

GUI I

Mobilne aplikacije

Stevan Gostojić

Fakultet tehničkih nauka, Novi Sad

25. oktobar 2022.

Pregled sadržaja

- 1 Resursi i konfiguracije uređaja
- 2 Grafički korisnički interfejs
- 3 Pogledi
- 4 Rasporedi
- 5 Stilovi i teme

Resursi i konfiguracije uređaja

- Android aplikacija je skup slabo povezanih komponenti
- Komponente pored klasa mogu da sadrže i resurse (deklaracije GUI, tekst, rastersku i vektorsku grafiku, audio i video klipove, itd.)
- Resurse treba eksternalizovati da bi se omogućilo prilagođavanje aplikacije različitim konfiguracijama uređaja (dimenzije, rezolucija i orijentacija ekrana, jezik i region, itd.) i lakša sinhronizacija između programera i grafičkih dizajnera.

Resursi

```
1 ExampleApp/  
2   app/  
3     src/  
4       main/  
5         java/  
6           ExampleActivity.java  
7         res/  
8           drawable/  
9             icon.png  
10          layout/  
11            main.xml  
12          raw/  
13          values/  
14            arrays.xml  
15            colors.xml  
16            strings.xml  
17            styles.xml  
18          xml/  
19            AndroidManifest.xml  
20        build/  
21          generated/  
22          source/  
23            r/  
24              R.java  
25
```

Resursi

Tip	Opis
anim	animacije
drawable	vektorska ili rasterska grafika
layout	deklaracije grafičkog korisničkog interfejsa
raw	"sirovi" podaci (audio i video klipovi)
values	proste vrednosti (nizovi, boje, eksteralizovani stringovi, stilovi, itd.)
xml	XML dokumenti

Table 1: Tipovi resursa.

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="vertical">
5
6     <TextView
7         android:id="@+id/text"
8         android:text="Hello World!"/ >
9
10 </LinearLayout>
11
```

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="vertical">
5
6     <TextView
7         android:id="@+id/text"
8         android:text="@string/hello_world" />
9
10 </LinearLayout>
11
```

strings.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="hello_world">Hello World!</string>
4 </resources>
5
```


Resursi

- Svaki resurs identifikovan je nazivom i tipom
- Android generiše jedinstveni identifikator svakog resura (nalazi se u R klasi)
- Resursima se može pristupiti iz Java koda (`R.layout.main`, `R.string.hello__world`) ili iz XML koda (`@layout/main`, `@string/hello__world`)

Konfiguracije uređaja

- Postoji veliki broj uređaja (sa različitim hardverskim karakteristikama) koji koriste Android platformu i veliki broj verzija Android platforme
- Konfiguracija uređaja apstrahuju njegove hardverske i softverske karakteristike
- Resursi se mogu definisati za različite konfiguracije uređaja (npr. ekran niske, srednje i visoke rezolucije)
- Različitim konfiguracijama uređaja odgovaraju resursi koji se nalaze u direktorijumima sa različitim sufiksima (ldpi, mdpi, hdpi)
- Moguće je istovremeno definisati resurse za više tipova konfiguracija (ekran visoke rezolucije u noćnom modu)

Konfiguracija uređaja

Tip	Vrednosti
language and region	en, fr, en-rUS, fr-rFR, fr-rCA, itd.
screen size	small, normal, large, xlarge
screen orientation	port, land
screen pixel density	ldpi, mdpi, hdpi, xhdpi, xxhdpi, xxxhdpi, nodpi, tvdpi
UI mode	car, desk, television, appliance, watch
touchscreen type	notouch, finger
night mode	night, notnight
platform version (API level)	v1, v2, v3, itd.

Table 2: Tipovi konfiguracije uređaja.

Konfiguracija uređaja

```
1 ExampleApp/  
2   res/  
3     drawable/  
4       icon.png  
5     drawable-ldpi/  
6       icon.png  
7     drawable-mdpi/  
8       icon.png  
9     drawable-hdpi/  
10      icon.png  
11     drawable-night-hdpi/  
12      icon.png  
13
```

Konfiguracija uređaja

- Aplikacija bi uvek trebalo da sadrži podrazumevane resurse
- Za različite rezolucije ekrana bi trebalo pripremiti slike različitih rezolucija
- Za različite veličine ili orijentacije ekrana bi trebalo pripremiti različite rasporede GUI-a
- Treba internacionalizovati stringove da bi se omogućila lokalizacija aplikacije na različite jezike

Pregled sadržaja

- 1 Resursi i konfiguracije uređaja
- 2 Grafički korisnički interfejs**
- 3 Pogledi
- 4 Rasporedi
- 5 Stilovi i teme

GUI

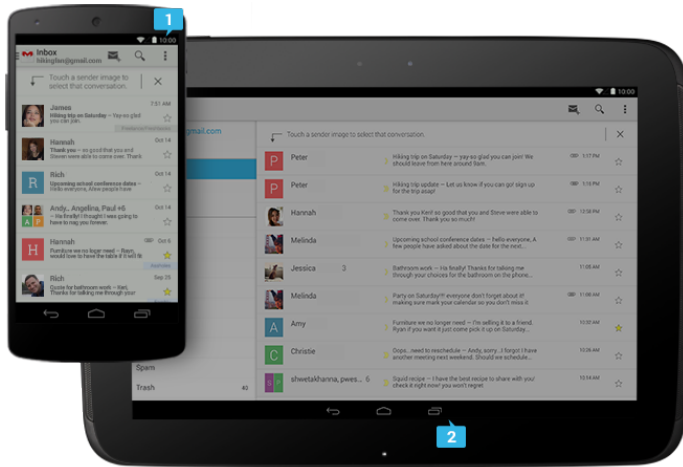


Figure 1: Statusna linija i navigaciona linija.

GUI

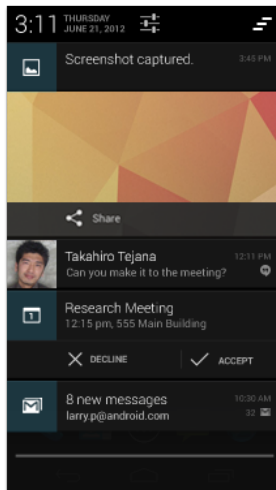


Figure 2: Obaveštenja.

- Home Screen
- All Apps
- Overview Space (Recents Screen)
- Notifications
- App Screen

GUI

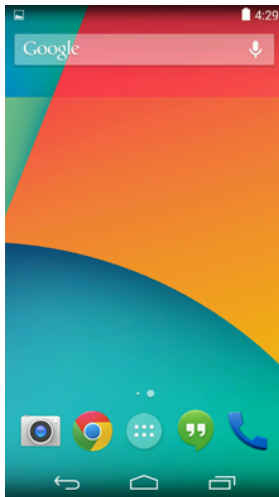


Figure 3: Home Screen.

GUI

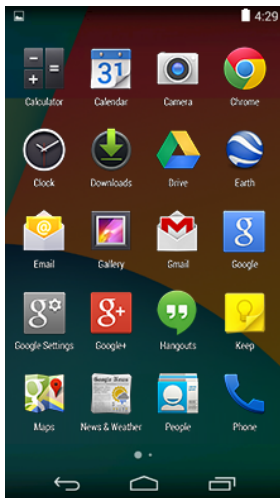


Figure 4: All Apps.

GUI

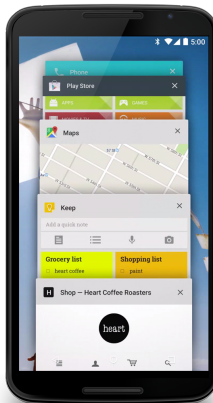


Figure 5: Overview Space

GUI

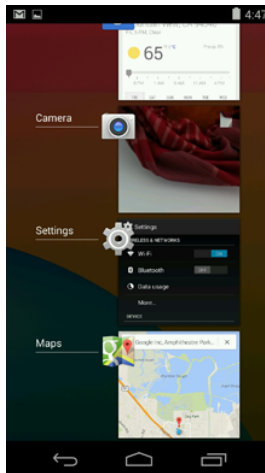


Figure 6: Recents Screen.

GUI

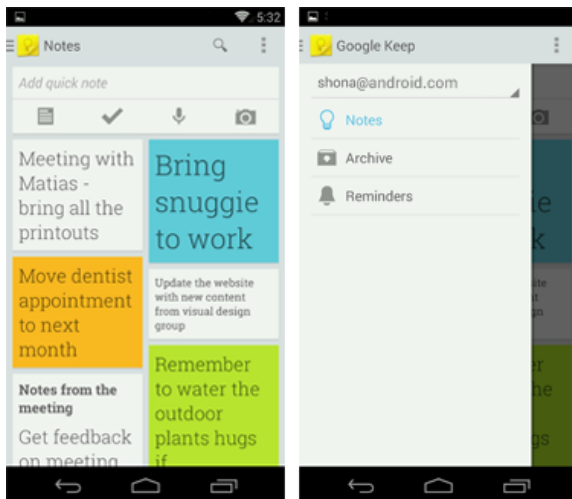


Figure 7: Toolbar, Navigation Drawer & Content Area.

GUI

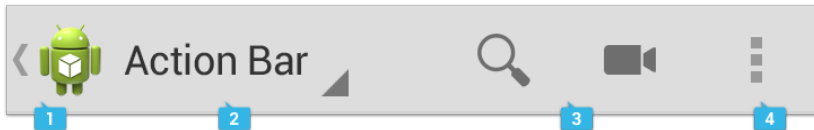


Figure 8: Toobar (App Bar).

- ❶ app icon
- ❷ view control
- ❸ action buttons
- ❹ action overflow

- Touch
- Long press
- Swipe/Drag
- Long press drag
- Double touch
- Double touch drag
- Pinch open
- Pinch close

Navigacija

- Modelovati podatke (npr. dijagram klasa, dijagram entiteta i veza, itd.)
- Skicirati korisnički interfejs kao skup ekrana i njihovih veza (npr. koristeći mock-up alat)
- Implementirati korisnički interfejs

Navigacija

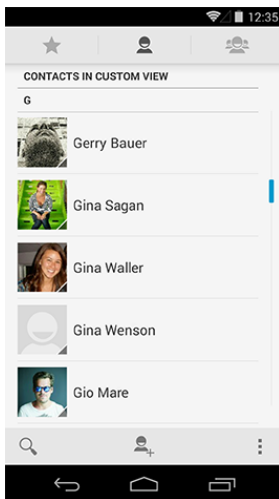


Figure 9: Kolekcija entiteta.

Navigacija

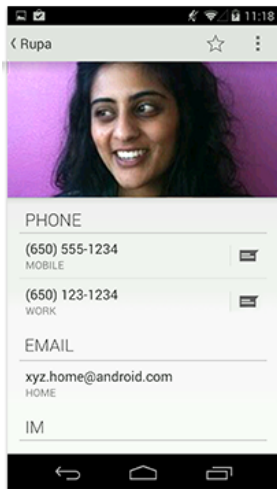


Figure 10: Pojedinačni entitet.

Navigacija

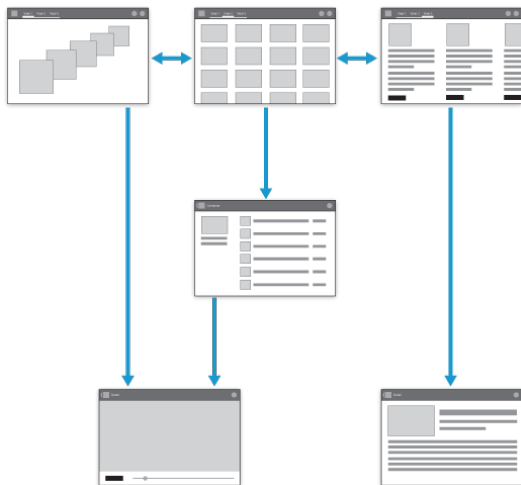
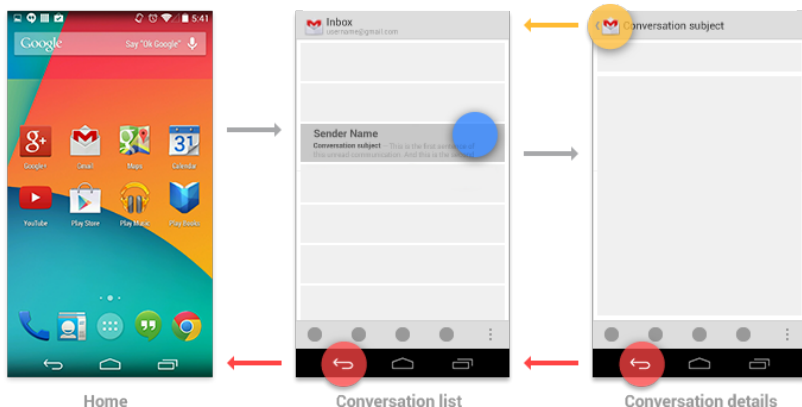
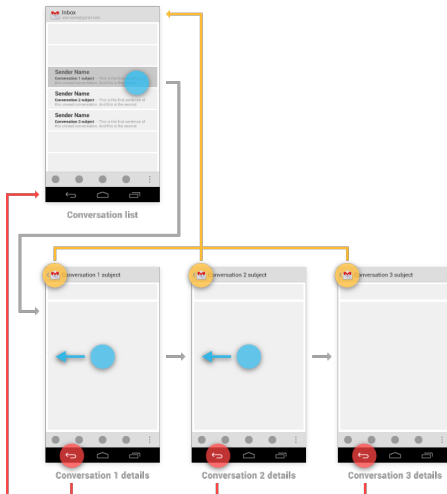


Figure 11: Struktura GUI-a.

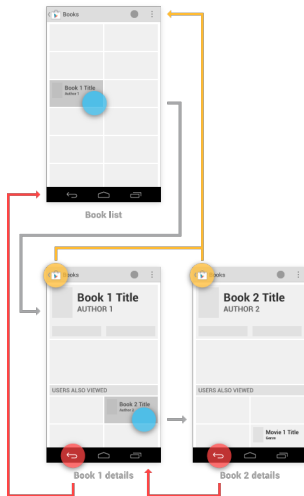
Navigacija



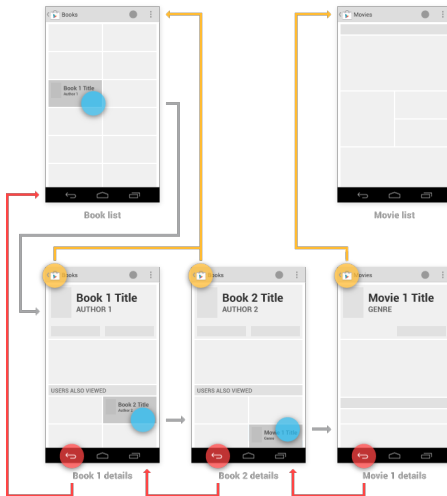
Navigacija



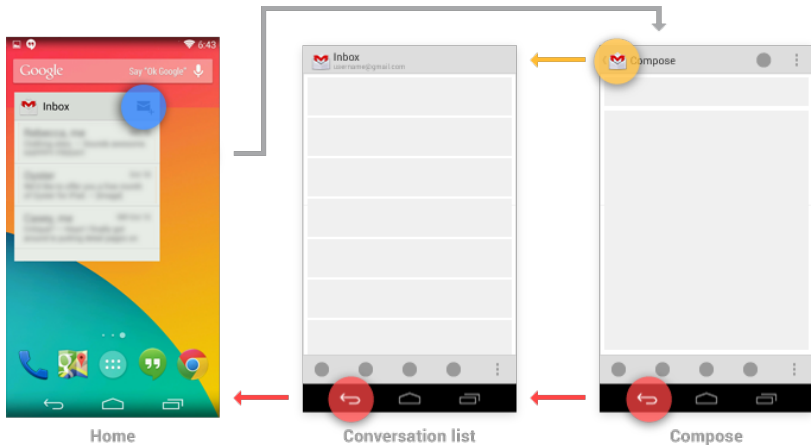
Navigacija



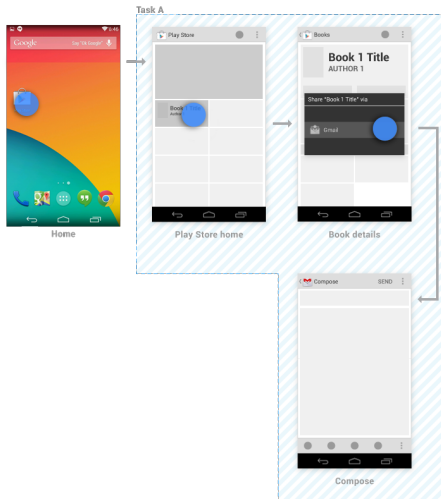
Navigacija



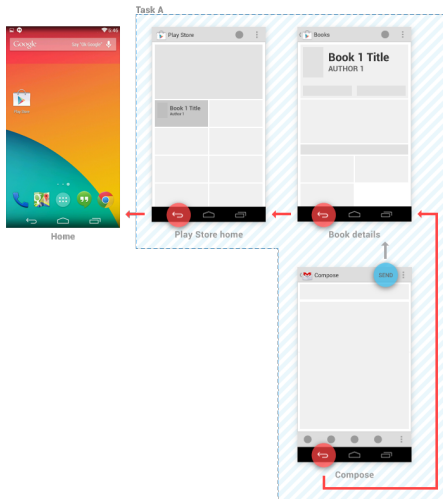
Navigacija



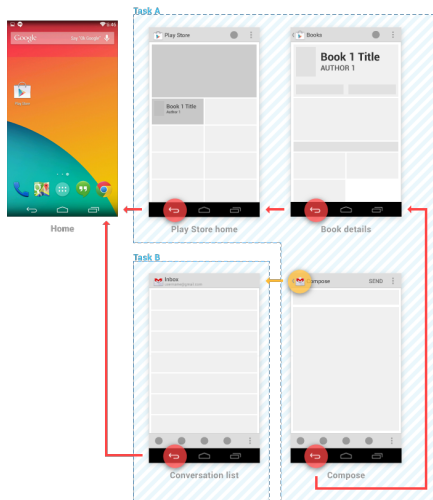
Navigacija



Navigacija



Navigacija



Pregled sadržaja

- 1 Resursi i konfiguracije uređaja
- 2 Grafički korisnički interfejs
- 3 **Pogledi**
- 4 Rasporedi
- 5 Stilovi i teme

Pogledi i rasporedi

- Grafički korisnički interfejs bilo koje aktivnosti može se predstaviti hijerarhijom pogleda (view) i rasporeda (layout)
- Pogledi predstavljaju elemente GUI-a
- Rasporedi su pogledi koji sadrže druge poglede i određuju kako se oni raspoređuju na ekranu

Pogledi i rasporedi

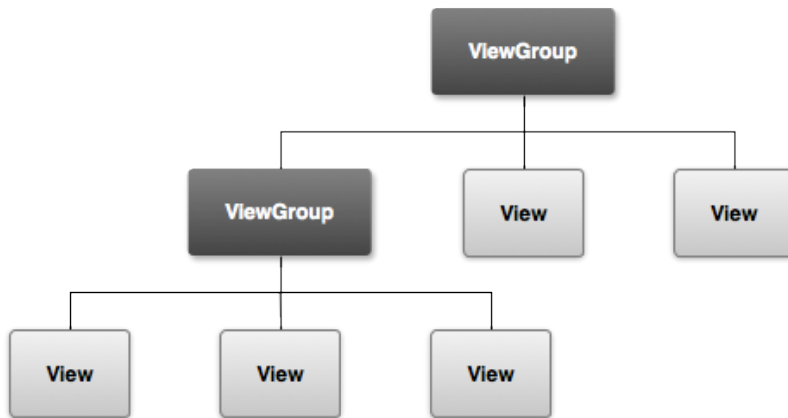


Figure 12: Hijerarhija pogleda i rasporeda GUI-a.

Pogledi

- Elementi Android GUI-a su pogledi
- Android sadrži predefinisane tipove pogleda (npr. labela, tekstualna polja, dugmad, itd.)
- Moguće je definisati nove tipove pogleda (mada se obično koriste predefinisani)

Svojstva pogleda

- Stanje pogleda određeno je njegovim svojstvima
- Postoje svojstva koja su zajednička za sve tipove pogleda (npr. vidljivost, transparentnost, itd.), a neki tipovi pogleda mogu da sadrže i posebna svojstva

Svojstva pogleda

Svojstvo	Opis
id	supply an identifier name for this view
clickable	defines whether this view reacts to click events
focusable	boolean that controls whether a view can take focus
visibility	controls the initial visibility of the view
alpha	controls the initial transparency of the view

Table 3: Važnija svojstva pogleda.

Događaji

- Pogledi mogu obrađivati različite događaje (dodir, pritisak tastera, promenu fokusa, itd.)

Pogledi

Događaj	Opis
onTouch()	the user performs an action qualified as a touch event)
onClick()	the user touches the item
onLongClick()	the user either touches and holds the item
onFocusChange()	the user navigates onto or away from the item
onKey()	the user is focused on the item and presses or releases a hardware key on the device
onCreateContextMenu()	this is called when a Context Menu is being built

Table 4: Važniji događaji.

Pravljenje pogleda

- Pogledi se mogu definisati instanciranjem objekata u Java kodu ili dodavanjem elemenata u XML kodu
- Na sličan način mogu se postaviti svojstva i obrađivači događaja pogleda

Pravljenje pogleda

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {  
2     protected void onCreate(Bundle state) {  
3  
4         Button button = new Button(this);  
5         LinearLayout layout = new LinearLayout(this);  
6         layout.addView(button);  
7         setContentView(layout);  
8     }  
9 }  
10
```

Pravljenje pogleda

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:id="@+id/linear_layout"
4   android:orientation="vertical">
5
6   <Button android:id="@+id/button" />
7
8 </LinearLayout>
9
```

Pravljenje pogleda

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {  
2     protected void onCreate(Bundle state) {  
3  
4         setContentView(R.layout.main);  
5     }  
6 }  
7
```


Postavljanje svojstava

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {
2     protected void onCreate(Bundle state) {
3
4         Button button = (Button) findViewById(R.id.button);
5         button.setLayoutParams(new LayoutParams(
6             ViewGroup.LayoutParams.WRAP_CONTENT,
7             ViewGroup.LayoutParams.WRAP_CONTENT));
8         button.setText(R.string.button_text);
9
10        LinearLayout layout = new LinearLayout(this);
11        layout.addView(button);
12
13        setContentView(layout);
14    }
15 }
16
```

Postavljanje svojstava

layout.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:id="@+id/linear_layout"
4   android:orientation="vertical">
5
6   <Button
7     android:id="@+id/button"
8     android:layout_width="wrap_content"
9     android:layout_height="wrap_content"
10    android:text="@string/button_text" />
11
12 </LinearLayout>
13
```

Pravljenje pogleda

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {  
2     protected void onCreate(Bundle state) {  
3  
4         setContentView(R.layout.main);  
5     }  
6 }  
7
```

Obrada događaja

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {
2     protected void onCreate(Bundle bundle) {
3
4         Button button = new Button(this);
5         button.setOnClickListener(new View.OnClickListener() {
6             public void onClick(View v) {
7                 Toast toast = Toast.makeText(this, "Button clicked!", Toast.
LENGTH_SHORT);
8                 toast.show();
9             }
10        }
11
12        LinearLayout layout = new LinearLayout(this);
13        layout.addView(button);
14
15        setContentView(layout);
16    }
17 }
18
```

Obrada događaja

layout.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:id="@+id/linear_layout"
4   android:orientation="vertical">
5
6   <Button
7     android:id="@+id/button"
8     android:layout_width="wrap_content"
9     android:layout_height="wrap_content"
10    android:text="@string/button_text"
11    android:onClick="click" />
12
13 </LinearLayout>
14
```

Pravljenje pogleda

ExampleActivity.java

```
1 public class ExampleActivity extends Activity {  
2  
3     protected void onCreate(Bundle state) {  
4         setContentView(R.layout.main);  
5     }  
6  
7     public void click(View view) {  
8         Toast toast = Toast.makeText(this, "Button clicked!", Toast.  
9             LENGTH_SHORT);  
10        toast.show();  
11    }  
12 }
```

Tipovi pogleda

- TextView
- ImageView
- EditText
- Button
- RadioButton
- ToggleButton
- Checkbox

TextView

Hello Android!

Figure 13: TextView.

- Pogled TextView prikazuje tekst i omogućava njegovo kopiranje

TextView

```
1 <TextView
2   android:id="@+id/email_address"
3   android:layout_width="fill_parent"
4   android:layout_height="wrap_content"
5   android:text="@string/hello_world" />
6
```

ImageView



Figure 14: ImageView.

- Pogled ImageView prikazuje proizvoljnu sliku iz različitih izvora
- Omogućava i skaliranje, odsecanje, primenu filtera, itd.

ImageView

```
1 <ImageView
2   android:id="@+id/icon"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:src="@drawable/my_image" />
6
```

EditText

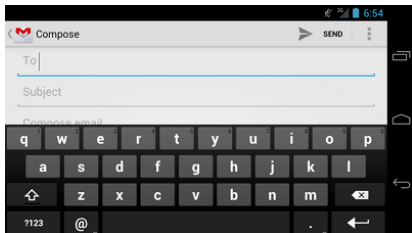


Figure 15: EditText.

- Pogled EditText omogućava unos teksta
- Pored unosa teksta, omogućava i niz drugih aktivnosti kao što su označavanje, isecanje, kopiranje, itd.
- Moguće je specificirati tip tastature (normalna, numerička, telefonska, itd.) ili ponašanje tastature (automatsko pretvaranje početnih slova reči u velika slova, itd.)

EditText

```
1 <EditText
2   android:id="@+id/email_address"
3   android:layout_width="fill_parent"
4   android:layout_height="wrap_content"
5   android:hint="@string/password_hint"
6   android:inputType="text | textPassword" />
7
```

EditText

Vrednost	Opis
text	normalna tastatura
textEmailAddress	normalna tastatura sa @ znakom
textUri	normalna tastatura sa / znakom
number	osnovna numerička tastatura
phone	telefonska tastatura

Table 5: Vrednosti inputType atributa.

EditText

Vrednost	Opis
textCapWords	normalna tastatura koja automatski pretvara početno slovo rečenice u veliko slovo
textAutoCorrect	normalna tastatura koja ispravlja česte pravopisne greške
textPassword	normalna tastatura koja unesene znakove prikazuje kao tačke
textMultiLine	normalna tastatura koja omogućava korisnicima da unose tekst u više redova

Table 6: Vrednosti inputType atributa.

Button



Figure 16: Button.

- Pogled Button prikazuje tekst ili sliku koja simbolizuju akciju
- Kada korisnik pritisne dugme generiše se click događaj
- Metoda koja obrađuje ovaj događaj specificira se onClick atributom i mora biti sadržana u aktivnosti kojoj je dugme pridruženo

Button

```
1 <Button
2     android:id="@+id/button_id"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:text="@string/button_text"
6     android:drawableLeft="@drawable/button_icon"
7     android:onClick="alarm" />
8
```

Button

```
1  /* Called when the user touches the button */
2  public void alarm(View view) {
3      Uri notification = RingtoneManager.getDefaultUri(
4          RingtoneManager.TYPE_NOTIFICATION);
5      Ringtone r = RingtoneManager.getRingtone(
6          getApplicationContext(), notification);
7      r.play();
8  }
```

RadioButton

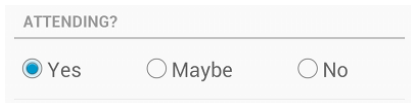


Figure 17: RadioButton.

- Pogled `RadioButton` omogućava korisniku da izabere jednu opciju iz skupa više opcija
- Svaka opcija predstavljena je objektom klase `RadioButton` koji su grupisani objektom klase `RadioGroup`

RadioButton

```
1 <RadioButton
2     android:id="@+id/yes"
3     android:layout_width="wrap_content"
4     android:layout_height="wrap_content"
5     android:text="@string/yes"
6     android:onClick="onRadioButtonClicked"/ >
7
```

RadioButton

```
1 <RadioGroup
2     android:layout_width="fill_parent"
3     android:layout_height="wrap_content"
4     android:orientation="vertical">
5
6     <RadioButton ... />
7     <RadioButton ... />
8
9 </RadioGroup>
10
```

RadioButton

```
1 public void onRadioButtonClicked(View view) {  
2     // Is the button now checked?  
3     boolean checked = ((RadioButton) view).isChecked();  
4     // Check which radio button was clicked  
5     switch(view.getId()) {  
6         case R.id.yes:  
7             if (checked)  
8                 // ...  
9                 break;  
10        case R.id.maybe:  
11            if (checked)  
12                // ...  
13                break;  
14        }  
15    }  
16 }
```

ToggleButton



Figure 18: ToggleButton.

- Pogled ToggleButton omogućava korisniku da promeni podešavanje između dva stanja

ToggleButton

```
1 <ToggleButton
2   android:id="@+id/togglebutton"
3   android:layout_width="wrap_content"
4   android:layout_height="wrap_content"
5   android:textOn="Vibrate on"
6   android:textOff="Vibrate off"
7   android:onClick="onToggleClicked"/ >
8
```


ToggleButton

```
1 public void onToggleClicked(View view) {  
2     // Is the toggle on?  
3     boolean on = ((ToggleButton) view).isChecked();  
4     if (on) {  
5         // Enable vibrate  
6     } else {  
7         // Disable vibrate  
8     }  
9 }  
10
```

CheckBox

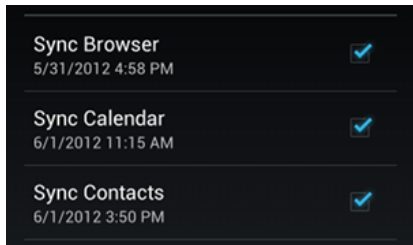


Figure 19: CheckBox.

- Pogled CheckBox omogućava korisniku da izabere jednu ili više opcija iz skupa opcija
- Opcije se obično prikazuju u vertikalnoj listi.

CheckBox

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/
   apk/res/android"
3     android:orientation="vertical"
4     android:layout_width="fill_parent"
5     android:layout_height="fill_parent">
6
7     <CheckBox
8         android:id="@+id/checkbox_browser"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        android:text="@string/meat"
12        android:onClick="onCheckboxClicked" />
13
14    <CheckBox
15        android:id="@+id/checkbox_calendar"
16        android:layout_width="wrap_content"
17        android:layout_height="wrap_content"
18        android:text="@string/cheese"
19        android:onClick="onCheckboxClicked" />
20
21 </LinearLayout>
22
```

CheckBox

```
1 public void onCheckboxClicked(View view) {  
2     // Is the view now checked?  
3     boolean checked = ((CheckBox) view).isChecked();  
4  
5     // Check which checkbox was clicked  
6     switch(view.getId()) {  
7         case R.id.checkbox_browser:  
8             if (checked)  
9                 // Synchronize browser  
10                break;  
11        case R.id.checkbox_calendar:  
12            if (checked)  
13                // Synchronize calendar  
14                break;  
15        }  
16    }  
17 }
```

Pregled sadržaja

- 1 Resursi i konfiguracije uređaja
- 2 Grafički korisnički interfejs
- 3 Pogledi
- 4 Rasporedi**
- 5 Stilovi i teme

Raspored

- Raspored (layout) je pogled koji sadrži druge poglede i raspoređuje ih po ekranu
- Kao i svaki drugi pogled, može se definisati proceduralno u Java kodu (instanciranjem klase) ili deklarativno u XML kodu (dodavanjem elementa)

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:orientation="vertical">
6
7   <TextView
8     android:id="@+id/text"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content"
11    android:text="Hello , I am a TextView" />
12
13   <Button
14     android:id="@+id/button"
15     android:layout_width="wrap_content"
16     android:layout_height="wrap_content"
17     android:text="Hello , I am a Button" />
18 </LinearLayout>
19
```

Svojstva pogleda

- Slično CSS box modelu, svaki pogled ima geometriju pravougaonika
- Poziciju i dimenzije pogleda određuje vrsta rasporeda koji ga sadrži i svojstva pogleda (koja mogu da zavise od vrste rasporeda)
- Neka svojstva (npr. padding i margin) ne zavise od vrste rasporeda
- Neka svojstva (npr. layout_width i layout_height) zavise od vrste rasporeda

Merne jedinice

Oznaka	Naziv	Opis
dp	density-independent pixels	apstraktna merna jedinica koja odgovara veličini piksela na ekranu rezolucije 160 dpi
sp	scale-independent pixels	kao i dp, samo što se piksel sklaire i u odnosu na faktor skaliranja fonta
pt	tačka	1/72 in
px	piksel	fizička veličina piksela na ekranu
mm	milimetar	
in	inč	

Table 7: Merne jedinice.

Svojstva pogleda

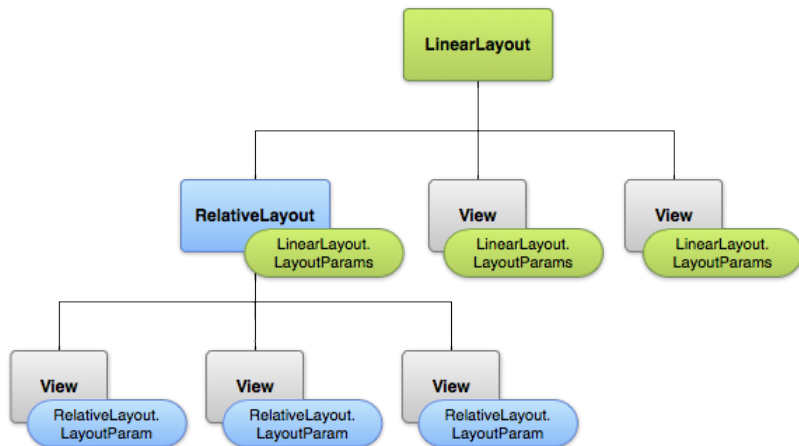


Figure 20: Svojstva pogleda.

Iscrtavanje pogleda

- Svaki pogled iscrtava sebe i svoju decu
- Iscrtavanje pogleda izvršava se u dva prolaza:
 - prolazu merenja (measure pass)
 - prolazu raspoređivanja (layout pass)

Vrste rasporeda

- `AbsoluteLayout`
- `GridLayout`
- `FrameLayout`
- `LinearLayout`
- `RelativeLayout`
- `DrawerLayout`
- `ConstraintLayout`
- `CoordinatorLayout`
- ...

LinearLayout



Figure 21: Linearni raspored.

- Linearni raspored (LinearLayout) je raspored koji raspoređuje decu u jednom pravcu (vertikalno ili horizontalno)
- Deca linearnog rasporeda raspoređena su jedno pored drugog, tako da vertikalni raspored ima samo jedno dete po vrsti (a horizontalni samo jedno dete po koloni)

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:paddingLeft="16dp"
6   android:paddingRight="16dp"
7   android:orientation="vertical">
8
9   <EditText
10     android:layout_width="fill_parent"
11     android:layout_height="wrap_content"
12     android:hint="@string/to" />
13
14   <EditText
15     android:layout_width="fill_parent"
16     android:layout_height="wrap_content"
17     android:hint="@string/subject" />
18
```

main.xml

```
1  <EditText
2      android:layout_width="fill_parent"
3      android:layout_height="0dp"
4      android:layout_weight="1"
5      android:gravity="top"
6      android:hint="@string/message" />
7
8  <Button
9      android:layout_width="100dp"
10     android:layout_height="wrap_content"
11     android:layout_gravity="right"
12     android:text="@string/send" />
13
14 </LinearLayout>
15
```

LinearLayout

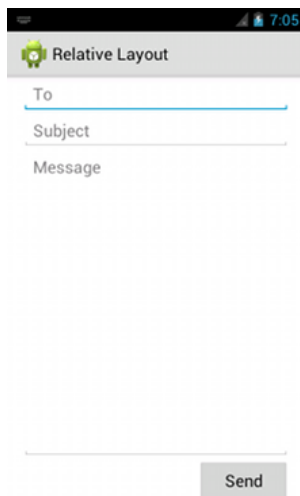


Figure 22: Linearni raspored.

LinearLayout

Svojstvo	Opis
layout_weight	assigns an "importance" value to a view in terms of how much space it should occupy on the screen
gravity	specifies how an object should position its content, on both the X and Y axes, within its own bounds
orientation	use "horizontal" for a row, "vertical" for a column orientation

Table 8: Svojstva linearnog rasporeda.

RelativeLayout



Figure 23: Relativni raspored.

- Relativni raspored (RelativeLayout) je raspored koji raspoređuje decu relativno u odnosu na sebe i jedno na drugo
- Pozicija pogleda može se specificirati u odnosu na elemente istog hijerarhijskog nivoa (levo, desno, iznad ili ispod drugog pogleda) ili u odnosu na roditelja (poravna sa levom, desnom, gornjom ili donjom ivicom)

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:paddingLeft="16dp"
6   android:paddingRight="16dp">
7
8   <EditText
9     android:id="@+id/name"
10    android:layout_width="fill_parent"
11    android:layout_height="wrap_content"
12    android:hint="@string/reminder" />
13
14   <Spinner
15     android:id="@+id/dates"
16     android:layout_width="0dp"
17     android:layout_height="wrap_content"
18     android:layout_below="@id/name"
19     android:layout_alignParentLeft="true"
20     android:layout_toLeftOf="@+id/times" />
21
```

main.xml

```
1  <Spinner
2      android:id="@id/times"
3      android:layout_width="96dp"
4      android:layout_height="wrap_content"
5      android:layout_below="@id/name"
6      android:layout_alignParentRight="true" />
7
8  <Button
9      android:layout_width="96dp"
10     android:layout_height="wrap_content"
11     android:layout_below="@id/times"
12     android:layout_alignParentRight="true"
13     android:text="@string/done" />
14
15 </RelativeLayout>
16
```

RelativeLayout

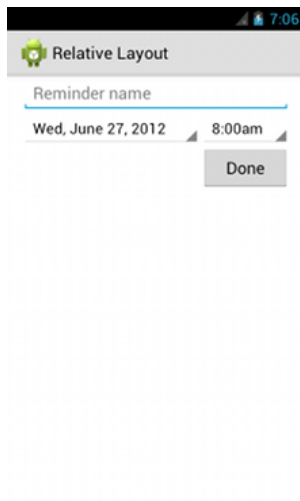


Figure 24: Relativni raspored.

Svojstva relativnog rasporeda

Svojstvo	Opis
<code>layout_alignParentTop</code>	If "true", makes the top edge of this view match the top edge of the parent
<code>layout_centerVertical</code>	If "true", centers this child vertically within its parent
<code>layout_below</code>	Positions the top edge of this view below the view specified with a resource ID
<code>layout_toRightOf</code>	Positions the left edge of this view to the right of the view specified with a resource ID

Table 9: Svojstva relativnog rasporeda.

ConstraintLayout

- Ograničavajući raspored (ConstraintLayout) je raspored koji omogućava određivanje pozicije i veličine pogleda na fleksibilan način
- Pozicije i veličine pogleda određuju se na osnovu ograničenja u odnosu na druge poglede, roditeljski raspored ili nevidljive vođice
- Ograničavajući raspored specijalizuje relativni raspored i prilagođen je radu u vizuelnom okruženju Android Studio

build.gradle

```
1 dependencies {  
2     compile 'com.android.support.constraint:constraint-layout:1.0.2'  
3 }  
4
```


main.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     app:layout_behavior="@string/appbar_scrolling_view_behavior"
9     tools:context="com.journaldev.constraintlayoutplaying.MainActivity"
10     >
11     <Button
12         android:id="@+id/button"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:text="Button"
16         app:layout_constraintBottom_toBottomOf="parent"
17         app:layout_constraintLeft_toLeftOf="parent"
18         app:layout_constraintRight_toRightOf="parent"
19         app:layout_constraintTop_toTopOf="parent" />
20
21 </android.support.constraint.ConstraintLayout>
22
23

```

ConstraintLayout

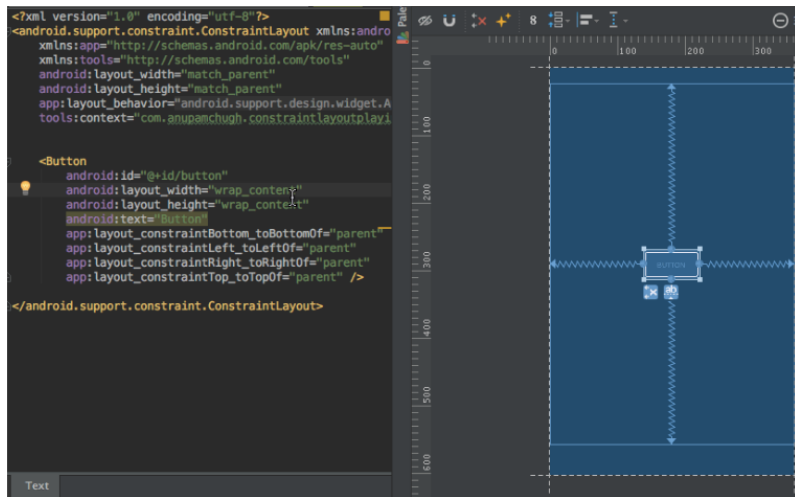


Figure 25: Ograničavajući raspored.

FrameLayout

- FrameLayout raspored prikazuje više pogleda koji će biti raspoređeni jedan na drugom (poslednji koji je dodat biće raspoređen na vrhu)
- Veličina FrameLayout rasporeda odgovara veličini najvećeg pogleda koga sadrži (ako to dozvoljava roditelj)

main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <FrameLayout
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     xmlns:android="http://schemas.android.com/apk/res/android">
6
7     <ImageView
8         android:id="@+id/iv_logo"
9         android:layout_height="match_parent"
10        android:layout_width="match_parent"
11        android:src="@drawable/logo" />
12
13    <TextView
14        android:id="@+id/tv_label"
15        android:text="@string/label"
16        android:layout_height="match_parent"
17        android:layout_width="match_parent" />
18
19 </FrameLayout>
20
```

FrameLayout

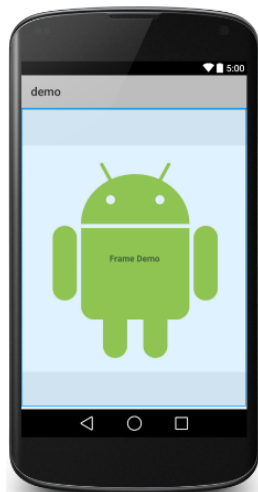


Figure 26: Okvir raspored.

CoordinatorLayout

- Koordinirajući raspored (CoordinatorLayout) upravlja interakcijom između pogleda koje sadrži
- Obično se koristi kao koreni raspored aktivnosti ili fragmenta

main.xml

```
1 <android.support.design.widget.CoordinatorLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:id="@+id/coordinator_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <android.support.design.widget.AppBarLayout
9         android:id="@+id/app_bar"
10        android:layout_width="match_parent"
11        android:layout_height="wrap_content">
12
13        <android.support.v7.widget.Toolbar
14            android:layout_width="match_parent"
15            android:layout_height="?attr/actionBarSize"/>
16
17    </android.support.design.widget.AppBarLayout>
18
```

main.xml

```
1  <!-- Main content -->
2
3  <android.support.design.widget.FloatingActionButton
4      android:layout_width="wrap_content"
5      android:layout_height="wrap_content"
6      android:layout_margin="16dp"
7      android:contentDescription="@string/add_item"
8      android:src="@drawable/ic_add_24dp"
9      app:layout_anchor="@id/app_bar"
10     app:layout_anchorGravity="bottom | right | end"/>
11
12 </android.support.design.widget.CoordinatorLayout>
13
```


CoordinatorLayout

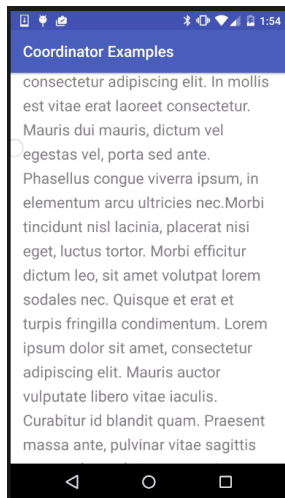


Figure 27: Koordinirajući raspored.

CoordinatorLayout

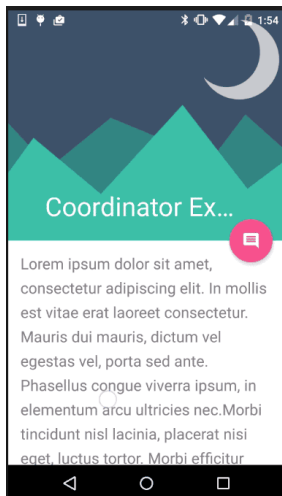


Figure 28: Koordinirajući raspored.

Pregled sadržaja

- 1 Resursi i konfiguracije uređaja
- 2 Grafički korisnički interfejs
- 3 Pogledi
- 4 Rasporedi
- 5 **Stilovi i teme**

Stilovi

- Stil je skup svojstava koja specificiraju izgled pogleda
- Neka od tih svojstava su visina (height), širina (width), punjene (padding), margina (margin), font (typeface), boja teksta (font color), veličina teksta (font size), boja pozadine (background color), itd.
- Stilovi se definišu se kao poseban resurs (odgovara im XML dokument u res/values direktorijumu) i moguće je ih nasleđivati

main__activity.xml

```
1 <TextView
2     android:id="@+id/text_view"
3     android:text="@string/hello_world"
4     android:layout_width="fill_parent"
5     android:layout_height="wrap_content"
6     android:typeface="monospace"
7     android:textSize="15sp"
8     android:textStyle="italic"
9     android:textColor="#00FF00" />
10
```

main__activity.xml

```
1 <TextView
2     android:id="@+id/text_view"
3     android:text="@string/hello_world"
4     android:layout_width="fill_parent"
5     android:layout_height="wrap_content"
6     style="@style/example_style" />
7
```

style.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <style name="example_style">
4     <item name="android:typeface">monospace</item>
5     <item name="android:textSize">15sp</item>
6     <item name="android:textStyle">italic </item>
7     <item name="android:textColor">#00FF00</item>
8   </style>
9 </resources>
10
```

style.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <style name="example_style" parent="@android:style/Theme.Material">
4     <item name="android:typeface">monospace</item>
5     <item name="android:textSize">15sp</item>
6     <item name="android:textStyle">italic </item>
7     <item name="android:textColor">#00FF00</item>
8   </style>
9 </resources>
10
```


Teme

- Tema je stil primenjen na celu aktivnost ili celu aplikaciju umesto na pojedinačan pogled
- Tada se na svaki pogled aktivnosti ili aplikacije primenjuje svako svojstvo teme koje pogled podržava
- Stilovi su lokalni, teme su globalne

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application android:theme="@android:style/Theme.Material" ... >
4     <!-- ... -->
5   </application>
6 </manifest>
7
```

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3   <application ... >
4     <activity android:name=". ExampleActivity"
5       android:theme="@android: style/example_style" ... >
6       <intent-filter >
7         <action android:name="android .intent .action .MAIN" />
8         <category android:name="android .intent .category .LAUNCHER" />
9       </intent-filter >
10    </activity>
11  </application>
12 </manifest>
13
```



All images copyrighted by Android Open Source Project (CC BY)