

# 桌面小部件 App Widget

## 1 App Widget简介

- App Widget是应用程序窗口小部件（Widget）是微型的应用程序视图，
- 它可以被嵌入到其它应用程序中（比如桌面）并接收周期性的更新。
- 你可以通过一个App Widget Provider来发布一个Widget。
- 本文参考Android官方文本，先介绍App Widget的主要组件，然后再以示例来详细说明。

## 2 App Widget主要的相关类介绍

### 2.1 AppWidgetProvider

AppWidgetProvider 继承自 BroadcastReceiver，它能接收 widget 相关的广播，例如 widget 的更新、删除、开启和禁用等。

AppWidgetProvider中的广播处理函数如下：

onUpdate()

当 widget 更新时被执行。

同样，当用户首次添加 widget 时，onUpdate() 也会被调用，这样 widget 就能进行必要的设置工作。

但是，如果定义了 widget 的 configure属性(即Android:config，后面会介绍)

那么当用户首次添加 widget 时，onUpdate()不会被调用；之后更新 widget 时，onUpdate才会被调用。

onAppWidgetOptionsChanged()

当 widget 被初次添加或者当widget 的大小被改变时，执行onAppWidgetOptionsChanged()。

你可以在该函数中，根据 widget 的大小来显示/隐藏某些内容。

可以通过 getAppWidgetOptions() 来返回 Bundle对象以读取 widget 的大小信息，Bundle中包括以下信息：

OPTION\_APPWIDGET\_MIN\_WIDTH -- 包含 widget 当前宽度的下限，以dp为单位。

OPTION\_APPWIDGET\_MIN\_HEIGHT -- 包含 widget 当前高度的下限，以dp为单位。

OPTION\_APPWIDGET\_MAX\_WIDTH -- 包含 widget 当前宽度的上限，以dp为单位。

OPTION\_APPWIDGET\_MAX\_HEIGHT -- 包含 widget 当前高度的上限，以dp为单位。

onAppWidgetOptionsChanged() 是 Android 4.1 引入的。

onDeleted(Context, int[])

当 widget 被删除时被触发。

onEnabled(Context)

当第1个 widget 的实例被创建时触发。

也就是说，如果用户对同一个widget增加了两次（两个实例），那么onEnabled()只会在第一次增加widget时触发。

onDisabled(Context)

当最后1个 widget 的实例被删除时触发。

onReceive(Context, Intent)

接收到任意广播时触发，并且会在上述的方法之前被调用。

总结，AppWidgetProvider 继承于 BroadcastReceiver。实际上，App Widge中的onUpdate()、onEnabled()、onDisabled()等方法都是在 onReceive()中调用的；是onReceive()对特定事情的响应函数。

### 2.2AppWidgetProviderInfo

AppWidgetProviderInfo描述一个App Widget元数据，比如App Widget的布局，更新频率，以及AppWidgetProvider 类。这个应该在XML里定义。下面以XML示例来对AppWidgetProviderInfo中常用的类型进行说明。

示例XML：

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="40dp"//指定App Widget布局需要的最小区域。
    android:minHeight="40dp"//指定App Widget布局需要的最小区域。
```

```
android:updatePeriodMillis="86400000"//定义 widget 的更新频率
android:previewImage="@drawable/preview"//指定预览图
android:initialLayout="@layout/example_appwidget"// widget 的布局资源文件
android:configure="com.example.android.ExampleAppWidgetConfigure"//定义了widget的配置Activity
android:resizeMode="horizontal|vertical"//widget 的调整尺寸的规则
android:widgetCategory="home_screen|keyguard"// widget 能显示的地方
android:initialKeyguardLayout="@layout/example_keyguard">//widget位于lockscreen中的布局资源文件
</appwidget-provider>
```

## 3 App Widget应用实例

- 第1步 新建Android工程

新建空白的Android工程，即不需要在建立Activity子类。

- 第2步 编辑manifest

添加代码

```
<!-- 声明widget对应的AppWidgetProvider -->

<receiver android:name=".ExampleAppWidgetProvider" >
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
        <action android:name="com.skywang.widget.UPDATE_ALL"/>
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/example_appwidget_info" />
</receiver>
```

说明：

(01)ExampleAppWidgetProvider是继承于的AppWidgetProvider类，用来响应widget的添加、删除、更新等操作。

(02) android.appwidget.action.APPWIDGET\_UPDATE，必须要显示声明的action！因为所有的widget的广播都是通过它来发送的；要接收widget的添加、删除等广播，就必须包含它。

(03) action android:name="com.skywang.widget.UPDATE\_ALL，这是我自己添加了，是为了接收服务所发送的更新图片的广播。

(04) <meta-data>指定了 AppWidgetProviderInfo 对应的资源文件android:name -- 指定metadata名，通过android.appwidget.provider来辨别data。android:resource -- 指定 AppWidgetProviderInfo 对应的资源路径。即，xml/exampleappwidgetinfo.xml。

(05) ExampleAppWidgetService 是用于更新widget中的图片的服务。

(06) android.appwidget.action.EXAMPLEAPPWIDGET\_SERVICE 用于启动服务的action。

## - 第3步 编辑AppWidgetProviderInfo对应的资源文件

在当前工程下新建xml目录(若xml目录不存在的话)；并在xml目录下新建exampleappwidgetinfo.xml文件。xml文件内容如下：

复制代码

```
<appwidget-provider xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="180dp"
    android:minHeight="180dp"
    android:previewImage="@drawable/preview"
    android:initialLayout="@layout/example_appwidget"
    android:resizeMode="horizontal|vertical"
    android:widgetCategory="home_screen|keyguard">

<!--
    android:minWidth : //最小宽度
    android:minHeight : //最小高度
    android:updatePeriodMillis : //更新widget的时间间隔(ms)，"86400000"为1个小时
    android:previewImage : //预览图片
```

```
android:initialLayout : //加载到桌面时对应的布局文件
android:resizeMode : //widget可以被拉伸的方向。horizontal表示可以水平拉伸，vertical表示可以竖直拉伸
android:widgetCategory : //widget可以被显示的位置。home_screen表示可以将widget添加到桌面，
                        keyguard表示widget可以被添加到锁屏界面。
android:initialKeyguardLayout : //加载到锁屏界面时对应的布局文件
-->

</appwidget-provider>
```

- 第4步 编辑example\_appwidget.xml等资源文件

新建layout/example\_appwidget.xml

- 第5步 编辑ExampleAppWidgetProvider.java