# 实验4 广播与服务

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 202231060901 | 姓名 | 刘洋 |
| 专业 | 物联网工程 | 年级 | 2022级 |

特别说明：实验报告必须命名为【学号紧跟姓名\_实验项目名】的格式，不包含括号【】。

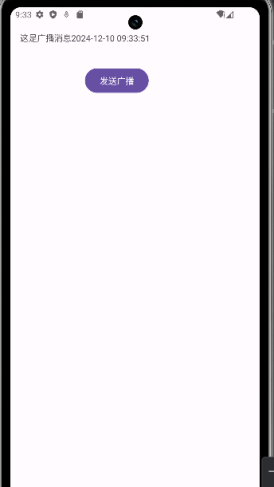
例如【202231060901刘洋\_Android实验4.docx】，不按此命名不得分。

**规范命名后，压缩为rar或者zip提交。**

实验报告提交地址：<https://pintia.cn/>

## 广播(30分)

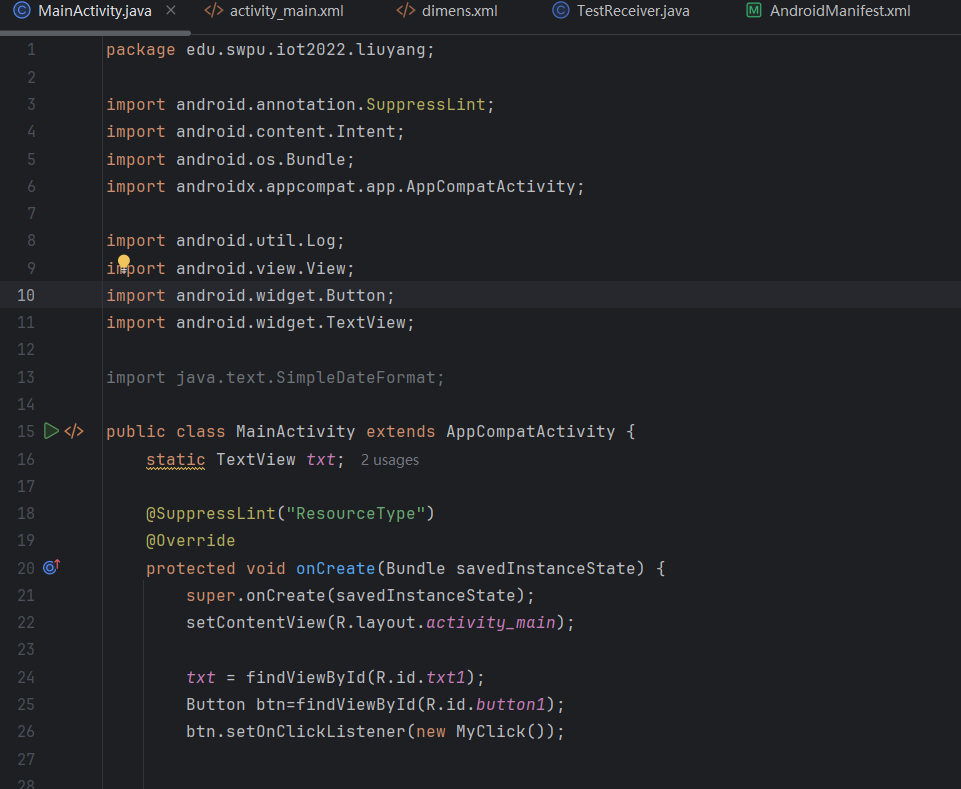
1. 运行效果图



1. MainActivity的Java代码(**是要文本，不是截图**)：

|  |
| --- |
| package edu.swpu.iot2022.liuyang;  import android.annotation.SuppressLint;  import android.content.Intent;  import android.os.Bundle;  import androidx.appcompat.app.AppCompatActivity;  import android.util.Log;  import android.view.View;  import android.widget.Button;  import android.widget.TextView;  import java.text.SimpleDateFormat;  public class MainActivity extends AppCompatActivity {  static TextView txt;  @SuppressLint("ResourceType")  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  txt = findViewById(R.id.txt1);  Button btn=findViewById(R.id.button1);  btn.setOnClickListener(new MyClick());  }  class MyClick implements View.OnClickListener  {  @Override  public void onClick(View v) {  Intent objInent=new Intent();  objInent.setAction("SwpuLiu");  objInent.setClassName("edu.swpu.iot2022.liuyang", "edu.swpu.iot2022.liuyang.TestReceiver");  Bundle bundle=new Bundle();  bundle.putString("MyKey" ,"这是广播消息");  objInent.putExtras(bundle);  sendBroadcast(objInent);  Log.d("MainActivity","点击事件");  }  }  } |

1. 对MainActivity代码的前面部分进行截图，要出现package语句：



## 服务(40分)

你构建的项目的相关代码

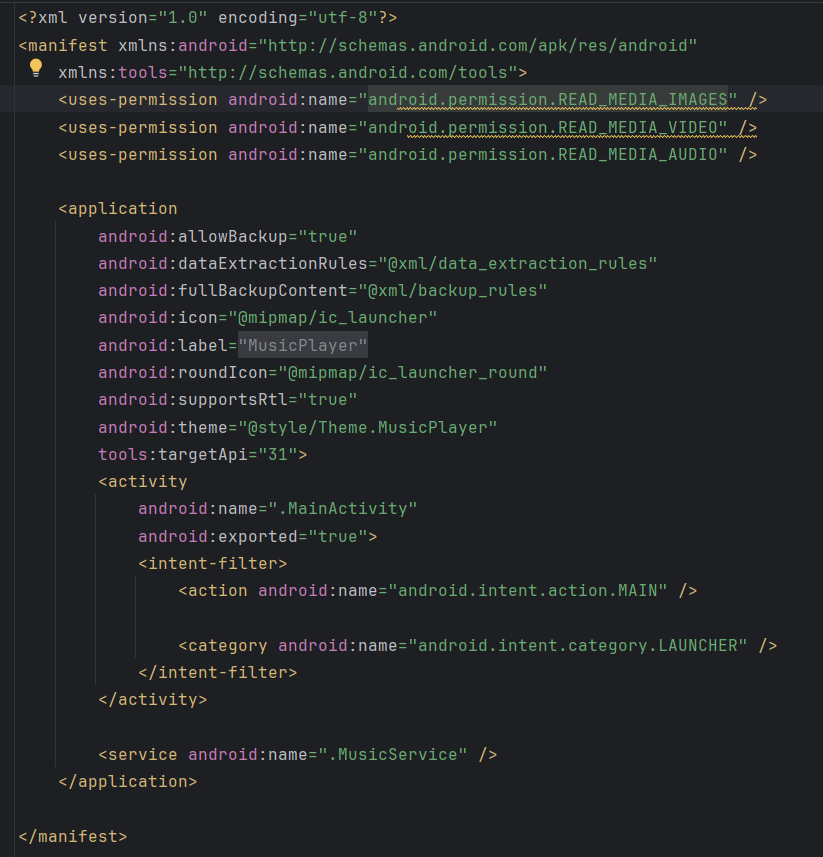
1. MainActivity的代码(是要文本，不是截图)

|  |
| --- |
| package edu.swpu.iot2022.musicplayer;  import android.Manifest;  import android.os.Build;  import android.content.ComponentName;  import android.content.Context;  import android.content.DialogInterface;  import android.content.Intent;  import android.content.ServiceConnection;  import android.content.pm.PackageManager;  import android.graphics.BitmapFactory;  import android.os.Bundle;  import android.os.Handler;  import android.os.IBinder;  import android.view.MenuItem;  import android.view.View;  import android.widget.AdapterView;  import android.widget.ImageView;  import android.widget.ListView;  import android.widget.SeekBar;  import android.widget.TextView;  import android.view.ContextMenu;  import android.view.Menu;  import android.widget.Toast;  import androidx.activity.EdgeToEdge;  import androidx.annotation.NonNull;  import androidx.appcompat.app.AlertDialog;  import androidx.appcompat.app.AppCompatActivity;  import androidx.core.app.ActivityCompat;  import androidx.core.content.ContextCompat;  import androidx.core.graphics.Insets;  import androidx.core.view.ViewCompat;  import androidx.core.view.WindowInsetsCompat;  import java.util.ArrayList;  import java.util.List;  public class MainActivity extends AppCompatActivity {  private static final int REQUEST\_CODE\_READ\_EXTERNAL\_STORAGE =1 ;  private static final int REQUEST\_CODE\_STORAGE\_PERMISSION = 2;  private List<MusicInfo> musicList = new ArrayList<MusicInfo>();  private ListAdapter mListAdapter;  private ListView mListView;  //记录长按的列表项坐标  private int currentSel;  //按钮  private ImageView btnPrevious;  private ImageView btnNext;  private ImageView btnPlay;  //文本  private TextView listTitle;  private TextView playingName;  //进度条  private SeekBar musicSeekBar;  //自定义Binder对象 用于调用服务中的方法  private MyBinderInterface myBinder;  //自定义服务连接对象  private MyServiceConnection conn;  //是否正在播放  private boolean isPlaying = false;  private Handler handler = new Handler();  //更新线程用于更新进度条  private Runnable updateThread = new Runnable() {  @Override  public void run() {  if (myBinder != null){  try {  if (myBinder.isPlaying()){  int duration = myBinder.getDuration();  int currentPos = myBinder.getCurrentPosition();  musicSeekBar.setMax(duration);  musicSeekBar.setProgress(currentPos);  int prg\_sec = currentPos/1000;  int max\_sec = duration/1000;  if (prg\_sec == max\_sec){  myBinder.PlayNext();  updateState();  }  }  }catch (Exception e){  e.printStackTrace();  }  }  handler.post(updateThread);  }  };  //定义服务连接  private class MyServiceConnection implements ServiceConnection {  @Override  public void onServiceConnected(ComponentName name, IBinder service) {  myBinder = (MyBinderInterface)service;  }  @Override  public void onServiceDisconnected(ComponentName name) {  }  };  //更新播放状态  private void updateState() {  int index = myBinder.getCurrentIndex();  mListAdapter.setFocusItemPos(index);  String currentMusicName = musicList.get(index).getMusic\_title();  playingName.setText(currentMusicName);  btnPlay.setImageBitmap(BitmapFactory.decodeResource(getResources(),R.drawable.play));  isPlaying = true;  }  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  // 检查是否已经授予权限  if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ\_MEDIA\_AUDIO) != PackageManager.PERMISSION\_GRANTED) {  ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ\_MEDIA\_AUDIO}, REQUEST\_CODE\_STORAGE\_PERMISSION);  }  //通过工具类MusicUtils获取音乐信息列表  musicList = MusicUtils.ResolveMusicToList(getApplicationContext());  //获取视图  initView();  //设置列表标题  String title = getResources().getString(R.string.title\_string).toString();  title += "（总数："+ musicList.size() + "）";  listTitle.setText(title);  //为mListView注册上下文菜单  registerForContextMenu(mListView);  conn = new MyServiceConnection();  //绑定服务  bindService(new Intent(this,MusicService.class),conn, Context.BIND\_AUTO\_CREATE);  handler.post(updateThread);  }  @Override  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  super.onRequestPermissionsResult(requestCode, permissions, grantResults);  if (requestCode == REQUEST\_CODE\_STORAGE\_PERMISSION) {  if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  } else {  Toast.makeText(this, "权限被拒绝", Toast.LENGTH\_SHORT).show();  }  }  }  //初始化视图  private void initView(){  listTitle = (TextView)findViewById(R.id.music\_list\_title);  playingName = (TextView)findViewById(R.id.music\_name);  musicSeekBar = (SeekBar)findViewById(R.id.music\_seek\_bar);  btnPrevious = (ImageView)findViewById(R.id.btn\_previous);  btnNext = (ImageView)findViewById(R.id.btn\_next);  btnPlay = (ImageView)findViewById(R.id.btn\_play);  mListView = (ListView)findViewById(R.id.music\_list);  mListAdapter = new ListAdapter(MainActivity.this,musicList);  mListView.setAdapter(mListAdapter);  setListener();  }  //设置监听事件  private void setListener(){  mListView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {  @Override  public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {  mListView.showContextMenu();  currentSel = position;  return true;  }  });  mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  myBinder.setCurrentIndex(position);  myBinder.Play();  mListAdapter.setFocusItemPos(position);  updateState();  }  });  btnPrevious.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (isPlaying == true){  myBinder.PlayPrev();  updateState();  }  }  });  btnNext.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (isPlaying == true){  myBinder.PlayNext();  updateState();  }  }  });  btnPlay.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (isPlaying == true){  btnPlay.setImageBitmap(BitmapFactory.decodeResource(  getResources(),R.drawable.play));  isPlaying = false;  myBinder.Pause();  return;  }  if (isPlaying == false){  if (myBinder.getCurrentIndex() == -1){  myBinder.setCurrentIndex(0);  mListAdapter.setFocusItemPos(0);  myBinder.Play();  updateState();  }  btnPlay.setImageBitmap(BitmapFactory.decodeResource(  getResources(),R.drawable.play));  isPlaying = true;  myBinder.Resume();  }  }  });  musicSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  @Override  public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  }  @Override  public void onStartTrackingTouch(SeekBar seekBar) {  }  @Override  public void onStopTrackingTouch(SeekBar seekBar) {  if (myBinder != null){  try {  myBinder.seekTo(seekBar.getProgress());  }catch (Exception e){  e.printStackTrace();  }  }  }  });  }  @Override  public void onCreateContextMenu(ContextMenu menu, View view, ContextMenu.ContextMenuInfo menuInfo){  menu.add(0,0,0,R.string.menu\_detail);  menu.add(0,1,1,R.string.menu\_play);  super.onCreateContextMenu(menu,view,menuInfo);  }  @Override  public boolean onContextItemSelected(MenuItem item) {  switch (item.getItemId()){  case 0:  StringBuilder msgBuilder = new StringBuilder();  msgBuilder.append("文件名：" + musicList.get(currentSel).getMusic\_name() + "\n");  msgBuilder.append("路 径：" + musicList.get(currentSel).getMusic\_path() + "\n");  msgBuilder.append("时 长：" + musicList.get(currentSel).getMusic\_duration()/1000 + " s\n");  String title = "文件详情";  new AlertDialog.Builder(MainActivity.this)  .setIcon(R.drawable.note)  .setTitle(title)  .setMessage(msgBuilder.toString())  .setPositiveButton(R.string.btn\_confirm, new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialog, int which) {  }}).create().show();  break;  case 1:  //不处于播放状态 或者 选择的歌曲和正在播放的歌曲不是同一首 则更新状态且播放  if (isPlaying == false || currentSel != myBinder.getCurrentIndex()){  myBinder.setCurrentIndex(currentSel);  updateState();  myBinder.Play();  }  //提示选择的歌曲已经在播放了  else{  Toast.makeText(MainActivity.this,R.string.str\_playing,Toast.LENGTH\_SHORT).show();  }  break;  default:  break;  }  return super.onContextItemSelected(item);  }  @Override  public boolean onCreateOptionsMenu(Menu menu) {  boolean retValue = super.onCreateOptionsMenu(menu);  getMenuInflater().inflate(R.menu.menu\_item,menu);  return retValue;  }  @Override  public boolean onOptionsItemSelected(MenuItem item) {  if (item.getItemId() == R.id.item\_about){  StringBuilder msgBuilder = new StringBuilder();  msgBuilder.append("MusicBox V1.0.0\n");  msgBuilder.append("作者：Leo\_Elegant\n");  msgBuilder.append("(C) 2019 ......");  String title = "关于";  new AlertDialog.Builder(MainActivity.this)  .setIcon(R.drawable.note)  .setTitle(title)  .setMessage(msgBuilder.toString())  .setPositiveButton(R.string.btn\_confirm, new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialog, int which) {  }}).create().show();  }  if (item.getItemId() == R.id.item\_exit){  onBackPressed();  }  return super.onOptionsItemSelected(item);  }  @Override  public void onBackPressed() {  super.onBackPressed();  String title = "提示";  new AlertDialog.Builder(MainActivity.this)  .setIcon(R.drawable.note)  .setTitle(title)  .setMessage("确定要退出吗?")  .setPositiveButton(R.string.btn\_confirm, new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialog, int which) {  myBinder.Release();  finish();  }  })  .setNegativeButton(R.string.btn\_cancel, new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialog, int which) {  }  }).create().show();  }  } |

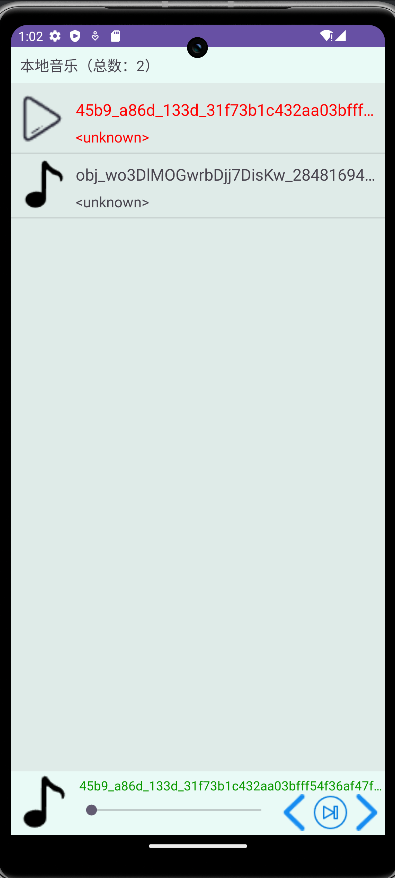
1. MusicService代码(是要文本，不是截图)

|  |
| --- |
| package edu.swpu.iot2022.musicplayer;  import android.app.Service;  import android.content.Intent;  import android.media.AudioManager;  import android.net.Uri;  import android.os.Binder;  import android.os.IBinder;  import androidx.annotation.Nullable;  import android.media.MediaPlayer;  import java.io.IOException;  import java.util.List;  public class MusicService extends Service {  private MediaPlayer mPlayer;  private int seekLength = 0;  private int currentIndex = -1;  private List<MusicInfo> musicList;  @Nullable  @Override  public IBinder onBind(Intent intent) {  return new MyBinder();  }  private void InitPlayer() {  mPlayer = new MediaPlayer();  mPlayer.setAudioStreamType(AudioManager.STREAM\_MUSIC);  }  @Override  public void onCreate() {  super.onCreate();  InitPlayer();  //通过工具类MusicUtils获取音乐信息列表  musicList = MusicUtils.ResolveMusicToList(getApplicationContext());  }  private class MyBinder extends Binder implements MyBinderInterface {  @Override  public void Pause() {  if (mPlayer.isPlaying()) {  mPlayer.pause();  seekLength = mPlayer.getCurrentPosition();  }  }  @Override  public void Resume() {  mPlayer.seekTo(seekLength);  mPlayer.start();  }  @Override  public void Play() {  mPlayer.reset();  Uri path = Uri.parse(musicList.get(currentIndex).getMusic\_path());  try {  mPlayer.setDataSource(String.valueOf(path));  mPlayer.prepare();  } catch (IOException e) {  e.printStackTrace();  }  mPlayer.seekTo(seekLength);  mPlayer.start();  }  @Override  public void PlayNext() {  currentIndex += 1;  if (currentIndex >= musicList.size()) {  currentIndex = 0;  }  seekLength = 0;  Play();  }  @Override  public void PlayPrev() {  currentIndex -= 1;  if (currentIndex <= 0) {  currentIndex = musicList.size() - 1;  }  seekLength = 0;  Play();  }  @Override  public void Release() {  mPlayer.reset();  mPlayer.stop();  mPlayer.release();  }  @Override  public boolean isPlaying() {  return mPlayer.isPlaying();  }  @Override  public int getDuration() {  return mPlayer.getDuration();  }  @Override  public int getCurrentPosition() {  return mPlayer.getCurrentPosition();  }  @Override  public void seekTo(int length) {  seekLength = length;  mPlayer.seekTo(length);  }  @Override  public int getCurrentIndex() {  return currentIndex;  }  @Override  public void setCurrentIndex(int currentIdx) {  currentIndex = currentIdx;  }  }  } |

1. AndroidManifest.xml文件的截图，此处是截图



1. 运行效果截图，多图





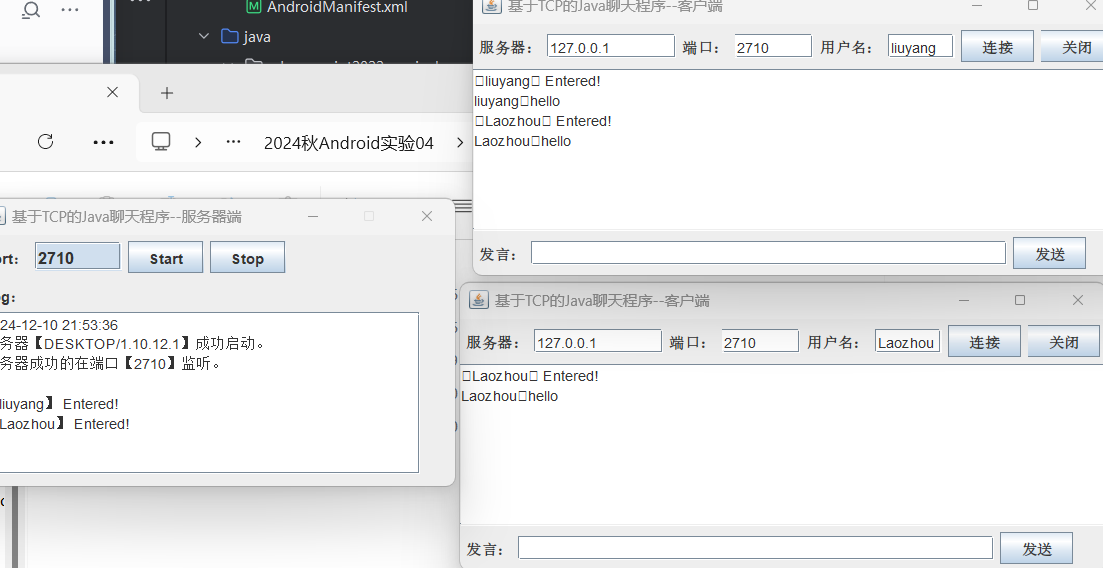


## 网络编程知识准备(20分)

1. URLDemo类的运行效果



1. 聊天服务的运行效果，注意，是两个或三个客户端连接到同一台服务器。截图中，聊天信息中，应该出现你的姓名信息（拼音字母）。



1. 解释客户端是怎么找到服务器的。

获取服务器地址和端口：

服务器在启动时通过 ServerSocket 在用户指定的端口（默认 2710）上监听。

服务器会通过 InetAddress.getLocalHost() 获取本机的 IP 地址和主机名，供日志记录和通知。

客户端连接服务器：

客户端需要知道服务器的 IP 地址和监听的端口号。

客户端通过 Socket 对象连接到服务器的指定 IP 地址和端口。

建立通信：

连接建立后，服务器通过 accept() 方法接收客户端的请求。

服务端为每个客户端创建一个专属的 ThreadSocket 线程，用于管理通信。

## 实验心得体会(10分)

要求不少于100汉字，必须是自己原创，不得雷同。

|  |
| --- |
| 本次实验中，碰到了安卓API33版本限制静态广播,如果想发送广播,需要指定相应的接收类objInent.setClassName("edu.swpu.iot2022.liuyang", "edu.swpu.iot2022.liuyang.TestReceiver");  编写音乐播放器时,遇到不弹出也不获取访问文件管理权限的问题,发现是高版本对权限申请做了细化,图片,音频,视频是三个不同的权限,分开在mainifest中注册,同时还要在通过代码动态申请,在onCreate回调中判断是不是有权限,没有就要申请.  编写音乐播放器时,遇到了不显示右上角菜单的问题,询问大模型后发现是mainifest主题theme的style被设置成NoactionBar的形式,默认不会显示菜单栏,需要手动修改 |