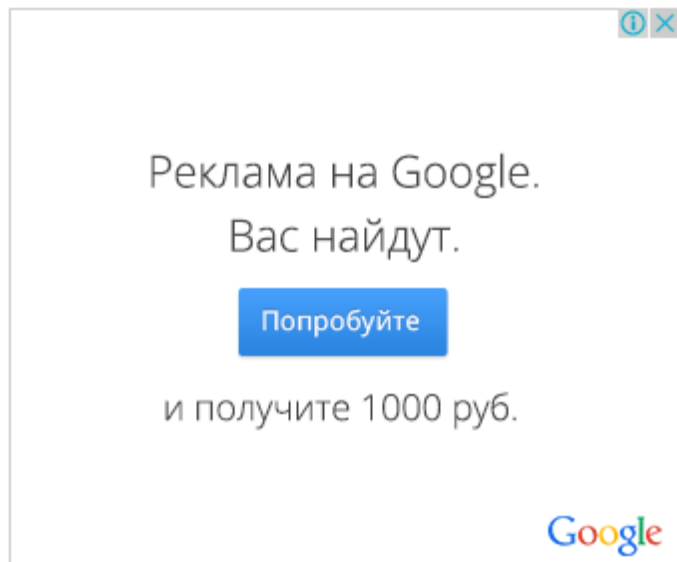


# Navigation Drawer — простейший пример

admin | 10.05.2014

Комментарии (8)



**Navigation Drawer** — это «выпадающее меню», которое появляется, когда пользователь кликает на иконку приложения в Action Bar в левом верхнем углу. Такое меню заслоняет лишь часть экрана, как бы «налаживаясь» сверху на левую часть активности. В выпадающем списке отображаются опции, позволяющие быстро перейти в нужную часть приложения. Паттерн проектирования Navigation Drawer улучшает юзабилити, поэтому он активно используется сегодня во многих популярных приложениях.



Создадим простейшее меню из трех опций, при клике на которые будет отображаться новое активити. По умолчанию (при запуске приложения) будет отображаться активити для первой опции.

Первое, что мы сделаем — это создадим файлы layout. Нам потребуется основной layout с использованием фрагмента и еще три для «заполнения» самого фрагмента (будут соответствовать трем пунктам меню). Обратите внимание, что мы будем использовать Support Library для поддержки устройств со старыми версиями Андроида.

Код **res/layout/activity\_main.xml**:

```
1 <android.support.v4.widget.DrawerLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/drawer_layout"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <!-- The main content view -->
8     <FrameLayout
9         android:id="@+id/content_frame"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent" />
12
13    <!-- The navigation drawer -->
14    <ListView android:id="@+id/left_drawer"
15        android:layout_width="240dp"
16        android:layout_height="match_parent"
17        android:layout_gravity="start"
18        android:choiceMode="singleChoice"
19        android:divider="@color/list_divider"
20        android:dividerHeight="1dp"
21        android:listSelector="@drawable/list_selector"
22        android:background="@color/list_background"/>
23
24 </android.support.v4.widget.DrawerLayout>
```

Само меню Navigation Drawer будет заполняться в ListView, которое первоначально будет скрыто и появляться при клике на иконку приложения в Action Bar. Мы будем использовать также паттерн **listSelector** для того, чтобы изменять бэкграунд для пунктов меню в различных состояниях (например, изменение фона при нажатии на пункт).

Макет для пункта меню Navigation Drawer — **res/layout/drawer\_list\_item.xml**:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/label"
4     android:layout_width="fill_parent"
5     android:layout_height="wrap_content"
6     android:text="test"
7     android:paddingLeft="10dp"
8     android:textColor="@color/drawer_item_color"
9     android:textSize="20sp">
10 </TextView>
```

Добавим в ресурсы **drawable** (папка **res**) макет **list\_selector.xml**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <item android:drawable="@drawable/list_item_bg_pressed" android:state_pressed="true"/>
5     <item android:drawable="@drawable/list_item_bg_pressed" android:state_activated="true"/>
6     <item android:drawable="@drawable/list_item_bg_normal" />
7
8 </selector>
```

Здесь мы определяем разные состояния пунктов и соответствующий им макет. Соответственно, добавим также в ресурсы **drawable** макеты:

### **list\_item\_bg\_pressed.xml**

```
1 <shape xmlns:android="http://schemas.android.com/apk/res/android"
2     android:shape="rectangle">
3     <gradient
4         android:startColor="@color/list_background_pressed"
5         android:endColor="@color/list_background_pressed"
6         android:angle="90" />
7 </shape>
```

### **list\_item\_bg\_normal.xml**

```
1 <shape xmlns:android="http://schemas.android.com/apk/res/android"
2     android:shape="rectangle">
3     <gradient
4         android:startColor="@color/list_background"
5         android:endColor="@color/list_background"
6         android:angle="90" />
7 </shape>
```

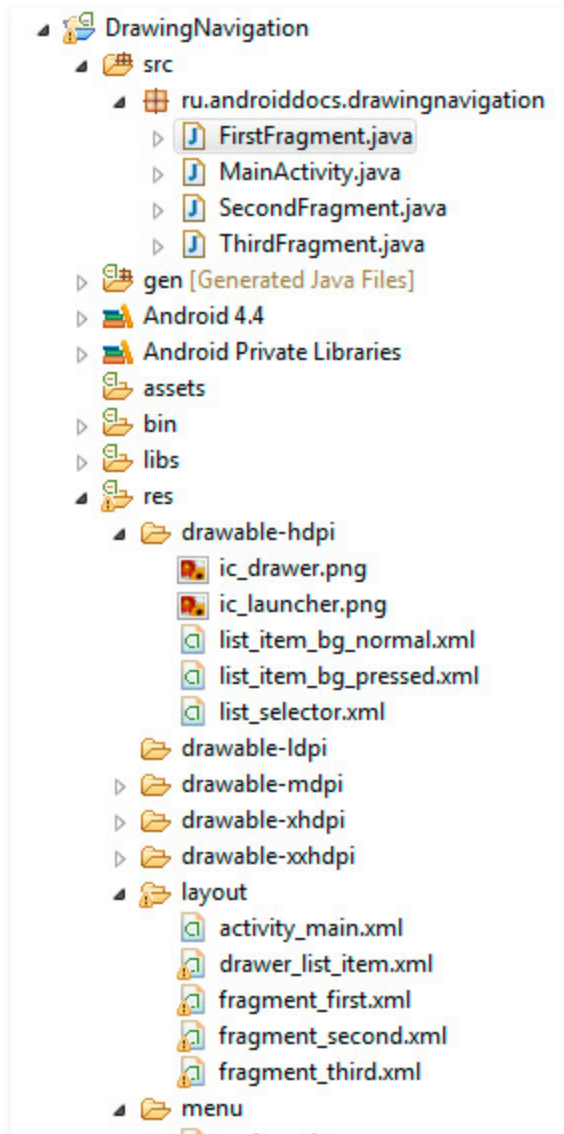
Нам требуется три **layout** для каждого фрагмента и **java**-файл для каждого из них. Я приведу код для одного фрагмента, для второго и третьего вы можете создать нужные файлы самостоятельно.

## res/layout/fragment\_first.xml:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <TextView
7         android:id="@+id/txtLabel"
8         android:layout_width="wrap_content"
9         android:layout_height="wrap_content"
10        android:layout_centerInParent="true"
11        android:textSize="16sp"
12        android:text="Main View"/>
13
14     <ImageView android:layout_width="wrap_content"
15        android:layout_height="wrap_content"
16        android:layout_below="@id/txtLabel"
17        android:src="@drawable/ic_launcher"
18        android:layout_centerHorizontal="true"
19        android:layout_marginTop="10dp"/>
20
21 </RelativeLayout>
```

## java-код для данного фрагмента (в папке src) **FirstFragment.java**:

```
1 package ru.androiddocs.drawingnavigation;
2
3 import android.app.Fragment;
4 import android.os.Bundle;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8
9 public class FirstFragment extends Fragment {
10
11     public FirstFragment() {}
12
13     @Override
14     public View onCreateView(LayoutInflater inflater, ViewGroup container,
15         Bundle savedInstanceState) {
16
17         View rootView = inflater.inflate(R.layout.fragment_first, container, false);
18
19         return rootView;
20     }
21 }
```



Вы уже обратили внимание, что мы используем ссылки на строковые ресурсы и ресурсы цветов:

### colors.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <color name="list_background">#000</color>
4     <color name="list_background_pressed">#ccc</color>
5     <color name="drawer_item_color">#fff</color>
6     <color name="list_divider">#5e5e5e</color>
7 </resources>
```

## strings.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">DrawingNavigation</string>
4   <string name="action_settings">Settings</string>
5   <string-array name="views_array">
6     <item>Первый экран</item>
7     <item>Второй экран</item>
8     <item>Третий экран</item>
9   </string-array>
10  <string name="menu">Меню</string>
11 </resources>
```

Также можно было бы вынести размерности и другие детали в ресурсы (dimens и styles), но здесь я этого не делал.

Добавьте иконку для Navigation Drawer, которая будет отображаться слева в Action Bar.

ic\_drawer.png (здесь разных размеров):



Добавим **MainActivity.java**

```
1 package ru.androiddocs.drawingnavigation;
2
3 import android.os.Bundle;
4
5 import android.app.Activity;
6 import android.app.Fragment;
7 import android.content.res.Configuration;
8 import android.support.v4.app.ActionBarDrawerToggle;
9 import android.app.ActionBar;
10 import android.support.v4.widget.DrawerLayout;
11 import android.util.Log;
12
13 import android.view.Menu;
14 import android.view.MenuItem;
15 import android.view.View;
16 import android.widget.AdapterView;
17 import android.widget.AdapterView.OnItemClickListener;
18 import android.widget.AdapterView.OnItemClickListener;
19
20
```

```

21
22 public class MainActivity extends Activity {
23
24     private DrawerLayout myDrawerLayout;
25     private ListView myDrawerList;
26     private ActionBarDrawerToggle myDrawerToggle;
27
28     // navigation drawer title
29     private CharSequence myDrawerTitle;
30     // used to store app title
31     private CharSequence myTitle;
32
33     private String[] viewsNames;
34
35     @Override
36     public void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.activity_main);
39
40         myTitle = getTitle();
41         myDrawerTitle = getResources().getString(R.string.menu);
42
43         // load slide menu items
44         viewsNames = getResources().getStringArray(R.array.views_array);
45         myDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
46         myDrawerList = (ListView) findViewById(R.id.left_drawer);
47
48         myDrawerList.setAdapter(new ArrayAdapter<String>(this, R.layout.drawer_list_item, view
49
50         // enabling action bar app icon and behaving it as toggle button
51         ActionBar actionBar = getActionBar();
52         actionBar.setDisplayHomeAsUpEnabled(true);
53
54         myDrawerToggle = new ActionBarDrawerToggle(this, myDrawerLayout,
55             R.drawable.ic_drawer, //nav menu toggle icon
56             R.string.app_name, // nav drawer open - description for accessibility
57             R.string.app_name // nav drawer close - description for accessibility
58         ){
59             public void onDrawerClosed(View view) {
60                 getActionBar().setTitle(myTitle);
61                 // calling onPrepareOptionsMenu() to show action bar icons
62                 invalidateOptionsMenu();
63             }
64
65             public void onDrawerOpened(View drawerView) {
66                 getActionBar().setTitle(myDrawerTitle);
67                 // calling onPrepareOptionsMenu() to hide action bar icons
68                 invalidateOptionsMenu();
69             }
70         };
71         myDrawerLayout.setDrawerListener(myDrawerToggle);
72
73         if (savedInstanceState == null) {
74             // on first time display view for first nav item
75             displayView(0);
76         }

```



77

78

```
myDrawerList.setOnItemClickListener(new DrawerItemClickListener());
```

79

```
}
```

```
80
81     private class DrawerItemClickListener implements ListView.OnItemClickListener {
82         @Override
83         public void onItemClick(
84             AdapterView<?> parent, View view, int position, long id
85         ) {
86             // display view for selected nav drawer item
87             displayView(position);
88         }
89     }
90
91     private void displayView(int position) {
92         // update the main content by replacing fragments
93         Fragment fragment = null;
94         switch (position) {
95             case 0:
96                 fragment = new FirstFragment();
97                 break;
98             case 1:
99                 fragment = new SecondFragment();
100                break;
101             case 2:
102                 fragment = new ThirdFragment();
103                 break;
104             default:
105                 break;
106         }
107
108         if (fragment != null) {
109             android.app.FragmentManager fragmentManager = getFragmentManager();
110             fragmentManager.beginTransaction()
111                 .replace(R.id.content_frame, fragment).commit();
112
113             // update selected item and title, then close the drawer
114             myDrawerList.setItemChecked(position, true);
115             myDrawerList.setSelection(position);
116             setTitle(viewsNames[position]);
117             myDrawerLayout.closeDrawer(myDrawerList);
118         } else {
119             // error in creating fragment
120             Log.e("MainActivity", "Error in creating fragment");
121         }
122     }
123
124     @Override
125     public boolean onCreateOptionsMenu(Menu menu) {
126         getMenuInflater().inflate(R.menu.main, menu);
127         return true;
128     }
```

```
129
130     @Override
131     public boolean onOptionsItemSelected(MenuItem item) {
132         // toggle nav drawer on selecting action bar app icon/title
133         if (myDrawerToggle.onOptionsItemSelected(item)) {
134             return true;
135         }
136         // Handle action bar actions click
137         switch (item.getItemId()) {
138             case R.id.action_settings:
139                 return true;
140             default:
141                 return super.onOptionsItemSelected(item);
142         }
143     }
144
145     /**
146     * Called when invalidateOptionsMenu() is triggered
147     */
148     @Override
149     public boolean onPrepareOptionsMenu(Menu menu) {
150         // if navigation drawer is opened, hide the action items
151         boolean drawerOpen = myDrawerLayout.isDrawerOpen(myDrawerList);
152         menu.findItem(R.id.action_settings).setVisible(!drawerOpen);
153         return super.onPrepareOptionsMenu(menu);
154     }
155
156     @Override
157     public void setTitle(CharSequence title) {
158         myTitle = title;
159         getActionBar().setTitle(myTitle);
160     }
161
162     /**
163     * When using the ActionBarDrawerToggle, you must call it during
164     * onCreate() and onConfigurationChanged()...
165     */
166     @Override
167     protected void onCreate(Bundle savedInstanceState) {
168         super.onCreate(savedInstanceState);
169         // Sync the toggle state after onRestoreInstanceState has occurred.
170         myDrawerToggle.syncState();
171     }
172
173     @Override
174     public void onConfigurationChanged(Configuration newConfig) {
175         super.onConfigurationChanged(newConfig);
176         // Pass any configuration change to the drawer toggls
177         myDrawerToggle.onConfigurationChanged(newConfig);
178     }
179 }
```

В методе **onCreate()** мы инициализируем нужные переменные, получаем доступ до ресурсов.

В строчке

```
1 myDrawerList.setAdapter(new ArrayAdapter<String>(this, R.layout.drawer_list_item, viewsNames))
```

задаем список пунктов меню для Navigation Drawer с помощью Array-адаптера. В качестве обработчика клика по пунктам меню используется класс **DrawerItemClickListener**.

При первой загрузке приложения мы инициализируем первое view, т.к. пользователь еще ничего не успел выбрать:

```
1 if (savedInstanceState == null) {  
2     displayView(0);  
3 }
```

Метод **displayView(int position)** получает в параметрах id нужного view и вызывает соответствующий код для заполнения фрагмента.

Выглядит это все примерно так.

