**Creating a MediaPlayer App**

**Step 1:**

**Open a new android project**  
After opening the Android Studio you have to create a **new project** using the **Empty Activity** with language as **Java** and give your project a unique name as you wish but don’t forget to keep the first alphabet capital.

2: Select**Empty Activity** as shown in the following screenshot.

3. Give your project a name, choose java and use lower level API so that your app can run on older version of android phones(I am using Api 16: Android 4.1 Jelly Bean).

Step 2:

**Designing the User Interface of the app**  
In this app, we have used 4 components:

* **a imageView– to show our given image for the song**
* **3 Buttons:**
  + **a play button to play our song**
  + **a pause button to pause our song**
  + **a stop button to stop our song**

These components are implemented on the below two layouts:

* **Vertical LinearLayout**
* **Horizontal LinearLayout**

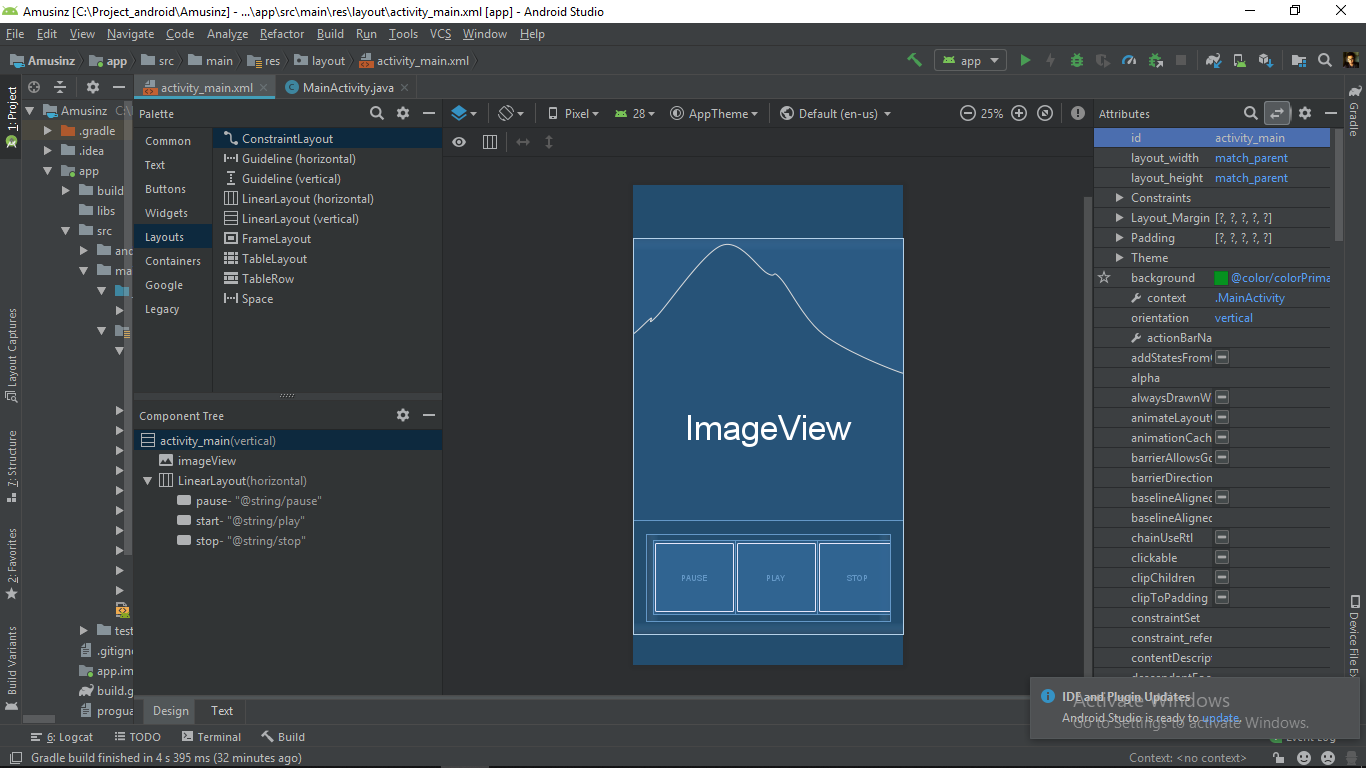
Inside the LinearLayout (vertical) there are two components:

* imageView component
* LinearLayout(horizontal)

Inside the LinearLayout (vertical) there are two components:

* imageView component
* LinearLayout(horizontal)

This layout will vertically divide our app screen in two halves. The **imageView** component will be on upper half and the **Horizontal Linear Layout** will be on the lower half. The horizontal layout will contain three buttons (play, pause and stop button). This horizontal layout will align these three buttons one after another horizontally on the lower half of our app screen.



In our app I have used different styles for play, pause and stop button by adding the following line of code:

*android:background=”@android:drawable/ic\_media\_play” for play button  
android:background=”@android:drawable/ic\_media\_pause” for pause button  
android:background=”@android:drawable/ic\_media\_stop” for stop button*

download 3 images for paly, pause and stop

save them as ic\_media\_play, ic\_media\_pause and ic\_media\_stop

paste them in ***project->app->src->main->res->drawable folder***

change the activity\_main.xml file as follows

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="<http://schemas.android.com/apk/res/android>"

    xmlns:tools="<http://schemas.android.com/tools>"

    android:id="@+id/activity\_main"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:background="@color/colorPrimary"

    android:orientation="vertical"

    android:theme="@style/Theme.AppCompat"

    tools:context=".MainActivity">

    <ImageView

        android:id="@+id/imageView"

        android:layout\_width="match\_parent"

        android:layout\_height="430dp"

        android:background="@drawable/download"

        android:contentDescription="@string/todo" />

    <LinearLayout

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent"

        android:layout\_margin="20dp"

        android:background="@color/colorAccent"

        android:orientation="horizontal"

        android:padding="10dp">

        <Button

            android:id="@+id/pause"

            style="@style/Widget.AppCompat.Button.Borderless.Colored"

            android:layout\_width="125dp"

            android:layout\_height="match\_parent"

            android:background="@android:drawable/ic\_media\_pause"

            android:onClick="musicpause" />

        <Button

            android:id="@+id/start"

            style="@style/Widget.AppCompat.Button.Borderless"

            android:layout\_width="125dp"

            android:layout\_height="match\_parent"

            android:background="@android:drawable/ic\_media\_play"

            android:onClick="musicplay" />

        <Button

            android:id="@+id/stop"

            style="@style/Widget.AppCompat.Button.Borderless"

            android:layout\_width="125dp"

            android:layout\_height="match\_parent"

            android:background="@android:drawable/ic\_delete"

            android:onClick="musicstop" />

    </LinearLayout>

</LinearLayout>

**Step 3 : Adding the music file to our app**

**Download a mp3 file**

Add the mp3 file to the raw folder. You can reach there by:

*app-> res-> raw*

If there is no raw folder, then create it by right-clicking on res directory then:

*res-> new-> directory*

Name the newly created directory as raw and add all the audio files in this folder. Drag and drop files here is not allowed. You have to copy your source file, then right-click on the raw directory and click paste.

File name - new name contains all small alphabets. The only valid characters are (a-z and 0-9 and \_ )

**Step 4: Let’s code the functionality of our App**

1. Make a object of **MediaPlayer** class named **music**. It is an inbuilt class in **android package**. All the properties of the MediaPlayer class can be used by this music object:

*MediaPlayer music*

1. We will add our music file to this newly created object by using **create** function :

*music = MediaPlayer.create(this, R.raw.filename);*

1. MediaPlayer class has an inbuilt function called **start** we will use this function for play button. It will start the song.

*public void playSong(View v){  
music.start();  
}*

1. For pause button we will use the inbuilt function **pause**. This will pause the song.

*public void pauseSong(View v) {  
mp.pause();  
}*

1. For **stop** button we will use the inbuilt **stop** function. This function also deletes the object (music), so we create a new object with the same name.

*public void stopSong(View v) {  
mp.stop();*

}

Java code for MainActivity.java

ublic class MainActivity

    extends AppCompatActivity {

    // Instantiating the MediaPlayer class

    MediaPlayer music;

    @Override

    protected void onCreate(

        Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        // Adding the music file to our

        // newly created object music

        music = MediaPlayer.create(

            this, R.raw.sound);

    }

    // Playing the music

    public void musicplay(View v)

    {

        music.start();

    }

    // Pausing the music

    public void musicpause(View v)

    {

        music.pause();

    }

    // Stopping the music

    public void musicstop(View v)

    {

        music.stop();

        music

            = MediaPlayer.create(

                this, R.raw.sound);

    }

}