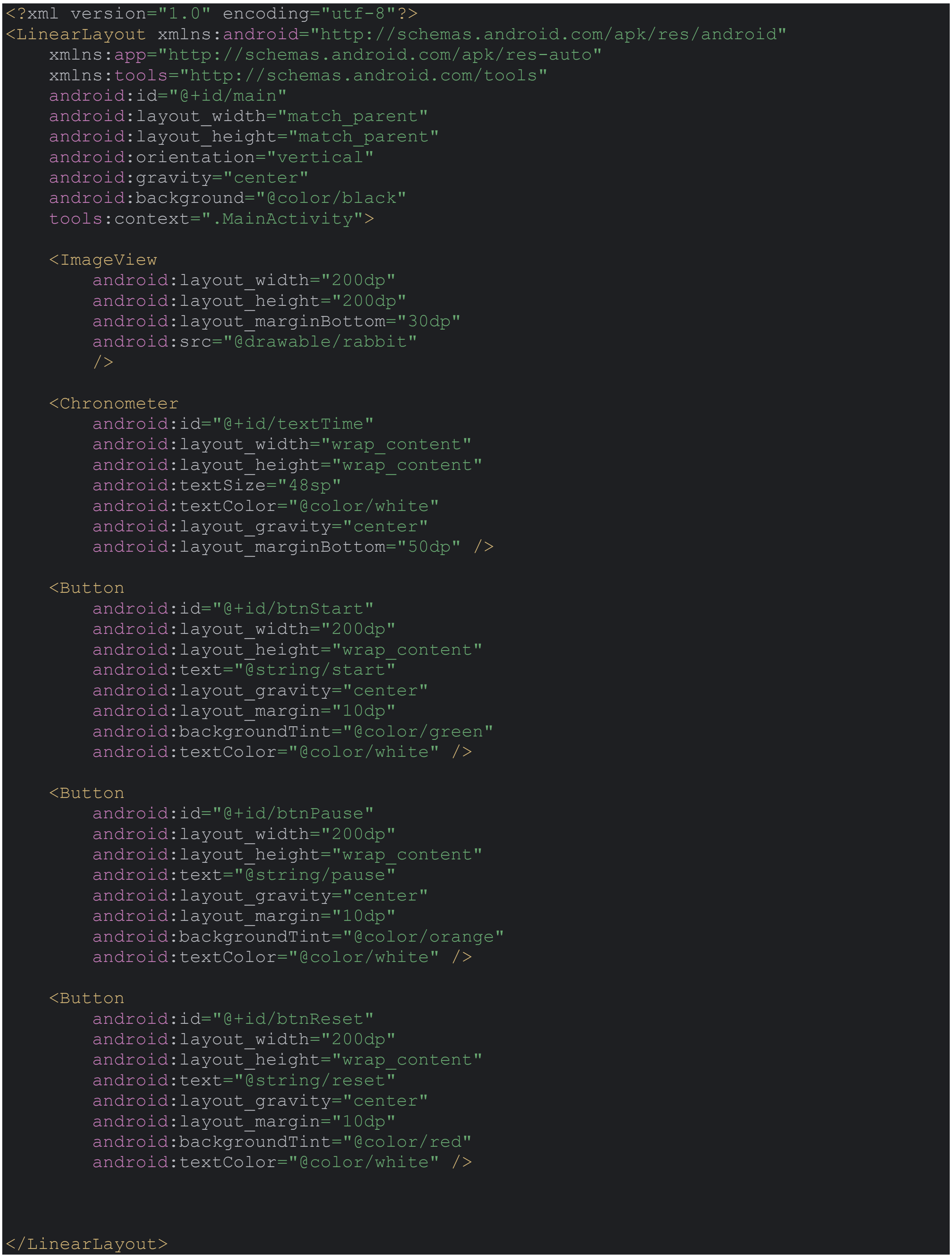
Лабораторная Работа № 3

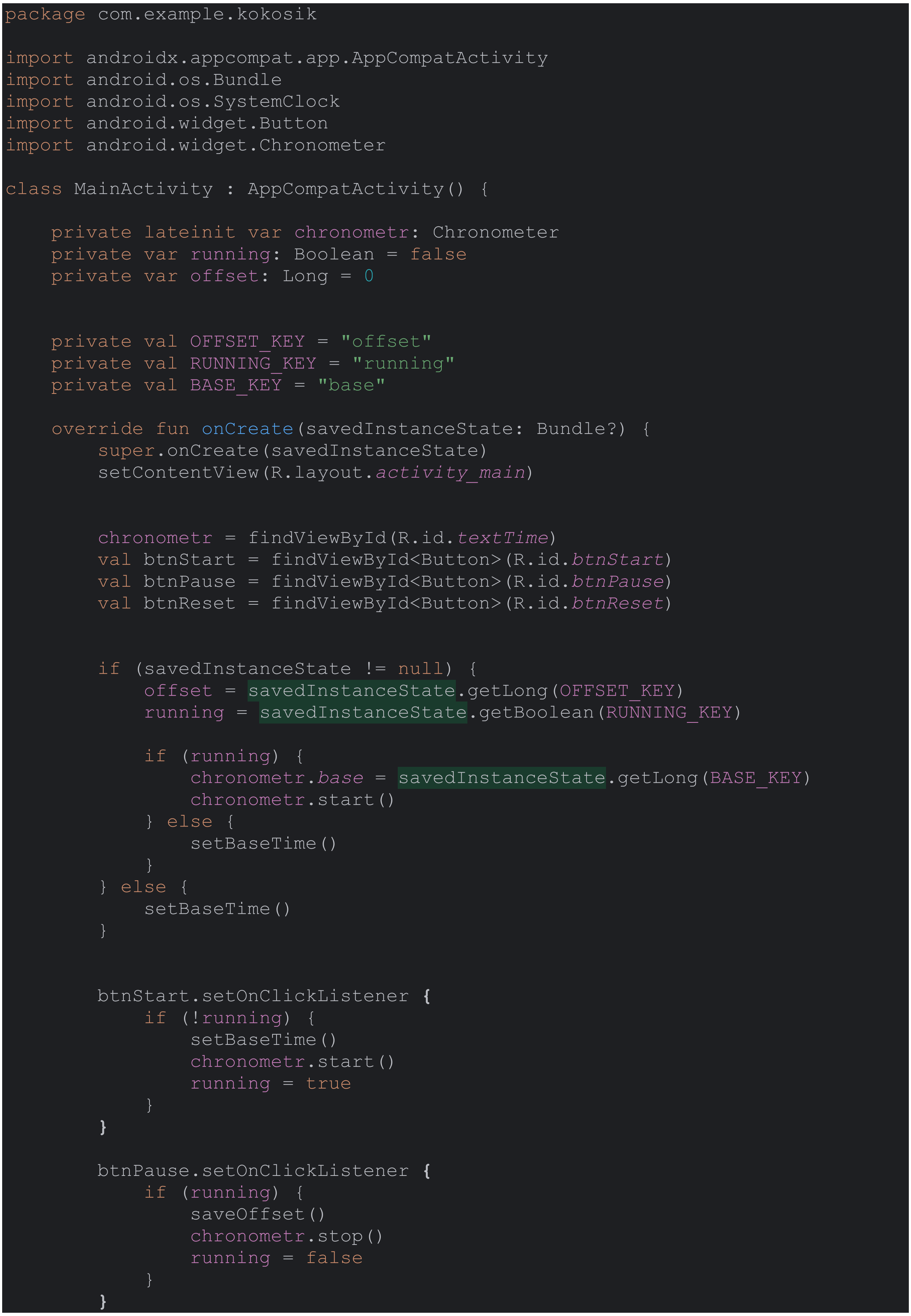
Секундомер

Activity.main.xml:

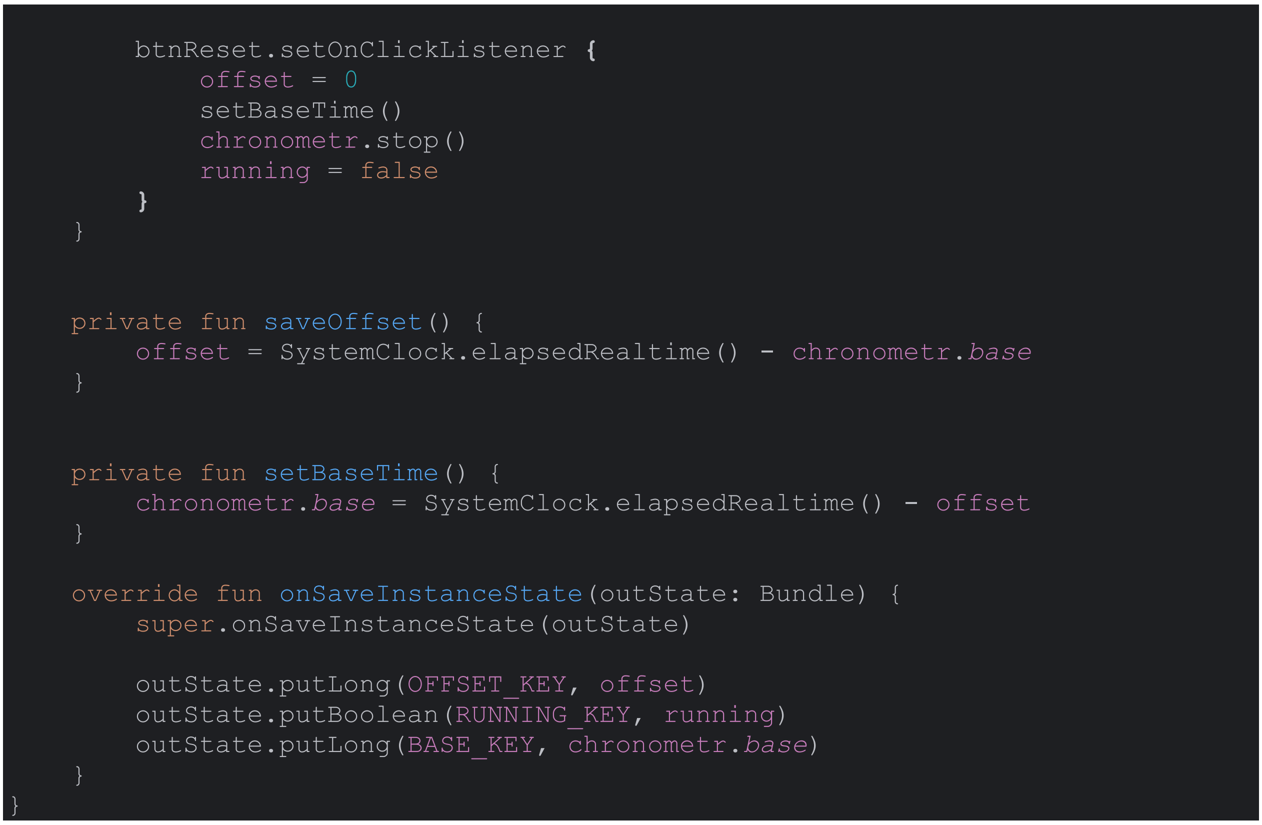


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@color/black"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_marginBottom="30dp"
        android:src="@drawable/rabbit"
        />

    <Chronometer
        android:id="@+id/textTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="48sp"
        android:textColor="@color/white"
        android:layout_gravity="center"
        android:layout_marginBottom="50dp" />

    <Button
        android:id="@+id/btnStart"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/start"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:backgroundTint="@color/green"
        android:textColor="@color/white" />

    <Button
        android:id="@+id/btnPause"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/pause"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:backgroundTint="@color/orange"
        android:textColor="@color/white" />

    <Button
        android:id="@+id/btnReset"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:text="@string/reset"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:backgroundTint="@color/red"
        android:textColor="@color/white" />



</LinearLayout>
```

Main.Activity.kt

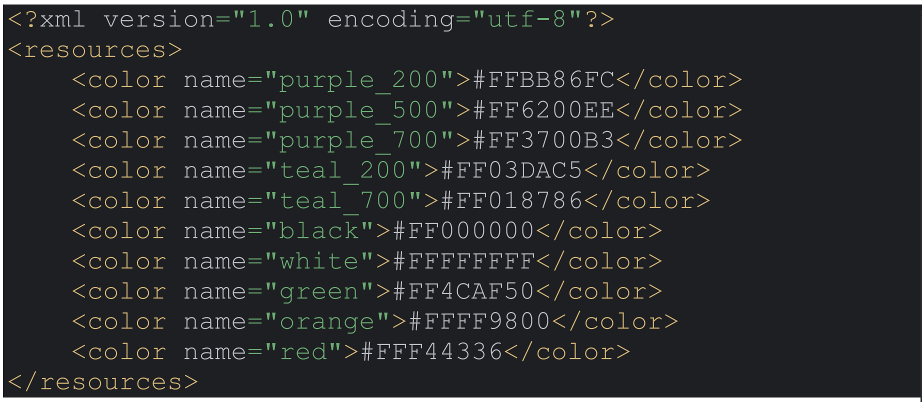


```
package com.example.kokosik

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.SystemClock
import android.widget.Button
import android.widget.Chronometer

class MainActivity : AppCompatActivity() {

    private lateinit var chronometr: Chronometer
    private var running: Boolean = false
    private var offset: Long = 0


    private val OFFSET_KEY = "offset"
    private val RUNNING_KEY = "running"
    private val BASE_KEY = "base"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)


        chronometr = findViewById(R.id.textTime)
        val btnStart = findViewById<Button>(R.id.btnStart)
        val btnPause = findViewById<Button>(R.id.btnPause)
        val btnReset = findViewById<Button>(R.id.btnReset)


        if (savedInstanceState != null) {
            offset = savedInstanceState.getLong(OFFSET_KEY)
            running = savedInstanceState.getBoolean(RUNNING_KEY)

            if (running) {
                chronometr.base = savedInstanceState.getLong(BASE_KEY)
                chronometr.start()
            } else {
                setBaseTime()
            }
        } else {
            setBaseTime()
        }


        btnStart.setOnClickListener {
            if (!running) {
                setBaseTime()
                chronometr.start()
                running = true
            }
        }

        btnPause.setOnClickListener {
            if (running) {
                saveOffset()
                chronometr.stop()
                running = false
            }
        }
```

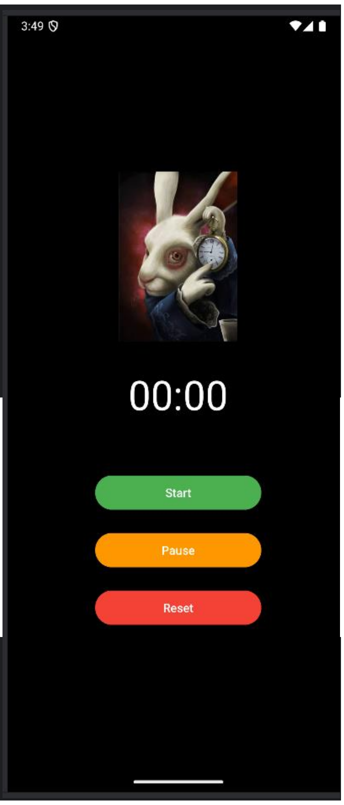


```kotlin
        btnReset.setOnClickListener {
            offset = 0
            setBaseTime()
            chronometr.stop()
            running = false
        }
    }


    private fun saveOffset() {
        offset = SystemClock.elapsedRealtime() - chronometr.base
    }


    private fun setBaseTime() {
        chronometr.base = SystemClock.elapsedRealtime() - offset
    }

    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)

        outState.putLong(OFFSET_KEY, offset)
        outState.putBoolean(RUNNING_KEY, running)
        outState.putLong(BASE_KEY, chronometr.base)
    }
}
```

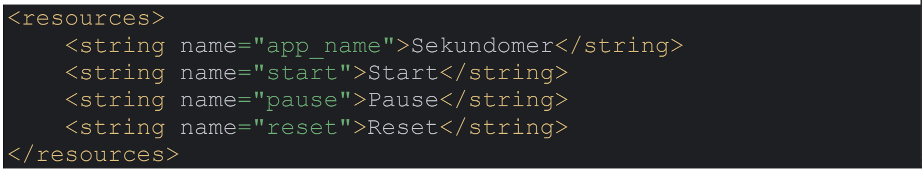
Colors.xml:



```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="green">#FF4CAF50</color>
    <color name="orange">#FFFF9800</color>
    <color name="red">#FFF44336</color>
</resources>
```



String:



```xml
<resources>
    <string name="app_name">Sekundomer</string>
    <string name="start">Start</string>
    <string name="pause">Pause</string>
    <string name="reset">Reset</string>
</resources>
```