

# Android Geliştirme | Hızlı Başlangıç

## A.Kurulum

Sırasıyla Kurmanız Gerekenler

1. JDK
2. Eclipse
3. Android SDK
4. ADT Eclipse Eklentisi

**Olası Problem:** JDK ve Eclipse, her ikisi de 32 bit veya 64 bit olmalı, birbirinden farklı olmamalı. Aksi halde Eclipse açılmayacaktır.

JDK: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Eclipse: <http://www.eclipse.org/downloads/>

Android SDK: <http://developer.android.com/sdk/index.html>

Eclipse ve Android SDK sıkıştırılmış halde hazır kurulu olarak gelmektedir. Tekrar kurulum ihtiyaçları yoktur. ADT eklentisi ise Eclipse içerisinden çevrimiçi olarak kurulmaktadır.

Eclipse ve Android SDK indikten sonra onları istediğimiz klasöre açıyoruz. Eclipse'i çalıştırıp projelerimizin kaydedileceği *workspace*'imizi seçtikten sonra *Help > Install New Software* yolunu takip ediyoruz. Karşımıza çıkan pencerede sol üst köşedeki *Add* butonuna tıklıyoruz. Karşımıza çıkan diyalogda ad kısmına *ADT Plugin*, URL kısmına ise <https://dl-ssl.google.com/android/eclipse/> ifadelerini yapıştırıyoruz. Bu aşamadan sonra talimatları takip etmek yeterli.

Eklentiye kurduktan sonra Eclipse yeniden başlatılacaktır, başlamaz ise kendiniz yeniden başlatın. Ardından *Window > Preferences > Android* menüsünü takip edin. *SDK Location* kısmından SDK'yı açtığınız yeri seçin ve onaylayın.

Eklentiye kurup, SDK'yı Eclipse'e gösterdikten sonra Eclipse'i kapatın. SDK'nın bulunduğu klasöre gidin ve *SDK Manager*'ı çalıştırın. *SDK Manager*'ın açılması biraz zaman alabilir. *SDK Manager* açıldıktan sonra var olan güncellemeleri kontrol edecektir, bittikten sonra tüm güncellemeleri listeleyecektir.

*Select*'in yanında link olarak bulunan *New* ve *Update*'e basıp yükleyebileceğimiz her şeyi seçip talimatları takip edip ve onaylayıp indirin. İndirme bittikten sonra Eclipse'ten *File > New > Project > Android > Android Application Project* yolundan yeni bir Android projesi başlatabilirsiniz.

## B.Uygulamaların Anatomisi

### Android Uygulamasının Öğeleri

- **src:** Yazacağımız tüm kodlar bu klasör içindeki paketlerimizin içerisinde bulunur.
- **gen:** Bu klasöre kesinlikle müdahale yapmıyoruz, bu klasör Android SDK tarafından projede her değişiklik yaptığımızda otomatik olarak yeniden oluşturulur. Bu klasörün içerisindeki kodların amacı XML dosyalarından oluşan formlarımız ve form elemanlarımız ile kodlarımız arasında köprü kurmaktır.
- **libs:** Dışarıdan ekleyeceğimiz kütüphaneleri ayarlar kısmından eklemeden önce bu klasörün içerisine atıyoruz ve oradan çağırıyoruz.
- **res:** Uygulama içerisindeki kod dışındaki tüm içeriğin bulunduğu klasördür, tablolar, görseller, metinler ve diğer her şey bu klasör içerisinde bulunur.

## C.İlk Uygulamamız

Daha önce bahsettiğimiz yoldan yeni bir uygulama açıyoruz. Projemizin ismi *ekranaismyazdirma* olsun, gördüğünüz gibi proje ismini değiştirdiğinizde alttaki kısımları otomatik olarak dolduruyor. Bir de SDK sürümü ayarlarını yapmamız gerekiyor.

- **Minimum Required SDK:** Uygulamamızın çalışmasını istediğimiz en düşük Android sürümü. Bunu ne kadar düşük tutarsanız o kadar fazla kullanıcıya ulaşırsınız ancak bunun bedeli de Android'in yeni özelliklerini kullanamamaktır. Şunu şöyle

açıklayabiliriz; elimizde 2 Android cihaz olsun, HTC Desire ve HTC Sensation. HTC Desire, Android 2.2'ye kadar güncelleme almıştır ve en fazla API 8'e kadar olan uygulamaları destekler. HTC Sensation ise Android 4.0'dır ve API 14'e kadar destek sunar. Elimizde API 8 ve API 14'te çalışan iki uygulama olduğunu düşünürsek Sensation her ikisini de çalıştıracakken, Desire yalnızca API 8 ile geliştirilen uygulamayı çalıştıracaktır. Şu anda uygun olan API 8'i seçmenizdir, bu ayar ile Android cihaz pazarının yüzde 95'ine ulaşabilirsiniz. Ancak gelecekte API 8'li cihazlar çok azalıp kullanıcı kitlesi azaldığında daha yüksek minimum API'lerde uygulama geliştirebilirsiniz.

- **Target SDK:** Uygulamanızın hitap ettiği API seviyesidir. Bu API cihazınızdan yüksek olsa bile Minimum Required SDK'da belirttiğiniz API seviyesinde çalışır, hedef API içerisinde, minimum API içerisinde olmayan bir özelliği kullanacak olursanız uygulamanızda hatalar çıkabilir. Şimdilik biz bunu da API 8 olarak seçiyoruz.
- **Compile With:** Uygulamanızın derleneceği API seviyesi. Uygulamanızda dışarıdan kütüphane kullanmayacaksanız bunu Target SDK ile eşit tutun. Eğer ki kullanacaksanız kullandığınız kütüphanelerin desteğine göre bu değeri vermeniz gerekir. Biz şu aşamada kullanmayacağımız için API 8 seçiyoruz.
- **Theme:** Uygulamamızın teması, bunu *None* yaparak boş olarak seçiyoruz.

Bu ayarlardan sonra elimizde şu görüntü olmalı:

Application Name:	<input type="text" value="ekranaisimyazdirma"/>
Project Name:	<input type="text" value="ekranaisimyazdirma"/>
Package Name:	<input type="text" value="com.example.ekranaisimyazdirma"/>
Minimum Required SDK:	API 8: Android 2.2 (Froyo) ▼
Target SDK:	API 8: Android 2.2 (Froyo) ▼
Compile With:	API 8: Android 2.2 (Froyo) ▼
Theme:	None ▼

**Olası Problem:** Bu uygulamayı örnek olsun diye yapıyoruz, eğer ki uygulamanızı Google Play Store'a koymak istiyorsanız Package Name'iniz com ifadesi ile başlamamalı. Aksi halde Play Store'a yüklerken bir takım sıkıntılarla karşılaşabilirsiniz. tr.gelistiriciadi.uygulamaadi veya uyg.sirketismi.uygulamaadi gibi anlamlı ifadeler kullanmak uygun olmalıdır.

Ayarları bitirince talimatları takip edip bir şey değiştirmeden ileri diyerek projenizi oluşturun. Karşınıza Hello world! Yazan boş bir telefon ekranı gelecektir. Eğer gelmediyse sol taraftan *ekranaisimyazdirma* > *res* > *layout* yolunu takip ederek içerisindeki XML

dosyasına çift tıklayın. Karşımıza tasarım ekranı geldi. Ekranın hemen solunda araçlarımız gözükecektir.

Şimdi *Hello world!* yazan *TextView*'e tıklayın, *Delete*'e basarak silin. Soldaki araç kutusundan *Text Fields* klasörünü açın ve üzerinde *abc* yazan *EditText*'i telefon ekranının sol üst köşesine sürükleyin. *EditText*'imiz yerleşmiş olmalı. Araç kutusunda *Form Widgets* klasörünü açın ve orada gördüğünüz *Button*'ı, yerleştirdiğimiz *EditText*'in hemen altına sürükleyin. Yerleştirdiğimiz *Button*'ın altına da araç kutusunda üzerinde *Large* yazan büyük *TextView*'i sürükleyin.

Telefon ekranının sağ üzerinde Android simgesi ve yanında 8 yazan bir seçenek göreceksiniz, o seçenek ile farklı Android sürümlerinde yaptıklarınızın nasıl gözükeceği hakkında bir fikir edinebilirsiniz, ancak bu görüntü asla cihazda birebir göreceğiniz görüntü olmayabilir.

Telefon ekranının hemen üzerinde de varsayılan olarak *Nexus One* yazan kısımdan da farklı ekran büyüklüklerinde uygulamamızın nasıl gözükeceği hakkında bir fikir edinebiliriz.

Telefon ekranının altındaki Graphical Layout ve XML dosyası tabından da oluşturmuş olduğumuz XML'in kaynağı ile görüntüsü arasında geçiş yapabiliriz.

Yaptıklarımız sonunda şöyle bir görüntü elde etmiş olmalıyız:

**Palette**

Palette

**Form Widgets**

TextView Large Medium Small

Button Small OFF

CheckBox RadioButton

CheckedTextView

Spinner  
Sub Item

QuickContactBadge

Text Fields

Layouts

Composite

Images & Media

Time & Date

Transitions

Advanced

Custom & Library Views

Graphical Layout activity\_main.xml

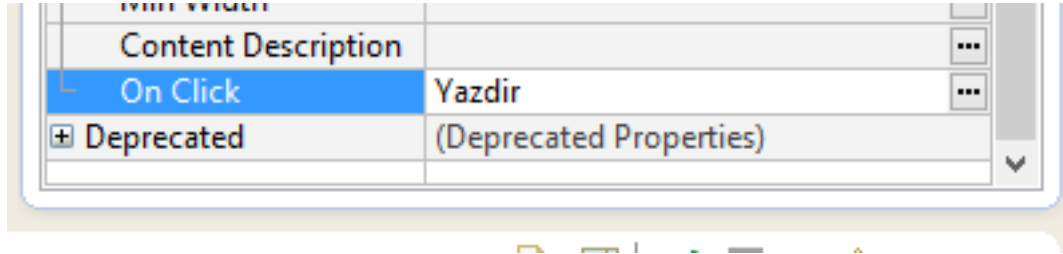
Nexus One AppTheme MainActivity 8

ekranaisimyazdirma

Button

Large Text

Eklediğimiz her bir araca tıkladığımızda varsayılan olarak sağda gelen *Properties*'teki ayarların değiştiğini görebiliriz. Şimdi butona tıklayınca ne olacağını ayarlamak için butona tıklayıp *Properties*'teki *OnClick* seçeneğini bulalım. İçerisine *Yazdir* yazalım. Aşağıdaki gibi görülmeli:



Artık grafik arayüzü ile işimiz bitti. src klasörümüzün içindeki paketlerde bulunan MainActivity sınıfımızı açıyoruz. Buradaki kodları aşağıdaki gibi göreceksiniz:

```
package com.example.ekranaisimyazdirma;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```



```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.activity_main, menu);
    return true;
}
}
```

Bu kodlar projemizi oluştururken SDK tarafından oluşturulan kodlar. Bu uygulamada telefon üzerindeki *Menü* tuşunu kullanmayacağımız için *onCreateOptionsMenu* sınıfını silmemizde bir sakınca yok. Bu fonksiyon *Menü* tuşuna basınca olucakları programlamamıza yarıyor. Onu silince kodlarımız şöyle olacaktır:

```
package com.example.ekranaisimyazdirma;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

İlk olarak eklediğimiz nesnelerden *TextView*'i ve *EditText*'i tanımlamamız gerekiyor, *Button*'a gerek yok, çünkü zaten onun hangi fonksiyonu çalıştıracağını tanımlamıştık. Tanımladıktan sonra kodlarımız şu hali alacaktır:

```
package com.example.ekranaisimyazdirma;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends Activity {
    private EditText editText;
    private TextView textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = (EditText) findViewById(R.id.editText1);
        textView = (TextView) findViewById(R.id.textView1);
    }
}
```

Gördüğünüz gibi *EditText* ve *TextView*'imizi tanımladık ve XML'deki hangilerinin bizim tanımladıklarımız olduğunu gösterdik. XML dosyasına gidecek olursanız *Properties* bölümünden eklediğimiz form araçlarının *id*'lerinin ne olduğunu görebilir ve değiştirebilirsiniz.

**Olası Problem:** XML'deki değişikliklerden sonra dosyanızı kaydetmelisiniz. Aksi halde kodlardaki form araçlarının tanımlanırken XML'de olması gereken id'leri henüz kaydetmediğiniz için bulamayacak ve kodu yazarken otomatik tamamlamayacaktır.

Şimdi *Button*'umuzun görevini programlayacağız. *Button*'umuzun çağırdığı fonksiyonu yazdıktan sonra kodlarımız şu şekilde olacaktır:

```
package com.example.ekranaisimyazdirma;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
public class MainActivity extends Activity {
    private EditText _editText;
    private TextView _textView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        _editText = (EditText) findViewById(R.id.editText1);
        _textView = (TextView) findViewById(R.id.textView1);
    }
    public void Yazdir(final View v) {
        _textView.setText(_editText.getText());
    }
}
```

Şimdi kodlarımızı çalıştırma vakti. Android telefonunuz var ise *Ayarlar > Uygulamalar* bölümünden bilinmeyen kaynaklardan gelen uygulamalara izin vermeli ve yine *Ayarlar > Uygulamalar > Geliştirme* bölümünden USB hata ayıklama seçeneğini aktif hale getirmelisiniz. Bu ayarlar farklı Android sürümlerinde farklı menü yolları içerisinde olabilir.

Bunu yaptıktan sonra telefonunuzu USB ile bilgisayara bağladığınızda sürücülerin yüklenmesini bekleyin. Olmuyorsa *Aygıt Yöneticisi'*nden sürücü yüklerken sürücü konumunu *Android SDK > extras > google > usb\_driver* olarak gösterin. Yine de olmuyorsa telefon üreticisinin yazılımını kurmanız gerekmektedir. Örnek olarak HTC cihazlar için *HTC Sync* yazılımını kurmalısınız.

Kodları *Run* butonuna basarak veya *Ctrl + F11* ile çalıştırabilirsiniz. İlk çalıştırmada uygulamanızın ne olduğunu soracaktır. *Android Application* seçeneğini işaretleyip ilerleyin. Cihaz bulunamadığına dair bir soru gelecektir, soruda hayır butonuna basın. Eğer ki cihazınız var ve doğru kurulmuş ise üstteki *Running Devices* kısmından seçin ve programınız cihazınızda çalışsın.

Cihazınız yok ise alttaki tabloyu seçin ve *Manager* butonuna tıklayın. Gelen pencerede *Device Definitions* tabına gelin ve bir cihaz seçin. Seçtikten sonra *Create AVD* diyerek ön tanımlı ayarları aynen kullanın. Ardından *Start* butonuna basıp emülatörü başlatın. Bu işlem dakikalar sürebilir. Emülatörü bir kere başlattıktan sonra her derlemede tekrar tekrar başlatmak zorunda değilsiniz. Emülatör açık dursun, uygulamanızdaki değişiklikleri halledin

ve *Run* butonuna basın. Ardından *Running Devices* bölümünden oluşturduğunuz emülatörü seçin. Tebrikler, ilk Android uygulamanızı oluşturdunuz.

Sadece buraya kadar olan bilgilerle hesap makinesi, birim çevirici ve buna benzer birçok uygulama yapabilirsiniz.

**Mustafa Tufan**  
Fırat Üniversitesi  
Teknoloji Fakültesi  
Yazılım Mühendisliği