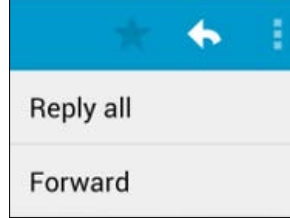


## DERS İÇERİĞİ:

1. Popup Menu
2. AppBar – Action Bar - Toolbar

### Popup Menu:

Popup Menü view'e bağlı olan bir menüdür. Eğer yer varsa ilgili view'in altında yoksa üstünde gösterilir.



Context Menü ile karıştırılmamalıdır. Context Menü ilgili view'de seçili olduğu seçeneği etkiler. Popup Menü kullanabilmek için API level 11 ve daha üstü gereklidir.

**Fig.1.** Gmail uygulamasında sağ üst köşedeki butona bağlı olan bir pop menü.

## Uygulama

**Amaç:** (Bu kısmı uygulamanın sonunda siz dolduracaksınız!!!)

.....

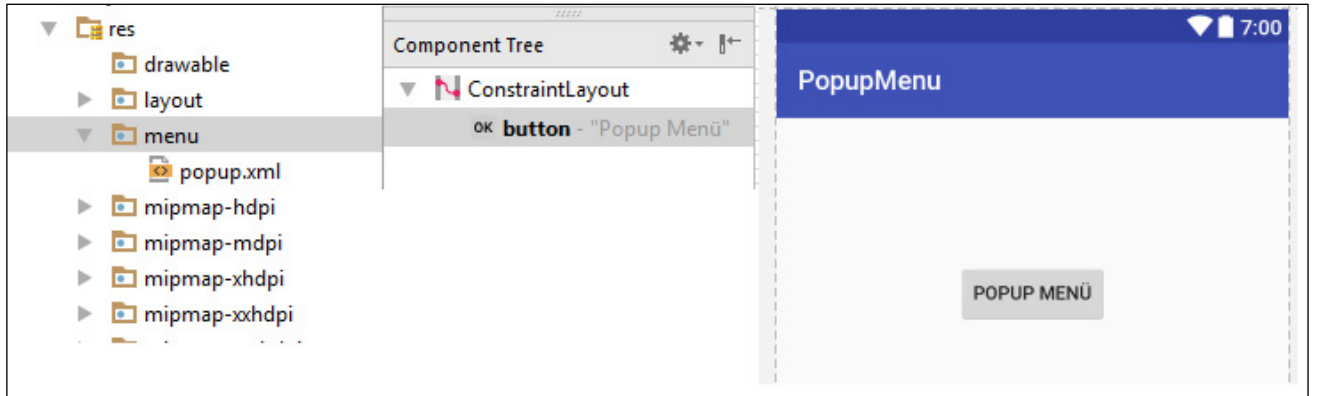
.....

.....

.....

.....

1. Start New Android Project, Appliation Name: **PopupMenu**, Min. SDK: **API20**, Empty Activity, Finish.
2. Aşağıdaki gibi bir tasarım oluşturun.



3. Yukarıdaki gibi bir menü klasörü oluşturun ve popup.xml'in içeriğini aşağıdaki gibi doldurun.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/tumuneYanıtla"
          android:title="Tümüne Yanıtla"/>
    <item android:id="@+id/ilet"
          android:title="İlet" />
</menu>
```

4. Aşağıdaki ifadenin işlevinin ne olduğunu arkadaşınızla tartışın ve bu ifadeyi nereye eklemeniz gerekiyorsa oraya

```
android:onClick="popupGoster"
```



MainActivity sınıfı PopupMenu.OnItemClickListener implement etmelidir! popup.inflate() metodu için API level 14 gereklidir.

ekleyin.

5. Aşağıdaki kodu MainActivity sınıfına ekleyin ve kodların yerine getirdiği işlevi arkadaşınızla tartışıp aşağıdaki boşluğa yazın.

```
public void showPopup(View v) {
    PopupMenu popup = new PopupMenu(this, v);
    popup.setOnMenuItemClickListener(this);
    popup.inflate(R.menu.popup);
    popup.show();
}
```

.....

.....

.....

.....

6. Son olarak item'lere tıklandığında gerçekleşmesini istediğimiz eylemleri tanımlarız.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

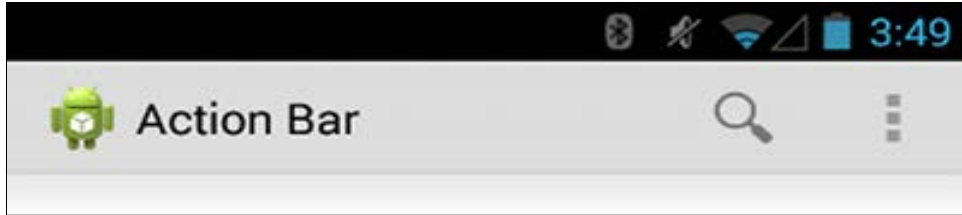
    switch (item.getItemId()) {
        case R.id.tumuneyanıtla:
            Toast.makeText(MainActivity.this, "Tümüne Yanıtla Tıklandı", Toast.LENGTH_LONG).show();
            return true;
        case R.id.ilet:
            Toast.makeText(MainActivity.this, "İlet Tıklandı", Toast.LENGTH_LONG).show();
            return true;
        default:
            return false;
    }
}
```

## AppBar – Action Bar – Toolbar:

**Action Bar - Toolbar**, uygulamanızın Activity'leri için en önemli tasarım bileşenlerinden birisidir. Diğer Android uygulamalarıyla tutarlı ve uygulamayı daha alışılabilir hale getiren bir arayüz özelliği sağlar.

Ana işlevleri şunlardır:

- Uygulamaya bir kimlik verir ve kullanıcının uygulamadaki yerini gösteren bir alan sağlar.
- Sıklıkla kullanılan eylemlere (Arama gibi) kısa yolla erişim sağlar.
- Navigasyon ve görüntü değiştirme desteği (sekmelerle veya açılır listelerle) sağlar.



En temel haliyle Action Bar, Activity'nin başlığını ve uygulamanın simgesini (solda) gösterir. En basit halinde bile kullanıcılara uygulama içerisinde nerede olduklarını bildirir. İsim karmaşıklığının sebebi başta sadece ActionBar olması, sonraki sürümlerde ise daha genel hali olan Toolbar'ın çıkmış olmasıdır.

## Uygulama

**Amaç:** (Bu kısmı uygulamanın sonunda siz dolduracaksınız!!!)

.....

.....

.....

.....

.....

1. Start New Android Project, Appliation Name: **Toolbar**, Min. SDK: **API15**, Empty Activity, Finish.
2. v7 appcompat destek kütüphanesinin projenize eklendiğinden emin olun.

```
import android.support.v7.app.AppCompatActivity;
```

3. MainActivity'nin "**AppCompatActivity**"i extends ettiğinden emin olun.
4. Manifest dosyasında **temayı** aşağıdaki gibi değiştirin. NoActionBar temalarından birini kullanmak uygulamanın varolan ActionBar ayarlarını kullanmasını engeller.

```
android:theme="@style/Theme.AppCompat.Light.NoActionBar"
```

5. Projenize Palette kısmından aşağıdaki gibi bir Toolbar ekleyin ve ilgili özelliklerini aşağıdaki gibi değiştirin.

```
android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
```

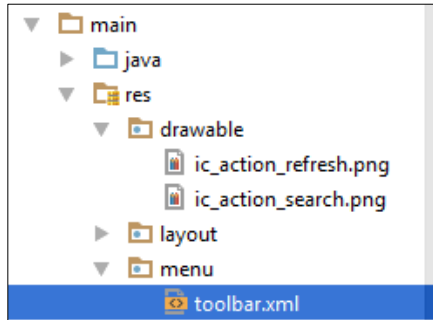
6. Aşağıdaki kodları MainActivity'ye ekleyin. Bu kodların yerine getirdiği işlev nedir? Aşağıya yazın.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar tb = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(tb);
}
```



Action Bar, uygulamadaki o anki context'le ilgili önemli işlemleri düğme olarak üzerine eklemeyi sağlar. Bu düğmeler bir simge ve/veya metin ile Action Bar'ın üzerinde görünür. Action Bar'a sığmayan veya yeterince önemli olmayan eylemler gizlenir. Buna **actions overflow** denilmektedir.

7. Föyle birlikte verilen resimleri aşağıdaki gibi drawable klasörüne kopyalayın.



8. ActionBar üzerindeki butonları tanımlamak için toolbar.xml adında bir menü oluşturun. İçeriğini aşağıdaki gibi doldurun.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item android:id="@+id/action_arama"
        android:icon="@drawable/ic_action_search"
        android:title="Ara"
        app:showAsAction="ifRoom" />

  <item android:id="@+id/action_yenile"
        android:icon="@drawable/ic_action_refresh"
        android:title="Yenile"
        app:showAsAction="ifRoom" />

  <item android:id="@+id/action_ayarlar"
        android:title="Ayarlar"
        app:showAsAction="never" />
</menu>
```

9. Son olarak onCreatOptionsMenu() ve onOptionsItemSelected() metodlarını Föy3'tekine benzer şekilde yazın.
10. app:showAction özelliğinin işlevini açıklayın.

.....

.....

.....

.....