

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

目录视图

摘要视图

订阅

从创业到再就业，浅述对程序员职业生​​涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Android之简易音乐播放器

2015-11-17 20:36 298人阅读 评论(0) 收藏 举报

分类： 移动开发 (38)

版权声明：本文为博主原创文章，未经博主允许不得转载。

布局主要代码之ListView：

[html]

```
01. <span style="font-size:14px;"> <ListView
02.     android:id="@+id/music_list"
03.     android:layout_width="match_parent"
04.     android:layout_height="wrap_content"
05.     android:layout_below="@id/handle_btnlayout"
06.     android:cacheColorHint="#00000000" /*设置为透明，防止滑动时背景变成白色*/
07.     android:dividerHeight="1dip"
08.     android:divider="#88888888" /*设置分割线的颜色和透明度*/
09.     android:listSelector="#00000000"
10.     android:layout_marginBottom="50dp" >
11. </ListView></span>
```

Android之UI界面代码：

[java]

```
01. <span style="font-size:14px;">public class Music extends BaseActivity{ //BaseActivity继承Activity作为所有活动的父类
02.     public static final String ACTION_UPDATEUI = "action.updateUI";
03.     private ListView ls = null;
04.     private Button play_music = null;
05.     private Button previous_music =null;
06.     private Button next_music =null;
07.     private Button repeat_music = null;
08.     private Button shuffle_music =null;
09.     private TextView music_title = null;
10.     private TextView music_duration =null;
11.     public static List<Map<String, String>> list ;
12.     private Boolean start = false;
13.     private Boolean isrepeat = MusicService.repeat;
14.     UpdateUIBroadcastReceiver broadcastReceiver;
15.     @Override
16.     protected void onCreate(Bundle savedInstanceState){
17.         super.onCreate(savedInstanceState);
18.         setContentView(R.layout.home_activity_layout);
19.         play_music = (Button) findViewById(R.id.play_music);
20.         previous_music = (Button) findViewById(R.id.previous_music);
21.         next_music = (Button) findViewById(R.id.next_music);
22.         repeat_music = (Button) findViewById(R.id.repeat_music);
23.         shuffle_music = (Button) findViewById(R.id.shuffle_music);
24.         music_title = (TextView) findViewById(R.id.music_title);
25.         music_duration = (TextView) findViewById(R.id.music_duration);
26.         music_title.setText(MusicService.music_title);
27.         music_duration.setText(MusicService.music_time);
28.
29.         play_music.setOnClickListener(new ButtonsetOnClickListener());
30.         previous_music.setOnClickListener(new ButtonsetOnClickListener());
31.         next_music.setOnClickListener(new ButtonsetOnClickListener());
```

```

32.         repeat_music.setOnClickListener(new ButtonsetOnClickListener());
33.         shuffle_music.setOnClickListener(new ButtonsetOnClickListener());
34.         if(MusicService.isplay){
35.             play_music.setBackgroundResource(R.drawable.pause_selector);
36.         }else{
37.             play_music.setBackgroundResource(R.drawable.play_selector);
38.         }
39.         if(isrepeat){
40.             repeat_music.setBackgroundResource(R.drawable.repeat_all_selector);
41.         }else{
42.             shuffle_music.setBackgroundResource(R.drawable.shuffle_selector);
43.         }
44.         //动态注册广播接收器
45.         IntentFilter filter = new IntentFilter();
46.         filter.addAction(ACTION_UPDATEUI);
47.         broadcastReceiver = new UpdateUIBroadcastReceiver();
48.         registerReceiver(broadcastReceiver, filter);
49.
50.         ls = (ListView) findViewById(R.id.music_list);
51.         //接下来要将这些数据封装成ArrayAdapter对象
52.         SimpleAdapter adapter = this.getAdapter();
53.         ls.setAdapter(adapter);
54.         ls.setOnItemClickListener(new ItemClick()); //点击
55.         ls.setOnItemLongClickListener(null); //长按
56.         ls.setOnItemSelectedListener(null); //勾选
57.     }
58.     private class ButtonsetOnClickListener implements OnClickListener{
59.         Intent intent =new Intent(Music.this, MusicService.class); //记得注册Service
60.         Bundle bundle = new Bundle();
61.         @Override
62.         public void onClick(View v) {
63.             switch(v.getId()){
64.                 case R.id.play_music:
65.                     if(start){
66.                         play_music.setBackgroundResource(R.drawable.play_selector);
67.                         bundle.putBoolean("start", false);
68.                         start = false;
69.                     }else{
70.                         play_music.setBackgroundResource(R.drawable.pause_selector);
71.                         bundle.putBoolean("start", true);
72.                         start = true;
73.                     }
74.                     break;
75.                 case R.id.previous_music:
76.                     bundle.putBoolean("previous", true);
77.                     break;
78.                 case R.id.next_music:
79.                     bundle.putBoolean("next", true);
80.                     break;
81.                 case R.id.repeat_music:
82.                     bundle.putBoolean("repeat", true);
83.                     repeat_music.setBackgroundResource(R.drawable.repeat_all_selector);
84.                     shuffle_music.setBackgroundResource(R.drawable.shuffle_none_selector);
85.                     break;
86.                 case R.id.shuffle_music:
87.                     bundle.putBoolean("shuffle", true);
88.                     shuffle_music.setBackgroundResource(R.drawable.shuffle_selector);
89.                     repeat_music.setBackgroundResource(R.drawable.repeat_none_selector);
90.                     break;
91.                 default:
92.                     break;
93.             }
94.             intent.putExtras(bundle); //Intent传递参数
95.             Music.this.startService(intent); //启动服务
96.         }
97.     }
98.     //定义广播接收器（内部类）
99.     private class UpdateUIBroadcastReceiver extends BroadcastReceiver{
100.         @Override
101.         public void onReceive(Context context, Intent intent) {
102.             music_title.setText(String.valueOf(intent.getExtras().getString("music_title")));
103.             music_duration.setText(String.valueOf(intent.getExtras().getString("music_time")));
104.         }
105.     }
106.     @SuppressWarnings("SimpleDateFormat")
107.     private SimpleAdapter getAdapter(){
108.         // 使用SimpleAdapter来封装数据时必须要把每个显示的数据封装成Map
109.         list=new ArrayList<Map<String, String>>();
110.         SimpleDateFormat format = new SimpleDateFormat("mm:ss");

```

```

111. //取得所有在外部存储卡上的音乐文件的信息 ps : MediaStore.Audio.Media.INTERNAL_CONTENT_URI 获得内置存储
112. Cursor cursor = Music.this.getContentResolver().query(
113.     MediaStore.Audio.Media.EXTERNAL_CONTENT_URI, null, null, null,
114.     MediaStore.Audio.Media.DEFAULT_SORT_ORDER);
115. cursor.moveToFirst();
116. for(int i=0;i<cursor.getCount();i++){
117.     String title = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.TITLE)); //歌曲的名称
118.     String url = cursor.getString(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DATA)); //歌曲文件的路
119.     int duration = cursor.getInt(cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DURATION)); //歌曲的总时长
120.     String hmsString = format.format(duration);
121.     Map<String, String>map =new HashMap<String, String>();
122.     map.put("title", title);
123.     map.put("time", hmsString);
124.     map.put("url", url);
125.     list.add(map);
126.     cursor.moveToNext();
127. }
128. SimpleAdapter adapter =new SimpleAdapter(this, //上下文, 显示对象
129.     list, // 要显示的数据, 一定要上面的格式进行封装
130.     R.layout.music_list, // 列表中要显示的单项信息的布局
131.     new String[]{"title","time"}, // map中的key
132.     new int[]{R.id.music_title_my,R.id.music_time_my}); // 显示到组件
133.
134. return adapter;
135. }
136. //点击事件
137. private class ItemClick implements OnItemClickListener{
138.     @Override
139.     public void onItemClick(AdapterView<?> parent, View view, int position,
140.         long id) {
141.         Intent intent =new Intent(Music.this, MusicService.class); //记得注册Activity
142.         Bundle bundle = new Bundle();
143.         bundle.putBoolean("playing", true);
144.         bundle.putInt("position", position);
145.         intent.putExtras(bundle); //Intent传递参数
146.         Music.this.startService(intent); //启动服务
147.         play_music.setBackgroundResource(R.drawable.pause_selector);
148.     }
149. }
150. }
151. </span>

```

[java]

```

01. <span style="font-size:14px;">public class MusicService extends Service{
02.     public static String music_title = "";
03.     public static String music_time = "";
04.     private MediaPlayer mp ;
05.     private List<Map<String, String>> list = Music.list;
06.     private int position ;
07.     private boolean playing = false;
08.     private boolean start = true;
09.     private boolean previous = false;
10.     private boolean next =false;
11.     public static boolean repeat = true;
12.     private boolean shuffle = false;
13.     public static boolean isplay = false;
14.     Random random = new Random();
15.     @Override
16.     public IBinder onBind(Intent intent) {
17.         return null;
18.     }
19.     @Override
20.     public void onCreate(){
21.         super.onCreate();
22.         /*
23.         final Intent intent = new Intent();
24.         intent.setAction(Music.ACTION_UPDATEUI);
25.         timer = new Timer();
26.         task = new TimerTask() {
27.             @Override
28.             public void run() {
29.                 intent.putExtra("count", ++count);
30.                 sendBroadcast(intent);
31.             }
32.         };

```

```

33.         timer.schedule(task, 1000,1000);
34.     */
35. }
36. @SuppressWarnings("NewApi")
37. @Override
38. public int onStartCommand(Intent intent, int flags, int startId){
39.     start = intent.getBooleanExtra("start", true);    //取不到值，默认false
40.     playing = intent.getBooleanExtra("playing", false);
41.     previous = intent.getBooleanExtra("previous", false);
42.     next = intent.getBooleanExtra("next", false);
43.     if(repeat){
44.         shuffle = intent.getBooleanExtra("shuffle", false);
45.         if(shuffle){
46.             repeat = false;
47.         }
48.     }else if(shuffle){
49.         repeat = intent.getBooleanExtra("repeat", false);
50.         if(repeat){
51.             shuffle = false;
52.         }
53.     }
54.     if(playing){
55.         position = intent.getIntExtra("position",1);
56.         Uri uri = Uri.parse(list.get(position).get("url"));
57.         playsong(uri);
58.     }else if(previous){
59.         previous song();
60.     }else if(next){
61.         next song();
62.     }
63.     if(mp!=null){
64.         if(start){
65.             mp.start();
66.             isplay = true;
67.         }else{
68.             mp.pause();    //暂停播放
69.             isplay = false;
70.         }
71.     }
72.     return super.onStartCommand(intent, flags, startId);
73. }
74. private void playsong(Uri uri){
75.     if(mp!=null&&mp.isPlaying()){
76.         mp.stop();    //停止播放
77.     }
78.     mp = MediaPlayer.create(this, uri);
79.     mp.start();
80.     mp.setOnCompletionListener(new CompletionListener());
81.     music_title = list.get(position).get("title");
82.     music_time = list.get(position).get("time");
83.     //发送广播，改变界面
84.     Intent intent = new Intent();
85.     intent.setAction(Music.ACTION_UPDATEUI);
86.     Bundle bundle = new Bundle();
87.     bundle.putString("music_title",music_title);
88.     bundle.putString("music_time", music_time);
89.     intent.putExtras(bundle);
90.     sendBroadcast(intent);
91. }
92. //获取随机数
93. private int randomInt(){
94.     int num = random.nextInt(list.size());
95.     if(num>=1&&num<=list.size()-1){
96.         return num;
97.     }else{
98.         return randomInt();
99.     }
100. }
101. private void previous song(){
102.     if(repeat){
103.         position--;
104.         if(position<=1){
105.             position = list.size()-1;
106.         }
107.     }else{
108.         position = randomInt();
109.     }
110.     String urlstr = list.get(position).get("url");
111.     Uri uri = Uri.parse(urlstr);

```

```
112.         playsong(uri);
113.     }
114.     private void nextsong(){
115.         if(repeat){
116.             position++;
117.             if(position>=list.size()){
118.                 position = 1;
119.             }
120.         }else{
121.             position = randomInt();
122.         }
123.         String urlstr = list.get(position).get("url");
124.         Uri uri = Uri.parse(urlstr);
125.         playsong(uri);
126.     }
127.     private class CompletionListener implements OnCompletionListener{
128.         @Override
129.         public void onCompletion(MediaPlayer mp) {
130.             nextsong();
131.         }
132.     }
133.     @Override
134.     public void onDestroy(){
135.         super.onDestroy();
136.         if(mp!=null) mp.release();    //释放资源
137.         stopSelf();
138.         // timer.cancel();
139.     }
140. }
```

顶 0 踩 0

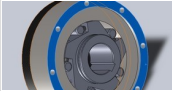
- [上一篇](#) Tomcat设置开启时自动访问某个servlet类存在的问题
- [下一篇](#) 关于jmf不能播放mp3的问题解决

相关文章推荐

- 博文汇总
 - 一个简易音乐播放器(android)
 - Android开发笔记之简易音乐播放器
 - 【Android】简易音乐播放器(介绍使用Service和...
 - 使用Qt做一个简易音乐播放器【Phonon浅谈--续】
- Android之简易的音乐播放器
 - 一个简易的音乐播放器
 - Android 调用系统播放器播放音乐或视频
 - 使用Flex制作一个简易版的音乐播放器 该有的功能...
 - Android简易音乐播放器之界面实现（第一篇）



代缴社保公司



逆变器



电磁加热设备



断路器测试仪



制作自己的网站



质量流量控制器



网站制作

猜你在找

- 机器学习之概率与统计推断
 - 机器学习之凸优化
 - 响应式布局全新探索
 - 深度学习基础与TensorFlow实践
 - 前端开发在线峰会
- 机器学习之数学基础
 - 机器学习之矩阵
 - 探究Linux的总线、设备、驱动模型
 - 深度学习之神经网络原理与实战技巧
 - TensorFlow实战进阶：手把手教你做图像识别应用