STOPWATCH

# Activity.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="vertical" android:background="#0F9D58" android:padding="16dp"  tools:context=".MainActivity">    <TextView  android:id="@+id/time\_view" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal"  android:textAppearance="@android:style/TextAppearance.Large" android:textSize="56sp" />    <Button  android:layout\_width="381dp" android:layout\_height="wrap\_content" android:onClick="onClickStart" android:text="@string/start" android:padding="16dp"/>    <Button  android:id="@+id/stop\_button" android:layout\_width="407dp" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:layout\_marginTop="8dp" android:onClick="onClickStop" android:text="@string/stop" android:padding="56dp"/>  <Button  android:id="@+id/reset\_button" android:layout\_width="412dp" android:layout\_height="wrap\_content" android:layout\_gravity="center\_horizontal" android:layout\_marginTop="8dp" android:onClick="onClickReset" android:text="@string/reset" android:padding="19dp"/>  </LinearLayout> |

# Java

|  |
| --- |
| package com.example.stopwatch;  import android.app.Activity; import android.os.Handler; import android.view.View; import android.os.Bundle; import java.util.Locale; import android.widget.TextView;    public class MainActivity extends Activity { private int seconds = 0;    private boolean running;    private boolean wasRunning;    @Override  protected void onCreate(Bundle savedInstanceState)  {  super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*); if (savedInstanceState != null) {    seconds  = savedInstanceState  .getInt("seconds"); running  = savedInstanceState .getBoolean("running"); wasRunning  = savedInstanceState  .getBoolean("wasRunning");  }  runTimer();  }    @Override  public void onSaveInstanceState( Bundle savedInstanceState)  {  savedInstanceState  .putInt("seconds", seconds); savedInstanceState  .putBoolean("running", running); savedInstanceState  .putBoolean("wasRunning", wasRunning); }    @Override  protected void onPause()  {  super.onPause(); wasRunning = running; running = false;  }    @Override |

|  |
| --- |
| protected void onResume()  {  super.onResume(); if (wasRunning) { running = true;  }  }    public void onClickStart(View view)  {  running = true;  }  public void onClickStop(View view)  {  running = false;  }    public void onClickReset(View view)  {  running = false; seconds = 0;  }  private void runTimer()  {    final TextView timeView  = (TextView)findViewById(  R.id.*time\_view*);  final Handler handler  = new Handler();    handler.post(new Runnable() {  @Override  public void run()  {  int hours = seconds / 3600; int minutes = (seconds % 3600) / 60; int secs = seconds % 60;  String time  = String  .*format*(Locale.*getDefault*(),  "%d:%02d:%02d", hours, minutes, secs); timeView.setText(time);    if (running) { seconds++;  }    handler.postDelayed(this, 1000); |

}

});

} }

# String.xml

<resources>

<string name="app\_name">GFG|Stopwatch</string>

<string name="start">Start</string>

<string name="stop">Stop</string>

<string name="reset">Reset</string>

</resources>

# OUTPUT

