

Android Navigation Drawer(Sliding Menu) Kullanımı

Navigation drawer ekranın sol tarafından bulunan, uygulamanın ana navigasyon seçeneklerini içeren bir paneldir. Çoğu zaman gizlidir, fakat kullanıcı parmağını ekranın sol kenarından sağa doğru kaydıracağı zaman veya uygulamanın en üstünde bulunan uygulama ikonuna bastığında ortaya çıkartabilir. Sliding menü olarak geçen bu yapı **Android 3** ile gelmiştir. Navigation drawer ile genelde ekran boyutu küçük olan cihazlara yönelik geliştirilen uygulamalarda çok rahat kullanım sağlamaktadır. Çünkü bu paneli dilediğimiz zaman **açıp gizleyebiliriz**.

Android 3.0(API Level 11) gelen bu özelliği **Android 2.1 (API Level 7)** ve üst versiyonlarda kullanabilmek için gereken işlemleri detaylı bir şekilde inceleyelim.

Öncelikle şunu hatırlatmakta fayda var: Android'te yeni versiyonlarda çıkmış bir özelliğin önceki versiyonlarda kullanılabilmesi için **destek kütüphaneleri(support library)** kullanılmaktadır. Bundan dolayı bu uygulamamızda biz de support library kullanacağız.

Şimdi uygulamayı geliştirmeye başlayalım:

res/layout/activity_main.xml dosyası

```
1 <android.support.v4.widget.drawerlayout xmlns:android="
2     android:id="@+id/drawer_layout"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5
6     <EditText
7         android:id="@+id/txtMain"
8         android:inputType="textMultiLine"
9         android:minLines="6"
10        android:gravity="top|left"
11        android:layout_height="wrap_content"
12        android:layout_width="fill_parent"
13        android:scrollbars="vertical"/>
14
15    <ListView
16        android:id="@+id/left_drawer"
17        android:layout_width="240dp"
18        android:layout_height="match_parent"
19        android:layout_gravity="start"
20        android:choiceMode="singleChoice"
21        android:divider="@android:color/transparent"
22        android:dividerHeight="0dp"
23        android:background="#111"/>
24 </android.support.v4.widget.DrawerLayout >
```

XML dosyalarında sıralama **z-index** sıralamasını ima eder. **activity_main** dosyası bir xml dosyası olduğu için ikinci sırada bulunan **ListView EditText**'in **z-index'inden** daha büyüktür. Bundan dolayı **ListView** açılıp kapandığında **EditText ListView** ekranda gösterildiğinde altta kalacaktır, **ListView** ise üstte kalacaktır.

android:layout_gravity="start" ile **DrawerLayout'un** soldan sağa doğru okunan dillerde sliding drawer'in sol tarafta, sağdan sola doğru okunan dillerde ise sağdan sola doğru açılır. Türkçe veya İngilizce dilinde cihazınızı kullanıyorsanız soldan sağa doğru, Arapça olarak kullanıyorsanız sağdan sola doğru açılır.

android:layout_width="240dp" ile **DrawerLayout'un** kaç dp genişliğe sahip olacağını ifade ediyoruz.

res/layout/drawer_list_item.xml dosyası

```
1 <TextView xmlns:android="
2     android:id="@android:id/text1"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:textAppearance="@android:style/TextAppearance.Medium"
6     android:gravity="center_vertical"
7     android:paddingLeft="16dp"
8     android:paddingRight="16dp"
9     android:textColor="#fff"
10    android:background="@drawable/list_selector_activated_holo_dark"
11    android:minHeight="48dip"/>
```

Bu dosyada bulunan **TextView** nesnesi **Activity** sınıfının onCreate metodunda **ArrayAdapter** ile kullanılacaktır. Navigation drawer uygulamamızda sliding menüdeki elemanlar textview olarak gösterilmesi bu dosya ile sağlanacaktır.

android:background = "@drawable/list_selector_activated_holo_dark" değeri ile sliding menünün arka plan renginin ne olacağını ifade ediyoruz. Bu dosyayı projenize eklediğiniz zaman android:background değeri altı çizili olarak gösteriliyorsa, **res/drawable-hdpi** dosyasına **bu resmi (/sources/docs/android/list_selector_activated_holo_dark.9.png)** indirip kopyalamanız gerekir.

Eğer Android 3.1 ve üstü sürümler için uygulama geliştiriyorsanız aşağıdaki kodları kullanınız:

```
1 <TextView xmlns:android="
2     android:id="@android:id/text1"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:textAppearance="@android:attr/textAppearanceListItemSmall"
6     android:gravity="center_vertical"
7     android:paddingLeft="16dp"
8     android:paddingRight="16dp"
9     android:textColor="#fff"
10    android:background="?android:attr/activatedBackgroundIndicator"
11    android:minHeight="?android:attr/listPreferredItemHeightSmall"/>
```

AndroidManifest.xml dosyası

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="
3     package="mucayufa.android.navigationDrawer.android7"
4     android:versionCode="1"
5     android:versionName="1.0">
6     <uses-sdk android:minSdkVersion="7"/>
7     <application android:label="@string/app_name" android:icon="@drawable/icon">
8         <activity android:name=".MainActivity"
9             android:label="@string/app_name"
10            android:theme="@style/Theme.AppCompat.Light.DarkActionBar"
11            android:launchMode="singleTop"> <!--alt sinifta up ikonuna basildiginda ana activity
12 sinifina doner-->
13             <intent-filter >
14                 <action android:name="android.intent.action.MAIN"/>
15                 <category android:name="android.intent.category.LAUNCHER"/>
16             </intent-filter >
17         </activity >
18     </application >
19 </manifest >
```

Görüldüğü gibi android **minSdkVersion 7** olarak belirtildi. Burada dikkat edilmesi gereken en önemli husus şudur: android:theme ifadesinin aldığı değer android.support_compatibility-v7-appcompat_19 kütüphanesinin res/values/themes.xml dosyası içinde tanımlanmıştır.

Android 3 ve üstü sürümlerde **android:theme** tanımlamaya gerek yoktur. Ayrıca **minSdkVersion 11** olarak değiştirilmelidir.

Android **support** kütüphanelerini eklemek için **maven** kullanıcıları aşağıdaki dependency'leri eklemelidir:

```
1 <dependency>
2     <groupId>android.support</groupId>
3     <artifactId>compatibility-v4</artifactId>
4     <version>19</version>
5 </dependency>
6 <dependency>
7     <groupId>android.support</groupId>
8     <artifactId>compatibility-v7-appcompat</artifactId>
9     <version>19</version>
10    <type>apklib</type>
11 </dependency>
12 <dependency>
13     <groupId>android.support</groupId>
14     <artifactId>compatibility-v7-appcompat</artifactId>
15     <version>19</version>
16     <type>jar</type>
17 </dependency>
```

Bu dependency'ler mvnrepository sitesinde bulunmadığı için **maven-android-sdk-deployer-master (/sources/codes/android/maven-android-sdk-deployer-master.zip)** dosyasını indirip açtıktan sonra o dizinde

```
1 | mvn install -P 2.2
```

komutunu çalıştırmaları gerekmektedir(Burada 2.2 Android versiyonunu temsil etmektedir. Örneğin Android 4.3 versiyonu için uygulama geliştiriyorsanız 2.2 yerine 4.3 yazmanız yeterlidir). Daha sonra bu dizinde bulunan extras/compatibility-v7-appcompat dizinine giderek şu komutu çalıştırmalıyız:

```
1 | mvn clean install
```

Tüm bu işlemler tamamlandıktan sonra maven kullanıcıları sorunsuz bir şekilde compatibility paketlerini kullanabilir. **Eclipse Ide** ile proje geliştiren kullanıcılar ise sırasıyla şu işlemleri yapmalıdır:

```
1 | File >> Import... >> Android (folder) >> Existing Android Code Into Workspace
2 | Browse to your sdk directory <sdk>/extras/android/support/v7/appcompat/
3 | Click Finish
4 | Right-click on the application -> properties...
5 | Select Android from the left menu
6 | Under Library section Click Add... button
7 | Select android-support-v7-appcompat
8 | Click OK
```

MainActivity Sınıfı

```
1 | package mucayufa.android.navigationDrawer.android7;
2 |
3 | import android.content.res.Configuration;
4 | import android.os.Bundle;
5 | import android.support.v4.app.ActionBarDrawerToggle;
6 | import android.support.v4.view.GravityCompat;
7 | import android.support.v4.widget.DrawerLayout;
8 | import android.support.v7.app.ActionBarActivity;
9 | import android.view.MenuItem;
10 | import android.view.View;
11 | import android.widget.*;
12 |
13 | public class MainActivity extends ActionBarActivity {
14 |     private DrawerLayout mDrawerLayout;
15 |     private ListView mDrawerList;
16 |     private ActionBarDrawerToggle mDrawerToggle;
17 |
18 |     private CharSequence mDrawerTitle;
19 |     private CharSequence mTitle;
20 |     private String[] mOperationList;
21 |     private EditText mTxtMain;
22 |
23 |
24 |     @Override
25 |     protected void onCreate(Bundle savedInstanceState) {
26 |         super.onCreate(savedInstanceState);
27 |         setContentView(R.layout.activity_main);
28 |
29 |         mTxtMain = (EditText) findViewById(R.id.txtMain);
30 |         mTitle = mDrawerTitle = getTitle();
31 |         mOperationList = getResources().getStringArray(R.array.operation_list);
32 |         mDrawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
33 |         mDrawerList = (ListView) findViewById(R.id.left_drawer);
34 |
35 |         // set a custom shadow that overlays the main content when the drawer opens
36 |         mDrawerLayout.setDrawerShadow(R.drawable.drawer_shadow, GravityCompat.START);
37 |         // set up the drawer's list view with items and click listener
38 |         mDrawerList.setAdapter(new ArrayAdapter<String>(this ,
39 |             R.layout.drawer_list_item, mOperationList));
40 |         mDrawerList.setOnItemClickListener(new DrawerItemClickListener());
41 |
42 |         // enable ActionBar app icon to behave as action to toggle nav drawer
43 |         getSupportActionBar().setDisplayHomeAsUpEnabled(true );
44 |         getSupportActionBar().setHomeButtonEnabled(true );
45 |
46 |         //ActionBarDrawerToggle sinifi sliding drawer ile
47 |         //action bar uygulama ikonu arasinda uyumlu bir entegrasyon saglar
48 |         mDrawerToggle = new ActionBarDrawerToggle(
49 |             this , //uzerinde islem yapacagi Activity sinifi
50 |             mDrawerLayout, //DrawerLayout nesnesi
51 |             R.drawable.ic_drawer, // 'Up' caret ile yer degisecek ikon
52 |             R.string.drawer_open, // "open drawer" erisim icin tanim
53 |             R.string.drawer_close // "close drawer" erisim icin tanim
54 |         ) {
55 |             public void onDrawerClosed(View view) {
56 |                 getSupportActionBar().setTitle(mTitle);
57 |                 invalidateOptionsMenu(); // onPrepareOptionsMenu() metodunu cagrir
58 |             }
59 |         }
```

```

59
60     public void onDrawerOpened(View drawerView) {
61         getSupportActionBar().setTitle(mDrawerTitle);
62         invalidateOptionsMenu(); // onPrepareOptionsMenu() metodunu cagrir
63     }
64 };
65 mDrawerLayout.setDrawerListener(mDrawerToggle);
66
67     if (savedInstanceState == null ) {
68         selectItem(0);
69     }
70
71 }
72 //The click listner for ListView in the navigation drawer
73 private class DrawerItemClickListener implements ListView.OnItemClickListener {
74     @Override
75     public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
76         selectItem(position);
77     }
78 }
79
80 /**
81  * Navigation Drawer'da App icona basildiginda acilmasi icin gerekli metoddur
82  * @param item
83  * @return
84  */
85 @Override
86 public boolean onOptionsItemSelected(MenuItem item) {
87     //ActionBarDrawerToggle'a meydana gelen event'i aktarir. Bu metoddan donen deger true olursa
88     //uygulama ikonunun tiklama event'i tarafından handle edilir
89     if (mDrawerToggle.onOptionsItemSelected(item)) {
90         return true ;
91     }
92     return super .onOptionsItemSelected(item);
93 }
94 private void selectItem(int position) {
95     String toBeAdded = "";
96     switch (position) {
97         case 0:
98             toBeAdded="Hello World";
99             break ;
100        case 1:
101            toBeAdded = "";
102            break ;
103    }
104    mTxtMain.setText(toBeAdded);
105    // update selected item and title, then close the drawer
106    mDrawerList.setItemChecked(position, true );
107    setTitle(mOperationList[position]);
108    mDrawerLayout.closeDrawer(mDrawerList);
109 }
110
111 @Override
112 public void setTitle(CharSequence title) {
113     mTitle = title;
114     getSupportActionBar().setTitle(mTitle);
115 }
116
117 /**
118  * When using the ActionBarDrawerToggle, you must call it during
119  * onCreate() and onConfigurationChanged()...
120  */
121 @Override
122 protected void onCreate(Bundle savedInstanceState) {
123     super .onCreate(savedInstanceState);
124     // Sync the toggle state after onRestoreInstanceState has occurred.
125     mDrawerToggle.syncState();
126 }
127
128 @Override
129 public void onConfigurationChanged(Configuration newConfig) {
130     super .onConfigurationChanged(newConfig);
131     // Pass any configuration change to the drawer toggls
132     mDrawerToggle.onConfigurationChanged(newConfig);
133 }
134 }
135 }

```

Dikkat ederseniz **MainActivity** sınıfı **Activity** sınıfını extend etmemiş, bunun yerine **ActionBarActivity** sınıfını extend etmiş. Çünkü **onCreate()** metodu içerisinde bulunan

```
1 //enable ActionBar app icon to behave as action to toggle nav drawer
2 getSupportActionBar().setDisplayHomeAsUpEnabled(true );
3 getSupportActionBar().setHomeButtonEnabled(true );
```

satırlardaki `getSupportActionBar()` metodu `ActionBarActivity` sınıfından gelmektedir. **API Level 11**'e göre uygulama geliştiriyor olsaydık `Activity` sınıfını `extend` ettikten sonra `getSupportActionBar()` metodu yerine `getActionBar()` metodu kullanmamız gerekecekti.