对于技术类的博客，我们永远追寻最新API脚步，在API22之前我们使用标题栏基本都是在ActionBarActivity的Activity中处理的，而API22之后，谷歌遗弃了ActionBarActivity，推荐我们也可以说是强制我们使用AppCompatActivity。

然而ActionBarActivity与AppCompatActivity的使用大同小异，下面我们将介绍最新的Activity组件之AppCompatActivity的使用。

最后的效果与微信基本相同，图如下：



**1.初探AppCompatActivity**

我们按照AndroidStudio的默认顺序创建项目，默认Activity继承自AppCompatActivity。代码如下：

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

}

}

运行提到如下所示的界面：



其结果与继承自Activity并无二致，丑爆了。下面我们来一步一步扩展其标题栏，让他的内容更丰富多彩。

**2.AppCompatActivity与Toolbar结合**

其实我们并不是要使用AppCompatActivity自带的标题栏，那样扩展会很麻烦，在14年的Android 5.0的时候就用Toolbar替代了ActionBar,而ActionBarActivity又被AppCompatActivity替代，那么使用方法就是引入Toolbar设置到Activity中。

㈠首先我们必须在配置文件中，写入Toolbar，代码如下：

<android.support.v7.widget.Toolbar

android:id="@+id/activity\_main\_toolbar"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:minHeight="?attr/actionBarSize"

android:background="?attr/colorPrimary">

</android.support.v7.widget.Toolbar>

解释①android:minHeight="?attr/actionBarSize"：设置标题栏最小高度为ActionBar的高度。

解释②android:background="?attr/colorPrimary"：该主题下的主色。也就是默认的灰色。

我们知道在AndroidManifest.xml清单文件下application中设置了android:theme="@style/AppTheme"而查看AppTheme看到如下样式

<resources>

*<!-- Base application theme. -->*

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

*<!-- Customize your theme here. -->*

</style>

</resources>

从名字我们就可以看出来，默认的标题栏为黑色。我们使用了toolbar就必须修改样式文件，将原来的标题栏去掉，修改后的样式文件如下：

<resources>

*<!-- Base application theme. -->*

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

*<!-- Customize your theme here. -->*

</style>

</resources>

如下这个时候运行，那么你将看到界面也是没有标题栏的，因为你并没有设置到AppCompatActivity中去。

㈡将Toolbar显示到界面

我们创建一个方法名为initActionBar()，首先当然是获取ToolBar控件，然后将Toolbar设置到Activity中去。代码如下：

public void initActionBar() {

Toolbar toolbar = (Toolbar) findViewById(R.id.*activity\_main\_toolbar*);

setSupportActionBar(toolbar);

}

现在运行，我们将得到如下界面：



依然不是你期待的效果，下面我们将介绍AppCompatActivity的细节。

**3.AppCompatActivity详解**

㈠设置title

我们可以看到，标题栏始终显示我们的项目名，这样显然给用户看是不理想的。那么怎么设置标题栏的字符串。我们可以在setSupportActionBar(toolbar)之前加入如下代码：

setTitle("liyuanjinglylyj");

此setTitle("liyuanjinglyj")是AppCompatActivity的方法，并不是toolbar的方法，那么我们运行一下程序将得到如下界面：



那么我们将刚才的AppCompatActivity的setTitle换成Toolbar的setTitle看看效果会怎么样，其实结果如上图一样，没有任何变化。

可能你希望title显示在中间，很遗憾，toolbar与AppCompatActivity并没有提供相关的方法，不过可以在toolbar中加入一个TextView，配置其属性，让其显示在最中间。

㈡设置回退按钮

假如我的这个界面并不是主界面，而是一个子界面，这个时候我有一个需求，需要回退到上一个界面那么怎么设置左边的图标并实现其方法。其实很简单，在setSupportActionBar(toolbar);后面加入如下代码：

toolbar.setNavigationIcon(R.drawable.*back*);

toolbar.setNavigationOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

finish();

}

});

为什么在setSupportActionBar(toolbar);后面加入而不在前面，你可以加入到前面试试，虽然可以显示回退的图片，但是点击并没有调用该按钮的点击事件，而在setSupportActionBar(toolbar);后面设置点击才有反应。一定要记住，不然回退是不会起作用的。

运行结果如下：



㈢设置logo

设置logo在标题栏并不少见，比如网易APP首页就的网易设置的logo，其实代码很简单，也仅仅需要一条代码即可，在setSupportActionBar(toolbar)前加入如下代码：

toolbar.setLogo(R.id.app\_logo);

运行结果如下：



㈣设置副标题

在setSupportActionBar(toolbar)前加入如下代码：：

toolbar.setSubtitle("liyuanjing");

运行界面后，如下：



㈤设置菜单

我们知道onCreateOptionsMenu是上下文菜单，同理可以直接在该方法中设置菜单，代码如下：

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.*menu\_main*, menu);

return true;

}

此方法为Activity方法。因为你在上面已经讲Toolbar设置到了Activity中，所以默认也会显示到标题栏中，如果你觉得这样不习惯，那么你可以添加一条语句，不过这条语句添加不添加都没关系。

toolbar.setOnCreateContextMenuListener(this);

运行程序，如下图所示：



**4.Toolbar的Style**

3标题下的Toolbar是蓝色，且有很好Style，那么这是怎么设置呢，虽然我们这里重点是剖析Activity但是遇到有关Activity标题栏样式的还是要提到。

我们粗略的讲解上面的标题栏用到了哪些Style：

①<item name="colorPrimary">#4876FF</item>：如第一步直接在toolbar控件下面设置的属性一样，就是toolbar的颜色。

②<item name="colorPrimaryDark">#3A5FCD</item>：状态栏颜色也就是标题栏上面的那个颜色，而我用的是小米1S测试的，其固件并没有很好的实现其功能，所以在该系统下并没有实现其效果。

③<item name="android:windowBackground">@android:color/white</item>：窗口背景色，也就是Activity标题栏下面所有地方的颜色。

当我们设置了如上三个属性，那么效果图就如上面所示了，完整代码如下：

*<!-- Base application theme. -->*

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

*<!-- toolbar（actionbar）颜色 -->*

<item name="colorPrimary">#4876FF</item>

*<!-- 状态栏颜色 -->*

<item name="colorPrimaryDark">#3A5FCD</item>

*<!-- 窗口的背景颜色 -->*

<item name="android:windowBackground">@android:color/white</item>

</style>

**5.Toolbar菜单样式**

我们来看看如果直接使用上面的点击后边的菜单会有什么效果：



你有看到那个APP菜单弹出挡住了标题栏吗？，肯定没有吧，要是有的话，你肯定立即卸载了该应用了。

下面我们的任务就是将该菜单的样式和其他APP一样，在标题栏下面弹出。

首先要说明一下，toolbar菜单默认样式的父类为Widget.AppCompat.Light.PopupMenu.Overflow，那么要更改toobar中菜单的弹出的样式，就必须继承这个父类的样式。

其代码如下：

*<!-- Base application theme. -->*

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

*<!-- toolbar（actionbar）颜色 -->*

<item name="colorPrimary">#4876FF</item>

*<!-- 状态栏颜色 -->*

<item name="colorPrimaryDark">#3A5FCD</item>

*<!-- 窗口的背景颜色 -->*

<item name="android:windowBackground">@android:color/white</item>

<item name="actionOverflowMenuStyle">@style/LYJMenuStyle</item>

</style>

<style name="LYJMenuStyle" parent="@style/Widget.AppCompat.Light.PopupMenu.Overflow">

<item name="overlapAnchor">false</item>

</style>

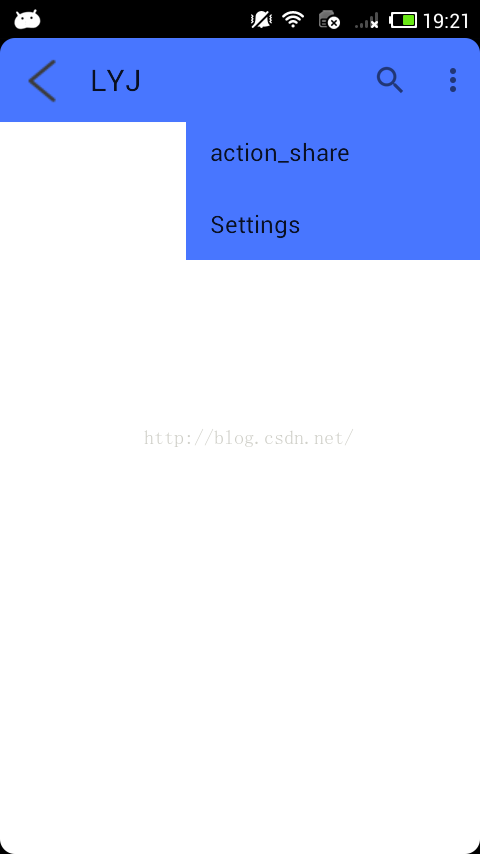
设置这一个属性那么其弹出的样式就与其他APP一样不会覆盖标题栏。

还有其他属性这里简要说明一下：

①<item name="android:popupBackground">?attr/colorPrimary</item>：弹出菜单背景色为标题栏的背景色

②<item name="android:dropDownVerticalOffset">0dip</item>：弹出菜单与标题栏的垂直间距

将这两条加入LYJMenuStyle中就可以了实现如下图所示的弹出菜单：



最后介绍一下菜单里面重要的几个属性：

app:showAsAction有三个值：

always：总是显示在界面上  
never：不显示在界面上，只让出现在右边的三个点中  
ifRoom：如果有位置才显示，不然就出现在右边的三个点中

可以用 | 同时使用两个上面的值。

系统也为菜单提供了默认的分享菜单与查询菜单，代码如下：

app:actionViewClass="android.support.v7.widget.SearchView"

app:actionProviderClass="android.support.v7.widget.ShareActionProvider"

从名字想必不需要过多解释，最后一个就是显示的优先级，也就是显示的顺序，谁在最前面，谁在后面：

android:orderInCategory数值越小，显示靠前，且优先级最大。