



MOBILNE APLIKACIJE

Vežbe 10

Multimedija

2022/2023

## Sadržaj

1. Snimanje zvuka .....	3
2. Reprodukcijs zvuka .....	6
3. Snimanje fotografije i video zapisa .....	7
4. Primer .....	10

## 1. Snimanje zvuka

Android platforma omogućava snimanje zvuka posredstvom mikrofona. Snimanje zvuka je moguće samo na fizičkim uređajima, a emulator nema tu mogućnost.

*MediaRecorder* je API za snimanje različitih audio i video formata.

Da bismo snimili zvuk potrebno je da izvršimo sledeće korake:

- Instancirati *MediaRecorder*.
- Postaviti audio izvor korišćenjem *setAudioSource* metode.
- Postaviti izlazni format korišćenjem *setOutputFormat* metode.
- Postaviti ime izlazne datoteke korišćenjem *setOutputFile* metode.
- Postaviti audio koder korišćenjem *setAudioEncoder* metode.
- Pozvati *prepare* metodu.
- Pozvati *start* metodu za početak snimanja zvuka.
- Pozvati *stop* metodu za završetak snimanja zvuka.
- Osloboditi instancu *MediaRecorder*-a pozivanjem *release* metode.

Za demonstriranje ovog primera kreirali smo klasu *MainActivity*. U njenoj *onCreate* metodi (slika 1) postavljamo naš layout *activity\_main.xml* (slika 2) i tražimo da se odobri pravo pristupa za snimanje audio zapisa. U drugom delu ove metode dobavljamo dugmiće sa *layout*-a i kreiramo putanju na kojoj će biti sačuvan audio zapis.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

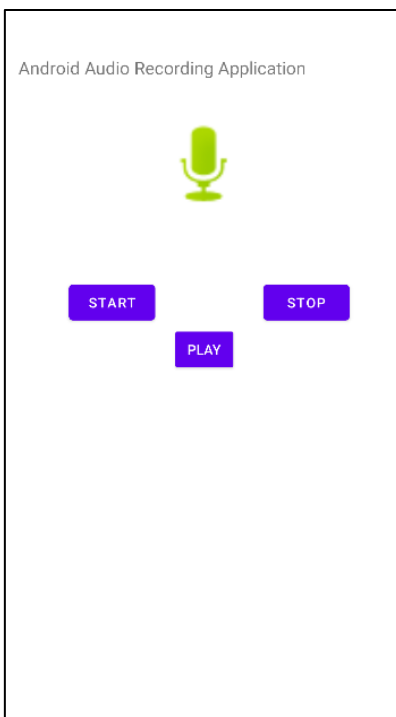
    ActivityCompat.requestPermissions( activity: this, permissions, REQUEST_RECORD_AUDIO_PERMISSION);

    start = (Button) findViewById(R.id.button1);
    stop = (Button) findViewById(R.id.button2);
    play = (Button) findViewById(R.id.button3);

    stop.setEnabled(false);
    play.setEnabled(false);
    outputFile = getExternalCacheDir().getAbsolutePath() + "/myrecording.3gp";

    myAudioRecorder = new MediaRecorder();
}
```

Slika 1. Metoda *onCreate*



Slika 2. *activity\_main.xml*

U *AndroidManifest* datoteci definisali smo i statička prava pristupa (slika 3) za snimanje zvuka (*RECORD\_AUDIO*) i za pisanje u eksterno skladište (*WRITE\_EXTERNAL\_STORAGE*).

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECORD_AUDIO" />
```

Slika 3. Definisanje statičkih prava pristupa

Metoda *start* se poziva kada korisnik klikne na dugme *start*. U ovoj metodi (slika 4) nad objektom klase *MediaRecorder* pozivamo metode:

- *setAudioSource* - postavljamo audio izvor.
- *setOutputFormat* - postavljamo izlazni format.
- *setAudioEncoder* - postavljamo audio koder.
- *setOutputFile* - postavljamo ime izlazne datoteke.

Inicijalizaciju završavamo pozivom metode *prepare*, nakon čega pozivamo metodu *start* za početak snimanja zvuka. Na samom kraju metode omogućavamo korisniku da klikne dugme *stop*, a dugme *start* postaje *disabled*.

```

public void start(View view) {
    try {
        myAudioRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
        myAudioRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
        myAudioRecorder.setAudioEncoder(MediaRecorder.OutputFormat.AMR_NB);
        myAudioRecorder.setOutputFile(outputFile);

        myAudioRecorder.prepare();
        myAudioRecorder.start();
    } catch (IllegalStateException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    start.setEnabled(false);
    stop.setEnabled(true);
    Toast.makeText(getApplicationContext(), text: "Recording started",
        Toast.LENGTH_LONG).show();
}

```

Slika 4. Metoda *start*

Kada korisnik klikne na dugme *stop*, poziva se *stop* metoda, koja oslobađa instancu *MediaRecorder*-a pozivanjem *release* metode (slika 5).

```

public void stop(View view) {
    myAudioRecorder.stop();
    myAudioRecorder.release();
    myAudioRecorder = null;
    stop.setEnabled(false);
    play.setEnabled(true);
    Toast.makeText(getApplicationContext(), text: "Audio recorded successfully",
        Toast.LENGTH_LONG).show();
}

```

Slika 5. Metoda *stop*

## 2. Reprodukcijska zvuka

Android platforma omogućava reprodukciju audio i video sadržaja u različitim formatima i preko različitih mrežnih protokola. Moguće je reprodukovati sadržaj iz datoteka koje su skladištene kao resursi, datoteka u internom ili eksternom skladištu podataka ili iz toka podataka, koji stiže preko mreže.

Klase koje koristimo za puštanje audio ili video sadržaja su:

- *MediaPlayer*
- *AudioManager*

*MediaPlayer* je API za reprodukciju audio i video sadržaja.

*AudioManager* upravlja audio izvorima i audio izlazom.

Korisnik može da reprodukuje audio zapis koji je prethodno snimio. Metoda *play* se poziva kada korisnik klikne na dugme *play* (slika 6). U toj metodi kreira se objekat klase *MediaPlayer* i postavlja mu se putanja do zvuka koji treba da se pusti, pozivanjem metode *setDataSource*. Kao i kod *MediaRecorder*-a, sledi pozivanje metode *prepare* i *start*.

```
public void play(View view) throws IllegalArgumentException,
    SecurityException, IllegalStateException, IOException {

    MediaPlayer m = new MediaPlayer();
    m.setDataSource(outputFile);
    m.prepare();
    m.start();
    Toast.makeText(getApplicationContext(), text: "Playing audio",
        Toast.LENGTH_LONG).show();
}
```

Slika 6. Reprodukcijska zvuka

### 3. Snimanje fotografije i video zapisa

Snimanje fotografija i video zapisa je moguće korišćenjem postojeće aplikacije (*Camera*). Kao i kod snimanja zvuka, koristimo *MediaRecorder*. Za snimanje fotografija i videa korišćenjem *MediaRecorder* API-a potrebno je izvršiti sledeće korake:

- Detektovati kameru.
- Pristupiti kameri.
- Napraviti *Preview* klasu.
- Napraviti *Preview* raspored.
- Podesiti obrađivače događaja.
- Snimiti fotografiju ili video u datoteku.
- Osloboditi kameru.

U *AndroidManifest* datoteci dodajemo novo pravo pristupa (CAMERA).

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
```

Slika 7. Definisanje statičkih prava pristupa za kameru

U metodi *onCreate* (slika 8), klase *MainActivity*, postavljamo naš layout *activity\_main.xml* i dobavljamo elemente sa *layout*-a. Potom vršimo proveru da li su permisije odobrene i ako nisu tražimo da se odobre.

```
33  @Override
34  protected void onCreate(@Nullable Bundle savedInstanceState) {
35      super.onCreate(savedInstanceState);
36      setContentView(R.layout.activity_main);
37
38      takePictureButton = (Button) findViewById(R.id.button_image);
39      imageView = (ImageView) findViewById(R.id.imageview);
40
41      startActivityResult = registerForActivityResult(
42          new ActivityResultContracts.StartActivityForResult(),
43          result -> {
44              if (result.getResultCode() == AppCompatActivity.RESULT_OK) {
45                  Intent data = result.getData();
46                  imageView.setImageURI(file);
47              }
48          }
49      );
50      if (ContextCompat.checkSelfPermission(context, this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
51          takePictureButton.setEnabled(false);
52          ActivityCompat.requestPermissions(activity, this, new String[] { Manifest.permission.CAMERA, Manifest.permission.WRITE_EXTERNAL_STORAGE }, requestCode: 0);
53      }
54  }
```

Slika 8. Metoda *onCreate*

Kada korisnik želi da uslika potrebno je da klikne na dugme *Take a picture*, koje će pozvati metodu *takePicture* (slika 9). U toj metodi instanciramo nameru, startujemo aktivnost i primamo rezultat od aktivnosti.

*GenericFileProvider* je klasa koju smo napravili i ona nasleđuje *FileProvider*.

EXTRA\_OUTPUT je URI koji određuje putanju i ime datoteke u koju će se sačuvati fotografija (ili video).

Ovo je primer dobijanja slike u punoj rezoluciji kamere.

```
82 public void takePicture(View view) {
83     Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
84     intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
85
86     File file = FileProvider.getUriForFile(context, this, GenericFileProvider.MY_PROVIDER, getOutputMediaFile());
87     intent.putExtra(MediaStore.EXTRA_OUTPUT, file);
88
89     startActivityForResult.launch(intent);
90 }
91
```

Slika 9. Metoda *takePicture*

Android *Camera* aplikacija može da sačuva fotografije ako joj postavimo putanju na kojoj treba da ih čuva. Uglavnom se fotografije, koje korisnik napravi, čuvaju na uređaju u javnom eksternom skladištu, tako da svaka aplikacija može da im pristupi. U svrhu generisanja putanje, napravljena je pomoćna metoda *getOutputMediaFile*.

Odgovarajući direktorijum za smeštanje deljenih fotografija se dobija pozivanjem metode *getExternalStoragePublicDirectory* i prosleđivanjem konstante *DIRECTORY\_PICTURES* (slika 10).

Pošto naziv datoteke (putanja) mora biti jedinstven, na naziv direktorijuma dodajemo datum i vreme kada je fotografija napravljena.

```
67 @ private static File getOutputMediaFile(){
68     File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(
69         Environment.DIRECTORY_PICTURES), child: "CameraDemo");
70
71     if (!mediaStorageDir.exists()){
72         if (!mediaStorageDir.mkdirs()){
73             return null;
74         }
75     }
76
77     String timeStamp = new SimpleDateFormat(pattern: "yyyyMMdd_HHmmss").format(new Date());
78     return new File(pathname: mediaStorageDir.getPath() + File.separator +
79         "IMG_" + timeStamp + ".jpg");
80 }
81
```

Slika 10. Pomoćna metoda *getOutputMediaFile*

U *AndroidManifest* datoteci definisali smo i *FileProvider*, a prethodno i neophodna prava pristupa *WRITE\_EXTERNAL\_STORAGE*.



```
<provider
    android:name="rs.reviewer.GenericFileProvider"
    android:authorities="rs.reviewer.provider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_paths"/>
</provider>
```

Slika 11. Definisanje *FileProvider*-a u *AndroidManifest* datoteci

## 4. Primer

Domaći se nalazi na *Canvas*-u (*canvas.ftn.uns.ac.rs*) na putanji *Вежбе/10 Вежбе/10 Задачах.pdf*.

Primer možete preuzeti na sledećem linku: <https://gitlab.com/antesevicceca/mobilne-aplikacije>.

Za dodatna pitanja možete se obratiti asistentima:

- Svetlana Antešević ([svetlanaantesevic@uns.ac.rs](mailto:svetlanaantesevic@uns.ac.rs))
- Jelena Matković ([matkovic.jelena@uns.ac.rs](mailto:matkovic.jelena@uns.ac.rs))