

Android Activity'nin Dialog Olarak Açılmasını Sağlamak

Bir **Activity** sınıfını dialog olarak açmak istiyorsak **AndroidManifest.xml** dosyasında, **Activity** set ederken tema değerini, android dialog temalarından birine set etmemiz gereklidir:

```
1 | <activity android:name=".UserAdd" android:theme="@android:style/Theme.Holo.Light.Dialog" />
```

Bu işlem tamamlandıktan sonra **Activity** sınıfı, dialog olarak açılacaktır. Eğer dialog'un boyutunu özelleştirmek istiyorsanız aşağıdaki kodları **onCreate()** metodu içerisinde, **setContentView()** metodundan **önce** kullanmak gereklidir:

```
1 | DisplayMetrics metrics = getResources().getDisplayMetrics();
2 | int screenWidth = (int) (metrics.widthPixels * 0.80);
3 | getWindow().setLayout(screenWidth, ViewGroup.LayoutParams.WRAP_CONTENT);
4 | setContentView(R.layout.user); //setContentView() metodundan önce kullanıldı
```

0.80 sayısı ile ekran genişliğini çarparak, dialog olarak açılacak **Activity**'nin **%80** genişlikte olması sağlanır.

Activity'nin Kapatılması

Aşağıdaki kodları kullanarak dialog olarak açılan **Activity**'nin layoutu dışında bir yere tıkladığımızda, bu **Activity**'nin kapatılıp kapatılmamasını sağlayabiliriz.

1. Boşluğa tıkladığında kapatılmasını **istiyorsak**:

```
1 | getWindow().setFlags(WindowManager.LayoutParams.FLAG_WATCH_OUTSIDE_TOUCH,
2 |   WindowManager.LayoutParams.FLAG_WATCH_OUTSIDE_TOUCH);
```

2. Boşluğa tıkladığında kapatılmasını **istemiyorsak**:

```
1 | getWindow().setFlags(WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL,
2 |   WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL);
```

Not: Yukarıda verilen kodlar **setContentView()** metodundan **önce** kullanılır.

3. Ayrıca **Activity** sınıfının **onTouchEvent()** metodu **override** edilir. Boşluğa tıkladığı zaman, **MotionEvent.ACTION_OUTSIDE == event.getAction()** değeri **true** olur.

```
1 | @Override
2 | public boolean onTouchEvent(MotionEvent event) {
3 |     // If we've received a touch notification that the user has touched
4 |     // outside the app, finish the activity.
5 |     if (MotionEvent.ACTION_OUTSIDE == event.getAction()) {
6 |         //finish(); todo finish() aktif yapılırsa activity disinda tiklandiginda dialog kapanir
7 |         return true;
8 |     }
9 |
10 |    // Delegate everything else to Activity.
11 |    return super.onTouchEvent(event);
12 | }
```

Dialog Başlığını Özelleştirmek

Dialog başlığını özelleştirmek için öncelikle custom bir layout eklememiz gereklidir. Örnek bir layout (dialog_title_content.xml):

```
1 | <?xml version="1.0" encoding="utf-8"?>
2 | <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3 |     android:orientation="vertical"
4 |     android:layout_width="match_parent"
5 |     android:layout_height="match_parent">
6 |     <TextView
7 |         xmlns:android="http://schemas.android.com/apk/res/android"
8 |         android:id="@+id/text_title_user_dialog"
9 |         android:layout_width="fill_parent"
10 |        android:layout_height="fill_parent"
11 |        android:textColor="@android:color/darker_gray"
12 |        android:gravity="center"/>
```

Activity sınıfının **onCreate()** metodu içerisinde eklenmesi gereken kodlar ise şunlardır:

```
1 requestWindowFeature(Window.FEATURE_CUSTOM_TITLE);
2 setContentView(R.layout.user);
3 getWindow().setFeatureInt(Window.FEATURE_CUSTOM_TITLE, R.layout.dialog_title_content);
4
5 //Baslik icerigini set ediyoruz
6 TextView title = (TextView) findViewById(R.id.text_title_user_dialog);
7 title.setText("Kullanıcı Giriş Sayfası");
8 title.setTextSize(20);
9 title.setTextColor(Color.parseColor("#ff0092f4"));
```

Örnek uygulamayı indirmek için **tıklayınız** (<http://www.softwarevol.com/sources/codes/android/activityAsDialog.zip>)