

# Mobil Uygulama Programlama

Dr.Öğ.Üyesi Ömer ÇETİN

Konu-3

## Bu Derste

---

- Uygulama konfigürasyonu ve manifest dosyası
- Aktiviteler, aktivite durumları, yeni aktivite oluşturma
- Intent, Service, Broadcast, Content Providers
- Resources (kaynaklar)

Dr.Öğ.Üyesi Ömer ÇETİN

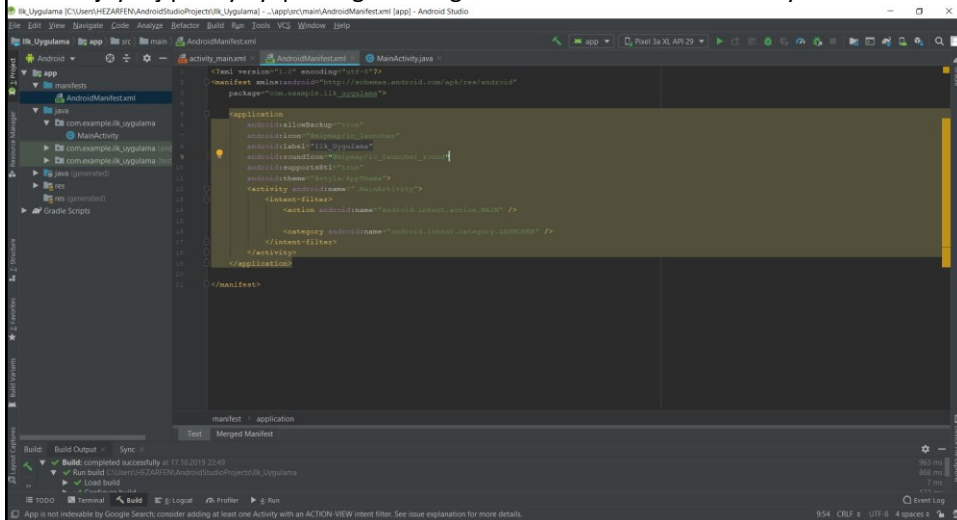
## Manifest Dosyası –AndroidManifest.xml

- Manifest dosyası proje **hiyerarşisinin en tepesinde** yer alır ve uygulamanın **bütün parçalarının bir araya geliş şeklini** tanımlar.
- Bu dosya içinde, kullandığınız bütün elemanları (Activity, Service, ContentProviders vs), uygulama gereksinimleri vb tanım yapılabilir.

Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml

- Projeyi açıp dosya yapısına göz atıldığında AndroidManifest.xml dosyasını



Dr.Öğ.Üyesi Ömer ÇETİN

# Manifest Dosyası –AndroidManifest.xml

- İlk uygulama projesi için otomatik olarak oluşturulan AndroidManifest.xml dosyası

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

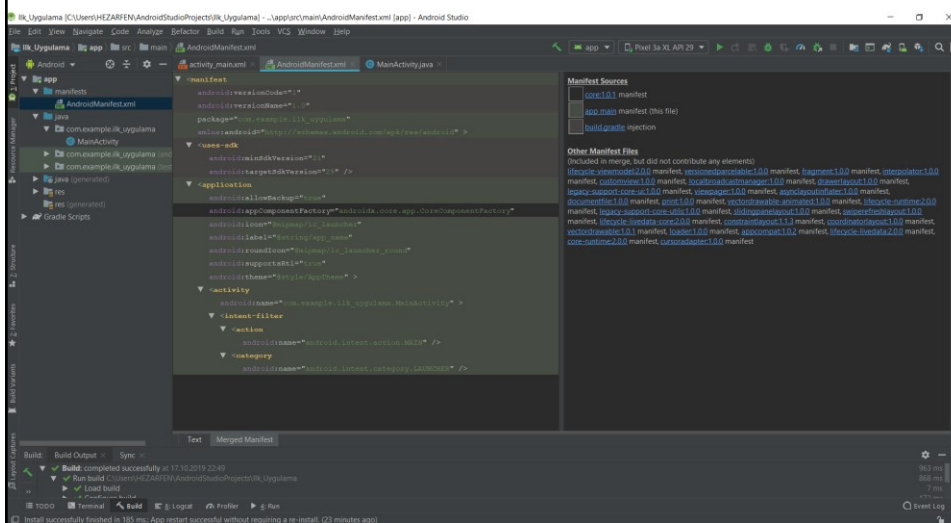
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Dr.Öğ.Üyesi Ömer ÇETİN

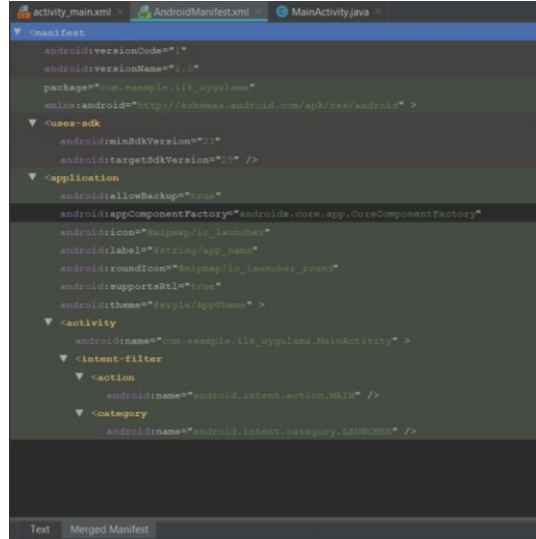
# Manifest Dosyası –AndroidManifest.xml

- İlk uygulama projesi için otomatik olarak oluşturulan AndroidManifest.xml dosyası



Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml



Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml

- Manifest dosyası için 5 ayrı sekme bulunur;
  - Manifest
  - Application (Uygulama)
  - Permissions (izinler)
  - Instrumentations (araçlar)
  - AndroidManifest

Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml

---

- **Version code:** Uygulamanızın versiyonu → Uygulamanın versiyonlanmasında kullanılır (Güncelleme)
- **Version name:** Uygulamanızın kullanıcılara gösterilecek olan versiyon adıdır.
- **package:** Java sınıflarının mantıksal gruplara ayrılması
- **Uses SDK:** Uygulamanız için çalışabilecek minimum, maksimum ve hedef Android SDK versiyonları
  - Hangi ekran tiplerinin desteklendiği, hangi klavye tipi ile çalışabileceği vb ayarlar
  - Fotoğraf çeken bir uygulama geliştirekseniz uygulamanın kurulabilmesi için cihazda kameranın bulunması şart koşulur.

Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml

---

- **Label** ve **icon** alanları uygulamanın simgesini ve simgenin altında yazılacak olan adı temsil eder
  - @string/app\_name
  - @drawable/ic\_launcher
- **Debuggable** uygulamanın adım adım hata ayıklama (debug) yapması için kullanılır.
  - Uygulama geliştirme süreciniz boyunca hata ayıklama işlemleri olacağından TRUE
- **Application Nodes** alanında kullanılan uygulama elemanları yer alır.
  - Activity, Service, ContentProvider, Broadcast Receivers...

Dr.Öğ.Üyesi Ömer ÇETİN

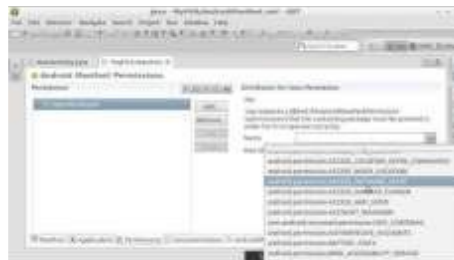
## Manifest Dosyası –AndroidManifest.xml

- İlk\_Uygulama projesinde tek bir activity vardı → MainActivity
- **Name** → aktivitenin sınıf adı
  - . → Manifest sekmesindeki package tanımına göre Activity sınıfının konumunu belirtir. (com.example.MainActivity)
- **Label**: Aktivite açıldığında ekranın üst tarafında ne yazılmasını isteniyorsa o isim.
- **Intent Filter**: Android İS bir uygulama başlatacağı zaman öncelikle manifest dosyasını okur ve içinde LAUNCHER kategorisine sahip MAIN action'ı içinde barındıran intent filter tanımını arar.
  - Bulduğu zaman ilgili Activity sınıfını başlatır. Default Aktivite seçimi aslında intent filter eklemek anlamındadır.

Dr.Öğ.Üyesi Ömer ÇETİN

## Manifest Dosyası –AndroidManifest.xml

- **Permissions**
  - Projenin hangi komponentleri kullanması gerektiğine karar verildikten sonra ilgili izinleri manifest dosyasına eklemek gerekir.
    - İnternet erişimi sağlamak gerekiyorsa ya da konum bazı bir işlem olacaksa ilgili izinleri ekleyip geliştirmeye devam etmeli → Bu yapılmazsa bu komponentler kullanılamaz
      - android.permission.Internet veya android.permission.ACCESS\_FINE\_LOCATION



Dr.Öğ.Üyesi Ömer ÇETİN

## Activity (Aktivite)

---

- Uygulamada temel olarak tek bir ekran
  - Web sitesine girildiğindeki ekran = Android uygulama activity
- Siteye girdiğinizde ilk çıkan ekran → **Default Activity**
- Oluşturulan proje → İlk Uygulama
  - **Tek bir ekran → Tek activity**
- MainActivity.java
  - public class MainActivity extends AppCompatActivity
  - → Her yeni ekran için **AppCompatActivity** sınıfından birer sınıf türetiliyor
- **Normal koşullarda uygulamada onlarca Activity tanımı yapıp kullanıcıların bu ekranlar arasında nasıl dolaşabileceklerine karar verilmesi gerekir.**

Dr.Öğ.Üyesi Ömer ÇETİN

## Activity (Aktivite)

---

- Activity sınıfını android uygulama geliştirirken bir çok kez kullanacağız. **Uygulamamızın ön planda yer alan sayfa yapısı olarak da adlandırabiliriz.**
- Bir Activity oluşturduğunuzda .java uzantılı bir class yapısı oluştuğunu göreceksiniz.
- Ayrıca, her bir Activity'nin kendine ait yaşam döngüsü vardır. Bu yaşam döngüsünün işleyişini çok iyi bilmemiz gerekiyor ki Activity yapısını düzgün bir şekilde kullanalım. Yaşam Döngüsünü incelediğimizde Activity başlatıldığında ilk çalışan method onCreate methodudur.

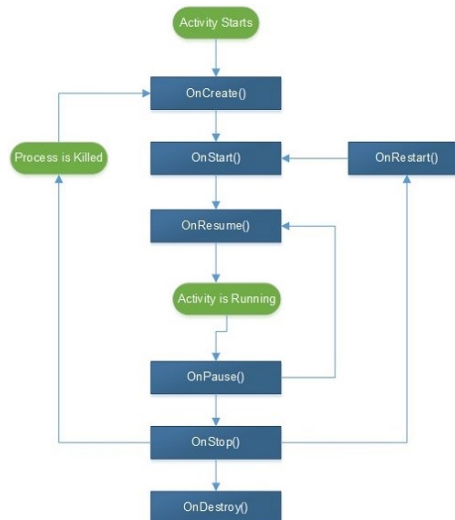
Dr.Öğ.Üyesi Ömer ÇETİN

## Activity (Aktivite)

- Android Studio da yeni bir android projesi oluşturduğunuzda ve ilgili adımları tamamladığımızda **ana Activity'e** sahip olan bir proje oluşturduğumuzu göreceksiniz. **MainActivity** android uygulamamız başlatıldığında **ilk çalışacak olan sınıftır**.
- Her Activity'de **onCreate** methodu içinde **setContentView** olduğunu göreceksiniz. Bu kısımda activityimizin hangi görsel yapıya sahip olduğunu belirliyoruz.
- Diğer bir özellik ise oluşturulan her bir Activity sınıfının **AndroidManifest.xml** de referansı olmak zorundadır yani bir tanımlama yapılması gerekiyor eğer Android Studio da new→Activity kısmından bir Activity oluşturduysak otomatik olarak Activityimizin referansının AndroidManifest.xml de oluştuğunu göreceksiniz.
- Diğer bir yöntem ise öncelikle bir new→Java Class oluştururuz daha sonra sınıfımızı Activity den türetiriz (extends ederiz) otomatik olarak bir layout oluşmadığından ayrıca bir layout oluştururuz ve **setContentView** de tanımlarız.

Dr.Öğ.Üyesi Ömer ÇETİN

## Activity (Aktivite)



Dr.Öğ.Üyesi Ömer ÇETİN



## Activity (Aktivite)

---

- **onCreate** : Activity başlatıldığında ilk çalışan method.
- **onStart** : onCreate methodu işlemini tamamladıktan sonra çalışan method.
- **onResume** : Activity çalıştıktan sonra bütün ön yüzdeki işlemler bittiğinde activity devamı için çalışan kısımdır. Activity arka plana geçip tekrar ön plana geldiğinde onResume methodu tetiklenir ve Activity çalışmaya devam eder.
- **onPause** : Activity arka plana alındığında bu metod tetiklenir.
- **onStop** : Aktivite arkaplana atıldığında çalışan bir diğer metoddur. Kullanıcı Aktivite'ye geri döndüğünde onRestart methodu tetiklenir ve onStart methodundan itibaren yaşam döngüsü devam eder. Eğer kullanıcı geri dönmezse ilgili Activity'e (bu durumda Activity sonlandı veya sistem tarafından kapatıldıysa) onDestroy methodu tetiklenir.
- **onDestroy** : Yaşam döngüsünün en son kısmıdır. Activity sonlandığında tetiklenir. Bellekte uygulama için kullanılan tüm kaynaklarda sonlandırılmış olur.

Dr.Öğ.Üyesi Ömer ÇETİN

## Intent

---

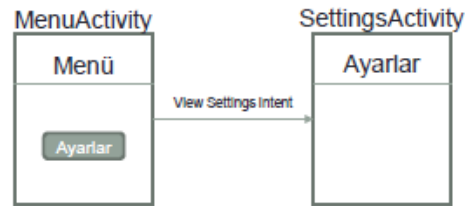
- Temel Android bileşenleri arasında gönderilen mesajları temsil eder.
  - Bu mesaj sayesinde bir aktivitenin başlamasını tetikleyebilir; bir servis çağrısı yapılabilir; cihazdaki bütün uygulamaları bir durumdan haberdar edilebilir.
- Kelime anlamı **niyet**
  - Android ortamında da niyeti;
    - Açıkça söyleyebilirsiniz → **explicit intent**
    - Üstü kapalı şekilde söyleyebilirsiniz → **implicit intent**

Dr.Öğ.Üyesi Ömer ÇETİN

# Intent

## • Örnek:

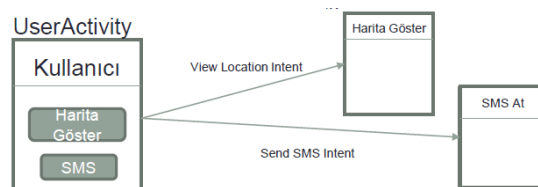
- Uygulamanın menü ekranına ayarlar düğmesi
- Düğmeye tıklandığında ayarlar aktivitesini başlatmanız gerekiyor.
- Yapmak istenilen açık → explicit intent



Dr.Öğ.Üyesi Ömer ÇETİN

# Intent

- Implicit Intent (Üstü Kapalı), ne yapmak istediğinizi bildiğiniz, ama bunu kimin yapması gerektiği konusunda bir karar verilmeyen durumlarda kullanılır.
- Örnek
  - Rehberdeki kişiler üzerinden konum bazlı bir uygulama
  - Bu kişilerin konumlarının harita üzerinde görülüp; onlara SMS atıldığını düşünelim.
  - Rehberdeki bir kişiyi seçtiniz ve kullanıcı ekranına geldiniz diyelim.
  - Bu kişinin konumu veritabanında saklanıp; harita üzerinde göstermek istenilir. → Telefonda birden fazla harita üzerinde konum gösterecek uygulama varsa kullanıcı hangi uygulamayı kullanacağını kendisi seçebilir.



Dr.Öğ.Üyesi Ömer ÇETİN

## Service

---

- **Servisler** uygulamada arka planda çalışan ve herhangi bir **arayüze sahip olmayan** parçalarıdır.
- Servis başlatıldıktan sonra hangi uygulama açık olursa olsun **çalışmaya devam eder**.
- **Kullanıcı etkileşimi olmayan** işlemleri servisler üzerinden yerine getirebilirsiniz.
  - Müzik çalar uygulaması → Uygulamayı kullanan kişiler müzik dinlerken başka işlemler yapabilmeleri gerekiyorsa müzik çalma işlemi için bir servis yazılmalı

Dr.Öğ.Üyesi Ömer ÇETİN

## Broadcast Receivers (Yayın Algılayıcıları)

---

- İşletim sistemi genelinde **olaylardan haberdar olmayı** ve bu olaylar gerçekleştiğinde **tepki verilebilmesini** sağlar.
- Aynı zamanda uygulamadan sistem geneline ya da uygulamanın başka bölümüne haber göndermek amacıyla kullanılabilir.
- Bir arama gerçekleştiğinde, SMS ulaştığında, telefon açıldığında/kapandığında vb.. İşletim sistemi olayın gerçekleştiğine dair bir broadcast yapar.
  - Haberdar olmak isteyen uygulamalar, bu yayınlara kayıt olur
    - Müzik uygulaması → kullanıcı kulaklığı çıkardığında müziği sonlandırma

Dr.Öğ.Üyesi Ömer ÇETİN

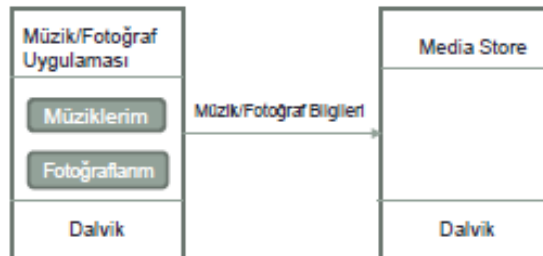
## Content Providers (İçerik Sağlayıcılar)

- Uygulamalar arasında **veri paylaşımı** için kullanılan arayüzlerdir.
- Uygulamada veritabanı oluşturulduğunda; bu veritabanı sadece uygulamanın erişimine açıktır.
  - Bazı bilgiler işletim sistemi genelinde paylaşılmalı
  - İçerik sağlayıcılar veritabanını farklı uygulamaların erişimine açılmasını ya da bu uygulamalar tarafından sağlanan içeriğe ulaşılmasını sağlar.
- Android İS kullanıma sunduğu çok sayıda içerik sağlayıcı vardır.
  - Contacts(Rahber) Provider, Settings Provider, Media Store...

Dr.Öğ.Üyesi Ömer ÇETİN

## Content Providers (İçerik Sağlayıcılar)

- **Örnek:** Media Store aslında herhangi bir arayüze sahip olmayan sadece içerik paylaşan bir uygulama olabilir.
- Uygulamanızdan bu içeriğe erişip yeni kayıt ekleyebilir (insert), mevcut kayıtları okuyabilir (read), silebilir (delete) ya da sorgulayabilirsiniz.
- Mantık; aynı içeriği isteyeninin istediği gibi görüntüleyebilmesini ve kullanabilmesine olanak sağlamaktır.
  - Farklı telefon üreticileri farklı rehber, müzik çalar vs.. sunar.



Dr.Öğ.Üyesi Ömer ÇETİN

## Application Context – Uygulama Ortamı

- Projedeki bütün komponentler, **contextlerden** türetilmiştir.
  - Uygulamadaki ilk component çalışmaya başladığı anda bir context oluşturulur → proje hayat döngüsü boyunca var olur
- Application Context kullanarak proje içindeki farklı kaynaklara (resource) erişim sağlayabilirsiniz.

Dr.Öğ.Üyesi Ömer ÇETİN

## Uygulama Öncelikleri

- Sistem bir şekilde bazı uygulamaları diğerlerinden ayırt edebilmeli ve sonlandırmak ya da bellekte tutmak arasında tercih yapmak durumunda uygulamaları önceliklendirebilmek için;
- 1. Uygulamanın o anda kullanılıyor olup olmadığı:**
    - Aktif durumdaki uygulamalar her zaman öncelikli olanlardır. Yok edilmeleri mümkün fakat listenin sonunda yer alırlar.
  - 2. Paused durumdaki aktiviteler:**
    - Görünür fakat başka dialog penceresi arkasında kalan ekran. ☐ mecbur kalınmadıkça yok edilmezler
  - 3. Çalışır durumdaki servisler:**
    - Müzik çalma işlemi servis aracılığıyla yapılıyorsa; müzik çalarken sistem bundan daha öncelikli process lere ihtiyaç duymadıkça bu servisi kullanan işlem sonlandırılmaz
  - 4. Çalışmayan arka plan servisleri**
  - 5. Uzun süredir kullanılmayan uygulamalar:**
    - Sistem kaynak sıkıntısı çekmedikçe bellekte tutulur.

Dr.Öğ.Üyesi Ömer ÇETİN

## Resources (Kaynaklar)

- XML yapılandırma dosyaları ve görseller için Android **statik kaynak oluşturma** imkanı sağlar. Bu tasarım, **kaynakları(resources)**, Android uygulamanızın kodundan ayırmanızı sağlar.
- Geliştirilen uygulama çoklu dil desteği eklemek istenirse → Her dil için proje tekrar derlenip farklı uygulama olarak kullanıcıya sunmak gerekir.
- Android İS için Resource kavramı bu boşluğu doldurur.

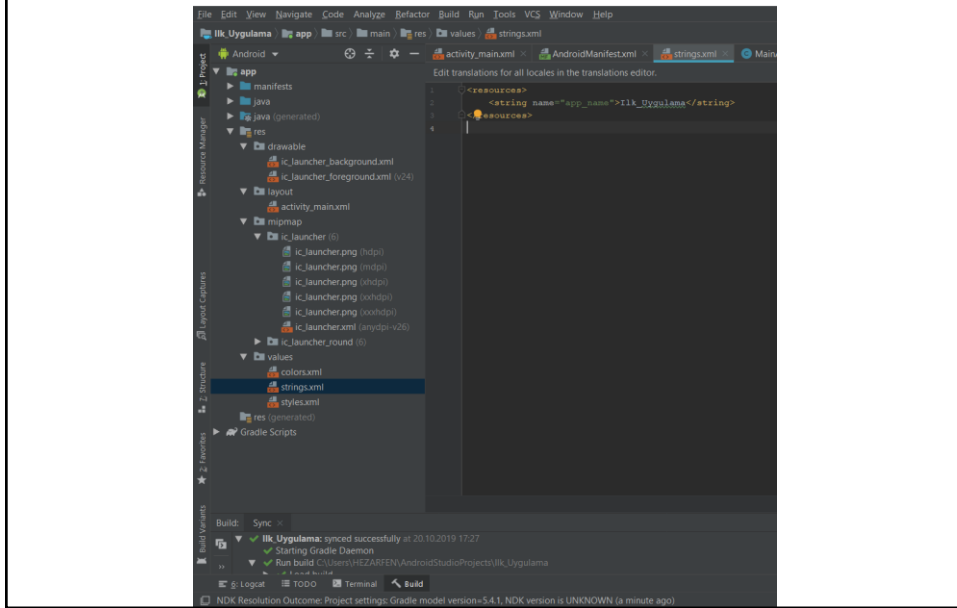
Dr.Öğ.Üyesi Ömer ÇETİN

## Resources (Kaynaklar)

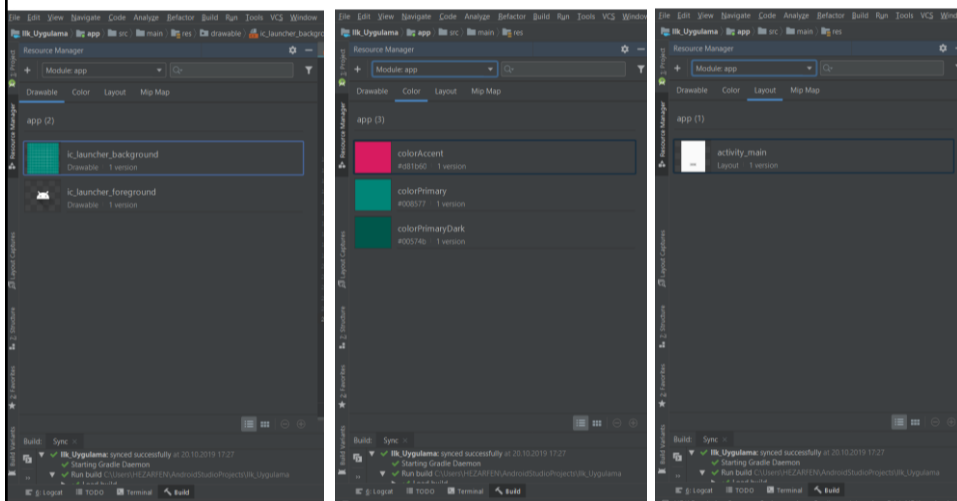
- String, ölçü, renk, resim, animasyon, hatta ekran tasarımı olabilir.
  - Bu kaynakları kullanarak uygulama farklı dillerde, farklı donanımdaki cihazlarda, farklı konumlarda, farklı davranırılabilir.
- Kaynak dosyaları, uygulamanızda daha önce oluşturulmuş olan **/res dizinin** altında bulunmalıdır. Daha özelleştirilmiş alt dizinler, kullanacağınız kaynak türüne göre değişebilir. Özel bir yapılandırma için kullanılacak kaynak klasörlerinin isimlendirilmesinde ek niteleyiciler kullanabilirsiniz. Bunlara **kaynak niteleyici (resource qualifiers)** denir. Örneğin, layout dosyasının sadece belli ekran boyutunda geçerli olduğunu belirtebilirsiniz.
- Projede → res klasörü
  - Bu klasör içinde drawable-hdpi, drawable-ldpi, drawable-mdpi, layout, values ve resource

Dr.Öğ.Üyesi Ömer ÇETİN

# Resources (Kaynaklar)

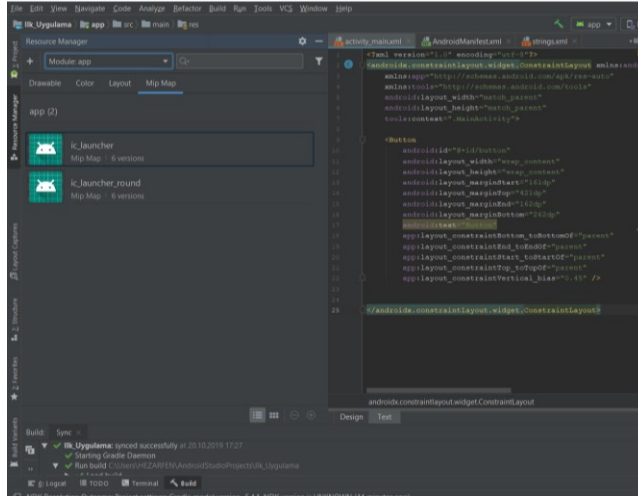


# Resources (Kaynaklar)



Dr.Öğ.Üyesi Ömer ÇETİN

## Resources (Kaynaklar)



Dr.Öğ.Üyesi Ömer ÇETİN

## Resources (Kaynaklar)

- Aşağıdaki tabloda desteklenen kaynakları ve bunların klasör isimlendirilmesinde kullanılan standart ön eklerini bulabilirsiniz.

Kaynak	Klasör	Açıklama
Çizilebilir	<i>/res/Drawables</i>	Görseller (png, jpeg dosyaları), vektörel çizimler ya da Android cihazın matlığını otomatik olarak ayarlayan XML dosyaları
Basit Değerler	<i>/res/Values</i>	Metin, renk, boyut, biçim ve string ya da tam sayılardan oluşan statik dizilerden oluşan XML dosyaları. Geleneksel olarak her tür ayrı dosya olarak saklanır. Örneğin, stringler res/values/strings.xml dosyasında tanımlanır.
Layouts (Yerleşim)	<i>/res/Layout</i>	Kullanıcı arayüzü tanımları için kullanılan activity'ler ve fragment'ları içeren yerleşim bildirimlerinden oluşan XML dosyaları
Biçimler ve temalar	<i>/res/Values</i>	Android uygulamanızın görünümü ile ilgili tanımlamaların yapıldığı dosyalar

Dr.Öğ.Üyesi Ömer ÇETİN



## Resources (Kaynaklar)

- Aşağıdaki tabloda desteklenen kaynakları ve bunların klasör isimlendirilmesinde kullanılan standart ön eklerini bulabilirsiniz.

Kaynak	Klasör	Açıklama
Animasyonlar	/res/animator	Animasyon API'si için XML dosyasında tanımlanan animasyonlar, nesnelerin zamanla nasıl şekilleneceğini belirlemenizi sağlar.
Ham veri	/res/raw	Bu dosyalar ham olarak kaydedilir. InputStream nesnesini kullanarak bunlara erişebilirsiniz.
Menüler	/res/menu	Uygulamanın araç çubuğunda kullanılan eylemleri tanımlamak için kullanılır.

Dr.Öğ.Üyesi Ömer ÇETİN

## String Resources

- Values klasörü açıldığında içinde strings.xml resource dosyası görülebilir.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Merhaba Dünya</string>
    <string name="hello_world">Hello World!</string>
    <string name="title_activity_merhaba_dunya">Merhaba
Dünya</string>
</resources>
```

- Projede kullanılacak olan string tanımları bu dosyada yapılabilir.
- Siz herhangi bir dosya oluşturmadan sistem arka planda bunu yapar.

Dr.Öğ.Üyesi Ömer ÇETİN

## String Resources

---

- Yapılan tanımların ekranda nerede kullanılmasına nasıl karar verildi?
  - AndroidManifest.xml üzerinde Application sekmesinde
    - Application Attributes ve Attributes for .MerhabaDunyaActivity bölümlerinde;
      - Label alanlarında @string/app\_name tanımları → ilgili alanlarda string resource içindeki app\_name mesajı kullanılır
- Yeni bir string tanımlı yapmak için doğrudan XML dosyasına

```
<string name="isim">Değer</string>
```

Veya resource ekranını kullanarak Add düğmesi yardımıyla string seçimini yaparak aynı işlem yerine getirilebilir.

Dr.Öğ.Üyesi Ömer ÇETİN

## Color Resources

---

- Renk kaynaklarının string kaynaklarından herhangi bir farkı yoktur.
- Tanımı yapılacak şey, uygulamada kullanılacak olan farklı renklerdir.
- Renk kaynakları için **colors.xml** dosyası oluşturup içine bir renk tanımlı eklenmeli.
  - Başka bir dosya ismi de oluşturulabilir – size bağlı → istenirse bütün basit kaynak tanımları tek bir kaynak dosyasında yapılabilir.

Dr.Öğ.Üyesi Ömer ÇETİN

## Color Resources

- colors.xml dosyasının içeriği kontrol edilirse, strings.xml dosyasına benzer bir yapıyla karşılaşılr.
- colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="ekran_arka_plani">#FFF</color>
</resources>
```

- Tanımları ayrı bir kaynak dosyasında yapıyor olmanın avantajı, sadece kaynak dosyasında değişiklik yaparak bütün ekranların arka planı tek seferde değiştirme olanağı
- Color Resource tanımında kullanılan değerler # ile başlar ve Alfa, Kırmızı, Yeşil ve Mavi değerlerini 16'lık düzende formatlarda alabilir

Dr.Öğ.Üyesi Ömer ÇETİN

## Dimension Resources (Ölçü Kaynakları)

- Ölçü değerlerini içeren kaynak dosyalarıdır.

<b>dp</b>	Dpi (dots per inch) 1 inch (2.54 cm) uzunluk içinde tanımlı renk noktası sayısını ifade eder. 1 dp Android için 160 dpi lik bir ekranda bir pixellik ölçüye denk gelir.
<b>sp</b>	dp ye benzer bir anlama gelir. Fakat kullanıcının font tercihiine göre değişkenlik gösterir.
<b>pt</b>	1/72 inch.
<b>px</b>	Pixel-ekrana göre değişkenlik gösterdiğinden kullanılması tavsiye edilir.
<b>mm</b>	Milimetre.
<b>in</b>	Inch.

Dr.Öğ.Üyesi Ömer ÇETİN

## Dimension Resources (Ölçü Kaynakları)

---

- **dimens.xml**

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

<resources>

    <dimen name="textbox_yukseklık">75dp</dimen>

</resources>
```

Dr.Öğ.Üyesi Ömer ÇETİN

## Drawable Resources

---

- Projede kullanılan her türlü çizim kaynağı.
  - Bir simge ya da arkaplan görüntü dosyası vb..
- res klasörü içerisinde drawable-hdpi, drawable-ldpi, drawable-mdpi gibi 3 grup
  - Drawable adıyla başlayan bu gruplarda çizim dosyaları bulunmaktadır.
  - Cihazın ekran çözünürlüğüne bağlı olarak hdpi → high, mdpi → medium, ldpi → low
  - Ekran boyutuna bağlı olarak farklı boyutlarda simgeler kullanılarak tutarsız görüntülerin oluşması engellenir.
  - Hangi kaynak dosyasının seçileceği sistem tarafından otomatik olarak belirlenir.

Dr.Öğ.Üyesi Ömer ÇETİN

## Drawable Resources

---

- Ayrıca AndroidManifest.xml dosyasında Application sekmesi açıldığında **Icon** alanında **@drawable/ic\_launcher** tanımı görülür
- Drawable resource tanımı string tanımlarından farkı ortada bir XML dosyasının bulunmaması ve @drawable ifadesinden sonra gelen ic\_launcher tanımının bir dosya adına denk gelmesi  
(örneğin ic\_launcher.png)

Dr.Öğ.Üyesi Ömer ÇETİN

## Layout Resources

---

- Aktivite arayüzlerinin XML dosyaları olarak tanımlanmasının genel adıdır.
- Bütün aktiviteler için bir tane resource dosyası olması beklenir.
- res klasörü içerisinde layout klasöründe Layout Resource tanımları yapılır.
- → Tek bir Activity olduğunda tek layout resource → main.xml ( siz layout a hangi ismi verdiyseniz)

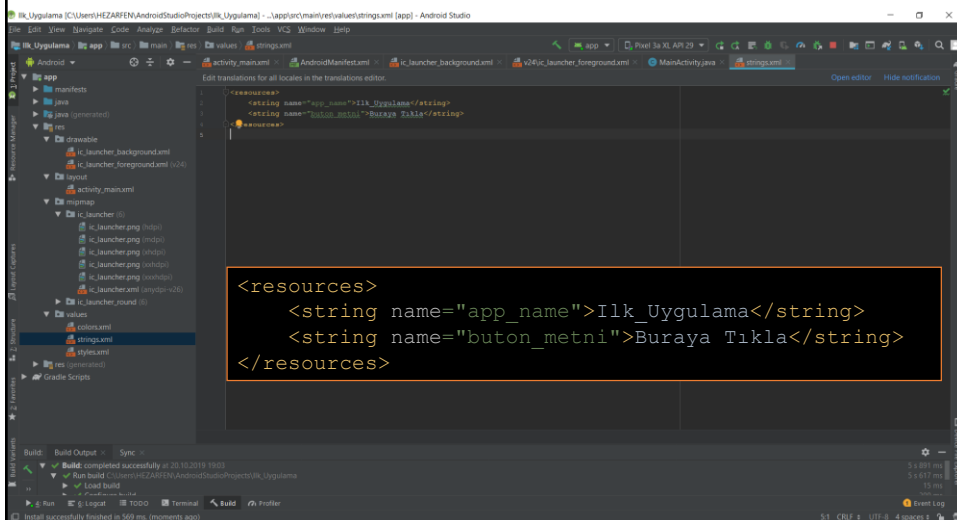
Dr.Öğ.Üyesi Ömer ÇETİN

## Layout Resources

- ADT eklentisi sayesinde grafiksel arayüz tasarımı yapılabilir.
- Sol taraftaki komponentlerden istenilen sürüklenerek aktivitenin üzerine bırakılabilir
  - main.xml dosyasının içeriğinin değiştiğini main.xml sekmesinden görülebilir.

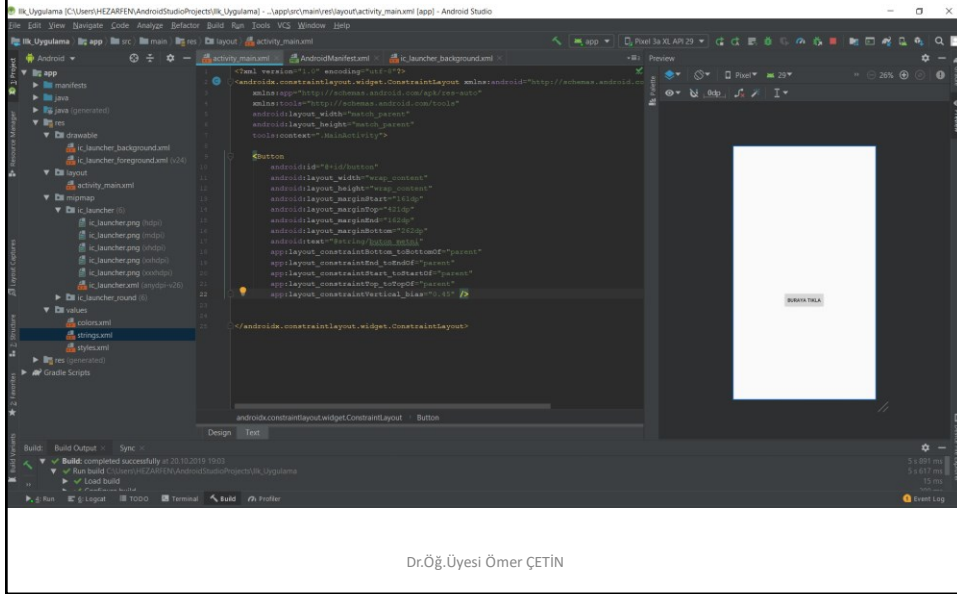
Dr.Öğ.Üyesi Ömer ÇETİN

## Örnek Uygulama



Dr.Öğ.Üyesi Ömer ÇETİN

# Örnek Uygulama



Dr.Öğ.Üyesi Ömer ÇETİN

# Örnek Uygulama

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="161dp"
    android:layout_marginTop="421dp"
    android:layout_marginEnd="162dp"
    android:layout_marginBottom="262dp"
    android:text="@string/buton_metni"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.45" />
```

Dr.Öğ.Üyesi Ömer ÇETİN

## Örnek Uygulama



Dr.Öğ.Üyesi Ömer ÇETİN

## Kaynaklar - Örnek

- Örnek: Stringlerin, String dizilerinin, Renklerin ve Boyutların Tanımlanması
- Örneğin; aşağıdaki /res/values dizindeki values.xml dosyasında birkaç string sabiti, string dizisi, renk ve boyut tanımlanmıştır.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Test</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

    <string-array name="operationsystems">
        <item>Ubuntu</item>
        <item>Android</item>
        <item>Microsoft Windows</item>
    </string-array>

    <color name="red">#ff0000</color>

    <dimen name="mymargin">10dp</dimen>
</resources>
```

Dr.Öğ.Üyesi Ömer ÇETİN



## Kaynak Dosyalar ve R.java

---

- res dizinindeki tüm kaynaklara, **Android build sistemi** tarafından bir **ID** atanır. Android, üretilen değerleri içeren bir **R.java** dosyası oluşturur. Bu referanslar statik tam sayı değerlerdir.
- **Eğer yeni bir kaynak dosyası eklerseniz, buna ilişkin referans R.java dosyasında otomatik olarak yaratılır.** R.java dosyasında doğrudan elle değişiklik yapmak gereksizdir, çünkü bu dosyayı güncelleyen araçlar bunun üzerine yazacaktır. Android sisteminin, bu kaynaklara **ID'leri kullanarak erişmenizi sağlayacak metotları** vardır.
- Örneğin; kaynak kodunuzda `R.string.yourString` ID'sine sahip bir string'e erişmek için `Context` sınıfında tanımlanmış `getString(R.string.yourString)` metodunu kullanabilirsiniz.

Dr.Öğ.Üyesi Ömer ÇETİN