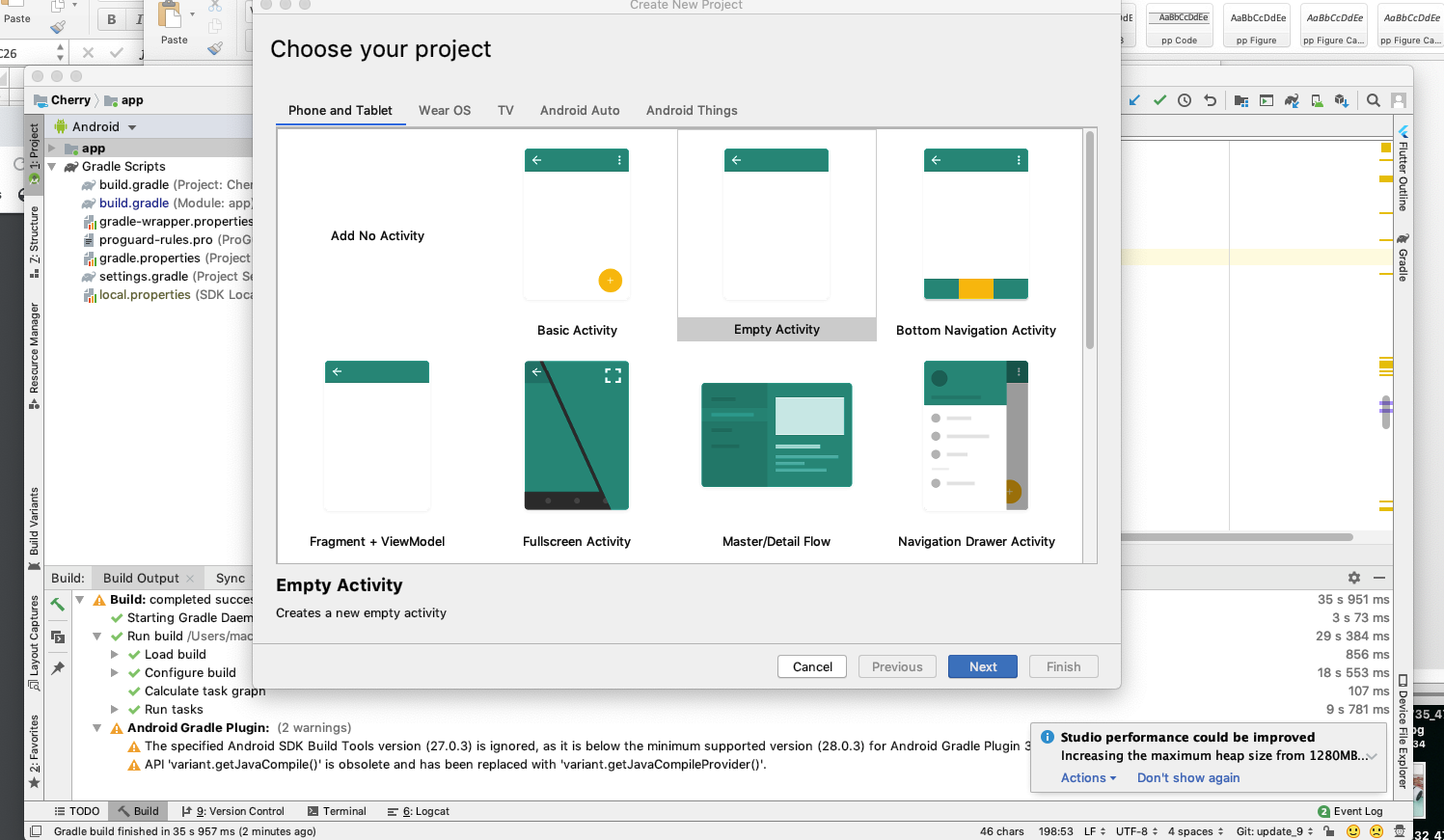
* Labguide này sẽ hướng dẫn các em cách viết một chương trình nghe nhạc đơn giản bằng cách sử dụng MediaPlayer và Bound Service.

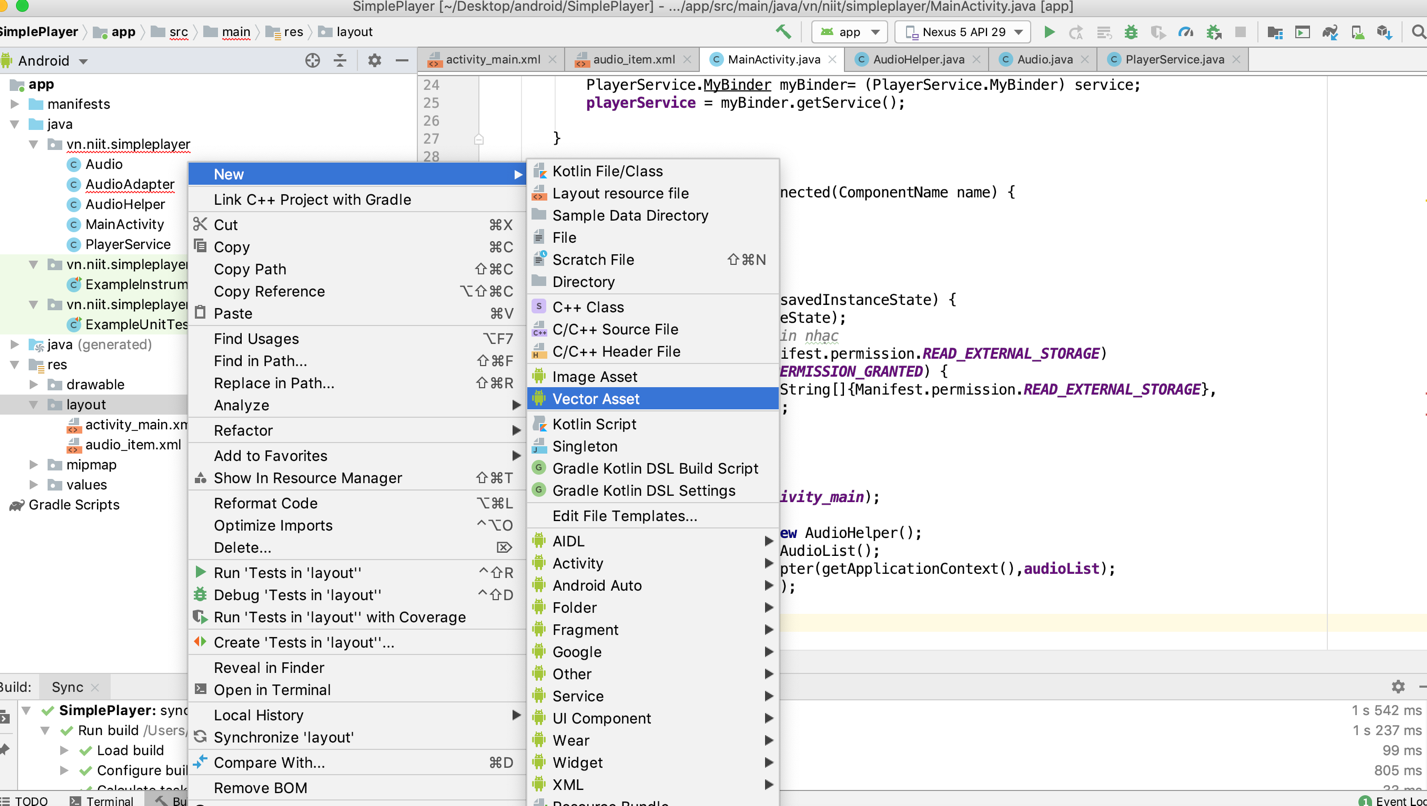
Bước 1: Tạo một dự án đặt tên là SimplePlayer, lưu ý chọn sẵn một Activity và đặt tên là MainActivity khi tạo dự án

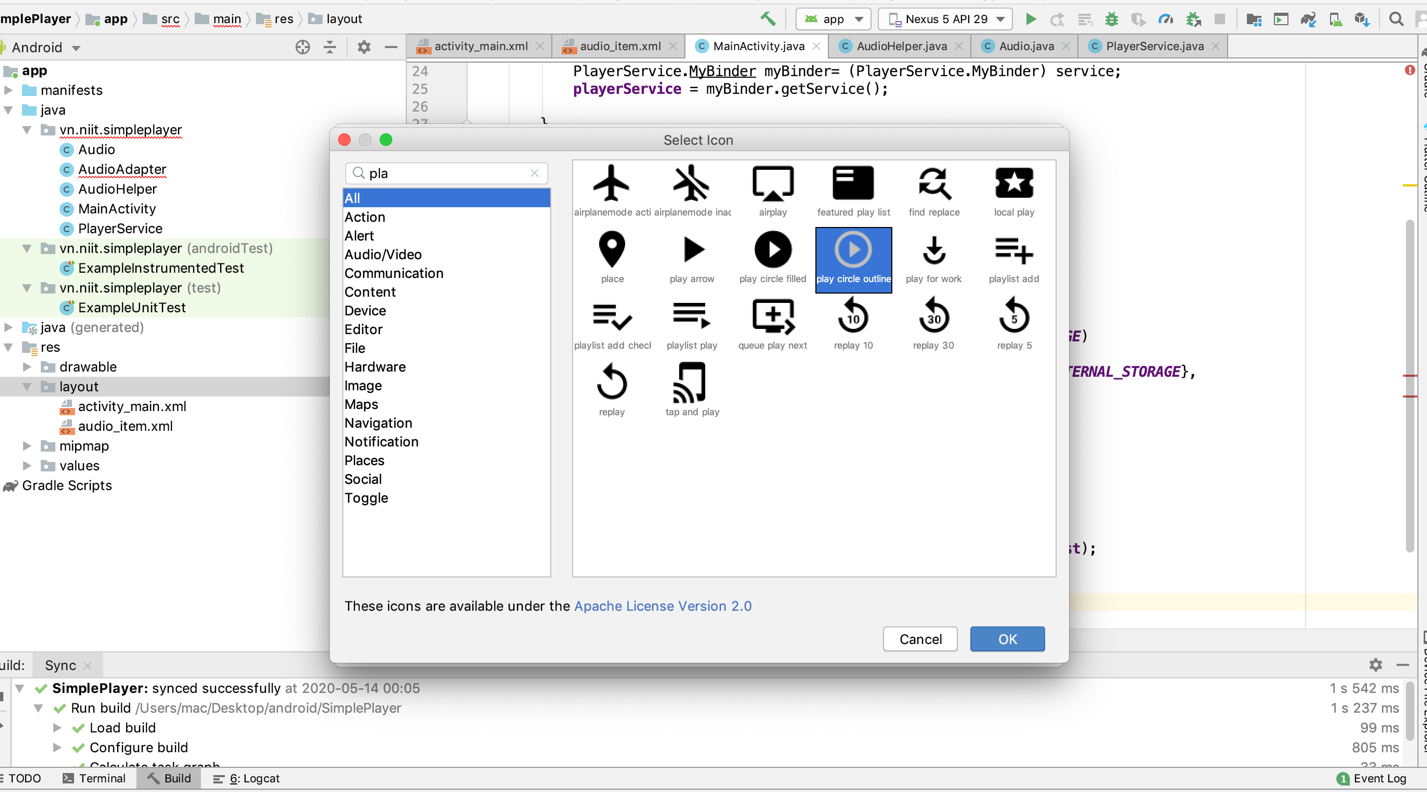


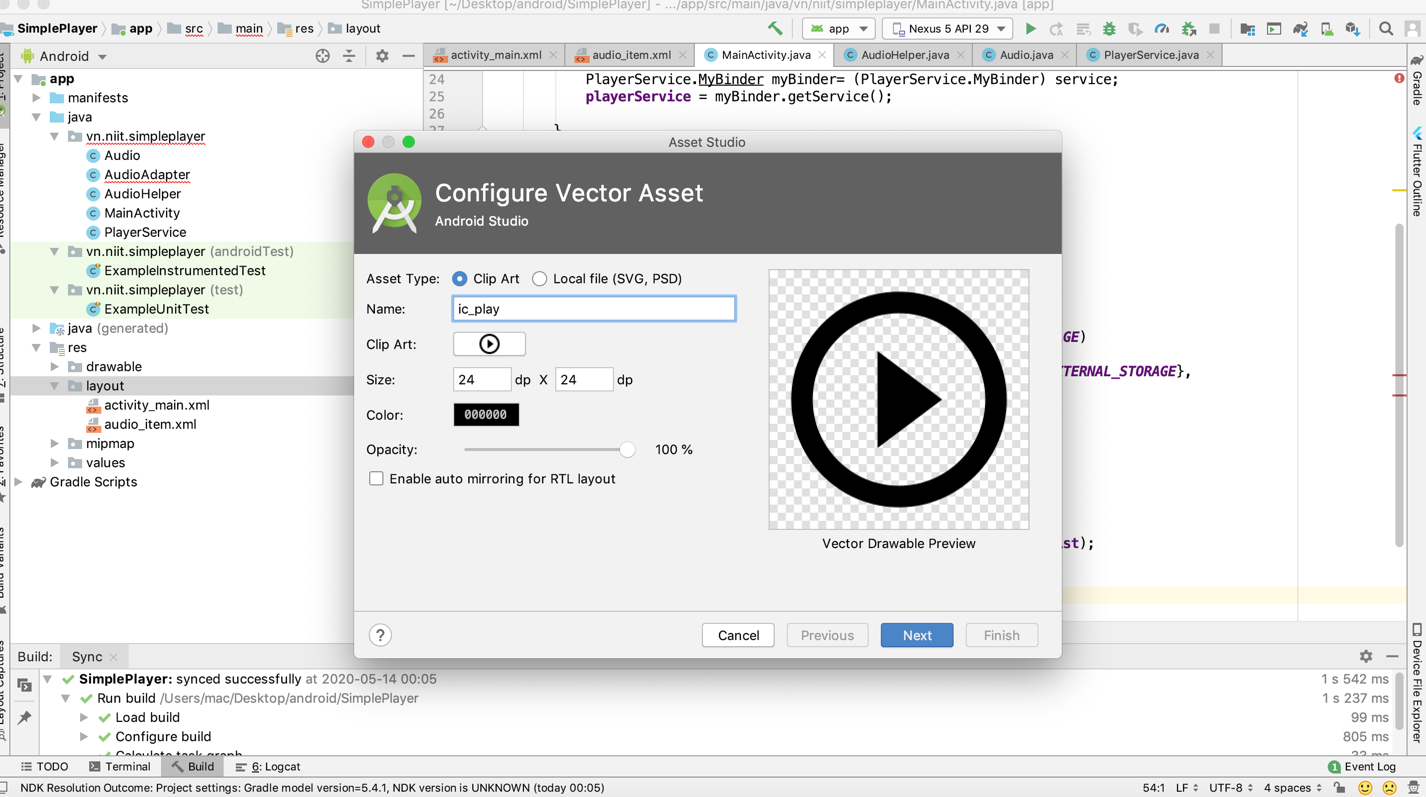
Buớc 2: Trong acitivity\_main.xml, viết mã sau

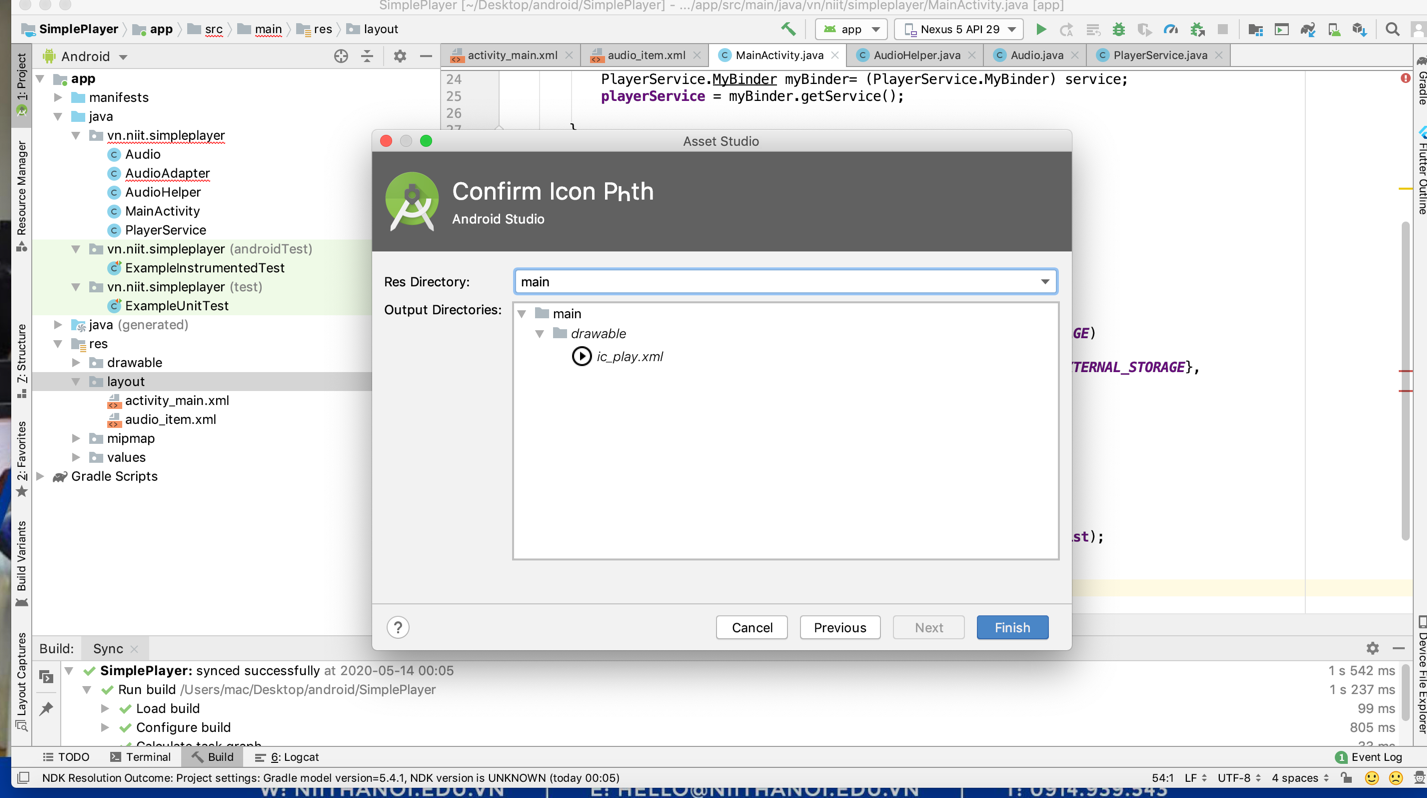
*<?***xml version="1.0" encoding="utf-8"***?>*<**ListView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/lst"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**/>

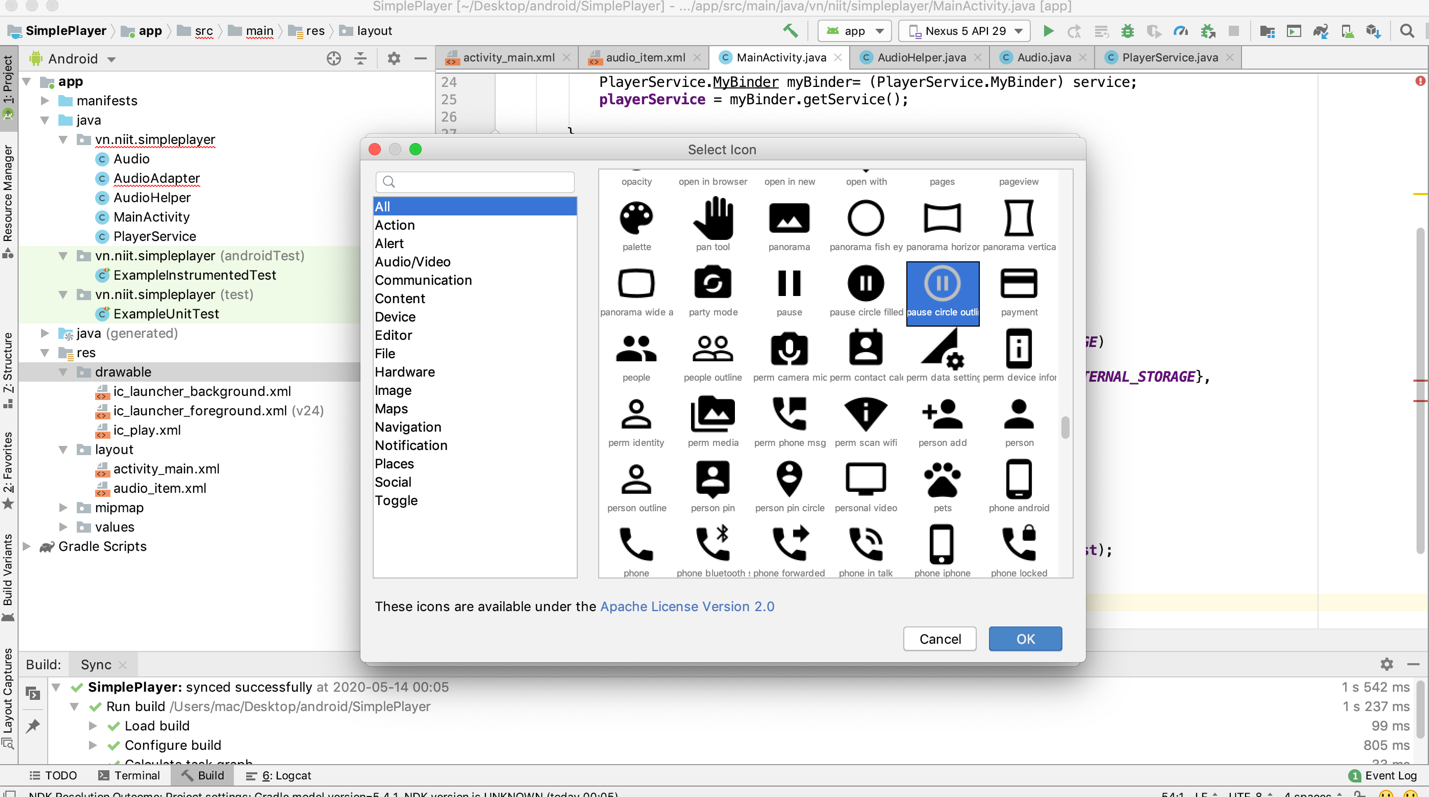
Bước 3: Thêm vào 2 icon play và pause từ kho vector của Android

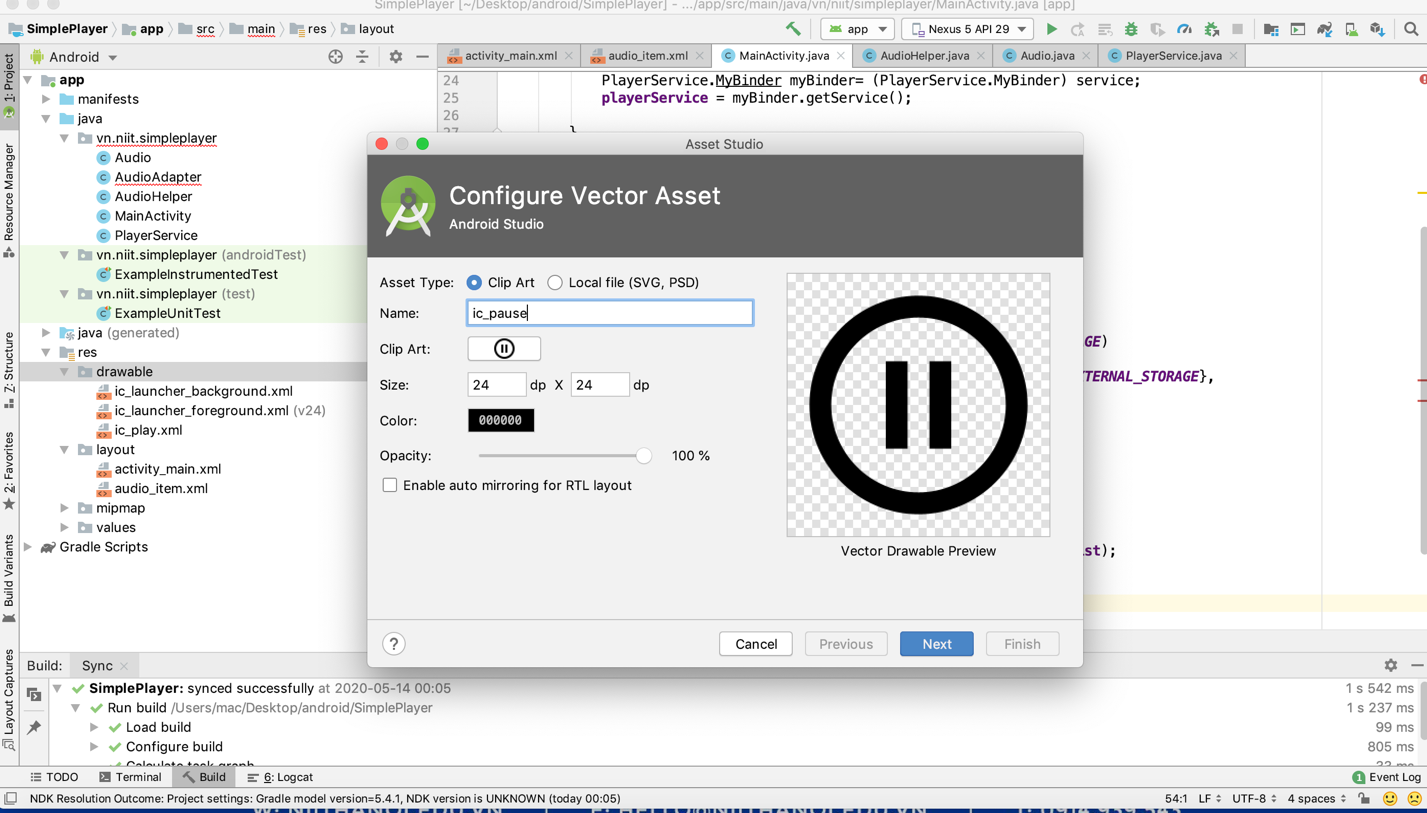


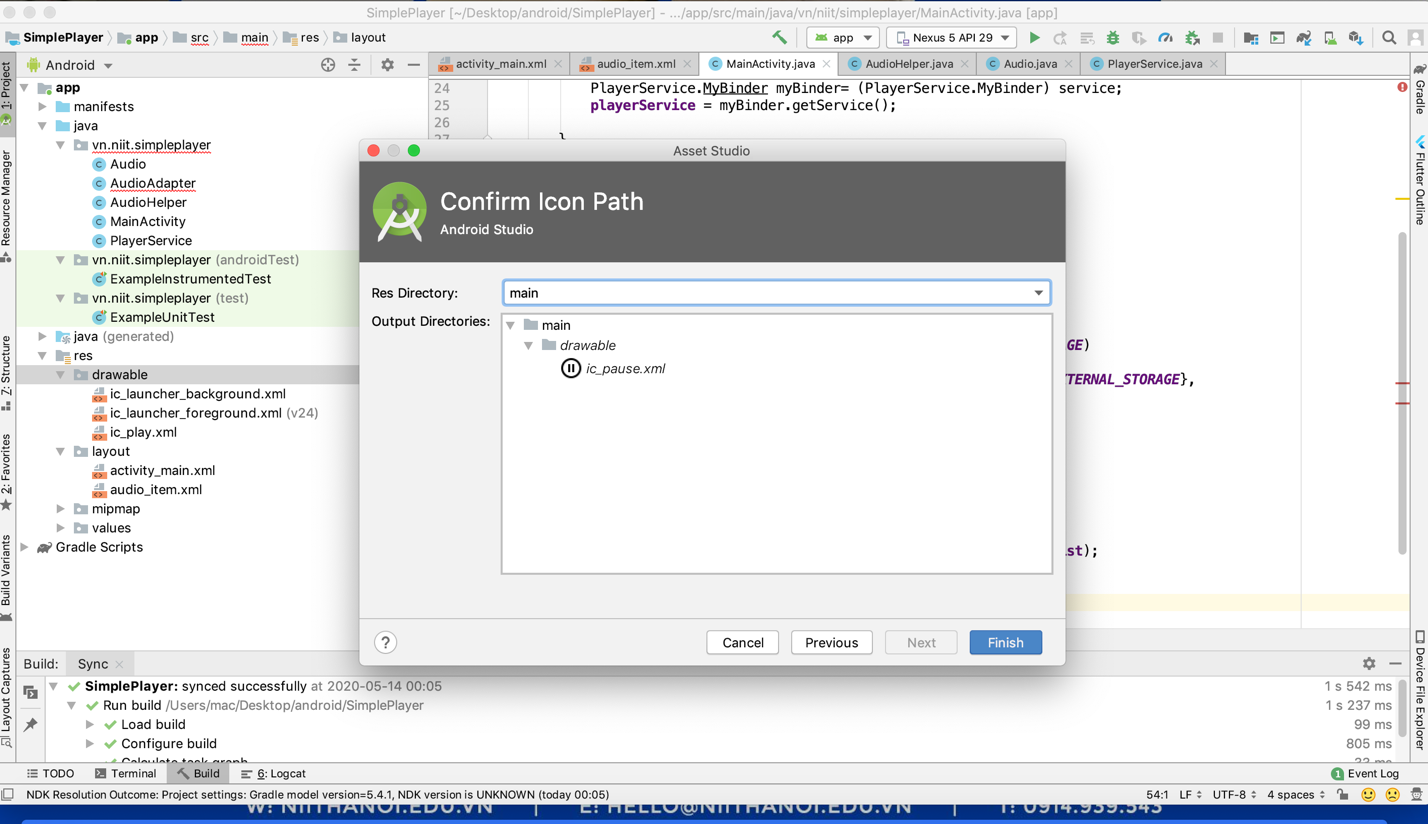








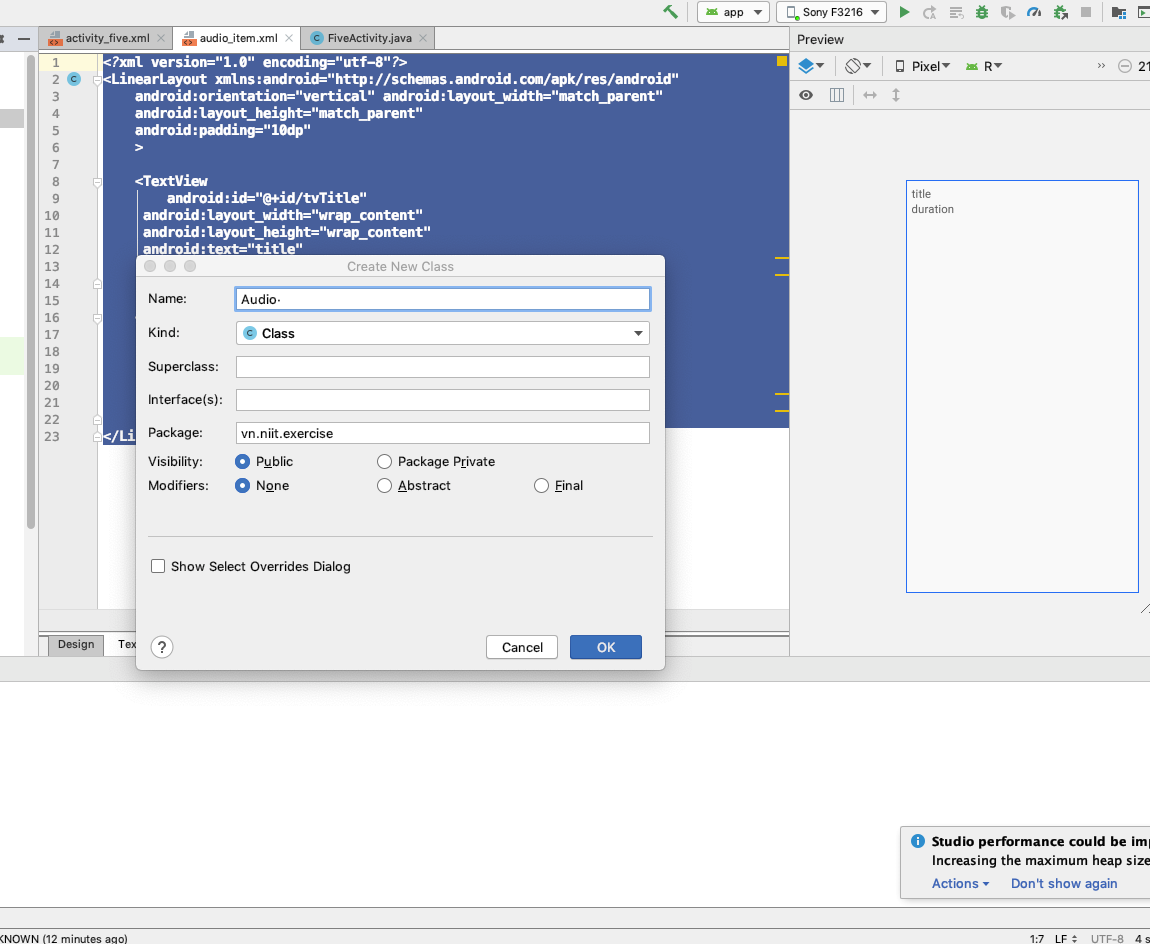




Buớc 4: Tạo file audio\_item.xml, viết mã cho file audio\_item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:orientation="horizontal"  
 android:padding="10dp"**>  
  
  
 <**ImageButton  
 android:id="@+id/btnPlay"  
 android:layout\_width="80dp"  
 android:layout\_height="80dp"  
 android:src="@drawable/ic\_play"  
 android:background="@null"  
 android:layout\_gravity="center"** />  
  
  
 <**TextView  
 android:id="@+id/tvTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="title"  
 android:textColor="@color/colorPrimaryDark"  
 android:textSize="20dp"  
 android:layout\_gravity="center"** />  
</**LinearLayout**>

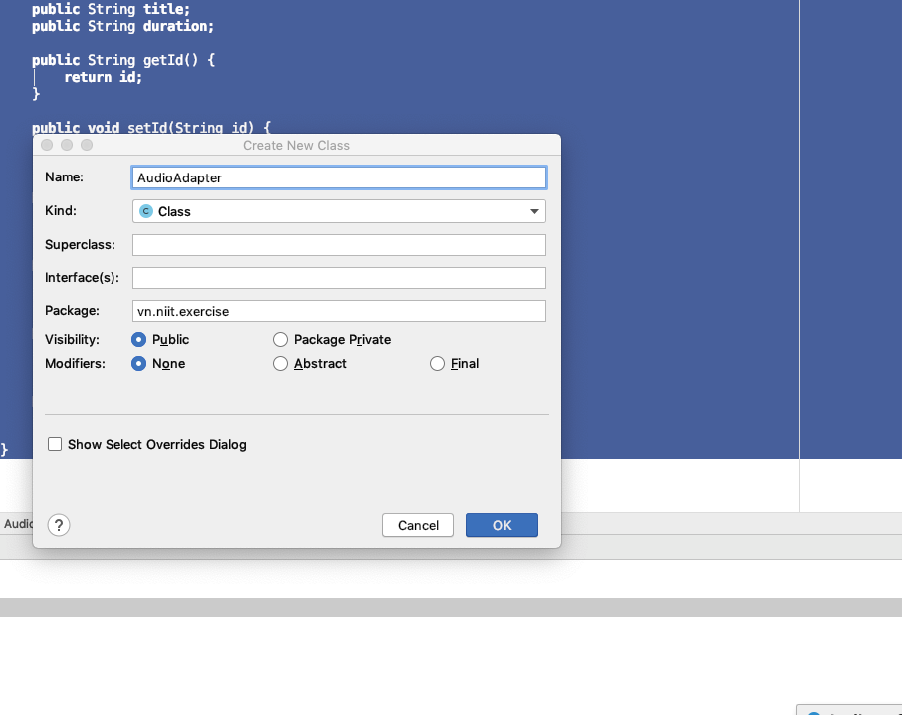
buớc 5: Tạo model Audio.java



Viết mã cho Audio.java file

**public class** Audio **implements** Serializable {  
 **public** String **title**;  
 **public** String **path**;  
 **public** Boolean **isPlaying** = **false**;  
  
 **public** String getTitle() {  
 **return title**;  
 }  
  
 **public void** setTitle(String title) {  
 **this**.**title** = title;  
 }  
  
 **public** String getPath() {  
 **return path**;  
 }  
  
 **public void** setPath(String path) {  
 **this**.**path** = path;  
 }  
  
 **public** Boolean getPlaying() {  
 **return isPlaying**;  
 }  
  
 **public void** setPlaying(Boolean playing) {  
 **isPlaying** = playing;  
 }  
}

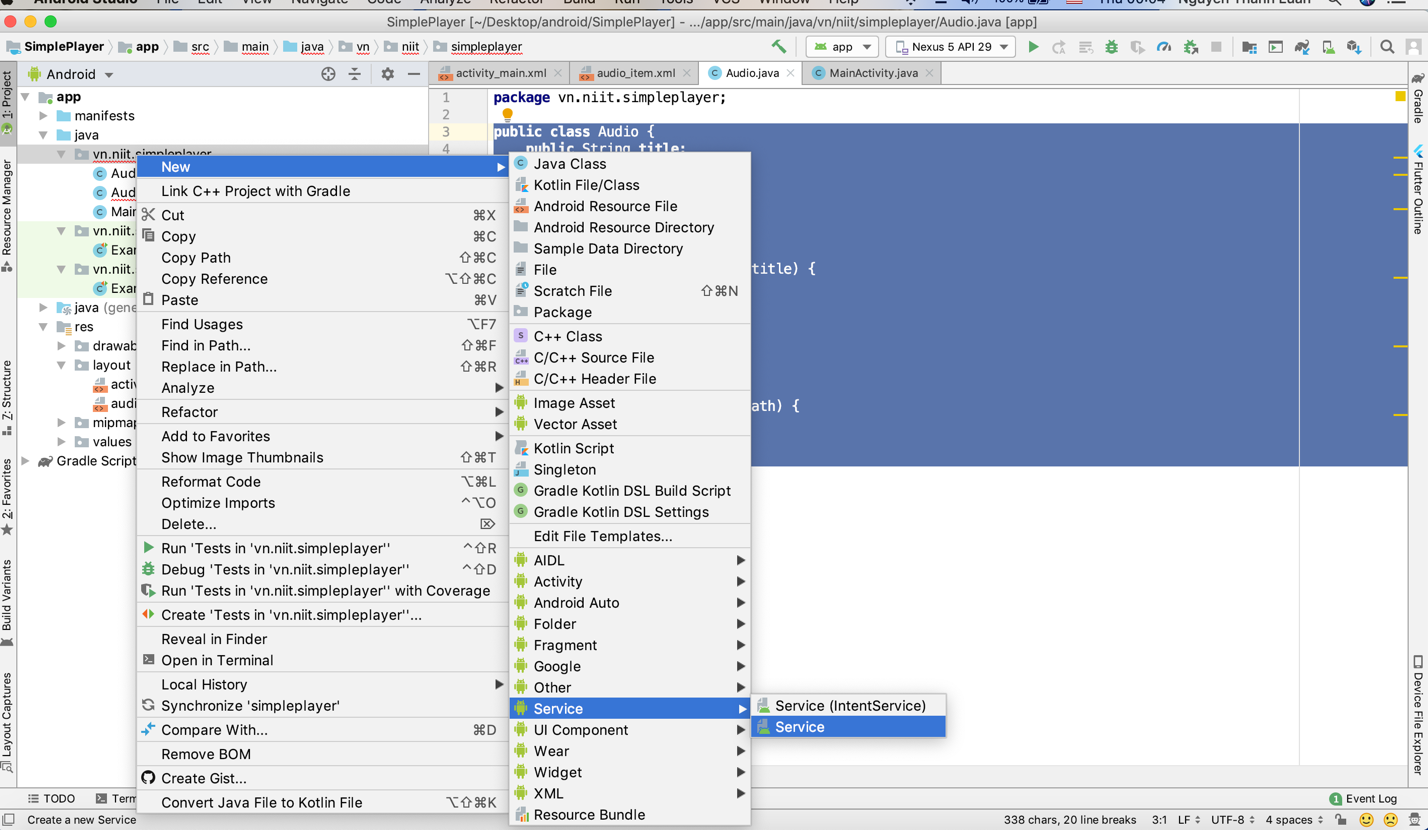
Buớc 6: Tạo AudioAdapter

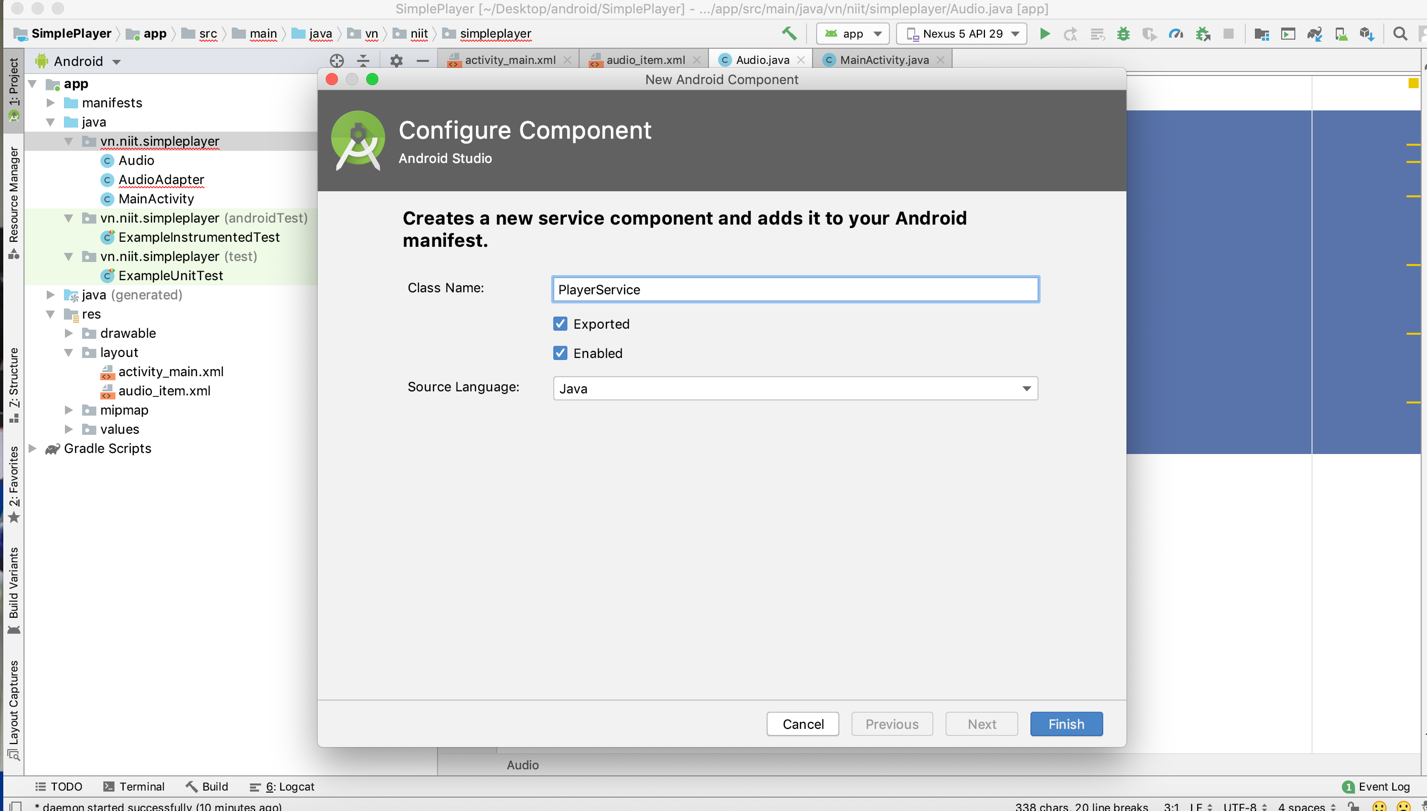


Viết mã cho AudioAdapter.java

**public class** AudioAdapter **extends** ArrayAdapter<Audio> {  
 ArrayList<Audio> **audios**;  
 **public** AudioAdapter(@NonNull Context context, @NonNull ArrayList<Audio> objects) {  
 **super**(context, R.layout.***audio\_item***, objects);  
 **this**.**audios** = objects;  
 }  
  
 @NonNull  
 @Override  
 **public** View getView(**int** position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 View view = convertView;  
 **if**(view==**null**){  
 view= LayoutInflater.*from*(getContext()).inflate(R.layout.***audio\_item***,**null**);  
 }  
 TextView tvTitle = view.findViewById(R.id.***tvTitle***);  
  
 tvTitle.setText(**this**.**audios**.get(position).getTitle());  
  
 ImageButton btnPlay = view.findViewById(R.id.***btnPlay***);  
 **if**(**this**.**audios**.get(position).getPlaying()){  
 btnPlay.setImageDrawable(getContext().getDrawable(R.drawable.***ic\_pause***));  
 }**else**{  
 btnPlay.setImageDrawable(getContext().getDrawable(R.drawable.***ic\_play***));  
 }  
  
 btnPlay.setTag(position);  
  
 btnPlay.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Integer position = (Integer) v.getTag();  
 **if**(**audios**.get(position).getPlaying()){  
 Intent intent = **new** Intent(**"PAUSE\_PLAYING"**);  
 intent.putExtra(**"INDEX\_AUDIO"**,position);  
 getContext().sendBroadcast(intent);  
 }**else**{  
 Intent intent = **new** Intent(**"START\_PLAYING"**);  
 intent.putExtra(**"INDEX\_AUDIO"**,position);  
 getContext().sendBroadcast(intent);  
 }  
 }  
 });  
 **return** view;  
 }  
}

Bước 7: Tạo một Service làm nhiệm vụ chơi nhạc trong background, trong Service này sẽ chứa lớp MediaPlayer.

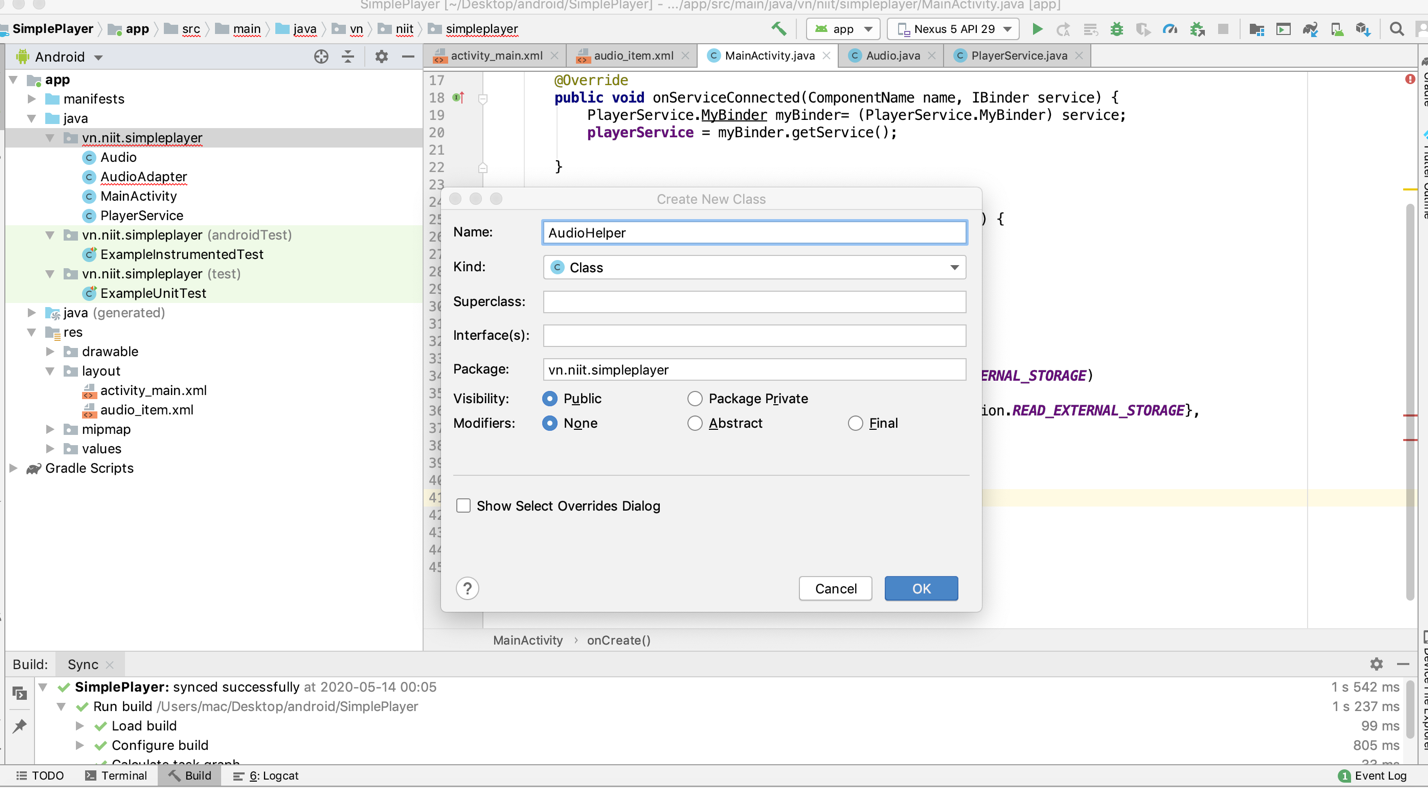




**Viết mã cho PlayerService**

**public class** PlayerService **extends** Service {  
 MediaPlayer **mediaPlayer** = **null**;  
 **private final** IBinder **myBinder** = **new** MyBinder();  
 NotificationManager **notificationManager**;  
 NotificationCompat.Builder **notificationBuilder**;  
  
  
 **public class** MyBinder **extends** Binder {  
 PlayerService getService() {  
 **return** PlayerService.**this**;  
 }  
 }  
  
 **public** PlayerService() {  
 **mediaPlayer** = **new** MediaPlayer();  
 }  
  
 @Override  
 **public** IBinder onBind(Intent intent) {  
 *//* ***TODO: Return the communication channel to the service.* return myBinder**;  
 }  
  
 **public void** showNotification(String content) {  
 NotificationManager notificationManager = (NotificationManager) getSystemService(***NOTIFICATION\_SERVICE***);  
 **if** (android.os.Build.VERSION.***SDK\_INT*** >= android.os.Build.VERSION\_CODES.***O***) {  
 String CHANNEL\_ID = **"controls\_channel\_id"**;  
 String CHANNEL\_NAME = **"controls\_channel"**;  
 NotificationChannel channel = **new** NotificationChannel(CHANNEL\_ID,  
 CHANNEL\_NAME, NotificationManager.***IMPORTANCE\_HIGH***);  
 channel.setSound(**null**,**null**);  
 channel.enableVibration(**false**);  
 notificationManager.createNotificationChannel(channel);  
 **notificationBuilder** = **new** NotificationCompat.Builder(getApplicationContext())  
 .setChannelId(CHANNEL\_ID)  
 .setSmallIcon(R.drawable.***ic\_launcher\_background***)  
 .setContentText(**"placeholder"**)  
 .setContentTitle(**"placeholder"**);  
  
 }**else**{  
 **notificationBuilder** = **new** NotificationCompat.Builder(  
 getApplicationContext());  
 }  
 Intent intent = **new** Intent(**this**, MainActivity.**class**);  
 PendingIntent contentIntent = PendingIntent.*getActivity*(**this**, 0,  
 intent, 0);  
  
 **notificationBuilder**.setContentText(content).setContentIntent(contentIntent).setSmallIcon(R.drawable.***ic\_launcher\_background***);  
 Notification notification =**notificationBuilder**.build();  
 notificationManager.notify(**"SimplePlayer"**,0,notification);  
  
  
 }  
  
 **public void** play(Audio audio) {  
 **if** (**mediaPlayer** != **null**) {  
 **mediaPlayer**.reset();  
 }  
  
 **try** {  
 **mediaPlayer**.setDataSource(audio.getPath());  
 **mediaPlayer**.prepare();  
 **mediaPlayer**.start();  
 showNotification(audio.getTitle());  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 **public void** pause() {  
 **if**(**mediaPlayer**!=**null** && **mediaPlayer**.isPlaying()){  
 **mediaPlayer**.pause();  
 }  
 }  
}

Buớc 8: Tạo một lớp AudioHelper, lớp này làm nhiệm vụ scan tất cả các file nhạc trong thẻ nhớ, trong bài thực hành này chúng ta không sử dụng MediStore để lấy thông tin nhạc.



**public class** AudioHelper {  
 String **MEDIA\_PATH** ;  
 **private** ArrayList<Audio> **audioList** = **new** ArrayList<Audio>();  
  
 **public** AudioHelper() {  
 **MEDIA\_PATH** = Environment.*getExternalStorageDirectory*().getPath();  
 }  
  
 */\*\*  
 \* Hàm để đọc toàn bộ file mp3 trong bộ nhớ và lưu vào arraylist  
 \* \*/* **public** ArrayList<Audio> getAudioList(){  
 File home = **new** File(**MEDIA\_PATH**);  
  
 **if** (home.listFiles(**new** FileExtensionFilter()).**length** > 0) {  
 **for** (File file : home.listFiles(**new** FileExtensionFilter())) {  
 Audio audio = **new** Audio();  
 audio.setPath(file.getPath());  
 audio.setTitle(file.getName().substring(0, (file.getName().length() - 4)));  
 **audioList**.add(audio);  
 }  
 }  
 *// return songs list array* **return audioList**;  
 }  
  
 */\*\*  
 \* filter file có đuôi là mp3  
 \* \*/* **class** FileExtensionFilter **implements** FilenameFilter {  
 **public boolean** accept(File dir, String name) {  
 **return** (name.endsWith(**".mp3"**) || name.endsWith(**".MP3"**) || name.endsWith(**".wav"**) || name.endsWith(**".ogg"**));  
 }  
 }  
}

Buớc 9: Viết mã cho MainActivity

**public class** MainActivity **extends** AppCompatActivity {  
 PlayerService **playerService** = **null**;  
 ListView **lst**;  
 AudioAdapter **audioAdapter**;  
 ArrayList<Audio> **audioList** = **new** ArrayList<>();  
  
 ServiceConnection **mConnection** = **new** ServiceConnection() {  
 @Override  
 **public void** onServiceConnected(ComponentName name, IBinder service) {  
 PlayerService.MyBinder myBinder= (PlayerService.MyBinder) service;  
 **playerService** = myBinder.getService();  
  
 }  
  
 @Override  
 **public void** onServiceDisconnected(ComponentName name) {  
 **playerService** = **null**;  
 }  
 };  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 *//yêu cầu quyền đọc thông tin nhạc* **if** (checkSelfPermission(Manifest.permission.***READ\_EXTERNAL\_STORAGE***)  
 != PackageManager.***PERMISSION\_GRANTED***) {  
 requestPermissions(**new** String[]{Manifest.permission.***READ\_EXTERNAL\_STORAGE***},  
 10);  
 **return**;  
 }  
 bindService(**new** Intent(**this**,PlayerService.**class**),**mConnection**,***BIND\_AUTO\_CREATE***);  
 setContentView(R.layout.***activity\_main***);  
 **lst**=findViewById(R.id.***lst***);  
 AudioHelper audioHelper = **new** AudioHelper();  
 **audioList** = audioHelper.getAudioList();  
 **audioAdapter** = **new** AudioAdapter(getApplicationContext(),**audioList**);  
 **lst**.setAdapter(**audioAdapter**);  
 IntentFilter intentFilter = **new** IntentFilter();  
 intentFilter.addAction(**"PAUSE\_PLAYING"**);  
 intentFilter.addAction(**"START\_PLAYING"**);  
 registerReceiver(**new** MyReceiver(),intentFilter);  
  
 }  
  
 **public class** MyReceiver **extends** BroadcastReceiver{  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 **switch** (action){  
 **case "PAUSE\_PLAYING"**:  
 Integer index= intent.getIntExtra(**"INDEX\_AUDIO"**,0);  
 **audioList**.get(index).setPlaying(**false**);  
 **playerService**.pause();  
 **break**;  
  
 **case "START\_PLAYING"**:  
 index= intent.getIntExtra(**"INDEX\_AUDIO"**,0);  
  
 **for** (**int** i = 0; i < **audioList**.size(); i++) {  
 **audioList**.get(i).setPlaying(**false**);  
 }  
 **audioList**.get(index).setPlaying(**true**);  
 **playerService**.play(**audioList**.get(index));  
 **break**;  
 }  
 **audioAdapter**.notifyDataSetChanged();  
 }  
 }  
}

Cuối cùng các em nhớ cấp quyền đọc bộ nhớ ở file Manifest

<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"**/>

Chạy thử chuơng trình

