

Pogledi (3)

Android App Development 6

Nikola Đukić

1. jun 2018.

Pregled sadržaja

- 1 Toasts/Snackbars
- 2 Obaveštenja
- 3 Dijalozi
- 4 Podešavanja
- 5 Toolbar
- 6 Navigation Drawer

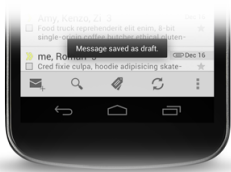
Toasts

- Toast je pop-up poruka koja automatski nestaje posle određenog vremena



Slika 1: Toast.

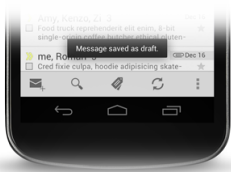
Toasts



Slika 1: Toast.

- Toast je pop-up poruka koja automatski nestaje posle određenog vremena
- Korisniku daje povratnu informaciju da je akcija izvršena

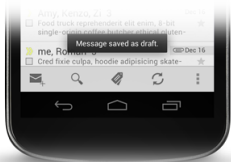
Toasts



Slika 1: Toast.

- Toast je pop-up poruka koja automatski nestaje posle određenog vremena
- Korisniku daje povratnu informaciju da je akcija izvršena
- Aktivnost na vrhu povratnog steka ostaje vidljiva i u fokusu

Toasts



Slika 1: Toast.

- Toast je pop-up poruka koja automatski nestaje posle određenog vremena
- Korisniku daje povratnu informaciju da je akcija izvršena
- Aktivnost na vrhu povratnog steka ostaje vidljiva i u fokusu
- U novijim verzijama Android platforme preporučljivo je koristiti Snackbar

MainActivity.java

```
1 Context context = getApplicationContext();
2 CharSequence message = "Hello World!";
3 int duration = Toast.LENGTH_SHORT;
4 Toast toast = Toast.makeText(context, message, duration);
5 toast.show();
```

MainActivity.java

```
1 Context context = getApplicationContext();
2 CharSequence message = "Hello World!";
3 int duration = Snackbar.LENGTH_SHORT;
4 Snackbar snackbar =
5     Snackbar.make(findViewById(R.id.parent), message, duration);
6 snackbar.setAction(R.string.undo_string, new UndoListener());
7 snackbar.show();
```


Pregled sadržaja

- 1 Toasts/Snackbars
- 2 **Obaveštenja**
- 3 Dijalozi
- 4 Podešavanja
- 5 Toolbar
- 6 Navigation Drawer

Obaveštenja

- Obaveštenje (notification) je poruka koja se prikazuje van korisničkog interfejsa aplikacije (u površini za obaveštenja ili fioci za obaveštenja)

Obaveštenja

- Obaveštenje (notification) je poruka koja se prikazuje van korisničkog interfejsa aplikacije (u površini za obaveštenja ili fioci za obaveštenja)
- Ne prekida korisnika u izvršavanju tekućeg zadatka

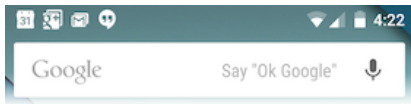
Obaveštenja

- Obaveštenje (notification) je poruka koja se prikazuje van korisničkog interfejsa aplikacije (u površini za obaveštenja ili fioci za obaveštenja)
- Ne prekida korisnika u izvršavanju tekućeg zadatka
- Obično se prikazuju obaveštenja o vremenski kritičnim događajima ili događajima u kojima učestvuju drugi ljudi

Obaveštenja

- Obaveštenje (notification) je poruka koja se prikazuje van korisničkog interfejsa aplikacije (u površini za obaveštenja ili fioci za obaveštenja)
- Ne prekida korisnika u izvršavanju tekućeg zadatka
- Obično se prikazuju obaveštenja o vremenski kritičnim događajima ili događajima u kojima učestvuju drugi ljudi
- Moguće je i izvršiti akciju iz obaveštenja

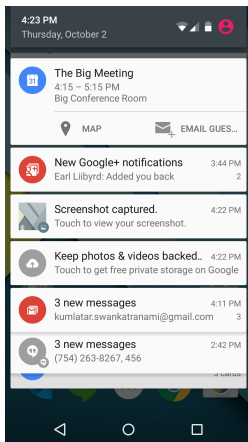
Obaveštenja



Slika 2: Površina za obaveštenja.

- Obaveštenje se prikazuje kao ikona u površini za obaveštenja (notification area)

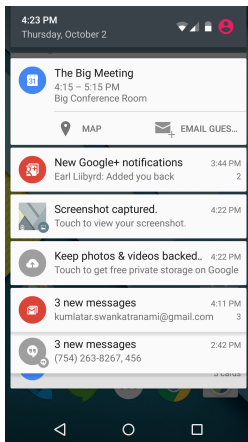
Obaveštenja



- Više informacija o obaveštenju prikazuje se u fioci za obaveštenja (notification drawer)

Slika 3: Fioka za obaveštenja.

Obaveštenja



- Više informacija o obaveštenju prikazuje se u fioci za obaveštenja (notification drawer)
- Iz obaveštenja je moguće izvršiti i akcije

Slika 3: Fioka za obaveštenja.

Obaveštenja

```
1 Context context = getApplicationContext();
2 NotificationCompat.Builder builder = new NotificationCompat.Builder(context)
3     .setContentTitle(title)                // Mandatory property
4     .setLargeIcon(R.drawable.large_icon)
5     .setContentText(text)                  // Mandatory property
6     .setContentInfo(info)
7     .setSmallIcon(R.drawable.small_icon)   // Mandatory property
8     .setWhen(when)
9     .setStyle(new Notification.BigPictureStyle().bigPicture(bigBitmap))
10    .setSound(soundURI)
11    .setLights(color, onDuration, offDuration)
12    .setVibrate(pattern)
13    .setPriority(priority)
```

Obaveštenja

Parametar	Opis
color	boja LED (RGB)
onDuration	interval u kome je LED uključena (ms)
offDuration	interval u kome je LED isključena (ms)

Tabela 1: Parametri setLights metode.

Obaveštenja

Parametar	Opis
pattern	interval u kome telefon vibrira (ms), interval u kome telefon ne vibrira (ms), itd.

Tabela 2: Parametri setVibrate metode.

Obaveštenja

Vrednost	Opis
priority	MAX (critical and urgent), HIGH (high priority), DEFAULT (default), LOW (low in urgency), MIN (contextual or background)

Tabela 3: Parametri setPriority metode.

Obaveštenja

```
1  // Creates an explicit intent
2  Intent intent = new Intent(this, ResultActivity.class);
3  // The stack builder contains an artificial back stack
4  TaskStackBuilder tsb = TaskStackBuilder.create(this);
5  // Adds the back stack for the Intent
6  tsb.addParentStack(ResultActivity.class);
7  // Adds the Intent to the top of the stack
8  tsb.addNextIntent(intent);
9  PendingIntent pe = tsb.getPendingIntent(0, pe.FLAG_UPDATE_CURRENT);
10 builder.setContentIntent(pe);
11 // Returns the handle to the notification manager
12 NotificationManager manager =
13     (NotificationManager).getSystemService(Context.NOTIFICATION_SERVICE);
14 // Notifies the user (id allows notification update later on)
15 manager.notify(id, builder.build());
```

Pregled sadržaja

- 1 Toasts/Snackbars
- 2 Obaveštenja
- 3 Dijalozi**
- 4 Podešavanja
- 5 Toolbar
- 6 Navigation Drawer

Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije

Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije
- Ne zauzima ceo ekran (aktivnost koja prikazuje dijalog se pauzira)

Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije
- Ne zauzima ceo ekran (aktivnost koja prikazuje dijalog se pauzira)
- Postoje predefinisani dijalozi kao što su:

Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije
- Ne zauzima ceo ekran (aktivnost koja prikazuje dijalog se pauzira)
- Postoje predefinisani dijalozi kao što su:
 - AlertDialog

Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije
- Ne zauzima ceo ekran (aktivnost koja prikazuje dijalog se pauzira)
- Postoje predefinisani dijalozi kao što su:
 - AlertDialog
 - DatePicker

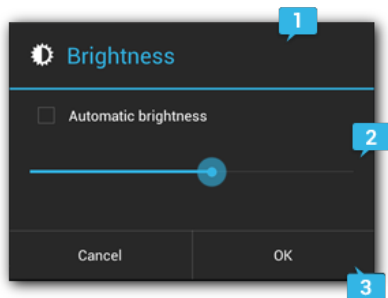
Dijalozi

- Dijalog je modalni prozor koji prikazuje poruku i (opciono) omogućava unos podataka i potvrdu izvršavanja akcije
- Ne zauzima ceo ekran (aktivnost koja prikazuje dijalog se pauzira)
- Postoje predefinisani dijalozi kao što su:
 - AlertDialog
 - DatePicker
 - TimePicker

Dijalozi

- Preporučljivo je da se umesto klase `Dialog` koristi klasa `DialogFragment` (ona vodi računa o životnom ciklusu dijaloga i omogućava ponovno korišćenje dijaloga kao elementa složenog korisničkog interfejsa)

Dijalozi

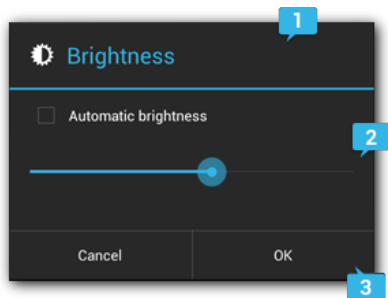


AlertDialog sadrži:

- 1 naslov

Slika 4: AlertDialog.

Dijalozi

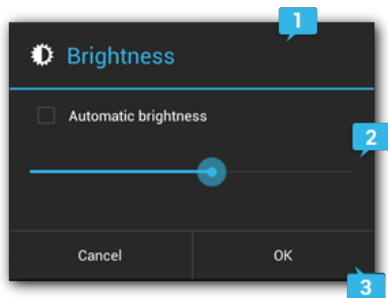


AlertDialog sadrži:

- 1 naslov
- 2 poruku, listu ili proizvoljan raspored

Slika 4: AlertDialog.

Dijalozi



AlertDialog sadrži:

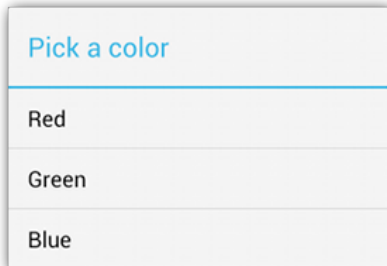
- ❶ naslov
- ❷ poruku, listu ili proizvoljan raspored
- ❸ do tri dugmeta (negativno, neutralno i pozitivno)

Slika 4: AlertDialog.

ExampleDialogFragment.java

```
1  public class ExampleDialogFragment extends DialogFragment {
2
3      @Override
4      public Dialog onCreateDialog(Bundle state) {
5          return new AlertDialog.Builder(getActivity())
6              .setMessage(R.string.message_id)
7              .setPositiveButton(
8                  R.string.btnOK,
9                  new DialogInterface.OnClickListener() {
10                     public void onClick(DialogInterface dialog, int id) {
11                         // ...
12                     }
13                 })
14              .create();
15     }
16 }
```

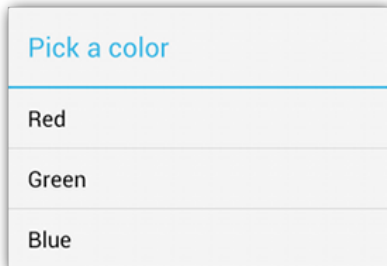
Dijalozi



- Alert dijalog može da sadrži:
 - listu

Slika 5: Dijalog sa listom.

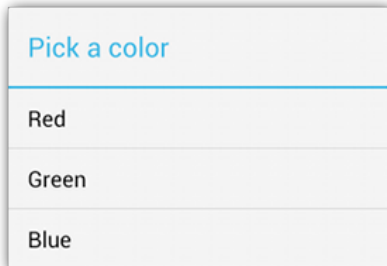
Dijalozi



Slika 5: Dijalog sa listom.

- Alert dijalog može da sadrži:
 - listu
 - radio buttons

Dijalozi



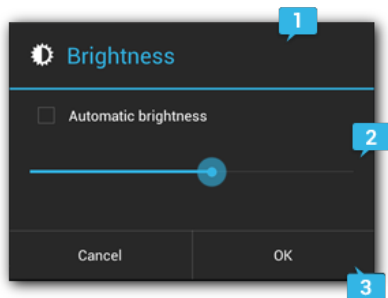
Slika 5: Dijalog sa listom.

- Alert dijalog može da sadrži:
 - listu
 - radio buttons
 - checkboxes
- Stavke se mogu definisati u statičkom nizu ili adapteru

ExampleDialogFragment.java

```
1  public class ExampleDialogFragment extends DialogFragment {  
2  
3      @Override  
4      public Dialog onCreateDialog(Bundle state) {  
5          return new AlertDialog.Builder(getActivity())  
6              .setTitle(R.string.pick_color)  
7              .setItems(  
8                  R.array.colors_array,  
9                  new DialogInterface.OnClickListener() {  
10                     public void onClick(DialogInterface dialog, int which) {  
11                         // ...  
12                     }  
13                 })  
14              .create();  
15      }  
16  }
```

Dijalozi



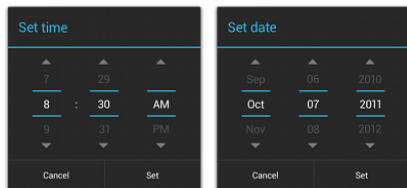
- Podrazumevano ponašanje je da raspored zauzme ceo ekran
- Moguće je dodati naslov i dugmad

Slika 6: Dijalog sa rasporedom.

ExampleDialogFragment.java

```
1  public class ExampleDialogFragment extends DialogFragment {
2
3      @Override
4      public Dialog onCreateDialog(Bundle state) {
5          return new AlertDialog.Builder(getActivity())
6              .setView(
7                  getActivity().getLayoutInflater().inflate(R.layout.dialog_layout, null))
8              .setPositiveButton(
9                  R.string.btnOK,
10                 new DialogInterface.OnClickListener() {
11                     @Override
12                     public void onClick(DialogInterface dialog, int id) {
13                         // ...
14                     }
15                 })
16              .create();
17      }
18  }
```

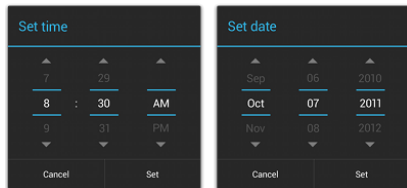
TimePicker & DatePicker



- 1 Za unos vremena koristi se predefinisani dijalog TimePicker

Slika 7: TimePicker & DatePicker.

TimePicker & DatePicker



Slika 7: TimePicker & DatePicker.

- 1 Za unos vremena koristi se predefinisani dijalog TimePicker
- 2 Za unos datuma koristi se predefinisani dijalog DatePicker

DatePicker

```
1  public class DatePickerFragment
2      extends DialogFragment
3      implements DatePickerDialog.OnDateSetListener {
4
5      @Override
6      public Dialog onCreateDialog(Bundle state) {
7          Calendar c = Calendar.getInstance();
8          int year = c.get(Calendar.YEAR);
9          int month = c.get(Calendar.MONTH);
10         int day = c.get(Calendar.DAY_OF_MONTH);
11         return new DatePickerDialog(getActivity(), this, year, month, day);
12     }
13
14     @Override
15     public void onDateSet(DatePicker view, int year, int month, int day) {
16         // ...
17     }
18 }
```

TimePicker

```
1  public class TimePickerFragment
2      extends DialogFragment
3      implements TimePickerDialog.OnTimeSetListener {
4
5      @Override
6      public Dialog onCreateDialog(Bundle state) {
7          Calendar c = Calendar.getInstance();
8          int hourOfDay = c.get(Calendar.HOUR_OF_DAY);
9          int minute = c.get(Calendar.MINUTE);
10         return new TimePickerDialog(getActivity(), this, hourOfDay, minute, true);
11     }
12
13     @Override
14     public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
15         // ...
16     }
17 }
```

Prikazivanje dijaloga

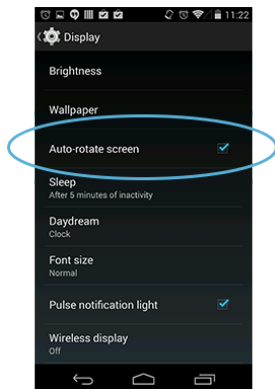
```
1 public void showDialog() {  
2     DialogFragment dialog = new DatePickerFragment();  
3     // Tag name is used to save and restore fragment state  
4     // and to get a handle to the fragment  
5     dialog.show(getSupportFragmentManager(), "tag_name");  
6 }
```

Pregled sadržaja

- 1 Toasts/Snackbars
- 2 Obaveštenja
- 3 Dijalozi
- 4 Podešavanja**
- 5 Toolbar
- 6 Navigation Drawer

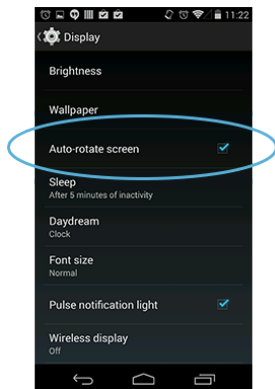
Podešavanja

- Za podešavanje aplikacije koristi se Preferece API (da bi ponašanje aplikacija bilo konzistentno)



Slika 8: Podešavanja.

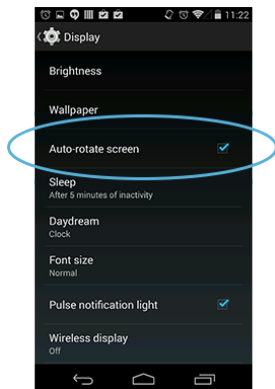
Podešavanja



- Za podešavanje aplikacije koristi se Preference API (da bi ponašanje aplikacija bilo konzistentno)
- Različitim tipovima parametrima odgovaraju različiti tipovi kontrola koje nasleđuju Preference klasu

Slika 8: Podešavanja.

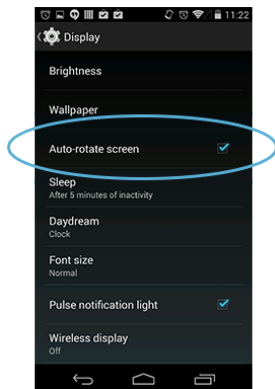
Podešavanja



Slika 8: Podešavanja.

- Za podešavanje aplikacije koristi se Preference API (da bi ponašanje aplikacija bilo konzistentno)
- Različitim tipovima parametrima odgovaraju različiti tipovi kontrola koje nasleđuju Preference klasu
- Kontrole se mogu grupisati u kategorije ili u podekrane

Podešavanja



Slika 8: Podešavanja.

- Za podešavanje aplikacije koristi se Preferece API (da bi ponašanje aplikacija bilo konzistentno)
- Različitim tipovima parametrima odgovaraju različiti tipovi kontrola koje nasleđuju Preference klasu
- Kontrole se mogu grupisati u kategorije ili u podekrane
- Vrednosti parametara se automatski učitavaju i snimaju (više o tome na jednom od narednih časova)

Podešavanja

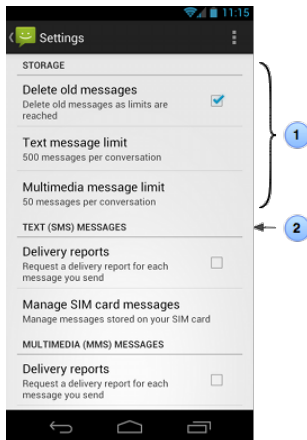
Tip parametra	Tip kontrole
Boolean	CheckBoxPreference
Float	EditTextPreference
Int	EditTextPreference
Long	EditTextPreference
String	EditTextPreference, ListPreference
Set<String>	MultiSelectListPreference

Tabela 4: Tipovi kontrola.

preferences.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <PreferenceScreen
3      xmlns:android="http://schemas.android.com/apk/res/android">
4
5      <CheckBoxPreference
6          android:key="pref_sync"
7          android:title="@string/pref_sync"
8          android:summary="@string/pref_sync_summ"
9          android:defaultValue="true" />
10
11     <ListPreference
12         android:dependency="pref_sync"
13         android:key="pref_syncConnectionType"
14         android:title="@string/pref_syncConnectionType"
15         android:dialogTitle="@string/pref_syncConnectionType"
16         android:entries="@array/pref_syncConnectionTypes_entries"
17         android:entryValues="@array/pref_syncConnectionTypes_values"
18         android:defaultValue="@string/pref_syncConnectionTypes_default" />
19
20 </PreferenceScreen>
```

Podešavanja



Slika 9: Podešavanja sa naslovima.

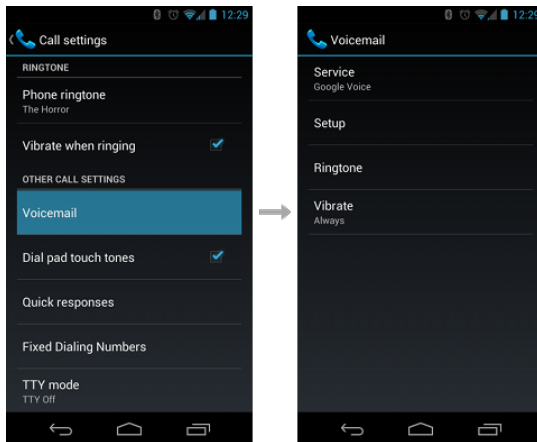
preferences.xml (1/2)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <PreferenceScreen
3      xmlns:android="http://schemas.android.com/apk/res/android">
4
5      <PreferenceCategory
6          android:title="@string/pref_sms_storage_title"
7          android:key="pref_key_storage_settings">
8
9          <CheckBoxPreference
10              android:key="pref_key_auto_delete"
11              android:summary="@string/pref_summary_auto_delete"
12              android:title="@string/pref_title_auto_delete"
13              android:defaultValue="false" />
```

preferences.xml (2/2)

```
1      <EditTextPreference
2          android:key="pref_key_sms_delete_limit"
3          android:dependency="pref_key_auto_delete"
4          android:summary="@string/pref_summary_delete_limit"
5          android:title="@string/pref_title_sms_delete" />
6
7      <EditTextPreference
8          android:key="pref_key_mms_delete_limit"
9          android:dependency="pref_key_auto_delete"
10         android:summary="@string/pref_summary_delete_limit"
11         android:title="@string/pref_title_mms_delete" />
12
13  </PreferenceCategory>
14
15  </PreferenceScreen>
```

Podešavanja



Slika 10: Podešavanja sa podekranima.

preferences.xml (1/2)

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <PreferenceScreen
3      xmlns:android="http://schemas.android.com/apk/res/android">
4
5      <!-- ... -->
6
7      <!-- opens a subscreen of settings -->
8      <PreferenceScreen
9          android:key="button_voicemail_category_key"
10         android:title="@string/voicemail"
11         android:persistent="false">
12
13         <ListPreference
14             android:key="button_voicemail_provider_key"
15             android:title="@string/voicemail_provider" />
```


preferences.xml (2/2)

```
1      <!-- opens another nested subscreen -->
2      <PreferenceScreen
3          android:key="button_voicemail_setting_key"
4          android:title="@string/voicemail_settings"
5          android:persistent="false">
6
7          <!-- ... -->
8
9      </PreferenceScreen>
10
11     <RingtonePreference
12         android:key="button_voicemail_ringtone_key"
13         android:title="@string/voicemail_ringtone_title"
14         android:ringtoneType="notification" />
15
16     <!-- ... -->
17
18 </PreferenceScreen>
19
20 <!-- ... -->
21
22 </PreferenceScreen>
```

Prikaz ekrana za podešavanja

```
1  public class SettingsFragment extends PreferenceFragment {  
2  
3      @Override  
4      public void onCreate(Bundle state) {  
5          super.onCreate(state);  
6          // Load the preferences from an XML resource  
7          addPreferencesFromResource(R.xml.preferences);  
8      }  
9  
10     // ...  
11  
12 }
```

Prikaz ekrana za podešavanja

```
1  public class SettingsActivity extends Activity {
2
3      @Override
4      protected void onCreate(Bundle state) {
5          super.onCreate(state);
6          // Display the fragment as the main content
7          getSupportFragmentManager()
8              .beginTransaction()
9              .replace(android.R.id.content, new SettingsFragment())
10             .commit();
11     }
12
13     // ...
14
15 }
```

Pregled sadržaja

- 1 Toasts/Snackbars
- 2 Obaveštenja
- 3 Dijalozi
- 4 Podešavanja
- 5 Toolbar**
- 6 Navigation Drawer

Toolbar

Toolbar je element GUI-a koji se (obično) nalazi na vrhu ekrana i obezbeđuje:

- branding aplikacije

Toolbar

Toolbar je element GUI-a koji se (obično) nalazi na vrhu ekrana i obezbeđuje:

- branding aplikacije
- navigaciju

Toolbar

Toolbar je element GUI-a koji se (obično) nalazi na vrhu ekrana i obezbeđuje:

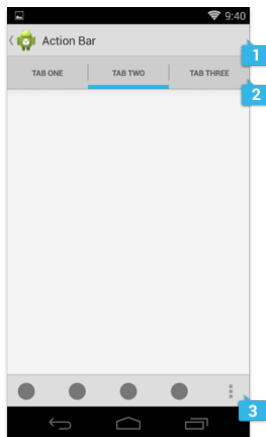
- branding aplikacije
- navigaciju
- promenu pogleda

Toolbar

Toolbar je element GUI-a koji se (obično) nalazi na vrhu ekrana i obezbeđuje:

- branding aplikacije
- navigaciju
- promenu pogleda
- izvršavanje akcija

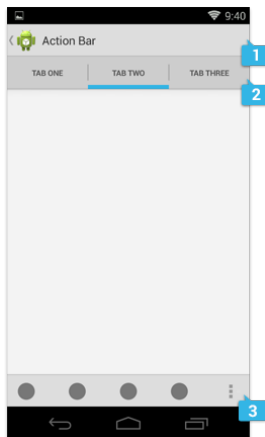
Toolbar



- 1 title area (identifikuje aplikaciju)

Slika 11: Toolbar.

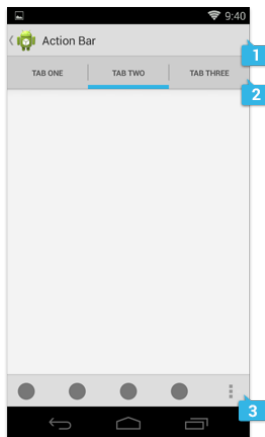
Toolbar



- 1 title area (identifikuje aplikaciju)
- 2 navigation area (omogućava navigaciju)

Slika 11: Toolbar.

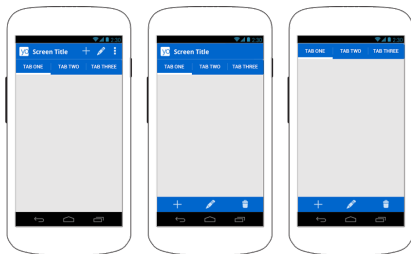
Toolbar



- ① title area (identifikuje aplikaciju)
- ② navigation area (omogućava navigaciju)
- ③ action area (omogućava izvršavanje akcija, ređe korišćene akcije su "prelivenne" u meni)

Slika 11: Toolbar.

Toolbar



Slika 12: Toolbar.

Može se podeliti u više delova:

- 1 glavni deo (prikazuje ikonu i omogućava navigaciju)

Toolbar



Slika 12: Toolbar.

Može se podeliti u više delova:

- 1 glavni deo (prikazuje ikonu i omogućava navigaciju)
- 2 gornji deo (omogućava promenu pogleda)

Toolbar



Slika 12: Toolbar.

Može se podeliti u više delova:

- 1 glavni deo (prikazuje ikonu i omogućava navigaciju)
- 2 gornji deo (omogućava promenu pogleda)
- 3 donji deo (omogućava izvršavanje akcija)

Pravljenje toolbar-a

- Dodati v7 appcompat support library u projekat

Pravljenje toolbar-a

- Dodati v7 appcompat support library u projekat
- Definirati klasu koja nasleđuje AppCompatActivity klasu i u onCreate() metodi pozvati setSupportActionBar()

Pravljenje toolbar-a

- Dodati v7 appcompat support library u projekat
- Definirati klasu koja nasleđuje AppCompatActivity klasu i u onCreate() metodi pozvati setSupportActionBar()
- Koristiti jednu od AppCompatActivity.NoActionBar tema (na taj način se sprečava korišćenje ugrađene ActionBar klase)

Pravljenje toolbar-a

- Dodati v7 appcompat support library u projekat
- Definirati klasu koja nasleđuje AppCompatActivity klasu i u onCreate() metodi pozvati setSupportActionBar()
- Koristiti jednu od AppCompatActivity.NoActionBar tema (na taj način se sprečava korišćenje ugrađene ActionBar klase)
- Dodati toolbar u raspored

build.gradle

```
1 dependencies {  
2     compile 'com.android.support:appcompat-v7:23.2.1'  
3 }
```

ExampleActivity.java

```
1  public class ExampleActivity extends AppCompatActivity {  
2  
3      @Override  
4      protected void onCreate(Bundle state) {  
5          super.onCreate(state);  
6          setContentView(R.layout.layout_main);  
7          Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
8          setSupportActionBar(toolbar);  
9      }  
10  
11 }
```

AndroidManifest.java

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

layout_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      <android.support.v7.widget.Toolbar
8          android:id="@+id/toolbar"
9          android:layout_width="match_parent"
10         android:layout_height="?attr/actionBarSize"
11         android:background="?attr/colorPrimary"
12         android:elevation="4dp"
13         android:theme="@style/ThemeOverlay.AppCompat.ActionBar"
14         app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />
15
16  </LinearLayout>
```

Izvršavanje akcija

- Deklarisati meni kao resurs

Izvršavanje akcija

- Deklarisati meni kao resurs
- Reagovati na akcije korisnika u `onOptionsItemSelected` metodi aktivnosti

action_bar.xml

```
1 <menu xmlns:android="http://schemas.android.com/apk/res/android">
2
3 <item
4     android:id="@+id/action_search"
5     android:icon="@drawable/ic_action_search"
6     android:title="@string/action_search"
7     android:showAsAction="ifRoom|withText|always" />
8
9 <item
10    android:id="@+id/action_compose"
11    android:icon="@drawable/ic_action_compose"
12    android:title="@string/action_compose" />
13
14 </menu>
```

ExampleActivity.java

```
1  @Override
2  public boolean onOptionsItemSelected(MenuItem item) {
3      switch (item.getItemId()) {
4          case R.id.action_search:
5              openSearch();
6              return true;
7          case R.id.action_compose:
8              composeMessage();
9              return true;
10         default:
11             return super.onOptionsItemSelected(item);
12     }
13 }
```

Up dugme

- U `AndroidManifest.xml` deklarirati aktivnost kao dete druge aktivnosti

Up dugme

- U `AndroidManifest.xml` deklarirati aktivnost kao dete druge aktivnosti
- U njoj aktivnosti pozvati `setDisplayHomeAsUpEnabled` metodu i proslediti joj argument `true`

AndroidManifest.xml

```
1  <manifest ... >
2    <application ... >
3      <!-- ... -->
4      <!-- The main/home activity (has no parent activity) -->
5      <activity android:name="com.example.MainActivity">
6        <!-- ... -->
7      </activity>
8
9      <!-- A child of the main activity -->
10     <activity
11       android:name="com.example.ChildActivity"
12       android:parentActivityName="com.example.MainActivity">
13       <!-- ... -->
14     </activity>
15   </application>
16 </manifest>
```

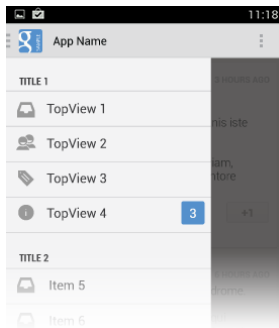
ExampleActivity.java

```
1  @Override
2  protected void onCreate(Bundle state) {
3      // ...
4      ActionBar toolbar = getSupportActionBar();
5      toolbar.setDisplayHomeAsUpEnabled(true);
6      // ...
7  }
```

Pregled sadržaja

- 1 Toasts/Snackbars
- 2 Obaveštenja
- 3 Dijalozi
- 4 Podešavanja
- 5 Toolbar
- 6 Navigation Drawer**

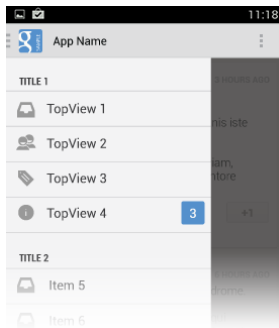
NavigationDrawer



- Deklarisati DrawerLayout raspored kao koreni raspored

Slika 13: NavigationDrawer.

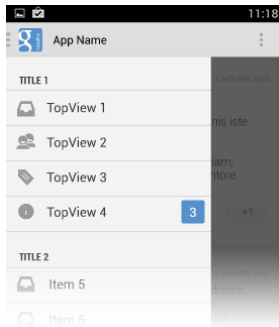
NavigationDrawer



Slika 13: NavigationDrawer.

- Deklarisati DrawerLayout raspored kao koreni raspored
- Dodati jedan pogled koji sadrži glavni sadržaj aktivnosti i drugi pogled koji sadrži sadržaj fioke za navigaciju

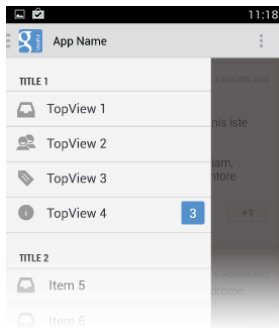
NavigationDrawer



Slika 13: NavigationDrawer.

- Deklarisati DrawerLayout raspored kao koreni raspored
- Dodati jedan pogled koji sadrži glavni sadržaj aktivnosti i drugi pogled koji sadrži sadržaj fioke za navigaciju
- Inicijalizovati fioku za navigaciju

NavigationDrawer



Slika 13: NavigationDrawer.

- Deklarisati DrawerLayout raspored kao koreni raspored
- Dodati jedan pogled koji sadrži glavni sadržaj aktivnosti i drugi pogled koji sadrži sadržaj fioke za navigaciju
- Inicijalizovati fioku za navigaciju
- Reagovati na akcije korisnika

layout_main.java

```
1 <android.support.v4.widget.DrawerLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:id="@+id/drawer_layout"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <!-- The main content view -->
8   <FrameLayout
9       android:id="@+id/content_frame"
10      android:layout_width="match_parent"
11      android:layout_height="match_parent" />
12
13   <!-- The navigation drawer -->
14   <ListView android:id="@+id/left_drawer"
15       android:layout_width="240dp"
16       android:layout_height="match_parent"
17       android:layout_gravity="start"
18       android:choiceMode="singleChoice"
19       android:divider="@android:color/transparent"
20       android:dividerHeight="0dp"
21       android:background="#111"/>
22
23 </android.support.v4.widget.DrawerLayout>
```

ExampleActivity.java

```

1  public class ExampleActivity extends Activity {
2
3      private String[] planetTitles;
4      private DrawerLayout drawerLayout;
5      private ListView drawerList;
6
7      @Override
8      public void onCreate(Bundle state) {
9          super.onCreate(state);
10         setContentView(R.layout.layout_main);
11
12         planetTitles = getResources().getStringArray(R.array.planets_array);
13         drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layout);
14         drawerList = (ListView) findViewById(R.id.left_drawer);
15
16         // Set the adapter for the list view
17         drawerList.setAdapter(new ArrayAdapter<String>(
18             this, R.layout.drawer_list_item, planetTitles));
19         // Set the list's click listener
20         drawerList.setOnItemClickListener(new DrawerItemClickListener());
21     }
22 }

```

ExampleActivity.java

```

1 private class DrawerItemClickListener implements ListView.OnItemClickListener {
2
3     @Override
4     public void onItemClick(AdapterView parent, View view, int position, long id) {
5         selectItem(position);
6     }
7
8     // Swaps fragments in the main content view
9     private void selectItem(int position) {
10         // Create a new fragment and specify the planet to show based on position
11         Fragment fragment = new PlanetFragment();
12         Bundle args = new Bundle();
13         args.putInt(PlanetFragment.ARG_PLANET_NUMBER, position);
14         fragment.setArguments(args);
15
16         // Insert the fragment by replacing any existing fragment
17         FragmentManager fragmentManager = getFragmentManager();
18         fragmentManager
19             .beginTransaction()
20             .replace(R.id.content_frame, fragment)
21             .commit();
22
23         // Highlight the selected item, update the title, and close the drawer
24         drawerList.setItemChecked(position, true);
25         setTitle(planetTitles[position]);
26         drawerLayout.closeDrawer(drawerList);
27     }
28
29     @Override
30     public void setTitle(CharSequence title) {
31         this.title = title;
32         getActionBar().setTitle(title);
33     }
34 }

```



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