

Android Action Bar Kullanımı

ActionBar bir activity'nin en üstünde yer alır. Activity'nin başlığını, ikonu, bazı action elementlerini, view'leri ve diğer interaktif elementleri içerir. Ayrıca uygulamamız içerisinde navigasyonu da sağlar. Action bar Android **SDK 11** veya **üst** versiyonlarda çalışır. Kullanılan temayı değiştirerek action bar devre dışı bırakılabilir, fakat varsayılan Android temasında, action bar aktiftir. **SDK 11'den** önceki sürümlerde action bar yerine options menü kullanılır. Options menü, kullanıcı option butonuna tıkladığı zaman gösterilir. **Action bar options menüden daha kullanışlıdır**, çünkü action bar açık bir şekilde görülebilmesine karşın, options menü sadece options tuşuna basıldığında görülür. Kullanıcı uygulamamızın options menüye sahip olup olmadığını options butonuna basmadıkça bilemez.

Şimdi action bar uygulaması için gereken adımları sırasıyla inceleyelim:

res/layout/activity_main.xml dosyası

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mainLayout"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/txtMain"
        android:inputType="textMultiLine"

        android:minLines="6"
        android:gravity="top|left"
        android:layout_height="wrap_content"
        android:layout_width="fill_parent"
        android:scrollbars="vertical"
    />

</LinearLayout>
```

res/menu/main.xml dosyası

Bu dosya action bar'a eklemek istediğimiz elemanları eklememizi sağlar. Aşağıdaki kodta görüldüğü gibi her bir eleman **item** elementi ile eklenmektedir.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/action_content_replace"
        android:icon="@drawable/action_search"
        android:title="@string/action_content_replace"
        android:showAsAction="collapseActionView|ifRoom"
        android:actionViewClass="android.widget.SearchView" />

    <item android:id="@+id/action_keyword"
        android:title="@string/action_keyword"
        android:showAsAction="ifRoom"/>

    <item android:id="@+id/action_important"
        android:title="@string/action_important"
        android:showAsAction="ifRoom"/>

</menu>
```

İlk **item** elementi ile action bar'a arama yapmayı sağlayacak bir **SearchView** ekliyoruz. Burada `android:showAsAction="collapseActionView|ifRoom"` ifadesindeki **ActionView** özelliği, arama butonuna tıkladığında bir **text box** açması, diğer durumda ise **text box**'ı gizlemesidir. **ifRoom** değeri ise action bar da boş yer varsa **item** ile belirtilen elementimizin gösterilmesini, aksi takdirde gösterilmemesini sağlar. **collapseActionView** ve **ifRoom** değerleri arasındaki **|** karakteri, birden çok değer eklerken değerler arasına konur. Action bar'a eklenebilecek **item sayısı** android cihazının **density-independent width** özelliğine göre Android işletim sistemi tarafından otomatik olarak belirlenir. Ayrıca **item** sayısının kapladığı alan action bar'ın **yarısını geçemez**.

res/values/string.xml dosyası

```
<resources>
    <string name="app_name">Action Bar</string>
    <string name="action_content_replace">R</string>
    <string name="action_important">I</string>
    <string name="action_keyword">K</string>
    <string name="search_hint">Search Replace</string>
</resources>
```

AndroidManifest.xml dosyası

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="mucayufa.android.actionBar"
    android:versionCode="1"
    android:versionName="1.0">
<uses-sdk android:minSdkVersion="14"/>
<application android:label="@string/app_name" android:icon="@drawable/icon">
    <activity android:name=".MainActivity"
        android:label="@string/app_name">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>
</manifest>

```

MainActivity Sınıfı

```

package mucayufa.android.actionBar;

import android.app.Activity;
import android.app.SearchManager;
import android.content.Context;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.*;

public class MainActivity extends Activity implements SearchView.OnQueryTextListener {
    private CharSequence mTitle;
    private EditText mTxtMain;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mTxtMain = (EditText) findViewById(R.id.txtMain);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        //Action bar'a kendi menu elementlerimizi eklemek için MenuInflater nesnesi gereklidir
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.main, menu);

        //SearchView nesnesini kullanmak için gerekli kodlar
        SearchManager searchManager = (SearchManager) getSystemService(Context.SEARCH_SERVICE);
        SearchView searchView = (SearchView) menu.findItem(R.id.action_content_replace)
            .getActionView();
        searchView.setSearchableInfo(searchManager
            .getSearchableInfo(getComponentName()));
        searchView.setOnQueryTextListener(this);
        return super.onCreateOptionsMenu(menu);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        //Action bar'a eklenen her item elementi otomatik id alır.
        //Bu elementlerden seçilene göre işlem yapmak için switch case kullanılması mantıklıdır
        int selectionStart;
        switch (item.getItemId()) {

            case R.id.action_content_replace:
                return true;
            case R.id.action_important:

                mTxtMain.getText().insert(mTxtMain.getSelectionStart(), "2,");
                return true;
            case R.id.action_keyword:

                mTxtMain.getText().insert(mTxtMain.getSelectionStart(), "3,");
                return true;
            default:
                return super.onOptionsItemSelected(item);
        }
    }
}

```

```

    }
}
@Override
public boolean onQueryTextSubmit(String query) {
    if (query.length() > 0) {
        Toast.makeText(getApplicationContext(), query, Toast.LENGTH_SHORT).show();
    }
    return false;
}

@Override
public boolean onQueryTextChange(String newText) {
    mTxtMain.setText(newText);
    return false;
}
@Override
public void setTitle(CharSequence title) {
    mTitle = title;
    getActionBar().setTitle(mTitle);
}
}
}

```

MainActivity sınıfı **Activity** sınıfını **extends** etmiş, ve **SearchView.OnQueryTextListener** interface'ini ise **implement** etmiştir. Bundan dolayı **SearchView.OnQueryTextListener** interface'i içerisinde bulunan **onQueryTextSubmit** ve **onQueryTextChange** metodlarını **implement** etmeliyiz. Hatırlarsak, action bar'a **SearchView** nesnesi eklemiştik. Bu nesne sayesinde arama işlemi yapabiliyoruz. Bu nesne sadece arama amaçlı kullanılmaz ayrıca herhangi bir işlem için de kullanılabilir. Örneğin **SearchView** sayesinde eklenen text box ile, yazılan değeri **Toast** nesnesi ile kullanabiliriz. Bunu yapabilmek için gerekent metodlar **onQueryTextSubmit** ve **onQueryTextChange** metodlarıdır. **onQueryTextSubmit** metodu, arama butonuna bastığımız zaman çalışırken, **onQueryTextChange** metodu **SearchView**'de girilen her karakter için çalışır.

onOptionsItemSelected metodu **Activity** sınıfı içerisinde bulunan bir metod'tur. **MainActivity** sınıfı **Activity** sınıfını extend ettiğinden dolayı bu metodu direkt kullanabiliriz. Bu metod action bar'a eklenen item elementlerinden herhangi birine tıklandığında çalışır.

onCreateOptionsMenu metodu, **Activity** sınıfının standart seçenek menüsünü initialize eder. Eğer bizim kendi menü elemanlarımız varsa, bu uygulamamızda **res/menu/main.xml** dosyasında bulunmaktadır, eklemek için kullanılır. Bu metod bir kez çalışır, eğer menü her gösterildiği zaman güncellemek istiyorsanız **onPrepareOptionsMenu** metodunu kullanmanız gereklidir.