**Design库-TabLayout属性详解**

[](http://www.jianshu.com/u/cfcd1814672c)

作者 [自导自演的机器人](http://www.jianshu.com/u/cfcd1814672c) 关注

2016.06.02 01:17\* 字数 1226 阅读 32231评论 86喜欢 162

**序**



上图是简书Android端的主页Tab,在其他的App中Tab也是很常见的，它的实现方式也有很多：TabHost，自定义控件（第三方库），RadioGroup等等。这里主要介绍Android Design库中的TabLayout的使用。

**1.什么是TabLayout**

在源码中给出了TabLayout的定义：

TabLayout provides a horizontal layout to display tabs.

意思很明显：TabLayout提供了一个水平的布局用来展示Tabs。

Design库：AS有直接的引用，如果是Eclipse这里提供一个通道-[Design库](http://download.csdn.net/detail/android_yh/9525604)

**特别说明：**

Caused by: java.lang.IllegalArgumentException: You need to use a Theme.AppCompat theme (or descendant) with the design library.

在清单文件中设置如下代码即可：

android:theme="@style/Theme.AppCompat"

**2.TabLayout的基本使用方式**

方式一：

1.在布局中加入该控件：

<android.support.design.widget.TabLayout

          android:id="@+id/tabLayout"

          android:layout\_width="wrap\_content"

          android:layout\_height="wrap\_content"/>

2.在代码中

tabLayout= (TabLayout) findViewById(R.id.tabLayout);

tabLayout.addTab(tabLayout.newTab().setText("Tab 1"));

tabLayout.addTab(tabLayout.newTab().setText("Tab 2"));

tabLayout.addTab(tabLayout.newTab().setText("Tab 3"));

3.显示效果



感觉还不错吧，挺简单就实现了这个UI效果。

方式二：

<android.support.design.widget.TabLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content">

<android.support.design.widget.TabItem

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Tab1"/>

...

</android.support.design.widget.TabLayout>

这样同样也可以实现方式一的效果，可是单单这样并不能满足我们。于是接下来看看有什么属性可以使用。

**3.改变下TabLayout的颜色**

上面的Tab颜色感觉不好看，打算换换：

1.改变选中字体的颜色

app:tabSelectedTextColor="@android:color/holo\_orange\_light"

2.改变未选中字体的颜色

app:tabTextColor="@color/colorPrimary"

3.改变指示器下标的颜色

app:tabIndicatorColor="@android:color/holo\_orange\_light"

4.改变整个TabLayout的颜色

app:tabBackground="color"

于是现在我的Tab变成了这个样子：



**4.改变TabLayout内部字体大小**

总觉得这个字体有点小了，于是想找方法把这个字变得大一点，

好像没有直接变大的方法，可是找到了这个：

app:tabTextAppearance="@android:style/TextAppearance.Holo.Large"//设置文字的外貌

效果：



**5.改变指示器下标的高度**

既然字体变大了，指示器太小就显得不太好看了，

设置指示器下标的高度：

app:tabIndicatorHeight="4dp"

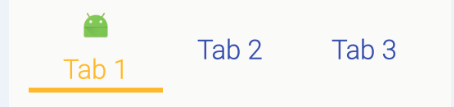
效果：



**6.添加图标**

有时候Tab只有文字感觉有点单调了：

tabLayout.addTab(tabLayout.newTab().setText("Tab 1").setIcon(R.mipmap.ic\_launcher));



**7.Tab的模式**

数据很多的时候我们应该怎么办呢，简书中的第二个Tab就是可以滑动的：

我们先多加几个tab：

tabLayout.addTab(tabLayout.newTab().setText("Tab 4"));

tabLayout.addTab(tabLayout.newTab().setText("Tab 5"));

tabLayout.addTab(tabLayout.newTab().setText("Tab 6"));

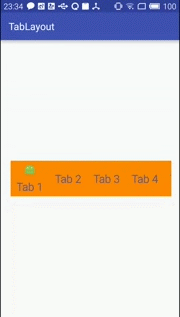
tabLayout.addTab(tabLayout.newTab().setText("Tab 7"));

然后设置属性为：

app:tabMode="scrollable"

默认是fixed：固定的，标签很多时候会被挤压，不能滑动。

效果如下：



**8.加入Padding**

设置Tab内部的子控件的Padding：

app:tabPadding="xxdp"

app:tabPaddingTop="xxdp"

app:tabPaddingStart="xxdp"

app:tabPaddingEnd="xxdp"

app:tabPaddingBottom="xxdp"

设置整个TabLayout的Padding：

app:paddingEnd="xxdp"

app:paddingStart="xxdp"

**9.内容的显示模式**

app:tabGravity="center"//居中，如果是fill，则是充满

**10.Tab的宽度限制**

设置最大的tab宽度：

app:tabMaxWidth="xxdp"

设置最小的tab宽度：

app:tabMinWidth="xxdp"

**11.Tab的“Margin”**

TabLayout开始位置的偏移量：

app:tabContentStart="100dp"

**12.TabLayout的监听事件**

选中了某个tab的监听事件OnTabSelectedListener()：

tabLayout.setOnTabSelectedListener(newTabLayout.OnTabSelectedListener() {

@Override

public voidonTabSelected(TabLayout.Tab tab) {

//选中了tab的逻辑

}

@Override

public voidonTabUnselected(TabLayout.Tab tab) {

//未选中tab的逻辑

}

@Override

public voidonTabReselected(TabLayout.Tab tab) {

//再次选中tab的逻辑

}

});

**13.和ViewPager的联动**

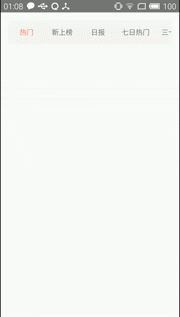
最后也是最重要的：

tabLayout.setupWithViewPager(Viewpager);

一行代码和ViewPager联动起来，简单粗暴。

和ViewPager联动，需要重写adapter 的getPageTitle,否则无法显示TabLayout上的标签

**14.简单模仿简书Android端的Tab**



主要就是设置下标的高度为0，相当于没有下标。

app:tabIndicatorHeight="0dp"

然后设置背景颜色以及选中文字颜色

app:tabSelectedTextColor="#ff7a61"

app:tabBackground="#f6f4f2"//这里不能直接写RGB，需要@color/xx

最后设置Tab的模式：

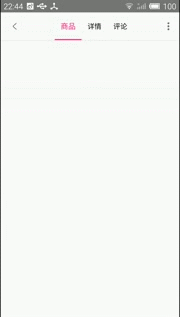
app:tabMode="scrollable"

当然Tablayout还可以做很多的样式，如果有关于Tablayout的注意事项或者相关知识的文章，希望大家可以分享给我，多谢！

**15.仿京东商品详情Android端的Tab**

主要原理是setCustomView()加载自定义视图，来实现字体大小的改变，默认的TabLayout不能改变。

[Design库-TabLayout仿京东商品详情Tab](http://www.jianshu.com/p/116f87abdaea)



**补充：默认选中某项**

tablayout.getTabAt(position).select();

**16.** **TabLayout用法详解及自定义样式**

TabLayout的默认样式:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1 | app:theme="@style/Widget.Design.TabLayout" |

从系统定义的该样式继续深入:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <style name="Widget.Design.TabLayout" parent="Base.Widget.Design.TabLayout">   <item name="tabGravity">fill</item>   <item name="tabMode">fixed</item>  </style>  <style name="Base.Widget.Design.TabLayout" parent="android:Widget">   <item name="tabMaxWidth">264dp</item>   <item name="tabIndicatorColor">?attr/colorAccent</item>   <item name="tabIndicatorHeight">2dp</item>   <item name="tabPaddingStart">12dp</item>   <item name="tabPaddingEnd">12dp</item>   <item name="tabBackground">?attr/selectableItemBackground</item>   <item name="tabTextAppearance">@style/TextAppearance.Design.Tab</item>   <item name="tabSelectedTextColor">?android:textColorPrimary</item>  </style> |

接着,看看系统定义Tab文本的样式(注意textAllcaps这个属性):

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | <style name="TextAppearance.Design.Tab" parent="TextAppearance.AppCompat.Button">   <item name="android:textSize">14dp</item>   <item name="android:textColor">?android:textColorSecondary</item>   <item name="textAllCaps">true</item>  </style> |

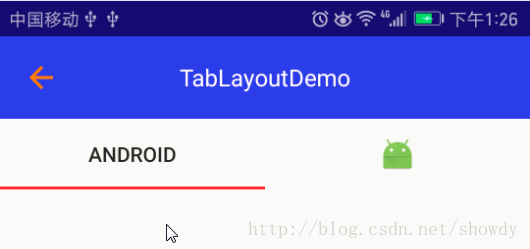
从系统定义TabLayout的默认样式可以看出,我们可以改变TabLayout对应的系统样式的属性值来适配我们自己的需求.

**TabLayout的基本用法**

TabLayout独立使用使用时,可以xml布局中静态添加tab个数及其样式,也可以动态添加Tab的个数及其样式,如:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <android.support.design.widget.TabLayout   android:id="@+id/tablayout"   android:background="@color/colorPrimary"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content">   <android.support.design.widget.TabItem    android:layout\_width="match\_parent"    android:layout\_height="wrap\_content"    android:text="Android"/>   <android.support.design.widget.TabItem    android:layout\_width="match\_parent"    android:layout\_height="wrap\_content"    android:icon="@mipmap/ic\_launcher"/>  </android.support.design.widget.TabLayout> |



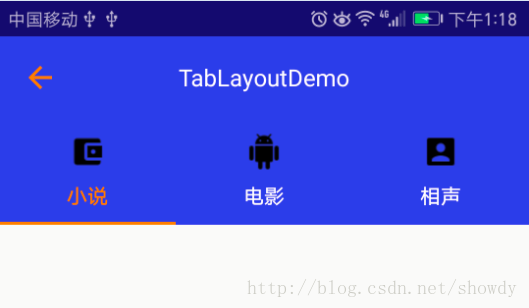
或者:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | <android.support.design.widget.TabLayout   android:id="@+id/tablayout"   android:background="@color/colorPrimary"   android:layout\_width="match\_parent"   android:layout\_height="wrap\_content"/> |

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | private int[] images = new int[]{       R.drawable.ic\_account\_balance\_wallet\_black,       R.drawable.ic\_android\_black,       R.drawable.ic\_account\_box\_black};   private String[] tabs = new String[]{"小说", "电影", "相声"};   TabLayout tabLayout = (TabLayout) findViewById(R.id.tablayout);   tabLayout.addTab(tabLayout.newTab().setIcon(images[0]).setText(tabs[0]),true);   tabLayout.addTab(tabLayout.newTab().setIcon(images[1]).setText(tabs[1]),false);   tabLayout.addTab(tabLayout.newTab().setIcon(images[2]).setText(tabs[2]),false); |



TabLayout在实际开发中最多的是与ViewPager联合使用,实现TabLayout与ViewPager的联动:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | <android.support.design.widget.TabLayout    android:id="@+id/tablayout"    android:layout\_width="match\_parent"    android:layout\_height="wrap\_content"    android:background="@color/colorPrimary"    app:tabGravity="fill"    app:tabIndicatorColor="@android:color/holo\_orange\_dark"    app:tabIndicatorHeight="2dp"    app:tabMode="fixed"    app:tabSelectedTextColor="@android:color/holo\_orange\_dark"    app:tabTextAppearance="@style/CustomTabTextAppearanceStyle"    app:tabTextColor="@android:color/white"    app:theme="@style/Widget.Design.TabLayout"/>   <android.support.v4.view.ViewPager    android:id="@+id/view\_pager"    android:layout\_width="match\_parent"    android:layout\_height="match\_parent"/> |

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | TabLayout tabLayout = (TabLayout) findViewById(R.id.tablayout);  ViewPager viewPager = (ViewPager) findViewById(R.id.view\_pager);  viewPager.setAdapter(new TabPagerAdapter(getSupportFragmentManager()));  tabLayout.setupWithViewPager(viewPager); |



值得注意的是:

在TabPagerAdapter中需要实现getPagerTitle()否则,TabLayout的Tab将不显示,先看TabLayout#setupWithPager()源码,发现Tab的添加是在populateFromPagerAdapter()中实现,实现源码如下,可以看出该方法调用了PagerAdpater#getPagerTitle()为Tab设置文本信息,如果我们自定义的Adapter没有实现getPagerTitle()将会导致Tab不显示文本信息.

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | void populateFromPagerAdapter() {    removeAllTabs();    if (mPagerAdapter != null) {     final int adapterCount = mPagerAdapter.getCount();     for (int i = 0; i < adapterCount; i++) {      addTab(newTab().setText(mPagerAdapter.getPageTitle(i)), false);     }     // Make sure we reflect the currently set ViewPager item     if (mViewPager != null && adapterCount > 0) {      final int curItem = mViewPager.getCurrentItem();      if (curItem != getSelectedTabPosition() && curItem < getTabCount()) {       selectTab(getTabAt(curItem));      }     }    }   } |

另外, 我们发现getPagerTitle()方法的返回值CharSequence而不是String,那么Tab的文本信息的设置将变得更加灵活,比如设置一个SpanableString,将图片和文本设置Tab的文本.

[?](http://www.jb51.net/article/101912.htm)

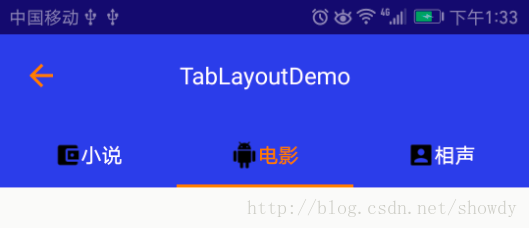
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | @Override    public CharSequence getPageTitle(int position) {     Drawable image = TablayoutActivity.this.getResources().getDrawable(images[position]);     image.setBounds(0, 0, image.getIntrinsicWidth()/2, image.getIntrinsicHeight()/2);     ImageSpan imageSpan = new ImageSpan(image, ImageSpan.ALIGN\_BOTTOM);     SpannableString ss = new SpannableString(" "+tabs[position]);     ss.setSpan(imageSpan, 0, 1, Spannable.SPAN\_EXCLUSIVE\_EXCLUSIVE);     return ss;    } |



但是Tab缺没有显示任何信息,一片空白,从上面提到的TabLayout的系统默认样式中我们发现: <item name="textAllCaps">true</item>,这会阻止ImageSpan渲染出来,我们只需要将textAllCaps改为false即可,如下定义,再次运行,成功显示

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3 | <style name="CustomTabTextAppearanceStyle" parent="TextAppearance.Design.Tab">   <item name="textAllCaps">false</item>  </style> |



  修改Indicator的长度:

从TabLayout的源码可以看出Indicator的绘制,是在其内部类SlidingTabStrip中绘制,而SlingTabStrip类继承LinearLayout,源码如下:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | @Override  public void draw(Canvas canvas) {   super.draw(canvas);   // Thick colored underline below the current selection   if (mIndicatorLeft >= 0 && mIndicatorRight > mIndicatorLeft) {    canvas.drawRect(mIndicatorLeft, getHeight() - mSelectedIndicatorHeight,      mIndicatorRight, getHeight(), mSelectedIndicatorPaint);   }  } |

在onDraw()中主要是就绘制一个Rect,并且宽度是根据mIndicatorLeft和mIndicatorRight设置的,而mIndicatorLeft等的宽度来自SlidingTabStrip的child,而Child就相当于一个Tab,这样我们就通过修改Child的margin来设置mIndicatorLeft的值.

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | public void setIndicator(TabLayout tabs, int leftDip, int rightDip) {    Class<?> tabLayout = tabs.getClass();    Field tabStrip = null;    try {     tabStrip = tabLayout.getDeclaredField("mTabStrip");    } catch (NoSuchFieldException e) {     e.printStackTrace();    }    tabStrip.setAccessible(true);    LinearLayout llTab = null;    try {     llTab = (LinearLayout) tabStrip.get(tabs);    } catch (IllegalAccessException e) {     e.printStackTrace();    }    int left = (int) TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP, leftDip, Resources.getSystem().getDisplayMetrics());    int right = (int) TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP, rightDip, Resources.getSystem().getDisplayMetrics());    for (int i = 0; i < llTab.getChildCount(); i++) {     View child = llTab.getChildAt(i);     child.setPadding(0, 0, 0, 0);     LinearLayout.LayoutParams params = new LinearLayout.LayoutParams(0, LinearLayout.LayoutParams.MATCH\_PARENT, 1);     params.leftMargin = left;     params.rightMargin = right;     child.setLayoutParams(params);     child.invalidate();    }   } |

然后在代码中调用即可,但是要注意,必须要在Tablayout渲染出来后调用,我们可以选择view.post()方法来实现:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | tabLayout.post(new Runnable() {    @Override    public void run() {     setIndicator(tabLayout, 20, 20);    }  }); |

最后得到效果图如下:



自定义TabLayout的TabItem及TabItem的点击事件

在TabLayout的Api是没有提供TabItem点击事件的方法,如果我们想实现如下效果图,怎么办?



先自定义一个TabItem:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"   android:layout\_width="match\_parent"   android:layout\_height="match\_parent"   android:gravity="center"   android:orientation="horizontal">   <TextView    android:id="@+id/txt\_title"    android:layout\_width="wrap\_content"    android:layout\_height="wrap\_content"    android:gravity="center"    android:textSize="14sp" />   <ImageView    android:id="@+id/img\_title"    android:src="@drawable/indicator"    android:layout\_width="wrap\_content"    android:layout\_height="wrap\_content"    android:layout\_marginLeft="5dp" />  </LinearLayout> |

在自定义的Adapter中可以定义一个getTabView的方法:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public View getTabView(int position){   View view = LayoutInflater.from(context).inflate(R.layout.tab\_item, null);   TextView tv= (TextView) view.findViewById(R.id.textView);   tv.setText(tabTitles[position]);   ImageView img = (ImageView) view.findViewById(R.id.imageView);   img.setImageResource(imageResId[position]);   return view;  } |

  重新设置点击事件:

[?](http://www.jb51.net/article/101912.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | viewPager.setAdapter(pagerAdapter);   tabLayout.setupWithViewPager(viewPager);   for (int i = 0; i < tabLayout.getTabCount(); i++) {    TabLayout.Tab tab = tabLayout.getTabAt(i);    if (tab != null) {     tab.setCustomView(pagerAdapter.getTabView(i));     if (tab.getCustomView() != null) {      View tabView = (View) tab.getCustomView().getParent();      tabView.setTag(i);      tabView.setOnClickListener(mTabOnClickListener);     }    }   }   viewPager.setCurrentItem(1); |

以上所述是小编给大家介绍的TabLayout用法详解及自定义样式，希望对大家有所帮助，如果大家有任何疑问请给我留言，小编会及时回复大家的。在此也非常感谢大家对脚本之家网站的支持！

**16.** X**TabLayout自定义**

<http://blog.csdn.net/a1533588867/article/details/53810409>

<https://github.com/AndroidKun/XTabLayout>

**FlycoTabLayout**

<https://github.com/H07000223/FlycoTabLayout>