

## Xây dựng các widget trong android

### 1. Khai báo widget trong file manifest

```
<!-- Broadcast Receiver that will process AppWidget updates -->
<receiver android:name=".WordWidget" android:label="android test">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
    </intent-filter>
    <meta-data android:name="android.appwidget.provider"
        android:resource="@xml/widget_world" />
</receiver>
```

### 2. Thiết lập cấu hình cho widget file **res/xml/widget\_world.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:minWidth="146dip"
    android:minHeight="72dip"
    android:initialLayout="@layout/widget_message"
    android:updatePeriodMillis="86400000"
/>
```

+ Tham khảo thiết kế giao diện cho widget tại

[http://developer.android.com/guide/practices/ui\\_guidelines/widget\\_design.html](http://developer.android.com/guide/practices/ui_guidelines/widget_design.html)

+ `android:updatePeriodMillis="86400000"` – với phiên bản 1.6, thời gian update nhỏ nhất là 30 phút

### 3. Xây dựng layout, giao diện widget **res/layout/widget\_message.xml**

Về cơ bản, bố cục giao diện của widget cũng giống với các app thông thường. Vẫn có thể sử dụng các layout cơ bản như linear, absolute, relative, frame.

*\*Note : Chưa thử với listview và gridview*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/widget_bg_normal" >
    <TextView
        android:id="@+id/text_widget"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
</LinearLayout>
```

#### 4. Tạo widget : WordWidget

```
package cnc.appwidget;

import android.appwidget.AppWidgetProvider;

public class WordWidget extends AppWidgetProvider {

}
```

#### 5. Thêm widget ra màn hình

#### 6. Ứng dụng làm clock Clock widget

```
@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
    int[] appWidgetIds) {
    Timer timer = new Timer();
    timer.scheduleAtFixedRate(new MyTime(context, appWidgetManager), 1,
        1000);
}

private class MyTime extends TimerTask {
    RemoteViews remoteViews;
    AppWidgetManager appWidgetManager;
    ComponentName thisWidget;
    java.text.DateFormat format = SimpleDateFormat.getInstance(
        SimpleDateFormat.MEDIUM, Locale.getDefault());

    public MyTime(Context context, AppWidgetManager appWidgetManager) {
        this.appWidgetManager = appWidgetManager;
        remoteViews = new RemoteViews(context.getPackageName(),
            R.layout.widget_message);
        thisWidget = new ComponentName(context, WordWidget.class);
    }
    @Override
    public void run() {
        remoteViews.setTextViewText(R.id.text_widget,
            "Time = " + format.format(new Date()));
        appWidgetManager.updateAppWidget(thisWidget, remoteViews);
    }
}

@Override
public void onReceive(Context context, Intent intent) {
    // v1.5 fix that doesn't call onDelete Action
    final String action = intent.getAction();
    if (AppWidgetManager.ACTION_APPWIDGET_DELETED.equals(action)) {
        final int appWidgetId = intent.getExtras().getInt(
            AppWidgetManager.EXTRA_APPWIDGET_ID,
            AppWidgetManager.INVALID_APPWIDGET_ID);
        if (appWidgetId != AppWidgetManager.INVALID_APPWIDGET_ID) {
            this.onDeleted(context, new int[] { appWidgetId });
        }
    } else {
        super.onReceive(context, intent);
    }
}
```

Xây dựng và sử dụng service : thành phần chạy ẩn trong Android. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.

### 1. Khai báo service trong manifest

```
<service android:name=".TestServices" android:enabled="true">
    <intent-filter>
        <action android:name="cnc.autoboot.TestServices"/>
    </intent-filter>
</service>
```

### 2. Khai báo class TestService

```
import java.util.*;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.os.Binder;
import android.os.IBinder;
import android.os.SystemClock;

public class TestServices extends Service {

    public void onCreate() {
        super.onCreate();
    }

    public class LocalBinder extends Binder {
        TestServices getService() {
            return TestServices.this;
        }
    }

    @Override
    public IBinder onBind(Intent intent) {
        return (mBinder);
    }

    private final IBinder mBinder = new LocalBinder();

    @Override
    public void onDestroy() {
        super.onDestroy();
    }
}
```

### 3. Thực thi một service

```
Intent serviceIntent = new Intent();
serviceIntent.setAction("cnc.autoboot.TestServices");
startService(serviceIntent);
```

Để kết thúc service ta gọi đến

```
stopService(serviceIntent);
```

#### 4. Sử dụng service

Service có thể coi tương tự activity, nó khác ở chỗ nếu có dung task manager để kết thúc ứng dụng thì service sẽ vẫn khởi động lại

Bản thân service không tự động chạy liên tục. Nó chỉ giúp cung cấp một process có thể chạy song song với hệ thống ngay cả khi ứng dụng không hoạt động. Muốn service chạy liên tục phải sử dụng timer, countdowntimer, hoặc implements runnable. Dưới đây là một ví dụ

```
public class ServiceName extends Service {
    private Timer timer = new Timer();
    protected void onCreate() {
        super.onCreate();
        startservice();
    }

    private void startservice() {
        timer.scheduleAtFixedRate( new TimerTask() {
            public void run() {
                //Do whatever you want to do every "INTERVAL"
            }
        }, 0, INTERVAL);
    }

    private void stopservice() {
        if (timer != null){
            timer.cancel();
        }
    }
}
```