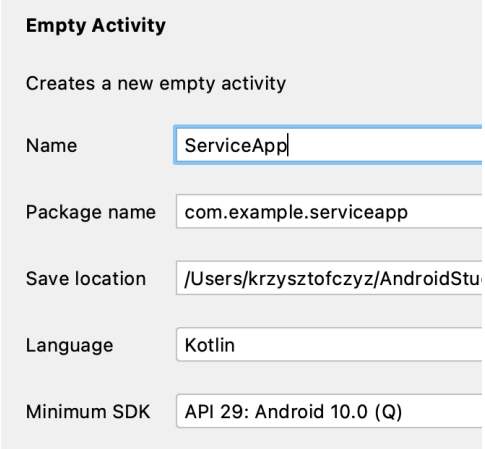
Usługi (services)

Usługi są wykonywane w tle. Przykład: zegar pracuje w czasie, gdy oglądamy film. Uruchamiamy nowy projekt:



Uruchamiamy viewBinding:

MainActivity.kt | build.gradle (:app)

```kotlin
package com.example.serviceapp

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import com.example.serviceapp.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
}
```

Oraz projektujemy nasz interfejs xml w pliku activity_main.xml :

activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".MainActivity">
```

```
10
11      <Button
12          android:id="@+id/btn_start"
13          android:layout_width="wrap_content"
14          android:layout_height="wrap_content"
15          android:layout_margin="10dp"
16          android:text="Start" />
17
18      <Button
19          android:id="@+id/btn_stop"
20          android:layout_width="wrap_content"
21          android:layout_height="wrap_content"
22          android:layout_margin="10dp"
23          android:text="Stop" />
24
25  </LinearLayout>
```

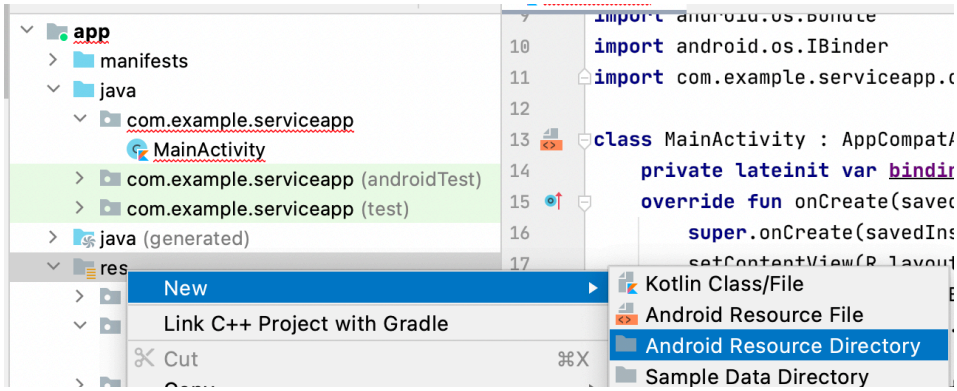
Dodajemy klasę MyService:

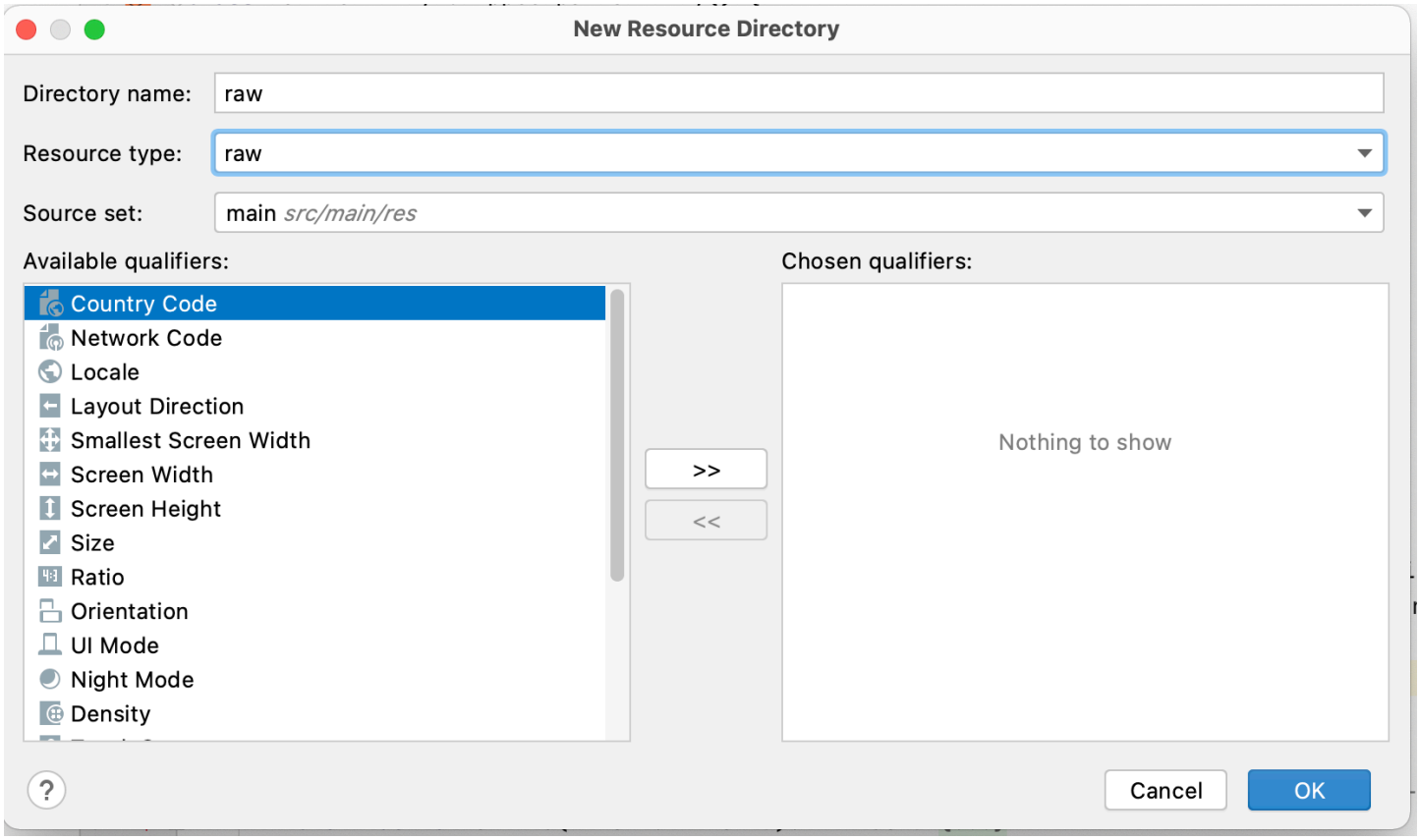
```
10  class MainActivity : AppCompatActivity() {
11      private lateinit var binding: ActivityMainBinding
12      override fun onCreate(savedInstanceState: Bundle?) {
13          super.onCreate(savedInstanceState)
14          setContentView(R.layout.activity_main)
15          binding = ActivityMainBinding.inflate(layoutInflater)
16          setContentView(binding.root)
17      }
18      class MyService: Service(){
19          override fun onBind(intent: Intent?): IBinder? {
20              return null
21          }
22      }
23  }
```

Rozbudowujemy klasę o zmienne i dodatkowe metody:

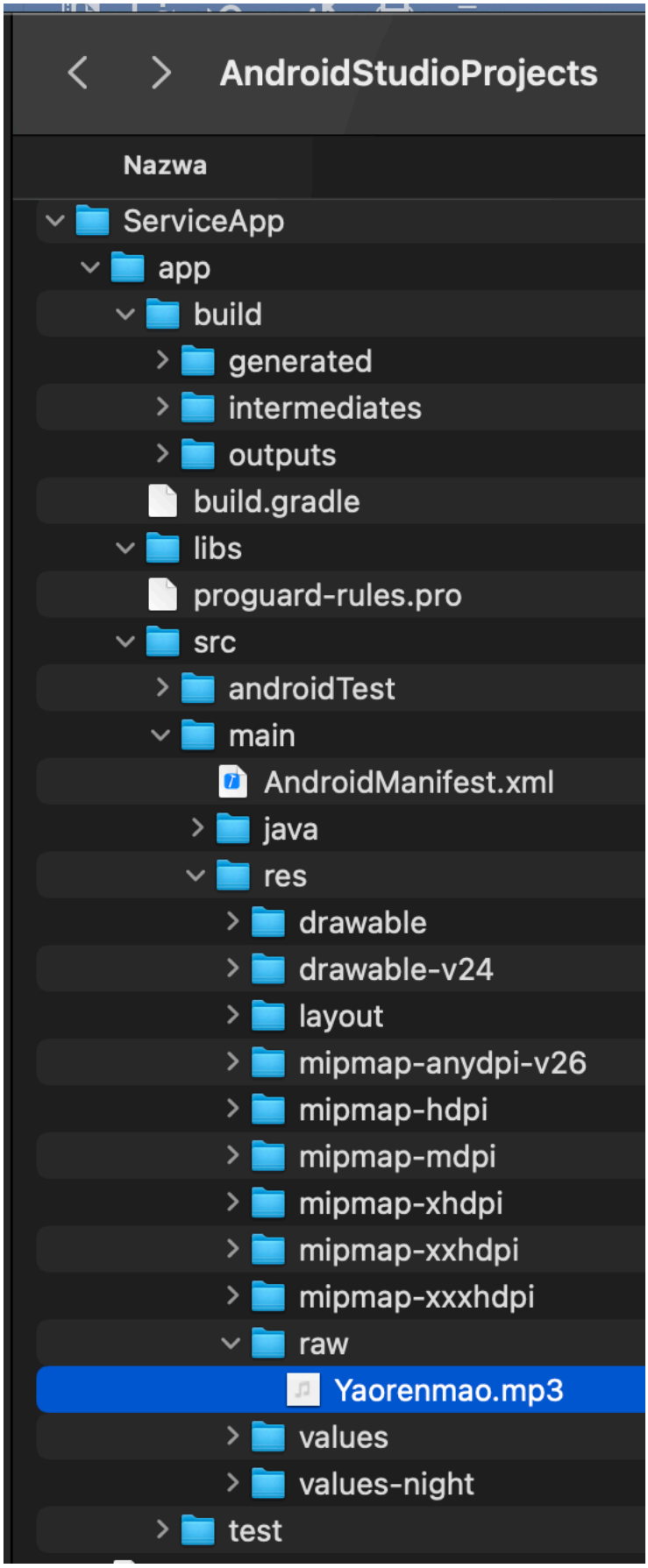
```
19      class MyService: Service(){
20          private lateinit var mediaPlayer: MediaPlayer
21          private var isPlay: Boolean = false
22          //-- metoda uruchamiająca nasz service
23          override fun onCreate() {
24              super.onCreate()
25          }
26          //-- metoda odnawiająca działanie servivce
27          override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
28              return super.onStartCommand(intent, flags, startId)
29          }
```

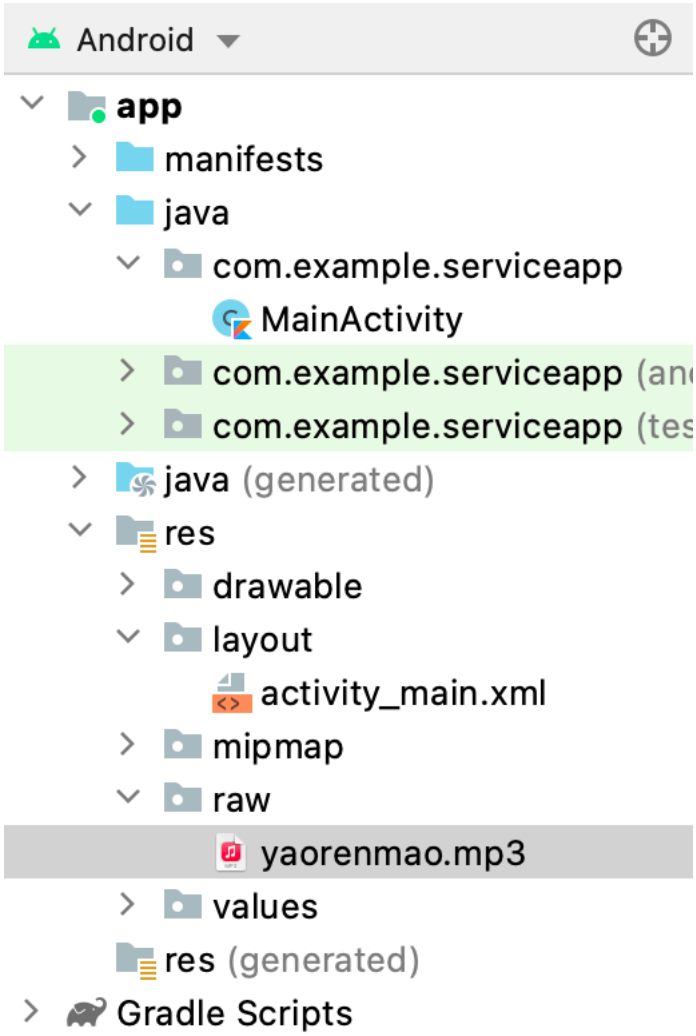
Dodajemy katalog na muzykę:





Teraz do folderu kopiujemy mp3-kę i na nowo uruchamiamy aplikację:





Warto zastosować małe litery w nazwach plików.

Tworzymy notyfikację (powiadomienie) w metodzie onStartCommand oraz metodę wyłączającą nase zdarzenie onDestroy:

MainActivity.kt

```
21      class MyService: Service(){                                    ⚠ 1  ✓ 18
22          private lateinit var mediaPlayer: MediaPlayer
23          private var isPlay: Boolean = false
24          //-- metoda uruchamiająca nasz service
25          override fun onCreate() {...}
28          //-- metoda odnawiająca działanie servivce
29          override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
30              if(!isPlay){
31                  val notificationIntent = Intent( packageContext: this,MainActivity::class.java)
32                  val pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0,
    notificationIntent, flags: 0)
33                  val notification = Notification.Builder( context: this, channelId: "Test")
    .setContentTitle("Przykład usługi").setContentText("Działa wq tle").setContentIntent
    (pendingIntent).build()
34                  mediaPlayer = MediaPlayer.create(applicationContext,R.raw.yaorenmao)
35                  mediaPlayer.start()
36                  isPlay = true
37                  startForeground( id: 1,notification)
38              }
39              return START_NOT_STICKY
40          }
41
42          override fun onDestroy() {
43              mediaPlayer.stop()
44              isPlay = false
45              super.onDestroy()
46          }
```

Teraz oprogramowujemy działanie przycisków:

MainActivity.kt

```
import android.app.Notification
import android.app.PendingIntent
import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.IBinder
import com.example.serviceapp.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
        val intent = Intent(applicationContext,MyService::class.java)
        binding.btnStart.setOnClickListener {  it: View!
            startService(intent)
        }
        binding.btnStop.setOnClickListener {  it: View!
            stopService(intent)
        }
    }
    class MyService: Service(){...}
}
```

W pliku Manifest musimy umożliwić uruchomienie naszej usługi:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.serviceapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="ServiceApp"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ServiceApp">
        <activity android:name=".MainActivity"...>
        <service android:name=".MainActivity$MyService"
            android:exported="false" />
    </application>

</manifest>
```

Dodajemy klasę tworzącą kanał

MainActivity.kt

```
class MyService: Service(){...}
class App : Application(){
    companion object{
        const val CHANNEL_ID = "Przykład"
    }
    override fun onCreate() {
        super.onCreate()
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
            val serviceChannel = NotificationChannel(CHANNEL_ID,
    name: "ExampleServiceChannel",NotificationManager.IMPORTANCE_DEFAULT)
            val mng = getSystemService(NotificationManager::class.java)
            mng.createNotificationChannel(serviceChannel)
        }
    }
}
}
```

W pliku Manifest dodajemy naszą nową klasę oraz zezwolenie na użycie kanału powiadomień:

MainActivity.kt | AndroidManifest.xml

```
<?xml v                        3"?>
<manife   Close. Alt-Click to Close Others   hemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.serviceapp">
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <application
        android:name=".MainActivity$App"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
```

Ostatnią rzeczą jest uruchomienie naszego kanału powiadomień w wywołaniu aplikacji:

MainActivity.kt

```
    }
    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        if(!isPlay){
            val notificationIntent = Intent( packageContext: this,MainActivity::class.java)
            val pendingIntent = PendingIntent.getActivity( context: this, requestCode: 0,
notificationIntent, flags: 0)
            val notification = NotificationCompat.Builder( context: this,CHANNEL_ID)
.setContentTitle("Przykład usługi").setContentText("Działa w tle").setContentIntent
(pendingIntent).build()
```

Można sprawdzić działanie przycisków.