# ANDROID - Ključni pojmovi

1. Widget

Kod Androida, widget je jedan element korisničkog sučelja. Polja, gumbi, oznake, popisi, padajući meniji svi se zovu widgeti. Korisničko sučelje aktivnosti sastoji se od jednog ili više widgeta.

1. Containers/ Layout Managers

Da bi se widgeti organizirani na zaslonu, koriste se različite klase spremnika koje se nazivaju „Layout Managers“. Oni omogućuju da se elementi stave u retke, stupce ili složenije rasporede prema potrebi. Zato je potrebno stvoriti datoteku resursa izgleda (activity\_main.xml) i staviti u mapu resursa projekta odakle ga Android preuzima izravno i automatski generira cijelo korisničko sučelje.

1. Resursi

Resursi u androidu odnose se na slike, animacije, zvukove i slično. U programiranju za android potrebno je ponekad puno takvih datoteka resursa.

1. Fragmenti  
   Dizajn aplikacije mora imati prilagodljivo korisničko sučelje odnosno mora raditi na svim vrstama uređaja: mobitelima, tabletima i televizorima. Na primjer, Gmail aplikacija na tabletu prikazat će vaš popis oznaka, popis razgovora u odabranom oznaka i popis poruka u odabranom razgovoru, sve u jednoj aktivnosti (zaslonu) na tabletu. Međutim, ista Gmail aplikacija na telefonu to ne može učiniti jer nema dovoljno prostora na ekranu, pa svaku od njih (oznake, razgovore, poruke) prikazuje u zasebnim aktivnostima (ekranima). Android isporučuje konstrukciju zvanu fragment kako bi olakšala implementaciju ovakvih efekata.
2. APK

Aplikacija dolazi u obliku APK datoteke. To je APK datoteka koju ćete prenijeti u Play Store ili distribuirati na neki drugi način. Važno je naučiti da svaka android aplikacija ima jedinstveni naziv paketa i da mora ispunjavati tri zahtjeva: a) To mora biti valjani naziv java paketa, jer će neki java izvorni kod generirati alate za izgradnju androida u ovom paketu. B) Dvije aplikacije ne mogu postojati na uređaju u isto vrijeme s istim paketom. 3) U Trgovini Play ne mogu se prenijeti dvije aplikacije koje imaju isti paket.

1. Android version i API level

Kako bi lakše pratili različite verzije operacijskog sustava (OS) koje su važne razvojnim programerima, Android ima razine API-ja. Nova razina API-ja definira se kada dođe verzija Androida koja sadrži promjene koje utječu na aplikacije koje razvijaju razvojni programeri. Kada se pokrene AVD emulator (vrsta Virtual Machine) za testiranje aplikacije, potrebno je naznačite koju razinu API-ja taj emulator treba emulirati. Kod distribucije aplikacije, naznačiti najstariju razinu API-ja koju aplikacija podržava, tako da aplikacija ne bude instalirana na starijim uređajima.

1. Glavne komponente ANDROIDA

* Aktivnosti, servisi, prijemnici poruka (broadcast receivers), dobavljači sadržaja (content resolver), pružatelji sadržaja (content providers)
* Detaljnije opisano u android 13

1. Android Manifest

Datoteka AndroidManifest.xml sadrži informacije o vašem paketu, uključujući komponente aplikacije kao što su aktivnosti, usluge, prijamnici emitiranja, davatelji sadržaja itd., određuje dozvole pristupa, prijavljuje koji API će aplikacija koristiti, pridružuje ikonu aplikaciji, upravlja cijelim paketom resursa i java programa.

# PRIMJER WEB servisa

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**ScrollView 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"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 tools:context=".MainActivity"**>  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:gravity="center"**>  
  
  
 <**Button  
 android:id="@+id/exit"  
 android:layout\_width="100dp"  
 android:layout\_height="50dp"  
 android:text="Izlaz"** />  
  
  
 <**TextView  
 android:id="@+id/txt\_temperature"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Dobivanje temperature..."**/>  
  
 </**LinearLayout**>  
  
</**ScrollView**>

MainActivity.java

**package** com.example.myapplication17;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.os.Bundle;  
**import** android.util.Log;  
  
**import** android.widget.Button;  
**import** android.widget.TextView;  
  
**import** java.io.BufferedReader;  
**import** java.io.InputStreamReader;  
**import** java.net.URL;  
  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 String **inputLine**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 getTemperature();  
  
  
  
 Button btnExit = findViewById(R.id.***exit***);  
 btnExit.setOnClickListener(v -> MainActivity.**this**.finish());  
  
  
 }  
  
 **void** getTemperature() {  
  
 **new** Thread(() -> {  
 **try** {  
  
 URL yahoo = **new** URL(**"https://vrijeme.hr/hrvatska\_n.xml"**);  
 BufferedReader in = **new** BufferedReader(  
 **new** InputStreamReader(  
 yahoo.openStream()));  
  
  
 **boolean** currently = **false**;  
  
 **while** ((**inputLine** = in.readLine()) != **null**) {  
 **if**(**inputLine**.equals(**"<GradIme>Rijeka</GradIme>"**)) {  
 currently = **true**;  
 **continue**;  
 }  
  
 **if**(currently && **inputLine**.contains(**"Temp"**)) {  
  
 MainActivity.**this**.runOnUiThread(() -> {  
 TextView textView = findViewById(R.id.***txt\_temperature***);  
String string1=**"Temperatura u Rijeci: "** +  
 **inputLine**.replace(**"<Temp>"**, **""**).replace(**"</Temp>"**, **""**).trim() + **" °C"**;  
 textView.setText(string1);  
 });  
  
 **break**;  
 }  
 }  
  
 in.close();  
 } **catch** (Exception e) {  
 Log.*e*(**"XML\_Error"**, **"XML Pasing Exeption = "** + e);  
  
 MainActivity.**this**.runOnUiThread(() -> {  
 TextView textView = findViewById(R.id.***txt\_temperature***);  
 String string2=**"Greška u dobivanju temperature!"**;  
 textView.setText(string2);  
 });  
 }  
  
 }).start();  
  
  
 }  
}

*AndroidManifest*

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
  
 package="com.example.myapplication17"**>  
<**uses-permission android:name="android.permission.INTERNET"**/>  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MyApplication17"**>  
 <**activity  
 android:name=".MainActivity"  
 android:exported="true"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

# Objavljivanje Android aplikacija

Kako bi aplikacija bila dostupna krajnjim korisnicima potrebno je proći dvije faze:

* priprema za puštanje u produkciju, aplikacija gradi (eng. *build* ) konačnu verziju
* puštanje u produkciju, aplikacija se javno objavljuje uz nadoplatu ili besplatno.

Postupak pripreme za puštanje u produkciju:

* prikupiti materijale za objavu

(pripremiti ugovor s korisnikom, generirati kriptografske ključeve, stvoriti početnu ikonu)

* konfigurirati aplikaciju za objavu

(čišćenje projekata od postavki zaostalih od prethodnih pokretanja i debugiranja, postavljanje postavki u manifest)

* izgraditi aplikaciju za objavu

(buildanje i potpisivanje aplikacije)

* pripremiti servere

(priprema servera za produkciju)

* testirati aplikaciju

(provjeriti ispravnost rada aplikacije)

1. **kriptografski ključ –** Android sustavi zahtijevaju da svaka aplikacija bude digitalno potpisana, kreiranje javnog certifikata za koji razvojni programer ima privatni ključ. Valjanost tim kriptografskim ključevima istječe nakon 22. listopada 2033. godine.
2. **Ikona** – predstavlja našu aplikaciju korisniku. Za objavljivanje aplikacije na Google Playu potrebno je izraditi ikonu u visokoj rezoluciji (32-bitni PNG dimenzija 512 x 512 px i max veličine 1024 KB.
3. **End-User Licence Agreement(EULA) –** Ovakav ugovor pomaže da se autor aplikacije zaštiti i predstavlja intelektualno vlasništvo autora.
4. **Ostalo –** opis aplikacije

**Konfiguriranje aplikacije**

1. **Odabir prikladnog imena aplikacije-** ime razvojne aplikacije mora biti u skladu s njenim sadržajem. Naknadna izmjena imena nije moguća.
2. **Isključiti logove i debugiranje –** brisanje poziva log metode u kodu i postavljanje android:debuggable na false pod application tagom u manifestu. Izbrisati i log datoteke ako smo ih kreirali.
3. **Očistiti projekt od suvišnih datoteka.**
4. **Pregledati manifest i Gradle postavke –** treba postaviti sve potrebne <user-permissions> te uklonit suvišne, ispravno postaviti android:label i android:icon vrijednosti te ispravno podesiti android:version>Code i android: versionName, te ograničenja za koje smo SDK verzije aplikaciju namijenili i testirali, android:minSdkVersion i android:targetSdkVersion
5. **Provjera URL adresa**
6. **Implementacija Licence**

**Otvaranje računa – 25$**

[**https://play.google.com/console/signup/playSignup**](https://play.google.com/console/signup/playSignup)