

android  
developers

# Android开发教程&笔记



android

www.eoeandroid.com  
最棒Android开发社区

# 多式样 ProgressBar

撰写：地狱怒兽

联系：zyf19870302@126.com



## 普通圆形 ProgressBar



该类型进度条也就是一个表示运转的过程，例如发送短信，连接网络等等，表示一个过程正在执行中。

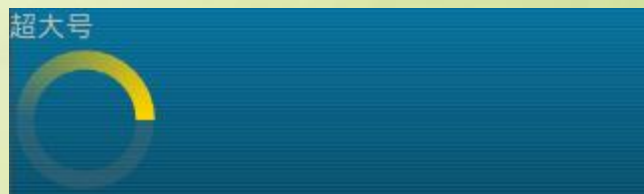
一般只要在 XML 布局中定义就可以了。

```
<ProgressBar android:id="@+id/widget43"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" android:layout_gravity="center_vertical">
</ProgressBar>
```

此时，没有设置它的风格，那么它就是圆形的，一直会旋转的进度条。

## 各大小样式圆形 ProgressBar

### 超大号圆形 ProgressBar



此时，给设置一个 style 风格属性后，该 ProgressBar 就有了一个风格，这里大号 ProgressBar 的风格是：

```
style="?android:attr/progressBarStyleLarge"
```

完整 XML 定义是:

```
<ProgressBar android:id="@+id/widget196"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleLarge">
</ProgressBar>
```

## 小号圆形 ProgressBar



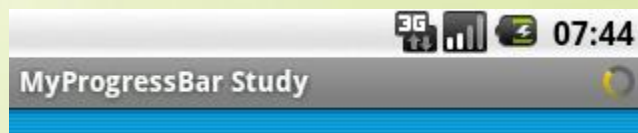
小号 ProgressBar 对应的风格是:

```
style="?android:attr/progressBarStyleSmall"
```

完整 XML 定义是:

```
<ProgressBar android:id="@+id/widget108"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleSmall">
</ProgressBar>
```

## 标题型圆形 ProgressBar



标题型 ProgressBar 对应的风格是:

```
style="?android:attr/progressBarStyleSmallTitle"
```

完整 XML 定义是:

```
<ProgressBar android:id="@+id/widget110"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleSmallTitle">
</ProgressBar>
```

代码中实现：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
    //请求窗口特色风格，这里设置成不明确的进度风格
    setContentView(R.layout.second);
    setProgressBarIndeterminateVisibility(true);
    //设置标题栏中的不明确的进度条是否可以显示
}
```

## 长形进度条

### 布局中的长形进度条



① 首先在 XML 进行布局

```
<ProgressBar android:id="@+id/progressbar_updown"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleHorizontal"
    android:layout_gravity="center_vertical"
    android:max="100"
    android:progress="50"
    android:secondaryProgress="70" >
```

讲解：

style="?android:attr/progressBarStyleHorizontal"	设置风格为长形
android:max="100"	最大进度值为 100
android:progress="50"	初始化的进度值
android:secondaryProgress="70"	初始化的底层第二个进度值
android:layout_gravity="center_vertical"	垂直居中

② 代码中运用

```
private ProgressBar myProgressBar;
//定义 ProgressBar
myProgressBar = (ProgressBar) findViewById(R.id.progressbar_updown);
```

```
//ProgressBar 通过 ID 来从 XML 中获取
myProgressBar.incrementProgressBy(5);
//ProgressBar 进度值增加 5
myProgressBar.incrementProgressBy(-5);
//ProgressBar 进度值减少 5
myProgressBar.incrementSecondaryProgressBy(5);
//ProgressBar 背后的第二个进度条 进度值增加 5
myProgressBar.incrementSecondaryProgressBy(-5);
//ProgressBar 背后的第二个进度条 进度值减少 5
```

## 页面标题中的长形进度条



代码实现:

① 先设置一下窗口风格特性

```
requestWindowFeature(Window.FEATURE_PROGRESS);
//请求一个窗口进度条特性风格
setContentView(R.layout.main);
setProgressBarVisibility(true);
//设置进度条可视
```

② 然后设置进度值

```
setProgress(myProgressBar.getProgress() * 100);
//设置标题栏中前景的一个进度条进度值
setSecondaryProgress(myProgressBar.getSecondaryProgress() * 100);
//设置标题栏中后面的一个进度条进度值
//ProgressBar.getSecondaryProgress() 是用来获取其他进度条的进度值
```



# ProgressDialog

## ProgressDialog 中的圆形进度条



ProgressDialog 一般用来表示一个系统任务或是开启任务时候的进度，有一种稍等的意思。  
代码实现：

```
ProgressDialog myDialog=new ProgressDialog(this);  
//实例化  
myDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);  
//设置进度条风格，风格为圆形，旋转的  
myDialog.setTitle("Google");  
//设置 ProgressDialog 标题  
myDialog.setMessage(getResources().getString(R.string.second));  
//设置 ProgressDialog 提示信息  
myDialog.setIcon(R.drawable.android);  
//设置 ProgressDialog 标题图标  
myDialog.setButton("Google",this);  
//设置 ProgressDialog 的一个 Button  
myDialog.setIndeterminate(false);  
//设置 ProgressDialog 的进度条是否不明确  
myDialog.setCancelable(true);  
//设置 ProgressDialog 是否可以按退回按键取消  
myDialog.show();  
//让 ProgressDialog 显示
```

## ProgressDialog 中的长形进度条



代码实现:

```
ProgressDialog myDialog=new ProgressDialog(this);
//实例化
myDialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
//设置进度条风格，风格为长形，有刻度的
myDialog.setTitle("地狱怒兽");
//设置 ProgressDialog 标题
myDialog.setMessage(getResources().getString(R.string.second));
//设置 ProgressDialog 提示信息
myDialog.setIcon(R.drawable.android);
//设置 ProgressDialog 标题图标
myDialog.setProgress(59);
//设置 ProgressDialog 进度条进度
myDialog.setButton("地狱曙光",this);
//设置 ProgressDialog 的一个 Button
myDialog.setIndeterminate(false);
//设置 ProgressDialog 的进度条是否不明确
myDialog.setCancelable(true);
//设置 ProgressDialog 是否可以按退回按键取消
myDialog.show();
//让 ProgressDialog 显示
```

# AlertDialog.Builder

## AlertDialog 中的圆形 ProgressBar



① 先来设计一个 Layout，待会儿作为一个 View，加入 AlertDialog.Builder

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_gravity="center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <LinearLayout android:id="@+id/LinearLayout01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </LinearLayout>
    <ProgressBar android:layout_gravity="center_vertical|center_horizontal"
        android:layout_height="wrap_content"
        android:progress="57"
        android:id="@+id/my View_ProgressBar2"
        android:layout_width="wrap_content">
```



```
</ProgressBar>  
</LinearLayout>
```

## ② 代码罗

```
private AlertDialog.Builder AlterD,AlterD2;  
//定义提示对话框  
private LayoutInflater layoutInflater;  
//定义布局过滤器  
private LinearLayout myLayout;  
//定义布局  
layoutInflater2=(LayoutInflater) getSystemService(this.LAYOUT_INFLATER_SERVICE);  
//获得系统的布局过滤服务  
myLayout2=(LinearLayout) layoutInflater2.inflate(R.layout.roundprogress, null);  
//得到事先设计好的布局  
AlterD2.setTitle(getResources().getString(R.string.RoundO));  
//设置对话框标题  
AlterD2.setIcon(R.drawable.ma);  
//设置对话框图标  
AlterD2.setMessage(getResources().getString(R.string.ADDView));  
//设置对话框提示信息  
AlterD2.setView(myLayout2);  
//设置对话框中的 View  
AlterD2.show();  
//让对话框显示
```




## AlertDialog 中的长形 ProgressBar（可控制）



① 先来设计一个 Layout，待会儿作为一个 View，加入 AlertDialog.Builder

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_gravity="center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <Button
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_width="50dp"
        android:id="@+id/myView_BT_Down">
    </Button>
    <ProgressBar
        android:layout_gravity="center_vertical"
        android:layout_height="wrap_content"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="100dp"/>
```



```

        android:id="@+id/myView_ProgressBar"
        android:progress="57"
        android:layout_width="178dp">
    </ProgressBar>
    <Button android:layout_height="wrap_content"
        android:text="+"
        android:layout_width="50dp"
        android:id="@+id/myView_BT_Up">
    </Button>
</LinearLayout>

```

## ② 代码罗

```

private AlertDialog.Builder AlterD,AlterD2;
//定义提示对话框

private LayoutInflater layoutInflater;
//定义布局过滤器

private LinearLayout myLayout;
//定义布局

layoutInflater=(LayoutInflater) getSystemService(this.LAYOUT_INFLATER_SERVICE);
//获得系统的布局过滤服务

myLayout=(LinearLayout) layoutInflater.inflate(R.layout.myview, null);
//得到事先设计好的布局


myup=(Button) myLayout.findViewById(R.id.myView_BT_Up);
mydown=(Button) myLayout.findViewById(R.id.myView_BT_Down);
mypro=(ProgressBar)myLayout.findViewById(R.id.myView_ProgressBar);
//通过 myLayout.findViewById 来获取自定义 View 中的 Widget 控件元素


myup.setOnClickListener(this);
//设置对话框 View 中的按钮监听器
mydown.setOnClickListener(this);
//设置对话框 View 中的按钮监听器
mypro.setProgress(Tag);
//设置一个 Tag 作为进度值
AlterD.setTitle(getResources().getString(R.string.RectO));
//设置对话框标题
AlterD.setIcon(R.drawable.mb);
//设置对话框图标
AlterD.setMessage(getResources().getString(R.string.ADDView));
//设置对话框提示信息
AlterD.setView(myLayout);
//设置对话框添加的 View
AlterD.setPositiveButton("OK", new DialogInterface.OnClickListener){

```

```

@Override
public void onClick(DialogInterface dialog, int which) {
    // TODO Auto-generated method stub
    MyProgressBar.Tag=mypro.getProgress();
    });
//设置对话框按钮，以及按钮的事件监听器
AlterD.show();
//让对话框显示

```

### ③ 进度条进度值的按钮事件

myup.setOnClickListener(this);  
 //设置对话框 View 中的按钮监听器  
 mydown.setOnClickListener(this);  
 //设置对话框 View 中的按钮监听器  
 对应的代码:

```

@Override
public void onClick(View button) {
    // TODO Auto-generated method stub
    SwitchUPorDown(button);
}

private void SwitchUPorDown(View button) {
    switch (button.getId()) {
        case R.id.myView_BT_Up: {
            mypro.incrementProgressBy(1);
        }
        break;
        case R.id.myView_BT_Down: {
            mypro.incrementProgressBy(-1);
        }
        break;
        default:
            break;
    }
}

```



# App Widget 中的进度条

## Widget 中的圆形 ProgressBar

这个很简单，在 Widget 中没有多大意思，不再赘述。

## Widget 中的长形 ProgressBar（可控制）



Widget 的实现就不再重复，假设您已经把 Widget 布局，相应设置已经设置好了。也可以在桌面加入类似上面图中的样式。

现在我们来实现一下按钮事件，与进度条的交互。

下面还是简单讲解一下 Widget 的设计与部署。

www.eoeandroid.com  
最棒Android开发社区

### ① 设计 Widget 布局

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:background="@drawable/widget"
    android:layout_height="74dp"
    android:layout_width="296dp">
    <Button
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_gravity="center_vertical"
        android:layout_width="50dp"
        android:id="@+id/widget_BT_Down"
        android:layout_marginLeft="10dp">
    </Button>
    <ProgressBar
        android:layout_gravity="center_vertical"
        android:layout_height="wrap_content"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="178dp"
        android:id="@+id/widget_ProgressBar">
    </ProgressBar>
    <Button
        android:layout_height="wrap_content"
        android:text="+"
        android:layout_gravity="center_vertical"
        android:layout_width="50dp"
        android:id="@+id/widget_BT_Up">
    </Button>
</LinearLayout>
```

### ② 新增一个.res/xml 目录，加入 appwidget-provider

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:initialLayout="@layout/widgetlayout"
    android:updatePeriodMillis="8660000"
    android:minWidth="296dp"
    android:minHeight="74dp">
</appwidget-provider>
```

### ③ 实现一个 AppWidgetProvider 子类

```
package zyf.test.ProgressBar;

import android.appwidget.AppWidgetManager;
import android.appwidget.AppWidgetProvider;
import android.content.Context;
import android.content.Intent;

public class App extends AppWidgetProvider {

    @Override
    public void onEnabled(Context context) {
        // TODO Auto-generated method stub
        super.onEnabled(context);
    }

    @Override
    public void onReceive(Context context, Intent intent) {
        // TODO Auto-generated method stub
        super.onReceive(context, intent);
    }


    @Override
    public void onUpdate(Context context, AppWidgetManager appWidgetManager,
        int[] appWidgetIds) {
        // TODO Auto-generated method stub
        super.onUpdate(context, appWidgetManager, appWidgetIds);
    }
}
```

### ④ 配置 Manifest, 进行注册

```
<receiver android:name="AppWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"></action>
    </intent-filter>
    <meta-data
        android:resource="@xml/appwidget"
        android:name="android.appwidget.provider">
    </meta-data>
</receiver>
```

这里实现按钮与进度条的交互。(Widget 自己广播发送与接收)

### ① 按钮的消息发送



```
@Override
public void onUpdate(Context context, AppWidgetManager appWidgetManager,
    int[] appWidgetIds) {
    // TODO Auto-generated method stub
    final int N = appWidgetIds.length;

    // Perform this loop procedure for each App Widget that belongs to this provider
    for (int i=0; i<N; i++) {
        int appWidgetId = appWidgetIds[i];
        RemoteViews views=
            new RemoteViews(context.getPackageName(), R.layout.widgetlayout);

        Intent UPintent=new Intent("zyf.test.widget.UP");
        Intent DOWNintent=new Intent("zyf.test.widget.DOWN");
        //实例化 两个带有 Action 的 Intent
        PendingIntent pendingIntentUp
            =PendingIntent.getBroadcast(context, 0, UPintent, 0);
        PendingIntent pendingIntentDown
            =PendingIntent.getBroadcast(context, 0, DOWNintent, 0);
        //实例化两个以 Intent 来构造的 PendingIntent
        views.setOnClickPendingIntent(R.id.widget_BT_Up, pendingIntentUp);
        views.setOnClickPendingIntent(R.id.widget_BT_Down, pendingIntentDown);
        //给 View 上的两个按钮绑定事件，这里是广播消息的发送
        appWidgetManager.updateAppWidget(appWidgetId, views);
    }
}
```

### ② Widget 自身消息接收，使用 intent.getAction()来获取 Action

```
@Override
public void onReceive(Context context, Intent intent) {
    // TODO Auto-generated method stub
    super.onReceive(context, intent);
    if(intent.getAction().equals("zyf.test.widget.UP")){
        Tag+=5;
        if(Tag>100){
            Tag=100;
        }
        views.setProgress(R.id.widget_ProgressBar, 100, Tag, false);
        appManager.updateAppWidget(this Widget, views);
    }
}
```



```

        if(intent.getAction().equals("zyf.test.widget.DOWN")){
            Tag-=5;
            if(Tag<0){
                Tag=0;
            }
            views.setProgressBar(R.id.widget_ProgressBar, 100, Tag, false);
            appManager.updateAppWidget(this Widget, views);
        }
    }
}

```

### ③ 进度条的进度值设置

```

views.setProgressBar(R.id.widget_ProgressBar, 100, Tag, false);
//设置 Widget 上的进度条的进度值
//第一个参数,Widget 上进度条 ID
//第二个参数, 进度条最大值
//第三个参数 Tag, 一个 int 值, 就是设置的进度值
//第四个参数, 是否是要进度条不确定

```

**注意了**, Widget 自身的 `onReceive()` 方法如果要接收其他的 Action 广播。那就必须在 Manifest 中, 在 Intent-filter 中添加 Action:

```

<receiver android:name="AppWidget">
    <intent-filter>
        <action android:name="android.appwidget.action.APPWIDGET_UPDATE"></action>
        <action android:name="zyf.test.widget.UP"></action>
        <action android:name="zyf.test.widget.DOWN"></action>
    </intent-filter>
    <meta-data
        android:resource="@xml/appwidget"
        android:name="android.appwidget.provider">
    </meta-data>
</receiver>

```

如有什么问题和不对地方, 请联系我: [zyf19870302@126.com](mailto:zyf19870302@126.com) (地狱怒兽)

android  
developers



android

www.eoeandroid.com  
最棒Android开发社区