



ANDROID GÜVENLİ UYGULAMA GELİŞTİRME



~ whoami

```
1
2
3  ibrahim süren
4  -----
5  Back-end Web      Android developer (2015)
6  PHP, Python, GO   Java, Kotlin
7
8  Freelance web jobs, Material Lock, HackPoint, WebDevTools Android Apps
9
10
11  Twitter, LinkedIn, GitHub, Medium -> @ibrahimsn98
12
13 |
```



~ cat android/contents.txt

- Android nedir?
- Android mimarisine genel bakış
- Android uygulamalarına genel bakış
- Android yaşam döngüsü (Lifecycle)
- Android uygulama temelleri (Activity, intent, layout etc)
- Logcat
- ADB



~ cat android/info.txt

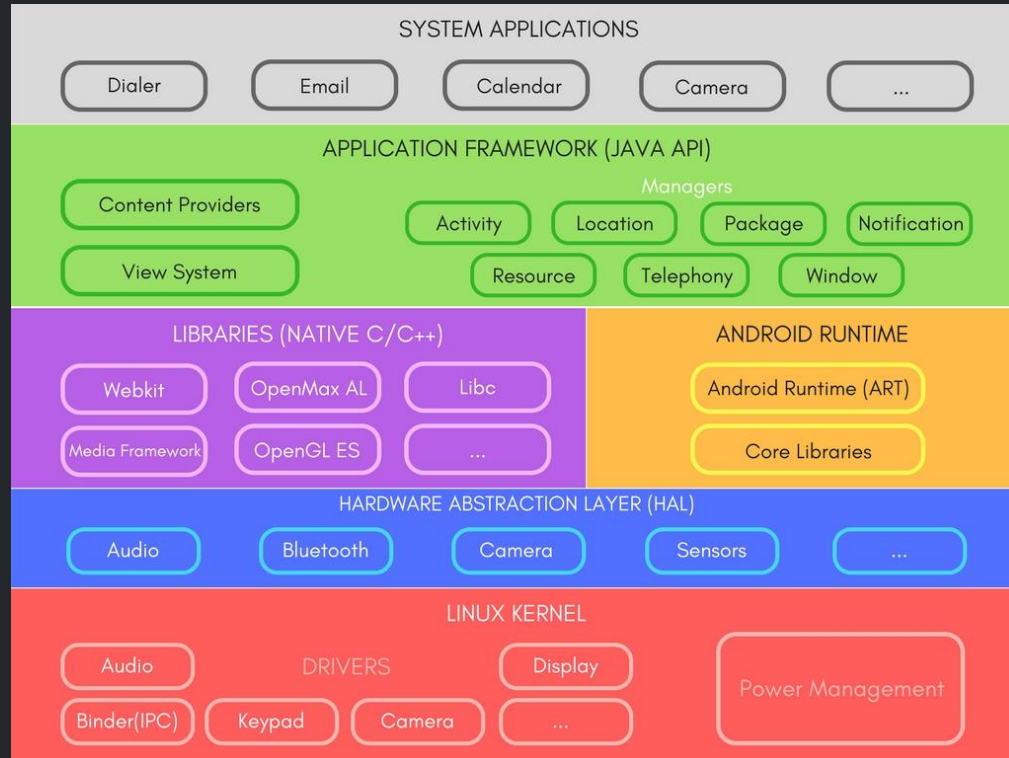
Android, Google ve Open Handset Alliance tarafından, mobil cihazlar için geliştirilmekte olan, Linux tabanlı özgür ve ücretsiz bir işletim sistemidir. Sistem açık kaynak kodlu olsa da, kodlarının ufak ama çok önemli bir kısmı Google tarafından kapalı tutulmaktadır. Desteklenen uygulama uzantısı ".apk"dır.

Java ve **Kotlin** dilleri ile native geliştirme yapılırken; Xamarin (C#), Flutter (Dart), React Native (JS) vb. cross-platform teknolojiler ile Android uygulamalar geliştirilebilir.

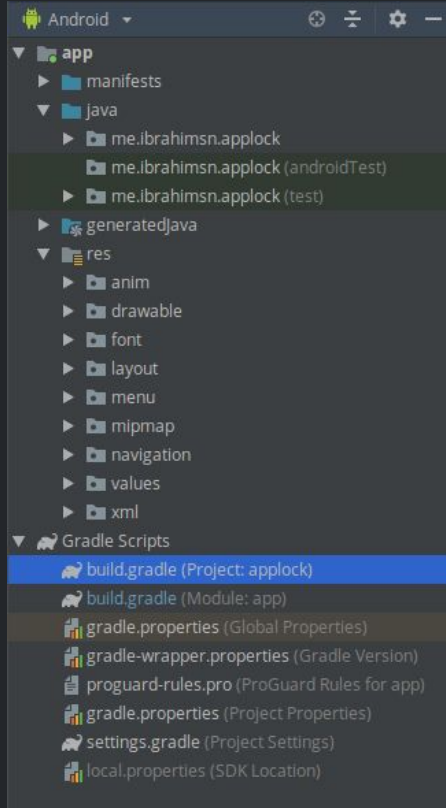
Native?: Native yazılan mobil uygulamalar, cihazın her türlü yeteneklerine ve bilgilerine erişebilir, cihaz yönetimi sağlayabilirler. Cross-platform teknolojilere göre daha güvenilir ve hızlı çalışırlar. Ancak cross-platform teknolojilerin avantajı, tek bir uygulama kodu ile? birden fazla işletim sistemi için uygulama çıkartabilmesidir.



~ cat android/architecture.txt



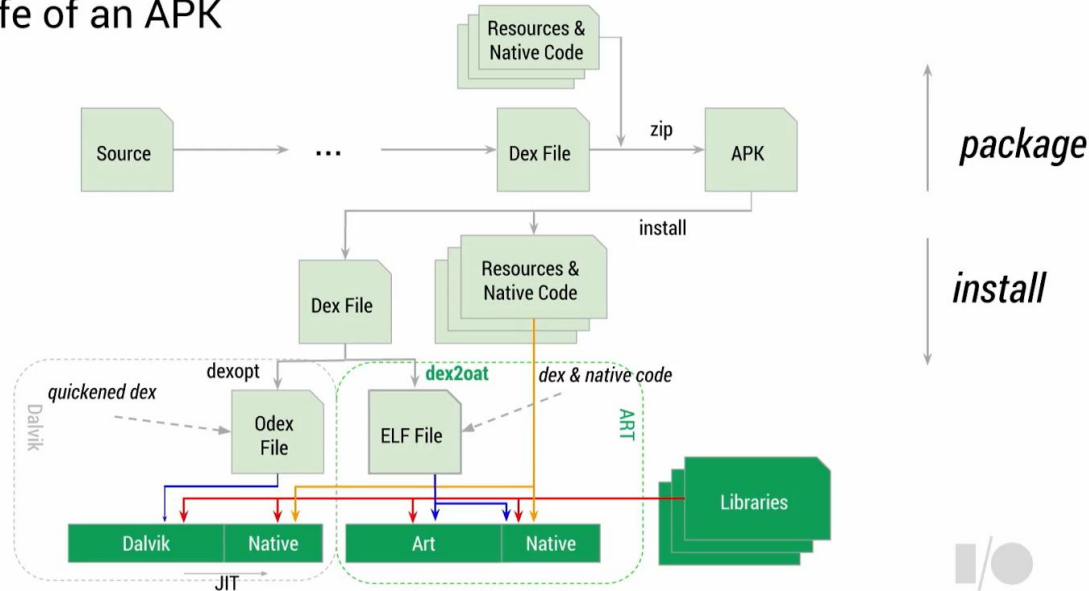
~ cat android/structure.txt



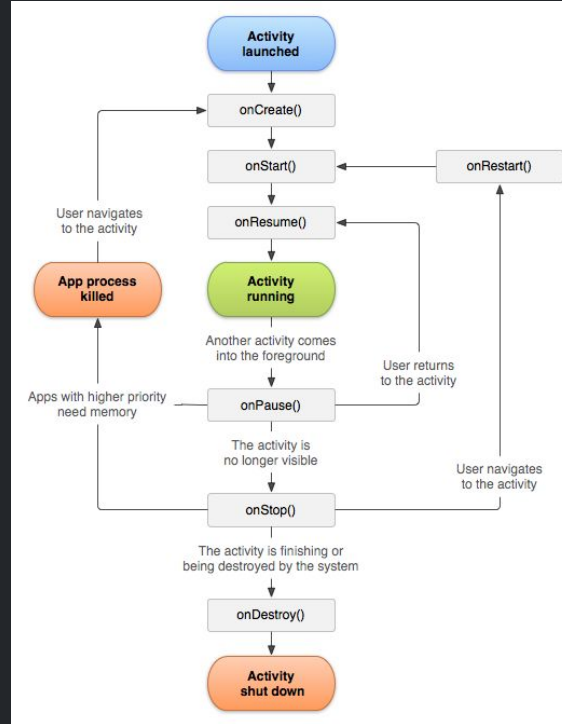
- **Java:** Uygulama mantığını içeren java dosyaları
- **Generated Java:** Compiler tarafından oluşturulan java dosyaları
- **Res:** Resimler, UI içerikleri, xml tasarım dosyaları gibi tüm kod dışı kaynaklar
- **Drawable:** Uygulama ihtiyacına göre değişen farklı tipte resimler
- **Layout:** Uygulamanın kullanıcı arayüzünü oluşturmak için kullandığımız xml tasarım dosyaları
- **Mipmap:** Resimler, logolar, ikonlar vs
- **Values:** Uygulamada kullanılan kelimeler, renk kodları, stiller
- **Manifest:** Uygulamanın kullanacağı izinler, servisler, uygulamaya ait olan activity'ler, gerekli cihaz özellikleri, bu klasörde bulunan AndroidManifest.xml dosyasında belirlenir.
- **Gradle Scripts:** Uygulamada kullanılacak modülleri, harici kütüphaneleri, minimum SDK versiyonunu vs içerir

~ cat android/life_of_apk.txt

The life of an APK



~ cat android/lifecycle.txt

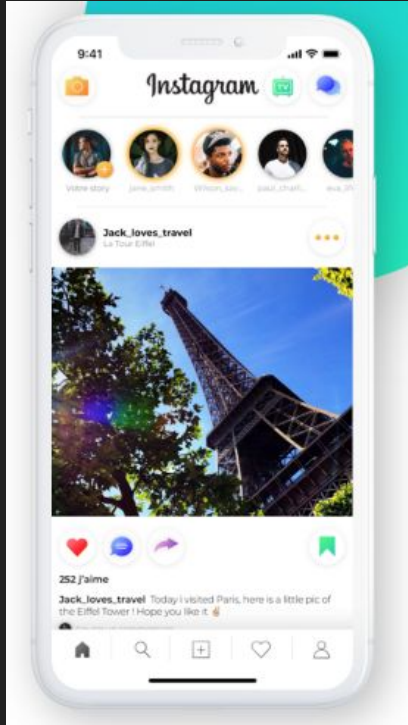




~ cat android/fundamentals.txt

- Activity
- Fragments
- Services
- Broadcast Receivers
- Layouts
- Logcat

~ cat android/activity.txt



- Activity'ler kullanıcının uygulama ile etkileşime geçebildiği pencere/arayüzlerdir.
- Her uygulama en az 1 adet Activity'e mutlaka sahiptir.

```
//Starting an Activity
startActivity(new Intent(packageContext, SignInActivity.class));

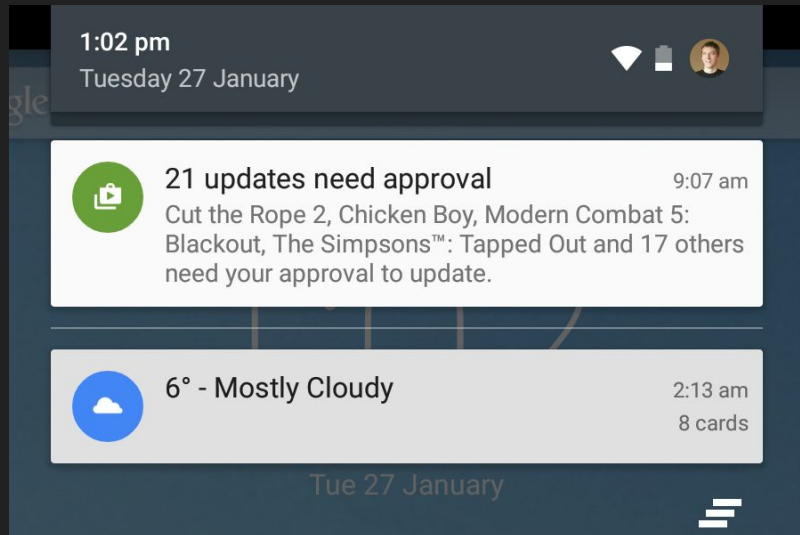
//Stopping an Activity
finish();
```

~ cat android/fragments.txt



- Yeniden kullanılabilmesi amaçlanan arayüz parçasıdır.
- Uygulamaları daha modüler ve dinamik hale getirir.
- Activity gibi kendine ait bir layout.xml dosyası ve yaşam döngüsü bulunur.
- Activity'lerin içine gömülür. Activity öldüğünde, fragment da ölür.

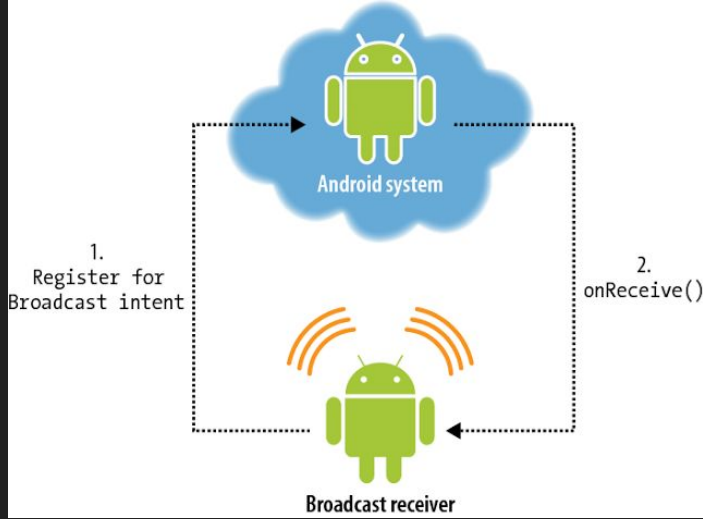
~ cat android/services.txt



- Servisler, arkaplanda çalışan uzun zamanlı operasyonlardır.
- Uygulamanın main thread'inden bağımsız çalışırlar. Uygulamayla birlikte ölebilir veya ayakta kalabilirler.

Örnek: Gerçek zamanlı bildirimler, indirme işlemleri

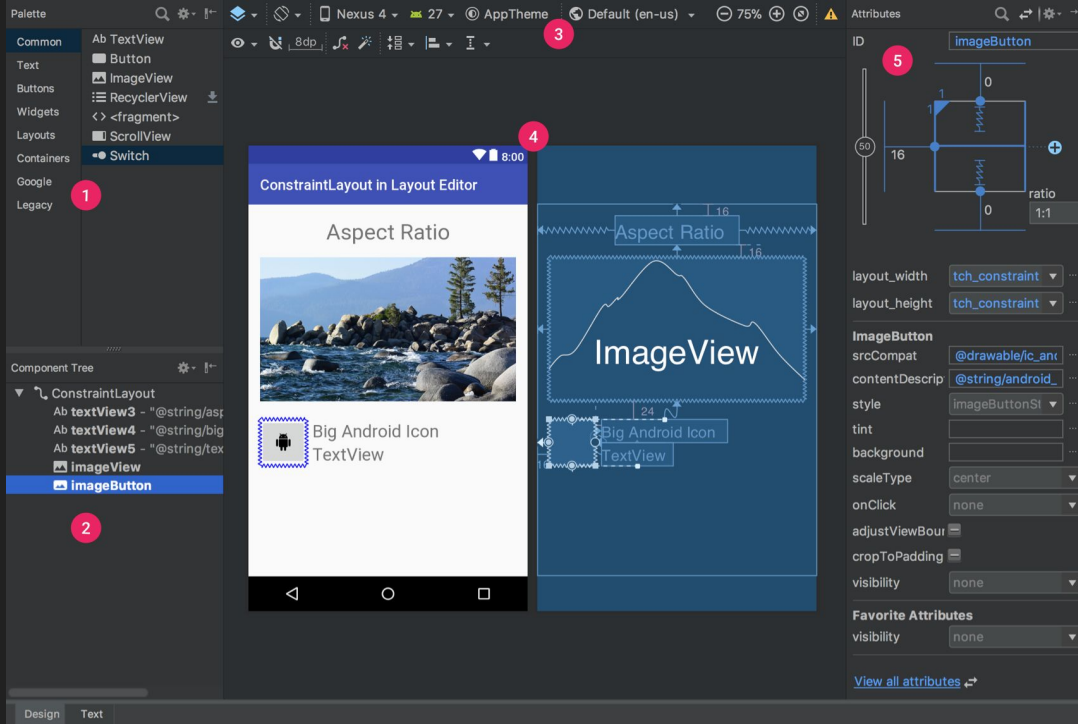
~ cat android/broadcast_receivers.txt



- Broadcast'ler, Android sistemi veya diğer uygulamalar tarafından gönderilen veya dinlenen olaylardır. BroadcastReceiver sınıfı tarafından dinlenirler.
- BroadcastReceiver sınıfları AndroidManifest dosyasında belirtilebilir veya program içinde dinamik olarak oluşturulabilir.
- Bazı broadcast dinleme işlemleri izin gerektirebilir.

Örnek: Cihazda her saat güncellemesinde, pil durumu değişikliğinde, boot veya ekran açma kapama durumlarında, sistem broadcastleri gönderilir. İhtiyacı olan uygulamalar bunları dinleyerek işlem yaparlar.

~ cat android/layouts.txt



Layout'lar kullanıcı için etkileşime geçebileceği görsel bir arayüz sunar.

Android tarafından sunulan birçok arayüz komponenti bulunur. Örnek: TextView, ImageView, Button, ProgressBar, LinearLayout, RelativeLayout, ImageButton vs.

~ cat android/layouts_example.txt

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="horizontal"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content"
7     android:paddingBottom="4dp">
8
9     <TextView
10         android:id="@+id/tvUser"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:textColor="@color/colorAccent"
14         android:text="root@aucc:~/$"
15         android:layout_marginStart="0dp"
16         android:layout_marginEnd="5dp"
17         android:textSize="15sp"/>
18
19     <TextView
20         android:id="@+id/tvCommand"
21         android:layout_width="wrap_content"
22         android:layout_height="wrap_content"
23         android:textColor="@color/colorTextPrimary"
24         android:textSize="15sp"/>
25
26 </LinearLayout>
```



~ cat android/logcat.txt

Logcat, geliştirici tarafından programa eklenen debug satırlarını veya uygulamada gelen hataları yığın izleriyle birlikte günlüğünü bırakan bir komut satırı aracıdır.

```
Logcat
Emulator APPLOCK Android 8.1.0  com.aucc.game (3284)  Verbose  Q  [x] Regex  [No Filters]
2019-03-03 14:36:41.367 3284-3289/? I/zygote: Increasing code cache capacity to 256MB
2019-03-03 14:36:41.367 3284-3289/? I/zygote: JIT allocated 71KB for compiled code of void android.widget.TextView.<init>(android.content.Context, android.util.Attr
2019-03-03 14:36:41.479 3284-3284/? D/####: Retrofit error: HTTP 404 Not Found
2019-03-03 14:36:46.283 3284-3304/com.aucc.game V/FA: Inactivity, disconnecting from the service
2019-03-03 14:36:46.285 3284-3284/com.aucc.game V/FA: onUnbind called for intent. action: com.google.android.gms.measurement.START
2019-03-03 14:36:46.286 3284-3284/com.aucc.game V/FA: Local AppMeasurementService is shutting down
2019-03-03 14:36:46.634 2925-2962/com.android.calendar D/InitAlarmsService: Clearing and rescheduling alarms.
2019-03-03 14:36:46.637 2301-2325/system_process E/memtrack: Couldn't load memtrack module
2019-03-03 14:36:46.637 2301-2325/system_process W/android.os.Debug: failed to get memory consumption info: -1
2019-03-03 14:36:46.637 2301-2564/system_process I/ActivityManager: Killing 2948:com.android.contacts/u0a8 (adj 906): empty #17
2019-03-03 14:36:46.637 2301-2328/system_process W/zygote: kill(-2948, 9) failed: No such process
2019-03-03 14:36:46.647 2301-2325/system_process E/memtrack: Couldn't load memtrack module
2019-03-03 14:36:46.647 2301-2325/system_process W/android.os.Debug: failed to get memory consumption info: -1
2019-03-03 14:36:46.673 2301-2328/system_process W/zygote: kill(-2948, 9) failed: No such process
2019-03-03 14:36:46.673 2301-2328/system_process I/zygote: Successfully killed process cgroup uid 10008 pid 2948 in 35ms
2019-03-03 14:36:47.660 1406-1406/? I/boot-pipe: done populatinq /dev/random
```




~ cat android/logcat_usage.txt

Logging from code

The `Log` class allows you to create log entries in your code that display in the logcat tool. Common logging methods include:

- `Log.v(String, String)` (verbose)
- `Log.d(String, String)` (debug)
- `Log.i(String, String)` (information)
- `Log.w(String, String)` (warning)
- `Log.e(String, String)` (error)

For example, using the following call:

KOTLIN

JAVA

```
Log.i("MyActivity", "MyClass.getView() - get item number " + position);
```



The logcat outputs something like:

```
I/MyActivity( 1557): MyClass.getView() - get item number 1
```



~ cat android/adb.txt

Android Debug Bridge (adb), bir aygıtla iletişim kurmanıza olanak sağlayan çok yönlü bir komut satırı aracıdır. Adb komutu, uygulama yükleme ve hata ayıklama gibi çeşitli aygıt eylemlerini kolaylaştırır ve aygıtta çeşitli komutları çalıştırmak için kullanabileceğiniz bir Unix kabuğuna erişim sağlar. Üç bileşen içeren bir client-server programıdır.

```
+ x adb -e shell
→ ~ adb -e shell
generic_x86:/ $ su
generic_x86:/ # ls
acct          dev              init.usb.configfs.rc  sdcard
bugreports    etc              init.usb.rc           storage
cache         fstab.ranchu     init.zygote32.rc      sys
charger       fstab.ranchu.early mnt                    system
config        init             oem                   ueventd.ranchu.rc
d             init.envIRON.rc  proc                  ueventd.rc
data          init.ranchu.rc   root                  var
default.prop  init.rc          sbin                  vendor
generic_x86:/ # logcat
----- beginning of main
03-03 14:35:29.880 1337 1337 W auditd : type=2000 audit(0.0:1): initialized
```

- **Client:** Adb komutu yazılan makinede çalışır.
- **Daemon (Adbd):** Komut gönderilen Android cihazda çalışır. Komutları yerine getirir.
- **Server:** Client ile aynı makinede çalışır. Client ile daemon iletişimini sağlar.



~ cat android/resources.txt

- <https://developer.android.com>
- <https://www.slideshare.net/scentsome/introduction-to-android-studio>