在开发中，我们有时会遇到顶部导航栏滑动切换页面的设计，如网易新闻。实现的方式有很多种，今天我们使用PagerSlidingTabStrip配合ViewPager实现顶部导航栏。  
  
效果图如下。  
  
  
PagerSlidingTabStrip是github上的一个开源项目，项目地址如下。  
<https://github.com/astuetz/PagerSlidingTabStrip>  
  
（一）PagerSlidingTabStrip的使用  
  
在使用之前，我们先来看一下PagerSlidingTabStrip中的自定义属性。

**[html]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **<declare-styleable** name="PagerSlidingTabStrip"**>**
2. **<attr** name="pstsIndicatorColor" format="color" **/>**
3. **<attr** name="pstsUnderlineColor" format="color" **/>**
4. **<attr** name="pstsDividerColor" format="color" **/>**
5. **<attr** name="pstsIndicatorHeight" format="dimension" **/>**
6. **<attr** name="pstsUnderlineHeight" format="dimension" **/>**
7. **<attr** name="pstsDividerPadding" format="dimension" **/>**
8. **<attr** name="pstsTabPaddingLeftRight" format="dimension" **/>**
9. **<attr** name="pstsScrollOffset" format="dimension" **/>**
10. **<attr** name="pstsTabBackground" format="reference" **/>**
11. **<attr** name="pstsShouldExpand" format="boolean" **/>**
12. **<attr** name="pstsTextAllCaps" format="boolean" **/>**
13. **</declare-styleable>**

各属性的详细介绍如下。  
pstsIndicatorColor：滑动条的颜色。  
pstsIndicatorHeight：滑动条的高度。  
pstsUnderlineColor：底部线条的颜色。（底部线条会填充屏幕宽度）  
pstsUnderlineHeight：底部线条的高度。  
pstsDividerColor：tab之间的竖直分割线的颜色。  
pstsDividerPadding：tab之间的竖直分割线，距离顶部和底部的距离，即它的paddingTop和paddingBottom。  
pstsTabPaddingLeftRight：单个tab内部的左间距和右间距，即它的paddingLeft和paddingRight。  
pstsTabBackground：单个tab的背景。  
pstsScrollOffset：当前tab滚动的偏移量。  
pstsShouldExpand：设置为ture，每个tab的权重一样，均分屏幕宽度，默认值false。  
pstsTextAllCaps：是否将tab中的字母转换成大写，默认值true。

下面，我们将PagerSlidingTabStrip使用到具体项目中。

首先将PagerSlidingTabStrip添加到工程module的gradle中。

**[html]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. dependencies {
2. compile 'com.astuetz:pagerslidingtabstrip:1.0.1'
3. }

接下来添加布局文件。

**[html]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **<com.astuetz.PagerSlidingTabStrip** xmlns:tab="http://schemas.android.com/apk/res-auto"
2. android:id="@+id/tab"
3. android:layout\_width="match\_parent"
4. android:layout\_height="wrap\_content"
5. android:background="#fafafa"
6. android:paddingBottom="10dp"
7. android:paddingTop="10dp"
8. android:textColor="#333333"
9. android:textSize="13sp"
10. tab:pstsDividerColor="@android:color/transparent"
11. tab:pstsIndicatorColor="#ed5955"
12. tab:pstsIndicatorHeight="2dp"
13. tab:pstsShouldExpand="true"
14. tab:pstsTabBackground="@android:color/transparent"
15. tab:pstsUnderlineColor="@android:color/transparent"**/>**

最后实现逻辑代码。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **package** net.csdn.blog.ruancoder;
3. **import** android.os.Bundle;
4. **import** android.support.v4.app.Fragment;
5. **import** android.support.v4.app.FragmentActivity;
6. **import** android.support.v4.view.ViewPager;
7. **import** android.view.Window;
9. **import** com.astuetz.PagerSlidingTabStrip;
11. **public** **class** MainActivity **extends** FragmentActivity {
12. @Override
13. **protected** **void** onCreate(Bundle savedInstanceState) {
14. **super**.onCreate(savedInstanceState);
15. requestWindowFeature(Window.FEATURE\_NO\_TITLE);
16. setContentView(R.layout.activity\_main);
17. **final** PagerSlidingTabStrip tabStrip = (PagerSlidingTabStrip) findViewById(R.id.tab);
18. ViewPager viewPager = (ViewPager) findViewById(R.id.viewpager);
20. Fragment[] fragments = {**new** NewsFragment(), **new** TechFragment(), **new** FinanceFragment(), **new** InternetFragment(), **new**
21. PhoneFragment()};
22. String[] titles = {"头条", "科技", "财经", "互联网", "手机"};
23. TabPagerAdapter adapter = **new** TabPagerAdapter(getSupportFragmentManager(), fragments, titles);
24. viewPager.setAdapter(adapter);
25. tabStrip.setViewPager(viewPager);
26. }
27. }

Activity中使用到的TabPagerAdapter类：

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **package** net.csdn.blog.ruancoder;
3. **import** android.support.v4.app.Fragment;
4. **import** android.support.v4.app.FragmentManager;
5. **import** android.support.v4.app.FragmentPagerAdapter;
7. **public** **class** TabPagerAdapter **extends** FragmentPagerAdapter {
8. **private** Fragment[] mFragments;
9. **private** String[] mTitles;
11. **public** TabPagerAdapter(FragmentManager fm, Fragment[] fragments, String[] titles) {
12. **super**(fm);
13. **this**.mFragments = fragments;
14. **this**.mTitles = titles;
15. }
17. @Override
18. **public** Fragment getItem(**int** position) {
19. **return** mFragments[position];
20. }
22. @Override
23. **public** **int** getCount() {
24. **return** mFragments.length;
25. }
27. @Override
28. **public** CharSequence getPageTitle(**int** position) {
29. **return** mTitles[position];
30. }
31. }

（二）PagerSlidingTabStrip的源码分析  
  
(1).类的声明。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **package** com.astuetz;
3. **public** **class** PagerSlidingTabStrip **extends** HorizontalScrollView {
4. }

PagerSlidingTabStrip继承自HorizontalScrillView，当tab数量较多超出屏幕时，可以横向滚动。  
  
(2).构造方法。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **public** PagerSlidingTabStrip(Context context, AttributeSet attrs, **int** defStyle) {
2. **super**(context, attrs, defStyle);
4. tabsContainer = **new** LinearLayout(context);
5. tabsContainer.setOrientation(LinearLayout.HORIZONTAL);
6. tabsContainer.setLayoutParams(**new** LayoutParams(LayoutParams.MATCH\_PARENT, LayoutParams.MATCH\_PARENT));
7. addView(tabsContainer);
9. // 这里省略了初始化各参数的代码
11. rectPaint = **new** Paint();
12. rectPaint.setAntiAlias(**true**);
13. rectPaint.setStyle(Style.FILL);
15. dividerPaint = **new** Paint();
16. dividerPaint.setAntiAlias(**true**);
17. dividerPaint.setStrokeWidth(dividerWidth);
19. defaultTabLayoutParams = **new** LinearLayout.LayoutParams(LayoutParams.WRAP\_CONTENT, LayoutParams.MATCH\_PARENT);
20. expandedTabLayoutParams = **new** LinearLayout.LayoutParams(0, LayoutParams.MATCH\_PARENT, 1.0f);
21. }

在构造方法中，先是创建了一个水平线性布局tabsContainer，并将其添加到PagerSlidingTabStrip中。因为PagerSlidingTabStrip继承自HorizontalScrillView，而HorizontalScrillView内部只能有一个子View，所以之后的tab都将添加到tabsContainer布局中。  
  
接下来创建了两个画笔rectPaint和dividerPaint，分别用来绘制滑动条和竖直分隔线。  
  
最后创建了两个布局参数LayoutParams，默认为defaultTabLayoutParams，自适应tab的宽度，以及expandedTabLayoutParams，tab均分屏幕宽度。当自定义属性pstsShouldExpand声明为true时，使用expandedTabLayoutParams，相反则使用defaultTabLayoutParams。  
  
  
(3).setViewPager(ViewPager pager)方法。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **public** **void** setViewPager(ViewPager pager) {
2. **this**.pager = pager;
4. **if** (pager.getAdapter() == **null**) {
5. **throw** **new** IllegalStateException("ViewPager does not have adapter instance.");
6. }
8. pager.setOnPageChangeListener(pageListener);
10. notifyDataSetChanged();
11. }

当我们获取到PagerSlidingTabStrip对象后，会调用它的setViewPager(ViewPager pager)方法。  
  
在该方法内部，为ViewPager对象添加了滑动监听PageListener，并调用了notifyDataSetChanged()方法。  
  
  
(4).notifyDataSetChanged()方法。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **public** **void** notifyDataSetChanged() {
3. tabsContainer.removeAllViews();
5. tabCount = pager.getAdapter().getCount();
7. **for** (**int** i = 0; i < tabCount; i++) {
9. **if** (pager.getAdapter() **instanceof** IconTabProvider) {
10. addIconTab(i, ((IconTabProvider) pager.getAdapter()).getPageIconResId(i));
11. } **else** {
12. addTextTab(i, pager.getAdapter().getPageTitle(i).toString());
13. }
15. }
17. updateTabStyles();
19. getViewTreeObserver().addOnGlobalLayoutListener(**new** ViewTreeObserver.OnGlobalLayoutListener() {
21. @SuppressWarnings("deprecation")
22. @SuppressLint("NewApi")
23. @Override
24. **public** **void** onGlobalLayout() {
26. **if** (Build.VERSION.SDK\_INT < Build.VERSION\_CODES.JELLY\_BEAN) {
27. getViewTreeObserver().removeGlobalOnLayoutListener(**this**);
28. } **else** {
29. getViewTreeObserver().removeOnGlobalLayoutListener(**this**);
30. }
32. currentPosition = pager.getCurrentItem();
33. scrollToChild(currentPosition, 0);
34. }
35. });
37. }

方法内部，遍历ViewPager的Adapter中的标题，生成Tab并添加到tabsContainer布局中。然后将滑动条滚动到第一个Tab下。  
  
(5).OnPageChangeListener的实现类，PageListener类。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. **private** **class** PageListener **implements** ViewPager.OnPageChangeListener {
2. @Override
3. **public** **void** onPageScrolled(**int** position, **float** positionOffset, **int** positionOffsetPixels) {
4. currentPosition = position;
5. currentPositionOffset = positionOffset;
7. scrollToChild(position, (**int**) (positionOffset \* tabsContainer.getChildAt(position).getWidth()));
9. invalidate();
11. // ......
12. }
14. @Override
15. **public** **void** onPageScrollStateChanged(**int** state) {
16. **if** (state == ViewPager.SCROLL\_STATE\_IDLE) {
17. scrollToChild(pager.getCurrentItem(), 0);
18. }
19. // ......
20. }
22. @Override
23. **public** **void** onPageSelected(**int** position) {
24. // ......
25. }
26. }

PageListener类监听ViewPager的滑动。

这里重点是onPageScrolled()方法，当ViewPager在滑动过程中该方法不断被回调。在其中执行了2行代码，scrollToChild()，移动tab；invalidate()，触发View的onDraw()的调用。  
  
(6).核心方法onDraw()。

**[java]** [view plain](http://blog.csdn.net/ruancoder/article/details/51992469) [copy](http://blog.csdn.net/ruancoder/article/details/51992469)

1. @Override
2. **protected** **void** onDraw(Canvas canvas) {
3. **super**.onDraw(canvas);
5. **if** (isInEditMode() || tabCount == 0) {
6. **return**;
7. }
9. **final** **int** height = getHeight();
11. // draw indicator line
12. rectPaint.setColor(indicatorColor);
14. // 默认在当前被选中的tab底部绘制滑动条
15. View currentTab = tabsContainer.getChildAt(currentPosition);
16. **float** lineLeft = currentTab.getLeft();
17. **float** lineRight = currentTab.getRight();
19. // 如果正在滑动，在当前tab和下一个tab之间绘制滑动条
20. **if** (currentPositionOffset > 0f && currentPosition < tabCount - 1) {
22. View nextTab = tabsContainer.getChildAt(currentPosition + 1);
23. **final** **float** nextTabLeft = nextTab.getLeft();
24. **final** **float** nextTabRight = nextTab.getRight();
26. lineLeft = (currentPositionOffset \* nextTabLeft + (1f - currentPositionOffset) \* lineLeft);
27. lineRight = (currentPositionOffset \* nextTabRight + (1f - currentPositionOffset) \* lineRight);
28. }
30. // 绘制滑动条，滑动条宽度为lineRight - lineLeft，高度为indicatorHeight
31. canvas.drawRect(lineLeft, height - indicatorHeight, lineRight, height, rectPaint);
33. // 绘制底部线条，线条宽度为PagerSlidingTabStrip的宽度，高度为underlineHeight
34. rectPaint.setColor(underlineColor);
35. canvas.drawRect(0, height - underlineHeight, tabsContainer.getWidth(), height, rectPaint);
37. // 在每个tab的右侧，绘制竖直分割线，分割线宽度为dividerWidth，高度为height - dividerPadding \* 2
38. dividerPaint.setColor(dividerColor);
39. **for** (**int** i = 0; i < tabCount - 1; i++) {
40. View tab = tabsContainer.getChildAt(i);
41. canvas.drawLine(tab.getRight(), dividerPadding, tab.getRight(), height - dividerPadding, dividerPaint);
42. }
43. }

通过在onPageScrolled()回调方法中得到的currentPosition和currentPositionOffset，实时绘制滑动条的位置，实现滑动条跟随手势移动。  
  
最后附上完整工程下载链接。  
<http://download.csdn.net/detail/ruancoder/9582974>