

Resursi i konfiguracije uređaja

Android App Development

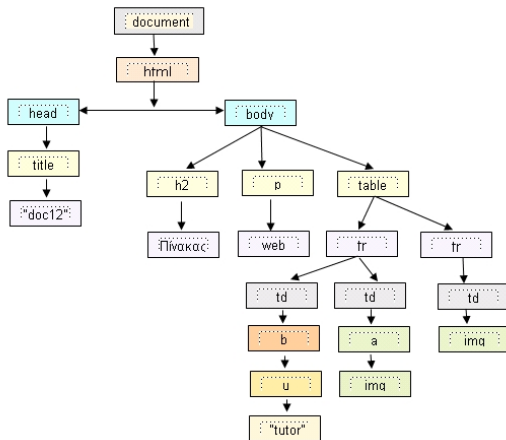
Pregled sadržaja

- 1 XML
- 2 Resursi
- 3 Konfiguracije uređaja
- 4 Prava pristupa
- 5 Varijante aplikacije

HTML dokument

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
2  <html>
3  <head>
4      <title>doc12</title>
5  </head>
6  <body>
7      <h2>Table</h2>
8      <p>web</p>
9      <table width="100%">
10         <tr>
11             <td><b>
12                 <u>tutor</u>
13             </b>
14         </td>
15         <td><a href="starweb.html">
16             
17         </a>
18         </td>
19     </tr>
20     <tr>
21         <td>
22     </td>
23 </tr>
24 </table>
25 </body>
26 </html>
```

HTML dokument



XML

- XML dokumenti ne sadrže unapred definisane elemente i attribute, nego ih autori sami definišu
- XML je metajezik, jezik za definisanje drugih jezika, skup pravila za definisanje drugih jezika
- HTML se može posmatrati kao jedan dijalekt XML-a (XHTML)

Struktura XML dokumenta

- sadrži čvorove različitog tipa uređene u stablo
- oznake su način da se hijerarhijska struktura serijalizuje u linearnu strukturu (niz znakova)

Tipovi čvorova

- elementi
- atributi
- tekst
- komentari
- reference na entitete
- itd.

Imena elemenata i atributa

- postoji razlika između velikih i malih slova
- mogu sadržati samo slova, cifre, donju crtdu, crticu, dvotačku i tačku
- moraju početi slovom ili donjom crtom
- ne mogu početi slovima `xml`

Elementi

- tag je tekstualna oznaka za početak i kraj elementa
- sadržaj elementa se nalazi između početnog (otvarajućeg) i krajnjeg (zatvarajućeg) taga
 - tekst
 - podelementi
 - mešavina teksta i podelemenata
 - prazan sadržaj

Elementi

```
1  <foo>bar</foo>
2
3
4  <foo>
5    <bar>qux</bar>
6    <baz>qux</baz>
7  </foo>
8
9
10 <foo>
11 bar <baz>qux</baz> bar
12 </foo>
13
14
15 <foo></foo>
16 <foo />
```

Atributi

- element može da ima attribute
- atribut ima ime i vrednost
- vrednost atributa je nestrukturiran tekst

Strukturiranje dokumenata

```
1  <name>
2    <first>Petar</first>
3    <last>Petrović</last>
4  </name>
5
6
7
8  <name first="Petar" last="Petrović" />
```

Komentari

- niz znakova između `<!--` i `-->`
- predstavljaju komentare ljudima
- programski se ne obrađuju

Reference na entitete i entiteti

Referenca na entitet	Entitet
<	<
&	&
>	>
"	”
'	'

Tabela 1: Reference na entitete i entiteti

Pregled sadržaja

- 1 XML
- 2 Resursi
- 3 Konfiguracije uređaja
- 4 Prava pristupa
- 5 Varijante aplikacije

Resursi

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

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>
```

strings.xml

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

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>
```

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
```

Resursi

Tip	Opis
<code>drawable</code>	vektorska ili rasterska grafika
<code>layout</code>	deklaracije grafičkog korisničkog interfejsa
<code>raw</code>	”sirovi” podaci (audio i video klipovi)
<code>values</code>	proste vrednosti (nizovi, boje, eksternalizovani stringovi, stilovi, itd.)
<code>xml</code>	XML dokumenti

Tabela 2: Tipovi resursa.

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`)

ExampleActivity.java

```
1  @Override
2  public void onCreate(Bundle savedInstanceState) {
3      super.onCreate(savedInstanceState);
4
5      setContentView(R.layout.main);
6  }
```

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>
```


Pregled sadržaja

- 1 XML
- 2 Resursi
- 3 Konfiguracije uređaja
- 4 Prava pristupa
- 5 Varijante aplikacije

Konfiguracije uređaja

- Postoji veliki broj uređaja (sa različitim hardverskim karakteristikama) koji koriste (različite verzije) Android platformu
- Konfiguracija uređaja apstrahuje 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), ali se mora voditi računa o redosledu sufiksa

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

Tabela 3: Vrste konfiguracije uređaja.

Konfiguracija uređaja

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

Konfiguracija uređaja

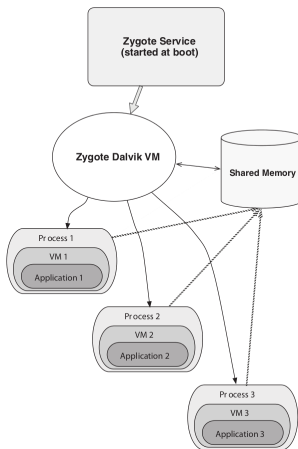
- Aplikacija bi uvek trebalo da sadrži podrazumevane resurse
- Za različite rezolucije ekrana bi trebalo pripremiti slike različitih rezolucija
- Za različite veličine ili orijentacije ekrana bi trebalo pripremiti različite rasporede GUI-a
- Treba internacionalizovati stringove da bi se omogućila lokalizacija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

Pregled sadržaja

- 1 XML
- 2 Resursi
- 3 Konfiguracije uređaja
- 4 Prava pristupa
- 5 Varijante aplikacije

Prava pristupa

Operativni sistem izoluje aplikacije (kako aplikacije međusobno tako i operativni sistem od aplikacija).



Prava pristupa

- Dodatne funkcije bezbednosti su implementirane mehanizmom prava pristupa.
- Aplikacija ne može da izvrši ni jednu operaciju koja može negativno da utiče na druge aplikacije, operativni sistem ili korisnike ukoliko joj to nije dozvoljeno
- Implementacija mehanizma prava pristupa razlikuje se do Androida 6.0 (statička prava pristupa) i od Androida 6.0 pa dalje (dinamička prava pristupa)

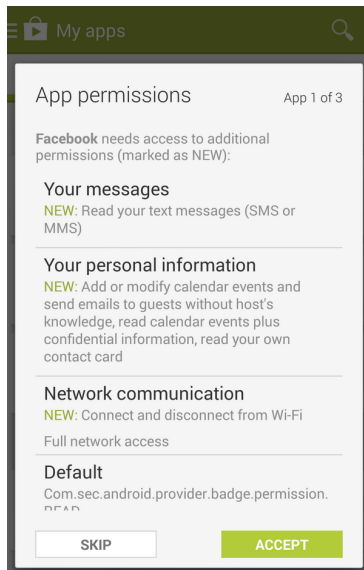
Prava pristupa

- Do Androida 6.0 prava pristupa koja su potrebna za izvršavanje aplikacije statički se deklarišu u AndroidManifest.xml.
- Korisnik može da aplikaciji prilikom instalacije dodeli prava pristupa koja traži ili da odustane od instalacije aplikacije
- Svaki pokušaj da aplikacija izvrši nedozvoljene operacije biće sprečen

Prava pristupa

```
1 <manifest ... >
2     ...
3     <uses-permission android:name="android.permission.INTERNET" />
4     ...
5 </manifest>
```

Prava pristupa



Prava pristupa

- Od Androida 6.0 pa dalje aplikacija dinamički traži prava pristupa koja su joj potrebna za izvršavanje
- Android može automatski odobriti aplikaciji pravo pristupa ili može zatražiti od korisnika da joj odobri pravo pristupa (u zavisnosti od osetljivosti operacije i resursa)
- Korisnik ima mogućnost da aplikaciji u svakom trenutku oduzme pravo pristupa
- To znači da aplikacija mora da svaki put pre nego što izvrši operaciju koja zahteva pravo pristupa proveriti da li ima to pravo

Prava pristupa

```
1  // Assume thisActivity is the current activity
2  int permissionCheck = ContextCompat.checkSelfPermission(thisActivity,
3      Manifest.permission.WRITE_CALENDAR);
```

Prava pristupa

```

1  // Here, thisActivity is the current activity
2  if (ContextCompat.checkSelfPermission(thisActivity,
3      Manifest.permission.READ_CONTACTS)
4      != PackageManager.PERMISSION_GRANTED) {
5
6      // Should we show an explanation?
7      if (ActivityCompat.shouldShowRequestPermissionRationale(thisActivity,
8          Manifest.permission.READ_CONTACTS)) {
9
10         // Show an explanation to the user *asynchronously* -- don't block
11         // this thread waiting for the user's response! After the user
12         // sees the explanation, try again to request the permission.
13
14     } else {
15
16         // No explanation needed, we can request the permission.
17
18         ActivityCompat.requestPermissions(thisActivity,
19             new String[] {Manifest.permission.READ_CONTACTS},
20             MY_PERMISSIONS_REQUEST_READ_CONTACTS);
21
22         // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
23         // app-defined int constant. The callback method gets the
24         // result of the request.
25     }
26 }

```

Prava pristupa

```

1  @Override
2  public void onRequestPermissionsResult(int requestCode,
3      String permissions[], int[] grantResults) {
4      switch (requestCode) {
5          case MY_PERMISSIONS_REQUEST_READ_CONTACTS: {
6              // If request is cancelled, the result arrays are empty.
7              if (grantResults.length > 0
8                  && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
9
10                 // permission was granted, yay! Do the
11                 // contacts-related task you need to do.
12
13             } else {
14
15                 // permission denied, boo! Disable the
16                 // functionality that depends on this permission.
17             }
18             return;
19         }
20
21         // other 'case' lines to check for other
22         // permissions this app might request
23     }
24 }

```

Prava pristupa

Constant	Meaning
CALL_PHONE	Allows an application to initiate a phone call.
SEND_SMS	Allows an application to send SMS messages.
RECORD_AUDIO	Allows an application to record audio.
CAMERA	Required to be able to access the camera device.
VIBRATE	Allows access to the vibrator.

Tabela 4: Prava pristupa.

Prava pristupa

Constant	Meaning
ACCESS_FINE_LOCATION	Allows an app to access precise location from location sources such as GPS, cell towers, and Wi-Fi.
INTERNET	Allows applications to open network sockets.
BLUETOOTH	Allows applications to connect to paired bluetooth devices.
NFC	Allows applications to perform I/O operations over NFC.

Tabela 5: Prava pristupa.

Pregled sadržaja

- 1 XML
- 2 Resursi
- 3 Konfiguracije uređaja
- 4 Prava pristupa
- 5 Varijante aplikacije

Varijante izgradnje aplikacije

- Android aplikacije se podrazumevano mogu izgraditi u dve varijante: **debug** i **release**.
- Debug varijanta služi programeru tokom razvoja aplikacije.
- Release varijanta je namenjena krajnjim korisnicima.
- Izgradnjom ovih varijanti projekta upravlja Gradle.
- Osim pomenutih varijanti, Gradle skriptom je moguće konfigurisati i tzv. dimenzije.
- U jednoj dimenziji se može razlikovati više "aroma"(eng. flavors).

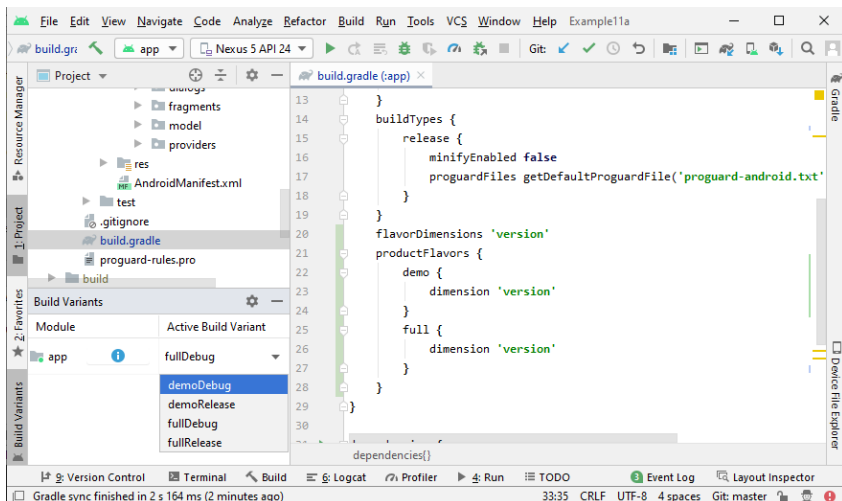
Build flavors

- Primer definisanja dimenzije i njenih flavor-a u gradle skripti:

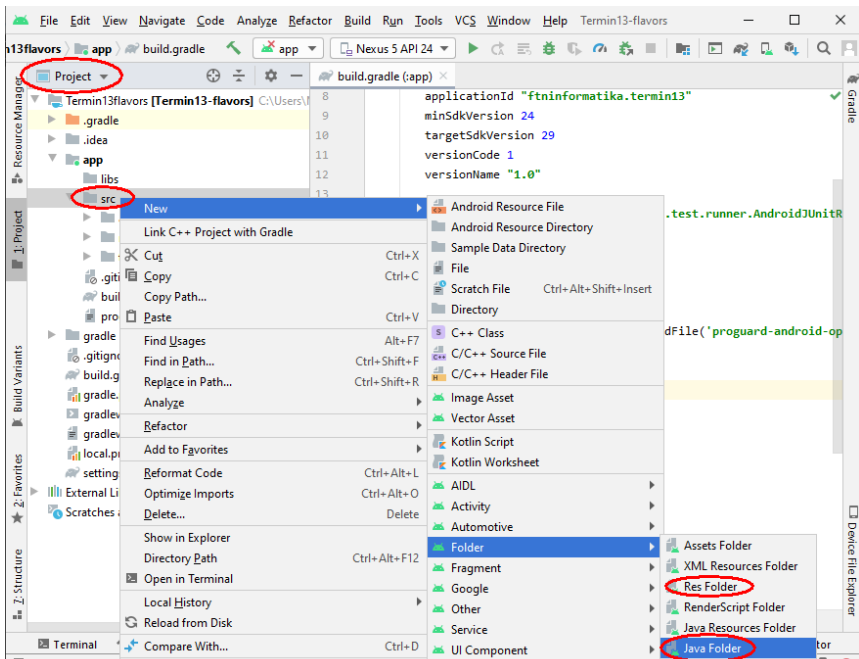
```
android {  
    ...  
    flavorDimensions 'version'  
    productFlavors {  
        demo {  
            dimension 'version'  
        }  
        full {  
            dimension 'version'  
        }  
    }  
}
```

- Dimenzije i flavor-e je moguće definisati i tako što se iz menija izabere:
 - Build
 - Edit Flavors...
 - Add Flavor Dimension
 - Add Product Flavor

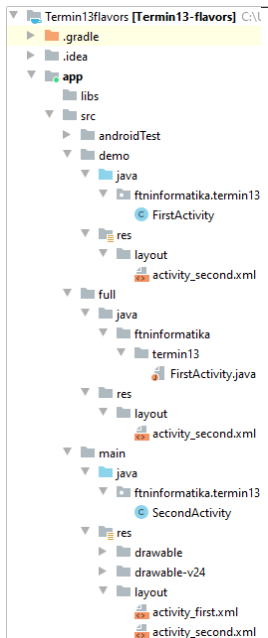
- Nakon dodavanja flavor-a, u prozoru 'Build Variants' (sa leve strane) će biti ponuđene sve varijante aplikacije.
- Izbor u ovoj padajućoj listi određuje koja varijanta aplikacije će se izgraditi prilikom pokretanja aplikacije.



- U okviru projekta je potrebno razgraničiti koje funkcionalnosti (tj. programski kôd) i koji sadržaji (tj. resursi) pripadaju kojoj varijanti.
- Za to koristimo zasebne foldere u projektu dok `main` folder ostaje zajednički.
- Ove foldere kreiramo tako što:
 - Za prikaz strukture projekta izaberemo `Project`.
 - Nad folderom `src` kreiramo novi `Java` folder i novi `Res` folder, najpre za jedan flavor, a potom i za drugi flavor.
- Putanje ovih foldera bi trebalo da glase:
 - `src/demo/java`
 - `src/demo/res`
 - `src/full/java`
 - `src/full/res`



- Nakon izgradnje, u demo verziji će se nalaziti:
 - programski kôd = `src/main/java` + `src/demo/java`
 - resursi = `src/main/res` + `src/demo/res`
- Slično, u full verziji će se nalaziti:
 - programski kôd = `src/main/java` + `src/full/java`
 - resursi = `src/main/res` + `src/full/res`
- Može se reći da verzije nastaju udruživanjem sadržaja `main` foldera i foldera te verzije.
- Pri tome treba voditi računa da pri ovom udruživanju resursi koji pod istim nazivom postoje u `main` folderu i folderu neke verzije rezultuju resursom iz foldera te verzije (odn. flavor folder će preklopiti `main` folder).
- Za programski kôd ovo ne važi odn. u `main` folderu i folderu neke verzije se ne mogu nalaziti istoimene klase.
- Za programski kôd je potrebno voditi računa i o tome da se u svim verzijama koristi jedinstven način nazivanja paketa.



- U prikazanoj strukturi projekta se nalaze dve aktivnosti pri čemu jedna aktivnost ima različitu funkcionalnost u zavisnosti od verzije, a druga različit izgled u zavisnosti od verzije.
- To je postignuto tako što:
 - Svaka verzija ima svoju klasu `FirstActivity` (ta klasa ne sme postojati i u main folderu)
 - Resurs `activity_first.xml` se nalazi samo u main folderu
 - Klasa `SecondActivity` se nalazi samo u main folderu
 - Svaka verzija ima svoj resurs `activity_second.xml` (on se nalazi i u main folderu)