

移动互联网技术及应用

大作业报告

姓名	班级	学号
史伯卿	2017211319	2017211770

2020.6

目录

- 1. 相关技术..... 3
- 2. 系统功能需求..... 3
- 3. 系统设计与实现..... 4
- 4. 系统可能的扩展..... 7
- 5. 总结体会..... 7

1. 相关技术

在本系统中，我主要负责页面设计以及视频信息流列表显示（包含封面图）。在这一部分需要完成的工作包括：1、到 API:<https://beiyou.bytedance.com/api/invoke/video/invoke/video> 获取视频信息，包括视频描述、作者名点赞数量、视频 url 等；2、使用 RecyclerView 显示视频列表（一页显示多个 item）；3、使用 Glide 加载封面图；4、为每个 item 添加点击响应，为之后点击就跳转到播放界面留出接口。

技术简述：

1、使用 retrofit 进行网络资源获取

Retrofit 是安卓上最流行的 HTTP Client 库之一，性能好，处理快，使用简单。

具体使用包括以下几个步骤：①定义一个接口（封装 URL 地址和数据请求）；②实例化 Retrofit；③通过 Retrofit 实例创建接口服务对象；④接口服务对象调用接口中方法，获得 Call 对象；⑤Call 对象执行请求（异步、同步请求）。

2、使用 RecyclerView 显示视频列表

RecyclerView 是 Android 一个更强大的控件，其不仅可以实现和 ListView 同样的效果，还有优化了 ListView 中的各种不足。其可以实现数据纵向滚动，也可以实现横向滚动（ListView 做不到横向滚动）。

具体使用包括以下几个步骤：①设计并实例化 LayoutManager 类；②设计并实例化 Adapter 类；③获取 RecyclerView 对象；④设置 RecyclerView 对象的 LayoutManager、Adapter。

3、使用 Glide 加载封面图

Glide 是一个被 google 所推荐的图片加载库，被广泛运用在 google 的开源项目中。

Glide 的一个完整的请求至少需要三个参数，一是上下文 context，二是源 url，三是目标 view，只要规定好这三个参数即可将图片展示到指定的 view 上。在系统中，展示的封面图我们选择视频的第一秒处的一帧，所以要用到 Glide 从视频中获取特定位置的一帧的功能，要进行额外的设置，主要通过 Glide 中的 RequestOptions 来实现。

2. 系统功能需求

需求分析：近年来抖音火的不行，并且多家短视频 APP 陆续上线并且都取得了不错的成绩，比如西瓜视频、微视等等。这反映了短视频市场规模仍存在上升空间，而且随着 5G 网络的加速建成，短视频行业或将进入下一个快速发展阶段。因此，如果像 QQ 推广微视那样将本产品也作为一个插件放到一款常用软件上，比如外卖软件，让用户等餐的时候一直刷视频，那就算这个短视频软件做的再难用，内容再无聊，也会有用户使用的。这就是为什么如今短视频软件如同雨后春笋一般冒出但市场仍未达到饱和的原因，也是我们现在仍然在做短视频软件的原因。

用户群体：闲得无聊打发时间的年轻人。

系统功能需求：本系统总体来说就是一个短视频播放器。主界面使视频信息列表，用户可以点击进入该视频的播放界面。由于现在的年轻人都讲究效率，所以视频信息流页面将每个 item 再不影响美观，不影响视频信息展示的前提下可以做的稍微小一些，使一页能够放下多个 item。用户在综合考虑视频的内容简介、作者、点赞数、封面图等信息后可以自行选择要不要点进去看。如果用户选择观看视频，那么点击这个 item 就会跳转到播放界面，并且从播放界面再回到视频信息流页面的时候是从上次进入的地方继续查看列表，比较符合人的一般认知。在播放界面视频可以单击暂停/继续播放，双击点赞冒出小心心，并且隐藏进度条避免造成累赘。双击点赞冒出小心心使借鉴的当前一些较大的短视频 APP，符合大众审美，增加流量。

3. 系统设计与实现

（此句阅后删除： 应包括 总体设计、系统组成、模块设计， 关键代码的解释）

本系统分为两个大模块：视频信息流页面和播放页面。我主要负责的是视频信息流页面。视频信息流页面主要包括两个模块，即上述网络资源获取模块、recyclerlist 模块，其中 recyclerlist 模块的设计比较复杂，内部要自己完成 LayoutManager、Adapter 的设计，在 Adapter 中还要再每个 item 加载的时候使用 Glide 完成封面图的加载。

网络资源获取模块：

我们按照①定义一个接口（封装 URL 地址和数据请求）；②实例化 Retrofit；③通过 Retrofit 实例创建接口服务对象；④接口服务对象调用接口中方法，获得 Call 对象；⑤Call 对象执行请求（异步、同步请求）的顺序完成本模块的应用。

第一步，我们定义一个接口，由于 baseUrl 会在实例化 Retrofit 的时候封装到 Retrofit 对象中，所以这里只需要写 api 地址的后半部分即可，如下左图所示。此外，为了将获取到的视频信息解析，沃尔玛还需要定义一个 VideoInfo 类，用来接收获得的视频信息并加载到 list 中。在分析了 api 中提供的视频信息内容后，我认为该 VideoInfo 类应当包含的成员如下右图所示。

```
public interface ApiService {
    @GET("invoke/video/invoke/video")
    Call<List<VideoInfo>> getArticles();
}

@SerializedName("_id")
public String _id;
@SerializedName("feedurl")
public String feedUrl;
@SerializedName("nickname")
public String nickname;
@SerializedName("description")
public String description;
@SerializedName("likecount")
public int likeCount;
@SerializedName("avatar")
public String avatarUrl;
```

第二步，我们按如下方式实例化 Retrofit，并封装 baseUrl“https://beiyou.bytedance.com/api/”

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

第三步，通过 Retrofit 实例创建接口服务对象

```
ApiService apiService = retrofit.create(ApiService.class);
```

第四步，此时便可以调用接口服务对象的接口方法进行数据获取，并将数据存储到内存数据结构中，便于之后使用。

```
apiService.getVideos().enqueue(new Callback<List<VideoInfo>>() {
    @Override
    public void onResponse(Call<List<VideoInfo>> call, Response<List<VideoInfo>> response) {
        if (response.body() != null) {
            Log.d( tag: "retrofit", Integer.toString(response.body().size()));
            List<VideoInfo> videoInfos = response.body();
            Log.d( tag: "retrofit", videoInfos.toString());
            if (videoInfos.size() != 0) {
                mAdapter.setContext(getApplicationContext());
                mAdapter.setData(videoInfos);
                mAdapter.notifyDataSetChanged();
            }
        }
    }

    @Override
    public void onFailure(Call<List<VideoInfo>> call, Throwable t) {
        Log.d( tag: "retrofit Fail", t.getMessage());
    }
});
```

在上图中看到，如果我们获取到了回应并且回应的 body 部分不为空，那么就可以解析 body，将 body 中 i 哦那个的 list 直接反序列化转化成 VideoInfo 的 list。转换成功之后我们便将其设置给 mAdapter，作为 recyclerlist 的数据源。

视频信息流展示模块：

此模块主要由 RecyclerView 实现，所以要实现的子模块包括 LayoutManager、Adapter。其中 LayoutManager 在扩展功能做 viewpage2 的时候需要自己重写。由于 viewpage1 和 viewpage2 在视频流信息的展示方面主要区别就在于 LayoutManager，别的地方区别不大，所以在这里一起阐述。

LayoutManager 部分：

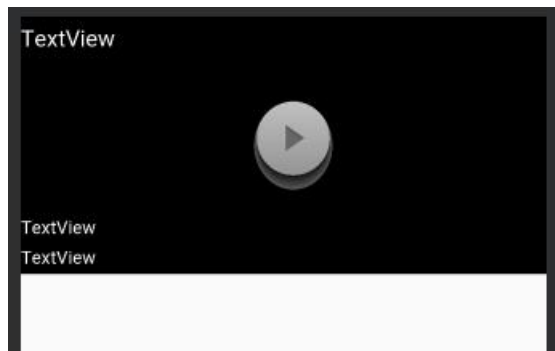
在 viewpage1 中，只要使用 androidx 中的 LinearLayoutManager 类即可，我们将它实例化便可用作 RecyclerView 的布局器。

在 viewpage2 部分，由于我们希望实现这样的功能：每页只显示一个 item，在滑动的时候根据我们滑动的幅度来判断我们是否希望翻到上/下一个视频，如果滑动出现了滑动事件，那么我们就自动计算当前屏幕上中间的 item 是哪个，这样就可以做到小肚肚华东不翻页，大幅度滑动才翻页，保证任何时候屏幕上都显示唯一一个完整的 item。具体实现见代码。

Adapter 部分：

两个 viewpage 的 Adapter 部分基本工作原理是一样的，以 viewpage1 为例进行说明。在重写 Adapter 类的时候，我们需要重写以下函数：onCreateViewHolder、onBindViewHolder、getItemCount。第一个在将 viewholder 类实例化的时候触发，做初始化用；第二个在每次下

滑 recyclerlist 出现一个新的 item 的时候触发，将新的 item view 绑定一个 viewholder；第三个用来返回当前 recyclerlist 中 item 总数。我们对一个 item 的设计是这样的：



它包括三个 textview，分别显示视频描述、视频作者以及视频点赞数，此外，有一个 imagebutton，用来加载封面图并提供点击效果，最后还有一个没什么用只是为了美观的播放键。在 viewholder 初始化的时候要将上述所有 view 找到并绑定到自身成员，方便后边是用。在 onBindViewHolder 中，我们对 view 们进行设置，用的就是 adapter 的数据 VideoInfo 的 list。我们通过 position 获取当前 item 是 recyclerview 中第几个 item，之后获取 VideoInfo list 中对应位置的的结构体，将机构体内数据分别赋值给上述 view。其中封面图的加载是比较特别的，它需要用到 Glide 库。具体的用法为：

```
Glide.with(mContext)
    .setDefaultRequestOptions(
        new RequestOptions()
            .frame(1000000)
            .centerCrop()
    )
    .load(mDataset.get(position).feedUrl)
    .into(holder.cover);
```

以上代码使用 Glide 实现了将 url 为 mDataset.get(position).feedUrl 的视频的第 1000000nm，即 1s 位置的一帧作为封面图加载到 imagebutton 上。

Recyclerlist 部分：

这一部分主要依赖于前面完成的 LayoutManager 和 Adapter。

```
recyclerView = (RecyclerView)findViewById(R.id.RV_covers);
layoutManager = new LinearLayoutManager(context, this);
recyclerView.setLayoutManager(layoutManager);
mVideoInfos = new ArrayList<>();
mAdapter = new MyAdapter();
recyclerView.setAdapter(mAdapter);
```

按照以上代码将 recyclerlist 的 LayoutManager 和 Adapter 设置完成之后，只要在 retrofit 获取到数据之后将数据传给 mAdapter，再提醒数据集发生改变进行更新即可正常使用。界面设计如下：



此外，在主界面还设计了 **vp2** 按键，可以切换到 **viewpage2** 模式。相应的，**viewpage2** 该位置是 **vp1** 按键，可以切换回来，让用户按照喜好进行选择。

4. 系统可能的扩展

该系统在之后还可以加入消息列表、根据推荐算法按照用户喜好进行视频推荐等功能，用户也可以选择喜欢的作者或作品进行收藏，在获取数据的时候多获取同类型的数据，或者同作者的视频。考虑到目前视频平台主要针对短视频播放，在客户量大之后，可能会根据不同的用户需求考虑长视频播放，例如电影、课程视频等。此外，还可以添加上传作品等功能，实现平台的多元化。

5. 总结体会

移动互联网技术为什么当下会越来越火，我认为就是因为移动二字。试问谁有那么多时间一直在一个地方呆着，随着手机以及移动网络的发展，促进了人们将时间碎片化，而时间的碎片化也会带来移动算以及移动互联网的进一步的发展，这二这是相辅相成的。我们在进行移动开发的时候也应当注意到这一点。可能人们等汽车加个油，或者等个外卖的时间，就会将碎片化的时间加以利用，此时，短视频无疑比电影之类的娱乐更为合适，可以说这是更为适

合移动互联网的内容。我们在进行移动应用开发的时候要总和移动互联网的特性以及人们对于移动网络的需求共同进行思考，这样才能开发出一个合大众胃口的应用。同理，在实现的时候，我们也应该进行类似的考虑，比如在封面图加载的时候先使用占位图，或者在视频播放的时候预加载下一个视频，这样可以针对移动互联网偶尔不稳定的问题大幅优化用户体验。本次使用 **android studio** 进行系统的开发我学会了移动开发时的思路以及方法，对移动互联网有了更深的理解。更具体的包括 **Recyclerlist** 的使用、**layoutmanager** 的重写等具体技术方面更是收获颇丰，总体来说是一次十分有意义的大作业。