些变化感兴趣,即哪些事件通知,比如窗口打开,滑动,焦点变化,长按等.具体的值可以在AccessibilityEvent类中查到,如typeAllMask表示接受所有的事件通知.

accessibilityFeedbackType:表示反馈方式,比如是语音播放,还是震动

canRetrieveWindowContent:表示该服务能否访问活动窗口中的内容.

也就是如果你希望在服务中获取窗体内容的化,则需要设置其值为true. notificationTimeout:

接受事件的时间间隔,通常将其设置为100即可. packageNames:

表示对该服务是用来监听哪个包的产生的事件

## 4. 启动服务

当我们做完以上操作,便可将app安装到手机.安装成功后,在设置->

辅助功能中便可以找到我们的服务.该服务默认处在关闭状态,需要手动开启.

## 5. 获取事件信息

上面我们说道,onAccessibilityEvent(AccessibilityEvent event)是该服务的核心方法,其中参数event封装来自界面相关事件的信息,比如我们可以获得该事件的事件类型,进而根据起类型选择不同的处理方式:

```
1  <code class="cs"> public void onAccessibilityEvent(AccessibilityEvent
2  event) {
3      int eventType = event.getEventType();
4      switch (eventType) {
5          case AccessibilityEvent.TYPE_VIEW_CLICKED:
6              //界面点击
7              break;
8          case AccessibilityEvent.TYPE_VIEW_TEXT_CHANGED:
9              //界面文字改动
10             break;
11     }
```

这里我们对AccessibilityEvent进行简单的说明: 当用户发生发生变化时,系统会发送一系列的AccessibilityEvent事件,比如按钮被点击时会发送TYPE\_VIEW\_CLICKED类型的事件.

方法	说明
getEventType()	事件类型

方法	说明
getSource()	获取事件源对应的结点信息
getClassName()	获取事件源对应类的类型,比如点击事件是有某个Button产生的,那么此时获取的就是Button的完整类名
getText()	获取事件源的文本信息,比如事件是有TextView发出的,此时获取的就是TextView的text属性.如果该事件源是树结构,那么此时获取的是这个树上所有具有text属性的值的集合
isEnabled()	事件源(对应的界面控件)是否处在可用状态
getItemCount()	如果事件源是树结构,将返回该树根节点下子节点的数量

系统不断的产生各种事件,有些是界面控件产生的,有些是系统产生的.

对于由界面控件的产生的事件,通常我们将该控件称之为事件源.

并不是所有的事件都能通过getSource()方法获取到事件源,比如像通知消息类型的事件(TYPE\_NOTIFICATION\_STATE\_CHANGED).

## 6. 获取窗口内容

仅仅知道事件的信息是不够的,我们还希望通过事件来获取发出该事件(事件源)的信息,比如Button按钮被点击时它的text.一个服务可以配置为允许服务检索窗口内容,即获取窗口内容.整个窗口内容本质上是关于AccessibilityWindowInfo和AccessibilityNodeInfo的树结构,我称之为内容树.(类似View Tree,但由不完全相同)

需要注意,该服务可能配置了只检测了部分事件,而不是全部事件,这就意味着,当内容树发生变化后,该服务可能并不知道,即该服务无法及时的了解当前的内容树是否发生了变化.比如说,你的服务只检测了点击事件,但是此时界面的输入焦点已经变化,这样整个结点树也发生了变化,但是你的服务却不知道,此时你在结点中拿到的窗口内容可能已经不是最新的了.因此,如果你想及时的获知当前窗口的内容,那么就在配置的时候,设置监听全部事件.

正如上面所提到的,要想获取窗口内容,在配置AccessibilityService时需设置canRetrieveWindowContent为true.之后,



便可以通过一下方法获取窗口内容:AccessibilityEvent.getSource(),findFocus(int),getWindow()或者getRootInActiveWindow()

## 7. 服务的生命周期

要理解该中服务的生命周期只需要记住一下三点即可:

该种服务完全由系统管理,并遵循已有的服务周期.

开启一个服务只能由用户在设置中打开,

而关闭则只能由用户在设置中关闭或者服务本身通过disableSelf()方法关闭(当然,

现在有些第三放软件也可以强制关闭该类型服务)

系统绑定该服务之后,会调用onServiceConnected()方法,这个方法可以被重写,在其中,你可以做一些初始化的操作.

## 8. 检测服务是否开启

介绍了一些AccessibilityService的基础知识之后,再补充一点关于检测某个服务是否开启的知识.

通常来说大体有以下两种方法来检测服务是否启用:

方法一:借助服务管理器AccessibilityManager来判断,但是该方法不能检测app本身开启的服务.

```
private boolean enabled(String name) {
    AccessibilityManager am = (AccessibilityManager) getSystemService(Context.ACCESSIBILITY_SERVICE);
    List serviceInfos = am.getEnabledAccessibilityServiceList(AccessibilityServiceInfo.FILTER_ENABLED);
    List installedAccessibilityServiceList = am.getInstalledAccessibilityServiceList();
    for (AccessibilityServiceInfo info : installedAccessibilityServiceList) {
        Log.d("MainActivity", "all -->" + info.getId());
        if (name.equals(info.getId())) {
            return true;
        }
    }
    return false;
}
```

既然谈到了AccessibilityManager,那么在这里我们就做个简单的介绍:

AccessibilityManager是系统级别的服务,用来管理AccessibilityService服务,比如分发事件,重



说明 获取服务列表(api 14之后废弃,用下面的方法代替) 获取已安装到系统的服务列表  
获取已启用的服务列表 判断服务是否启用 发送事件

## 方法

getAccessibilityServiceList()

getInstalledAccessibilityServiceList() setEnabledAccessibilityServiceList(int feedbackTypeFlags) isEnabled() sendAccessibilityEvent()

方法二:我们知道大部分的系统属性都在settings中进行设置,比如wifi,蓝牙状态等,而这些信息主要是存储在settings对应的的数据库中(system表和serure表),这就意味我们可以通过直接读取setting设置来判断相关服务是否开启:

```
private boolean checkStealFeature1(String service) {  
    int ok = 0;  
    try {  
        ok = Settings.Secure.getInt(getApplicationContext().getContentResolver(), Se  
    } catch (Settings.SettingNotFoundException e) {  
    }  
    TextUtils.SimpleStringSplitter ms = new TextUtils.SimpleStringSplitter(':');  
    if (ok == 1) {  
        String settingValue = Settings.Secure.getString(getApplicationContext().getC  
        if (settingValue != null) {  
            ms.setString(settingValue);  
            while (ms.hasNext()) {  
                String accessibilityService = ms.next();  
                if (accessibilityService.equalsIgnoreCase(service)) {  
                    return true;  
                }  
            }  
        }  
    }  
}
```



# 实际应用

到现在有关AccessibilityService的一些知识,我们已经讲完,下面我们就看它的具体使用,其中典型的应用就是抢红包插件.

## 1. 抢红包插件

先回顾一下抢红包的流程:

- 状态栏出现"[微信红包]"的消息提示,点击进入聊天界面
- 点击相应的红包信息,弹出抢红包界面
- 在抢红包界面点击"开",打开红包
- 在红包详情页面,查看详情,点击返回按钮返回微信聊天界面.

以上是不在微信聊天界面时的流程.如果你所在的微信聊天窗口出现红包,则不会执行步骤1,而是直接执行2,3,4.如果是在微信好友列表时,收到红包,则会在列表项中显示[微信红包],需要点击该列表项,进入聊天界面,随后执行2,3,4.为了方便演示,这里我们暂时不考虑好友列表时出现红包的情况.明白了抢红包流程,之后我们通过AccessibilityService获取通知栏信息及微信聊天窗口界面,继而通过模拟点击实现打开红包,抢红包等操作. AccessibilityService配置如下:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <accessibility-service xmlns:android="http://schemas.android.com/apk/res/android"
  android:accessibilityEventTypes="typeNotificationStateChanged|typeWindowStateChange|typeWindowContentChanged" android:accessibilityFeedbackType="feedbackGeneric"
  android:accessibilityFlags="flagDefault" android:canRetrieveWindowContent="true"
  android:notificationTimeout="100" android:packageNames="com.tencent.mm" />
```

具体实现代码如下:

```
1 <code class="cs">public class RobService extends AccessibilityService {
2
3     @Override
4     public void onAccessibilityEvent(AccessibilityEvent event) {
5         int eventType = event.getEventType();
6         switch (eventType) {
```

```

6         case AccessibilityEvent.TYPE_NOTIFICATION_STATE_CHANGED:
7             handleNotification(event);
8             break;
9         case AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED:
10        case AccessibilityEvent.TYPE_WINDOW_CONTENT_CHANGED:
11            String className = event.getClassName().toString();
12            if (className.equals("com.tencent.mm.ui.LauncherUI")) {
13                getPacket();
14            } else if
15            (className.equals("com.tencent.mm.plugin.luckymoney.ui.LuckyMoneyReceiveUI"))
16            {
17                openPacket();
18            } else if
19            (className.equals("com.tencent.mm.plugin.luckymoney.ui.LuckyMoneyDetailUI"))
20            {
21                close();
22            }
23            break;
24        }
25    }
26    /**
27     * 处理通知栏信息
28     *
29     * 如果是微信红包的提示信息,则模拟点击
30     *
31     * @param event
32     */
33    private void handleNotification(AccessibilityEvent event) {
34        List<CharSequence> texts = event.getText();
35        if (!texts.isEmpty()) {
36            for (CharSequence text : texts) {
37                String content = text.toString();
38                //如果微信红包的提示信息,则模拟点击进入相应的聊天窗口
39                if (content.contains("[微信红包]")) {
40                    if (event.getParcelableData() != null &&
41                        event.getParcelableData() instanceof Notification) {
42                        Notification notification = (Notification)
43                        event.getParcelableData();
44                        PendingIntent pendingIntent =
45                        notification.contentIntent;
46                        try {
47                            pendingIntent.send();
48                        } catch (PendingIntent.CanceledException e) {
49                            e.printStackTrace();
50                        }
51                    }
52                }
53            }
54        }
55    }
56    /**

```

```

57     * 关闭红包详情界面, 实现自动返回聊天窗口
58     */
59     @TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
60     private void close() {
61         AccessibilityNodeInfo nodeInfo = getRootInActiveWindow();
62         if (nodeInfo != null) {
63             //为了演示, 直接查看了关闭按钮的id
64             List<AccessibilityNodeInfo> infos =
65 nodeInfo.findAccessibilityNodeInfosById("@id/ez");
66             nodeInfo.recycle();
67             for (AccessibilityNodeInfo item : infos) {
68                 item.performAction(AccessibilityNodeInfo.ACTION_CLICK);
69             }
70         }
71     }
72     /**
73     * 模拟点击, 拆开红包
74     */
75     @TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
76     private void openPacket() {
77         AccessibilityNodeInfo nodeInfo = getRootInActiveWindow();
78         if (nodeInfo != null) {
79             //为了演示, 直接查看了红包控件的id
80             List<AccessibilityNodeInfo> list =
81 nodeInfo.findAccessibilityNodeInfosById("@id/b9m");
82             nodeInfo.recycle();
83             for (AccessibilityNodeInfo item : list) {
84                 item.performAction(AccessibilityNodeInfo.ACTION_CLICK);
85             }
86         }
87     }
88     /**
89     * 模拟点击, 打开抢红包界面
90     */
91     @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
92     private void getPacket() {
93         AccessibilityNodeInfo rootNode = getRootInActiveWindow();
94         AccessibilityNodeInfo node = recycle(rootNode);
95         node.performAction(AccessibilityNodeInfo.ACTION_CLICK);
96         AccessibilityNodeInfo parent = node.getParent();
97         while (parent != null) {
98             if (parent.isClickable()) {
99                 parent.performAction(AccessibilityNodeInfo.ACTION_CLICK);
100                 break;
101             }
102             parent = parent.getParent();
103         }
104     }
105     /**
106     * 递归查找当前聊天窗口中的红包信息
107     */

```

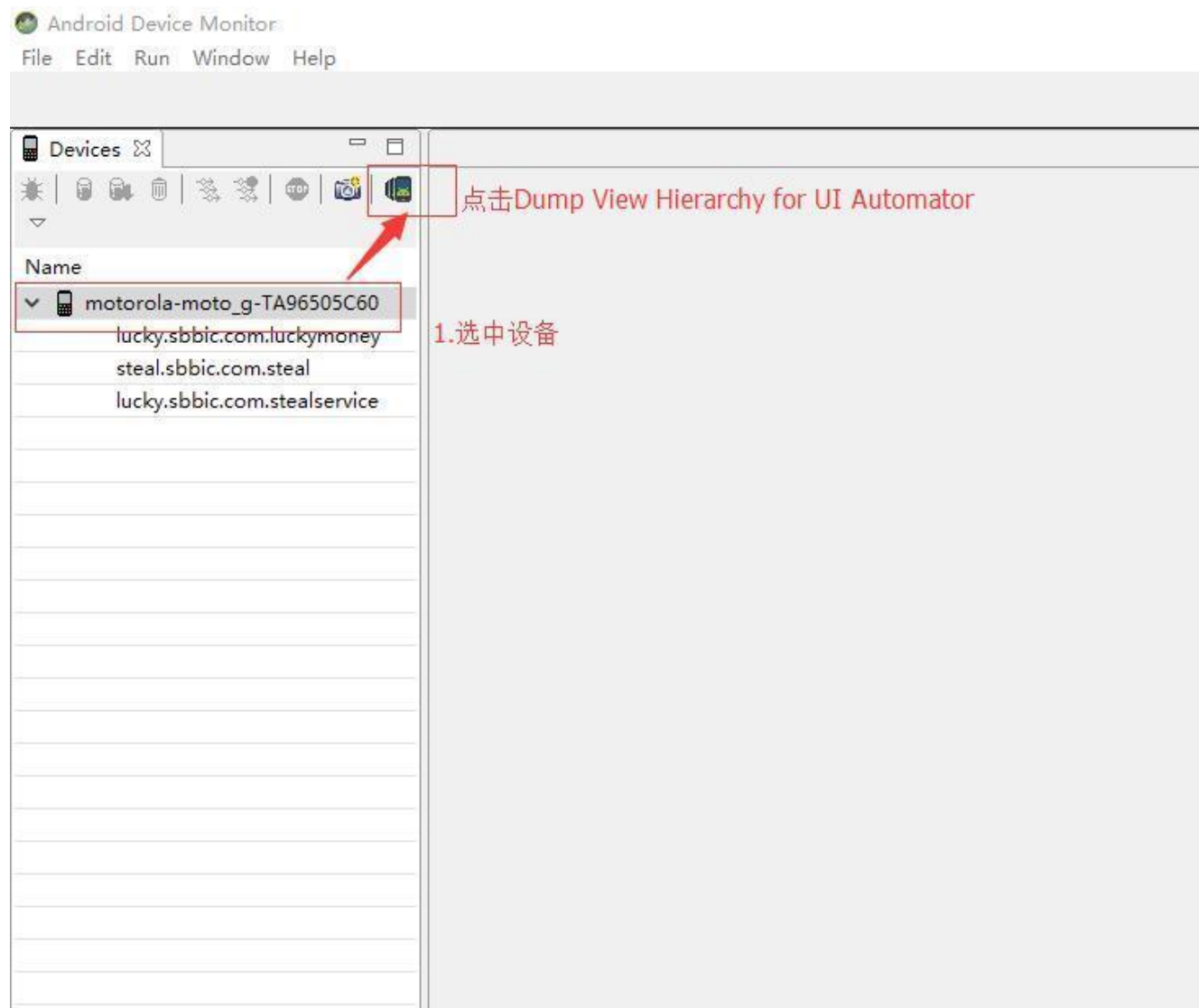


```
108     * 聊天窗口中的红包都存在"领取红包"一词, 因此可根据该词查找红包
109     *
110     * @param node
111     */
112     public AccessibilityNodeInfo recycle(AccessibilityNodeInfo node) {
113         if (node.getChildCount() == 0) {
114             if (node.getText() != null) {
115                 if ("领取红包".equals(node.getText().toString())) {
116                     return node;
117                 }
118             }
119         } else {
120             for (int i = 0; i < node.getChildCount(); i++) {
121                 if (node.getChild(i) != null) {
122                     recycle(node.getChild(i));
123                 }
124             }
125         }
126         return node;
127     }
128     @Override
129     public void onInterrupt() {
130     }
131     @Override
132     protected void onServiceConnected() {
133         super.onServiceConnected();
134     }
135 }
```

上面的代码简单演示了抢红包的原理,为了方便起见,我直接通过findAccessibilityNodeInfosByViewId()获取制定id控件.在实际中,这种方法不太可靠,到目前为止,微信已经改过几次相关控件的id了.

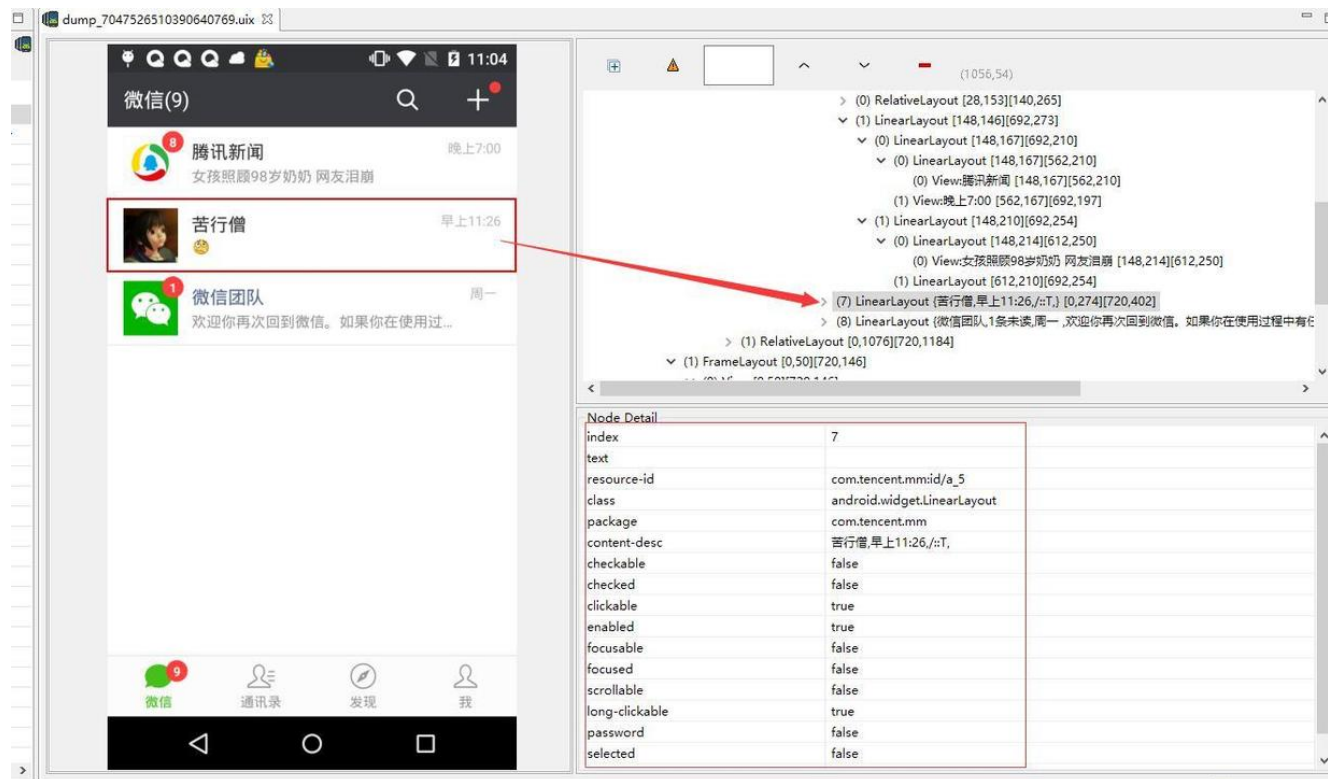
有童鞋问,怎么样知道该控件的id呢.其实很简单,android中已经为我们提供了相关的工具:在Android Studio中开启Android Device Monitor,选择设备后点击Dump View Hierarchy for UI Automator,如下:





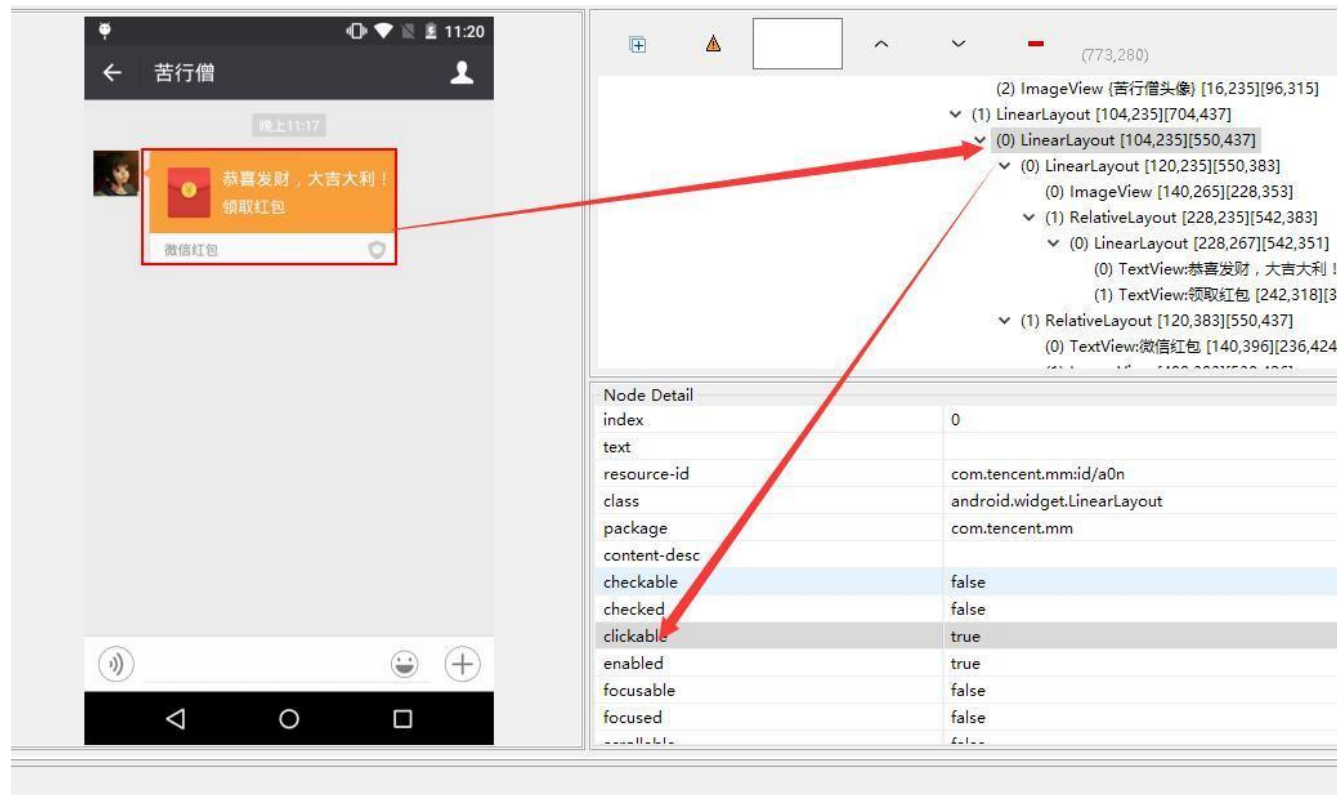
这里写图片描述

稍等片刻之后,便会出现当前设备的窗口,在该窗口中点击相关控件,便会显示该控件的属性.借助该工具,可以帮我们快速的分析界面结构,帮助我们其他app布局策略中学习



这里写图片描述

我们用Dump View Hierarchy for UI Automator分析聊天界面微信红包信息:

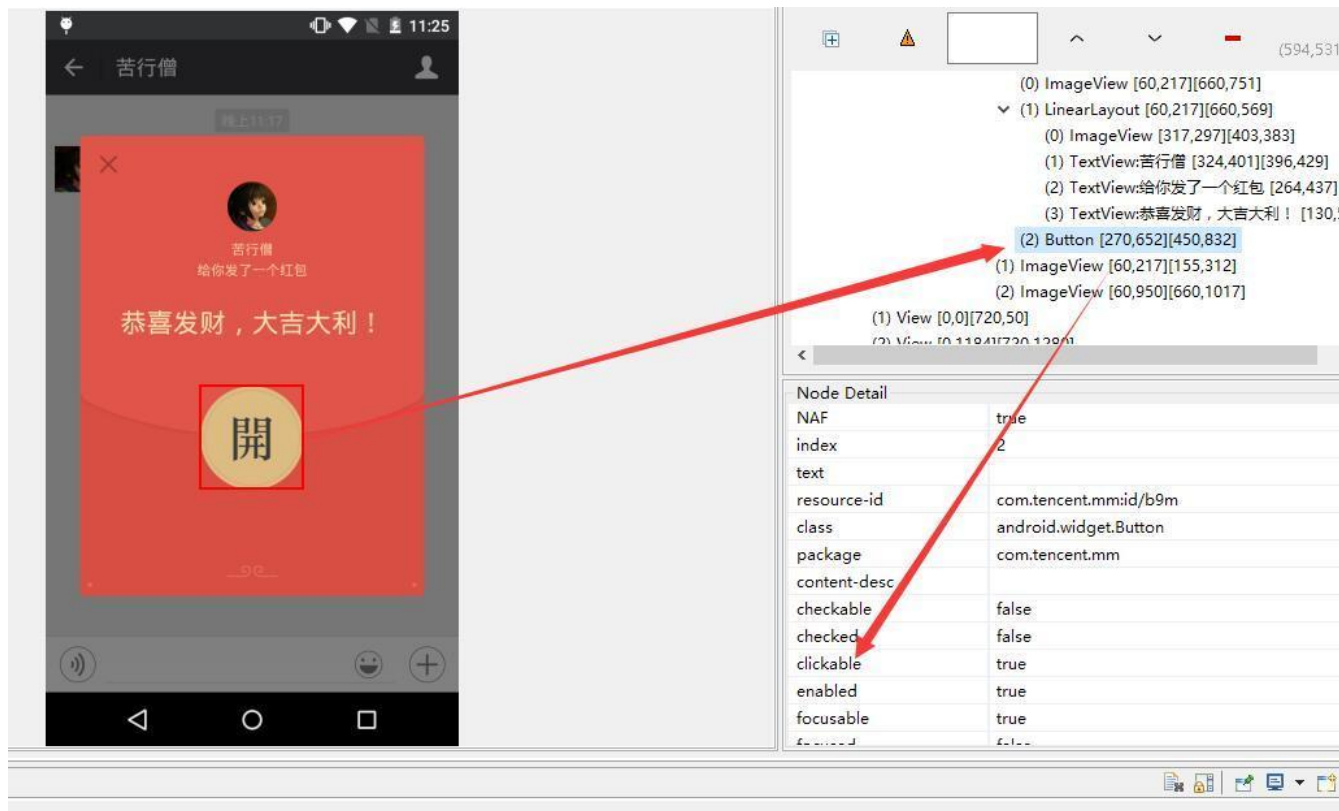


这里写图片描述

抢红包界面:







这里写图片描述

## 2. App自动安装

讲完了微信红包插件的实现原理,不难发现其本质是根据相关的界面状态,模拟后续的操作(比如点击等). 既然这样,那么我们完全可以利用该服务实现更多的功能,比如apk自动安装,传统的安装过程大概是如下流程:

点击apk文件,弹出安装信息界面,在该界面点击"下一步",然后在点击"安装",最后在安装完成界面点击"完成".

不难发现,该流程完全可以通过模拟点击操作完成.

现在我们简单的讲一下AccessibilityService在这方面的具体应用:

我们知道系统的安装程序由PackageInstaller负责,其包名是com.android.packageinstaller,那么我们只需要监听该package下的安装信息界面和安装完成界面,并模拟点击"下一步","安装",



完成""操作即可实现自动安装.

AccessibilityService配置如下:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <accessibility-service
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:accessibilityEventTypes="typeAllMask"
  android:accessibilityFeedbackType="feedbackGeneric"
  android:accessibilityFlags="flagDefault"
  android:canRetrieveWindowContent="true"
  android:description="@string/auto_service_des"
  android:notificationTimeout="100"
  android:packageNames="com.android.packageinstaller"/>
```

具体实现代码如下:

```
1 <code class="cs">public class InstallService extends AccessibilityService {
2     @Override
3     public void onAccessibilityEvent(AccessibilityEvent event) {
4         Log.d("InstallService", event.toString());
5         checkInstall(event);
6     }
7
8     private void checkInstall(AccessibilityEvent event) {
9         AccessibilityNodeInfo source = event.getSource();
10        if (source != null) {
11            boolean installPage =
12            event.getPackageName().equals("com.android.packageinstaller");
13            if (installPage) {
14                installAPK(event);
15            }
16        }
17    }
18    @TargetApi(Build.VERSION_CODES.JELLY_BEAN)
19    private void installAPK(AccessibilityEvent event) {
20        AccessibilityNodeInfo source = getRootInActiveWindow();
21        List<AccessibilityNodeInfo> nextInfos =
22        source.findAccessibilityNodeInfosByText("下一步");
23        nextClick(nextInfos);
24        List<AccessibilityNodeInfo> installInfos =
25        source.findAccessibilityNodeInfosByText("安装");
26        nextClick(installInfos);
27        List<AccessibilityNodeInfo> openInfos =
28        source.findAccessibilityNodeInfosByText("打开");
29        nextClick(openInfos);
30        runInBack(event);
31    }
32    private void runInBack(AccessibilityEvent event) {
```

```

32         event.getSource().performAction(AccessibilityService.GLOBAL_ACTION_BACK)
33     }
34     private void nextClick(List<AccessibilityNodeInfo> infos) {
35         if (infos != null)
36             for (AccessibilityNodeInfo info : infos) {
37                 if (info.isEnabled() && info.isClickable())
38                     info.performAction(AccessibilityNodeInfo.ACTION_CLICK);
39             }
40     }
41     @TargetApi(Build.VERSION_CODES.JELLY_BEAN_MR2)
42     private boolean checkTilt(AccessibilityNodeInfo source) {
43         List<AccessibilityNodeInfo> infos =
44             getRootInActiveWindow().findAccessibilityNodeInfosById("@id/app_name");
45         for (AccessibilityNodeInfo nodeInfo : infos) {
46             if (nodeInfo.getClassName().equals("android.widget.TextView")) {
47                 return true;
48             }
49         }
50         return false;
51     }
52     @Override
53     public void onInterrupt() {
54     }
55     @Override
56     protected void onServiceConnected() {
57         Log.d("InstallService", "auto install apk");
58     }
59 }

```

### 3. 检测前台服务:

在很多情况下,我们需要检测自己的app是不是处在前台,借助该服务同样也能够完成该检测操作.

下面,我们就演示一下如何实现:

AccessibilityService配置如下:

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <accessibility-service
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:accessibilityEventTypes="typeWindowStateChanged"
    android:accessibilityFeedbackType="feedbackGeneric"
    android:accessibilityFlags="flagDefault"
    android:canRetrieveWindowContent="true"
    android:description="@string/auto_detection"
    android:notificationTimeout="100" />

```



具体实现代码如下:

```
1 public class DetectionService extends AccessibilityService {
2     private static volatile String foregroundPackageName = "error";
3     /**
4      * 检测是否是前台服务
5      *
6      * @param packageName
7      * @return
8      */
9     public static boolean isForeground(String packageName) {
10         return foregroundPackageName.equals(packageName);
11     }
12     @Override
13     public void onAccessibilityEvent(AccessibilityEvent event) {
14         if (event.getEventType() ==
15             AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED) {
16             foregroundPackageName = event.getPackageName().toString();
17         }
18     }
19     @Override
20     public void onInterrupt() {
21     }
```

在使用时,需要引导用户开启该服务,之后通过调用DetectionService.isForeground()即可.

## 4. 窃取信息

上面的所有示例演示的都是AccessibilityService在具体应用中发挥的良好作用.

但是该服务也存在一定的风险,很多人利用该服务做一些"坏事",比如窃取短信验证码,窃取短信内容,想要看看自己女朋友最近在和谁聊QQ,偷偷安装流氓软件等.

上面我们了解微信抢红包插件的原理,那么利用该AccessibilityService编写相应的反抢红包插件:通过模拟微信红包的通知信息,发送虚假的事件通知.不出意外,我们编写的反抢红包插件会让失眠绝大多数的抢红包插件.这里我就不做深入的解释,有兴趣的同学可以自行研究.



你现在是不是想能否借助该服务直接获取一些app的密码呢？  
凡是EditText中设置inputType为password类型的,都无法获取其输入值.除此之外,  
大多数软件都针对该中风险做了提前的防范.因此,  
你想要借助该服务来实现窃取密码还是比较有难度的,所以,少年觉悟吧.

本文链接:<http://www.ithtw.com/thread-12022.htm>

转载请注明:6月前 于 IT十万个为什么 发表

&lt;

&gt;

## 推荐阅读

- › android解决bitmap内存溢出
- › 移动端的加解密
- › 饿了么开源的Android跨进程事件分发框架HermesEventBus
- › 一个字符解决Gradle aar编译参数传递问题是怎样一种体验
- › Android中Instrumentation使用简介(续)
- › Android之断点续传下载
- › Android Training - 性能优化小技巧
- › android 仿QQ搜索框自动弹出、收起输入法
- › Android状态栏微技巧，带你真正理解沉浸式模式
- › SwipeRefreshLayout简单玩法

## 最新回复 (0)

[⚡ 快速回复](#)

## 回复

[登录](#) 发表 or 还没有账号？去 [注册](#)

