**ActionBar Навигация табы и выпадающий список**

OCT 7TH, 2012 | [COMMENTS](http://android-helper.com.ua/actionbar-navigation/#disqus_thread)

Добрый день друзья.

Нашел сегодня интересную статью. Которая еще более детально описывает работу с actionbar.

Прошу к прочтению.

На прошлом уроке мы рассмотрели добавление элементов ActionBar. Пользователю эти элементы удобно использовать для совершения каких-либо операций в приложении. Например, в почтовой программе это может быть удаление письма, пересылка, ответ и т.д.

Теперь рассмотрим, как добавить элементы для навигации по приложению. Есть два типа элементов для навигации: табы и выпадающий список. Создадим простое приложение и реализуем в нем сначала один, затем второй способ.

Создадим проект:

**Project name**: P1081\_ActionBarNavigation  
**Build Target**: Android 4.1   
**Application name**: ActionBarItems  
**Package name**: ru.startandroid.develop.p1081actionbarnavigation  
**Create Activity**: MainActivity

Никакие строки не добавляем, основной layout не трогаем.

**Табы**

Кодим **MainActivity.java**:

import android.app.ActionBar;

import android.app.ActionBar.Tab;

import android.app.Activity;

import android.app.FragmentTransaction;

import android.os.Bundle;

import android.util.Log;

public class MainActivity extends Activity implements ActionBar.TabListener {

final String LOG\_TAG = "myLogs";

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

ActionBar bar = getActionBar();

bar.setNavigationMode(ActionBar.NAVIGATION\_MODE\_TABS);

Tab tab = bar.newTab();

tab.setText("tab1");

tab.setTabListener(this);

bar.addTab(tab);

tab = bar.newTab();

tab.setText("tab2");

tab.setTabListener(this);

bar.addTab(tab);

}

@Override

public void onTabReselected(Tab tab, FragmentTransaction ft) {

Log.d(LOG\_TAG, "reselected tab: " + tab.getText());

}

@Override

public void onTabSelected(Tab tab, FragmentTransaction ft) {

Log.d(LOG\_TAG, "selected tab: " + tab.getText());

}

@Override

public void onTabUnselected(Tab tab, FragmentTransaction ft) {

Log.d(LOG\_TAG, "unselected tab: " + tab.getText());

}

}

Смотрим, что нам нужно, чтобы включить табы.

В **onCreate** мы получаем доступ к ActionBar и устанавливаем для него режим навигации в [NAVIGATION\_MODE\_TABS](http://developer.android.com/reference/android/app/ActionBar.html#NAVIGATION_MODE_TABS). Далее идет добавление табов. Тут все несложно – создаем, пишем текст, присваиваем обработчика, добавляем в ActionBar.

Обработчиком для табов мы сделали MainActivity, оно реализует методы интерфейса [ActionBar.TabListener](http://developer.android.com/reference/android/app/ActionBar.TabListener.html):

[onTabReselected](http://developer.android.com/reference/android/app/ActionBar.TabListener.html) – выбран уже выбранный таб

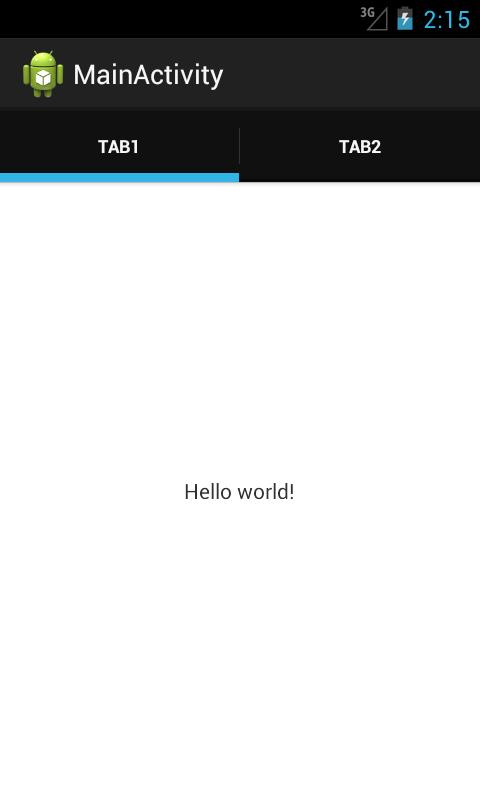
[onTabSelected](http://developer.android.com/reference/android/app/ActionBar.TabListener.html)– таб выбран

[onTabUnselected](http://developer.android.com/reference/android/app/ActionBar.TabListener.html)– таб более не выбран

В эти методы поместим запись в лог и посмотрим, как оно все там работает.

Все сохраняем и запускаем приложение.

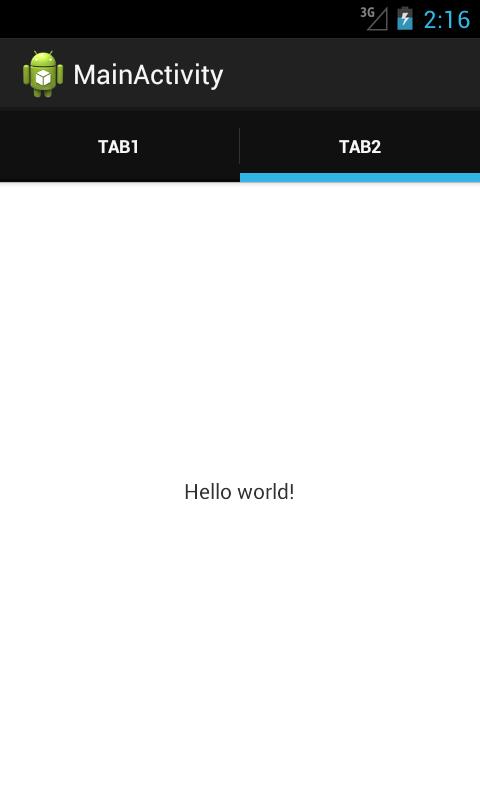
Видим табы



Сейчас выбран первый таб и лог сразу об этом сообщает:

\_selected tab: tab1\_

Нажмем на второй таб,



смотрим логи:

\_unselected tab: tab1

\_\_selected tab: tab\_2

Первый таб «развыбран», а второй выбран.

Еще раз нажмем на второй таб:

\_reselected tab: tab2\_

Второй таб перевыбран. Все логично и понятно.

Я, чтобы не перегружать урок, не стал реализовывать какие-либо операции по навигации в приложении. А, вообще, подразумевается, что в методах обработчика мы кодим операции с фрагментами. Нам даже любезно предоставляют объект **FragmentTransaction**для этих целей. При этом хелп предупреждает, что в этих методах нам не надо самим вызывать метод **commit**, а также мы не можем добавлять транзакцию в **BackStack**.

**Выпадающий список**

Теперь посмотрим, как работает навигация с выпадающим списком.

Перепишем **MainActivity.java**:

import android.app.ActionBar;

import android.app.Activity;

import android.os.Bundle;

import android.util.Log;

import android.widget.ArrayAdapter;

public class MainActivity extends Activity implements

ActionBar.OnNavigationListener {

String[] data = new String[] { "one", "two", "three" };

final String LOG\_TAG = "myLogs";

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

ActionBar bar = getActionBar();

bar.setNavigationMode(ActionBar.NAVIGATION\_MODE\_LIST);

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

android.R.layout.simple\_spinner\_item, data);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

bar.setListNavigationCallbacks(adapter, this);

}

@Override

public boolean onNavigationItemSelected(int itemPosition, long itemId) {

Log.d(LOG\_TAG, "selected: position = " + itemPosition + ", id = "

+ itemId + ", " + data[itemPosition]);

return false;

}

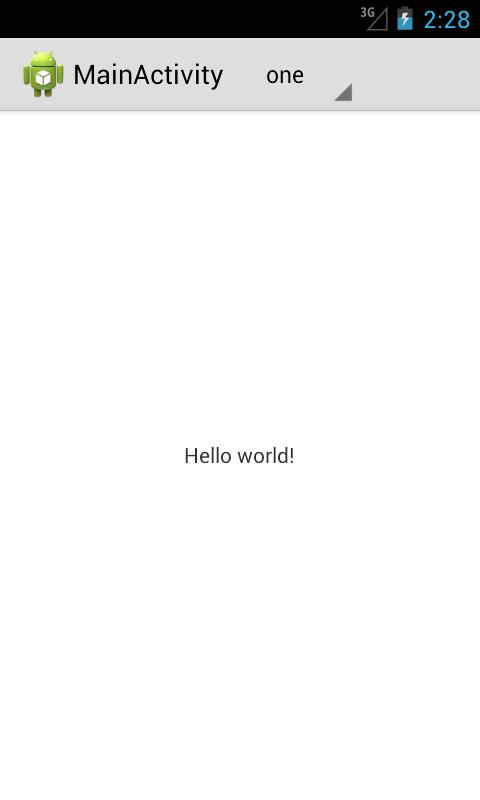
}

В **onCreate** получаем ActionBar и включаем ему режим навигации [NAVIGATION\_MODE\_LIST](http://developer.android.com/reference/android/app/ActionBar.html#NAVIGATION_MODE_LIST). Для выпадающего списка необходимо создать адаптер, реализующий SpinnerAdapter. Пусть это будет ArrayAdapter. При создании используем массив из трех строк. Далее вызываем метод [setListNavigationCallbacks](http://developer.android.com/reference/android/app/ActionBar.html" \l "setListNavigationCallbacks(android.widget.SpinnerAdapter,%20android.app.ActionBar.OnNavigationListener)), в который передаем адаптер и обработчик.

Обработчиком у нас снова является Activity, реализует метод [onNavigationItemSelected](http://developer.android.com/reference/android/app/ActionBar.OnNavigationListener.html#onNavigationItemSelected(int,%20long)) интерфейса [ActionBar.OnNavigationListener](http://developer.android.com/reference/android/app/ActionBar.OnNavigationListener.html). Этот метод дает нам позицию и id выбранного из списка элемента. Выводим в лог эту инфу и соответствующий элемент массива.

В манифесте я прописал для MainActivity атрибут темы: android:theme=“@android:style/Theme.Holo.Light”. Иначе был темный текст на темном фоне.

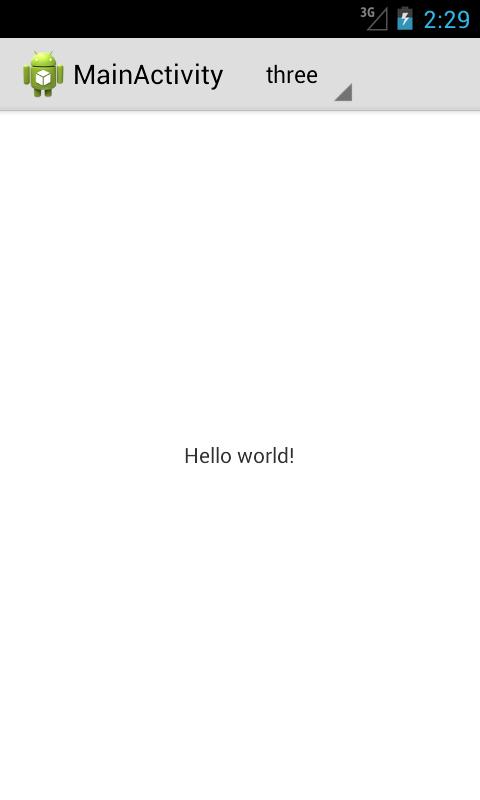
Все сохраняем и запускаем.



Первый элемент сразу выбран. В логах видим:

\_selected: position = 0, id = 0, one\_

Выберем какой-нить другой элемент из списка



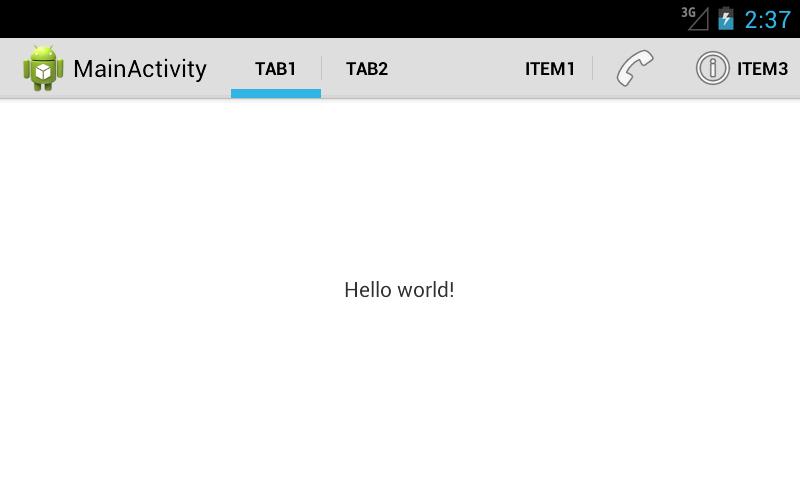
В логах:

\_selected: position = 2, id = 2, three\_

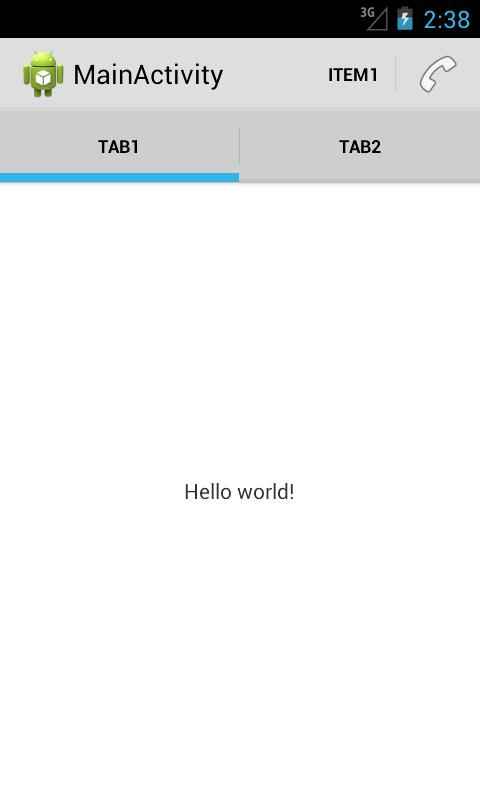
Снова все логично и понятно.

Для обоих видов навигации мы всегда можем получить текущий выбранный элемент с помощью метода [getSelectedNavigationIndex](http://developer.android.com/reference/android/app/ActionBar.html" \l "getSelectedNavigationIndex()). А для навигации с табами есть также метод [getSelectedTab](http://developer.android.com/reference/android/app/ActionBar.html" \l "getSelectedTab()), возвращающий текущий выбранный таб.

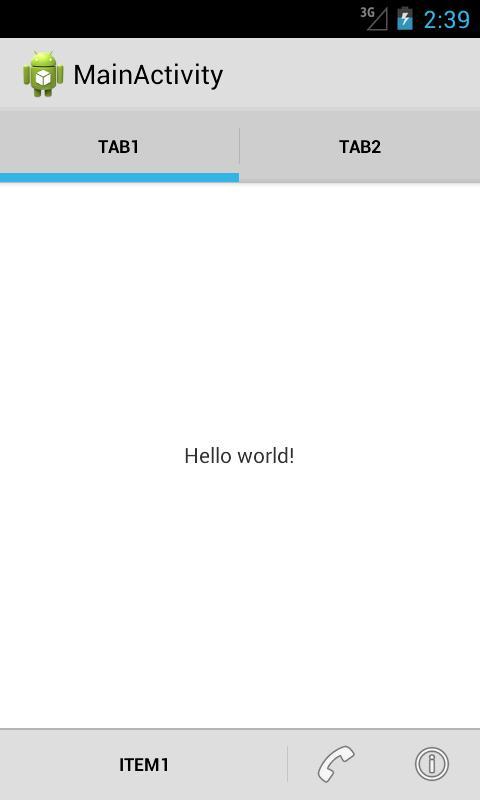
Разумеется, что вместе с навигацией мы можем добавлять в ActionBar и обычные элементы, которые были рассмотрены на прошлом уроке.



При этом, если места будет недостаточно, то ActionBar разделит все свои элементы на две полосы.



Начиная с четвертой версии Андроид, можно использовать атрибут **uiOptions** для Activity или Application в манифесте. Если присвоить ему значение**splitActionBarWhenNarrow**, то результат при нехватке места получится такой:



Элементы ушли вниз. Правда, при этом почему-то перестает работать withText в showAsAction.

Статья взята из сайта [startandroid.ru](http://startandroid.ru/uroki/vse-uroki-spiskom/178-urok-108-android-3-actionbar-navigatsija-taby-i-vypadajuschij-spisok.html)

Если вам понравилась статья, оставляйте свои комментарии. И не забывайте рассказать друзьям о нас.