移动互联网技术及应用

大作业报告

姓名	班级	学号
吴佩鞠	2017211307	2017211352

2020.6

目录

1. 相关技术

- App 启动动画
- 用 retrofit 解析 json 文件
- 使用 recycleView 显示视频列表,显示每个视频的必要信息
- 使用 Glide 加载封面图
- 从视频流点击某个视频封面进入播放页面
- 根据视频信息的 URL 播放视频
- 单击视频窗口暂停视频
- 双击视频窗口弹出点赞爱心图标

2. 系统功能需求

2.1 信息流列表显示（包含封面图）

2.2 视频播放

从视频信息流点击某个视频封面进入播放页面

根据视频信息的 url 播放视频

单击视频窗口暂停/继续

3. 系统设计与实现

3.1 App 启动动画

3.1.1 新建一个叫 Splash 的 Activity，将准备好的启动图片放到 drawable 目录下，并修改 Splash 的 xml 布局文件

//隐藏状态栏、标题栏达到全屏效果，用 sleep()使程序休眠三秒再启动 MainActivity
public class Splash extends AppCompatActivity {

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

```
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
    getSupportActionBar().hide();
    setContentView(R.layout.activity_splash);
    Thread myThread=new Thread(){//创建子线程
        @Override
        public void run() {
            try{
                sleep(3000);
                Intent it=new Intent(getApplicationContext(), MainActiv
ity.class);
                startActivity(it);
                finish();//关闭当前活动
            }catch (Exception e){
                e.printStackTrace();
            }
        }
    };
    myThread.start();//启动线程
   }
}
```

3.1.2 在 AndroidManifest 中将启动页面修改为 Splash：

```
<activity android:name=".Splash" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <action android:name="android.intent.action.VIEW" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

3.2 用 retrofit 解析 json

3.2.1 新建一个类来存放视频信息

```
public class VideoBean {
    private String id;
    private String feedurl;
    private String nickname;
    private String description;
    private int  likecount;
    private String avatar;

    public String getId(){
        return id;
    }
    public void setId(String id){
        this.id=id;
```

```
    }

    public String getFeedurl(){
        return feedurl;
    }
    public void setFeedurl(String feedurl){
        this.feedurl=feedurl;
    }

    public String getNickname(){
        return nickname;
    }
    public void setNickname(String nickname){
        this.nickname=nickname;
    }

    public String getDescription(){
        return description;
    }
    public void setDescription(String description){
        this.description=description;
    }

    public String getAvatar(){
        return avatar;
    }
    public void setAvatar(String avatar){
        this.avatar=avatar;
    }

    public int getLikecount(){
        return likecount;
    }
    public void setLikecount(int likecount){
        this.likecount=likecount;
    }
}
```

Retrofit 是一个 HTTP Client 库，把 REST API 返回的数据转化为 Java 对象方便操作。将解析好的数据放入 ListmDataset，提供给 MainActivity。

```
Retrofit retrofit = new Retrofit.Builder()
        .baseUrl("https://beiyou.bytedance.com/")
        .addConverterFactory(GsonConverterFactory.create())
        .build();

ApiService apiService = retrofit.create(ApiService.class);
apiService.getR().enqueue(new Callback<JsonArray>() {
    @Override
```

```
    //解析 JsonArray
    public void onResponse(Call<JsonArray> call, Response<JsonArray> re
sponse) {
        Log.d("Vretrofit", String.valueOf(response.body().isJsonArray
()));
        Log.d("Vretrofit", response.body().toString());

        String str=response.body().toString();
        Type type=new TypeToken<List<VideoBean>>(){}.getType();

        Gson gson=new Gson();
        mDataset=gson.fromJson(str,type);
        }
      }
```

3.3 使用 recycleView 显示视频列表

3.3.1 确定一个 item 的内容

如图所示，一个 item 由两个 ImageView,三个 TextView 组成。其中，大的 ImageView 用来加载视频的封面图，小的 ImageView 用来加载爱心图标；三个 TextView 分别用来加载视频作者、视频描述和视频点赞数。

3.3.2 将 item 作为 RecycleView 的一个组件

```
<android.support.v7.widget.RecyclerView
    android:id="@+id/rv_list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    tools:listitem="@layout/im_list_item" />
<!--im_list_item.xml 是一个 item 的描述-->
```

3.3.3 编写 RecycleView 的适配器 MyAdapter,显示每个视频必要的信息

```
//RecyleView 里一个 item 的所有组件
public class NumberViewHolder extends RecyclerView.ViewHolder implement
s View.OnClickListener {

    private final  ImageView imageView;
    private final TextView textAuthor;
    private final TextView textDescription;
    private final TextView textLikecount;

    public NumberViewHolder(@NonNull View itemView) {
        super(itemView);

        imageView =(ImageView)itemView.findViewById(R.id.ImageView);
        textAuthor=(TextView)itemView.findViewById(R.id.textView_Autho
r);
        textDescription=(TextView)itemView.findViewById(R.id.textView_d
es);
        textLikecount=(TextView)itemView.findViewById(R.id.textView_lik
ecount) ;

        itemView.setOnClickListener(this);
    }

//为每个子项绑定数据
public void onBindViewHolder(@NonNull final NumberViewHolder numberView
Holder, int position) {

            numberViewHolder.textAuthor.setTextSize(18);
            numberViewHolder.textDescription.setTextSize(18);
            numberViewHolder.textLikecount.setTextSize(18);
            numberViewHolder.textAuthor.setTextColor(Color.WHITE);
            numberViewHolder.textDescription.setTextColor(Color.WHITE);
            numberViewHolder.textLikecount.setTextColor(Color.WHITE);
            numberViewHolder.textAuthor.setText("@"+mDataset.get(positi
on%10).getNickname());
            numberViewHolder.textDescription.setText("#"+mDataset.get(p
osition%10).getDescription());
            numberViewHolder.textLikecount.setText(String.valueOf(mData
set.get(position%10).getLikecount()));
```

```
}
```

3.4 使用 Glide 加载封面图

Glide 是一个 Android 上的图片加载和缓存库，其目的是实现平滑的图片列表滚动效果。

```
//截取视频的第一帧作为视频的封面
Glide.with(numberViewHolder.imageView.getContext())

        .setDefaultRequestOptions(

                new RequestOptions()

                        .frame(1000000)//单位微秒

                        .centerCrop()

        )

        .load(mDataset.get(position%10).getFeedurl())
        .into(numberViewHolder.imageView);
```

3.5 从视频流点击某个视频封面进入播放页面

3.5.1 在 MainActivity 监听用户点击的是哪一个 Item，将 Item 的位置信息（播放页面要播放的是哪个视频）通过 Intent 来传递。

```
public void onListItemClick(int clickedItemIndex) {
    Log.d(TAG, "onListItemClick: ");
    if (mToast != null) {
        mToast.cancel();
    }
    String toastMessage = "Item #" + (clickedItemIndex+1) + " clicked.";
    mToast = Toast.makeText(this, toastMessage, Toast.LENGTH_LONG);

    Intent intent1 = new Intent();
    intent1.setClass(this,VideoActivity.class);//播放页面
    String str1 = String.valueOf(clickedItemIndex);
    String str2 = mDataset.get(clickedItemIndex%10).getFeedurl();
    intent1.putExtra("item",str1);
    intent1.putExtra("video",str2);
    startActivity(intent1);
```

```
}
```

3.5.2 在新建的 VideoActivity 根据 video 标签找到要播放的视频 url

```
String feedurl =bundle.getString("video");
```

3.6 根据视频信息的 URL 播放视频

使用 Uri.parse（）来解析 url,播放视频

```
String feedurl =bundle.getString("video");

 videoView.setVideoURI(Uri.parse(feedurl));
```

3.7 单击视频窗口暂停视频

3.7.1 实现一个 View.OnTouchListener 的接口 MyClickListener

```
public class MyClickListener implements View.OnTouchListener {

    private static int timeout=400;
    private int clickCount = 0;//记录连续点击次数
    private Handler handler;
    private MyClickCallBack myClickCallBack;
    public interface MyClickCallBack{
        void onSimpleClick();//点击一次的回调
        void onDoubleClick();//连续点击两次的回调
    }


    public MyClickListener(MyClickCallBack myClickCallBack) {
        this.myClickCallBack = myClickCallBack;
        handler = new Handler();
    }
```

3.7.2 增加布局监听事件

```
public void onSimpleClick() {
    //showToast("单击了");
    if (videoView.isPlaying()==true){
        videoView.pause();
        Log.d("flag",String.valueOf(videoView.isPlaying()));

    }
    else
    {
```

```
            videoView.start();
            //Log.d("flag","start");
            Log.d("flag",String.valueOf(videoView.isPlaying()));
            //flag=0;
        }

}
```

3.8 双击视频窗口弹出点赞爱心图标

双击弹出点赞爱心图标源代码来自
https://github.com/KevinYou128/HotHeart/tree/master

3.8.1 在项目根目录的 build.gradle 文件中加入

```
allprojects {
      repositories {
            ...
            maven { url 'https://jitpack.io' }
      }
}
```

3.8.2 添加依赖项

```
dependencies {
      implementation 'com.github.KevinYou128:HotHeart:v1.1'
}
```

3.8.3 在布局文件里调用

3.8.4 在 VideoActivity 里添加布局的监听事件

```
heartFrameLayout.setOnTouchListener(new MyClickListener
        (new MyClickListener.MyClickCallBack() {

            @Override
            public void onSimpleClick() {
                if (videoView.isPlaying()==true){
                    videoView.pause();
                    Log.d("flag",String.valueOf(videoView.isPlaying()));

                }
                else
                {
                    videoView.start();
                    Log.d("flag",String.valueOf(videoView.isPlaying()));
```

```
            }

        }

        @Override
        public void onDoubleClick() {
        }
    }));
```

4. 系统可能的扩展

4.1 将来可以进一步可使用 ViewPage2 来代替 RecycleView，使得 APP 的效果与抖音更加接近。

4.2 可以对视频进行评论。

4.3 加入视频录制拍摄功能。

4.4 加入视频上传功能。

5. 总结体会

现今的生活已经离不开手机了，生活中处处可见移动互联网给千家万户带来的便利。在之前，我只是一个移动互联网的使用者，单纯地享受它为我提供的服务。现在，我学会了转换的角度，从一个开发者的角度来看待移动互联网和移动互联网技术。通过设计一个小小的视频播放器，让我明白了任何一个成功的应用里都有许多不被人知道的努力与坚持，每一个小小的功能都需要经过精细的设计，才能在一个大程序里发挥它的作用。