



# **Android Studio Dersleri**



**Ümit KÖSE**

**[Umiitkose.com](http://Umiitkose.com)**

**Herkese Merhaba, Ben Ümit Köse. Fırat Üniversitesi Bilgisayar Mühendisliği 4. Sınıf öğrencisiyim. 2 sene boyunca blogumda yer verdiğim android derslerini pdf kitap halinde toplamak istedim. Benimle [umiitkose.com](http://umiitkose.com) üzerinden ya da sosyal ağlar ile irtibata geçebilirsiniz.**

**Herkese iyi öğrenmeler, iyi okumalar.**

**Sosyal Ağlar'ım..**



# İçindekiler

[Android Programlama Giriş](#)

[1-Android Studio Kurulum](#)

[2-Giriş ve Emülatörde Hello World Uygulaması](#)

[3-Button Tanımlama ve Android Manifest Uygulaması](#)

[4-Intent Kavramı ve Yeni Ekran oluşturma](#)

[5-Thread Kavramı ve Splash Ekran](#)

[6-Uygulamayı Tam Ekran Yapma](#)

[7-Button Still Tanımlama](#)

[8-Layoutlar](#)

[9-webView Kullanımı ve Uygulamada Web Sayfası Açma](#)

[10-Titreşim Kullanımı \(Vibrator\)](#)

[11-Android'te Yazı Tipi Değiştirme](#)

[12-Mac Adresi, Üretici ve Model Bilgileri](#)

[13-Option \(Menü\) Oluşturma](#)

[14-Arama Yapma](#)

[15-Bluetooth Kontrolü](#)

[16-Alert Dialog](#)

[17-Progress Dialog](#)

[18-Rating Bar](#)

[19-CheckBox](#)

[20-RadioButton](#)

[21-Button'ları tek bir listener ile kontrol etme](#)

[22-Wifi Kullanımı](#)

[23-Telefon Kamerası Kullanımı](#)

[24-Video Çekme, Oynatma](#)

[25-ScrollView Kullanımı](#)

[26-Android Studio Dil Desteği](#)

[27-MediaPlayer ile Ses Kontrolü](#)

[28-Paylaş Butonu](#)

[29-PushBots ile Bildirim Göndermek](#)

[30-İnternet Kontrolü](#)

## Uygulamalar

[1-Hesap Makinesi](#)

[2-İnternet Tarayıcısı](#)

[3-Flash Uygulaması](#)

[Android Studio Apk dosyası oluşturma](#)

[Android Genymotion Kurulumu](#)

## Android Studio Programlama Giriş

---

# Android Nedir?

Artık tabletlerde çoğu telefonlarda neredeyse hayatımızın çoğu işleminde android kullanıyoruz. En basit açıklaması hayatımızı kolaylaştırmak için yazdığımız ve google tarafından bizlere bir açık kaynaklı mobil işletim sistemi olarak sunulan teknolojidir.

Android, Google, Open Handset Alliance ve özgür yazılım topluluğu tarafından geliştirilen, Linux tabanlı, mobil cihaz ve cep telefonları için geliştirilmekte olan, açık kaynak kodlu bir mobil işletim sistemidir. Desteklenen uygulama uzantısı “.apk”dır.

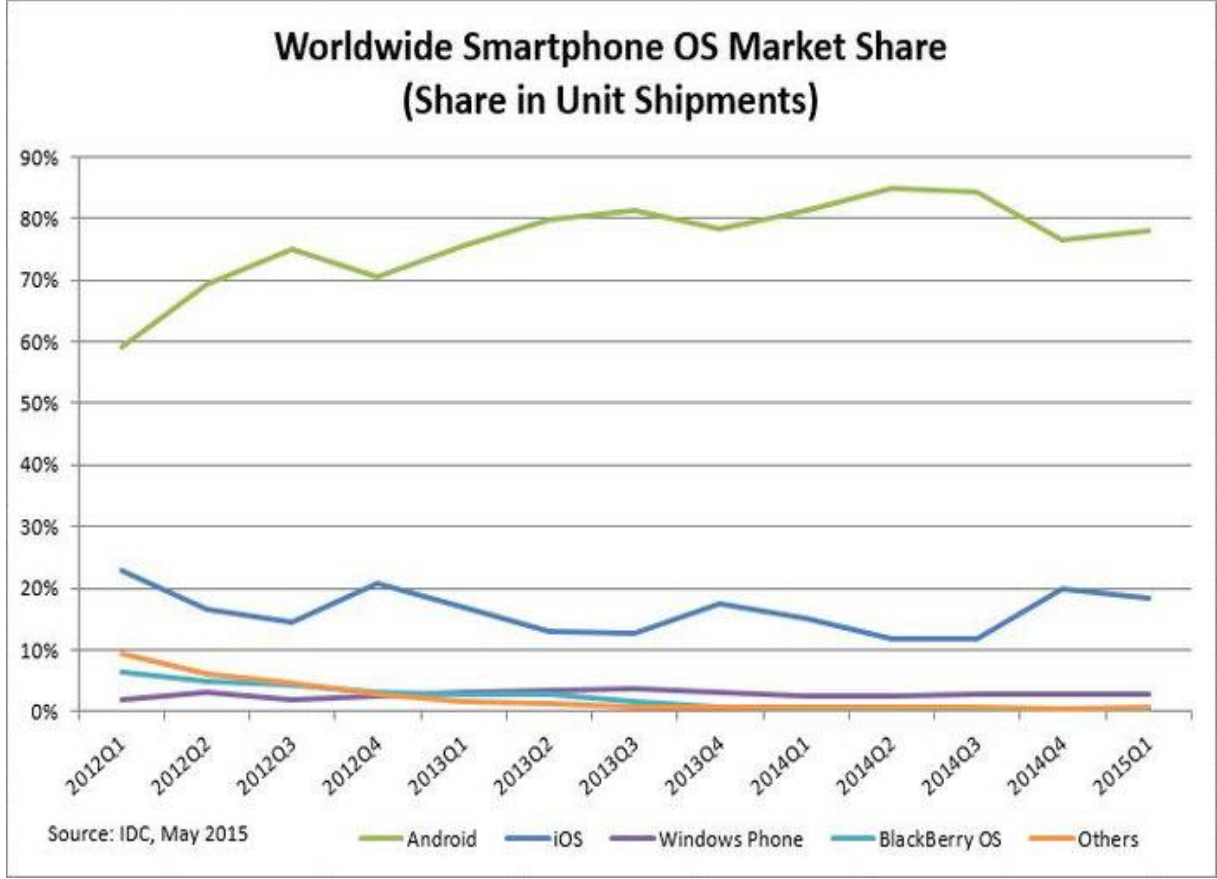
Android, aygıtların fonksiyonelliğini genişleten uygulamalar yazan geniş bir geliştirici grubuna sahiptir. Google Play, Android işletim sistemi uygulamalarının çeşitli sitelerden indirilebilmesinin yanı sıra, Google tarafından işletilen kurumsal uygulama mağazasıdır.

Android, linux çekirdeği üzerine inşa edilmiş bir mobil işletim sistemidir, bu sistemde ara katman yazılımı, kütüphaneler ve API C diliyle yazılmıştır. Uygulama yazılımları ise, Apache harmony üzerine kurulu java-uyumlu kütüphaneler ihtiva eden uygulama iskeleti üzerinden çalışır. Android, derlenmiş java kodunu çalıştırmak için dinamik çevirmeli (JIT) Dalvik sanal makinesini kullanır ve cihazların fonksiyonerliğini artıran uygulamaların geliştirilmesi için çalışan geniş bir programcı-geliştirici çevresine sahiptir.

Android ile ilgili bilgileri verdikten sonra bir de google'in Temmuz 2005'te satın aldığı android'te bugüne kadar ki gelişmelere bakarsak.

## Android'in Pazar Payı.

Dünyaca ünlü araştırma şirketi IDC'nin verilerine göre dünya üzerindeki akıllı telefon satış rakamı, 2015'in ilk çeyreğinde yüzde 334,4 milyon adet olarak gerçekleşti. Günümüzde bu büyük pazarda yarışan dört önemli mobil işletim sistemi var: Android, iOS, Windows Phone ve BlackBerry 10.



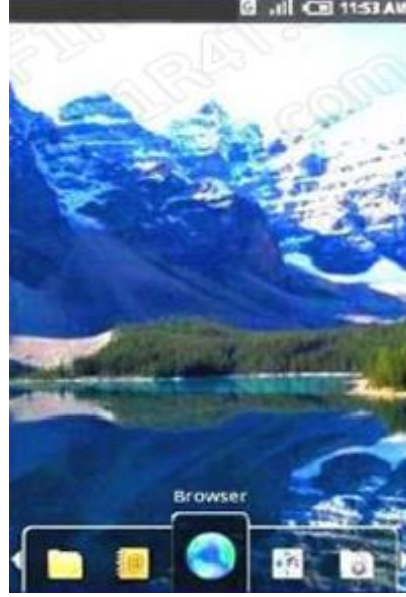
Android'in bu pazardaki gelişimini incelediğimizde yine IDC'nin araştırmasına göre 2011'den bu yana düzenli olarak büyüdüğünü ve pazarın hâkimi konumunu sürekli olarak daha daha yukarı taşıma eğiliminde olduğunu görüyoruz. 2011'in ikinci çeyreğindeki pazar payı yalnızca yüzde 36,1 iken, bu rakam 2015'in ilk çeyreğinde yüzde 78'e ulaştı.

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

## Android sürümlerinin tarihine bakarsak

**Android 1,0:** HTC Dream'in işletim sistemi olan bu sürüm ile android işletim sistemi adını duyurmaya başladı. Bu sürüm kamera desteği, web tarayıcı, wifi, Youtube, alarmlı saat, galeri, bluetooth, sesli arama gibi özellikleri barındırmaktadır.

**Android 1,1:** 9 Şubat 2009'da yayınlanan bu sürümde bir önceki sistemin hatalarının düzeltilmesi ve API'yi geliştirme üzerine çalıştı.



**Android 1,5: Cupcake** 30 Nisan 2009 da yayımlandı. Bu sürümde Bluetooth desteği, kamera kaydı, video gösterimi, Picasa, sık kullanılanlar klasörü ve Widget'ları kişiselleştirebilme, yazı tahmin edebilen klavye ve animasyonlu ekran özellikleri Android'e kazandırıldı.



**Android 1,6: Donut** 15 Eylül 2009'da yayınlandı. Ekran çözünürlüğü yükseltildi. Android Market, galeride çoklu silme, sesli Arama ve WVGA ekran çözünürlüğü desteği geldi.



**Android 2,0 ve 2,1: Eclair** 29 Ekim 2009'da yayınlandı. HTML5 ve Bluetooth 2.1 desteği geldi. Sanal klavye ve yüksek ekran çözünürlüğü ile özelleştirilebilen arayüz özelliği eklendi. Google maps geliştirildi. Hareketli duvar kağıtları ve dijital yakınlaştırma özellikleri geldi.



**Android 2,2: Froyo** Mayıs 2010'da yayınlandı. 720p ekran çözünürlüğü desteği, USB, Wifi tarayıcı özelliği ve Flash Player 10.1 desteği geldi.



**Android 2.3: Gingerbread** Şubat 2011'de yayımlandı. Çoklu kamera ve Çoklu dokunmatik desteği eklendi. Video arama özelliği eklendi.

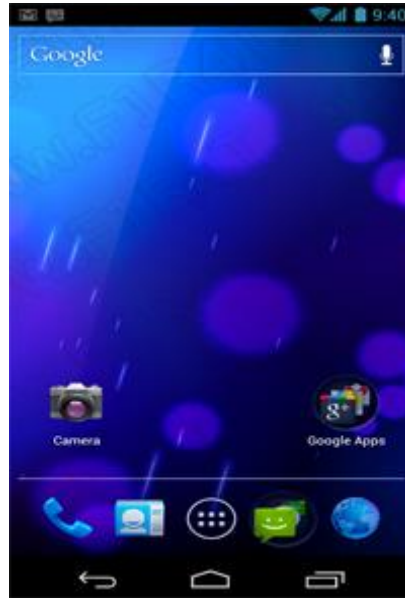


**Android 3,0: Honeycomb** Şubat 2011'de yayınlandı. İşletim sistemi tabletlere uyumlu hale getirildi.





**Android 4,0: Ice Cream Sandwich** Ekim 2011’de yayınlandı. Yüz tanıma fonksiyonları ile NFC dosya paylaşım özelliği eklendi. Mart 2012’de Android market adını Google play Store olarak değiştirdi.



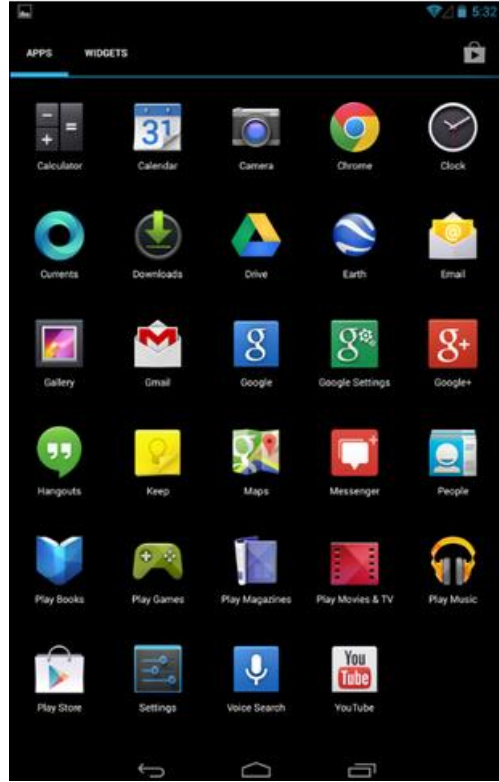
**Android 4,1: Jelly Bean** Temmuz 2012’de çıktı. Aynı Anda iki uygulama açma özelliği kazandırıldı. Enerji verimliliği ve takılmadan kayan ekran eklendi.

**Android 4.2: Jelly Bean Plus** Ekim 2012’de yayınlandı. Çoklu kullanıcı, 360 derece panoramik fotoğraf kazandırıldı.

**Android 4,2,2:** 11 Şubat 2013 tarihinde yayınlandı. Isınma ve donma problemleri giderildi.

**Android 4,3: Jelly Bean** yayınlandı. Çoklu kullanıcı için ebevent kontrolü, oyunlar için 3.0 Open Gl Es desteği, Akıllı bluetooth ve Wifi teknolojisi eklendi. Ayrıca uygulama izinleri bu sürümle kontrol edilebilir hale getirildi.

**Android 4,4: Kitkat** 31 Ekim 2013’te yayınlandı. Görsel arayüzde bir çok değişiklik yapıldı.



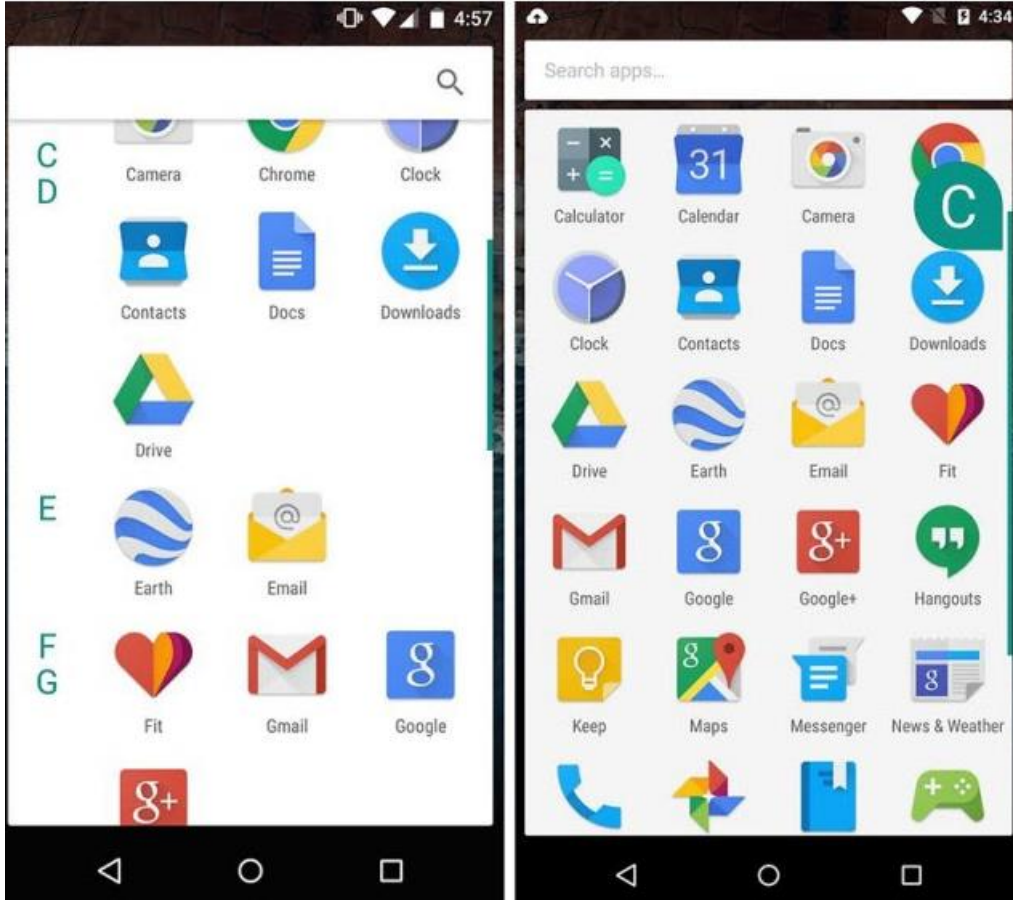
3 Kasım 2014 tarihinde Google Android 5.0 Lollipop sürümünü çıkarttı. Tüm uygulamalar ve hizmetler “*Material Design*” altında yenilendi.



2 Aralık 2014 tarihinde Google Android 5.0.1 güncellemesi yayınlandı.

19 Aralık 2014 tarihinde Google bazı hata düzeltmeleri içeren Android 5.0.2 sürümünü yayınladı.

29 Mayıs 2015 tarihinde Google yaptığı I/O 2015 konferansında Android M 6.0'ı duyurdu ve önizleme sürümünü yayınladı. M = **Marshmallow**



**Marshmallow** ile detaylı uygulama izinleri, gelişmiş parmak izi okuma, Döndürülebilir ana ekran, Arayüzde geliştirme gibi yeni özellikler eklendi.

Android uygulamalara erişebileceğiniz ve kendi uygulamanızı yayınlayabileceğiniz marketler listesi:

**Google Play Store** : \$25 dolar'lık ücret ile üye oluyorsunuz.

<https://play.google.com/store>

**Samsung App Store**:<http://www.samsung.com/tr/apps/mobile/galaxyapps/>

**Amazon app store**: <http://www.amazon.com/mobile-apps/b?ie=UTF8&node=2350149011>

**Opera App store**: [http://apps.opera.com/tr\\_tr/](http://apps.opera.com/tr_tr/)

**Turkcell T Market**: <http://www.t-market.com/>

### Android Studio Kurulum



### Android Studio İçin..

Öncelikle android studio'yu indiriyoruz. [Link için Tıklayın](#). Bir uygulama yüklemiştir. Yine aynı sistem. Eğer Android studioyu kurarken jdk hatası alırsanız [bu adresten](#) jdk'yu indirip kurmanız gerekmektedir. Jdk'yu kurup bu hata ile karşılaşırsanız..

**Error:** cannot start Android Studio. No JDK found. Please validate either ANDROID\_STUDIO\_JDK, or JDK\_HOME or JAVA\_HOME points to valid JDK installation. ECHO is off. Press any key to continue ..

#### Çözümü :

- 1-Bilgisayarına sağ tıklayıp özellikleri seçin.
  - 2-Gelişmiş tabına geçin.
  - 3-Ortam değişkenlerine tıklayın.
  - 4-Sistem değişkenlerinin altından yeni değişken yaratın.
  - 5-Adını **JAVA\_HOME**.
  - 6-Değerini **C:\Program Files\Java\jdk1.7.0\_25** olarak ayarlayın.
  - 7-Sistem değişkenlerinden **path** 'i bulun ve değerini **C:\Program Files\Java\jdk1.7.0\_25\bin** olarak değiştirin.
- Buradaki "**jdk1.7.0\_25**" yerine sisteminizdeki jdk versiyonunu yazmanız gerekiyor. Sonra hatadan kurtulup android studio'ya başlayabilirsiniz.

## Resim üzerinden anlatırsak;

# Download

Before installing Android Studio or the standalone SDK tools, you must agree to the following terms and conditions.

## Terms and Conditions

This is the Android Software Development Kit License Agreement

### 1. Introduction

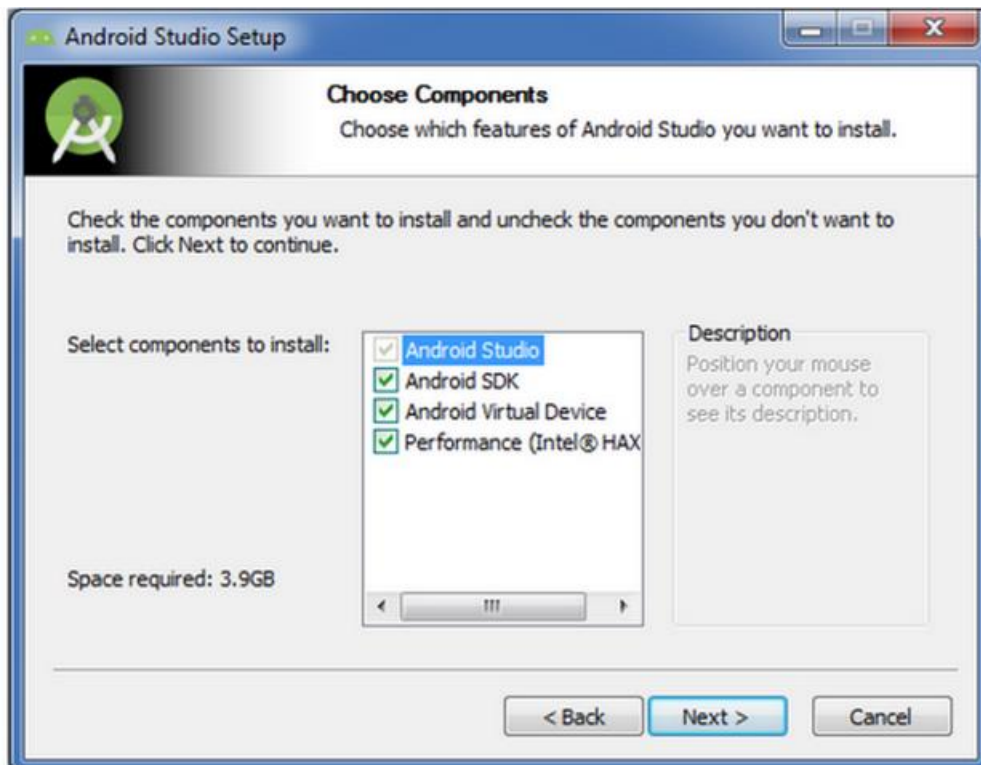
1.1 The Android Software Development Kit (referred to in this License Agreement as the "SDK" and specifically including the Android system files, packaged APIs, and Google APIs add-ons) is licensed to you subject to the terms of this License Agreement. This License Agreement forms a legally binding contract between you and Google in relation to your use of the SDK.

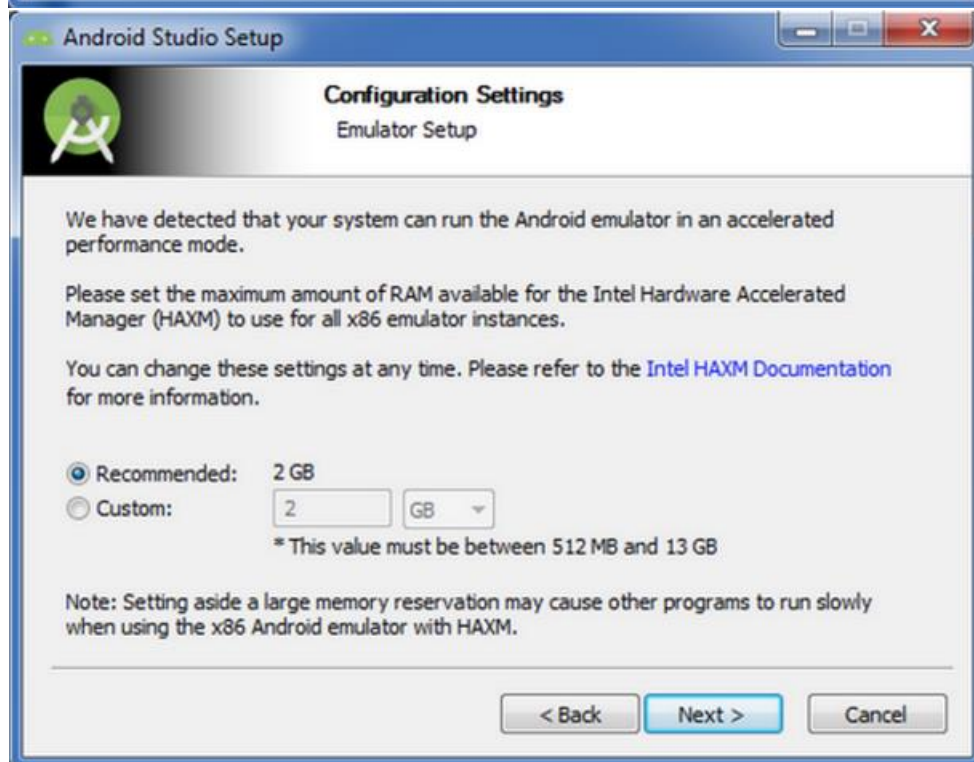
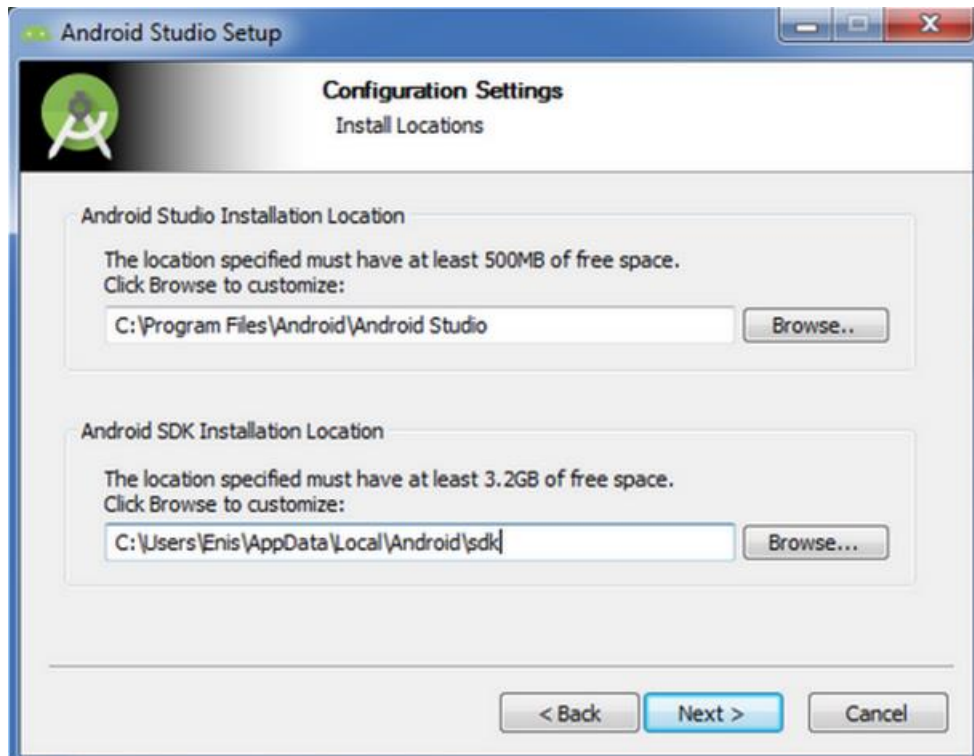
1.2 "Android" means the Android software stack for devices, as made available under the Android Open Source Project, which is located at the following URL: <http://source.android.com/>, as updated from time to time.

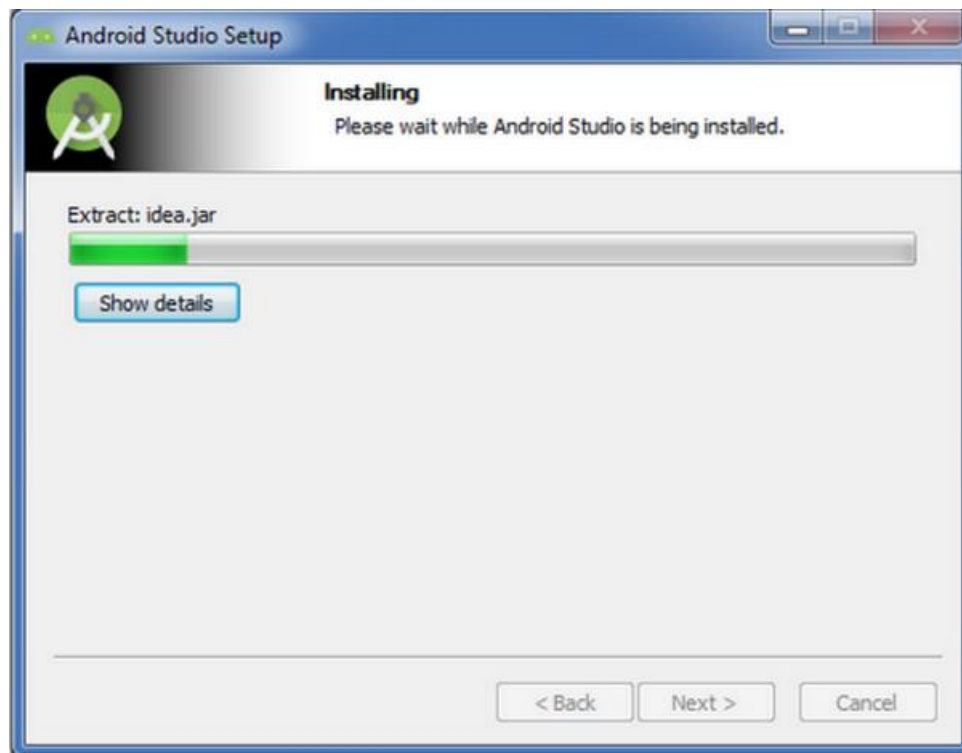
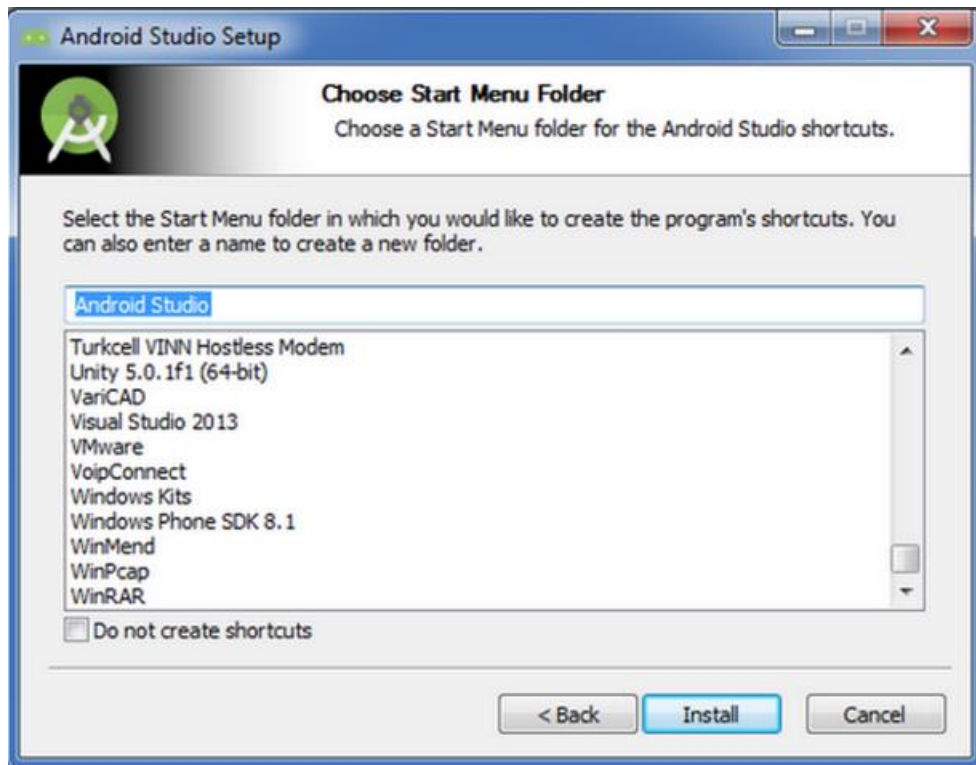
I have read and agree with the above terms and conditions

DOWNLOAD ANDROID STUDIO FOR WINDOWS













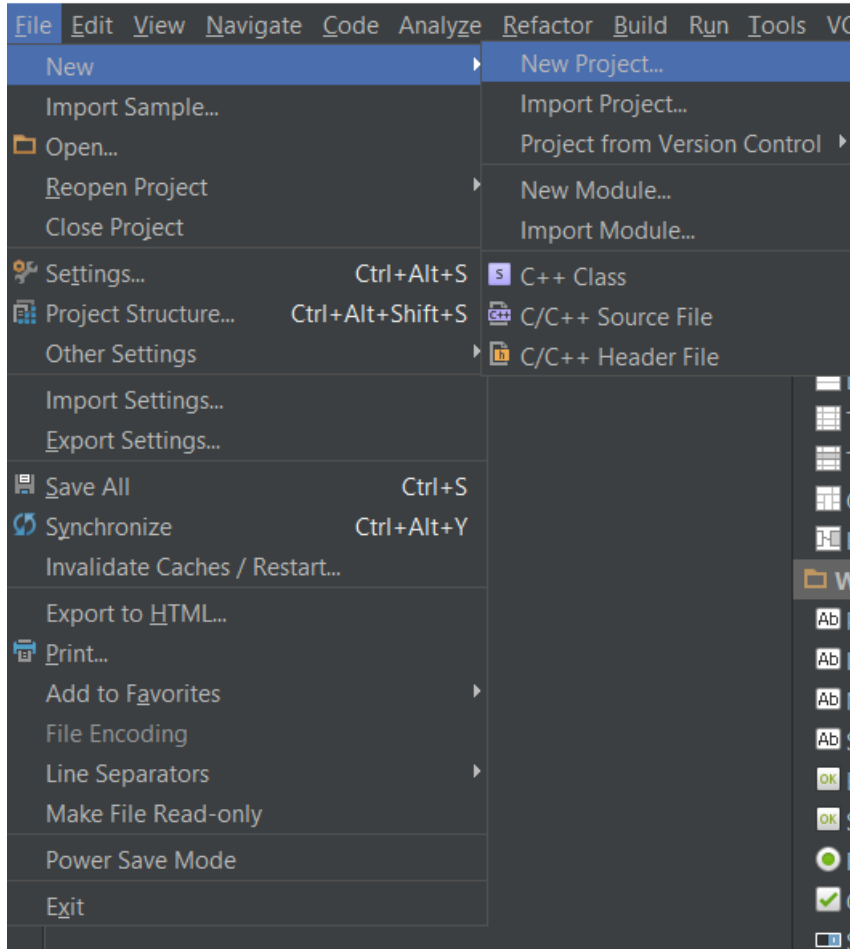
## Android Studio Programlama Bölüm 2

### Giriş ve Emülatör'de Hello World Uygulaması

Android'i tanımaya ve her uygulamada olduğu gibi Hello World yazdırmaya ne dersiniz. İlk kısımda herhangi bir kodlama yazmıyor olacağız. Sadece bir başlangıç yapacağız. İlk başta android studio'ya yakından bakalım.

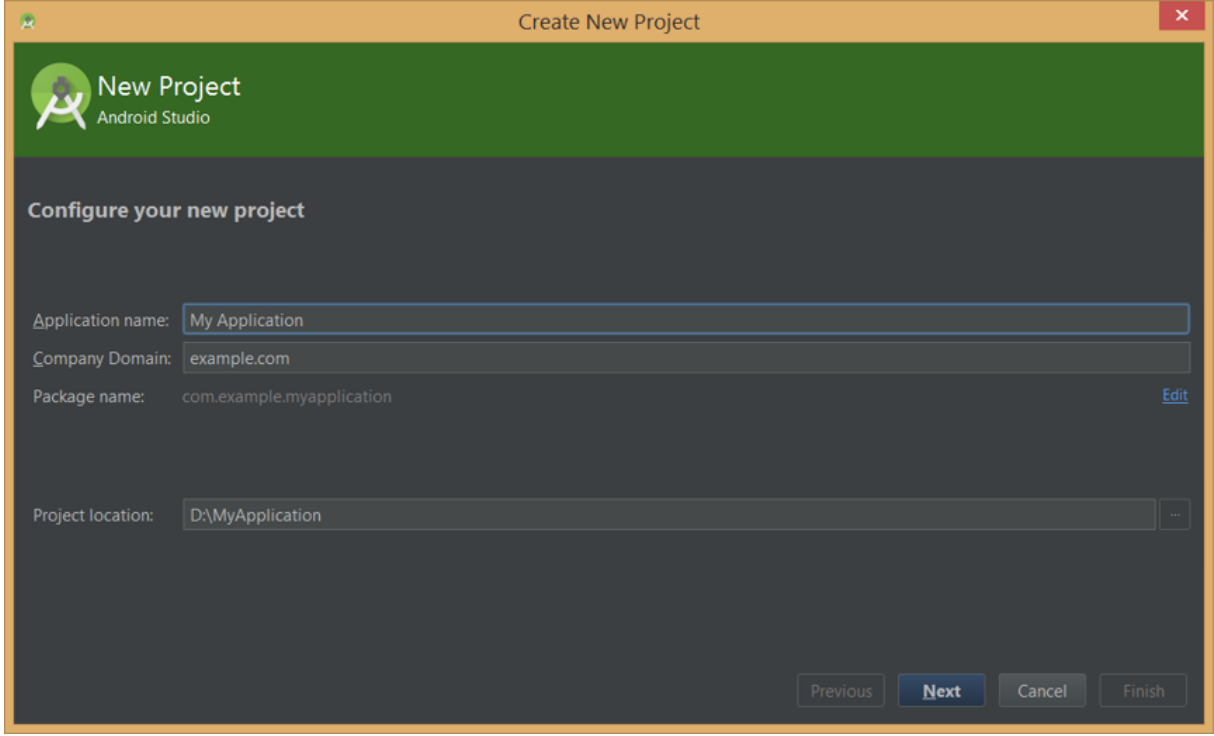
Öncelikle Yeni Bir proje oluşturma kısmı ile başlayalım.

**File-> New -> New Project** ile yeni bir proje oluşturuyoruz.



Yeni proje oluşturma tıkladıktan sonra karşımıza **Create New Project** kısmı geliyor..

Daha yakından incelersek:



The screenshot shows the 'Create New Project' dialog in Android Studio. The dialog is titled 'Create New Project' and 'New Project Android Studio'. It contains the following fields and buttons:

- Application name:** My Application
- Company Domain:** example.com
- Package name:** com.example.myapplication (with an [Edit](#) link)
- Project location:** D:\MyApplication (with a browse button)
- Buttons:** Previous, Next, Cancel, Finish

**Application name:** Uygulama ismini buraya yazıyoruz. Dosyamızın ismi'de aynı zamanda bu olmaktadır. Deęiřtirmek için location'u yazınız.

**İlk harf her zaman büyük olmalıdır. Yoksa hata alırsınız. Rakam ve işaret ile başlamamalıdır.**

**Company Domain:** Hayali bir şirket ismi diye düşünün.

**Package name:** Burası application isminize göre oluşmaktadır. Sizin yaptığınız bir şey yoktur.

**Project location:** Uygulamanızın yolu. Nereye kaydetmek istiyorsanız o dizini belirtin.

Create New Project ×

## Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK: API 9: Android 2.3 (Gingerbread)

Lower API levels target more devices, but have fewer features available. By targeting API 9 and later, your app will run on approximately **99,7%** of the devices that are active on the Google Play Store.

[Help me choose](#)

Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

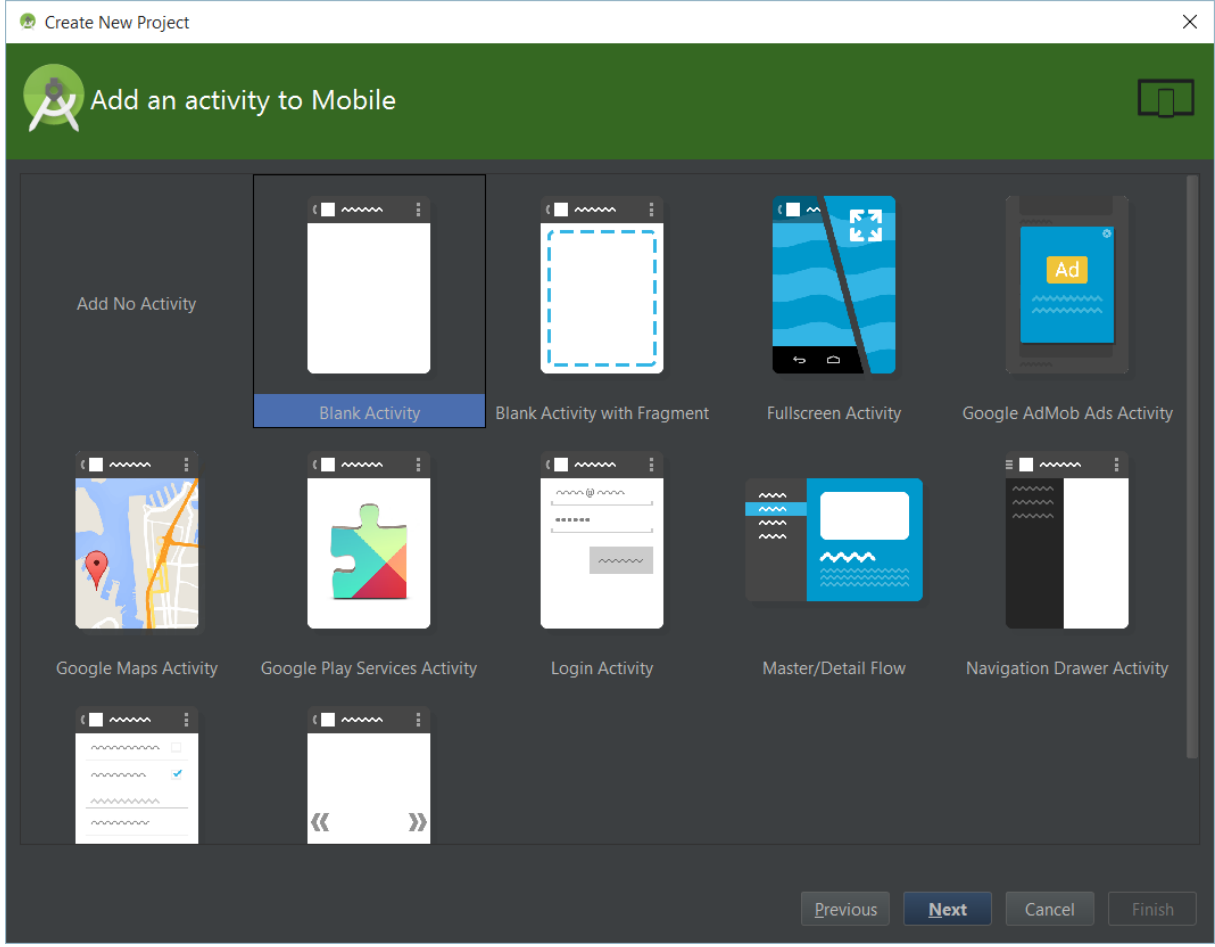
Android Auto

Glass (Not Installed)

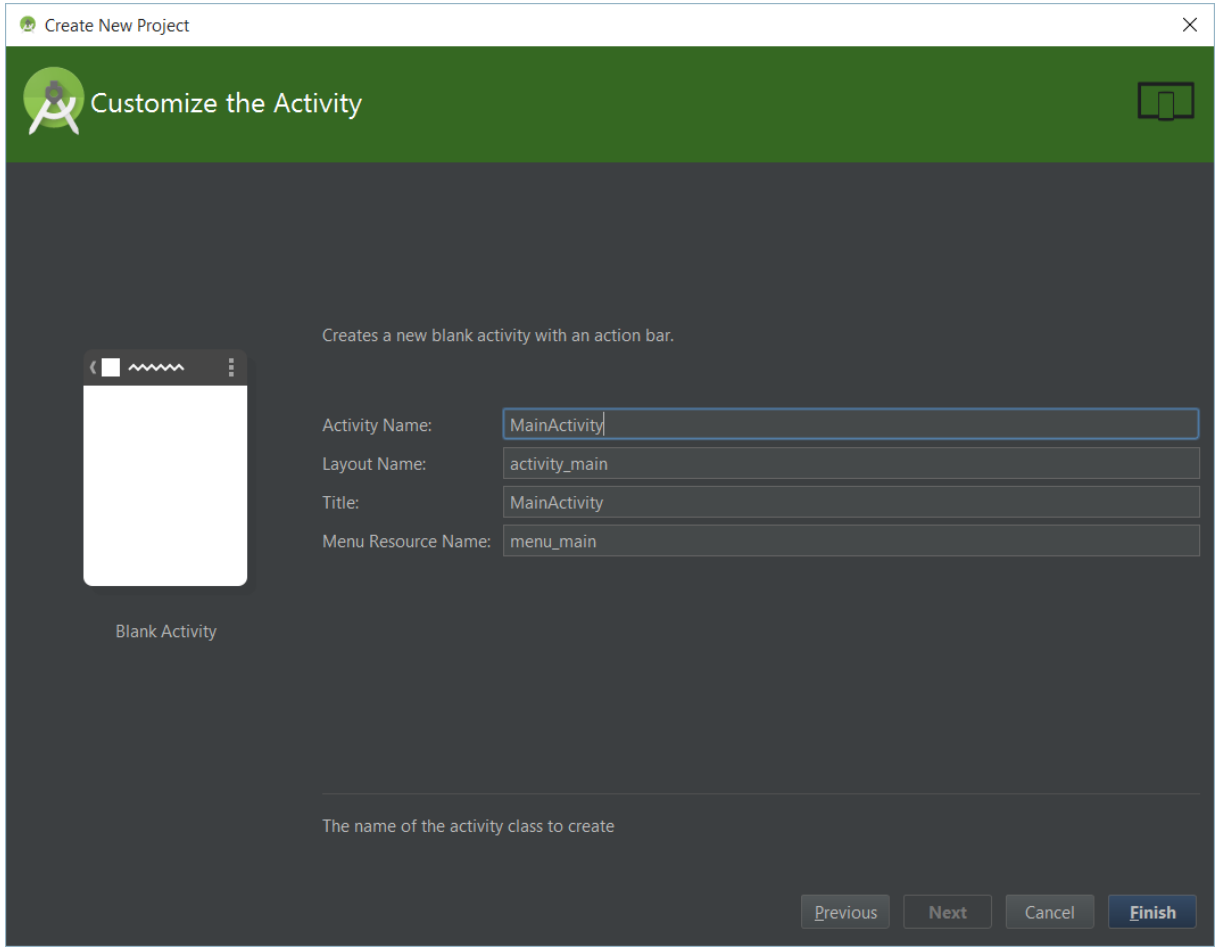
Minimum SDK:

Previous **Next** Cancel Finish

Sonraki kısımda uygulamayı minimum hangi android işletim sisteminde çalıştırmak istediğinize karar veriyorsunuz. Ayrıca Tv Glass ya da Wear (akıllı saat, gözlük ya da Televizyon) için'de ayarlamalarınızı burada yapıyorsunuz.

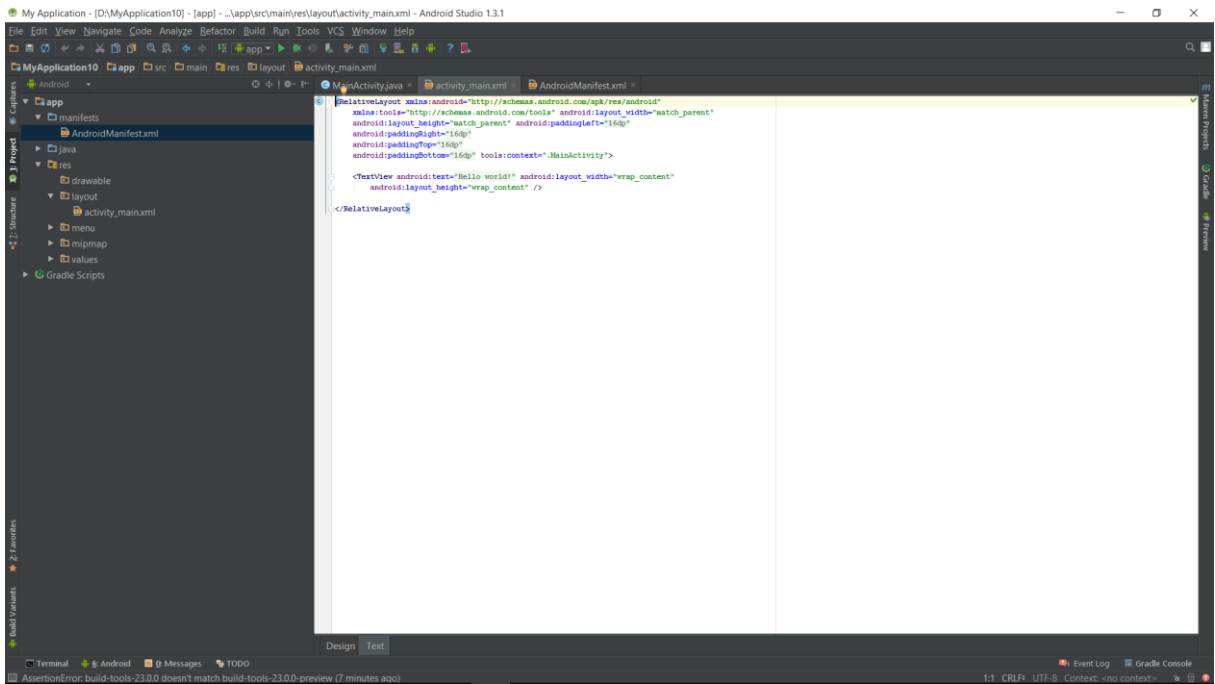


Uygulamamızın activity layout yani tasarım kısmını burada ayarlayabilirsiniz. Bir tavsiye eğer dışarıdan proje aktarıyorsanız boş bir activity oluşturun. Ve klasör olarak taşıyın.



Bu kısım ise tasarım dediğim Activity kısmının isimlendirilmesidir. Burayı kendinize göre ayarlayabilirsiniz. Finish ile yeni projemiz'i oluşturduk.

Yeni proje oluşturduktan sonraki görünüm aşağıdaki gibi.



Şimdi sağdaki android proje kısmını anlatacağım.

**Manifests:** Android programlama için izin kısmı burasıdır. Örneğin wifi'yi kullanmak için telefon'a bunun için izin almamız gerektiğini buraya yazacağız.

**Java:** Java kaynak kodlarını burada bulunduracağız.

**Res:** Resource kısmı'dır. Yani kaynak kodlarımız burada olacak. Menu kısmı, Metin yazıları, resim ve layout kısmı..

**drawable:** Resim dosyalarımızı burada bulunduracağız. Dosyalarımız PNG ya da JPEG formatında olabilir.

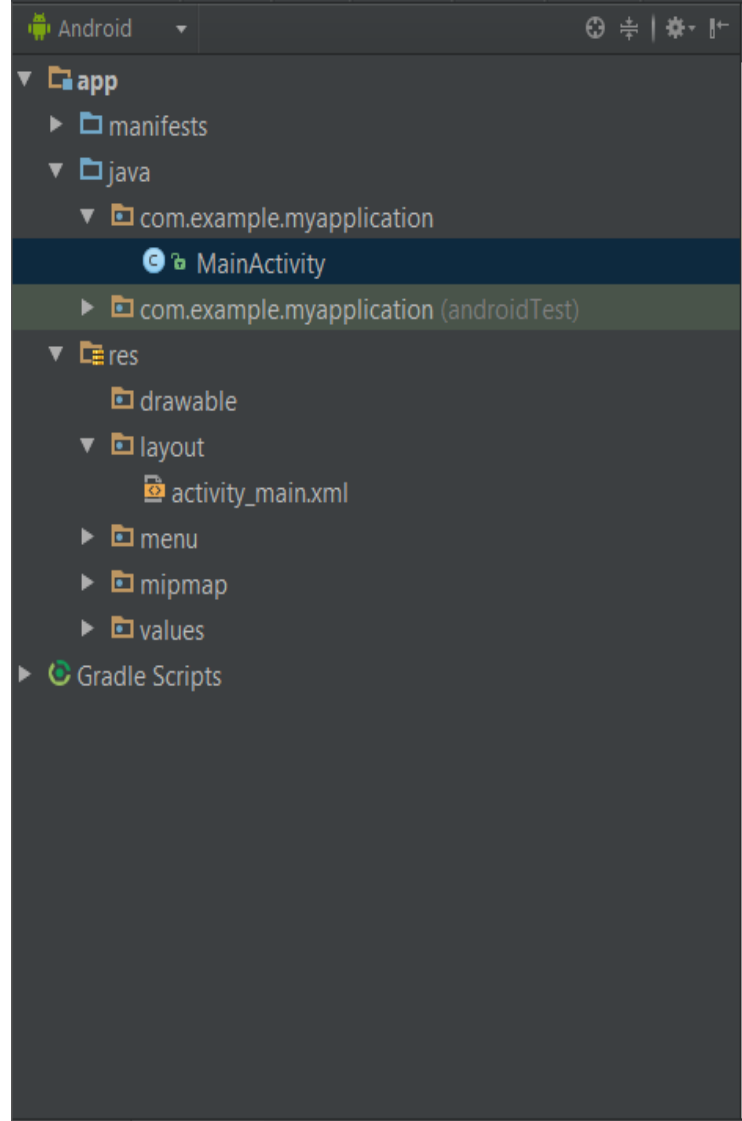
**layout:** Bu kısım tasarım dosyalarımızın olduğu kısımdır. XML (extensible markup language) dili ile yazılmıştır. Buradaki tasarım dosyalarımızı javada Oncreate metodun'da çağırırız.

**menu:** Menu kısmını burada aktif edebiliriz.

**mipmap:** Uygulamanızın açılış logolarını içerir.

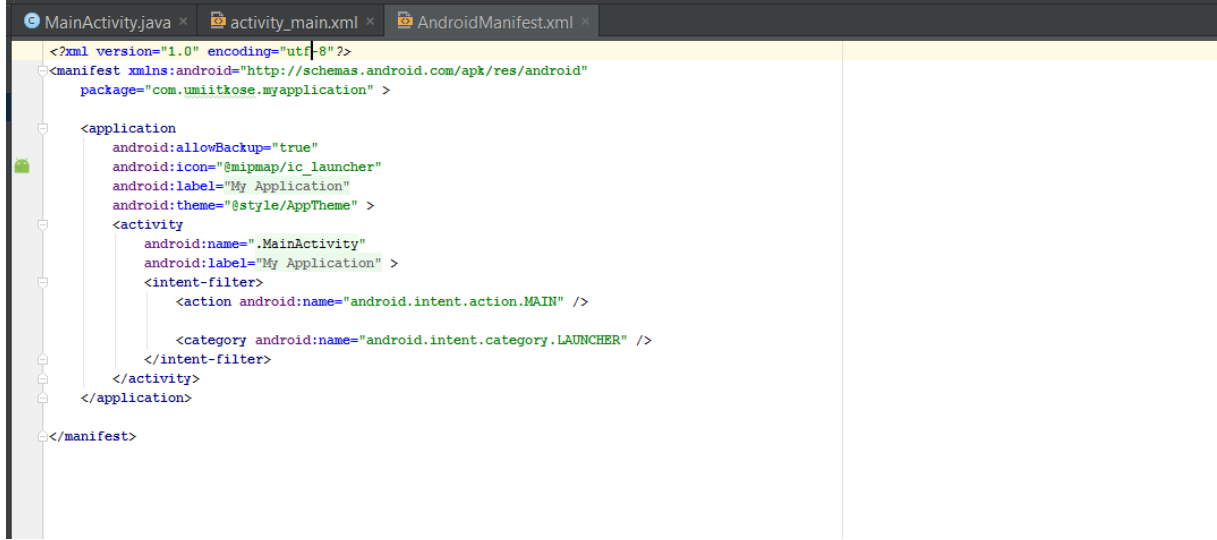
**values:** : Uygulamada kullanılan sabit değişkenler burada saklanabilir. strings.xml dosyası uygulamada kullanılan ve ekranlarda kullanıcıya gösterilen her türlü metni saklar.

**Gradle Scripts:** Gradle, projelerimizi built etmemize yarayan yeni nesil bir sistemdir. Gradle en güzel özelliği test, debug, release esnasında almanız gereken proje paketlerini otomatikleştirmektedir.



Aşağıda tasarım kısmı, Android manifest ve Java kısmının ilk halleri gösterilmiştir. Hadi Hello World'u emülatör de çalıştıralım.

Manifest dosyamıza bakarsak paketimizin ismi, iconumuz'un bulunduğu konumu ilk açılan sayfamızın izni gibi bilgiler burada tanımlanmıştır.



```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.umitkose.myapplication" >

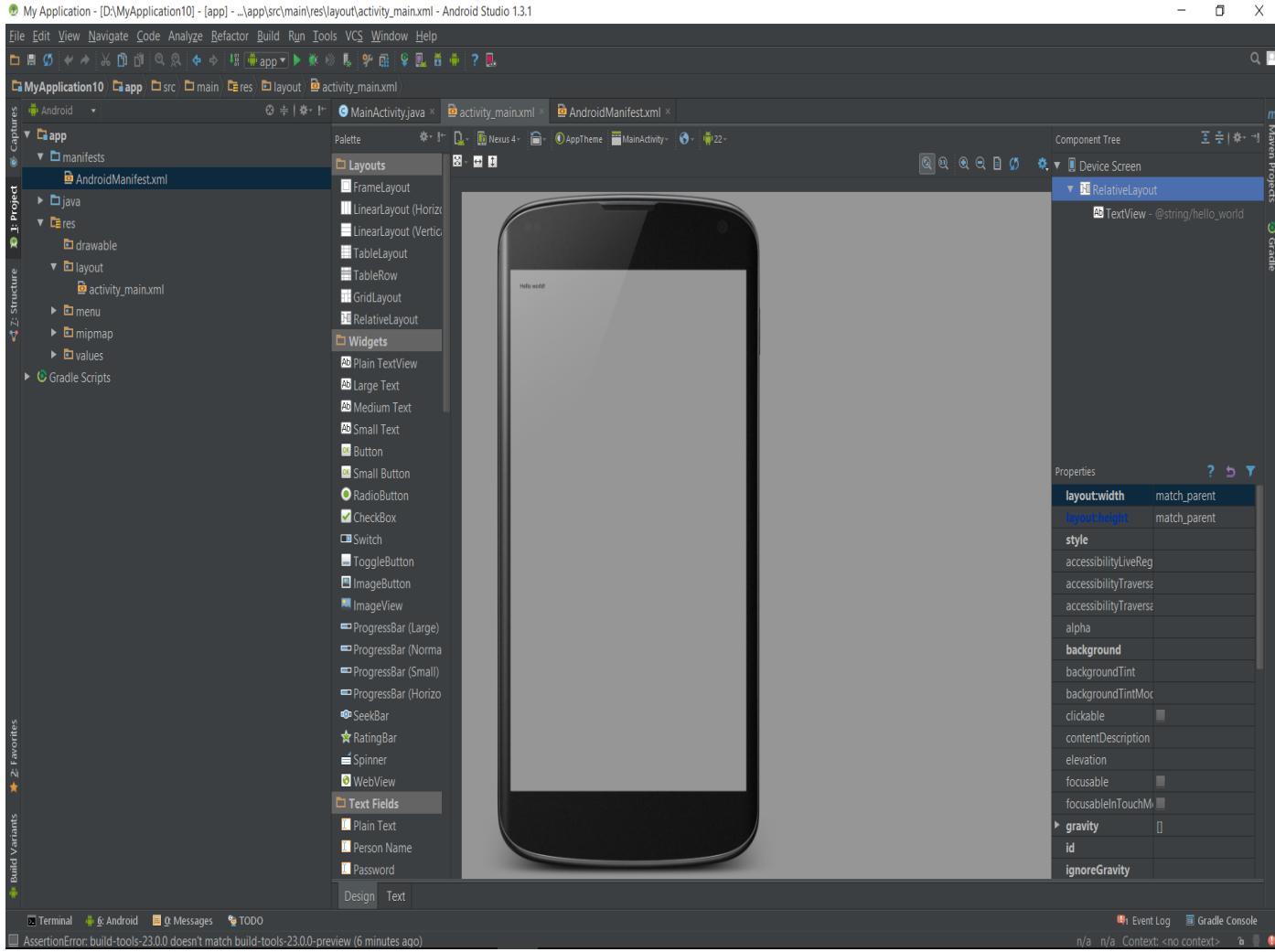
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="My Application" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Tasarım kısmında ise Palette dediğim yerden çektiğimiz (sürükle bırak) ile butonlar, yazıları, layout ya da etiket gibi bir sürü elemanı kullanabilirsiniz. Sağ alt tarafta properties kısmında aldığınız elemanlara ait düzenlemeler yapabilirsiniz.(Boyutu, Rengi gibi)

Elemanları içinde bulundurduğumuz kısım ise activity yani sayfa diyebiliriz. Bir sayfa stack mantığı ile çalışır. En son eklenen sayfa en üsttedir.





Geldik java kısmına. onCreate kısmı bizim bulunduğumuz activity'de (tasarım ekranımız) 'da işlemleri gerçekleştirdiğimiz kısımdır. onOptionsItemSelected ile menü ayarlamalarını yapabiliyoruz.

```
MainActivity.java × activity_main.xml × AndroidManifest.xml ×
package com.example.myapplication;

import ...

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

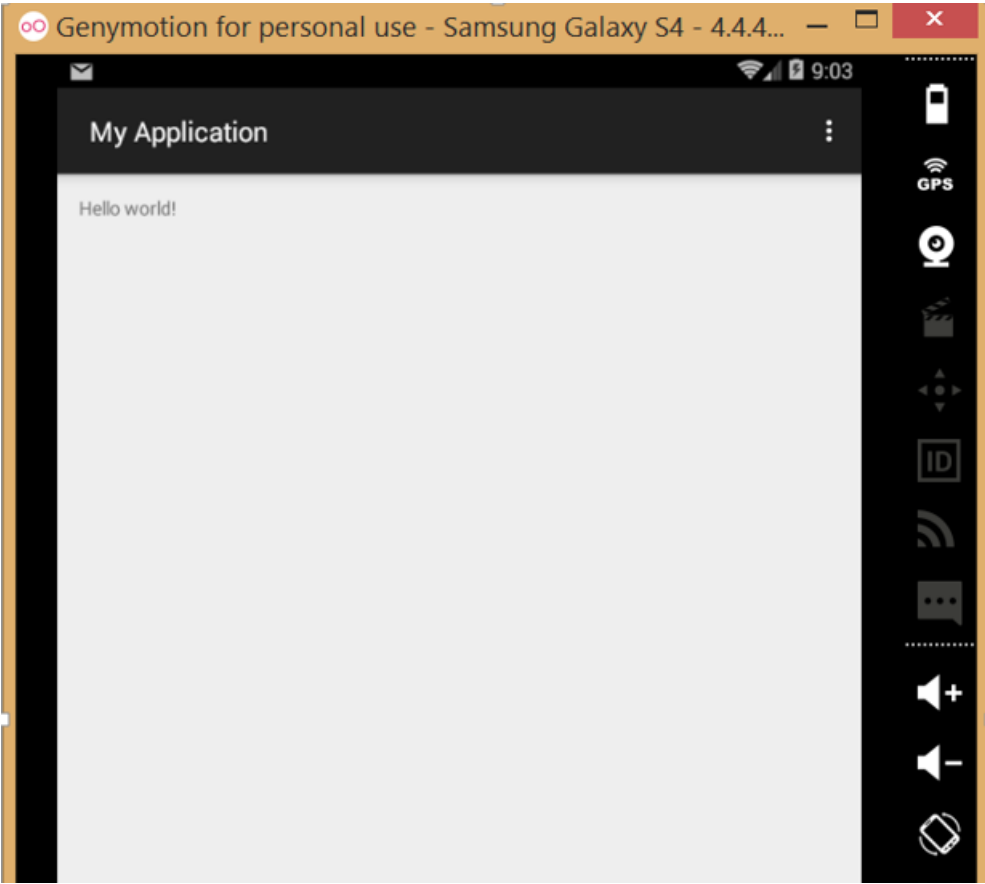
Ben android studiodaki projelerimde emülatör olarak genymotion kullanıyorum. [Kurulumu için](#)

Android studioda ki emülatörü kullanmıyorum. Sebebi ise tabiki de genymotion hem çok hızlı hemde pratik. Sizde öneririm.

Emülatör yaptığımız projeleri test etmemiz için sanal bir android işletim sistemini barındırdığımız programdır. Projemizi burada test edeceğiz.

Projemizi play tuşuna basarak çalıştırabilirsiniz.

İlk uygulamanızı yaptınız.Hello World ☺



## Android Studio Programlama Bölüm 3

### Buton Tanımlama ve Android Manifest Dosyası

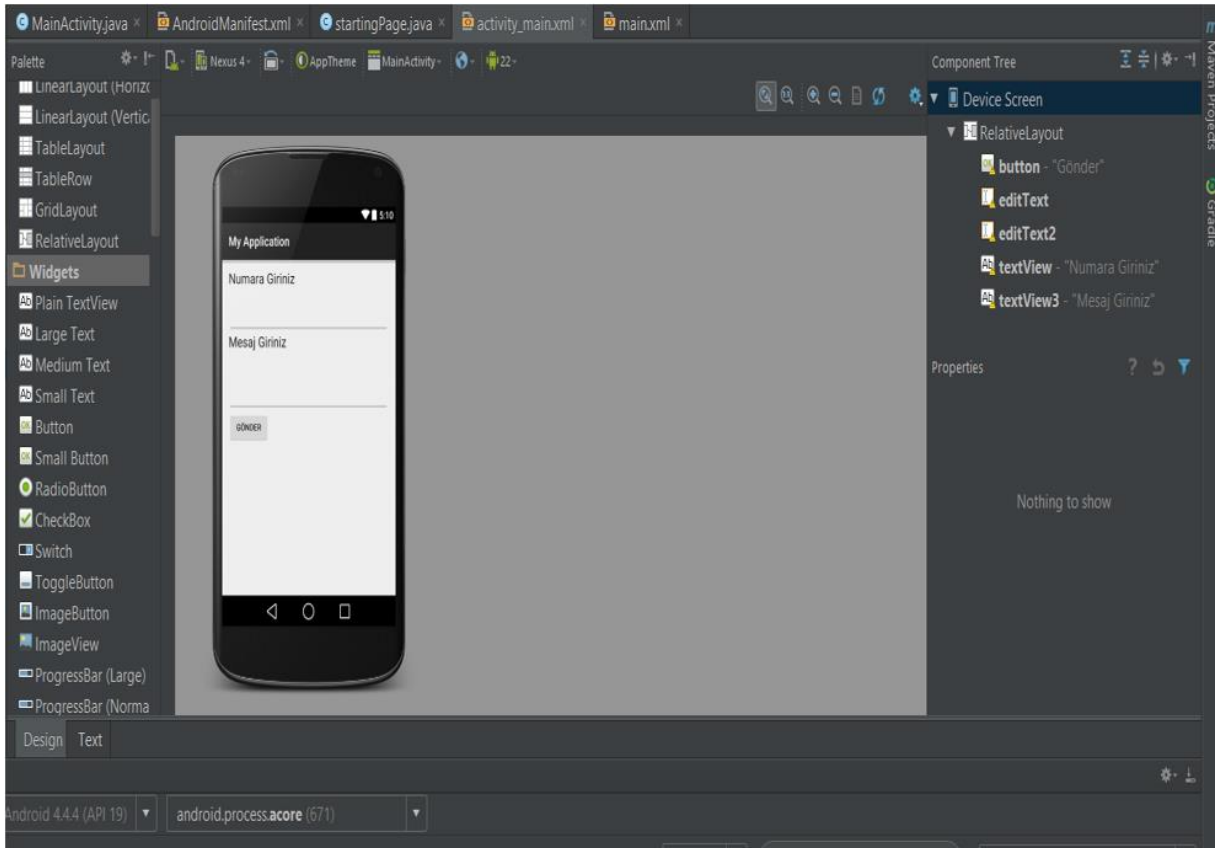
Bu dersimizde android'le ilgili sms uygulaması yapacağız. Sebebi ne dersiniz aslında görüldüğünden daha kolay bir programlama olduğunu göstermek. İlerdeki konularada bir giriş niteliği taşıması.

Öncelikle 2 tane textView tanımlayıp 2 adet edittext'i 1 tane button aktif ederek gönderceğiz. Bu dersten öğrenecekleriniz.

- Android Manifest Dosyasını yakından incelenilecek
- Mesajlaşma kısmını android üzerinden nasıl gerçekleştiği
- Bir butona işlevsellik vererek onu aktif etme

#### Daha da basit indirgemek için tasarım diyorum :

Soldaki paletta kutusu yani tasarım araçlarının bulunduğu alandan 2 tane text View 2 tane editText ve 1 tane button alarak ekliyoruz.(Sürükle bırak ile ekrana sürükleyin.) Basit bir tasarım oluşturduk bile.



Android'te kodlarımızı java da ki MainActivity sınıfımıza yazıyoruz. Orada öncelikle 2 tane editText i tanımlayacağız.

Final değişkenini açıklarsam öncelikle tanımlanan ilk değişken değeri aldıktan sonra değiştirilmemesidir. Telno olarak kalıcak. Aynı mesaj da öyle. Kullanmasanız da olur.

Sıra Button olayının aktifleşmesine gelirse:

**Button gonder=(Button)findViewById(R.id.button);** Kalıbını kullanacağız. Button'u idsini çekiyoruz. Gonder değişkenine eşitliyoruz.

```
gonder.setOnClickListener(new view.OnClickListener) {
```

```
public void onClick(View v){
```

Burası ise button'a tıkladınızda ne olacak kısmıdır. Button'un işlevselliğini burada tanımlıyoruz.

```
}}
```

Button tanımlanması böyle. Ben Smsmaneger'ı aktif etmesini ve editTextlerdeki verileri çekerek mesajı göndermesini istedim. Olayımız bu kadar

```
public class MainActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        final EditText telno=(EditText)findViewById(R.id.editText);  
        final EditText mesaj=(EditText)findViewById(R.id.editText2);  
        Button gonder=(Button)findViewById(R.id.button);  
        gonder.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                // TODO Auto-generated method stub  
                android.telephony.SmsManager sms=android.telephony.SmsManager.getDefault();  
                sms.sendTextMessage(telno.getText().toString(), null, mesaj.getText().toString(), null, null);  
            }  
        });  
    }  
}
```

Ve gerekli olan son bir şey ise en sonda daha yakından inceliyeceğimiz Android Manifest dosyamız. Android sistemi bir uygulama bileşenini başlatmadan önce, uygulamanın AndroidManifest.xml dosyasını okuyarak bileşenin varlığını kontrol etmelidir. Bir uygulamaya ait bütün bileşenlerin uygulamanın ana proje dizininde olması gereken bu dosya içinde bildirim yapılmalıdır.

## **Manifest dosyası uygulamaya ait bileşenlerin bildirimine ek olarak aşağıdaki işlemleri de yapar:**

- > İnternet girişi veya kullanıcı bilgilerinin sadece okunur girişi gibi kullanıcı izinlerini belirlemek.
- > Uygulamanın gerektirdiği minimum API seviyesini bildirmek.
- > Uygulamanın gerektirdiği veya kullandığı donanım ve yazılım özelliklerini bildirmek (kamera, bluetooth hizmeti gibi).
- > Android API'leri dışında uygulamanın gerektirdiği API kütüphaneleri (Google Maps kütüphanesi gibi).

## **Tüm uygulama bileşenlerinin bildirimini benzer şekilde yapılmalıdır:**

- <activity> Aktivite elemanları
- <service> Hizmet elemanları
- <receiver> Mesaj alıcıları elemanları
- <provider> İçerik sağlayıcı elemanları

## **Hızlı işlemler :**

application tagları arasında biz kendi activityimizi koyuyoruz. İzin alırken **<uses-permission android:name="Alıncak isim"/>** şeklinde alıyoruz.

activity android:name ve labeli kullanarak istersek yeni bir activity için izin alabiliriz. Sonraki kısımlarda değinelecektir.

```
MainActivity.java x AndroidManifest.xml x startingPage.java x activity_main.xml x main.xml x
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication" >

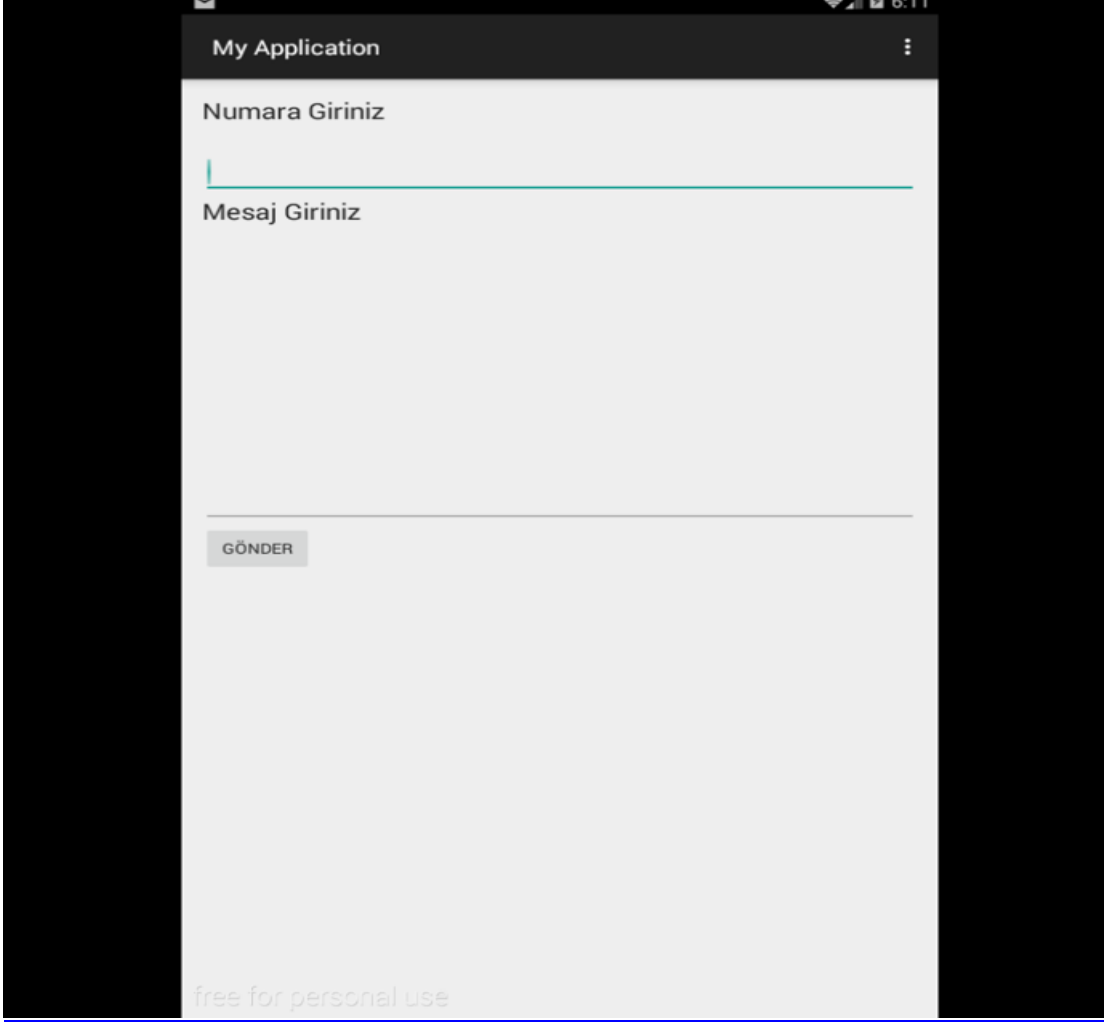
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:theme="@style/AppTheme" >
        <uses-permission android:name="android.permission.SEND_SMS"/>
        <activity
            android:name=".MainActivity"
            android:label="My Application" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".startingPage"
            android:label="My Application" >

        </activity>
    </application>
</manifest>
```

Uygulamayı çalıştırdığınız da işlem yapmadığını göreceksiniz. Sebebi ise emülatör'ün içinde herhangi bir sms göndermek işlev yapacak birim yok.. Telefonda denediğinizde sonuca ulaşacaksınız. Uygulamanın emülatörde çalışması. Telefonda çalıştırmak içinde apk dosyası yaratmalısınız. [Apk dosyası kısmı için tıklayın.](#)

Bu dersimizde android ile ilgili bir uygulamanın nasıl yapıldığını, sürecin nasıl işledi, bir uygulamada izin almaya ilk adımı attık ve güzel bir uygulama yaptık. Aşağıdaki dersleri de inceleyerek daha da ileri seviye uygulama yapacağız. Bu dersi şu sebepten işledim tasarım olsun, kodlama olsun çok kolay olmasada android'i öğrenmek kolay ve hızlıdır.



**Not:**

**findViewById:** Onun idsiyle xmlde olan görünümü bulur ve nesnemizi gösterir..

**R.id :** bizim herşeyi içinde barındıran üzerinde pek değişiklik yapmadığımız gen classından geliyor



## Android Studio Programlama Bölüm 4

---

### Intent Kavramı ve Yeni Ekran Oluşturma

#### Öncelikle Intent Nedir?

Türkçe anlamı olarak amaç'tır. Ben buna amacınız ne ? Diyorum. Yapacağımız uygulamanın yapısına göre farklı aktivite sınıfları oluşturup, intent'ler aracılığı ile bunlar arasında geçiş yaparız. Android işletim sistemi olarak bir aktivite'nin yada servisin vb. birbirlerini çağırmasını ve iletişime geçebilmesini sağlamak için ortak bir yapı oluşturmuştur. Bu yapıya intent denmektedir. Intent'ler aracılığı ile başka bir aktivite'nin çalıştırılması istenebilir, hatta o aktiviteye bazı bilgiler gönderilip, sonucu istenebilir. Intent'ler ile başka kategoride uygulamalarda çalıştırılması istenebilir, mesela uygulamanızda bir web sitesini görüntülemek istiyorsanız, bir web tarayıcı yazmak zorunda değilsiniz. Siz web tarayıcı açmak istediğinizi söylersiniz, gerekli bilgileri verirsiniz, Android uygun olan programı açar, eğer birden fazla seçenek var ise kullanıcıya sorar.

Intentler ikiye ayrılır, dolaylı intentler ve doğrudan intentler. Dolaylı intent'lere örnek olarak web tarayıcı örneğini verebiliriz. Doğrudan intentler ise doğrudan isim vererek (kendimizin kontrol edebildiği ) uygulamalardır. Şimdi doğrudan intent örneği ile ekran geçişlerimizi nasıl yaptığımızı göreceğiz.

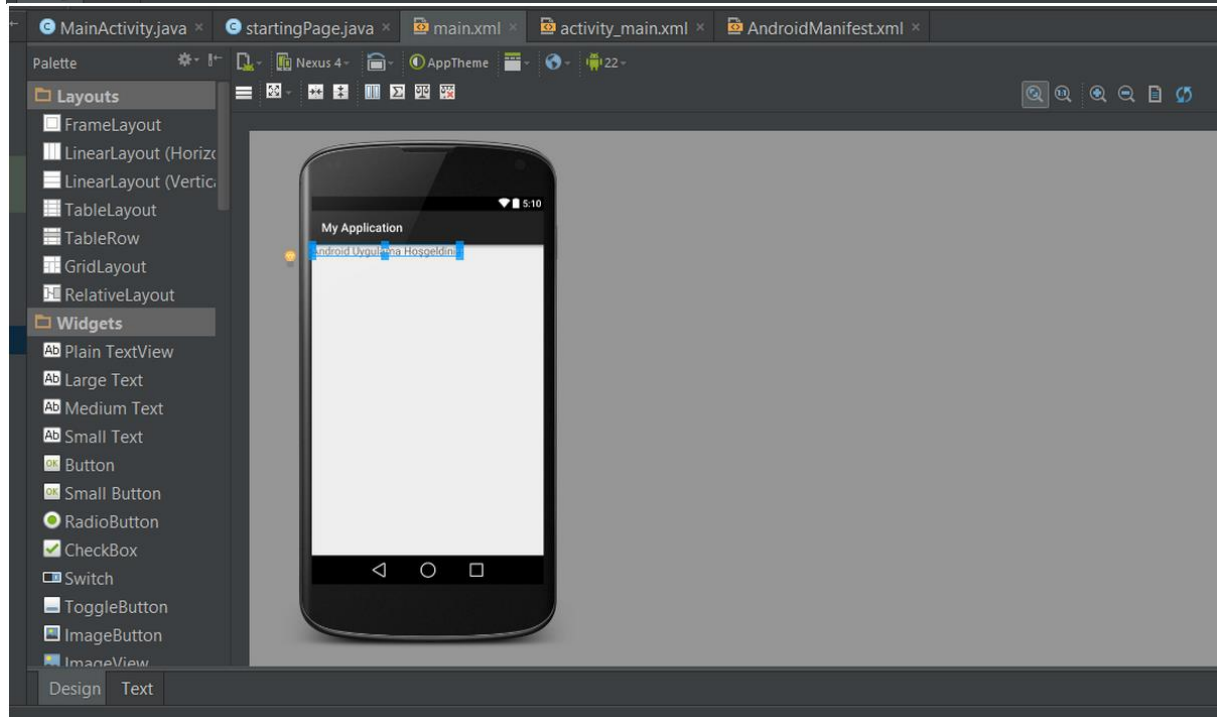
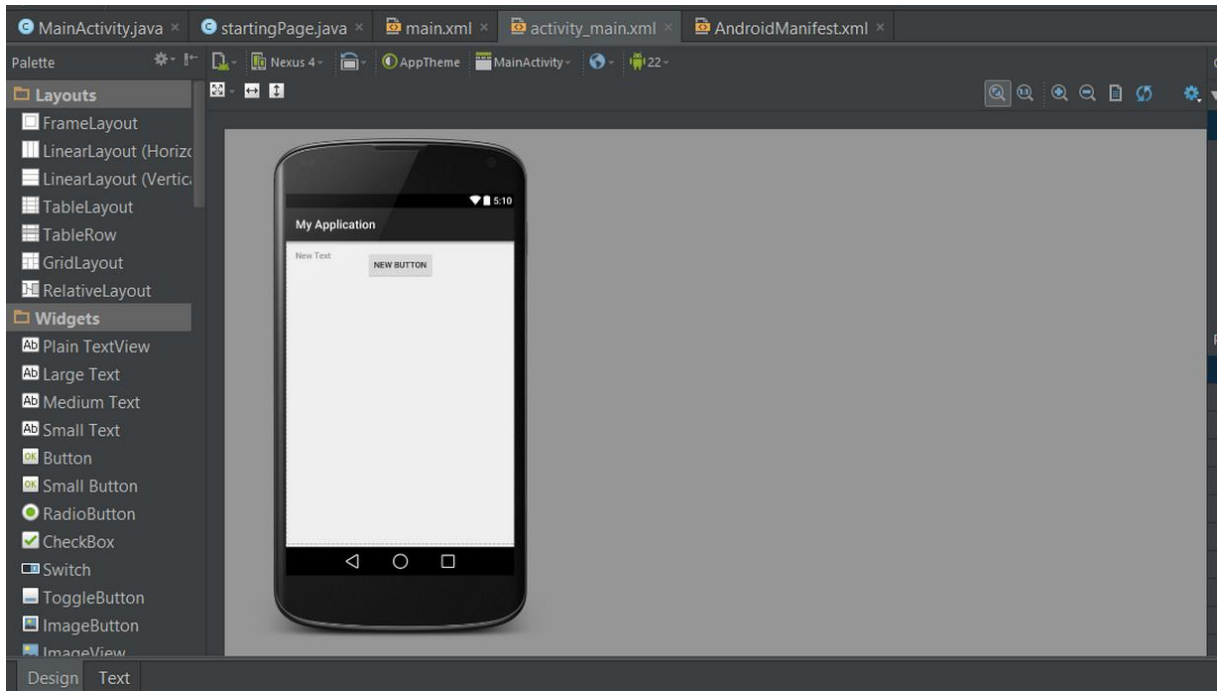
Her zamanki gibi yeni projemizi açtık. Sonrasında yeni bir activity ve java dosyamızı oluşturacağız.

Öncelikle yeni bir activity oluşturacağız. (**layout -> Sağ tık -> new -> Layout Resource file**)

sonra java dosyamızı oluşturalım. Java dosyasıyla da yeni oluşturduğumuz layoutu kontrol edeceğiz. (**Java -> sağ tık -> New -> Java class**)

Uygulama çok basit olacak. Bir tuşa basıcaksınız ve diğer ekrana ulaşacaksınız.

Öncelikle ilk activity'e bir tane buton atın. 2. Yeni oluşturulan activity'e ise Android uygulamasına Hoşgeldiniz başlıklı bir tane text atın.



## 1. Java dosyamızda ise setContentView altına

```

Button button=(Button)findViewById(R.id.button); // activitydeki butonu bul
    button.setOnClickListener(new View.OnClickListener() {
//butona dinleme olayı ver
        @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, startingPage.class);
//Butona basınca intent ile bu activityden yeni activityye git
            MainActivity.this.startActivity(intent);//intent'i başlat
            MainActivity.this.finish();
        }
    });

```

```
}  
});
```

yapıştırın. Burada öncelikle bir butonu aktik ettik. Yani bir dinleme verdik. Butona bir şey olunca olay olunca onun içindeki işlemi gerçekleştir diye düşünebilirsiniz. Intent intent=new Intent(Butonun bulunduğu kısım, butonun gideceği class) kısmını hep bu şekil olur.Resmi ;

```
import android.widget.Button;  
  
public class MainActivity extends ActionBarActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Button button=(Button)findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                Intent intent = new Intent(MainActivity.this, startingPage.class);  
                MainActivity.this.startActivity(intent);  
                MainActivity.this.finish();  
            }  
        });  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.menu_main, menu);  
        return true;  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) {  
        // Handle action bar item clicks here. The action bar will  
        // automatically handle clicks on the Home/Up button, so long  
        // as you specify a parent activity in AndroidManifest.xml.  
        int id = item.getItemId();  
    }  
}
```

Şimdi 2. oluşturduğumuz ve benim adını startingPage verdiğim java dosyama doldurmaya.

```
package com.example.myapplication;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.widget.Toast;  
  
/**  
 * Created by um on 20.7.2015.  
 */  
public class startingPage extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

```
}  
}
```

Aynen bu şekil olacak. setContentView'de ki main benim layoutta yeni oluşturduğum dosyamın adı. Resmi ;

```
package com.example.myapplication;  
  
import android.app.Activity;  
import android.os.Bundle;  
import android.widget.Toast;  
  
/**  
 * Created by um on 20.7.2015.  
 */  
public class startingPage extends Activity {  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

Bunları yaptıktan sonra uygulamayı çalıştırın. İlk sayfa geldiğini ve button'a tıklayınca kapandığını göreceksiniz. İzin kısmını almadık. Android Manifest Dosyasında şimdi yeni bir activity için izin almamız gerekiyor.

```
<activity  
    android:name=".startingPage"  
    android:label="@string/app_name" >  
</activity>
```

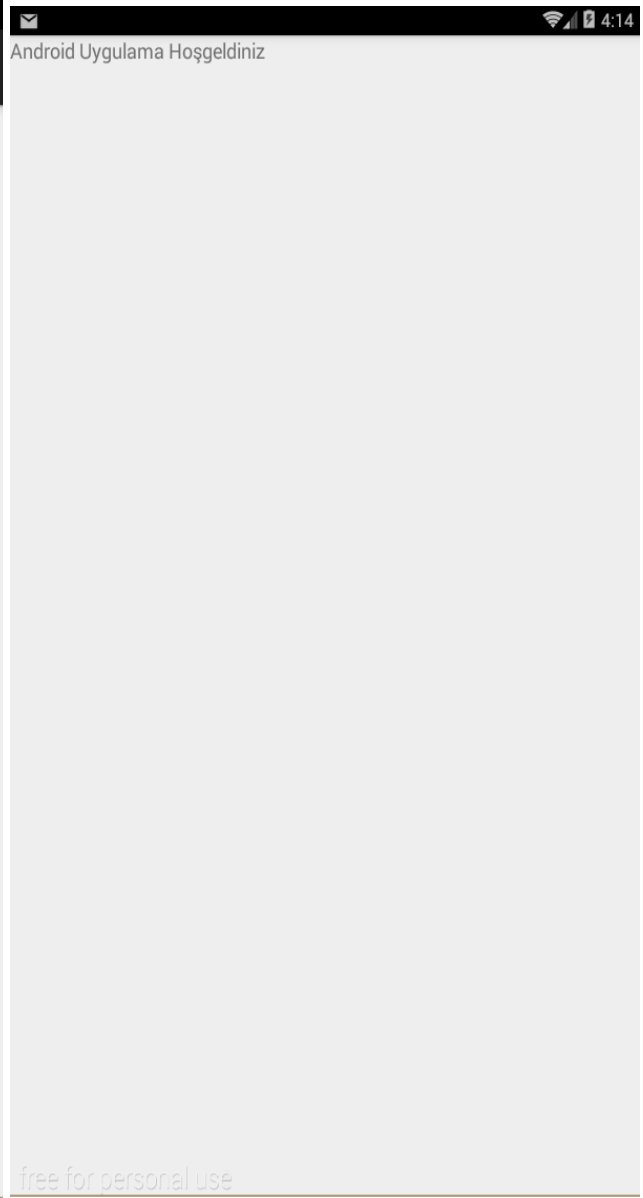
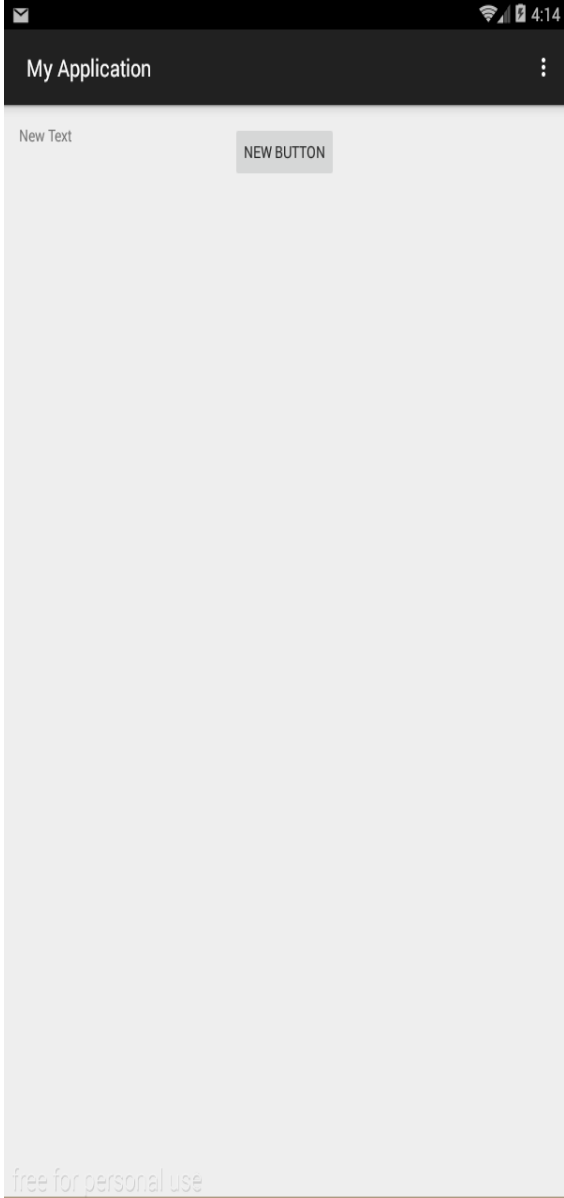
Genel kavram olarak göstereceğim.

```
<activity
```

```
    Android:name=".classİsmin"
```

```
></activity>
```

Uygulamanızı çalıştırın. Aşağıdaki görüntüyü elde ediceksiniz..



## Android Studio Programlama Bölüm 5

---

### Thread Kavramı ve Splash Ekranı (Android Studio Giriş Sayfası yapma)

Android dersimizde Splash ekran ve thread kavramını göreceğiz. Thread iş parçacığı demektir. Bir programın paralel olarak çalışan ve birbirine bağımlı yada bağımsız işlemler yapan alt parçacıklara verilen isimdir. 2 sayfa oluşturacağız. Yukardaki derste bunun nasıl yapıldığını öğrenmiştik.

Öncelikle 1 tane layout dosyası oluşturcaz. (**layout -> Sağ tık -> new -> Layout Resource file**) ve 1 tane da java dosyası oluşturcaz. (**Java -> sağ tık -> New -> Java class**)

Oluşturduktan sonra android manifest'te yeni bir tane java dosyası oluşturduğunu haber vericez. Oluşturduğumuz java dosyasının ismini name="Java dosyanız" ve etiketinizi burada oluşturuyorsunuz.

```
<activity|
    android:name=".startingPage"
    android:label="@string/app_name" >
</activity>
```

Java dosyamıza bir thread aktif edelim. setContentView altına şu kodları atıcaksınız.

```

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Thread MyScreen = new Thread() {
            public void run() {
                try {
                    sleep(5000);
                    //5 saniye sonunda class davet et.
                    startActivity(new Intent(getApplicationContext(), startingPage.class));
                }
                catch (InterruptedException e)
                {
                    e.printStackTrace();
                }
                finally
                {
                    finish();
                }
            }
        };
        MyScreen.start();
    }
}

```

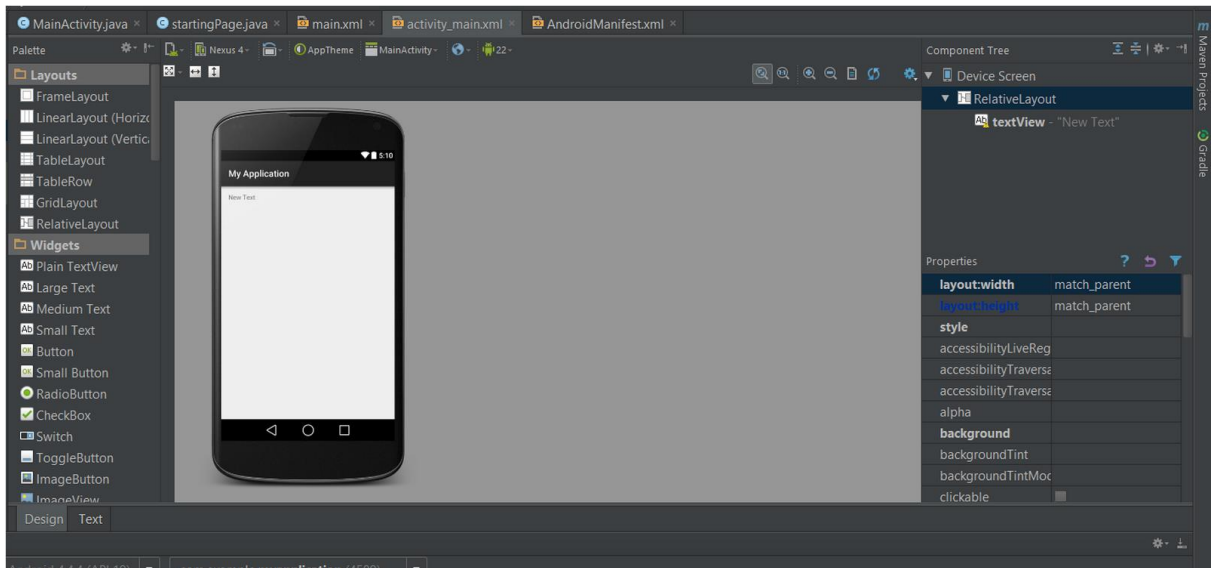
Kolaylık olsun diye kodlar..

```

Thread MyScreen = new Thread() {
    public void run() {
        try {
            sleep(5000);
            //5 saniye sonunda class davet et.
            startActivity(new Intent(getApplicationContext(), startingPage.class));
        }
        catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        finally
        {
            finish();
        }
    }
};
MyScreen.start();
}

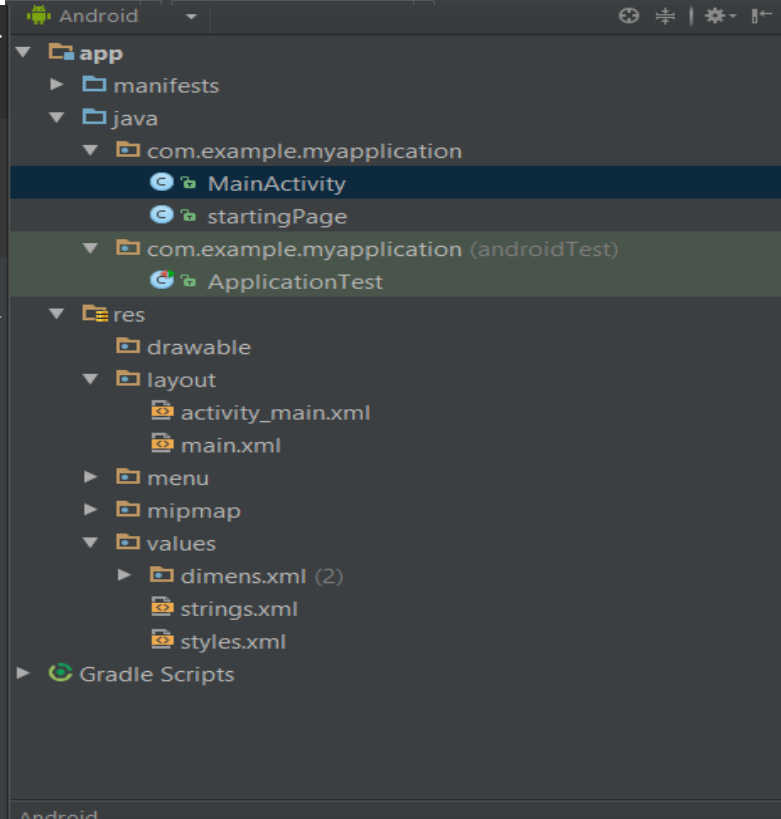
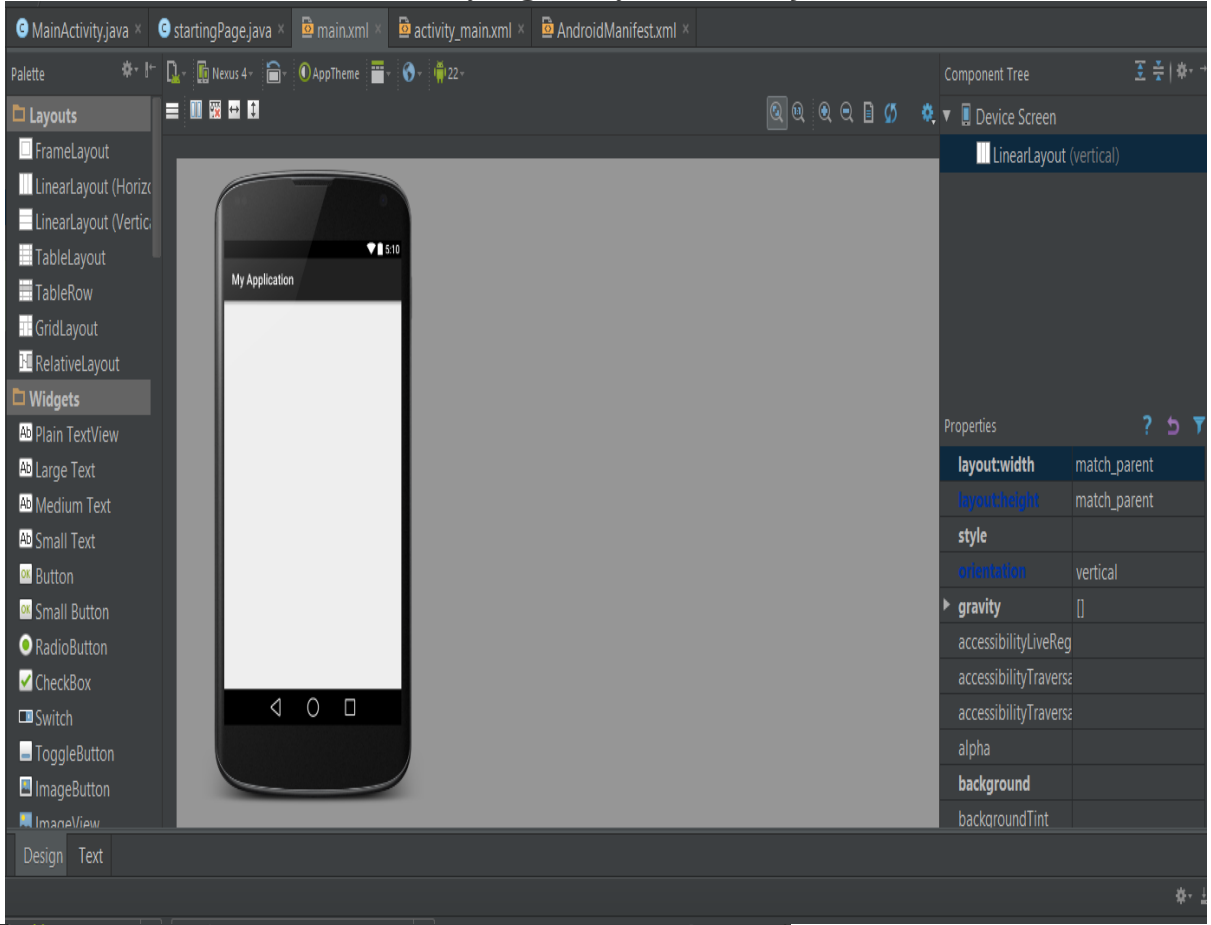
```

activity\_main dosyamda bir tane text attım.





Main adını koyduğum layoutumu boş bıraktım.



Eğer dosyaları merak ediyorsanız soldaki resim gibi olucak

Ve çalıştırdıktan 5 sn sonra yeni boş activity gelecek. Yakında içlerini de dolduracağız.

## Android Studio Programlama Bölüm 6

# Uygulamayı tam ekran yapma

Android studio'da uygulamasını tam ekran yapmak isteyenler olacaktır. Bildirim çubuğu kısmından kurtulmak için izlenecek yollar.

1- Main activity'iz kalıtım olarak Activity sınıfından almalıdır. Yani gösterimi :  
public class MainActivity extends Activity {

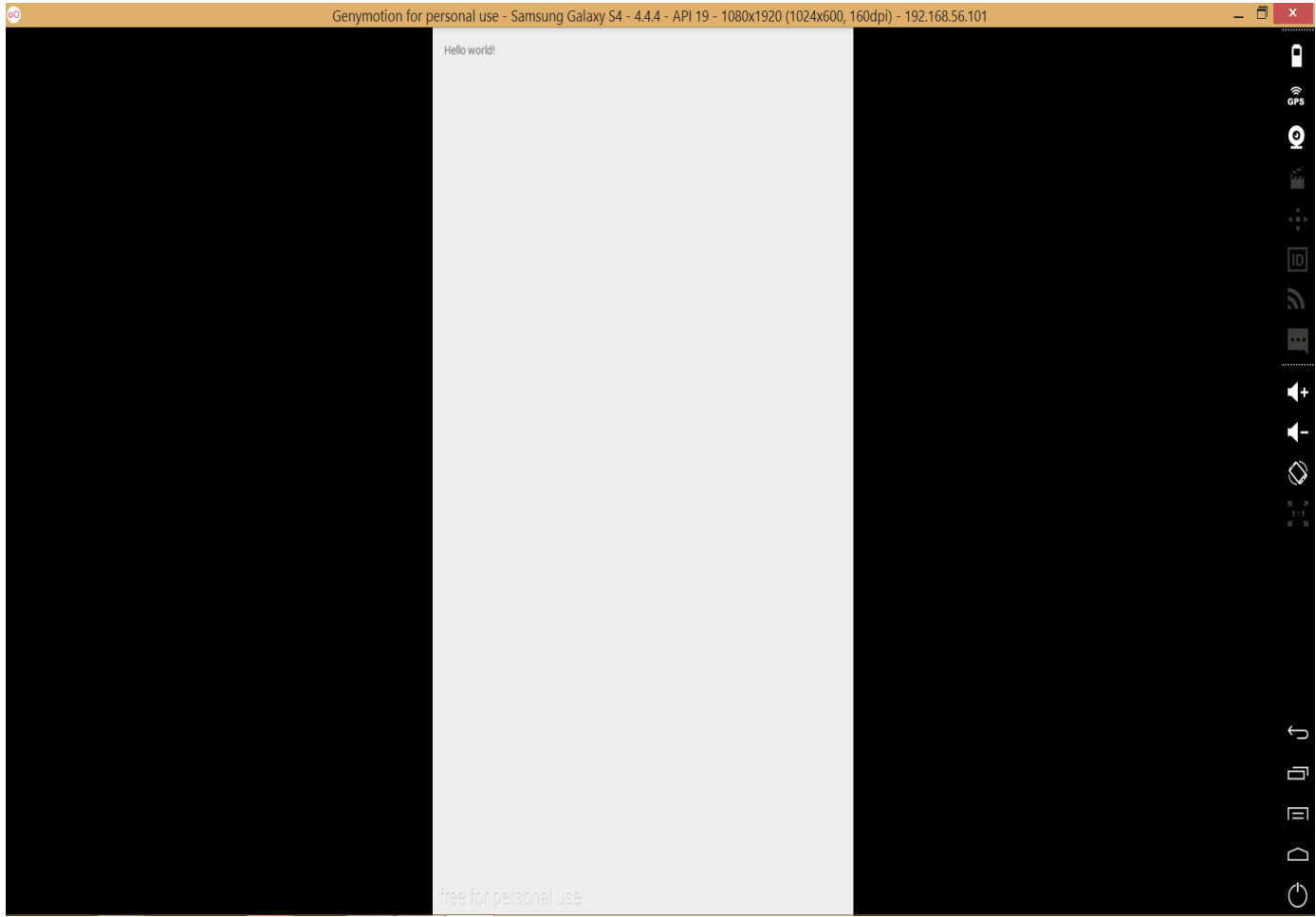
2- Hangi layoutta kullanacaksanız öncelikle setContentView(layout)'an önce kodumuzu yapıştırıyoruz.

```
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
    WindowManager.LayoutParams.FLAG_FULLSCREEN);
```

3- Uygulamayı çalıştırdığımızda tam ekran olduğunu görüyoruz.

Resimler :

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
            WindowManager.LayoutParams.FLAG_FULLSCREEN);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        // Inflate the menu; this adds items to the action bar if it is present.  
        getMenuInflater().inflate(R.menu.menu_main, menu);  
        return true;  
    }  
}
```



# Android Studio Programlama Bölüm 7

## Button Still'i Tanımlama

Android'te button still'i dediğimiz olay bir button'a tıklayınca şeklinin değişmesidir.



Mesela Yandaki 2 simgeden birine tıkladığımızda diğeri gelicek.

Öncelikle bu 2 resmi kaydedin ve projenizin dizinine gelerek res klasöründe ki drawable'ye atın. Çok karmaşık isimler de sorun çıkabiliyor. Ben olsam a b gibi basit harfler kullanırım. Sonra drawable klasörüne sağ tık **Drawable Resource file** oluşturun. **Drawable -> Sağ Tık -> Drawable Resource file**

Bir xml dosyası oluşturup tıklama olaylarını gözlemliyeceğiz. Gonder.xml tanımlayın.

Resim :

```
MainActivity.java x AndroidManifest.xml x startingPage.java x activity_main.xml x gonder.xml x
<?xml version="1.0" encoding="utf-8" ?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:state_enabled="true"
        android:state_pressed="false"
        android:drawable="@drawable/a"/>
  <item android:state_enabled="true"
        android:state_pressed="true"
        android:drawable="@drawable/b"/>
</selector>
```

Kodlar:

```
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:state_enabled="true"
        android:state_pressed="false"
        android:drawable="@drawable/a"/>
  <item android:state_enabled="true"
        android:state_pressed="true"
        android:drawable="@drawable/b"/>
```

</selector>

Sonraki işlemde ise yapacağımız olay style.xml de ( values'in orada ki)

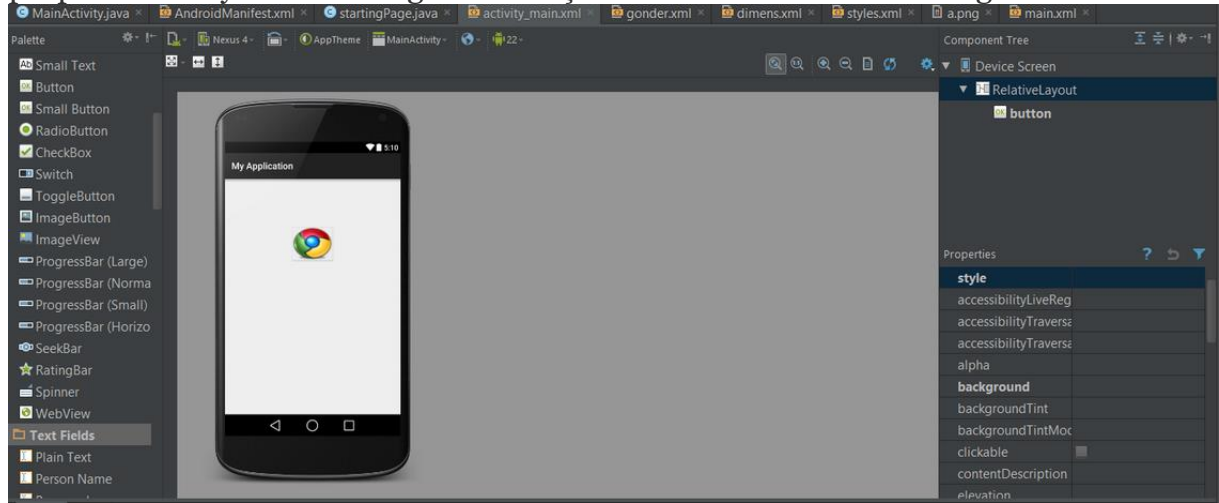
Resim :

```
</style>
<style name="gonder" parent="@android:style/Widget.Button">
  <item name="android:background">@drawable/gonder</item>
</style>
</resources>
```

Kodlar :

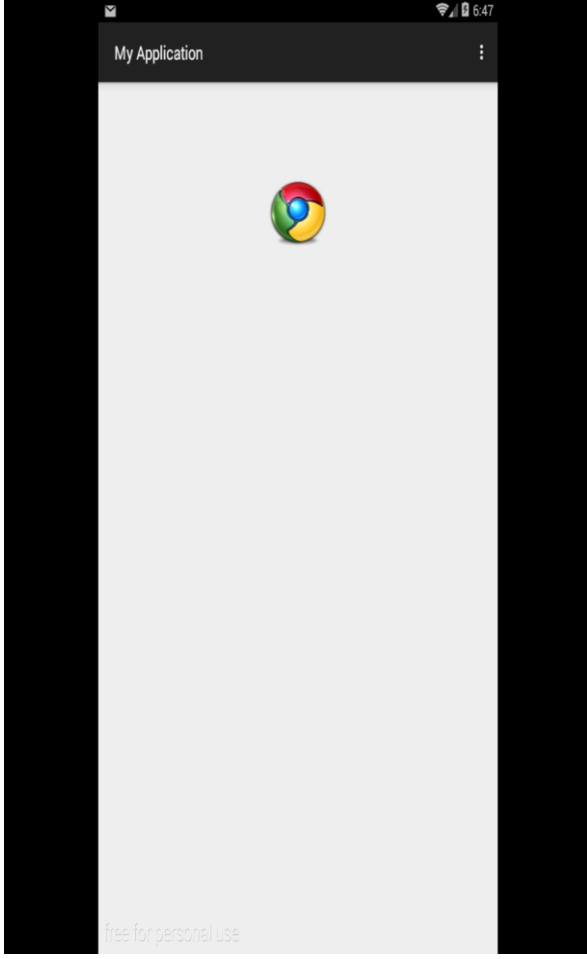
```
<style name="gonder" parent="@android:style/Widget.Button">
  <item name="android:background">@drawable/gonder</item>
</style>
```

Bir sonra ki adım ise activity\_main.xml'de bir button nesnesi atamak ve içeriğini boş bırakmak. F2 tuşu ile id ve nesnenin içeriğini ayarlayabilirsiniz. Sonra ise sağ altta ki properties'ta style kısmında gonder'i seçmek ve resim olarak 1 resim gelecek.

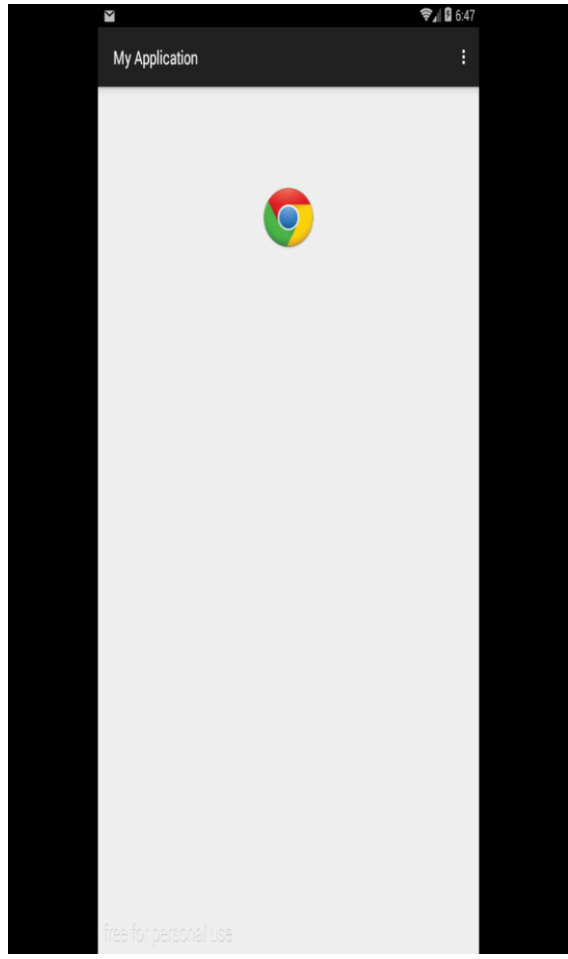


Uygulama bu kadar çalıştırılmı..

Tıklamadan Önce



Tıkladığımız da



# Android Studio Programlama Bölüm 8

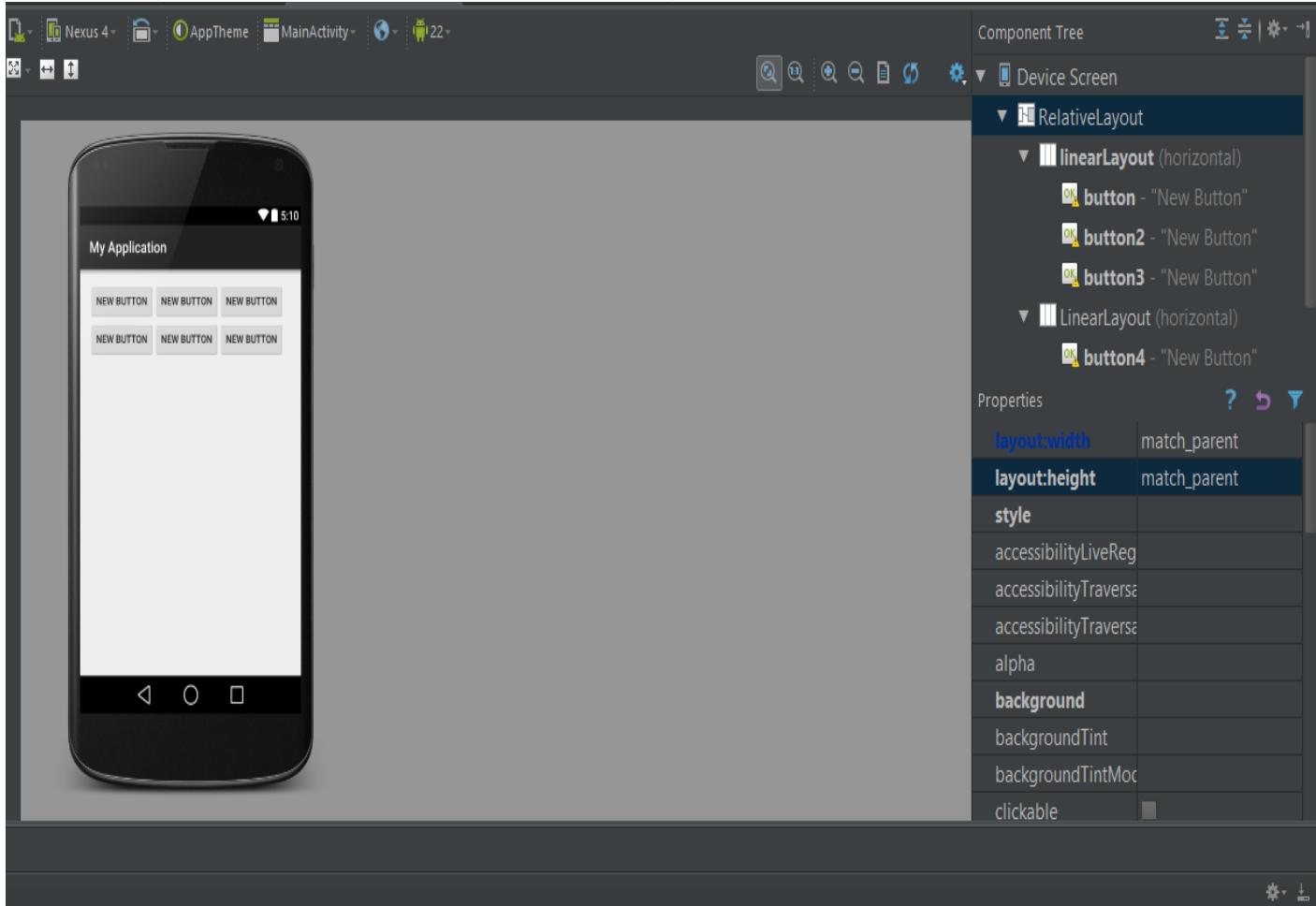
## Layoutlar

Android dersi için programlama kadar önemli bir konu varsa oda tasarım'dır. Tasarım'da bugün Layout yani nesnelimizi, itemlerimizi doğru ve güzel bir şekilde kullanıcılarımıza sunduğumuz yeri görecekiz.

### 1-)LinearLayout'lar:

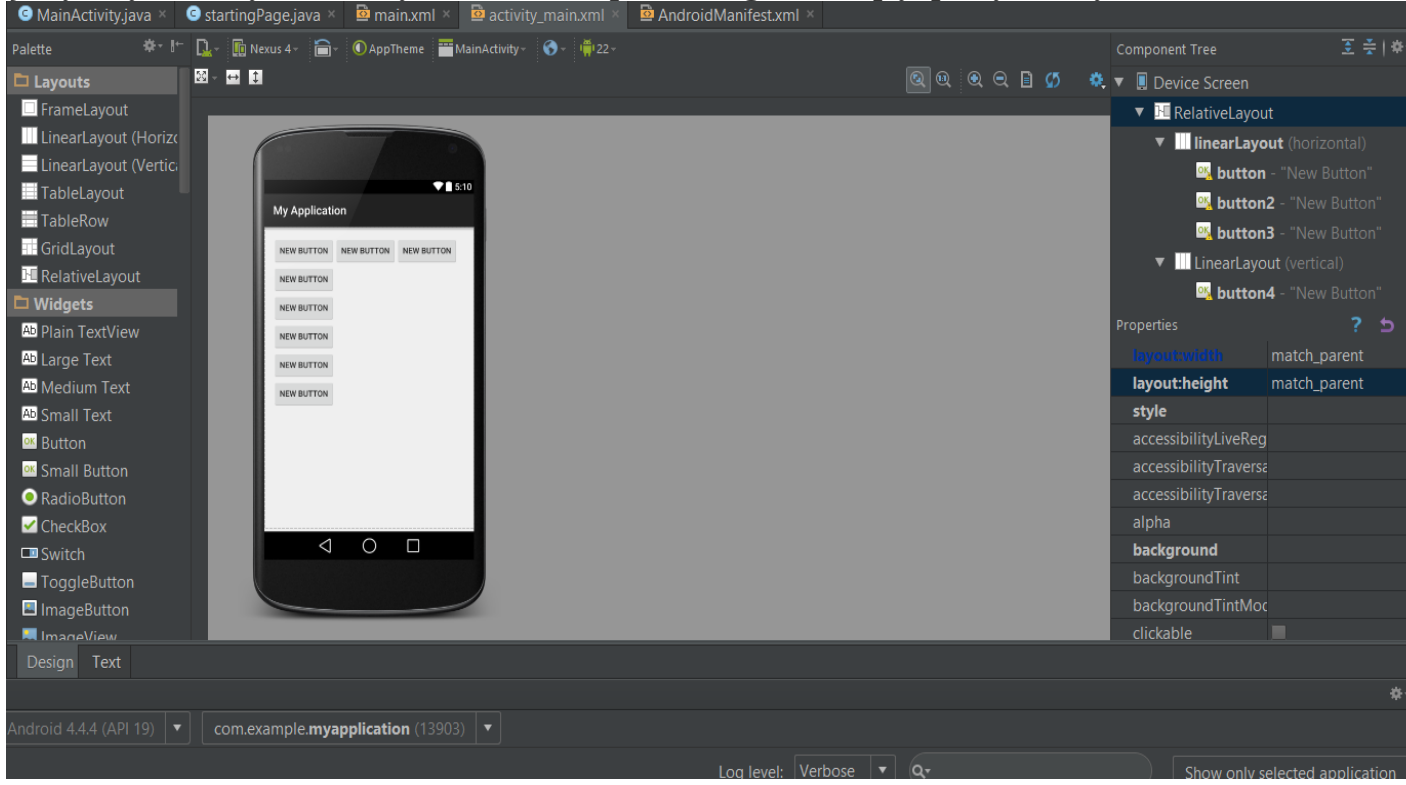
Nesnelerimiz doğrusal eklenmektedir. Eğer **vertical linear layout** kullanırsak alt alta, **horizontal linear layout** kullanırsak yan yana eklenir.

Horizontal'ı kullanarak düzgün bir şekilde layout tasarımı gerçekleştirdim. Aynıısını sizde yapabilirsiniz. 2 tane linear layout kullandım..



Peki horizontal ile vertical arasındaki fark ne ? Biri yan yana dizerken diğeri vertical yani alt alta nesneleri yerleştirir. Eğer bir layout atıp tüm ekranı kapattığından

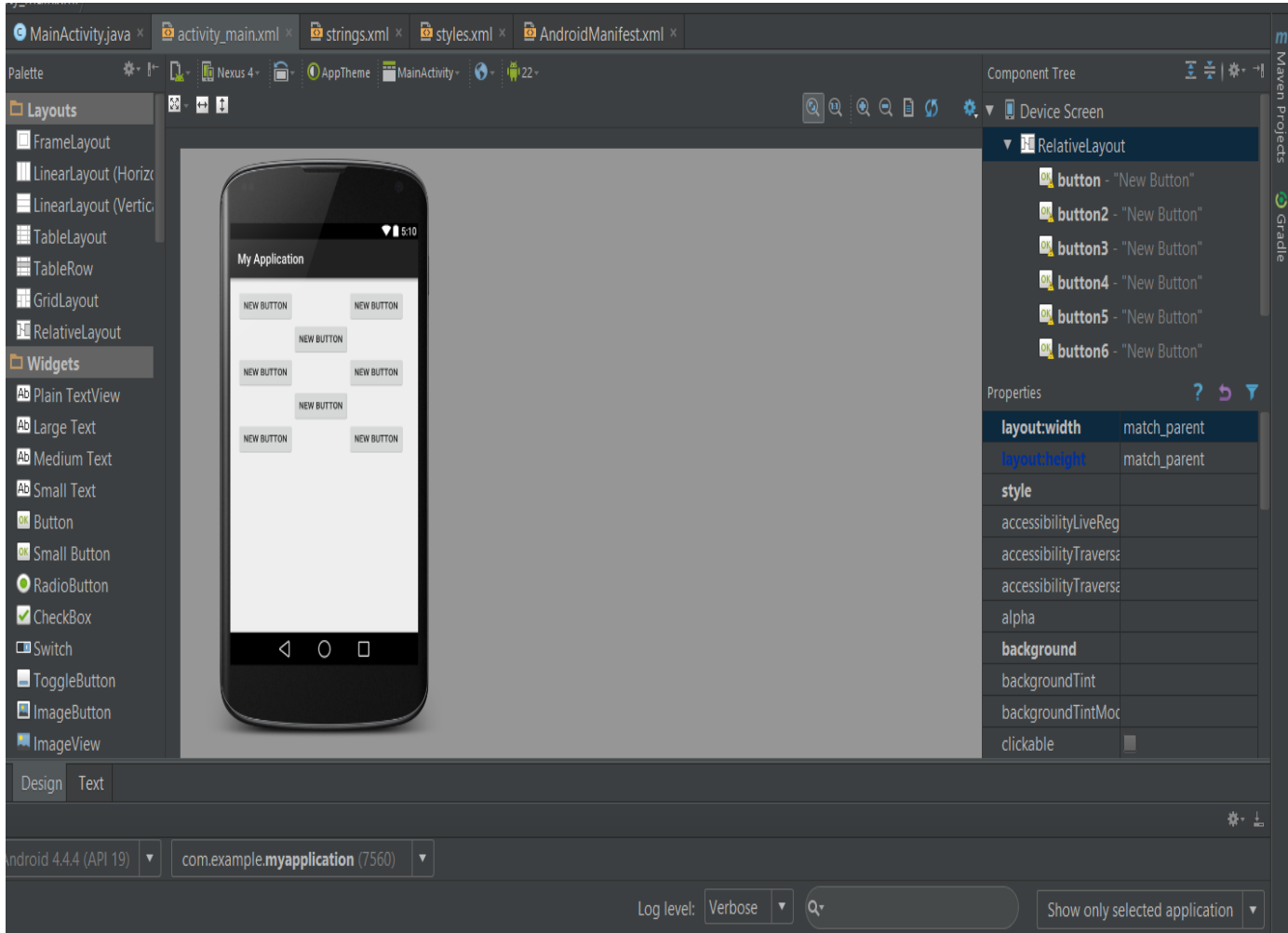
şikayetciyseniz layout'un boyutunu `match_parent` değilde wrap yapmayı deneyin.



## 2-)Relative Layout:

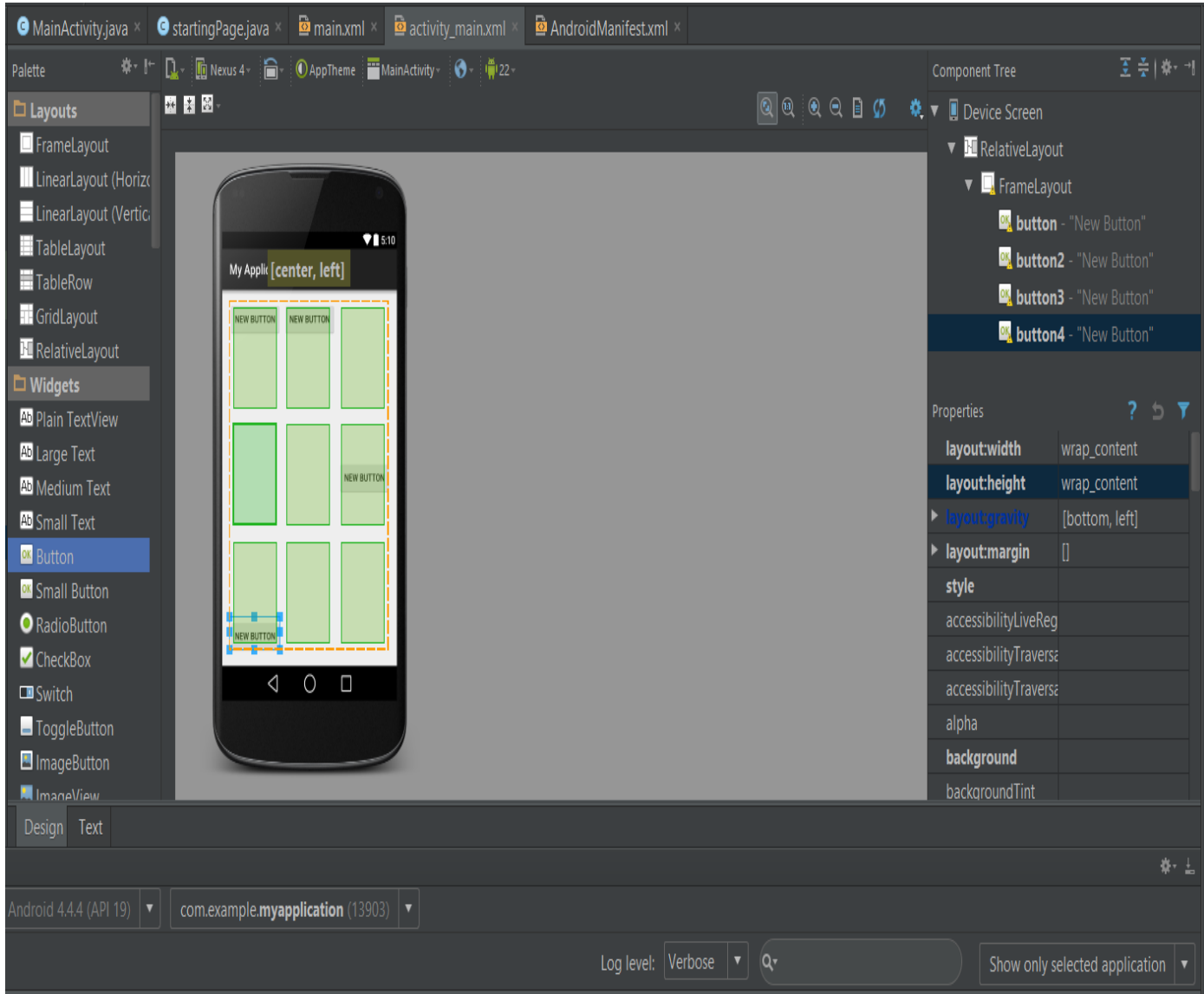
Otomatik sayfa açıldığında gelen Layouttur. Yani nesnemizi(Burda Buton) istediğimiz yerde kullanabiliriz..Ve benim en çok kullandığım bir layout çeşididir. Direk istediğimiz yerde nesnelerimizi ayarlayabiliyoruz.





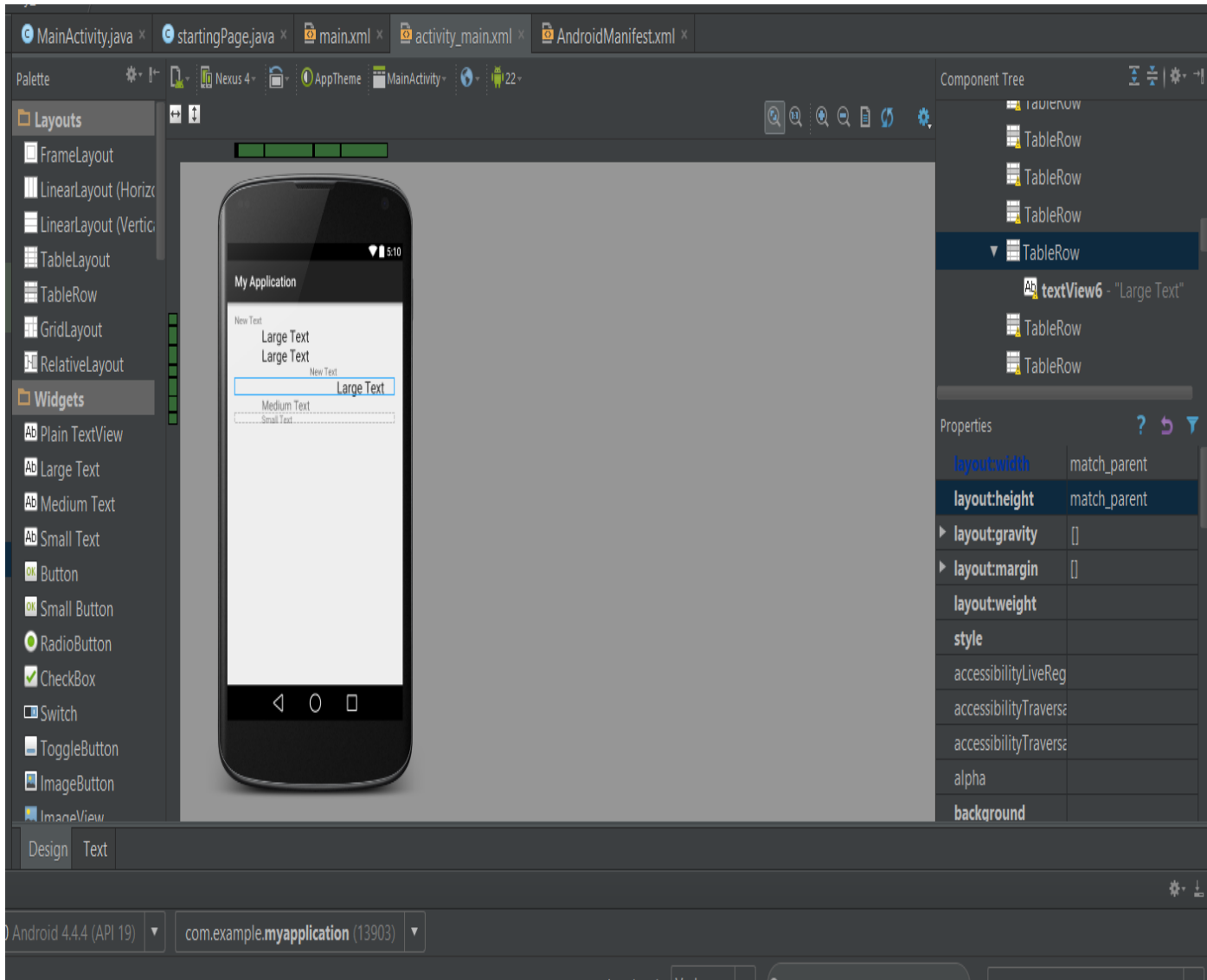
### 3-)FrameLayout:

Bu layout tipinde nesneler üst üste biner. Genelde aynı boyutlardaki butonların üst üste gelip, birinin kaybolduğunda diğerinin gözükmesi için kullanılır. Bunun için, bir önceki örnekte xml kısmında **LinearLayout** yazısını silip **FrameLayout** yazabiliriz ya da araç panelinden tekrar bir **FrameLayout** ekleyebiliriz. Tabiki yeniden eklemek daha iyi olacaktır.



#### 4-)TableRow:

Bir tablo düzeninde nesnelimizi yerleştiririz.

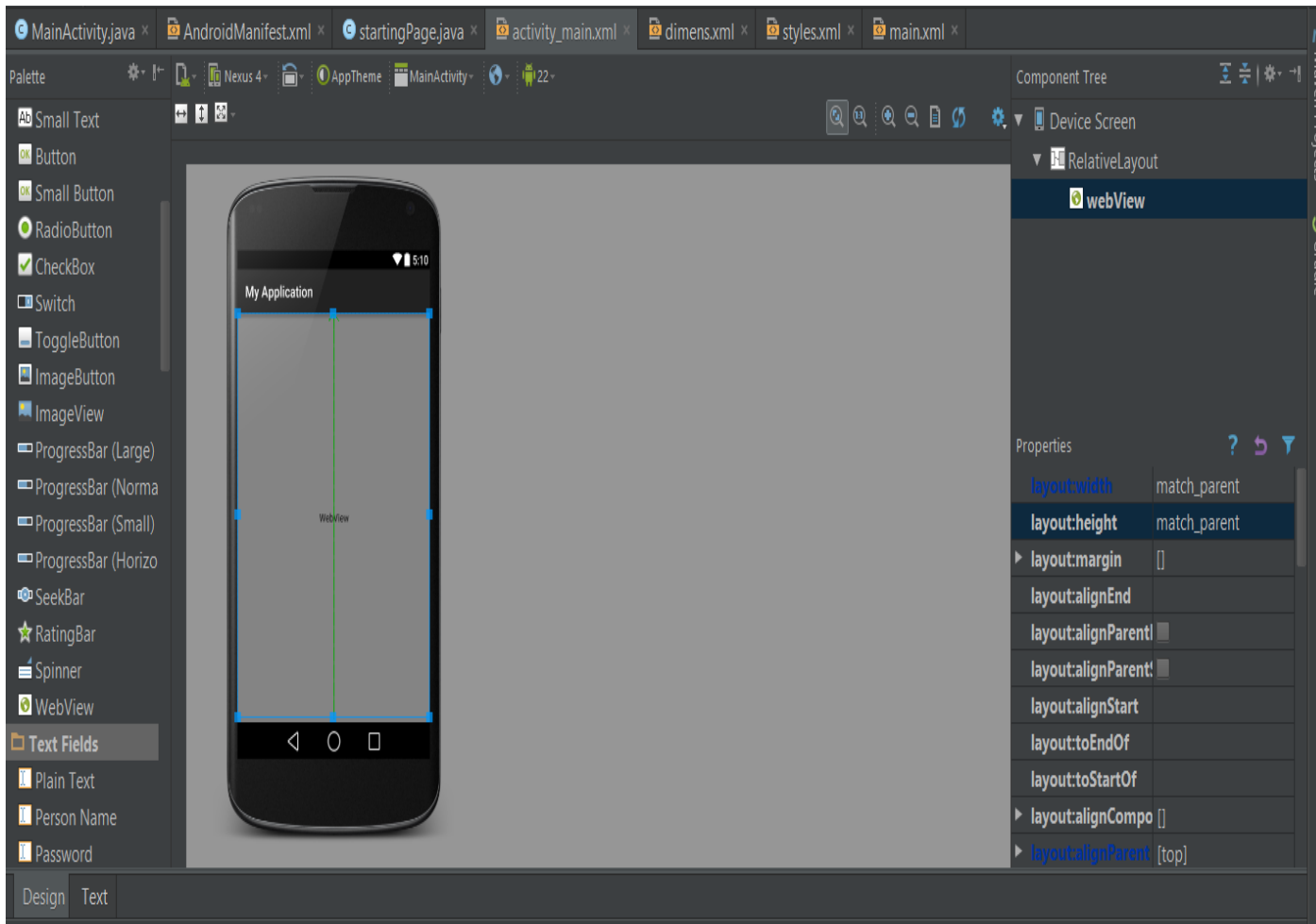


## Android Studio Programlama Bölüm 9

### WebView Kullanımı ve Uygulamada Web Sayfası Açma


Bu ders'te web sayfaları ile etkileşimi göreceğiz. **webView** içinde bir web sayfası açmayı görecez. **WebView**, hem uzak sunuculardaki web sayfalarını hem de uygulamanız içerisine koyduğunuz **html** sayfalarını görüntüleyebileceğiniz bir bileşendir. Bu bileşeni kullanarak uygulamanızdan çıkmadan web sayfalarını görüntüleyebilirsiniz.

Basit bir [umiitkose.com](http://umiitkose.com)'u açma işlemi gerçekleştirelim. Öncelikle Activity'de paletta'dan bir webView sürükleyip bırakıyoruz.



Bir sonraki adım'da ise kod kısmımız var. Öncelikle webView'i tanımlıyoruz. setJavaScriptEnabled ile java scriptleri çalıştırıyoruz. setWebViewClient ile kendimiz tarayıcıda açmamasını sağlıyoruz. load.URL ile de yüklenmesini sağlıyoruz.

```

public class MainActivity extends ActionBarActivity {
    
    WebView webV;

    //tarayıcı referansını oluşturalım.

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        webV=(WebView) findViewById(R.id.webView);

        //tarayıcımızı tanıtalım.

        webV.getSettings().setJavaScriptEnabled(true);

        //javascriptleri çalıştırmasını sağlayalım.

        webV.setWebViewClient(new WebViewClient());

        //bu kodu yazmadığımız takdirde telefonun tarayıcısında açıyor...
        webV.setWebChromeClient(new WebChromeClient());
        //bu kod youtube tarzı sitelerde videoyu oynatması için. Eski sürüm androidlerde çalışmayabilir...

        webV.loadUrl("http://www.umiitkose.com");

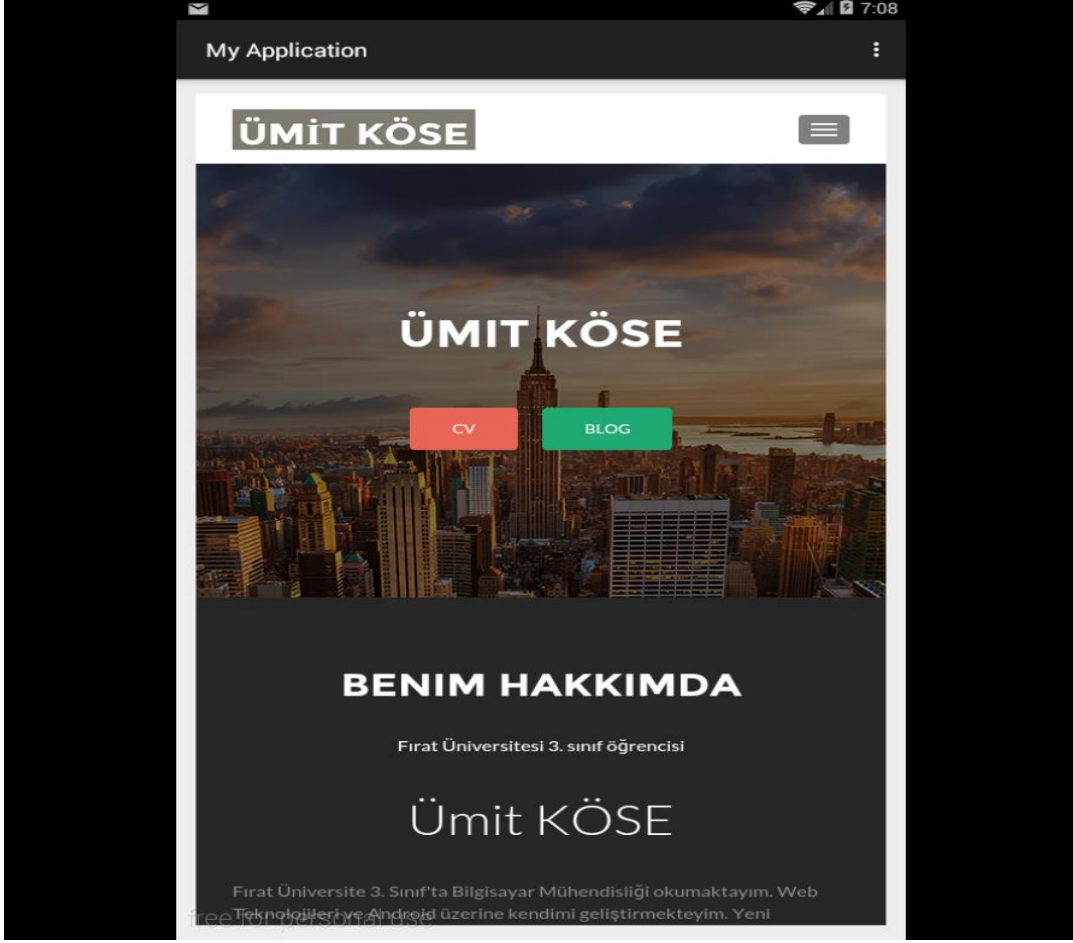
    } //bir url'yi çağırmasını istiyoruz.

```

Projenin son adımı ise her zamanki gibi izinler.

<uses-permission android:name="android.permission.INTERNET"/> satırını android manifest'te ekliyerek kullanabilirsiniz.

Uygulama Çalıştığında..



Daha gelişmiş bir uygulama olan [Uygulama 2 internet tarayıcısı yapma](#) Yazısını incelemenizi öneririm.

# Android Studio Programlama Bölüm 10

---

## Android'te Titreşim Kullanımı

Android'te bu dersimizde **Vibrator** sınıfını kullanacağız. Öncelikle arayüzümüze bir Button atalım. Button'a basınca kaç ms titreşim olacağını ayarlıyalım..

Arayüze bir buton bıraktık. Sonrasında ise kullanacağımız sınıf Vibrator.

### Kod olarak

```
Vibrator degisken=(Vibrator).getSystemService(Context.VIBRATOR_SERVICE);  
komutunu ekleyip  
degisken.vibrate(200); diyerek titreşimi onaylatıyoruz.  
Burada 200 ms olarak alınır ona göre hesaplama yapınız.
```

### Ve tabiki izin için..

```
<uses-permission android:name="android.permission.VIBRATE" />  
Komutunu ekliyoruz.
```

Eğer telefon ses modu komutlarını kontrol edicekseniz yararlı olacak komutlar:  
AudioManager sesKontrol = (AudioManager)getSystemService(Context.AUDIO\_SERVICE);

```
switch (sesKontrol.getRingerMode()) {  
case AudioManager.RINGER_MODE_SILENT:  
Log.i("voiceControl", "Telefon Mode: Sessiz");  
break;  
case AudioManager.RINGER_MODE_VIBRATE:  
Log.i("voiceControl", "Telefon Mode: Titreşim");  
break;  
case AudioManager.RINGER_MODE_NORMAL:  
Log.i("voiceControl", "Telefon Mode: Normal");  
break;  
}
```

---

[Android Studio'da apk dosyası oluşturmak için tıklayın.](#)

---

# Android Studio Programlama Bölüm 11

## Android'te Yazı Tipi Değiştirme

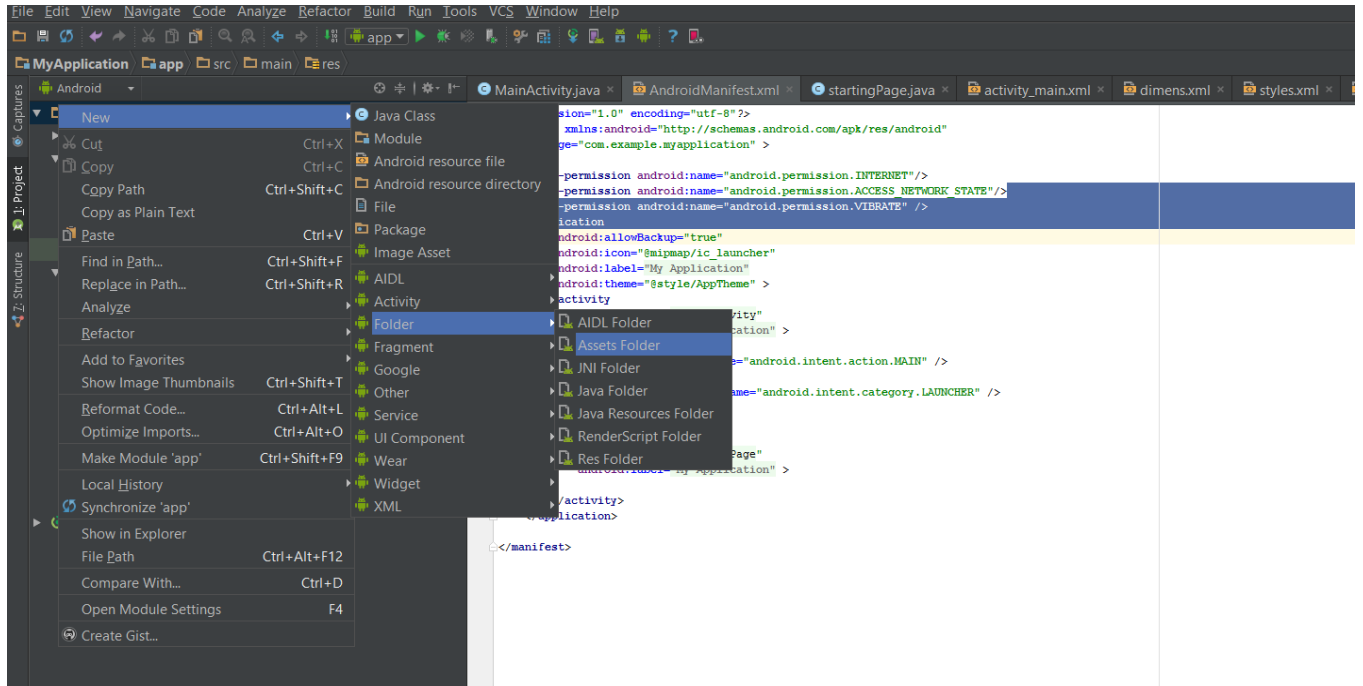
Öncelikle yazı tipi değiştirmek için bizim fonts'lara ihtiyacımız var. Fontlar içerisinde karakterleri barındıran yazı dizisidir. ttf uzantılıdır. Bir kaç tane güzel font olan siteler aşağıdadır.

<http://www.urbanfonts.com/fonts.php?fontauthor=2761> burdan güzel olan yazı tiplerini indirin arkadaşlar..

<http://www.fontr.com/trliste.aspx?sayfaNo=13&grup=Çeşitli&sys=1>

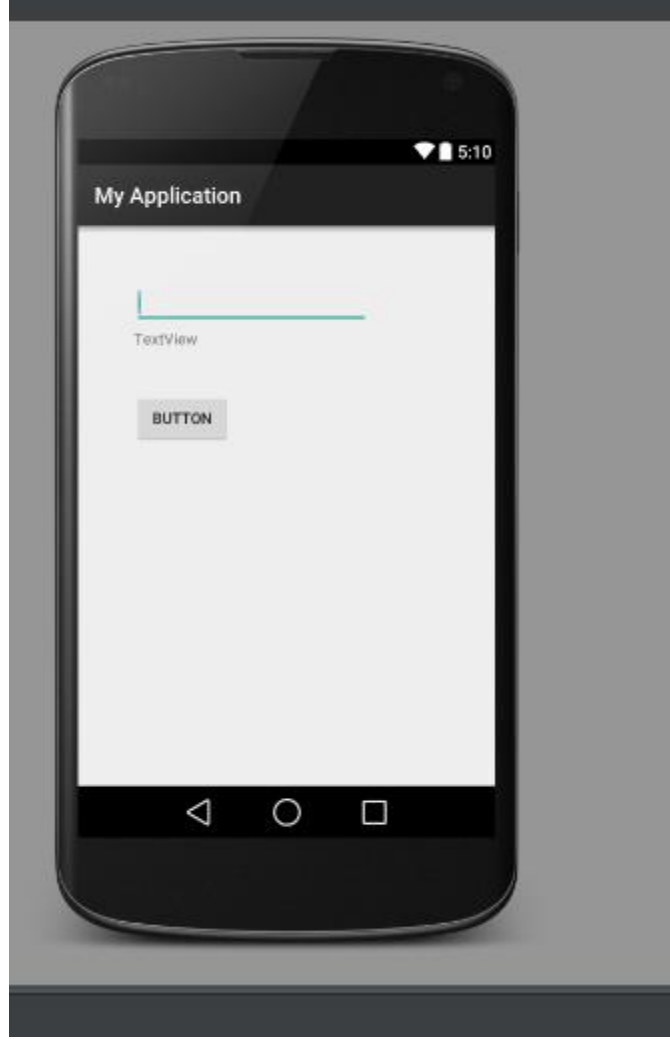
Öncelikle indirdiğimiz ttf uzantılı yazı fontlarımızı assets klasörü altında fonts klasörünün içine atıyoruz. Yoksa sağ tık new folder dememiz yeterli olacaktır.

Assets dosyasını ister res klasörünün dosya konumuna elle oluşturabilirsiniz. İsterseniz android studio içinden aşağıdaki yolları izleyerek oluşturabilirsiniz.





Oluşan assests dosyasına sağ tık ile bir directory içinde fonts klasörü oluşturun ve oneday.ttf'i ben kendim için indirdim sizde kendinize özel tff'i indirip içine atınız.



İkinci iş Tasarım oluşturmak.. Edittext,textview ve button olayı şu. Edittext'e yazdığımız yazı buttona basıyoruz seçtiğimiz font'ta bize text viewde gösteriyor..

Xml Kodumuz:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >
```

```
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginLeft="36dp"
```

```
android:layout_marginTop="136dp"  
android:text="Button" />
```

```
<TextView  
android:id="@+id/textView1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/button1"  
android:layout_alignParentTop="true"  
android:layout_marginTop="79dp"  
android:text="TextView" />
```

```
<EditText  
android:id="@+id/editText1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_alignLeft="@+id/textView1"  
android:layout_alignParentTop="true"  
android:layout_marginTop="33dp"  
android:ems="10" >
```

```
<requestFocus />  
</EditText>
```

```
</RelativeLayout>
```

Javada kısmındaysa, Edittext'ten girdiğimiz karakterleri string tipinde alıyoruz ve textView'e setText komutu ile yazdırıyoruz..

Button'a basınca da t1.de font oneway.ttf ayarladığımız font olacak.

Java Kodumuz:

```

Button b1;
EditText e1;
TextView t1;
Typeface tf1;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    e1=(EditText) findViewById(R.id.editText1);
    t1=(TextView) findViewById(R.id.textView1);
    b1=(Button) findViewById(R.id.button1);
    tf1=Typeface.createFromAsset(getAssets(), "fonts/oneway.ttf");
    b1.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            t1.setText(e1.getText().toString());
            t1.setTypeface(tf1);
        }
    });
}

```

```
package com.example.yazitipi;
```

```

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

```

```

public class MainActivity extends Activity {
    Button b1;
    EditText e1;
    TextView t1;
    Typeface tf1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1=(EditText)findViewById(R.id.editText1);
        t1=(TextView)findViewById(R.id.textView1);
        b1=(Button)findViewById(R.id.button1);
        tf1=Typeface.createFromAsset(getAssets(), "fonts/oneway.ttf");
        b1.setOnClickListener(new View.OnClickListener() {

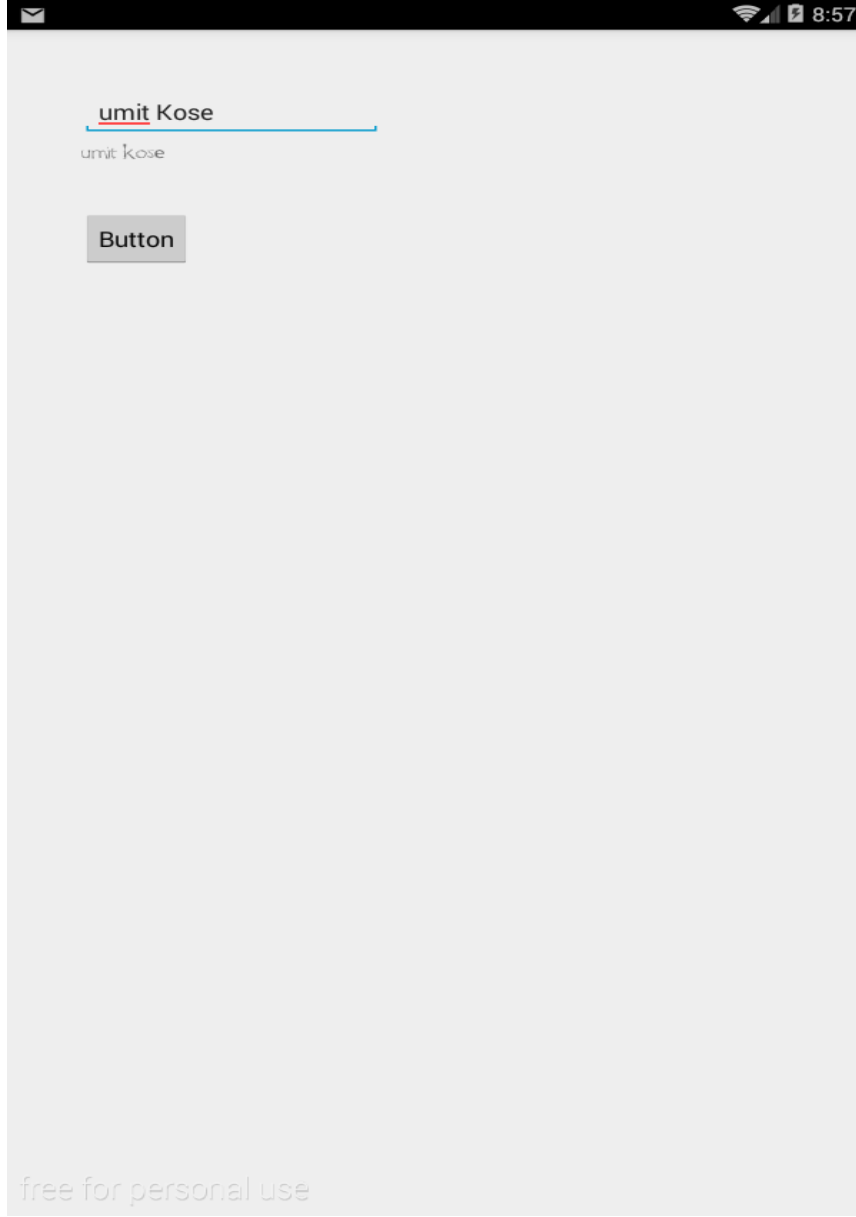
```

```

            @Override
            public void onClick(View v) {
                t1.setText(e1.getText().toString());
                t1.setTypeface(tf1);
            }
        });
    }
}

```

Uygulamamızın Resimli hali: Üstteki editText'teki yazı ile altında ki textView'deki yazı farklı



---

## Android Studio Programlama Bölüm 12

---

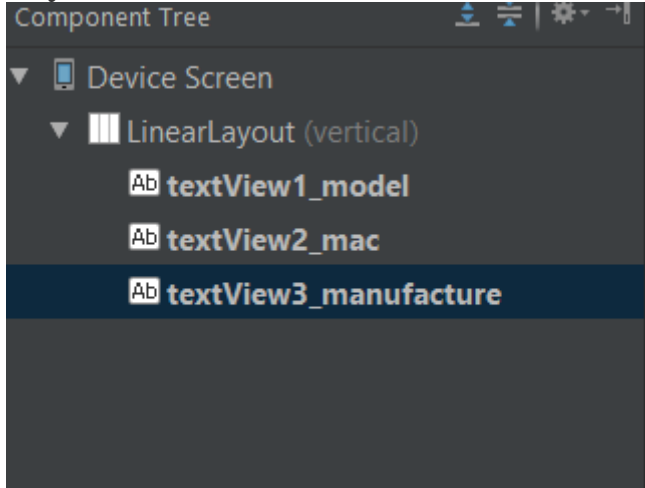
### Android Studio Mac Adresi, Üretici ve Model Bilgileri

Android programlama yazarken bazen Mac Adresi, Model ya da Üretici bilgileri gibi bazı bilgilere ihtiyacımız olabilir. Şimdi onlarla ilgili örnek yapacağız.

import **android.os.Build**; -> **Build** sınıfında bir çok bilgilere erişebiliyoruz. Mesela Şuan üstte yazdığım bilgiler.

Şimdi Hemen başlayalım. Yeni Projemizde Öncelikle boş 3 tane textView atıyoruz.. Tasarım kısmı ve Kodu..

#### Layout Kodumuz :



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_height="wrap_content" android:text=""
        android:layout_width="wrap_content"
        android:id="@+id/textView1_model">
    </TextView>

    <TextView
        android:id="@+id/textView2_mac"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge" >

    </TextView>
    <TextView
```

```
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_height="wrap_content" android:text=""
        android:layout_width="wrap_content"
        android:id="@+id/textView3_manufacture">
    </TextView>
</LinearLayout>
```

## Java Kodumuz :

```
private TextView mac,model,manufacturer;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    model=(TextView) findViewById(R.id.textView1_model);
    mac=(TextView) findViewById(R.id.textView2_mac);
    manufacturer=(TextView) findViewById(R.id.textView3_manufacture);
    model.setText("Model => "+ Build.MODEL);

    String mac_adresi=get_mac_address();
    mac.setText("Mac => "+mac_adresi);
    manufacturer.setText("Manufacturer => "+Build.MANUFACTURER);
}

private String get_mac_address()
{
    ConnectivityManager connMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo wifi = connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

    try{
        WifiManager wifiMan = (WifiManager) this.getSystemService(Context.WIFI_SERVICE);
        WifiInfo wifiInf = wifiMan.getConnectionInfo();
        return wifiInf.getMacAddress();
    }
    catch (Exception e) {
        return "00:00:00";
    }
}
}
```

```
package com.example.myapplication;

import android.app.Activity;
import android.content.Context;
import android.net.ConnectivityManager;
```

```

import android.net.NetworkInfo;
import android.net.wifi.WifiInfo;
import android.net.wifi.WifiManager;
import android.os.Build;
import android.os.Bundle;
import android.widget.TextView;

public class startingPage extends Activity {

    private TextView mac,model,manufacturer;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        model=(TextView) findViewById(R.id.textView1_model);
        mac=(TextView) findViewById(R.id.textView2_mac);
        manufacturer=(TextView) findViewById(R.id.textView3_manufacture);
        model.setText("Model => "+ Build.MODEL);

        String mac_adresi=get_mac_address();
        mac.setText("Mac => "+mac_adresi);
        manufacturer.setText("Manufacturer => "+Build.MANUFACTURER);

    }

    private String get_mac_address()
    {
        ConnectivityManager connMgr = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo wifi = connMgr.getNetworkInfo(ConnectivityManager.TYPE_WIFI);

        try{
            WifiManager wifiMan = (WifiManager) this.getSystemService(Context.WIFI_SERVICE);
            WifiInfo wifiInf = wifiMan.getConnectionInfo();
            return wifiInf.getMacAddress();
        }
        catch (Exception e) {
            return "00:00:00";
        }
    }
}

```

## ve android Manifest Dosyamız.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication" >

    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

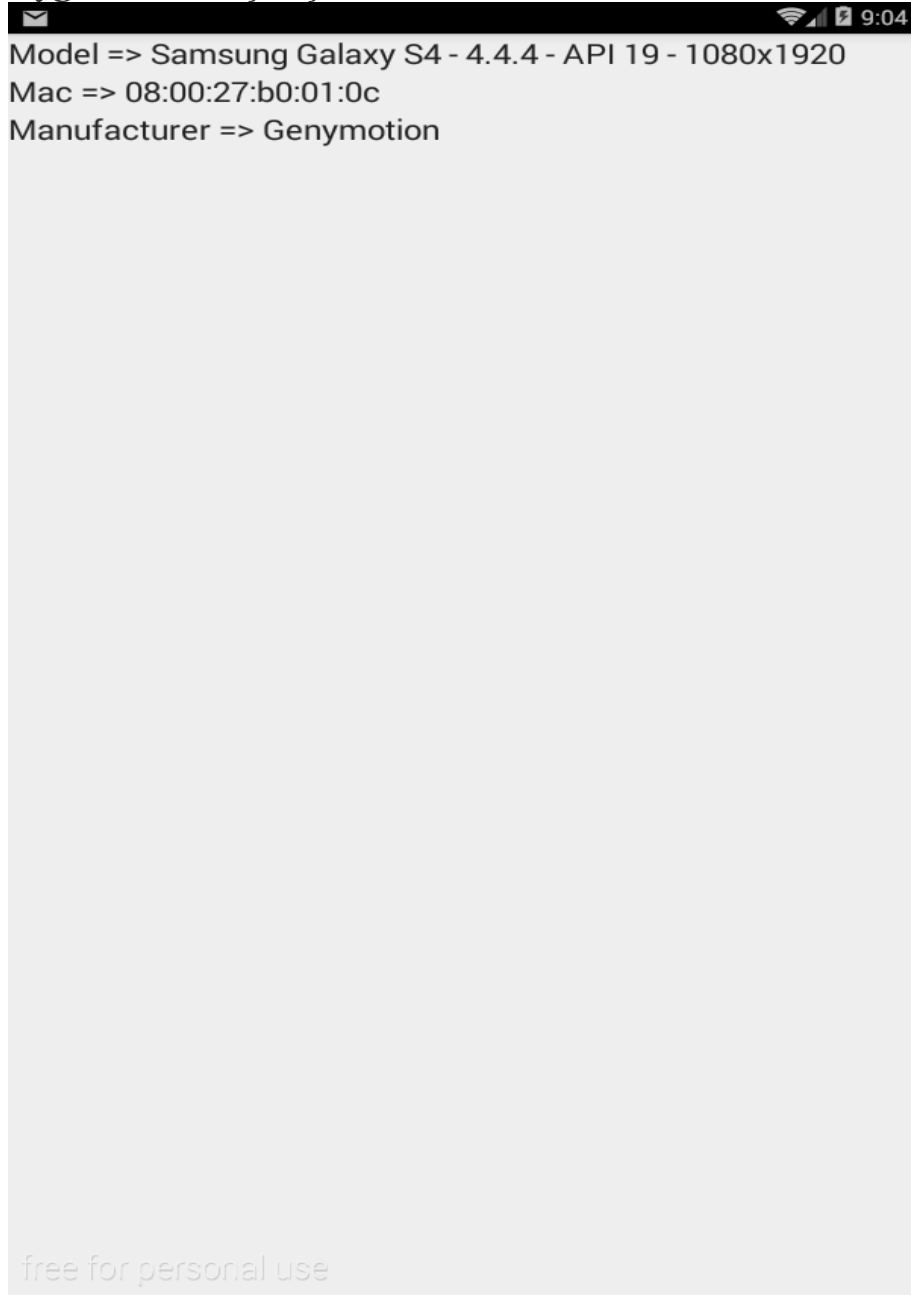
```

```
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".startingPage"
    android:label="@string/app_name" >

    </activity>
</application>

</manifest>
```

### Uygulamanın Çalışması :





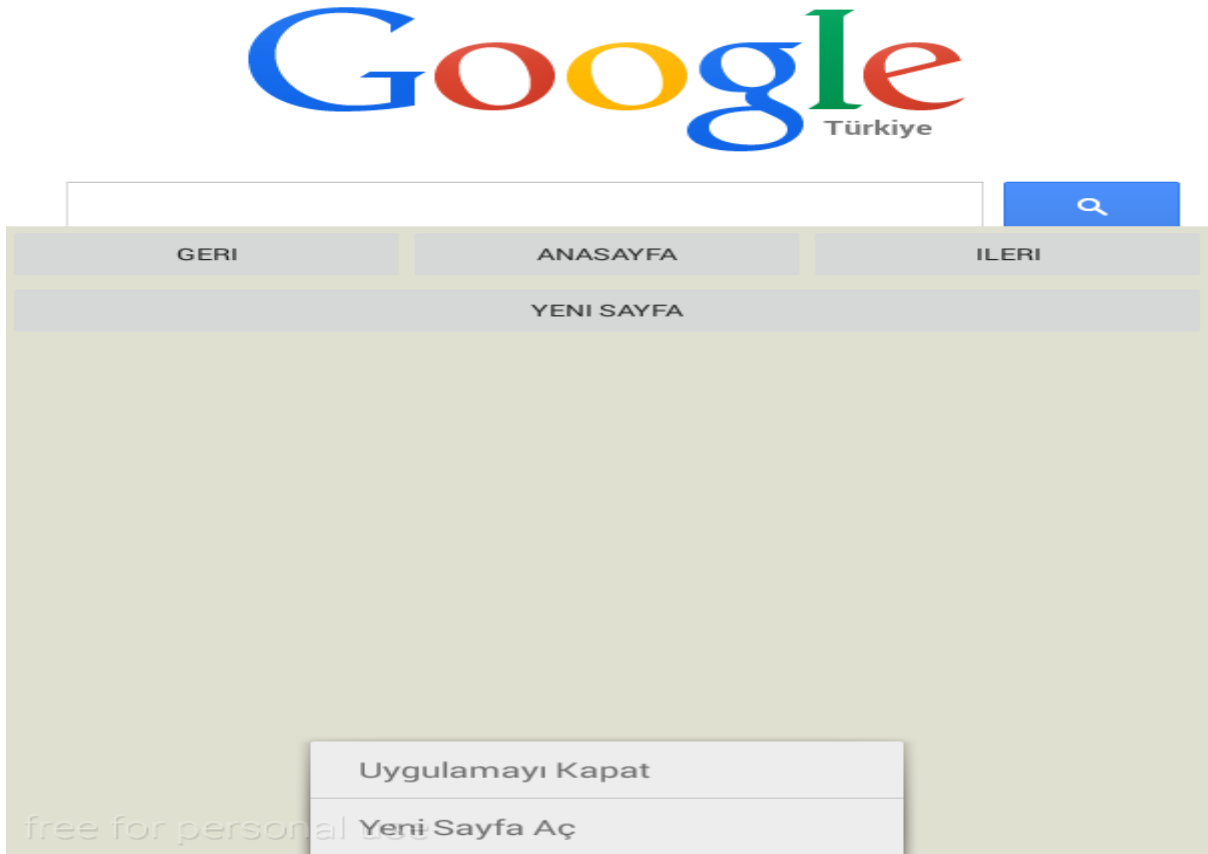
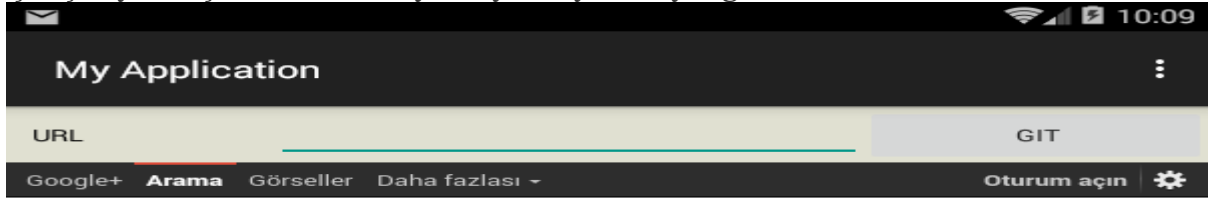
## Android Studio Programlama Bölüm 13

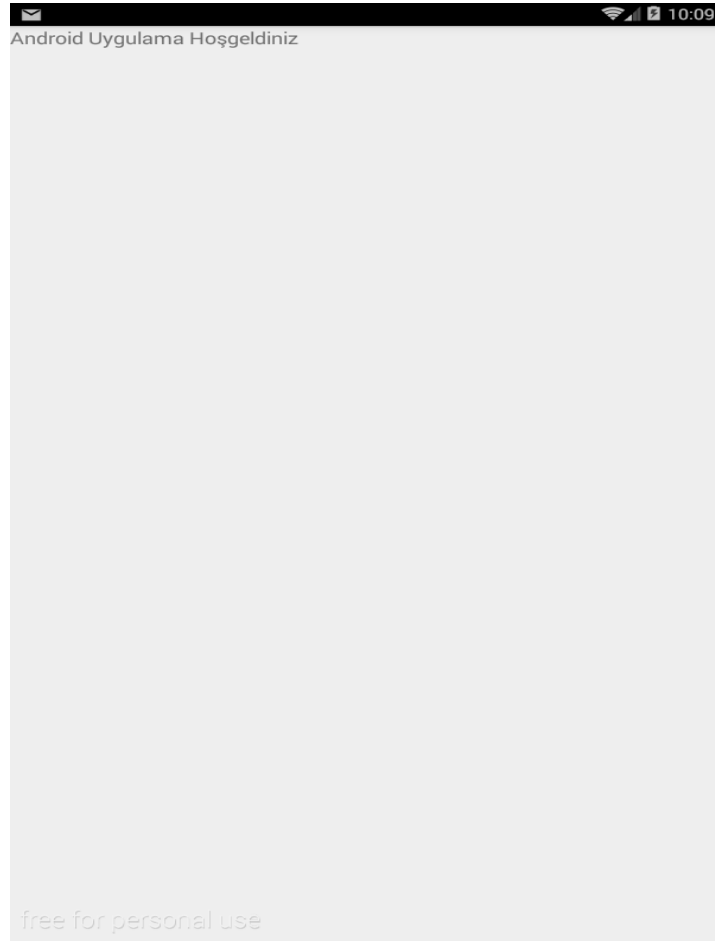
### Android Studio Option Menü oluşturma

Menü uygulamamızı anlatacağım. Mesela android telefonda sağ tuş menüye tıkladınız hakkında, çıkış sayfası geldi diye düşünün. Şimdi onun kodlarla nasıl yapılacağını göreceğiz.

Öncelikle yeni bir android Sayfası oluşturcaz ve menüye basıp 2. sayfa mesela bu hakkında sayfası olsun.Sayfaya tıkladığımda 2. activity'e geçiş yapsın. Çıkış tuşuna basınca da çıksın..[Yeni sayfa oluşturma vs. tekrar öğrenmek istenlere..](#)

Uygulamamızın activity kısmı ve kodları : Ana sayfa'da sağ altta bu şekilde olucak.. Çıkış Diyince çıkıcak. Yeni sayfa diyince yeni sayfa gelicek.





### Java Kodlarımız :

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    switch (id){
        case R.id.item1: System.exit(0);
            return true;
        case R.id.newpage:
            Intent a=new Intent(MainActivity.this,startingPage.class);
            startActivity(a);
    }
    return true;
}

return false;
}
```

### Resim :

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    switch (id){
        case R.id.item1: System.exit(0);
            return true;
        case R.id.newpage:
            Intent a=new Intent(MainActivity.this,startingPage.class);
            startActivity(a);
    }
    return true;
}

return false;
}
}
}

```

Menu.xml'de menü klasörü içinde ise 2 tane yeni item ekleyeceğiz. Bunlardan 1 tane vardır ve adı da büyük ihtimal settings'tir. Ayarlayarak halledebilirsiniz.

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools" tools:context=".MainActivity">
    <item android:id="@+id/item1" android:title="Uygulamayı Kapat" />
    <item android:id="@+id/newpage" android:title="Yeni Sayfa Aç"
        android:orderInCategory="100" app:showAsAction="never" />
</menu>

```

Uygulamayı çalıştırıp menü tuşuna basınca yukardaki uygulama gibi boş ekranda uygulamayı kapat ve yeni sayfa aç çıkacaktır.

## Android Studio Programlama Bölüm 14

### Android Studio Button Resmi Deęiřtirme, Arama Yapma

Bu bölümde Arama yapacağız. Bildiđiniz numara arıycaz. Bir buton olacak bu butona basınca direk arama yapacağız. Hadi 155 i arayalım tabii işletmeyelim.. 😊



řimdi öncelikle uygulamamıza bir buton atacağız.

Yeni Projemizde öncelikli olarak butonumuzu res klasörü içindeki drawable-hdpi içine atıyoruz..(sürükleyip bırak)

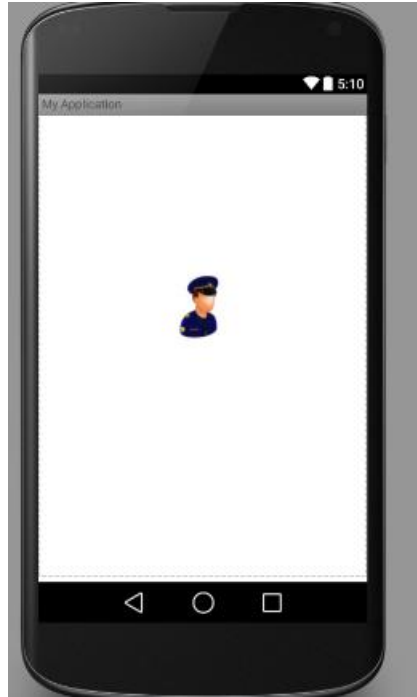
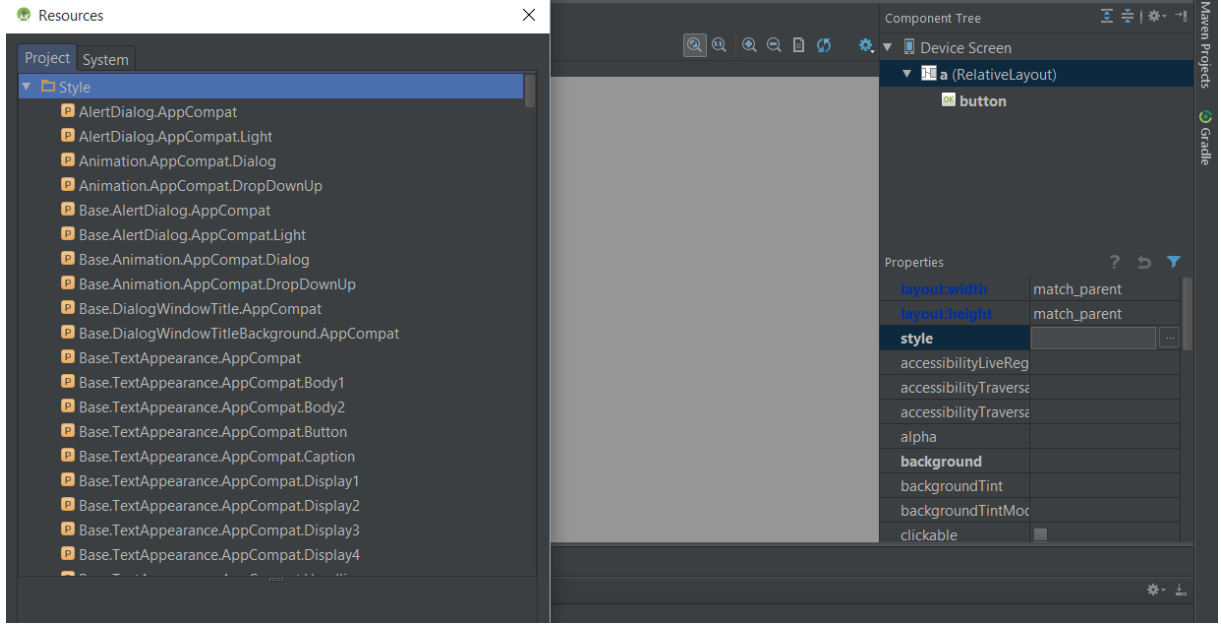
řimdi ise styles'de butonumuzun görseli tanımlıyoruz..(Values Klasörünün içinde)

```
<style name="c" parent="@android:style/Widget.Button">
<item name="android:background">@drawable/g</item>
```

Kod bu görüntü ise..

```
styles.xml
resources
  <!-- Base application theme. -->
  <style name="c" parent="@android:style/Widget.Button">
    <item name="android:background">@drawable/g</item>
  </style>
</resources>
```

řimdi ise Bir button'u layoutumuzun içine atın. Ve sağ tık Edit Style ile button görünümünü deęiřtiriyoruz..



Şimdi ise aramamız için gerekli java kodumuz.İntent ile Bir numarayı aradık.. ve en son activitemizi çalıştırdık..

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button button=(Button) findViewById(R.id.button);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent arama=new Intent(Intent.ACTION_CALL);
            arama.setData(Uri.parse("tel:+155));
            startActivity(arama);
        }
    });
}

```

Biz bunları yaptıkta İzin aldık mı? Tabiki de hayır. Android manifestte izin için Permissions ve Add Sonra name de CALL PHONE seçelim..

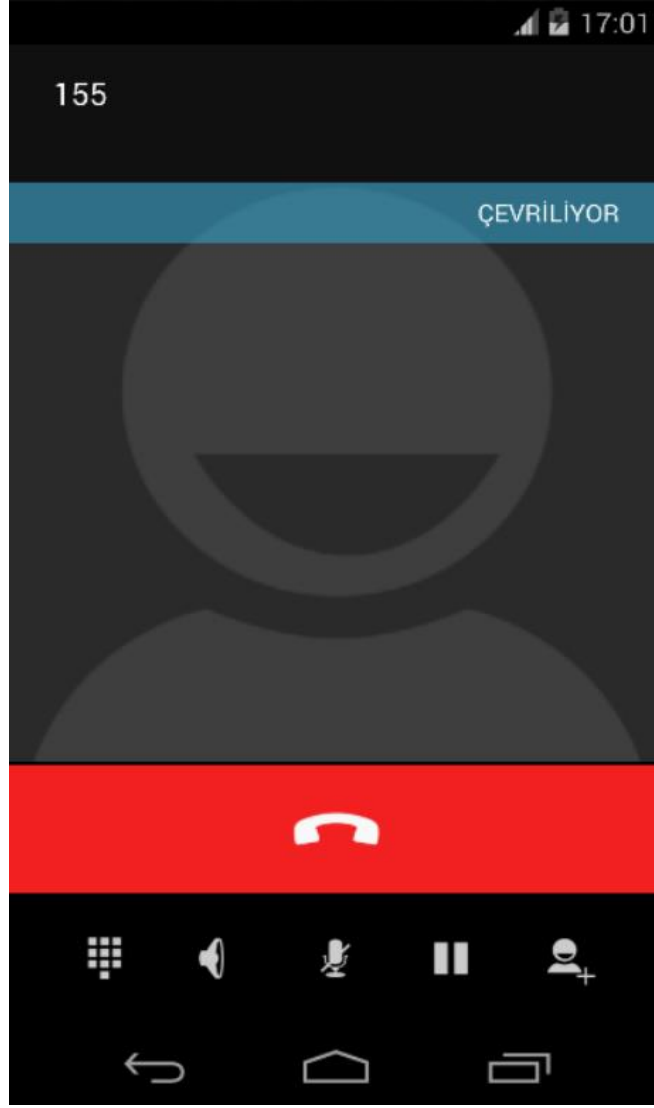
```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.umitkose.myapplication" >
    <uses-permission android:name="android.permission.CALL_PHONE"></uses-permission>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        >
        <activity
            android:name=".MainActivity"
            android:label="My Application" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

Hadi çalıştıralım.. Çalıştırdığımızda kodlarımızın görünümü..





Kısa olması için button'u image button olarak'ta ayarlayabilirsiniz. Tabi resmi bir button gibi göstercektir.

**URI:** Burada kullandığımız uri kısmı (uniform resource identifier) nizami kaynak belirteci olarak geçmekte ve bir kaydağı ya da veriyi isimlendirmek için kullanılmaktadır. Kaynağı nitelendirir.

---



---

## Android Studio Programlama Bölüm 15

---

### Android Studio Bluetooth Kontrolü

Bu yazımda android studio da bluetooth kontrolüne bakacağız. Bluetooth'u günümüzde kullanmayan akıllı telefon kalmadı. Bluetooth elektronik aletlerin kısa mesafelerde (10 metre gibi) birbirleri ile iletişim kurabilmesini sağlayan teknolojidir. Bluetooth ile cihazlar arası dosya paylaşımı yapılabilir. Ayrıca Bluetooth destekli kulaklıklar ile ses iletişimi de sağlanabilir.

Bir cep telefonu ile bir tableti, laptop bilgisayarı ya da PDA'yı Bluetooth aracılığı ile birbirine bağlayıp internete girmeniz mümkün olabilir.

Şimdi sizinle bir android cihazda bluetooth açma, kapama ve görünür yapma işlemlerini yapcaz.

Tasarım kısmında 3 tane button kullanacağız. Aç, Kapa ve görünür yap olacak bunlar..

activity\_main kodlarımız :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
```

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true">
```

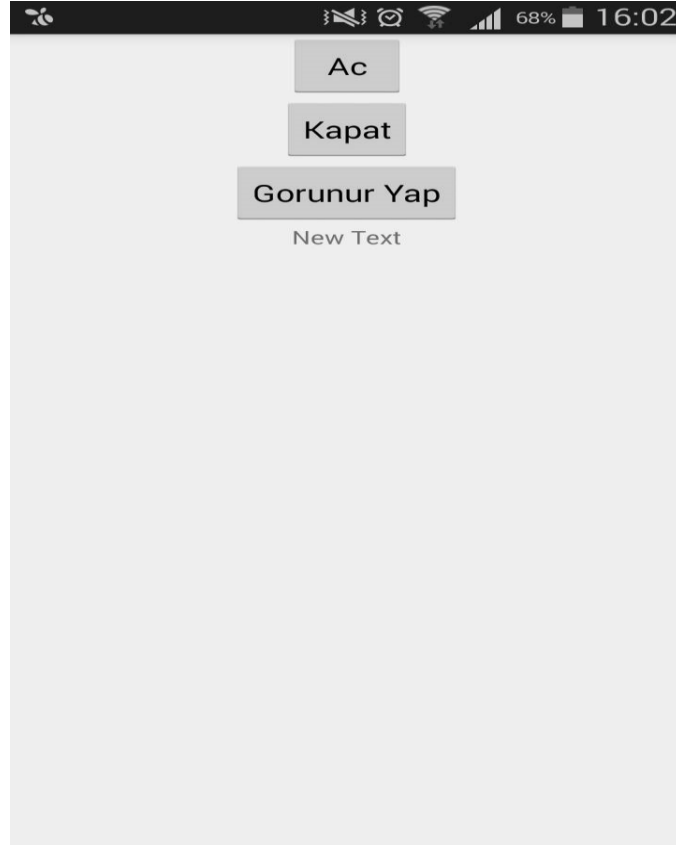
```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ac"
    android:id="@+id/ac"
    android:layout_gravity="center_horizontal" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kapat"
    android:id="@+id/kapat"
    android:layout_gravity="center_horizontal" />
```

```
<Button
    android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Gorunur Yap"
        android:id="@+id/gyap"
        android:layout_gravity="center_horizontal" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Text"
    android:id="@+id/textView"
    android:layout_gravity="center_horizontal" />
</LinearLayout>
</RelativeLayout>
```



Sonraki işlem ise java kodlarını kullanmak olacaktır. 3 butonu tanımladık. Burada BluetoothAdapter sınıfını kullanacağız. Bluetooth'un durumunu alarak buton'a işlevlerini vereceğiz.

**BluetoothAdapter.Action\_Request\_enable** kullanarak açma isteği işlemini gerçekleştireceğiz.

**BluetoothAdapter.Action\_Request\_discoverable** kullanarak görünür yapma isteği işlemini gerçekleştireceğiz.

Kapatmak için ise adaptorumuzu disable modunu kullanırız.

## JAVA KODLARIMIZ :

```
package com.umiitkose.myapplication;

import android.bluetooth.BluetoothAdapter;
import android.content.Intent;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;

public class MainActivity extends Activity {

    Button ac;
    Button kapat;
    Button gorunurYap;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ac=(Button)findViewById(R.id.ac);
        gorunurYap =(Button)findViewById(R.id.gyap);
        kapat=(Button)findViewById(R.id.kapat);
        final BluetoothAdapter adaptor = BluetoothAdapter.getDefaultAdapter();
        ac.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                if(adaptor == null )
                {
                    Toast.makeText(MainActivity.this, "Blueetooth Aygıtı Bulunamadı",
Toast.LENGTH_SHORT).show();
                }

                else
                {
                    if(!adaptor.isEnabled())
                    {
                        Intent bluetoothBaslat = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                        startActivityForResult(bluetoothBaslat,1);
                        Toast.makeText(MainActivity.this, "Bluetooth Aygıtı Açık", Toast.LENGTH_SHORT).show();
                    }
                    else {adaptor.disable();}
                }
            }
        });
        gorunurYap.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub

                Intent gorunurYap = new Intent (adaptor.ACTION_REQUEST_DISCOVERABLE);
                startActivityForResult(gorunurYap,1);
                Toast.makeText(MainActivity.this, "Görünür Hale geldi", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
    }  
  });  
  kapat.setOnClickListener(new View.OnClickListener() {  
  
      @Override  
      public void onClick(View v) {  
// TODO Auto-generated method stub  
        if(!adaptor.isEnabled())  
        {}else{  
            adaptor.disable();  
            Toast.makeText(MainActivity.this, "Kapatıldı", Toast.LENGTH_SHORT).show();  
        }  
    }  
  });  
}
```

Ve geriye 2 tane izin almamız gerekmektedir. Android Manifest Dosyamızda <application></application> taglarının üstüne

```
<uses-permission android:name="android.permission.BLUETOOTH"></uses-permission>  
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"></uses-permission>
```

İzinlerimizi ekleriz.

Emülatörde denerseniz hata alacaksınız'dır. Emülatörde blueetooth aygıtı bulunmamaktadır. Apk dosyası ile telefonunuz'da denerseniz çalıştığını göreceksiniz.

[Android Studio'da apk dosyası oluşturmak için tıklayın.](#)

Uygulamanın görüntüsü :

Saving screenshot...

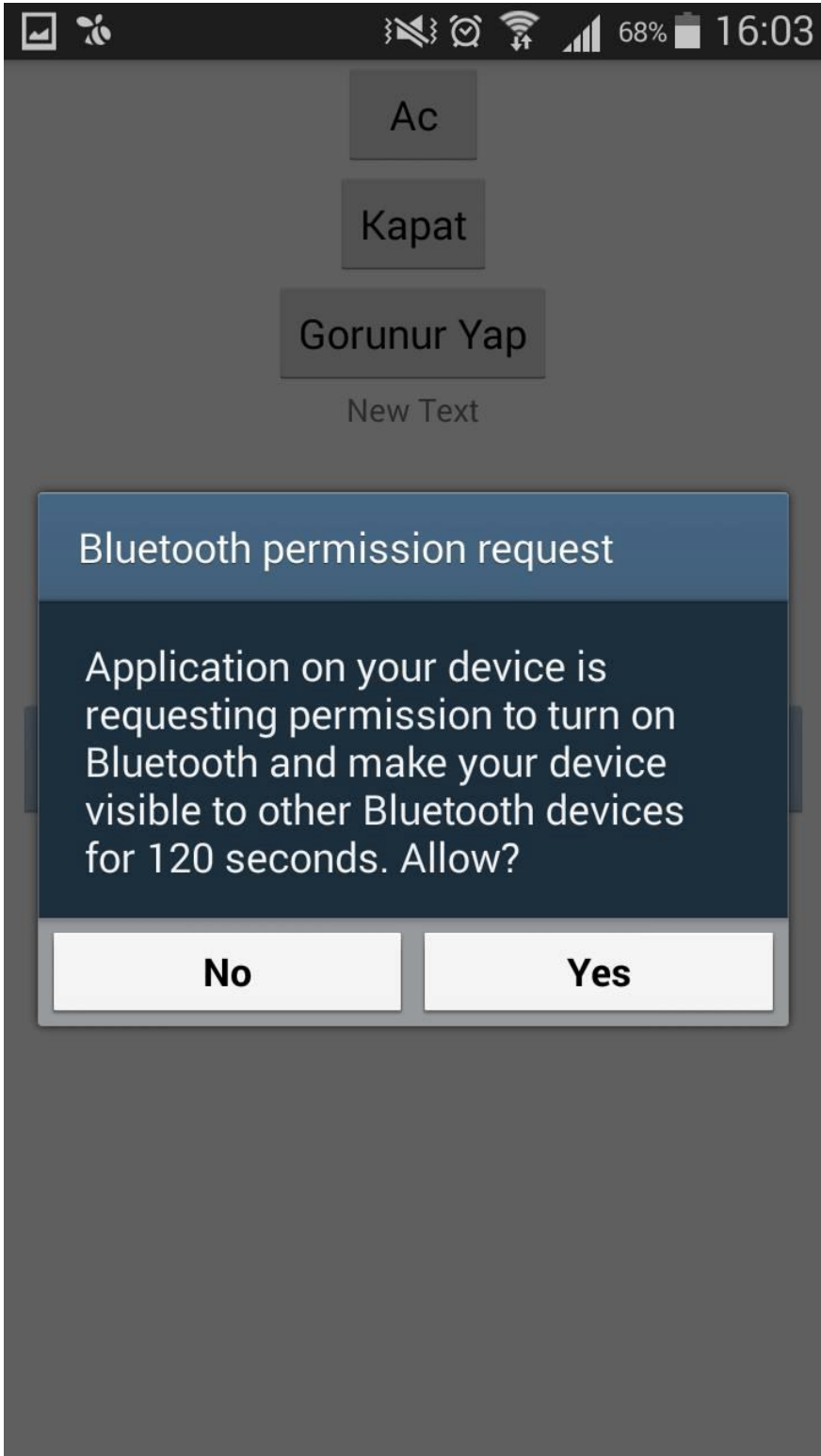
Ac

Kapat

Gorunur Yap

New Text

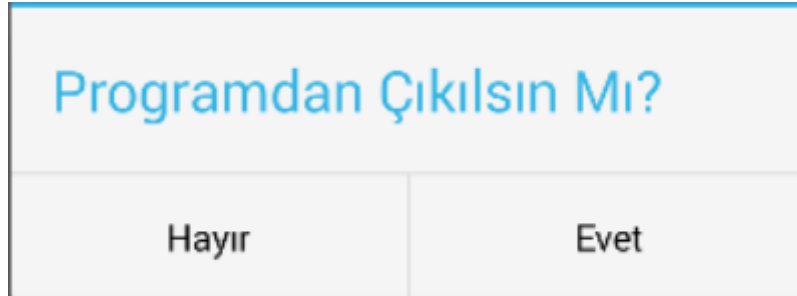
Kapatıldı



## Android Studio Programlama Bölüm 16

### Android Studio Alert Dialog

Android studio'da Alert Dialog konusuna geldik. Alert Dialog bizim ufak bir pencere ile yönlendirmeler yapmamıza yarayan bir uygulama kullanıcı diyalog penceresi diyebiliriz. Çünkü ufak bir pencere açılır ve ne yapmak istersiniz diye sorar. Örneğin Aşağıdaki resimde ki gibi..



Alert Dialog kullanarak bir mesajı kullanıcıya sunabiliriz. Mesela geri tuşuna basınca uygulamadan çıkmak mı istiyorsunuz ya da bir butona basınca bir uyarı penceresi çıkarabiliriz. Birazdan onun örneğini göreceğiz.

Öncelikle Kodlara Geçelim.. Layout kısmında bir işlem yapmamıza gerek yok. Sadece bir button atın ve Android Manifest'te de bir işleme gerek yok. Hepsi sadece java kodu ile hallediceğiz. Öncelikle geri buttonuna basınca gitmesi kısmı..

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {  
        //Eğer geri butonuna basılırsa
```

Burasını öncelikle **code -> Override -> onKeyDown** kısmından getirin. Sonrasında ise keyCode'a bir tıklama olayında ki Arka Back tuşuna basılırsa algılamasını istiyoruz.

```
final AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);  
    builder.setTitle("Uyarı !!");  
    builder.setMessage("Web sitesi Açılsın mı?");  
    builder.setCancelable(false);  
    builder.setIcon(R.drawable.a);  
  
    builder.setPositiveButton("Evet", new DialogInterface.OnClickListener() {  
  
        @Override  
        public void onClick(DialogInterface dialog, int which) {  
            // TODO Auto-generated method stub
```

```

        Toast.makeText(MainActivity.this, "Açılıyor.", Toast.LENGTH_LONG).show();
    }
});
builder.setNegativeButton("Hayır", new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        // TODO Auto-generated method stub
        Toast.makeText(MainActivity.this, "Pekala.", Toast.LENGTH_LONG).show();
    }
});

AlertDialog dialog = builder.create();
dialog.show();
}

```

Uygulamada ki 2 kısımda ki kodlar da aşağıda. AlertDialog mantığını anlatıyorum. Öncelikle bir kütüphaneden bulunan activity'de bir alertDialog pencere oluşturulması istendi.

`builder.setTitle`: En yukarda ki uyarı..

`builder.setMessage` : Kullanıya sunacağınız seçenek, mesaj..

`builder.setCancelable(false)` : Burada ki false bu mesajı cevaplamadan dialoga bir cevap vermeden dialog kapanmıyacak demek. True ya da yazmazsanız o zamanda ekranda bir yere tıklayarak bu kısımdan kurtulabilir. Yani dialogumuzu cevaplama zorunluluğu kılıyoruz.

Geri kalan işlemler de evet ve hayır adında iki kısım oluşturup yapılan cevaba göre toast mesaj göstermek ya da geri butonunda ki gibi uygulamayı kapatmak.

Uygulamanın kodları ve çalışma resimleri aşağıdadır. Başarılar..

Kodlar :

```

package com.umiitkose.egitim2;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends Activity {

    @Override

```



```

public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && event.getRepeatCount() == 0) {
//Eğer geri butonuna basılırsa

        try{
            AlertDialog.Builder alertDialogBuilder=new AlertDialog.Builder(this); //Mesaj Penceresini
Yaratalım
            alertDialogBuilder.setTitle("Programdan Çıkılsın
Mı?").setCancelable(false).setPositiveButton("Evet", new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int id) { //Eğer evet butonuna basılırsa
                    dialog.dismiss();
                    android.os.Process.killProcess(android.os.Process.myPid());
//Uygulamamızı sonlandırıyoruz.
                }

            }).setNegativeButton("Hayır", new DialogInterface.OnClickListener() {
//Eğer hayır butonuna basılırsa

                @Override
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(getApplicationContext(), "Programdan çıkmaktan vazgeçtiniz.",
Toast.LENGTH_SHORT).show();
                }
            });

            alertDialogBuilder.create().show();
//son olarak alertDialogBuilder'ı oluşturup ekranda görüntüleyiyoruz.
            return super.onKeyDown(keyCode, event);
        }
        catch(IllegalStateException e){ //yapımızı try-catch blogu içerisine aldık
//hata ihtimaline karşı.
            e.printStackTrace();
        }
        return true;
    }
    return super.onKeyDown(keyCode, event);
}

protected void onCreate(Bundle savedInstanceState) {
// TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final Button a=(Button)findViewById(R.id.button);
a.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        final AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
        builder.setTitle("Uyarı !!");
        builder.setMessage("Web sitesi Açılsın mı?");
        builder.setCancelable(false);
        builder.setIcon(R.drawable.a);

        builder.setPositiveButton("Evet", new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {
                // TODO Auto-generated method stub
                Toast.makeText(MainActivity.this, "Açılıyor.", Toast.LENGTH_LONG).show();
            }
        });
        builder.setNegativeButton("Hayır", new DialogInterface.OnClickListener() {

```

```
@Override
public void onClick(DialogInterface dialog, int which) {
    // TODO Auto-generated method stub
    Toast.makeText(MainActivity.this, "Pekala.", Toast.LENGTH_LONG).show();
}
});

AlertDialog dialog = builder.create();
dialog.show();
}

});
}}
```

Hello world!

New Button



Uyarı !!

Web sitesi Açılsın mı?

Hayır

Evet



Hello world!

New Button

Programdan Çıkılsın Mı?

Hayır

Evet

Sayfaya Yönlendiriliyorsunuz.



## Android Studio Programlama Bölüm 17

---

### Android Studio Progress Dialog

Android studio 'da progress dialog'u göreceğiz. Öncelikle nedir bu progres Dialog? Progress Dialog belirli bir işlem yaparken işlemin uzun sürmesine karşılık kullanıcılara sunduğumuz uyarı penceresidir. Mesela Dosya indirirken ki 100'e kadar dolma işlemi. Hadi bunu biz bir progress örneğinde yapalım

Activity\_main , Layout Kodumuz : Sadece 1 tane button attık.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Progress Dialog Tıkla"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

Java Kodlarımız :

```
package com.umiitkose.egitim2;

import android.app.Activity;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

/**
 * Android ProgressDialog example
 * @author Prabu Dhanapal
 * @version 1.0
 * @since SEP 01 2013
 */
public class MainActivity extends Activity {
    Button button;
    ProgressDialog progressDoalog;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    button = (Button) findViewById(R.id.button1);
    button.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            progressDoalog = new ProgressDialog(MainActivity.this);
            progressDoalog.setMax(100);
            progressDoalog.setMessage("Yükleniyor...");
            progressDoalog.setTitle("Progres Dialog Örneği");
            progressDoalog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
            progressDoalog.show();
            new Thread(new Runnable() {
                @Override
                public void run() {
                    try {
                        while (progressDoalog.getProgress() <= progressDoalog
                            .getMax()) {
                            Thread.sleep(200);
                            handle.sendMessage(handle.obtainMessage());
                            if (progressDoalog.getProgress() == progressDoalog
                                .getMax()) {
                                progressDoalog.dismiss();
                            }
                        }
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }).start();
        }

        Handler handle = new Handler() {
            @Override
            public void handleMessage(Message msg) {
                super.handleMessage(msg);
                progressDoalog.incrementProgressBy(1);
            }
        };
    });
}
}

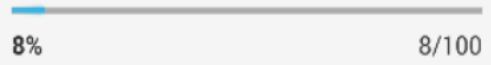
```

Uygulamamızın Çalışması :

Progress Dialog Tıkla

## Progres Dialog Örneği

Yükleniyor...



# Android Studio Programlama Bölüm 18

---

## Android Studio Rating Bar

Android studio'da rating Bar kullanımına geldik. Rating bar mesela bir işlem yapacaksınız puanlama oylama sistemi. Anket sistemi gibi düşünün. Onun için bir puan sistemi oluşturduunuz. İşte burada Rating bar kullanıacaksınız.

Layout'umuza öncelikle 1 tane rating bar, 1 tane button ve sonuc adlı text View oluşturduk.

Activity\_main Kodları :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/oylaTV"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Oyla"
    />

    <RatingBar
        android:id="@+id/ratingBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:numStars="5"
        android:stepSize="1.0" />

    <Button
        android:id="@+id/gonderButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gönder" />

    <LinearLayout
        android:id="@+id/linearLayout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/sonucTV"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Sonuç : "
        />

        <TextView
            android:id="@+id/sonucCiktiTV"
```



```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
    />
</LinearLayout>

</LinearLayout>

```

## Java Kodumuz ise :

```

package com.umiitkose.egitim2;

import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RatingBar;
import android.widget.Toast;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RatingBar;
import android.widget.RatingBar.OnRatingBarChangeListener;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
    private RatingBar ratingBar;
    private TextView ratingDegeri;
    private Button gonderButton;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        addListenerOnRatingBar();
        addListenerOnButton();
    }

    public void addListenerOnRatingBar() {

        ratingBar = (RatingBar) findViewById(R.id.ratingBar);
        ratingDegeri = (TextView) findViewById(R.id.sonucCiktiTV);

        //Rating değıştiđi anda bunu TextView'da görüntüle
        ratingBar.setOnRatingBarChangeListener(new OnRatingBarChangeListener() {
            public void onRatingChanged(RatingBar ratingBar, float rating,
                boolean fromUser) {

                ratingDegeri.setText(String.valueOf(rating));
            }
        });
    }

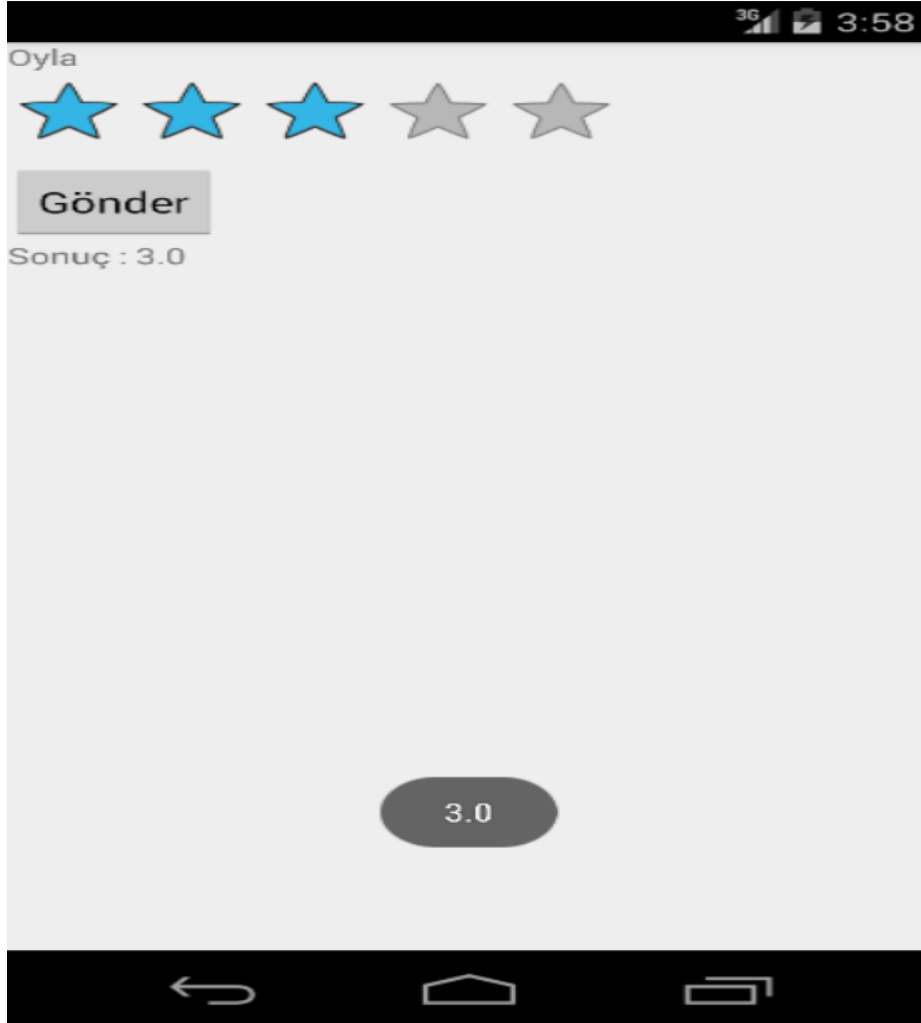
    public void addListenerOnButton() {

```

```
ratingBar = (RatingBar) findViewById(R.id.ratingBar);  
gonderButton = (Button) findViewById(R.id.gonderButton);
```

```
gonderButton.setOnClickListener(new OnClickListener() {  
    public void onClick(View v) {  
        //Buttona tıklandığında o anki rating değerini içeren bir toast mesajı göster  
        Toast.makeText(MainActivity.this,  
            String.valueOf(ratingBar.getRating()),  
            1).show();  
    }  
});  
}
```

Uygulamayı çalıştırdığımız da :



## Android Studio Programlama Bölüm 19

---

### Android Studio checkBox Kullanımı

Android studio'da checkbox kullanımını göreceğiz. Türkçe de işaret kutucuğu olarak adlandırılan checkbox'u anket örneği olarak mesela Cinsiyet sorusuna işaretlediğiniz Erkek Bayan örneği gibi düşünebilirsiniz.

Öncelikle bir tasarım oluşturalım. Tasarım bir textView'deki Hangi takımı tutuyorsunuz sorusuna 6 tane checkbox'tan birini seçerek cevap vermesi ile olsun. Gönder butonuna basınca da "Tuttuğunuz Takım -İşaretlediğimiz-" kısım olsun.

activity\_main Kodu :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_margin="10dp"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hangi Takımı Tutuyorsunuz ?"
        android:textSize="16sp"
        android:textStyle="bold"
    />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" >

        <LinearLayout
            android:layout_width="odp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:orientation="vertical" >

            <CheckBox
                android:id="@+id/checkBox1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Beşiktaş"
                android:checked="false" />

            <CheckBox
                android:id="@+id/checkBox2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Galatasaray"
                android:checked="false" />

            <CheckBox
                android:id="@+id/checkBox3"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Bursaspor"
```

```

        android:checked="false" />
</LinearLayout>

<LinearLayout
    android:layout_width="odp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:orientation="vertical" >

    <CheckBox
        android:id="@+id/checkBox4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fenerbahçe"
        android:checked="false" />

    <CheckBox
        android:id="@+id/checkBox5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Trabzonspor"
        android:checked="false" />

    <CheckBox
        android:id="@+id/checkBox6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Diğer"
        android:checked="false" />
</LinearLayout>
</LinearLayout>

<Button
    android:id="@+id/gonder"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Gönder"
    android:layout_marginTop="10dp"
    android:layout_gravity="center_horizontal" />

</LinearLayout>

```

Java Kodu :

Java kodumuzda öncelikle nesnelerimizi tanıttık.

```

cb1.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(cb1.isChecked())
            Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
    }
});

```

Kodu ile cb1 seçili ise bir Toast mesaj gösterdik. Tıklađığımız kutucuk yazı olarak önümüze çıkıcak. takım isimli String deđişkenimizde her checkbox için isChecked() metodu ile seçili olup olmadığını seçili ise de button'a tıklayınca Toast şeklinde göstermesini istedik.

## Genel hali ile Checkbox Kullanımı MainActivity kodları..

```
package com.umiitkose.egitim2;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.Toast;

public class MainActivity extends Activity {

    private CheckBox cb1;
    private CheckBox cb2;
    private CheckBox cb3;
    private CheckBox cb4;
    private CheckBox cb5;
    private CheckBox cb6;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //Tasarımdaki Checkbox'lar ı ekiyoruz.
        cb1 = (CheckBox)findViewById(R.id.checkBox1);
        cb2 = (CheckBox)findViewById(R.id.checkBox2);
        cb3 = (CheckBox)findViewById(R.id.checkBox3);
        cb4 = (CheckBox)findViewById(R.id.checkBox4);
        cb5 = (CheckBox)findViewById(R.id.checkBox5);
        cb6 = (CheckBox)findViewById(R.id.checkBox6);

        //CheckBox'ların Listener'larını tanımlıyoruz.
        cb1.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(cb1.isChecked())
                    Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
            }
        });
        cb2.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(cb2.isChecked())
                    Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
            }
        });
        cb3.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(cb3.isChecked())
                    Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
            }
        });
        cb4.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
```

```

        if(cb4.isChecked())
            Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
    }
});
cb5.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(cb5.isChecked())
            Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
    }
});
cb6.setOnCheckedChangeListener(new OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if(cb6.isChecked())
            Toast.makeText(getApplicationContext(), buttonView.getText(),
Toast.LENGTH_SHORT).show();
    }
});

// Butonu tıklayıyoruz ve tıklayıldığında işaretli betikleri ekrana basıyoruz.
Button show = (Button) findViewById(R.id.gonder);
show.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        String takim="Tuttuğunuz Takım : \n";
        if(cb1.isChecked())
            takim+= " "+cb1.getText();
        if(cb2.isChecked())
            takim+= " "+cb2.getText();
        if(cb3.isChecked())
            takim+= " "+cb3.getText();
        if(cb4.isChecked())
            takim+= " "+cb4.getText();
        if(cb5.isChecked())
            takim+= " "+cb5.getText();
        if(cb6.isChecked())
            takim+= " "+cb6.getText();

        Toast.makeText(getApplicationContext(), takim, Toast.LENGTH_LONG).show();
    }
});
}
}
}

```

3G 5:38

Hangi Takimi Tutuyorsunuz ?

Beşiktaş  Fenerbahçe

Galatasaray  Trabzonspor

Bursaspor  Diğer

Gönder

3G 5:47

Hangi Takimi Tutuyorsunuz ?

Beşiktaş  Fenerbahçe

Galatasaray  Trabzonspor

Bursaspor  Diğer

Gönder

Cevaplar:  
Beşiktaş

# Android Studio Programlama Bölüm 20

---

## Android Studio Radio Button

Android Studio'da sıra Radio Button kullanımında. Radio button web sayfalarındaki çokça kullandığımız checkbox gibi bir işaret kutucuğu diyebiliriz. CheckBox'tan farkı ise checkbox'ta birden fazla işaretçiyi seçebilirken, Radio Button'da tek bir işaretçiyi seçmemiz gerekmektedir. Kodlamaya Başlırsak

Öncelikle activity\_main :

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <RadioGroup
        android:id="@+id/radioSex"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <RadioButton
            android:id="@+id/radioMale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Erkek"
            android:checked="true" />

        <RadioButton
            android:id="@+id/radioFemale"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Bayan" />

    </RadioGroup>

    <Button
        android:id="@+id/btnDisplay"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gonder" />

</LinearLayout>
```

MainActivity kodlarımız:

Bir buton'da işaretli olan radio button'u aldık. Button'a tıklanıldığında radio Group'taki seçili id'yi aldık ve button'da bir Toast Mesajı ile gösterdik.

```
package com.umiitkose.egitim2;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
```



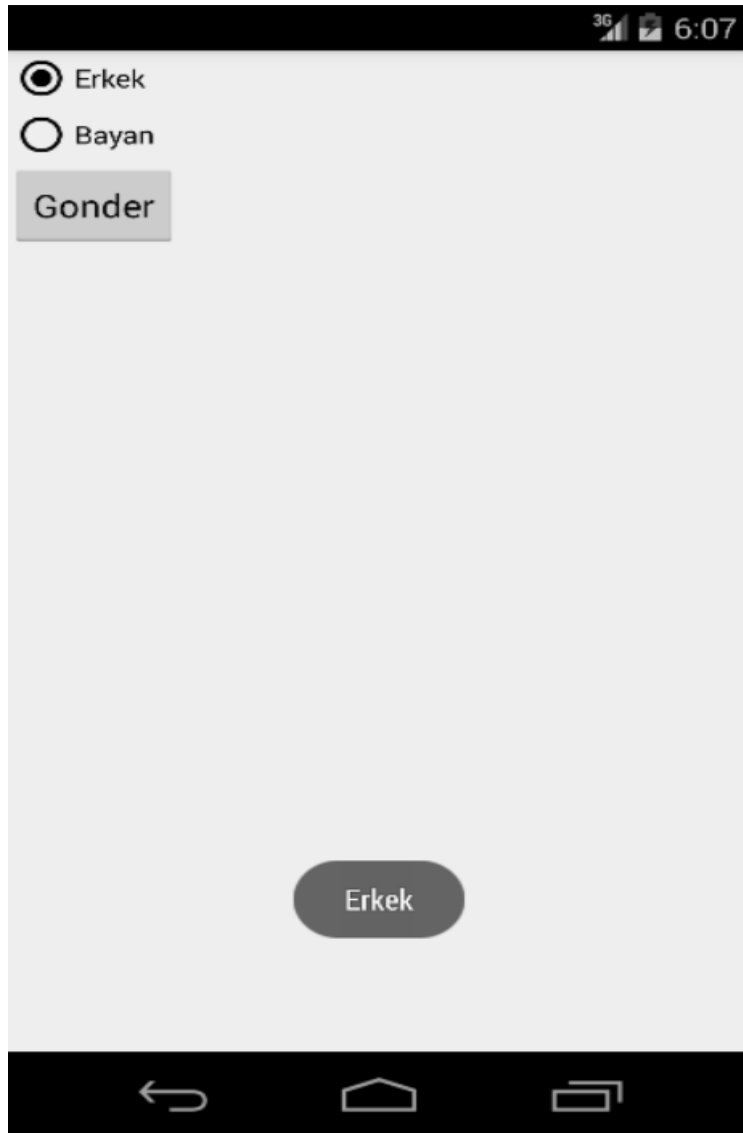
```
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

public class MainActivity extends Activity {

    private RadioGroup rG;
    private RadioButton rB;
    private Button gonder;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rG= (RadioGroup) findViewById(R.id.rG);
        gonder= (Button) findViewById(R.id.gonder);
        gonder.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View v) {
                int selectedId = rG.getCheckedRadioButtonId();
                rB= (RadioButton) findViewById(selectedId);
                Toast.makeText(MainActivity.this, rB.getText(), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Uygulamanın Çalışması :



---

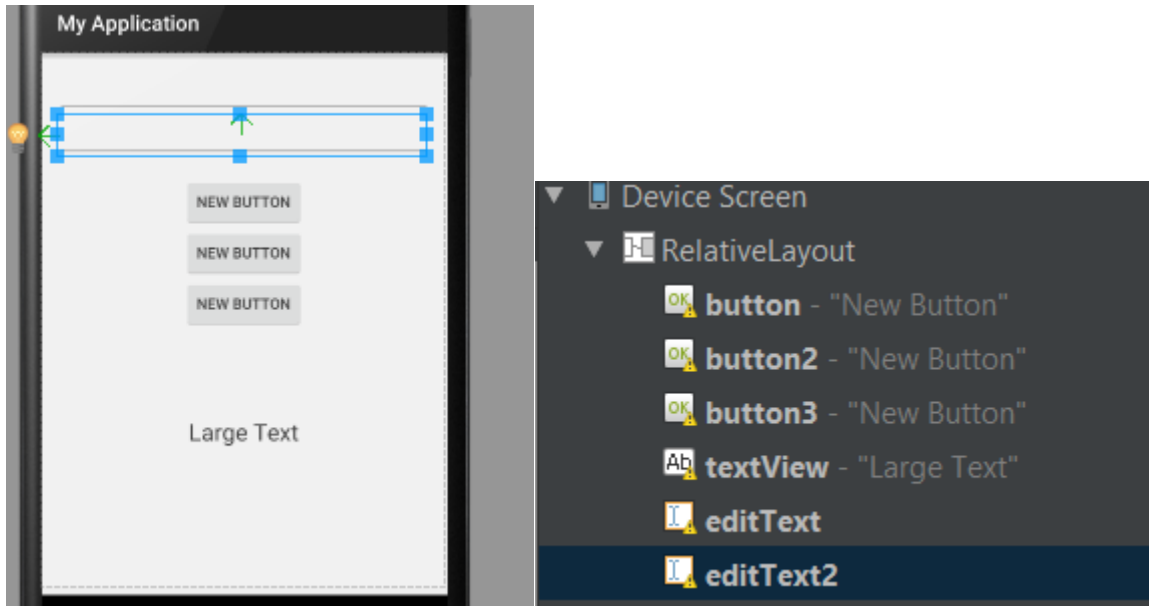
---

## Android Studio Programlama Bölüm 21

### Android Studio Buttonların tek bir listener ile kontrolü

Bu seferki dersimizin amacı switch case yapısı kullanarak birden fazla butonu tek bir listener ile kontrol edicez. Yani biz her button için tek tek tıklama olayı tanımlamıştık Şimdi ise tek bir tıklamada hepsini kontrol ediceğiz. Mesela 1 button 2 sayıyı toplasın. 1 butonumuz text View'e Ali yazdırsın. 1 Buton'da Toast Mesaj göstererek Button1 'e tıkladınız desin.

Tasarım Kısmı:



Tasarımın Kodları :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button"
    android:layout_below="@+id/button2"
    android:layout_centerHorizontal="true" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button2"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
```

```
        android:layout_marginTop="101dp" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button3"
    android:layout_below="@+id/button"
    android:layout_centerHorizontal="true" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Large Text"
    android:id="@+id/textView"
    android:layout_below="@+id/button3"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="84dp" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:layout_below="@+id/editText"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

</RelativeLayout>
```

Java Kodumuz : View.OnClickListener'dan 'dan kalıtım implement alıyoruz. Ve butonlarımızı tanıtıyoruz. ve hepsini tek bir noktada tıklama veriyoruz. Altta ise aldığımız getId'leri ile her buton'un idsi için işlem yaptırıyoruz.

```

public class MainActivity extends ActionBarActivity implements View.OnClickListener{
    TextView tv;
    Button b1,b2,b3;
    EditText a1,a2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=(Button)findViewById(R.id.button);
        b2=(Button)findViewById(R.id.button2);
        b3=(Button)findViewById(R.id.button3);
        b1.setOnClickListener(this);
        b2.setOnClickListener(this);
        b3.setOnClickListener(this);
        tv=(TextView)findViewById(R.id.textView);
        a1=(EditText)findViewById(R.id.editText);
        a2=(EditText)findViewById(R.id.editText2);
    }
}

```

```

@Override
public void onClick(View view) {
    switch (view.getId()){
        case R.id.button:
            Toast.makeText(getApplicationContext(), "Buton 1'e tiklandi.", Toast.LENGTH_SHORT).show();
            break;
        case R.id.button2:
            tv.setText("Ali");
            break;
        case R.id.button3:
            float sayi1float=Float.parseFloat(a1.getText().toString());

            float sayi2float=Float.parseFloat(a2.getText().toString());

            tv.setText(String.valueOf(sayi1float+sayi2float));

            break;
    }
}
}

```

## Java Kodumuz :

```

package com.umiitkose.myapplication;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

```

```
import android.widget.TextView;
import android.widget.Toast;
```

```
public class MainActivity extends ActionBarActivity implements View.OnClickListener{
    TextView tv;
    Button b1,b2,b3;
    EditText a1,a2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=(Button)findViewById(R.id.button);
        b2=(Button)findViewById(R.id.button2);
        b3=(Button)findViewById(R.id.button3);
        b1.setOnClickListener(this);
        b2.setOnClickListener(this);
        b3.setOnClickListener(this);
        tv=(TextView)findViewById(R.id.textView);
        a1=(EditText)findViewById(R.id.editText);
        a2=(EditText)findViewById(R.id.editText2);

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()){
            case R.id.button:
                Toast.makeText(getApplicationContext(), "Buton 1'e tiklandı.", Toast.LENGTH_SHORT).show();
                break;
            case R.id.button2:
                tv.setText("Ali");
                break;

            case R.id.button3:
                float sayi1float=Float.parseFloat(a1.getText().toString());

                float sayi2float=Float.parseFloat(a2.getText().toString());

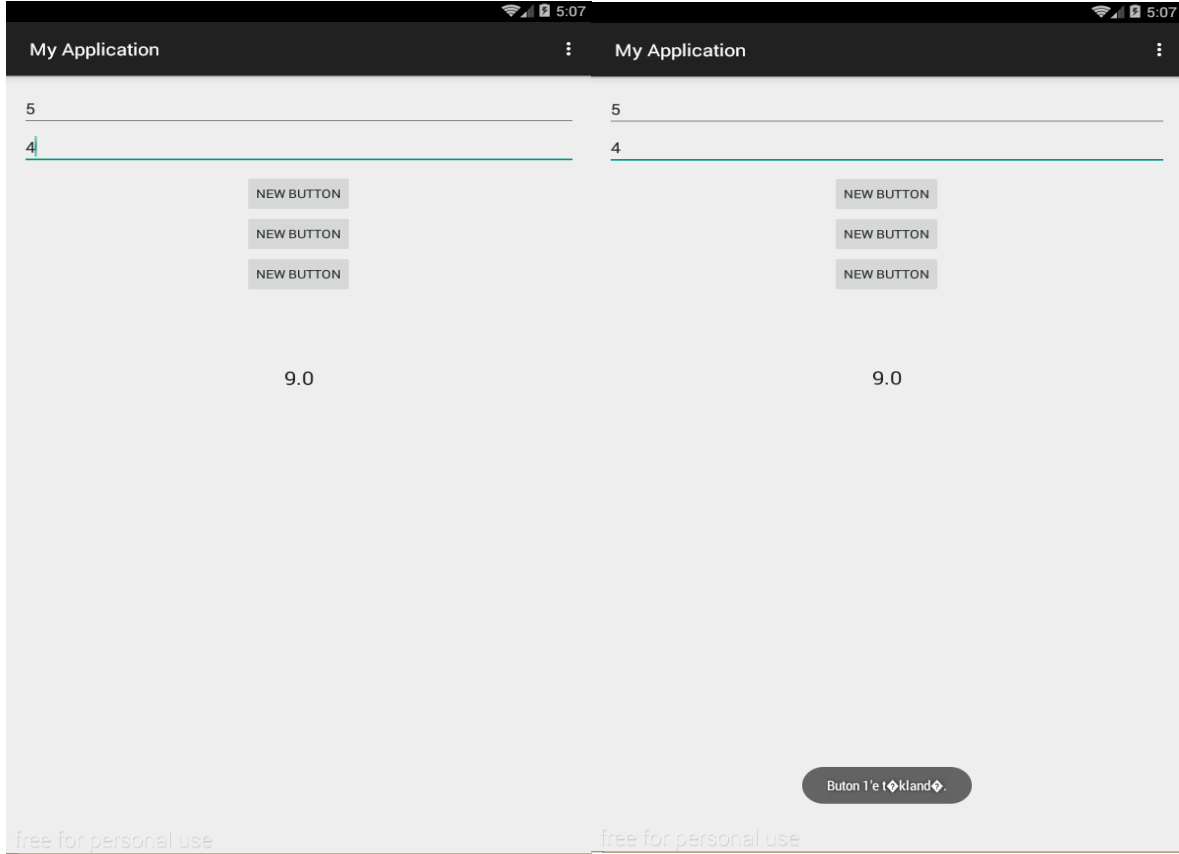
                tv.setText(String.valueOf(sayi1float+sayi2float));
        }
    }
}
```

```
}  
} break;  
}
```

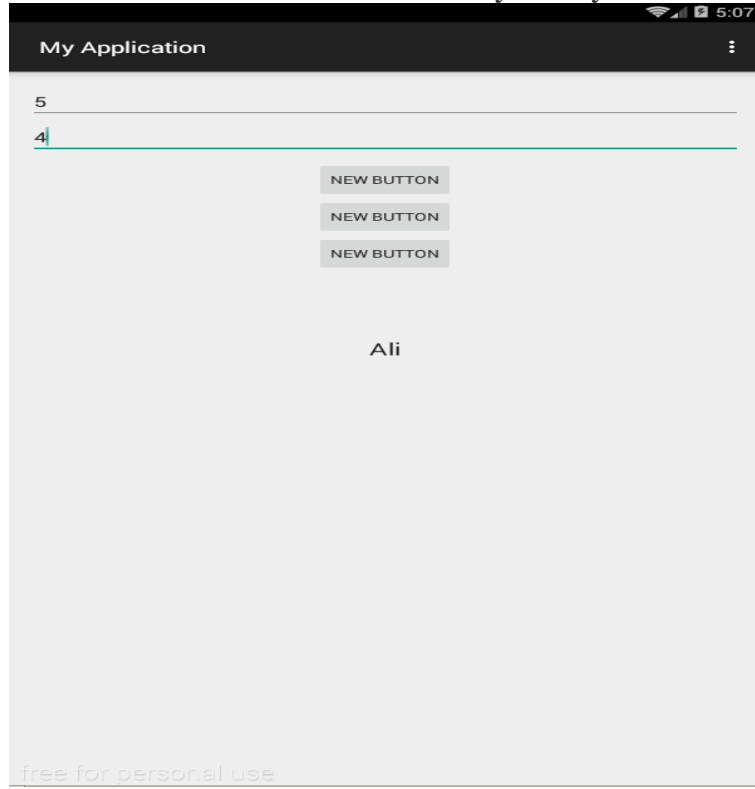
Uygulamanın Çalışması.

Bir buton'da Toplama işlemi

Bir butonda Toast mesaj



Bir butonda da Ekрана ali yazdırıyor.





## Android Studio Programlama Bölüm 22

---

### Android Studio Wifi Kullanımı

Android Studio'da 20. Dersimiz kablosuz internet erişimleri sağlayan wifi'nin çalışmasıyla ilgili. Wifi akıllı telefonlarda kablosuz internet bağlantısı erişimine olanak verir. Bir toggle button ile Android Studio'da wifi kontrolü sağlayarak işlemlerimizi gerçekleştireceğiz. Toggle button on ve off şeklinde içerisinde 2 kontrol mekanizması olan button'dur.

Activity\_main:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <ToggleButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/a"
        android:id="@+id/toggleButton"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />
</RelativeLayout>
```

Java Kodları : Açıklamaları kodların yanında yaptım.

```
package com.umiitkose.myapplication;

import android.bluetooth.BluetoothAdapter;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.MediaRecorder;
import android.net.Uri;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.app.Activity;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends Activity {
    //Wifi managerdan modem değişkeni alıyoruz.

    WifiManager modem=null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        /*ToggleButton'u bulup aktif ediyoruz. Eğer on kısmındaysa seçiliyse wifiAc metoduna geçecek ve wifi
        Acıcağ. Off kısmında ise wifi kapat metodunu çağırıp kapatıcak.
```

```

*/
final ToggleButton btn=(ToggleButton)findViewById(R.id.toggleButton);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(btn.isChecked()){
            wifiAc();
        }
        else
            wifiKapat();
    }
});
} //Modem'in wifi durumu kapalıysa wifiyi açıyor. Açıkça zaten açık diyor.
private void wifiAc(){
    modem=(WifiManager) getSystemService(Context.WIFI_SERVICE);
    if(modem.getWifiState()== WifiManager.WIFI_STATE_DISABLED){
        modem.setWifiEnabled(true);
    }else if(modem.getWifiState()==WifiManager.WIFI_STATE_ENABLING){
        Toast.makeText(MainActivity.this, "Açık", Toast.LENGTH_SHORT).show();
    }
}

//Eğer wifi açık ise kapatıyor.
private void wifiKapat(){
    modem=(WifiManager) getSystemService(Context.WIFI_SERVICE);
    if(modem.getWifiState()== WifiManager.WIFI_STATE_ENABLED){
        modem.setWifiEnabled(false);
    }else if(modem.getWifiState()==WifiManager.WIFI_STATE_DISABLING){
        Toast.makeText(MainActivity.this, "Zaten Kapalı", Toast.LENGTH_SHORT).show();
    }
}
}
}

```

### Android Manifest İzinlerimiz:

```

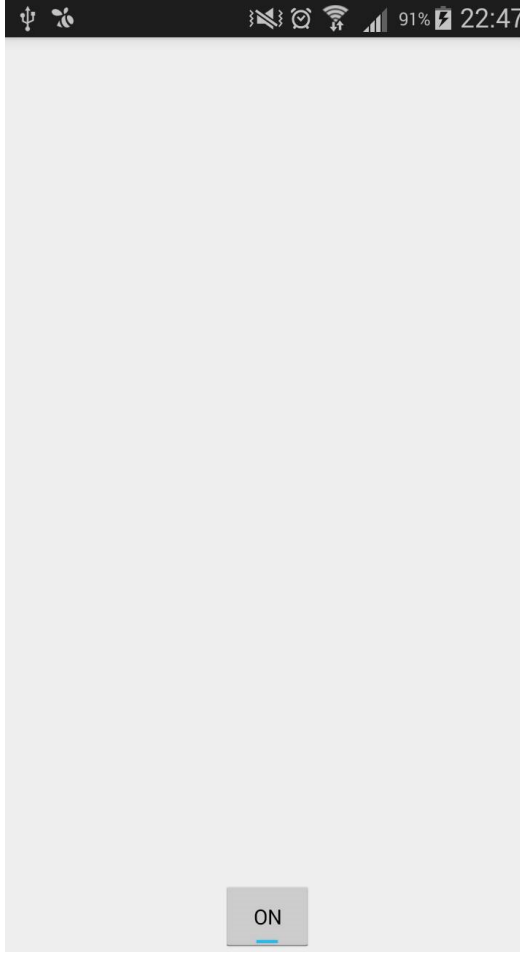
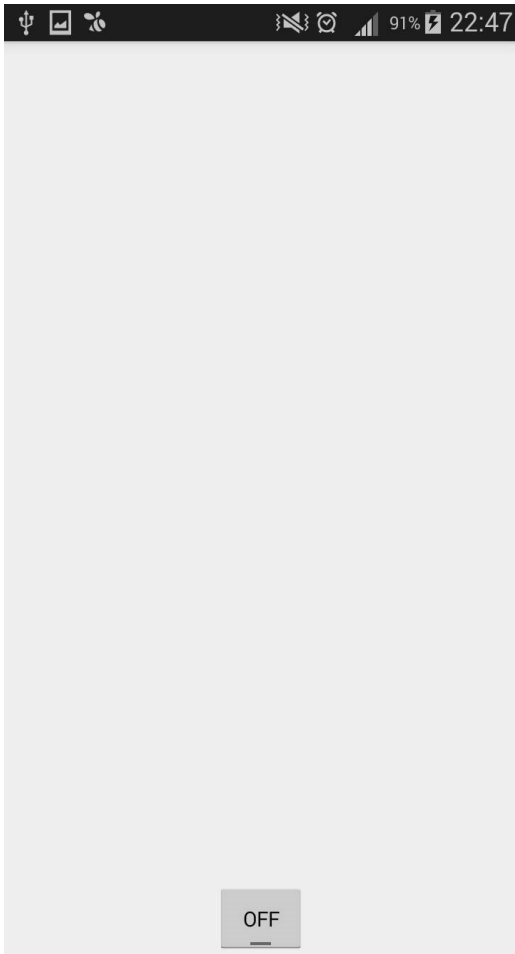
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />

```

Uygulamamızın çalışması. Emülatör'de denemeyin. [Android Studio'da apk dosyası oluşturmak için tıklayın.](#)

Off Kısmı Wifi Kapalı

On kısmı wifi açık.



## Android Studio Programlama Bölüm 23

---

### Android Studio Kamera'dan Görüntü Alma ve Görüntüyü Gösterme

Android studio'da Kamera kullanımındayız. Kamera işlemi yaparken android tarafında neler yapıyoruz. Hep birlikte gelin bunu inceleyelim.

1 button ve image view ile activity\_main dosyamızı oluşturduk.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="New Button"
    android:id="@+id/button"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/imageView"
    android:layout_above="@+id/button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginBottom="108dp" />
```

```
</RelativeLayout>
```

Java Dosyamız :

Java dosyasında öncelikle resim çekme işlemi aktif ediyoruz. İd ile isteğimizi onaylayıp imageView'e atarak gösterimini sağlıyoruz.

```
package com.umiitkose.myapplication;

import android.bluetooth.BluetoothAdapter;
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.media.MediaRecorder;
import android.net.Uri;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.app.Activity;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
```

```

import android.widget.Toast;
import android.widget.ImageView;

public class MainActivity extends Activity {

    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn=(Button)findViewById(R.id.button);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent kamera=new Intent(MediaStore.ACTION_IMAGE_CAPTURE); // Resim çekme isteği ve
                activity başlatılıp id'si tanımlandı
                startActivityForResult(kamera,33);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        if(requestCode==33){
            Bitmap image=(Bitmap)data.getExtras().get("data");//Çekilen resim id olarak bitmap şeklinde alındı
            ve imageView'e atandı
            ImageView resim= (ImageView)findViewById(R.id.imageView);
            resim.setImageBitmap(image);
        }

        // super.onActivityResult(requestCode, resultCode, data);
    }
}

```

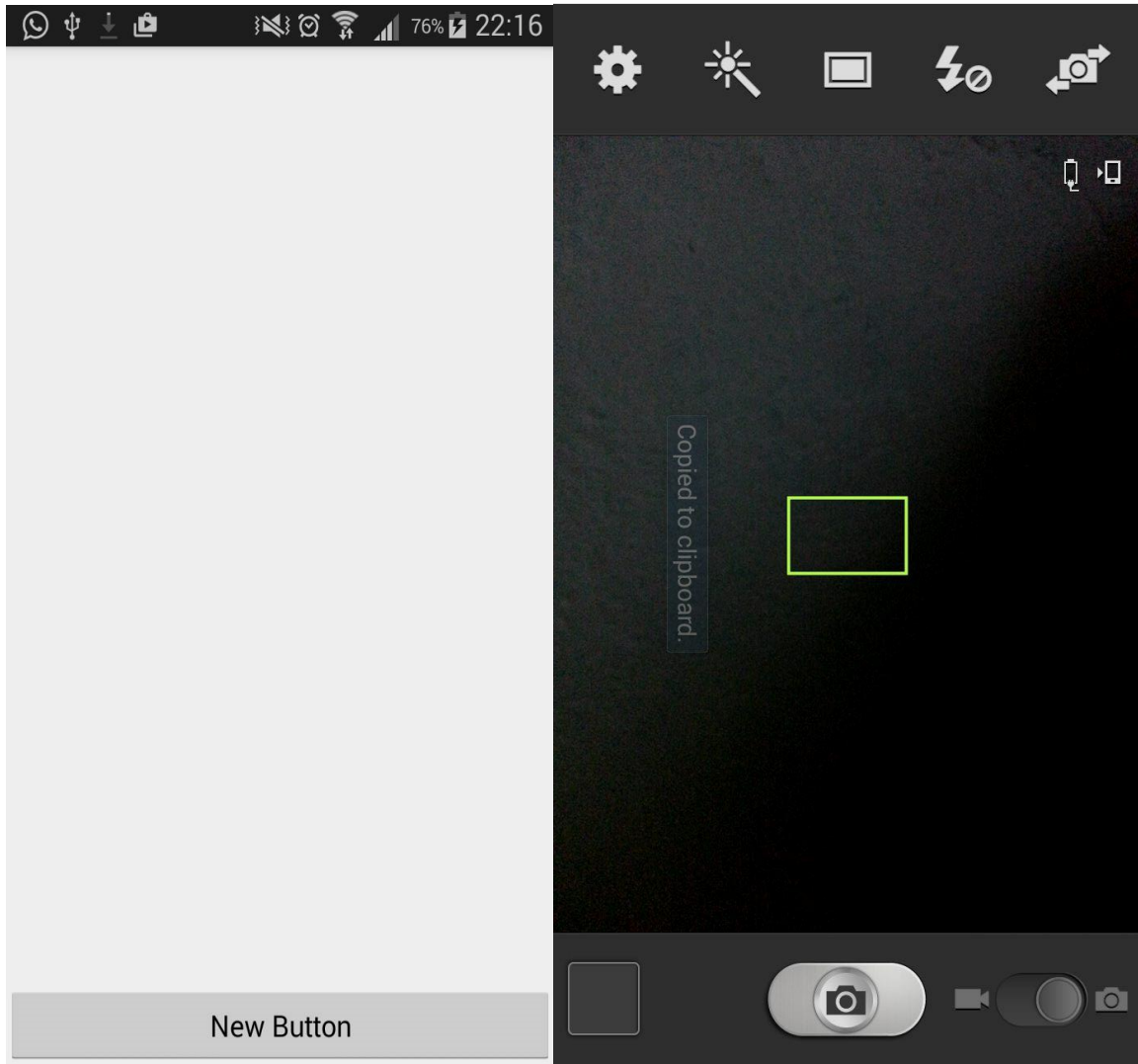
android studio Manifest İçin gerekli izinler..

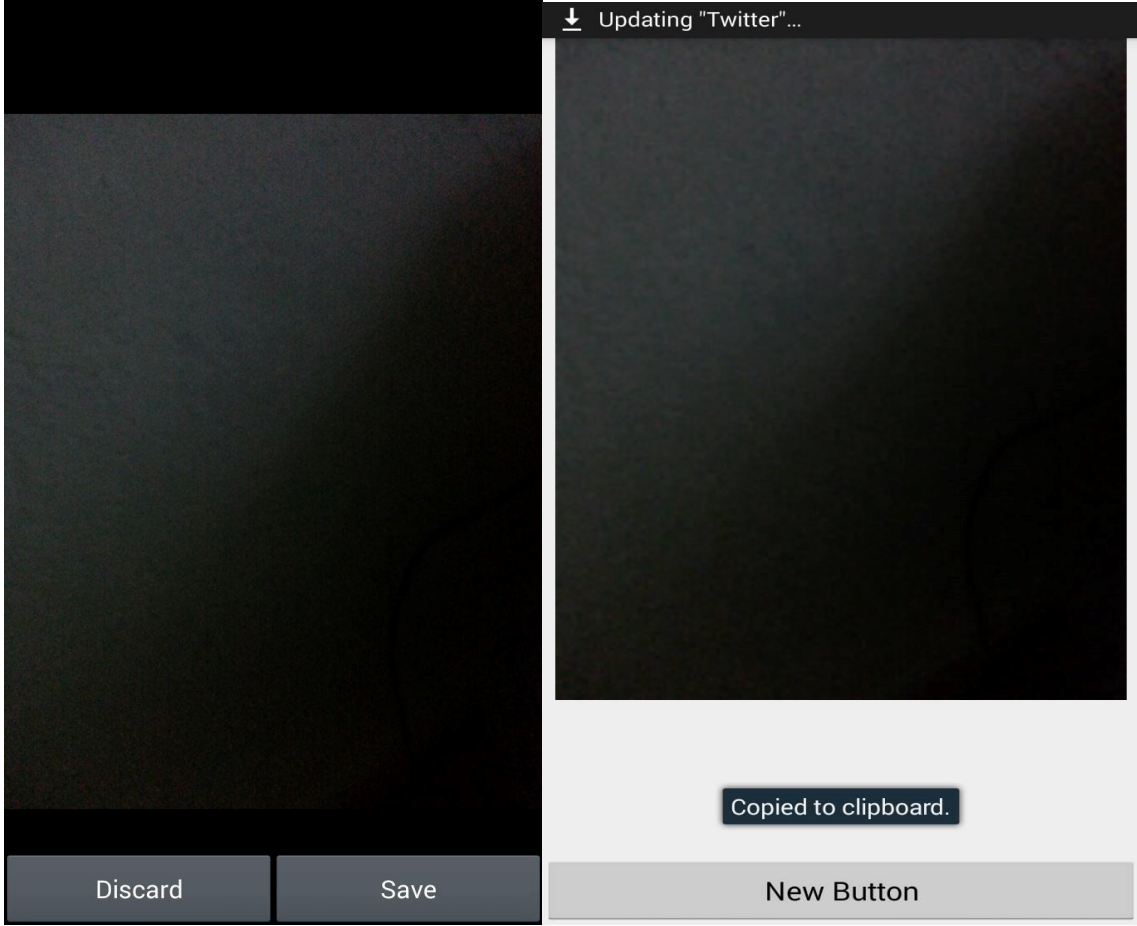
```

<uses-permission android:name="android.permission.CAMERA"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
<uses-feature android:name="android.hardware.Camera"></uses-feature>

```

Uygulamayı apk olarak android cihazınız da çalıştırınız [.Android Studio'da apk dosyası oluşturmak için tıklayın.](#)





Uygulama öncelikle Button'a basınca kameraya bağlanıyor. Kameradan aldığımız görüntüyü kaydedip, kaydetmeyeceğimizi soruyor. Kaydederseniz Görüntü Telefonda gözüküyor.

---

## Android Studio Programlama Bölüm 24

---

### Android Studio Video Çekme ve Oynatma

Android studio'da video çekme ve oynatma işlemleri yapacağız. Android'te hepimiz videolarla ilgilenmişizdir. Peki bir video çekerken android ne gibi işlemlerden geçiyor hiç düşündünüz mü ? Hep birlikte ona bakalım..

Öncelikle 1 tane videoView ve button'a ihtiyacımız var.

activity\_main kodları:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <VideoView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/videoView"
        android:layout_above="@+id/button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="180dp" />

</RelativeLayout>
```

Şimdi Java kodlarımız :

1 tane button'a Video kaydetme işlemini verdik. ve bir id atadık. 33 ile kamera görüntüleme idsi oldu. Bu istek ile verimizi aldık ve video view'e çektiğimiz veri aracılığıyla aktardık. Aktarmak içinde dikkat ettiyseniz 33 idsini kullandık.

```
package com.umiitkose.myapplication;
```

```
import android.bluetooth.BluetoothAdapter;
import android.content.Context;
import android.content.Intent;
import android.media.MediaRecorder;
```



```

import android.net.Uri;
import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.app.Activity;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.Toast;
import android.widget.VideoView;

public class MainActivity extends Activity {

    Button btn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btn=(Button)findViewById(R.id.button);

        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent kamera=new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
                startActivityForResult(kamera,33);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        VideoView video=(VideoView)findViewById(R.id.videoView);
        if(requestCode==33){
            Toast.makeText(MainActivity.this, "Kamera Butonu Kullanıldı.", Toast.LENGTH_SHORT).show();
            Uri abc=data.getData();
            video.setVideoPath(String.valueOf(abc));
            video.start();
        }

        // super.onActivityResult(requestCode, resultCode, data);
    }
}

```

## Android Studio için gerekli izinler..

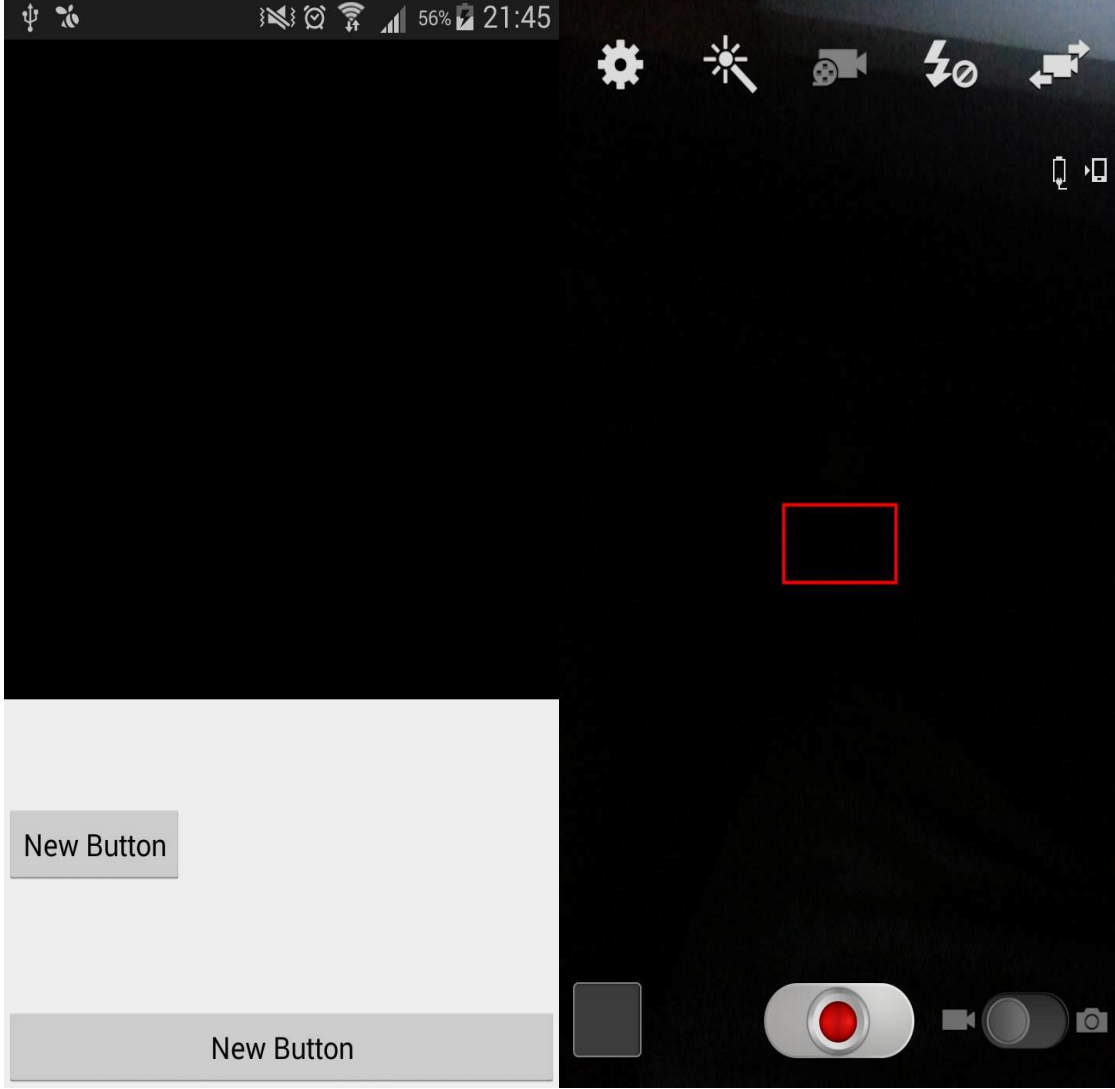
```

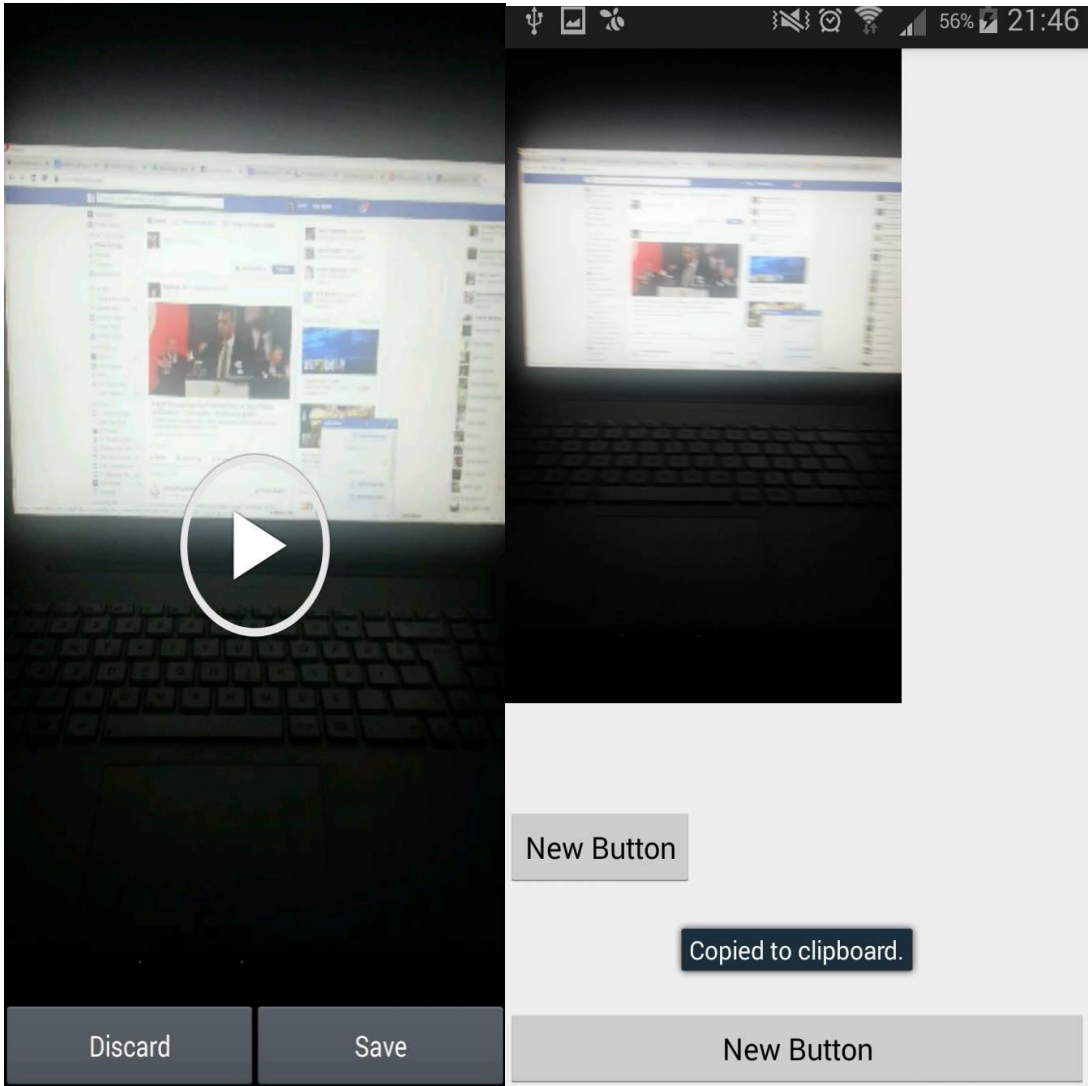
<uses-permission android:name="android.permission.CAMERA"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-
permission>
<uses-feature android:name="android.hardware.Camera"></uses-feature>

```

Emülatörde lütfen denemeyiniz çalışmamaktadır. [Android Studio'da apk dosyası oluşturmak için tıklayın.](#)

Uygulamanın çalışması :





# Android Studio Programlama Bölüm 25

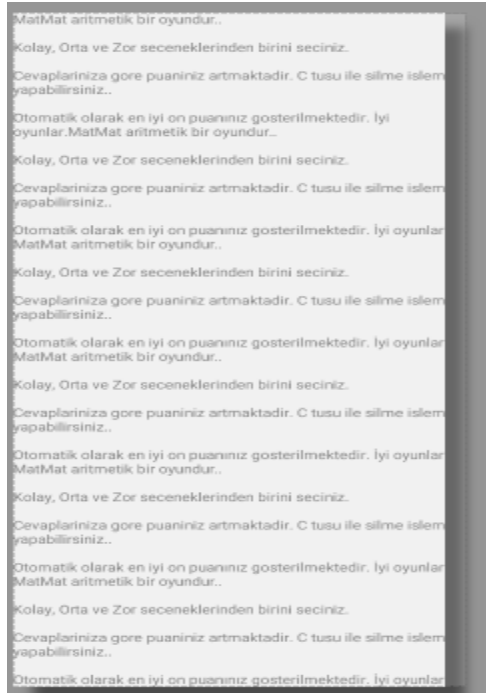
## Android Studio ScrollView Kullanımı

Android'te işlem yaparken bazen öyle bir uygulama yazarsınız ki ekran'ınız yetmeyebilir. Bunun için scrollview kullanmanız gerekmektedir. Android Studio'da scroll view kullanırken öncelikle bu uygulama da scrollview'i kullanıp içerisinde bir linear layout attım. Ve values'teki string.xml'e girdiğim text'i ekrandan büyük bir yazı olarak ayarladım.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:weightSum="1">

        <TextView
            android:id="@+id/tv_long"
            android:layout_width="403dp"
            android:layout_height="match_parent"
            android:text="@string/yazi">
        </TextView>

    </LinearLayout>
</ScrollView>
```



---

## Android Studio Programlama Bölüm 26

---

### Android Studio Dil Desteđi

Android studio'da kullanıcılara dil desteđi verebiliyoruz. Mesela Siz telefonunuzu Türkçe kullanıyorsunuz ve uygulamanızın yurt dıřında da ilgi görmesini istiyorsunuz. Bunun için yapacađınız uygulamınıza dil paketi eklemek.

Basit bir uygulama geliřtirelim. Hello world'u Türkçe dil desteđi ile Merhaba Dünya'ya dönüřtürelim

1-Res klasörünün içine values-tr klasörünü ekleyin.

2-Values klasöründeki strings.xml'i kopyalayıp values-tr içine atınız.

```
MainActivity.java × activity_main.xml × styles.xml × strings.xml × tr\strings.xml ×
Edit translations for all locales in the translations editor.
<resources>
  <string name="app_name">My Application</string>
  <string name="hello_world">Merhaba Dünya!</string>
  <string name="action_settings">Settings</string>
</resources>
```

Hello World yazısını bizim stringimizde Merhaba Dünya olarak deđiřtirin.

Uygulamayı çalıştırdığınızda Hello World yazar. Dil ayarlarından Türkçeyi seçerseniz Merhaba Dünya ile karşılaşacaksınız.

## Android Studio Programlama Bölüm 27

### Android Studio MediaPlayer sınıfı ile ses kontrolü

Bu dersimizde basit bir ses uygulaması geliştireceğiz. Ses'in oynatılması, durdurulması ve bekletilmesi olayını yapacağız.

MediaPlayer sınıfına ihtiyacımız var bunun için. ve bir tane de ses dosyasına. Ses dosyasını res klasöründe raw klasörü açıp içine atınız. İsim olarak kısa birşeyler koyunuz. Bazı karakterleri kabul etmiyor. Türkçe karakter vs.

Tasarımımız bu şekilde olacak. Kodları da yanında verilmiştir.



Activity\_main kodlarımız :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">
```

```
<Button
```

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Oynat"
android:id="@+id/button"
android:layout_marginTop="151dp"
android:layout_marginLeft="38dp"
android:layout_marginStart="38dp"
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Durdur"
    android:id="@+id/button2"
    android:layout_alignTop="@+id/button3"
    android:layout_toRightOf="@+id/button3"
    android:layout_toEndOf="@+id/button3" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Beklet"
    android:id="@+id/button3"
    android:layout_alignTop="@+id/button"
    android:layout_centerHorizontal="true" />

```

```

</RelativeLayout>

```

MainActivity'miz bu şekilde olacak. Kodları yanında açıkladık.

```

package com.umiitkose.myapplication;

import android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    Button b1,b2,b3;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final MediaPlayer ses= MediaPlayer.create(this, R.raw.ses);//raw klasöründe ses adlı dosyam var
        b1=(Button)findViewById(R.id.button);
        b2=(Button)findViewById(R.id.button2);
        b3=(Button)findViewById(R.id.button3);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                ses.start(); //ses butona basınca oynatılıyor
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```
        ses.stop();} //ses butona basınca duruyor
    });
    b3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) { // Bu kısım bekletme kısmı.
            ses.pause();
        }
    });
}
```

Eğer bekletme ve durdurma işlemini bir butonda yapıcaksanızda kodumuz basit.

```
if(ses.isPlaying()){
    ses.pause();
}
else{
    ses.start();
}}
```



## Android Studio Programlama Bölüm 28

---

### Android Studio Paylaş Butonu

Android'te yaptığımız uygulamayı belirli yerlerde paylaşmak isteriz. Bunun için hazır paylaş butonu kısmı var. Bu dersimizde onunla ilgili örnek vericeğim.

Activity\_main.xml kodları :

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/btnShare"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/lblContent"
        android:layout_marginTop="50dp"
        android:text="Paylaş" />

    <EditText
        android:id="@+id/txtContent"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:layout_toRightOf="@+id/lblContent"
        android:ems="10" >

        <requestFocus />
    </EditText>

    <TextView
        android:id="@+id/lblContent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Paylaşılacak yazı" />

</RelativeLayout>
```

Basit bir editText ile tanımlanmış yazıda textView içine girilen veri alınarak bir buton yardımıyla paylaşabiliyoruz.

MainActivity.java dosyasının kodları  
package googleyardim.example.info.paylas;

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Button btnShare = (Button) findViewById(R.id.btnShare);
final EditText txtContent = (EditText) findViewById(R.id.txtContent);
btnShare.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

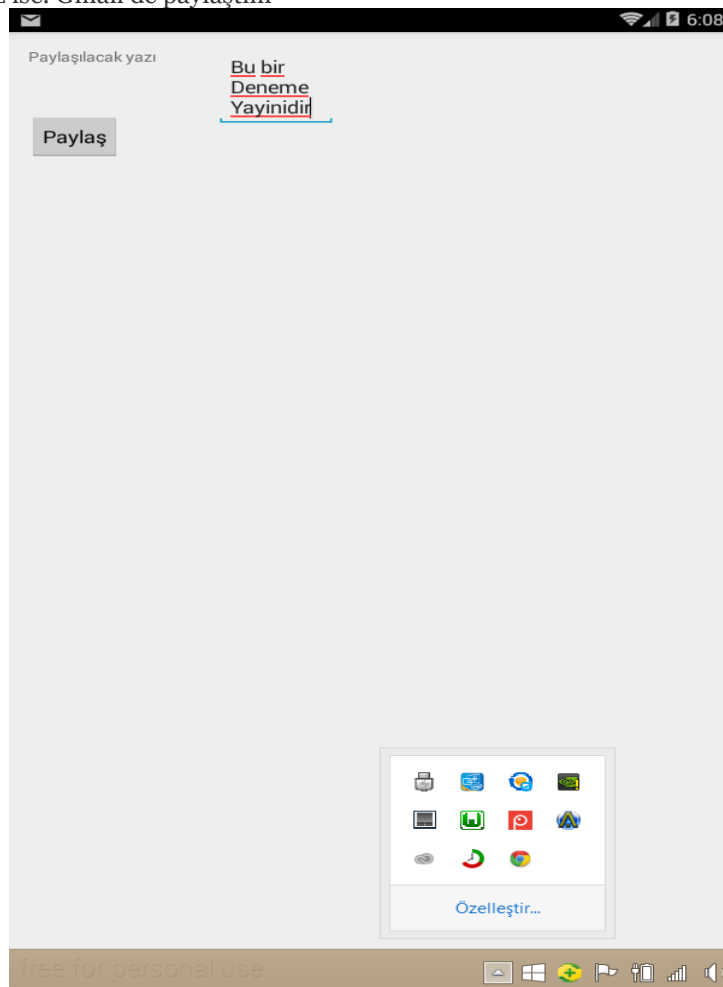
        // EditText'in içindeki yazıyı alır.
        String message = txtContent.getText().toString();

        shareMyMessage(message);
    }
});
}

protected void shareMyMessage(String _message) {
    // TODO Auto-generated method stub
    Intent share = new Intent(Intent.ACTION_SEND);
    share.setType("text/plain");
    share.putExtra(Intent.EXTRA_TEXT, _message);
    startActivity(Intent.createChooser(share, "Gönderiyi paylaş !!"));
}
}

```

Son olarak görselimiz ise. Gmail'de paylaştım



asdsad

Gelen Kutusu x



uuumtkose@gmail.com <uumtkose@gmail.com>

01:02 (0 dakika önce)



Alıcı: bana

Bu bir Deneme Yayınıdır



[Yanıtla](#) veya [Yönlendir](#)

15 GB'lık kotasının 0 GB'ı (%0) kullanılıyor  
[Yönet](#)

[Şartlar](#) - [Gizlilik](#)

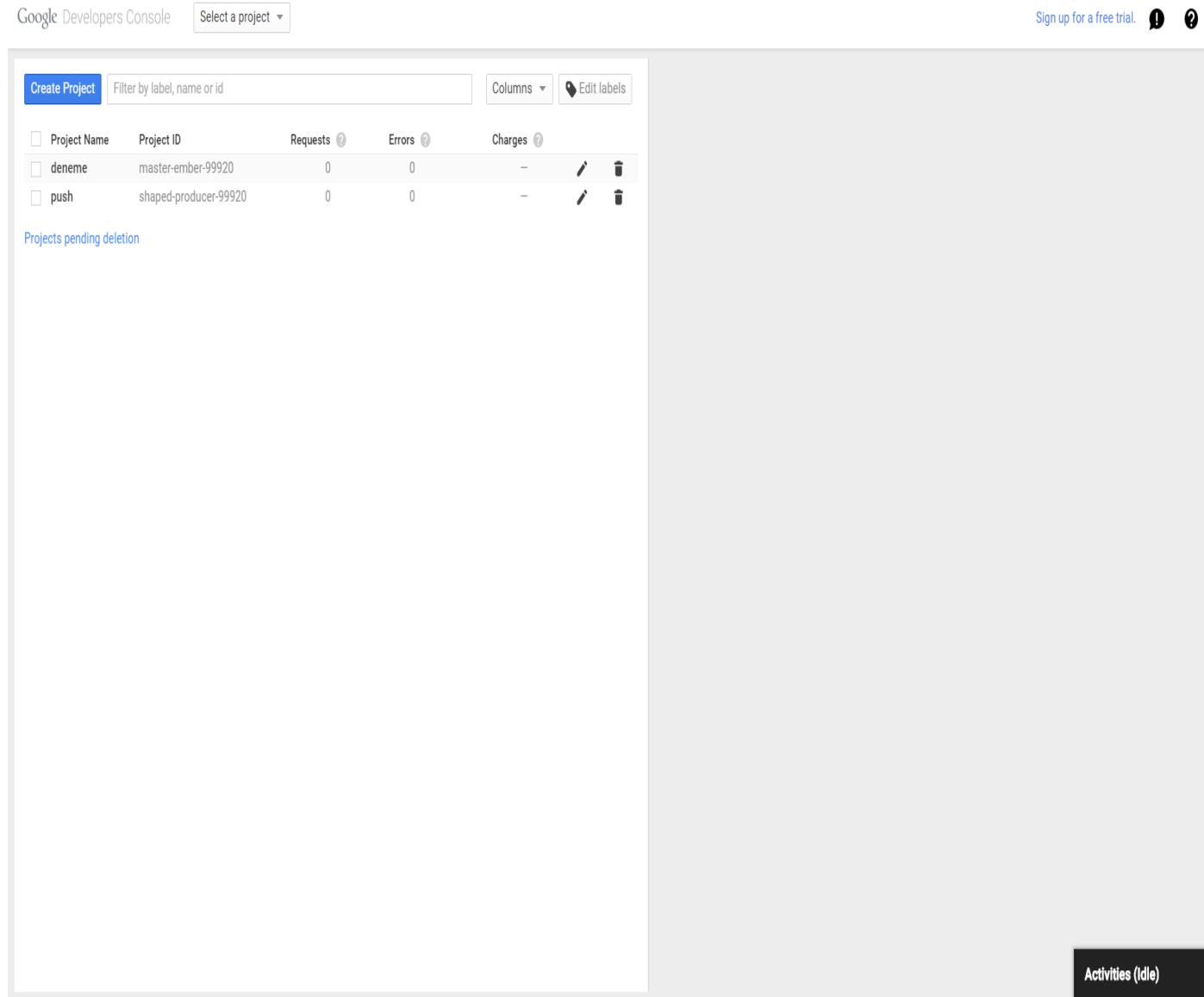
Son hesap etkinliği: 12 May  
[Ayrıntılar](#)

# Android Studio Programlama Bölüm 29

## Android Studio Pushbots ile Bildirim Göndermek

Telefonda herhangi bir olayda bildirim gelebiliyor. Örneğin facebook'ta yeni yorum gibi. Bizde dışardan kendi uygulamamız ile ilgili bilgiler vericek bir bildirim örneği vericeğiz. Öncelikle bunu bir web sitesi ile yapcaz. Tabi Google developers'tanda api key almamız gerekmektedir.

<https://console.developers.google.com/project> ' a girerek bir proje oluşturuyoruz. Create Project'ten proje oluşturduktan sonra uygulamamızın adını giriyoruz. Ve projemiz kaydoluyor.



Google Developers Console Select a project Sign up for a free trial. ! ?

[Create Project](#)  Columns Edit labels

<input type="checkbox"/>	Project Name	Project ID	Requests ?	Errors ?	Charges ?		
<input type="checkbox"/>	deneme	master-ember-99920	0	0	-		
<input type="checkbox"/>	push	shaped-producer-99920	0	0	-		

[Projects pending deletion](#)

Activities (Idle)

-Proje İsmimizi giriyoruz. Ve Kare içindekileri not alıyoruz.

### New Project


Project name ?


Your project ID will be possible-origin-99921 ? [Edit](#)


[Show advanced options...](#)


Project ID: possible-origin-99921 Project Number: 460071397837 Estimated charges this month: \$0.00


### Project Dashboard


 **Deploy a Hello World app**  
App Engine lets you build and deploy an app without worrying about the underlying infrastructure. Learn how to use App Engine to create a Hello World app and deploy it in the cloud.

 **Try out the VM quickstart**  
Spin up virtual machines using Google Compute Engine, NodeJS, and MongoDB to create a guestbook app

 **Enable Google APIs for use in your apps**  
Let your apps take advantage of Google's products and services

 **Create a Cloud Storage bucket**  
Store your unstructured data safely and with high availability using Google Cloud Storage

 **Try BigQuery with population data**  
Run queries against huge public data sets to see how BigQuery can help you analyze your own data

 **Start with a prebuilt solution**  
Launch your favorite software packages on Google Cloud Platform with just a few clicks

**Activities (Idle)** 🗑️ - ✕

Create Project: deneme	✓
Create Project: push	✓
Create Project: deneme	✓
Create Project: Pushbots-project	✓

[See all activity](#)

Burada ise Cloud Messaging for Android'e girerek Android için işlemimizi aktif ediyoruz.

- Overview
- Permissions
- APIs & auth
- APIs
- Credentials
- Consent screen
- Push
- Monitoring
- Source Code
- Deploy & Manage
- Compute
- Networking
- Storage
- Big Data

API Library Enabled APIs (6)

Search all 100+ APIs

Popular APIs



Google Cloud APIs

- Compute Engine API
- BigQuery API
- Cloud Storage API
- Cloud Datastore API
- Cloud Deployment Manager API
- Cloud DNS API
- More



Google Maps APIs

- Google Maps Android API
- Google Maps SDK for iOS
- Google Maps JavaScript API
- Google Places API for Android
- Google Places API for iOS
- Google Maps Roads API
- More



Google Apps APIs

- Drive API
- Calendar API
- Gmail API
- Google Apps Marketplace SDK
- Admin SDK
- Contacts API
- CalDAV API



Mobile APIs

- Cloud Messaging for Android
- Google Play Game Services
- Google Play Developer API
- Google Places API for Android



Social APIs

- Google+ API
- Blogger API
- Google+ Pages API
- Google+ Domains API



YouTube APIs

- YouTube Data API
- YouTube Analytics API





Advertising APIs





- AdSense Management API
- DCM/DFA Reporting And Trafficking API
- Ad Exchange Seller API
- Ad Exchange Buyer API
- DoubleClick Search API
- Analytics API



Other popular APIs

- Translate API
- Custom Search API
- URL Shortener API
- PageSpeed Insights API
- Fusion Tables API
- Web Fonts Developer API

Activities (Idle)  

Create Project: deneme	
Create Project: push	
Create Project: deneme	
Create Project: Pushbots-project	

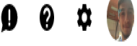
[See all activity](#)

## Enable Api'ye tıklayın ve aktifleştirin.

Google Developers Console

deneme

[Sign up for a free trial.](#)



Overview

Permissions

APIs & auth

APIs

Credentials

Consent screen

Push

Monitoring

Source Code

Deploy & Manage

Compute

Networking

Storage

Big Data

← Enable API

### Google Cloud Messaging for Android

Google Cloud Messaging allows for push messaging to Android devices.

[Learn more](#)

Activities (Idle) C \_ X

Create Project: deneme	✓
Create Project: push	✓
Create Project: deneme	✓
Create Project: Pushbots-project	✓

[See all activity](#)

Apıs & auth'tan credentials'tan Create new Key'e tıklayarak server key - Create işlemini gerçekleştiriyoruz.



- Overview
- Permissions
- APIs & auth
- APIs
- Credentials
- Consent screen
- Push
- Monitoring
- Source Code
- Deploy & Manage
- Compute
- Networking
- Storage
- Big Data

### OAuth

No client IDs found.

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

[Create new Client ID](#)

### Public API access

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

[Create new Key](#)

#### Create a new key



The APIs represented in the Google Developers Console require that requests include a unique project identifier. This enables the Console to tie a request to a specific project in order to monitor traffic, enforce quotas, and handle billing.

- Server key
- Browser key
- Android key
- iOS key



NO CLIENT IDS FOUND.

specific data with you

e

if

## Create a server key and configure allowed IPs

**This key should be kept secret on your server.**

Every API request is generated by software running on a machine that you control. Per-user limits will be enforced using the address found in each request's `userIp` parameter, (if specified). If the `userIp` parameter is missing, your machine's IP address will be used instead. [Learn more](#)

### Accept requests from these server IP addresses (Optional)

One IP address or subnet per line. Example: 192.168.0.1, 172.16.0.0/12, 2001:db8::1 or 2001:db8::/64

Or if you leave this blank, requests will be accepted from any address. Be sure to add IP addresses before using this key in production.

Create

Cancel

Bize verdiği APİ key'i not alıyoruz.

Overview

Permissions

APIs & auth

APIs

Credentials

Consent screen

Push

Monitoring

Source Code

Deploy & Manage

Compute

Networking

Storage

Big Data

### OAuth

No client IDs found.

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private.

[Learn more](#)

[Create new Client ID](#)


### Public API access

Use of this key does not require any user action or consent, does not grant access to any account information, and is not used for authorization.

[Learn more](#)

[Create new Key](#)

### Key for server applications

API key	AlzaSyAyFGFDIJGItiMc1Ha5CsUdXC5JK6oRfS4	
IPs	Any IP allowed	
Activation date	Jul 7, 2015, 5:17:00 PM	
Activated by	umitkose@gmail.com (you)	

[Edit allowed IPs](#) [Regenerate key](#) [Delete](#)

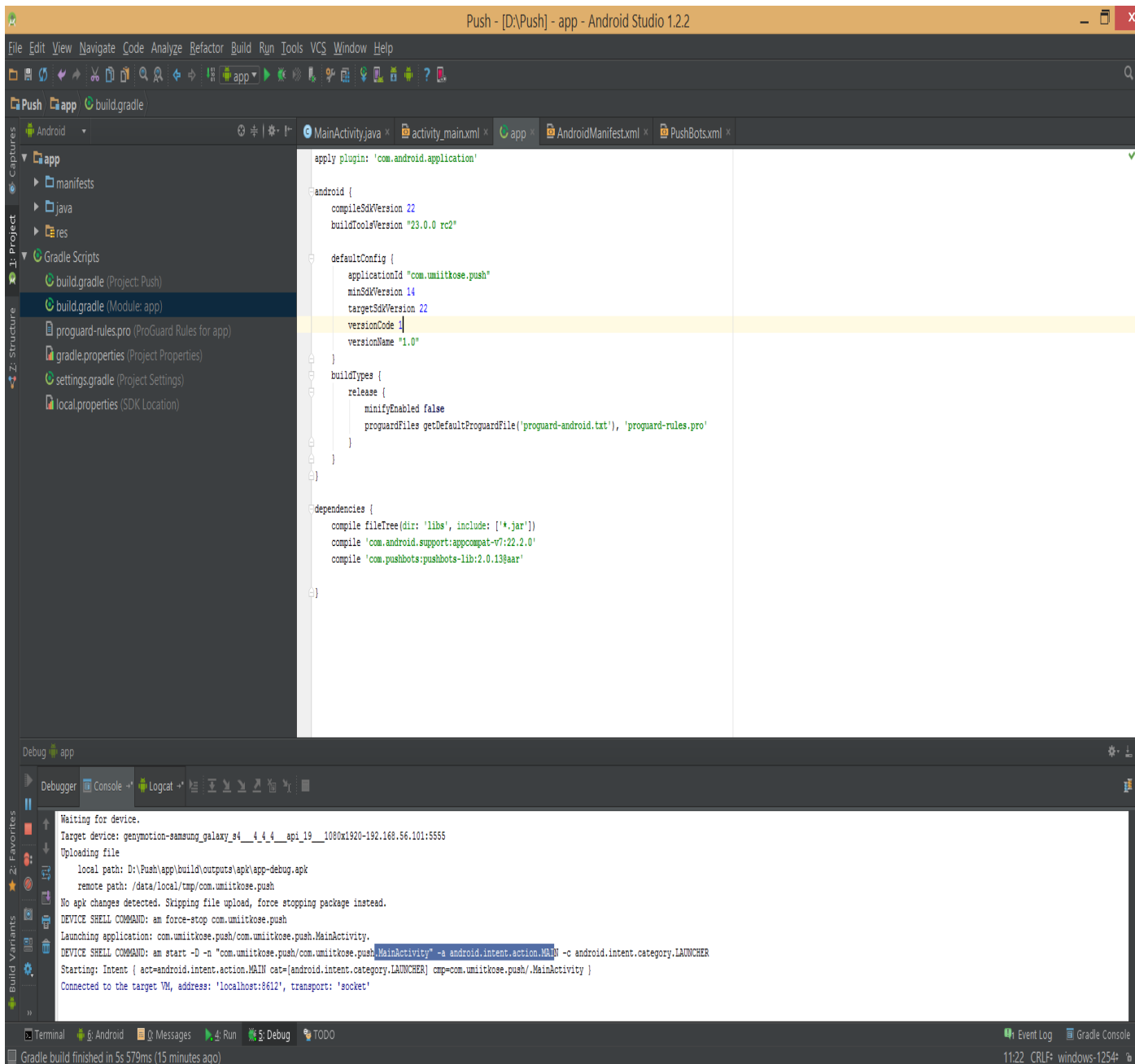
Şimdi elimizde api key ve üstte kaydettiğimiz project Number'ımız var.

Yeni Android Projesi oluşturuyoruz.

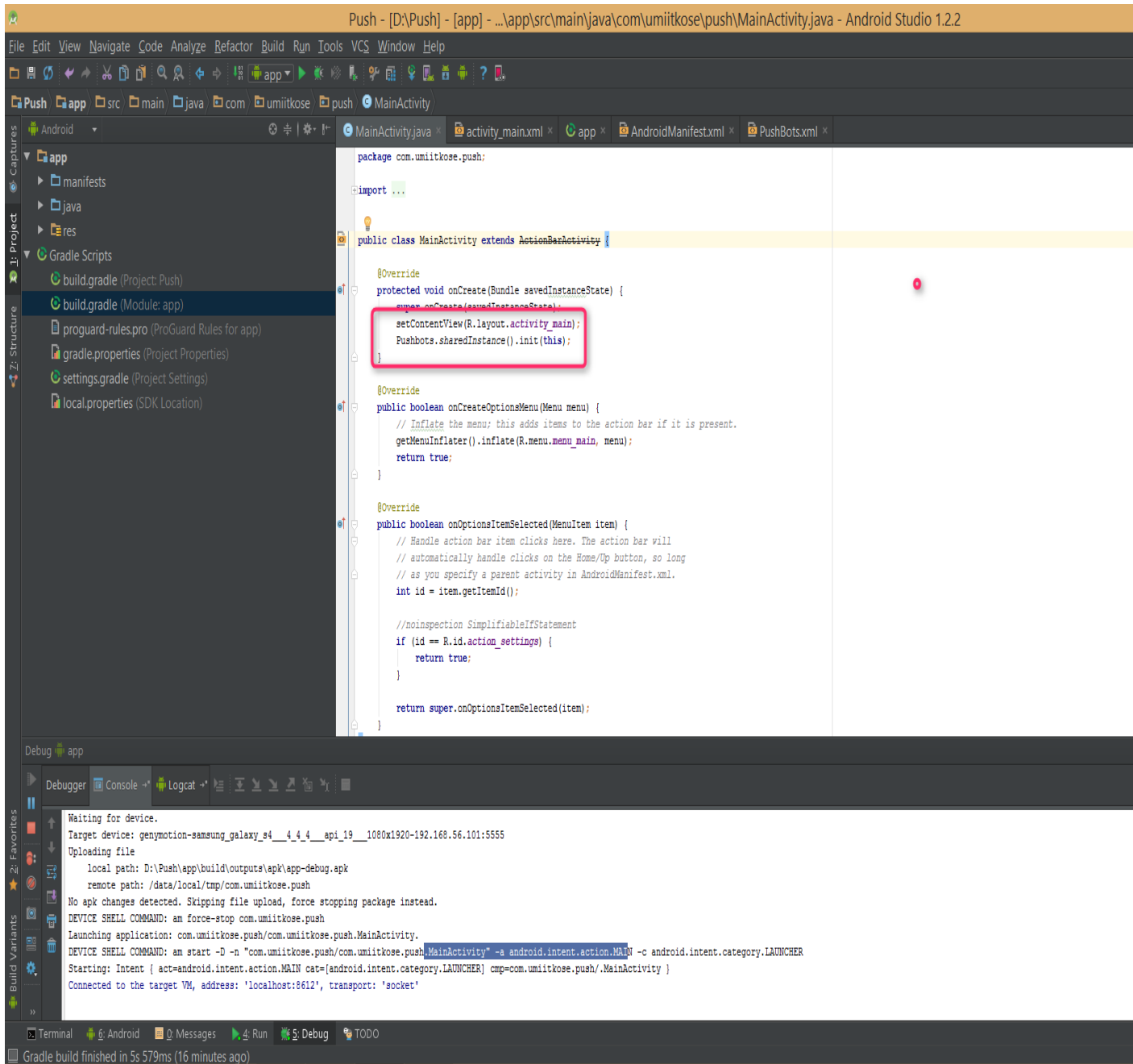
build.app'yi açarak FileTree altına

[compile](#) 'com.pushbots:pushbots-lib:2.0.13@aar'

yazıyoruz. Sync ediyoruz.



MainActivity.java da ise Pushbots.sharedInstance().init(this); ibaresini ekliyoruz.



AndroidManifest'iniz bu şekil olacak. Kodları :

Add the following permissions and change com.example.sampleapp with your app identifier. project isimlerini com.example.sampleapp'yı kendi proje isminizle değiştirin.

```
<!-- GCM connects to Google Services. -->
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<!-- GCM requires a Google account. -->
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

```
<permission
android:name="<spanclass="ddd">com.example.sampleapp.permission.C2D_MESSAGE"
android:protectionLevel="signature" />
<uses-permission android:name="com.example.sampleapp.permission.C2D_MESSAGE" />
<!-- This app has permission to register and receive data message. -->
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
```

The following intent-filter in your main activity: activity main kısmında intent filter ekleyin

```
<intent-filter>
  <action android:name="com.example.sampleapp.MESSAGE" />
  <category android:name="android.intent.category.DEFAULT" />
</intent-filter>
```

Add the following activity, receiver and service before the end of your application tag, and in all of the above code change com.example.sampleapp with your app identifier. (Application taglarından sonunda receiver ve service'leri ekliyeceksiniz. Ve kendi project name'i com.example.sampleapp'yi değiştirmeyi unutmayın)

```
<receiver
  android:name="com.pushbots.google.gcm.GCMBroadcastReceiver"
  android:permission="com.google.android.c2dm.permission.SEND" >
  <intent-filter>
    <!-- Receives the actual messages. -->
    <action android:name="com.google.android.c2dm.intent.RECEIVE" />
    <!-- Receives the registration id. -->
    <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
    <category android:name="com.example.sampleapp" />
  </intent-filter>
</receiver>
<receiver android:name="com.pushbots.push.DefaultPushHandler" />
<service android:name="com.pushbots.push.GCMIntentService" />
```

```
MainActivity.java × activity_main.xml × app × AndroidManifest.xml × PushBots.xml ×
↑ ↓ + - 🔍 Match Case Regex Words
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.umiitkose.push" >
    <!-- GCM connects to Google Services. -->
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <!-- GCM requires a Google account. -->
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <permission android:name="com.umiitkose.push.permission.C2D_MESSAGE" android:protectionLevel="signature" />
    <uses-permission android:name="com.umiitkose.push.permission.C2D_MESSAGE" />
    <!-- This app has permission to register and receive data message. -->
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Push"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="Push" >
            <intent-filter>
                <action android:name="com.umiitkose.push.MESSAGE" />
                <category android:name="android.intent.category.DEFAULT" />
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <receiver
            android:name="com.pushbots.google.gcm.GCMBroadcastReceiver"
            android:permission="com.google.android.c2dm.permission.SEND" >
            <intent-filter>
                <!-- Receives the actual messages. -->
                <action android:name="com.google.android.c2dm.intent.RECEIVE" />
                <!-- Receives the registration id. -->
                <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
                <category android:name="com.umiitkose.push" />
            </intent-filter>
        </receiver>
        <receiver android:name="com.pushbots.push.DefaultPushHandler" />
        <service android:name="com.pushbots.push.GCMIntentService" />
    </application>
</manifest>
```

Res Values'te yeni bir xml oluşturun. Pushbots.xml'de Kodlarımız. Burada pb\_senderid : Proje Number pb\_appid ise pushbots'a kaydedicez şimdi uygulamayı oradaki ID.

```
<?xml version="1.0" encoding="utf-8"?> <resources> <!-- Pushbots Application ID -  
-> <string name="pb_appid">548ef5901doab1</string> <!-- GCM Sender ID --  
> <string name="pb_senderid">48849973</string> <!-- Pushbots Log Level log Tag  
"PB2" --> <string name="pb_logLevel">DEBUG</string> </resources>
```

Push - [D:\Push] - [app] - ...app\src\main\res\values\PushBots.xml - Android Studio 1.2.2

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Push app src main res values PushBots.xml

Android MainActivity.java activity\_main.xml app AndroidManifest.xml PushBots.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <!-- Pushbots Application ID -->  
  <string name="pb_appid">548ef5901doab1</string>  
  <!-- GCM Sender ID -->  
  <string name="pb_senderid">48849973</string>  
  <!-- Pushbots Log Level log Tag "PB2" -->  
  <string name="pb_logLevel">DEBUG</string>  
</resources>
```

Debug app

Debugger Console Logcat

Waiting for device.  
Target device: genymotion-samsung\_galaxy\_s4\_\_4\_4\_\_api\_19\_\_1080x1920-192.168.56.101:5555  
Uploading file  
 local path: D:\Push\app\build\outputs\apk\app-debug.apk  
 remote path: /data/local/tmp/com.umitkose.push  
No apk changes detected. Skipping file upload, force stopping package instead.  
DEVICE SHELL COMMAND: am force-stop com.umitkose.push  
Launching application: com.umitkose.push/com.umitkose.push.MainActivity.  
DEVICE SHELL COMMAND: am start -D -n "com.umitkose.push/com.umitkose.push.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER  
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.umitkose.push/.MainActivity }  
Connected to the target VM, address: 'localhost:8612', transport: 'socket'

Terminal Android Messages Run Debug TODO

Gradle build finished in 5s 579ms (17 minutes ago)



PushBots'a kaydolalım. Bildiğiniz Üyelik işlemi. Dashboard sistemi aşağıda. Yeni bir proje oluşturcaz.

The screenshot displays the PushBots dashboard interface. At the top, there is a navigation bar with the PushBots logo, a home icon, a search bar labeled 'Search Applications', and the user's name 'umiitkose' with a profile icon and a settings gear. Below the navigation bar, the dashboard is divided into several sections. On the left, a dark sidebar contains a 'Welcome umiitkose' message, a 'Free account' status, and an 'Upgrade now' button. Below this, there are navigation links for 'Dashboard', 'Account', and 'Docs'. The main content area features four summary cards: 'API REQUESTS \*' (0% of 1,000,000), 'NOTIFICATIONS SENT \*' (0% of 1,500,000), 'TOTAL DEVICES' (1 of unlimited), and 'TOTAL APPS' (1 of unlimited). A modal dialog titled 'Create App' is open in the center, asking 'What will you call it?'. To the right of the modal, there is a card for an application named 'push', showing '1 devices' and '0 sent', with navigation options for 'PUSH', 'DEVICES', 'SETTINGS', 'ANALYTICS', and 'SANDBOX'. A small note at the top right states '\* API requests and number of notifications statistics start from Jul 05, 2015'.

Platform'a Android GCM API KEY'E yukarda google'a kaydolurkenki API keyimizi ekliyoruz. Ve keys'e geliyoruz.

The screenshot shows the PushBots dashboard interface. On the left is a dark sidebar with the PushBots logo, a home icon, and a search bar. The main content area is titled "Dashboard - push settings" and has a navigation bar with tabs for "General", "Keys", "Platforms", "Collaborators", and "Twitter". The "Keys" tab is active. Below the navigation bar, there are three sections: "Apple Push Certificate", "Android GCM API Key", and "Chrome API Key". The "Android GCM API Key" section is highlighted with a red box and contains the text "AlzaSyAdCxINCG9qTcG3B0GZHc65RJYRE\_gqT8s". The "Chrome API Key" section contains the text "Enter Chrome API Key and press Enter".

PushBots

Search Applications

umiitkose

Welcome umiitkose Free account Upgrade now

Dashboard

Account

Docs

Dashboard - push settings

General Keys Platforms Collaborators Twitter PRO FEATURE

Apple Push Certificate

To develop and deploy an app, you must get SSL certificates from Apple Member Center. Each certificate is limited to a single app, identified by its bundle ID. Each certificate is also limited to one of two development environments. Click here for a detailed step-by-step guide.

No certificate Uploaded yet Upload P12 Certificate

Android GCM API Key

To use GCM, you need to obtain a long-lived token (simple API key). Click here for a detailed step-by-step guide.

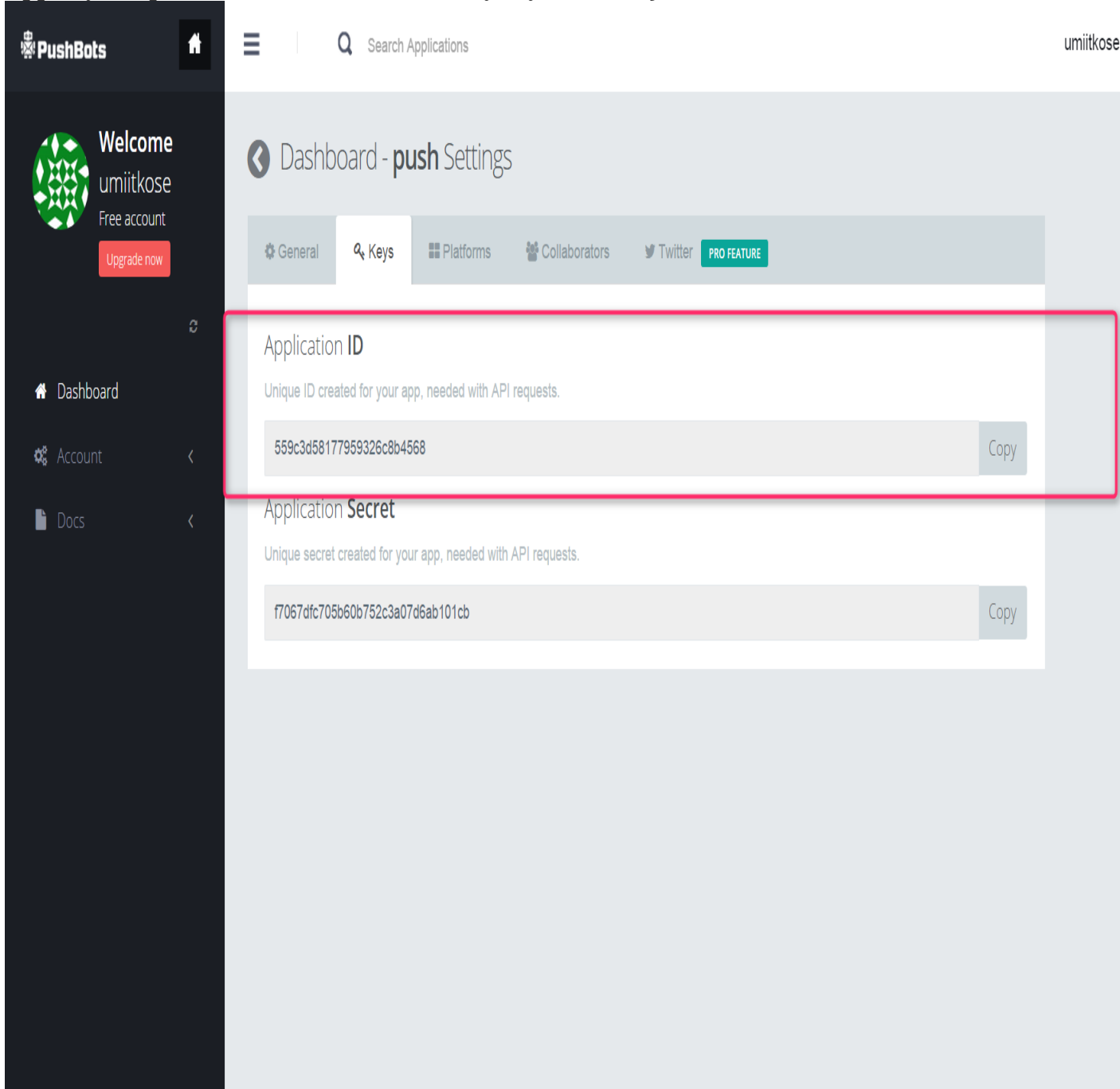
AlzaSyAdCxINCG9qTcG3B0GZHc65RJYRE\_gqT8s

Chrome API Key

To use Chrome Push notifications, you need to obtain a long-lived token (simple API key). Click here for a detailed step-by-step guide.

Enter Chrome API Key and press Enter

AppID'yi ise pushbots.xml'de senderID'Ye yazıyoruz. Ve işlem bu kadar



The screenshot shows the PushBots dashboard for a user named 'umiitkose'. The dashboard is titled 'Dashboard - push Settings'. The 'Keys' tab is selected, showing the 'Application ID' and 'Application Secret'. The Application ID is 559c3d58177959326c8b4568 and the Application Secret is 77067dfc705b60b752c3a07d6ab101cb. Both fields have a 'Copy' button next to them. The Application ID field is highlighted with a red box.

PushBots

Welcome umiitkose  
Free account  
Upgrade now

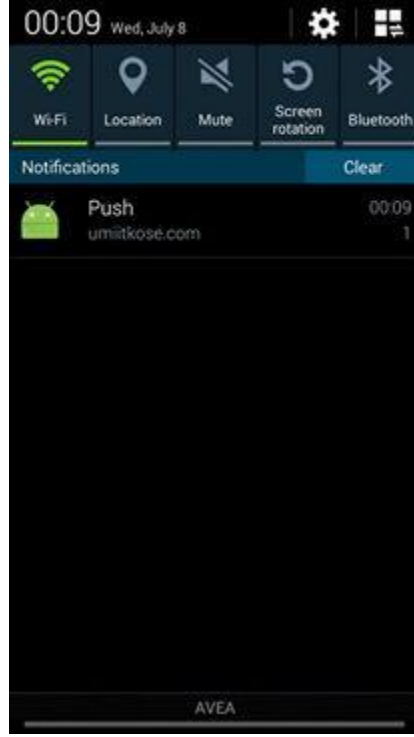
Dashboard - push Settings

General Keys Platforms Collaborators Twitter PRO FEATURE

Application ID  
Unique ID created for your app, needed with API requests.  
559c3d58177959326c8b4568 Copy

Application Secret  
Unique secret created for your app, needed with API requests.  
77067dfc705b60b752c3a07d6ab101cb Copy

Uygulamayı telefonunuz yükleyip Send a push'ta yeni bir mesaj yolladıktan sonraki hali..



Detaylı Bilgi İçin [Tıklayın..](#)

---

# Android Studio Programlama Bölüm 30

---

## Android Studio İnternet Varlığı

Herkese merhaba internetin olup olmadığını android üzerinden nasıl kontrol edebiliriz onu açıklayacağım. Kod olarak

```
public boolean İnternetKontrol() {
    ConnectivityManager manager = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
    if (manager.getActiveNetworkInfo() != null
        && manager.getActiveNetworkInfo().isAvailable()
        && manager.getActiveNetworkInfo().isConnected()) {
        return true;
    } else {
        return false;
    }
}
```

Yazacağız.

```
if(İnternetKontrol()){
```

```
// Bağlantı var
```

```
}else{
```

```
//Bağlantı yok
```

```
}
```

Şeklinde kullanacağız. Yok olan yerde toast mesaj gösterebilir ya da uygulamadan çıkabilirsiniz. Var olan yerde de bir web view'e yönlendirebilirsiniz.

Manifest dosyasında ise bu izni kullanacağız.

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

---

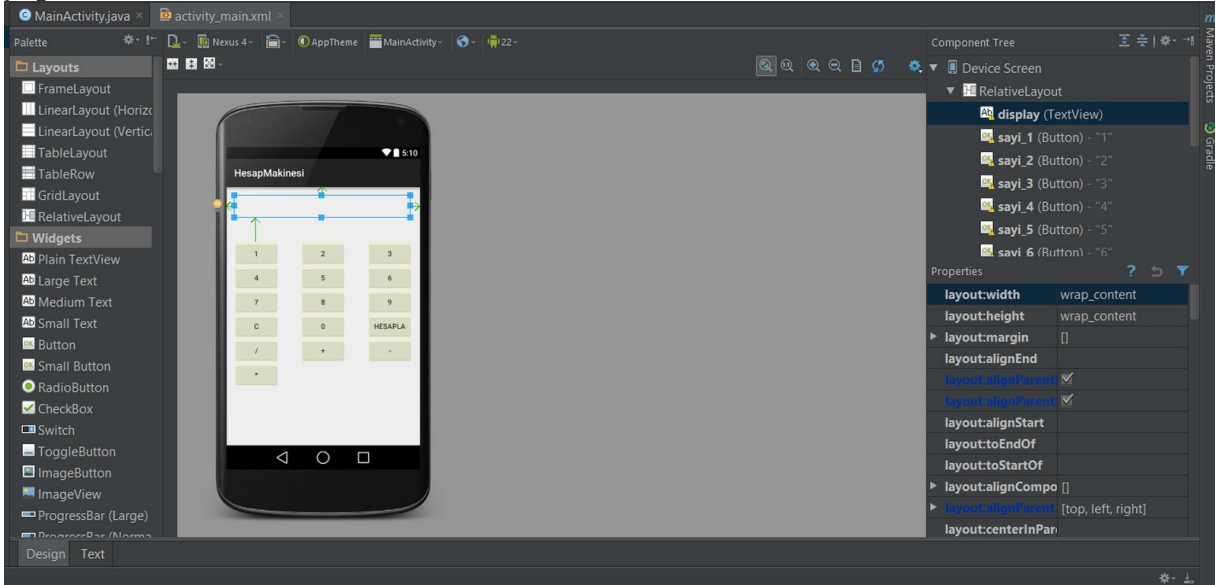
# Android Studio Uygulamalar 1 – Hesap Makinesi

## Hesap Makinesi

Android'te 1. uygulama olarak hesap makinesi yapımımlı anlatacađım. Basitçe 0-9'a kadar sayılar olucak. Onları display'de göstereceđiz. Topla, Hesapla, Sil, Çıkar, Böl, Çarp işlemleri olucak. Eđer kullanıcı sil'e basmışsa direk displayi boşaltıcaz. İlk sayıyı girdik. Herhangi bir işleme bastık. Onu hafızaya alıp yeni bastığımız işlemi hesapla ile hesaplattırcaz. Basit bir hesap makinesi. Hadi işe koyulalım.

Projenin kod hali isteyenler için.. [HesapMakinesi](#)

Öncelikle basit bir tasarım yapıyoruz. Ben üstüne düşmeden aşğıdaki şekil yaptım.



Sonrasında tüm butonlarımı aktif ettim.

```
MainActivity.java x activity_main.xml x
public class MainActivity extends ActionBarActivity {
    Button sayi_0,sayi_1,sayi_2,sayi_3,sayi_4,sayi_5,sayi_6,sayi_7,sayi_8,sayi_9,sayi_topla,sayi_cikar,sayi_carp,sayi_bol,sayi_sil,hesapla;
    TextView display;
    private float ilkdeger=0;
    private double sonuc=0;
    private String islem;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        hesapla=(Button)findViewById(R.id.hesapla);
        sayi_0=(Button)findViewById(R.id.sayi_0);
        sayi_1=(Button)findViewById(R.id.sayi_1);
        sayi_2=(Button)findViewById(R.id.sayi_2);
        sayi_3=(Button)findViewById(R.id.sayi_3);
        sayi_4=(Button)findViewById(R.id.sayi_4);
        sayi_5=(Button)findViewById(R.id.sayi_5);
        sayi_6=(Button)findViewById(R.id.sayi_6);
        sayi_7=(Button)findViewById(R.id.sayi_7);
        sayi_8=(Button)findViewById(R.id.sayi_8);
        sayi_9=(Button)findViewById(R.id.sayi_9);
        sayi_topla=(Button)findViewById(R.id.sayi_topla);
        sayi_cikar=(Button)findViewById(R.id.sayi_cikar);
        sayi_carp=(Button)findViewById(R.id.sayi_carp);
        sayi_bol=(Button)findViewById(R.id.sayi_bol);
        sayi_sil=(Button)findViewById(R.id.sayi_sil);
        display=(TextView)findViewById(R.id.display);
        sayi_0.setOnClickListener((view) -> { display.setText(display.getText()+"0"); });

        sayi_1.setOnClickListener((view) -> {
            display.setText(display.getText()+"1");
        });

        sayi_2.setOnClickListener((view) -> {
            display.setText(display.getText()+"2");
        });
    }
}
```

```
MainActivity.java x activity_main.xml x
        sayi_1.setOnClickListener((view) -> {
            display.setText(display.getText()+"1");
        });

        sayi_2.setOnClickListener((view) -> {
            display.setText(display.getText()+"2");
        });

        sayi_3.setOnClickListener((view) -> {
            display.setText(display.getText()+"3");
        });

        sayi_4.setOnClickListener((view) -> {
            display.setText(display.getText()+"4");
        });

        sayi_5.setOnClickListener((view) -> {
            display.setText(display.getText()+"5");
        });

        sayi_6.setOnClickListener((view) -> {
            display.setText(display.getText()+"6");
        });

        sayi_7.setOnClickListener((view) -> {
            display.setText(display.getText()+"7");
        });

        sayi_8.setOnClickListener((view) -> {
            display.setText(display.getText()+"8");
        });

        sayi_9.setOnClickListener((view) -> {
            display.setText(display.getText()+"9");
        });

        try{

```

"display.setText(display.getText()."sayi")" komutu ile bastığım tuşu displayde gösterdim.

Try-catch bloğu ile de eğer eğer toplama tuşuna basılmışsa hesaplada ilk değeri alıp Toplama stringine eşitleyip işlemi yaptırđım. Diđer butonlar içinde işlem aynı. Eski değeri ilkdeğere eşitleyerek kaydediyorum.

```
MainActivity.java x activity_main.xml x

try{
    sayi_topla.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            ilkdeger=0;
            sonuc=0;
            if(!display.getText().toString().equals("")){
                ilkdeger=Float.parseFloat(display.getText().toString());
                display.setText("");
                islem="Toplama";
            }
        }
    });
}
catch (Exception c){}

try{
    sayi_cikar.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            ilkdeger=0;
            sonuc=0;
            if(!display.getText().toString().equals("")){
                ilkdeger=Float.parseFloat(display.getText().toString());
                display.setText("");
                islem="Cikarma";
            }
        }
    });
}
}
```

Hesap'la da ise Eđer toplama işlemine bastıysam toplama işlemi yapıcak. Yani yukardan aldığım ilk değeri ile şimdi ki değeri toplayacak. Diđer işlemlerde aynı mantıkta.



```
Example / info / hesapmakinesi / MainActivity
MainActivity.java x activity_main.xml x
sayi_bol.setOnClickListener((view) -> {
    ilkdeger=0;
    sonuc=0;
    if(!display.getText().toString().equals("")){
        ilkdeger=Float.parseFloat(display.getText().toString());
        display.setText("");
        islem="Bolme";
    }
});
}
catch (Exception c){}
hesapla.setOnClickListener((view) -> {
    if(!display.getText().toString().equals("") && !String.valueOf(ilkdeger).equals("0")){
        if(islem.equals("Toplama")){
            sonuc=ilkdeger + Float.parseFloat(display.getText().toString());
        }
        else if(islem.equals("Cikarma")){
            sonuc=ilkdeger - Float.parseFloat(display.getText().toString());
        }
        else if(islem.equals("Carpma")){
            sonuc=ilkdeger * Float.parseFloat(display.getText().toString());
        }
        else if(islem.equals("Bolme")){
            sonuc=ilkdeger / Float.parseFloat(display.getText().toString());
        }
        display.setText(String.valueOf(sonuc));
        ilkdeger=0;
        sonuc=0;
    }
}
```

```
MainActivity.java × activity_main.xml ×
else if (islem.equals("Cikarma")){
    sonuc=ilkdeger - Float.parseFloat(display.getText().toString());
}
else if (islem.equals("Carpma")){
    sonuc=ilkdeger * Float.parseFloat(display.getText().toString());
}
else if (islem.equals("Bolme")){
    sonuc=ilkdeger / Float.parseFloat(display.getText().toString());
}
display.setText(String.valueOf(sonuc));
ilkdeger=0;
sonuc=0;
}
else{
    display.setText("Deger Girmediniz..");
}
if (R.id.sayi_sil == view.getId()) {
    ilkdeger=0;
    sonuc=0;
    display.setText("");
}
});
}
}
```

Projenin kod hali isteyenler için → [HesapMakinesi](#)

---

---

# Android Studio Uygulama 2

---

## Android Studio İnternet Tarayıcısı

Bu yazımızda da yeni bir internet tarayıcısı yapacağız. Öncelikle webView hakkında bilginiz yoksa basit bir konu anlatımı için [tıklayın](#).

Şimdi basit bir tarayıcı yapacağız. Öncelikle tasarımını halledelim.

Activity\_main Kodları:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#E0E0D1" >
```

```
<LinearLayout
    android:id="@+id/iki"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
```

```
<TextView
    android:id="@+id/textViewURL"
    android:layout_width="50dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:paddingLeft="13dp"
    android:textColor="#000"
    android:text="URL" />
```

```
<EditText
    android:id="@+id/URL"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:ems="10" >
</EditText>
```

```
<Button
    android:id="@+id/Git"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Git" />
</LinearLayout>
```

```
<WebView
    android:id="@+id/webV"
    android:layout_width="fill_parent"
    android:layout_height="850dp"
    android:layout_below="@id/iki" />
```

```
<LinearLayout
    android:id="@+id/bir"
    android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:weightSum="3"
android:layout_below="@+id/webV"
>

<Button
    android:id="@+id/Geri"
    android:layout_width="odp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Geri" />

<Button
    android:id="@+id/Anasayfa"
    android:layout_width="odp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Anasayfa" />

<Button
    android:id="@+id/Ileri"
    android:layout_width="odp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Ileri"
/>
</LinearLayout>
</RelativeLayout>
```

Şimdi Java kodlarına başlayacağız :  
Resimleri:

```

package com.example.myapplication;

import ...

public class MainActivity extends ActionBarActivity {
    Button Geri,Ileri, Anasayfa, Git;
    WebView webV;
    EditText URL;
    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        webV = (WebView) findViewById(R.id.webV);
        webV.getSettings().setJavaScriptEnabled(true);
        webV.setWebViewClient(new MyWebViewClient());
        webV.loadUrl("http://www.google.com.tr");
        URL = (EditText) findViewById(R.id.URL);

        Geri = (Button) findViewById(R.id.Geri);
        Ileri = (Button) findViewById(R.id.Ileri);
        Anasayfa = (Button) findViewById(R.id.Anasayfa);
        Git = (Button) findViewById(R.id.Git);

        Git.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if (URL.getText().toString().length() > 10) {
                    Toast.makeText(
                        getApplicationContext(),
                        URL.getText().toString()
                            + " adresi yükleniyor "
                            + URL.getText().toString()
                            + " adresi yükleniyor...",
                        Toast.LENGTH_SHORT).show();

                    if (InternetKontrol())
                        webV.loadUrl("http://" + URL.getText().toString());
                    else
                        BaglantiHatasiVer();
                } else {
                    Toast.makeText(getApplicationContext(), "Geçersiz Adres",
                        Toast.LENGTH_SHORT).show();
                }
            }
        });

        Anasayfa.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if (InternetKontrol())
                    webV.loadUrl("http://www.google.com.tr");
                else
                    BaglantiHatasiVer();
            }
        });

        Geri.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if (webV.canGoBack()) {
                    webV.goBack();
                }
            }
        });
    }
}

```

```

    },

    Ileri.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View arg0) {
            if (webV.canGoForward()) {
                webV.goForward();
            }
        }
    });
}

private class MyWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
}

public void BaglantiHatasiVer() {
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(
        MainActivity.this);
    alertDialogBuilder.setTitle("Sunucu Hatasi");
    alertDialogBuilder.setMessage(
        "internet bağlantınızı kontrol edip tekrar deneyin")
        .setCancelable(true);
    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}

public boolean InternetKontrol() {
    ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    if (manager.getActiveNetworkInfo() != null
        && manager.getActiveNetworkInfo().isAvailable()
        && manager.getActiveNetworkInfo().isConnected()) {
        alertDialog.show();
    }
}

public boolean InternetKontrol() {
    ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    if (manager.getActiveNetworkInfo() != null
        && manager.getActiveNetworkInfo().isAvailable()
        && manager.getActiveNetworkInfo().isConnected()) {
        return true;
    } else {
        return false;
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

```

Kodlar:

```
package com.example.myapplication;
```

```
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.webkit.WebChromeClient;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

```
public class MainActivity extends ActionBarActivity {
    Button Geri,Ileri, Anasayfa, Git;
    WebView webV;
    EditText URL;
    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);
        webV = (WebView) findViewById(R.id.webV);
        webV.getSettings().setJavaScriptEnabled(true);
        webV.setWebViewClient(new MyWebViewClient());
        webV.loadUrl("http://www.google.com.tr");
        URL = (EditText) findViewById(R.id.URL);

        Geri = (Button) findViewById(R.id.Geri);
        Ileri = (Button) findViewById(R.id.Ileri);
        Anasayfa = (Button) findViewById(R.id.Anasayfa);
        Git = (Button) findViewById(R.id.Git);

        Git.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View arg0) {
                if (URL.getText().toString().length() > 10) {
                    Toast.makeText(
                        getApplicationContext(),
                        URL.getText().toString()
                            + " adresi yükleniyor...",
                        Toast.LENGTH_SHORT).show();

                    if (InternetKontrol())
                        webV.loadUrl("http://" + URL.getText().toString());
                    else
                        BaglantiHatasiVer();
                } else {
                    Toast.makeText(getApplicationContext(), "Geçersiz Adres",
                        Toast.LENGTH_SHORT).show();
                }
            }
        });

        Anasayfa.setOnClickListener(new View.OnClickListener() {
```

```

    @Override
    public void onClick(View argo) {
        if (InternetKontrol())
            webV.loadUrl("http://www.google.com.tr");
        else
            BaglantiHatasiVer();
    }
});

Geri.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View argo) {
        if (webV.canGoBack()) {
            webV.goBack();
        }
    }
});

Ileri.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View argo) {
        if (webV.canGoForward()) {
            webV.goForward();
        }
    }
});

}
private class MyWebViewClient extends WebViewClient {
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, String url) {
        view.loadUrl(url);
        return true;
    }
}

public void BaglantiHatasiVer() {
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(
        MainActivity.this);
    alertDialogBuilder.setTitle("Sunucu Hatasi");
    alertDialogBuilder.setMessage(
        "internet bağlantınızı kontrol edip tekrar deneyin")
        .setCancelable(true);
    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}

public boolean InternetKontrol() {
    ConnectivityManager manager = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
    if (manager.getActiveNetworkInfo() != null
        && manager.getActiveNetworkInfo().isAvailable()
        && manager.getActiveNetworkInfo().isConnected()) {
        return true;
    } else {
        return false;
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}
}

```



```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}
}
```

Ve son olarak Manifest dosyasına da 2 tane izin ekliyoruz.

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

Uygulamanın çalıştırılması..



Google.com.tr bu dili kullansın: [English](#)

Bilinmiyor

## Android Studio Uygulama 3

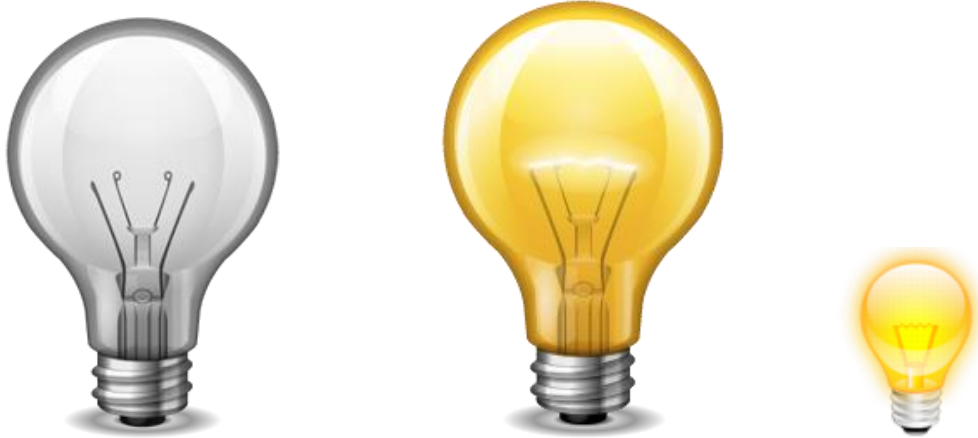
### Android Studio Flashlight Uygulaması

Android Studio'da 3. uygulamayı geliştireceğiz. 3 uygulamada bir image button atıp onu toggle button şeklinde kullanacağız. Toggle button'u açma kapama tuşu olarak düşünelim. Tuşa bastığımızda açılacak, Tuşa bastığımızda kapatılacak. Button'a basınca biz kamera'dan flash'a bağlanıp açacağız, diğer bir tıklamada flash'ı kapatacağız. Hadi başlayalım. Yeni bir proje açıyoruz.

İlk iş olarak 2 tane izin alarak başlayalım. Android manifest Dosyasına aşağıdaki kodları ekleyin.

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

İzinleri aldıktan sonra uygulamamızın icon'u değiştirelim. Güzel bir görüntü elde edelim. Mipmap içine küçük iconu atın. Diğer lambalarda kaydedin. Bunları uygulamada mesela kapalıyken gri olanını, açıkken sarı olanını göstereceğiz.



Activity\_main dosyamızın kodları

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="#cae5e0">

    <ImageButton
        android:id="@+id/btnSwitch"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dip"
        android:src="@drawable/btn_switch_on"
        android:background="@null"
```

```
        android:contentDescription="@null"
        />
</RelativeLayout>
```

### Şimdide MainActivity.java dosyamızın kodları:

```
package com.umiitkose.myapplication;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.pm.PackageManager;
import android.hardware.Camera;
import android.hardware.Camera.Parameters;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnCompletionListener;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;

import com.umiitkose.myapplication.R;

public class MainActivity extends Activity {

    ImageButton btnSwitch;
    private Camera camera;
    private boolean flashAcik;
    private boolean flashVarmi;
    Parameters params;
    MediaPlayer mp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnSwitch = (ImageButton) findViewById(R.id.btnSwitch); //ImageButton'u tanımladık
        //İlk işlem kameranın flash'ının olup olmadığı
        flashVarmi =
        getApplicationContext().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_FLASH);

        if (!flashVarmi) {
            //FlashYoksa alertDialog ile ekrana flash'ın olmadığını çıkardık
            AlertDialog alert = new AlertDialog.Builder(MainActivity.this).create();
            alert.setTitle("HATA");
```

```

        alert.setMessage("Bu aygıt Flash desteklemiyor");
        alert.setButton("Tamam", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                // Tamam'a basınca uygulamayı kapatıyoruz.
                finish();
            }
        });
        alert.show();
        return;
    }

    // camera'ya erişiyoruz
    cameraAc();

    //Flash'ın durumuna göre button'un resmini ayarlıyoruz
    toggleButtonImage();

    /*
    * Button'a her tıklandığında olacaklar. Flash açıksa kapat, Kapalıysa aç
    */
    btnSwitch.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            if (flashAcik) { // Flash Acik true dönücek. Eğer Flash Açıkça

                flashKapat();

            } else {

                // Açık değilse button'a basınca açıcak
                flashAc();

            }

        }

    });

}

/*
* Get the camera
*/
private void cameraAc() {
    if (camera == null) {

        camera = Camera.open();
        params = camera.getParameters();

    }

}

/*
* Flash'ı açma kısmı burada
*/
private void flashAc() {
    if (!flashAcik) {
        if (camera == null || params == null) {
            return;
        }
        params = camera.getParameters();
        params.setFlashMode(Parameters.FLASH_MODE_TORCH);
        camera.setParameters(params);
        camera.startPreview();
        flashAcik = true;

        // Flash Button'u Değişiyor
        toggleButtonImage();
    }
}

```

Kapatıcak

```

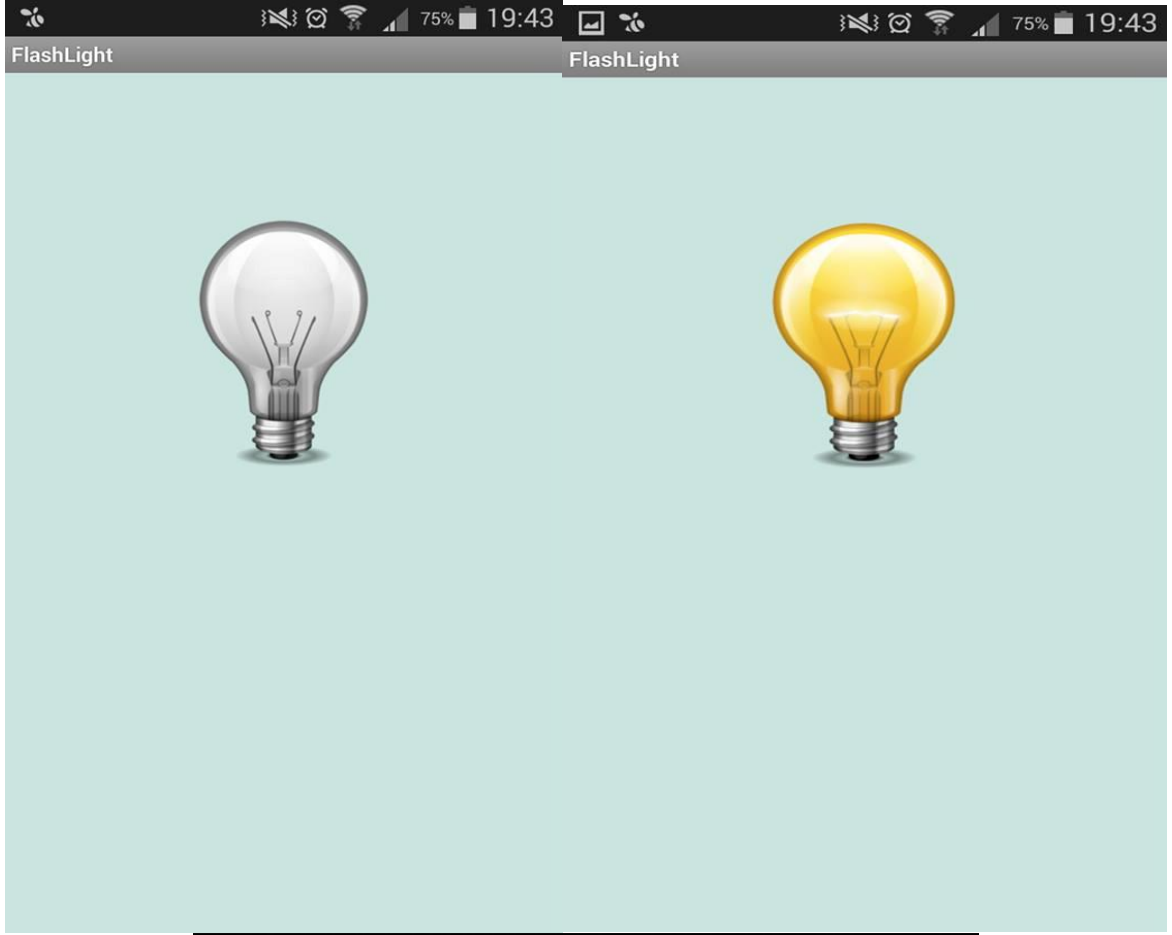
    }
}
/*
 * Flash'ı kapatma kısmı
 */
private void flashKapat() {
    if (flashAcik) {
        if (camera == null || params == null) {
            return;
        }
        params = camera.getParameters();
        params.setFlashMode(Parameters.FLASH_MODE_OFF);
        camera.setParameters(params);
        camera.stopPreview();
        flashAcik = false;

        // changing button/switch image
        toggleButtonImage();
    }
}

/*
 * Button'un deęiřtirmesi
 */
private void toggleButtonImage(){
    if(flashAcik){
        btnSwitch.setImageResource(R.drawable.flash_ac);
    }else{
        btnSwitch.setImageResource(R.drawable.flash_kapa);
    }
}
}}

```

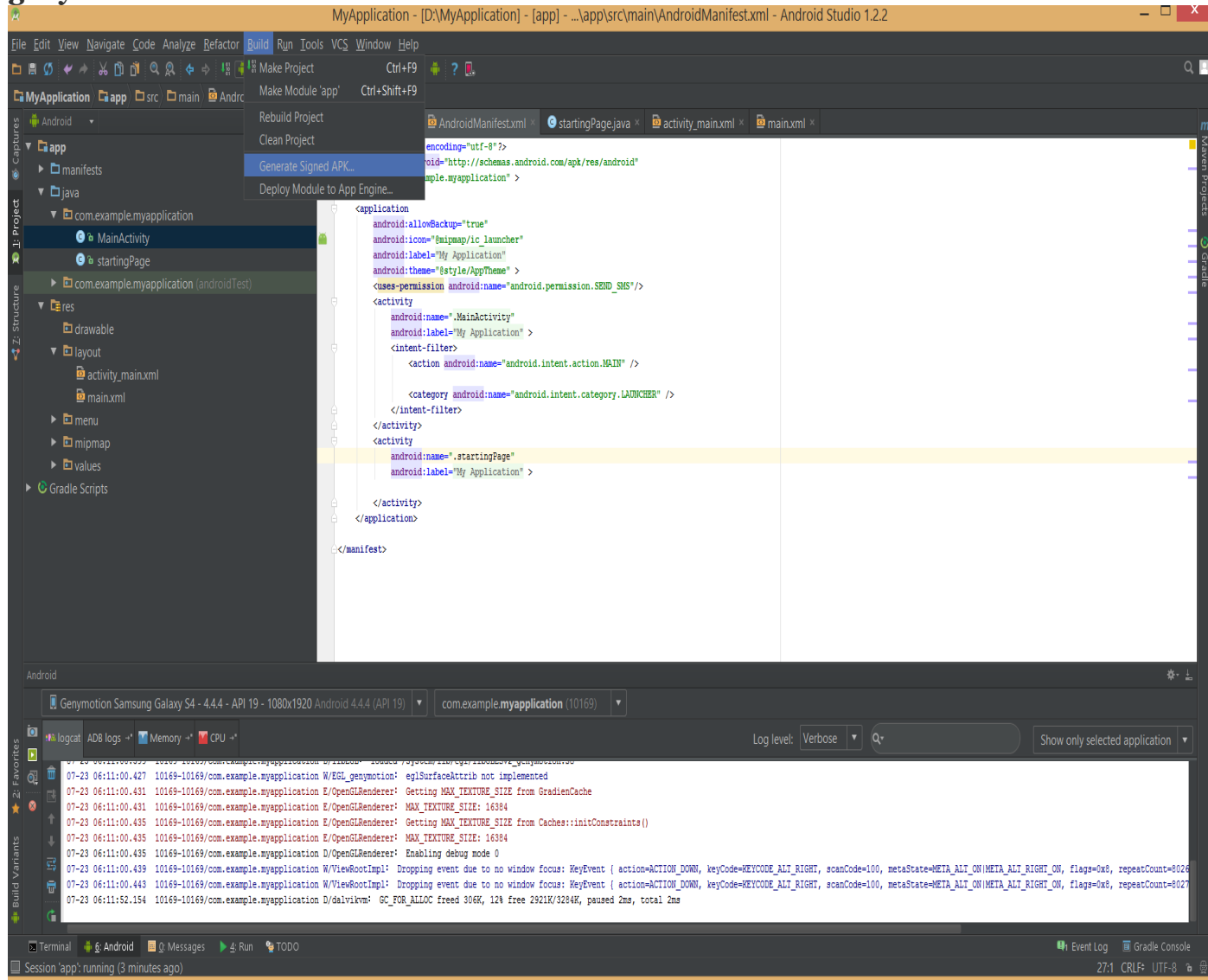
Uygulamayı telefonda deneyiniz. Yoksa Flash desteklemiyor'u alırsınız. Telefonda açıkken ve kapalıykenki görünümü.. [Apk dosyası oluşturarak deneyebilirsiniz.](#)



# Android Studio Apk Oluřturma

Android studio kullandıkça yenilikler farkettim. Mesela apk dosyası eclipse de önceden emülatörü çalıştırınca direk oluşuyordu. Fakat android studio da bunun olmadığını öğrendim. Şimdi bu bölümde de android studio'da nasıl apk dosyası oluşturulur onu öğreneceğiz.

## İşlem 1 : Android Studio ' da Build menüsünde Generate Signed APK..'ya giriyoruz.



The screenshot shows the Android Studio 1.2.2 interface. The 'Build' menu is open, and 'Generate Signed APK...' is selected. The main editor displays the AndroidManifest.xml file. The interface includes the Project view, Logcat, and Run/Debug tabs.

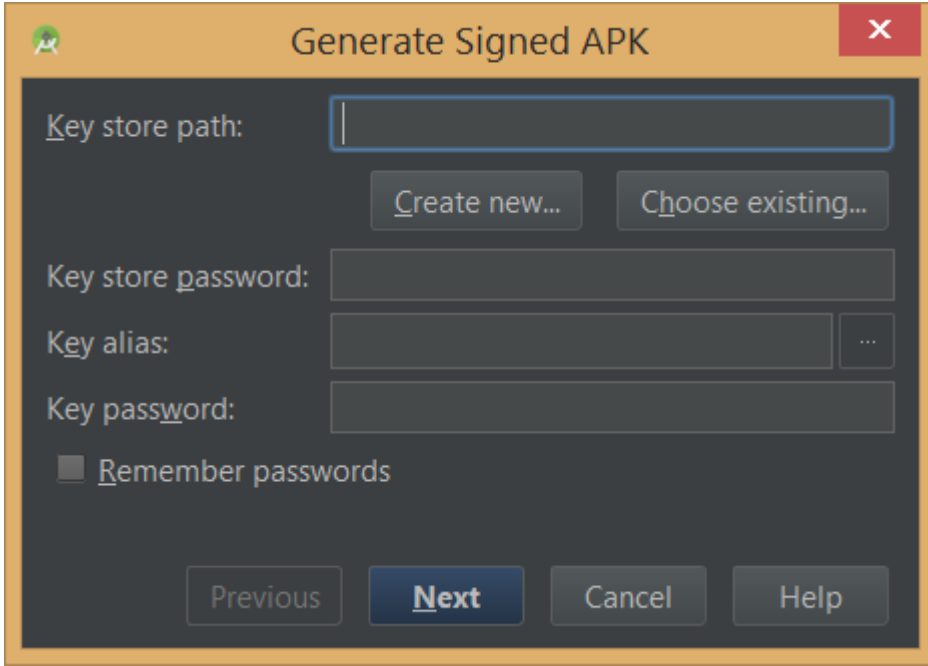
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication" >
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <uses-permission android:name="android.permission.SEND_SMS"/>
        <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>
        <activity
            android:name=".startingPage"
            android:label="@string/app_name" />
        </activity>
    </application>
</manifest>
```

The Logcat window shows the following logs:

```
07-23 06:11:00.427 10169-10169/com.example.myapplication W/EGL_genymotion: eglSurfaceAttrib not implemented
07-23 06:11:00.431 10169-10169/com.example.myapplication E/OpenGLESRenderer: Getting MAX_TEXTURE_SIZE from GradienCache
07-23 06:11:00.431 10169-10169/com.example.myapplication E/OpenGLESRenderer: MAX_TEXTURE_SIZE: 16384
07-23 06:11:00.435 10169-10169/com.example.myapplication E/OpenGLESRenderer: Getting MAX_TEXTURE_SIZE from Caches::initConstraints()
07-23 06:11:00.435 10169-10169/com.example.myapplication E/OpenGLESRenderer: MAX_TEXTURE_SIZE: 16384
07-23 06:11:00.435 10169-10169/com.example.myapplication D/OpenGLESRenderer: Enabling debug mode 0
07-23 06:11:00.439 10169-10169/com.example.myapplication W/ViewRootImpl: Dropping event due to no window focus: KeyEvent { action=ACTION_DOWN, keyCode=KEYCODE_ALT_RIGHT, scanCode=100, metaState=META_ALT_ON|META_ALT_RIGHT_ON, flags=0x8, repeatCount=8026}
07-23 06:11:00.443 10169-10169/com.example.myapplication W/ViewRootImpl: Dropping event due to no window focus: KeyEvent { action=ACTION_DOWN, keyCode=KEYCODE_ALT_RIGHT, scanCode=100, metaState=META_ALT_ON|META_ALT_RIGHT_ON, flags=0x8, repeatCount=8027}
07-23 06:11:52.154 10169-10169/com.example.myapplication D/dalvikvm: GC_FOR_ALLOC freed 306K, 128 free 2921K/3284K, paused 2ms, total 2ms
```



**İşlem 2 : Karşımızda Generate Signed APK çıkıyor. Create new'e tıklıyoruz. Bir anahtar yolu seçerek'ten apkmızı oluşturuyoruz.**



Generate Signed APK

Key store path:

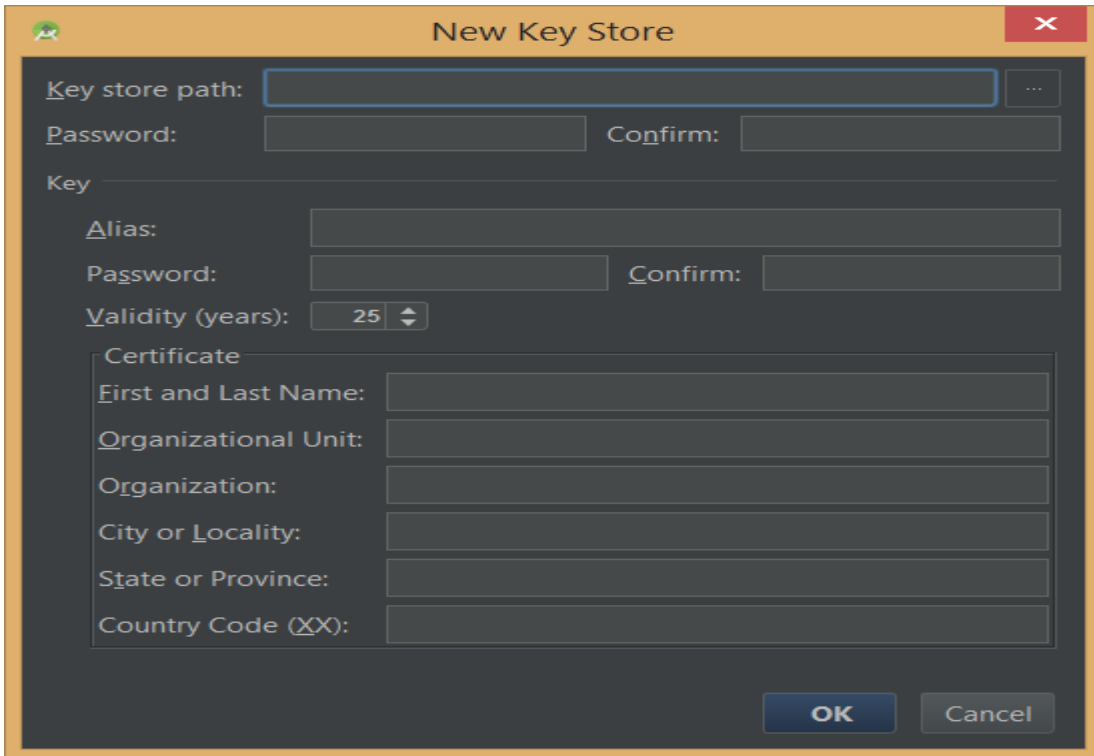
Key store password:

Key alias:  ...

Key password:

Remember passwords

**İşlem 3 : 3 noktaya tıklayarak bir yol seçiyoruz**



New Key Store

Key store path:  ...

Password:  Confirm:

Key

Alias:

Password:  Confirm:

Validity (years): 25

Certificate

First and Last Name:

Organizational Unit:

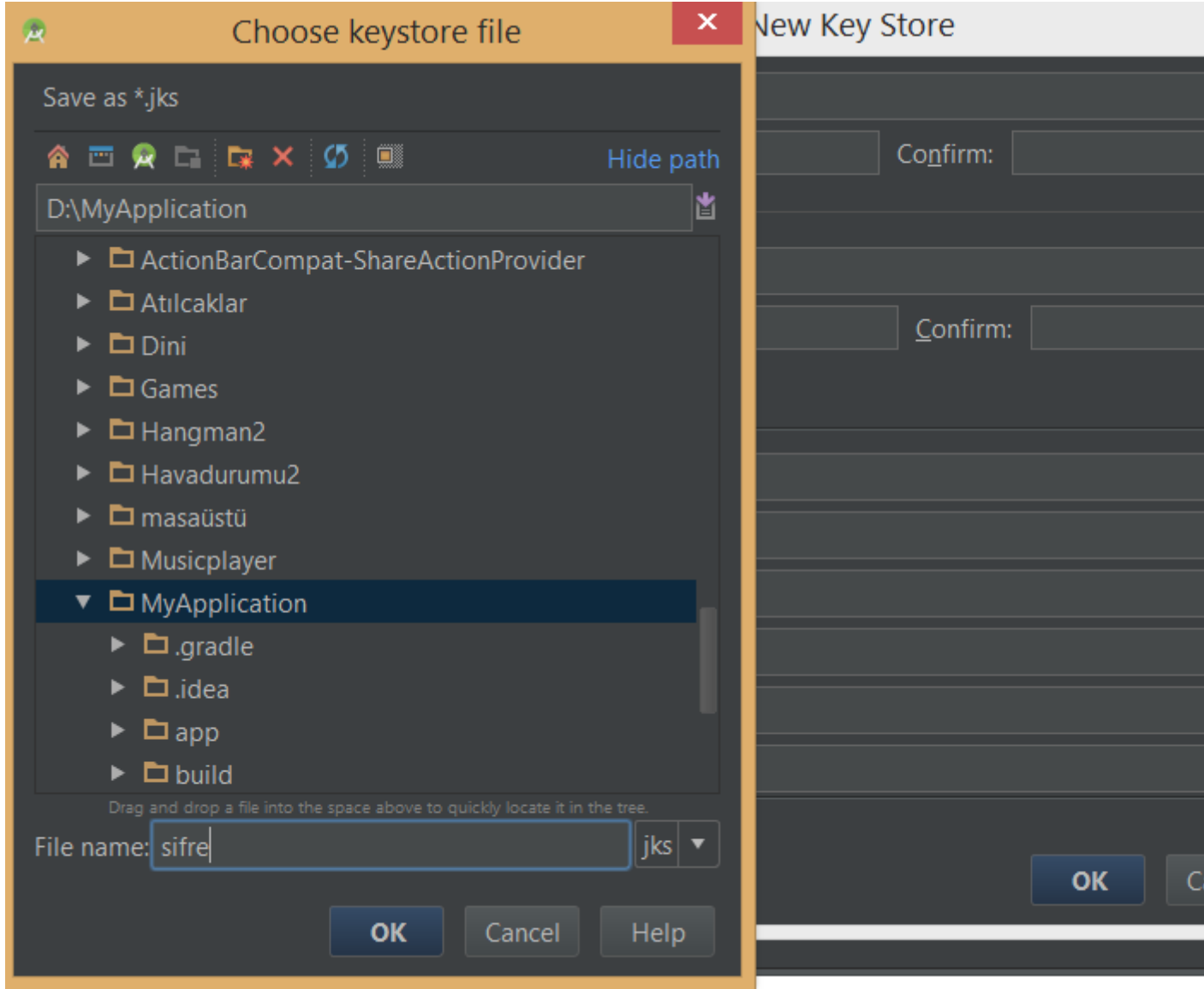
Organization:

City or Locality:

State or Province:

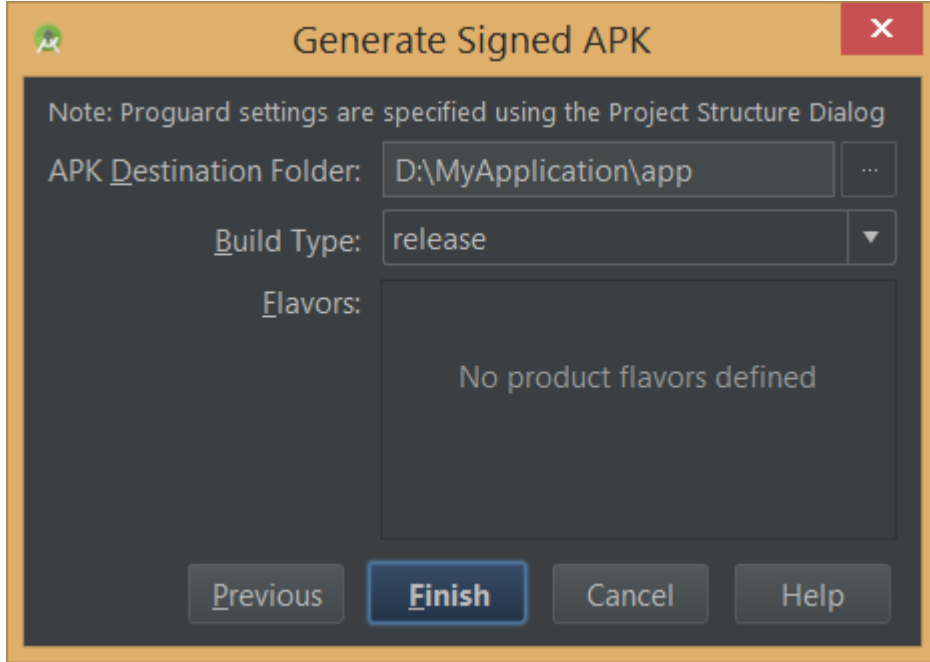
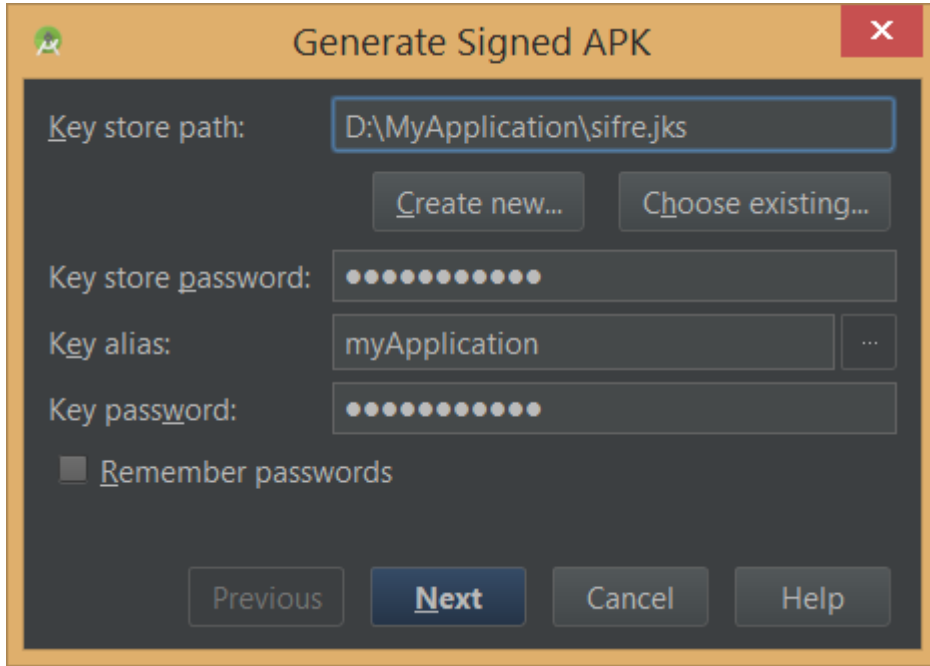
Country Code (XX):

**İşlem 4 : Ben uygulamanın içine şifreyi oluşturdum. Ve oraya da apk dosyasını oluştuyorum. Yolunu belirliyoruz.**



ation W/EGL\_genymotion: eglSurfaceAttrib not implemented

**Son işlem ise şifremizi vs oluşturup doldurduktan sonra burası gelecek Next ve finish diyerek bize en sonda apk dosyası oluşturuldu bilgisi veriliyor. Show Explore yolu göstererek apk dosyasına ulaşıyorsunuz.**



Not: Apk dosyasını Telefona yüklemek için bilinmeyen kaynakları açmanız gerekmektedir. Telefonunuzun ayarlar bölümünde olması lazım. Tıkladığınız an bu kadar uygulama yüklenmiştir.

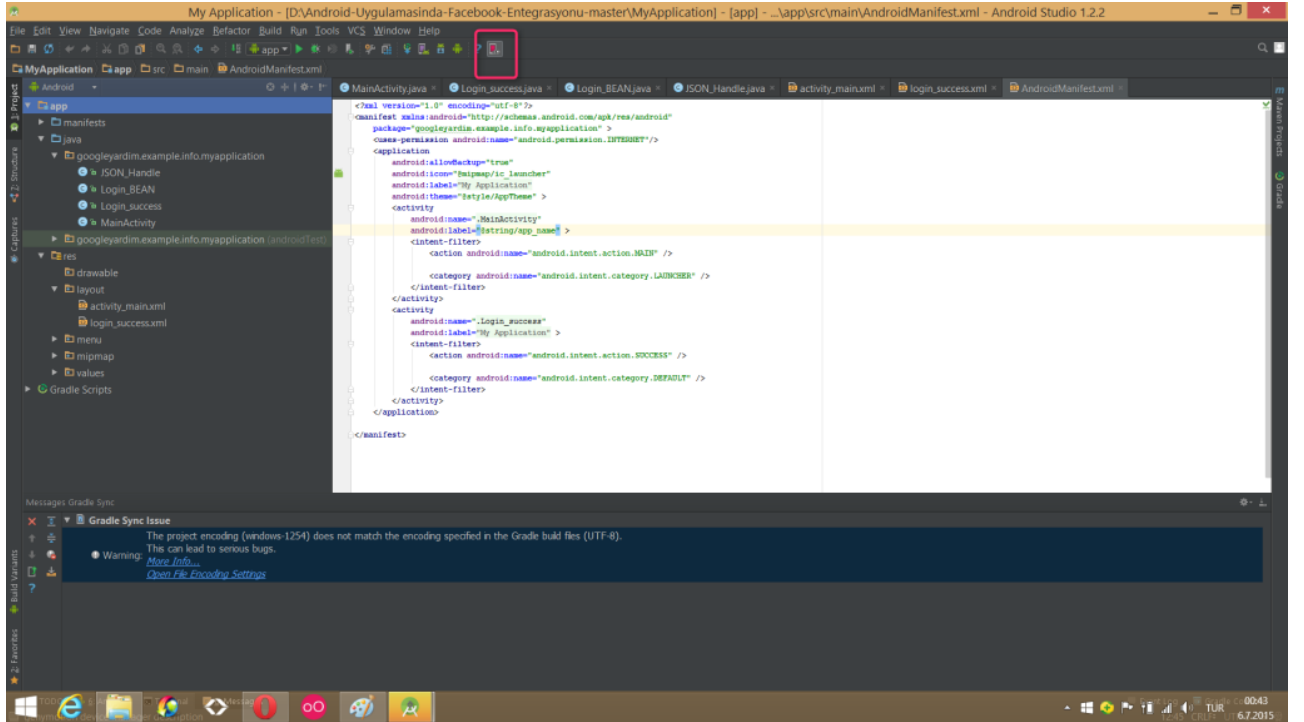
# Android Studio Genymotion Kurulumu

## Android Studio Genymotion Kurulumu

Android için hızlı bir emülatör olan Genymotion kurulumundan bahsedeceğim bugün. Öncelikle bir üyelik karşılığında <https://www.genymotion.com/> sitesinden emülatörü free olarak indirin. Dediğim gibi bir üyelik isteyecek. Öncelikle üye olduktan sonra direk indirebilirsiniz. Üye id ve pass'ı unutmayın İndirdiğiniz dosyayı kurulacak yeri seçip next next diyoruz yüklüyoruz.

Kurulumu yaptıktan sonra üye id ve pass'ı girerek genymotion'a telefonunuzu seçerek yüklüyorsunuz. Belirli dosya boyutunda telefonunuzun sürücüsünü indiriyor.

Android Studio'da ise **File --> Settings'e** giriyoruz **Plugins'e** tıklayarak genymotion'u kuruyoruz.



Karede'ki simgesi geldikten sonra Tıklayarak cihazımı start'a basarak çalıştırıyoruz. Artık Genymotion kurulumu hazır

## Kaynakça

---

Öncelikle bu kitabı yazarken yararlandığım bir çok site vardı. Onlarla ilgili tüm hatırladıklarım'ı buraya ekledim.

**Google**: Google'la ilgili dökümanlar olmadan olmazdı zaten.

**Androidhive**: Ravi Tamada'nın yazdığı İngilizce harika bir android sitesi. Kesinlikle takip etmenizi isteyim.

**GokselGuren**: Yazılar eski olsa da takip ettiğim bir site.

**MehmetKirazlı**: Yine takip ettiğim bir blog sitesi. Ve yine eskisi gibi güncel değil.

**AkıllıYazılım**: İçerisinde güncel ve harika yazılar var.

**Mobilhanem**: Android ile ilgili güzel yazılar var.

**GeleceğiYazanlar**: Turkcell tarafından kurulmuş içerisinde bir çok işletim sistemine göre yapabileceğiniz uygulamalar olduğu site.

Bunun gibi daha bir çok site var. Benim aklıma gelenler şuan bunlar. Daha çok internette araştırma yaparak zamanı geçiriyorum. Ama en çok problemlerime çözüm bulan siteler bunlardı. [Umiitkose.com](http://Umiitkose.com) 'da benim sitem. Takip etmeyi unutmayın. En kısa zamanda Google play' de uygulamalarım olacak. Ve bu kitabın devam yazıları orada yayınlanacak. Giriş yaptığımızı göre artık işleri daha da zorlaştırma vakti geldi. Herkese iyi okumalar. Size bu kitap umarım bişeyler katmıştır.