

Android微信自动回复功能



MeloDev (/u/f5909165c1e8) [+ 关注](#)

2016.07.15 10:17* 字数 2457 阅读 8607 评论 99 喜欢 225 赞赏 1
(/u/f5909165c1e8)

Android微信自动回复功能

本文原创，转载请经过本人准许。

写在前面：

最近接到老大的一个需求，要求在手机端拦截微信的通知（Notification），从而获得联系人和内容。之后将联系人和内容发送到我们的硬件产品上，展示出来之后，再将我们想回复内容传给微信，并且发送给相应联系人。

老大还提示我需要用**AccessibilityService**去实现它，当然在此之前我并不知道**AccessibilityService**是什么鬼，不过没关系，**just do IT**！

AccessibilityService

AccessibilityService官方文档（需翻墙）

(<https://developer.android.com/reference/android/accessibilityservice/AccessibilityService.html>)

上面这个链接是AccessibilityService的官方文档，可以翻墙点进去了解下，我再给大家总结一下：



AccessibilityService是Android系统框架提供给安装在设备上应用的一个可选的导航反馈特性。AccessibilityService 可以替代应用与用户交流反馈，比如将文本转化为语音提示，或是用户的手指悬停在屏幕上一个较重要的区域时的触摸反馈等。

如果感觉上面的描述比较抽象，没关系，也许你见过下面这张图：



辅助功能中的服务

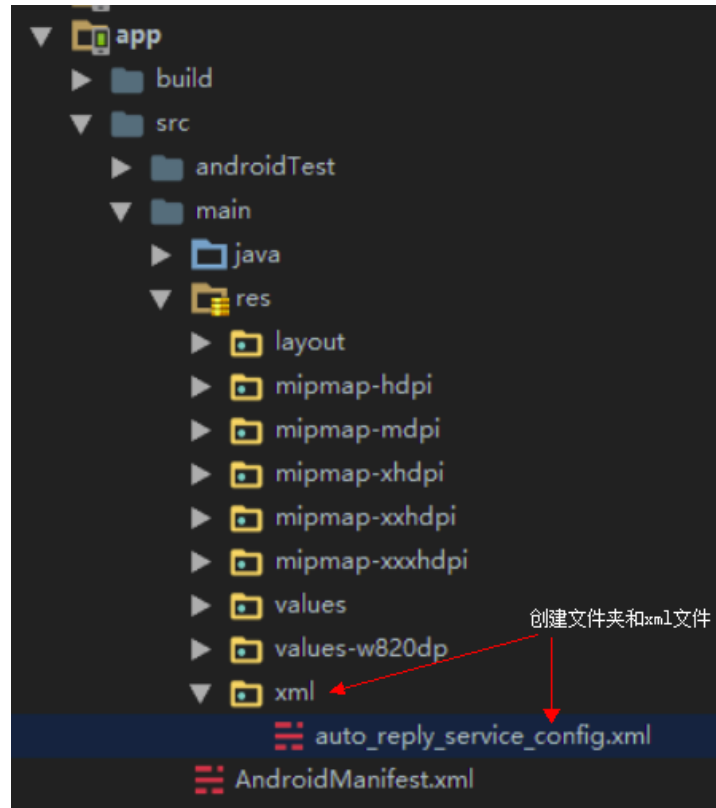
打开你手机的设置--辅助功能中，有很多APP提供的服务，他们都是基于AccessibilityService编写的，AccessibilityService可以侦听你的点击，长按，手势，通知栏的变化等。并且你可以通过很多种方式找到窗体中的EditText，Button等组件，去填充他们，去点击他们来帮你实现自动化的功能。

像360助手的自动安装功能，它就是侦听着系统安装的APP，然后找到“安装”按钮，实现了自动点击。微信自动抢红包功能，实现方式都是如此。



配置AccessibilityService

首先我们在res文件夹下创建xml文件夹，然后创建一个名为auto_reply_service_config的文件，一会我们会在清单文件中引用它。



AccessibilityService配置文件

代码：



```
<accessibility-service          xmlns:android="http://schemas.android.com/apk/r
    android:accessibilityFeedbackType="feedbackGeneric"
    android:accessibilityFlags="flagDefault"
    android:canRetrieveWindowContent="true"
    android:description="@string/accessibility_description"
    android:notificationTimeout="100"
    android:packageNames="com.tencent.mm" />
```

这个文件表示我们对AccessibilityService服务未来侦听的行为做了一些配置，比如 **typeNotificationStateChanged** 和 **typeWindowStateChanged** 表示我们需要侦听通知栏的状态变化和窗体状态改变。

android:packageNames="com.tencent.mm" 这是微信的包名，表示我们只关心微信这一个应用。

代码不打算带着大家一行一行看了，如果有不明白的，去看看文档，或者下面回复我，我给大家解答~

创建AccessibilityService

下面贴出AccessibilityService类的全部代码，注释还算详尽，如有疑问，下方回复。



```
package com.ileja.autoreply;

import android.accessibilityservice.AccessibilityService;
import android.annotation.SuppressLint;
import android.app.ActivityManager;
import android.app.KeyguardManager;
import android.app.Notification;
import android.app.PendingIntent;
import android.content.ClipData;
import android.content.ClipboardManager;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.os.PowerManager;
import android.text.TextUtils;
import android.view.KeyEvent;
import android.view.accessibility.AccessibilityEvent;
import android.view.accessibility.AccessibilityNodeInfo;

import java.io.IOException;
import java.util.List;

public class AutoReplyService extends AccessibilityService {
    private final static String MM_PNAME = "com.tencent.mm";
    boolean hasAction = false;
    boolean locked = false;
    boolean background = false;
    private String name;
    private String scontent;
    AccessibilityNodeInfo itemNodeinfo;
    private KeyguardManager.KeyguardLock kl;
    private Handler handler = new Handler();

    /**
     * 必须重写的方法，响应各种事件。
     * @param event
     */
    @Override
    public void onAccessibilityEvent(final AccessibilityEvent event) {
        int eventType = event.getEventType();
        android.util.Log.d("maptrix", "get event = " + eventType);
        switch (eventType) {
            case AccessibilityEvent.TYPE_NOTIFICATION_STATE_CHANGED:// 通知栏事件
                android.util.Log.d("maptrix", "get notification event");
                List<CharSequence> texts = event.getText();
                if (!texts.isEmpty()) {
```



```

for (CharSequence text : texts) {
    String content = text.toString();
    if (!TextUtils.isEmpty(content)) {
        if (isScreenLocked()) {
            locked = true;
            wakeAndUnlock();
            android.util.Log.d("maptrix", "the screen is locked");
            if (isAppForeground(MM_PNAME)) {
                background = false;
                android.util.Log.d("maptrix", "is mm in foreground");
                sendNotifacationReply(event);
                handler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        sendNotifacationReply(event);
                        if (fill()) {
                            send();
                        }
                    }
                }, 1000);
            } else {
                background = true;
                android.util.Log.d("maptrix", "is mm in background");
                sendNotifacationReply(event);
            }
        } else {
            locked = false;
            android.util.Log.d("maptrix", "the screen is unlocked");
            // 监听到微信红包的notification, 打开通知
            if (isAppForeground(MM_PNAME)) {
                background = false;
                android.util.Log.d("maptrix", "is mm in foreground");
                sendNotifacationReply(event);
                handler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        if (fill()) {
                            send();
                        }
                    }
                }, 1000);
            } else {
                background = true;
                android.util.Log.d("maptrix", "is mm in background");
                sendNotifacationReply(event);
            }
        }
    }
}
}
}
}
}

```



```

        break;
    case AccessibilityEvent.TYPE_WINDOW_STATE_CHANGED:
        android.util.Log.d("maptrix", "get type window down event");
        if (!hasAction) break;
        itemNodeinfo = null;
        String className = event.getClassName().toString();
        if (className.equals("com.tencent.mm.ui.LauncherUI")) {
            if (fill()) {
                send();
            } else {
                if (itemNodeinfo != null) {
                    itemNodeinfo.performAction(AccessibilityNodeInfo.ACTION_
                        handler.postDelayed(new Runnable() {
                            @Override
                            public void run() {
                                if (fill()) {
                                    send();
                                }
                                back2Home();
                                release();
                                hasAction = false;
                            }
                        }, 1000);
                    break;
                }
            }
        }

        //bring2Front();
        back2Home();
        release();
        hasAction = false;
        break;
    }
}

/**
 * 寻找窗体中的“发送”按钮，并且点击。
 */
@SuppressWarnings("NewApi")
private void send() {
    AccessibilityNodeInfo nodeInfo = getRootInActiveWindow();
    if (nodeInfo != null) {
        List<AccessibilityNodeInfo> list = nodeInfo
            .findAccessibilityNodeInfosByText("发送");
        if (list != null && list.size() > 0) {
            for (AccessibilityNodeInfo n : list) {
                if (n.getClassName().equals("android.widget.Button") && n.isEnabled()
                {
                    n.performAction(AccessibilityNodeInfo.ACTION_CLICK);
                }
            }
        }
    }
}

```



```

        }

        } else {
            List<AccessibilityNodeInfo> liste = nodeInfo
                .findAccessibilityNodeInfosByText("Send");
            if (liste != null && liste.size() > 0) {
                for (AccessibilityNodeInfo n : liste) {
                    if(n.getClassName().equals("android.widget.Button") && n.isE
                    {
                        n.performAction(AccessibilityNodeInfo.ACTION_CLICK);}
                    }
                }
            }
        }
        pressBackButton();
    }
}

/**
 * 模拟back按键
 */
private void pressBackButton(){
    Runtime runtime = Runtime.getRuntime();
    try {
        runtime.exec("input keyevent " + KeyEvent.KEYCODE_BACK);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/**
 *
 * @param event
 */
private void sendNotifacationReply(AccessibilityEvent event) {
    hasAction = true;
    if (event.getParcelableData() != null
        && event.getParcelableData() instanceof Notification) {
        Notification notification = (Notification) event
            .getParcelableData();
        String content = notification.tickerText.toString();
        String[] cc = content.split(":");
        name = cc[0].trim();
        scontent = cc[1].trim();

        android.util.Log.i("maptrix", "sender name =" + name);
        android.util.Log.i("maptrix", "sender content =" + scontent);
    }
}

```




```

        PendingIntent pendingIntent = notification.contentIntent;
        try {
            pendingIntent.send();
        } catch (PendingIntent.CanceledException e) {
            e.printStackTrace();
        }
    }

    @SuppressWarnings("NewApi")
    private boolean fill() {
        AccessibilityNodeInfo rootNode = getRootInActiveWindow();
        if (rootNode != null) {
            return findEditText(rootNode, "正在忙,稍后回复你");
        }
        return false;
    }

    private boolean findEditText(AccessibilityNodeInfo rootNode, String content) {
        int count = rootNode.getChildCount();

        android.util.Log.d("maptrix", "root class=" + rootNode.getClassName() + ", "+
        for (int i = 0; i < count; i++) {
            AccessibilityNodeInfo nodeInfo = rootNode.getChild(i);
            if (nodeInfo == null) {
                android.util.Log.d("maptrix", "nodeinfo = null");
                continue;
            }

            android.util.Log.d("maptrix", "class=" + nodeInfo.getClassName());
            android.util.Log.e("maptrix", "ds=" + nodeInfo.getContentDescription());
            if (nodeInfo.getContentDescription() != null) {
                int nindex = nodeInfo.getContentDescription().toString().indexOf("name=");
                int cindex = nodeInfo.getContentDescription().toString().indexOf("content=");
                android.util.Log.e("maptrix", "nindex=" + nindex + " cindex=" + cindex);
                if (nindex != -1) {
                    itemNodeInfo = nodeInfo;
                    android.util.Log.i("maptrix", "find node info");
                }
            }
        }
        if ("android.widget.EditText".equals(nodeInfo.getClassName())) {
            android.util.Log.i("maptrix", "=====");
            Bundle arguments = new Bundle();
            arguments.putInt(AccessibilityNodeInfo.ACTION_ARGUMENT_MOVEMENT_GRANULARITY_WORD,
                AccessibilityNodeInfo.MOVEMENT_GRANULARITY_WORD);
            arguments.putBoolean(AccessibilityNodeInfo.ACTION_ARGUMENT_EXTEND_SELECTION,
                true);
            nodeInfo.performAction(AccessibilityNodeInfo.ACTION_PREVIOUS_AT_MOVEMENT_GRANULARITY,
                arguments);
        }
    }

```



```

        nodeInfo.performAction(AccessibilityNodeInfo.ACTION_FOCUS);
        ClipData clip = ClipData.newPlainText("label", content);
        ClipboardManager clipboardManager = (ClipboardManager) getSystemService(Context.CLIPBOARD_SERVICE);
        clipboardManager.setPrimaryClip(clip);
        nodeInfo.performAction(AccessibilityNodeInfo.ACTION_PASTE);
        return true;
    }

    if (findEditText(nodeInfo, content)) {
        return true;
    }
}

return false;
}

@Override
public void onInterrupt() {

}

/**
 * 判断指定的应用是否在前台运行
 *
 * @param packageName
 * @return
 */
private boolean isAppForeground(String packageName) {
    ActivityManager am = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    ComponentName cn = am.getRunningTasks(1).get(0).topActivity;
    String currentPackageName = cn.getPackageName();
    if (!TextUtils.isEmpty(currentPackageName) && currentPackageName.equals(packageName)) {
        return true;
    }

    return false;
}

/**
 * 将当前应用运行到前台
 */
private void bringToFront() {
    ActivityManager activityManager = (ActivityManager) getSystemService(Context.ACTIVITY_SERVICE);
    List<ActivityManager.RunningTaskInfo> runningTaskInfos = activityManager.getRunningTasks(1);
    for (ActivityManager.RunningTaskInfo runningTaskInfo : runningTaskInfos) {
        if (this.getPackageName().equals(runningTaskInfo.topActivity.getPackageName())) {
            activityManager.moveTaskToFront(runningTaskInfo.id, ActivityManager.RunningTaskInfo.FLAG_TO_FRONT);
            return;
        }
    }
}

```



```
    }  
}  
  
/**  
 * 回到系统桌面  
 */  
private void back2Home() {  
    Intent home = new Intent(Intent.ACTION_MAIN);  
  
    home.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);  
    home.addCategory(Intent.CATEGORY_HOME);  
  
    startActivity(home);  
}  
  
/**  
 * 系统是否在锁屏状态  
 *  
 * @return  
 */  
private boolean isScreenLocked() {  
    KeyguardManager keyguardManager = (KeyguardManager) getSystemService(Context.  
        return keyguardManager.inKeyguardRestrictedInputMode();  
}  
  
private void wakeAndUnlock() {  
    //获取电源管理器对象  
    PowerManager pm = (PowerManager) getSystemService(Context.POWER_SERVICE);  
  
    //获取PowerManager.WakeLock对象，后面的参数|表示同时传入两个值，最后的是调试用的Tag  
    PowerManager.WakeLock wl = pm.newWakeLock(PowerManager.ACQUIRE_CAUSES_WAKEUP  
  
    //点亮屏幕  
    wl.acquire(1000);  
  
    //得到键盘锁管理器对象  
    KeyguardManager km = (KeyguardManager) getSystemService(Context.KEYGUARD_SER  
    kl = km.newKeyguardLock("unLock");  
  
    //解锁  
    kl.disableKeyguard();  
}  
  
private void release() {  
  
    if (locked && kl != null) {  
        android.util.Log.d("maptrix", "release the lock");  
        //得到键盘锁管理器对象
```



```
        kl.reenableKeyguard();  
        locked = false;  
    }  
}  
}
```

接着配置清单文件，权限和service的配置比较重要。



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.ileja.autoreply">

    <uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.BIND_ACCESSIBILITY_SERVICE" />
    <uses-permission android:name="android.permission.GET_TASKS" />
    <uses-permission android:name="android.permission.REORDER_TASKS" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <service
            android:name=".AutoReplyService"
            android:enabled="true"
            android:exported="true"
            android:label="@string/app_name"
            android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE">
            <intent-filter>
                <action android:name="android.accessibilityservice.AccessibilityServ
            </intent-filter>

            <meta-data
                android:name="android.accessibilityservice"
                android:resource="@xml/auto_reply_service_config"/>
        </service>
    </application>
</manifest>
```

为了使用某些必要的API，最低API level应该是18



运行程序，打开服务，看看效果如何把~



打开辅助服务

接着用其他手机试着发送给我几条微信





自动回复微信

可以看到，自动回复功能就实现了。

写在后面：

代码没有给大家详细讲解，不过看注释应该可以看懂个大概。当微信程序切换到后台，或者锁屏（无锁屏密码）时，只要有通知出现，都可以实现自动回复。

关于**AccessibilityService**可以监控的行为非常多，所以我觉得可以实现各种各样炫酷的功能，不过我并不建议你打开某些流氓软件的AccessibilityService服务，因为很有可能造成一些安全问题，所以，自己动手写就安全多了嘛。

github项目地址：

WcAutoReply (<https://github.com/itsMelo/WcAutoReply>)





MeloDev (/u/f5909165c1e8)

写了 106547 字，被 2358 人关注，获得了 3329 个喜欢

(/u/f5909165c1e8)

+ 关注

Github : <https://github.com/itsMelo> 欢迎 follow ~ itsCoder 主页 : <http://itscoder.com> 欢迎加入 ~ Email : mel...

❤ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 225



更多分享

(http://cwb.assets.jianshu.io/notes/images/4804184/weibo/image_1

被以下专题收入，发现更多相似内容



Android... (/c/b1fa46ec3b08?utm_source=desktop&utm_medium=notes-included-collection)



Android... (/c/5139d555c94d?utm_source=desktop&utm_medium=notes-included-collection)



@IT·互联网 (/c/V2CqjW?utm_source=desktop&utm_medium=notes-included-collection)



首页投稿 (/c/bDHhpK?utm_source=desktop&utm_medium=notes-included-collection)



Android (/c/4a21aecbb11d?utm_source=desktop&utm_medium=notes-included-collection)




安卓 (/c/21d1a252fbbe?utm_source=desktop&utm_medium=notes-included-collection)



android (/c/b0b5b0d5b6a7?utm_source=desktop&utm_medium=notes-



 included-collection)

展开更多 ∨

