

《Android应用开发》综合设计说明书

**题 目：**基于Android的视频分享软件

**组长学号：** 1800301613

**组长姓名：** 黄中翔

**组员学号：** 1800300217

**组员姓名：** 黄锦泰

**组员学号：** 1800300814

**组员姓名：** 黄振倚

**指 导 教 师：** 秦兴国

2020 年 9 月 20 号

# 选定任务列表

C．基于Android的视频分享软件**。**用户可将视频分享至平台以供其他用户浏览，用户可对喜欢的视频进行点赞保存分享等操作。

# 摘 要

随着人们生活水平的提高，人们的日常生活也日益丰富，人们对社交的软件需求就变得愈发强烈。一个界面美观、流畅的社交软件可以让用户在分享自己的状态与观看别人分享的状态时有更好的体验感。而视频是一种能很好的分享自己状态的一种方式。在Btvideo 软件上，用户可以将自己拍摄的视频分享到平台上以供其他用户浏览，没有太多繁杂的内容，只有简单的标题以及视频。同时用户可以对自己喜欢的视频进行点赞收藏。

关键词：Android 视频分享 视频上传 视频点赞 视频收藏

目录

[1.前言 1](#_Toc28158738)

[1.1课题背景，目的和意义 1](#_Toc28158739)

[1.2相关技术说明 1](#_Toc28158740)

[2.概要设计 1](#_Toc28158741)

[2.1需求描述 1](#_Toc28158742)

[2.2.整体开发设计架构 2](#_Toc28158743)

[3.详细设计 2](#_Toc28158744)

[3.1安卓部分 2](#_Toc28158745)

[3.2后端部分 9](#_Toc28158746)

[4．心得体会 22](#_Toc28158747)

[4.1综合学习心得 22](#_Toc28158748)

5.项目分工……………………………………………………………………………………………………...

[附录 23](#_Toc28158749)

# 1.前言

## 1.1课题背景，目的和意义

随着经济的发展，人们的日常生活越来越好，温饱已经不是人们的基本需求了。人民的生活日渐丰富，而分享已经成为日常生活中的一部分。就像微信的朋友圈以及新浪的微博一样，这类社交软件的使用可以拉近人与人之间的距离，丰富我们的日常生活。

视频分享APP是社交APP中的一个细分领域。视频分享APP开发为年轻人提供分享视频的平台，用户可以随时随地把自己喜欢的视频分享给其他人，进而拉近人与人之间的距离。本次课题所制作的视频分享软件 App分享内容只保留了视频以及标题，并设计了美观的使用界面。用户可以拥有更好的观感体验。在Btvideo App中，用户可以对自己喜欢的视频进行点赞收藏，或者进行下载。

## 1.2相关技术说明

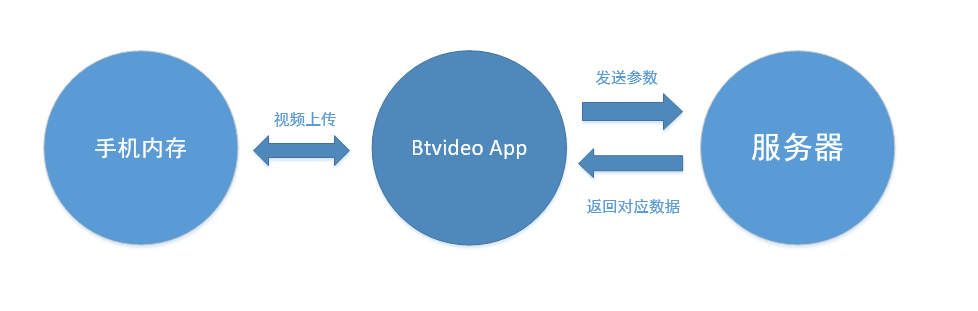
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **开发部分** | **开发平台** | **开发环境** | **开发语言** | **系统运行环境** |
| **安卓** | android studio | Java1.8 | Java | android 4.0及以上 |
| **后端** | IntelliJ IDEA | Java1.8 | Spring boot | Windows10 |

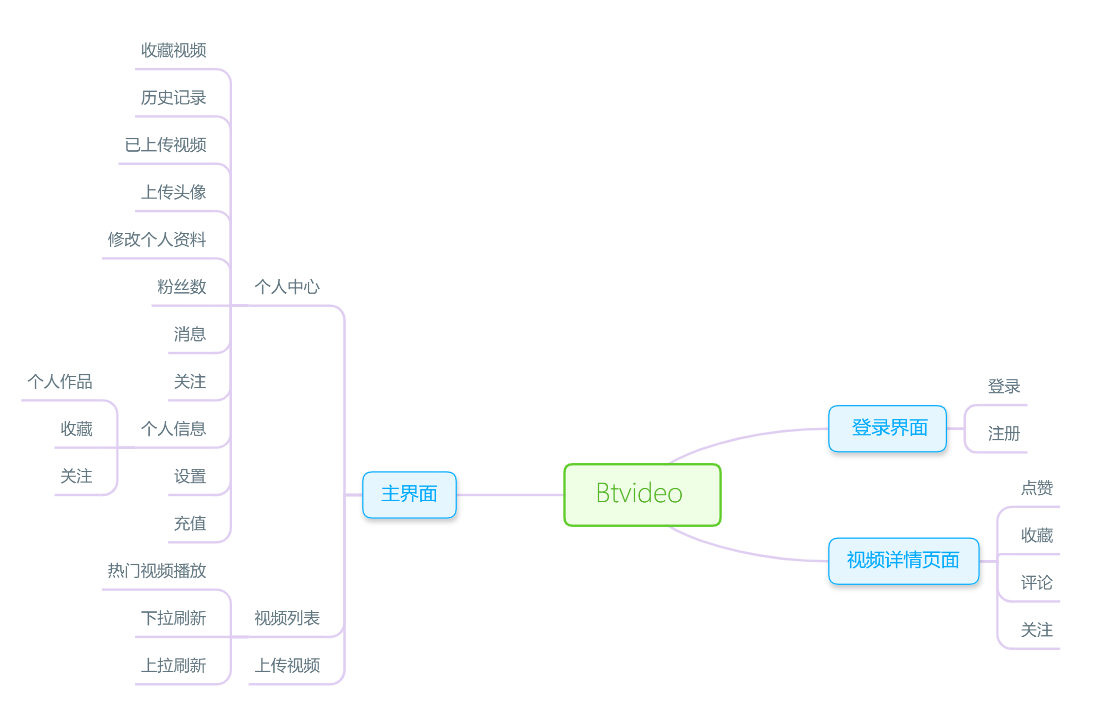
# 2.概要设计

## 2.1需求描述

制作一款APP，通过这款APP，用户可以将自己手机中的视频分享至平台以供其他用户浏览。对于分享至平台的视频，用户可对喜欢的视频进行点赞和收藏操作。可以查看自己发布的视频、点赞的以及收藏的视频。还可以将喜欢的视频下载到手机本地。

## 2.2.整体开发设计架构





# 3.详细设计

## 3.1安卓部分

#### 3.1.1 开发运行环境

1.后端语言：Spring boot

2.前端语言：XML

3.开发平台：android studio 3.4

4.运行环境: android 4.0及以上系统。

#### 3.1.2功能实现分析

（1）登陆

首先打开的是APP的登陆，在用户输入账号密码后，点击登陆按钮后，会通过Rxjava框架向服务器中对应的接口发送带参的请求，根据返回的信息判断登陆是否成功。

点击**记住密码**后，若登陆成功，后端返回登录成功的信息，并且自动生成一个token信息,以便下次登录的时候直接登录。通过SharedPreferences将密码保存在该软件的内部文件夹中。下次打开登陆界面时可以直接读取文件夹中的内容。

（2）注册

注册界面与登录界面大体相同，都是通过后端验证来判断输入是否有误。通过Retrofit对数据进行传递，注册成功会返回数据的信息。

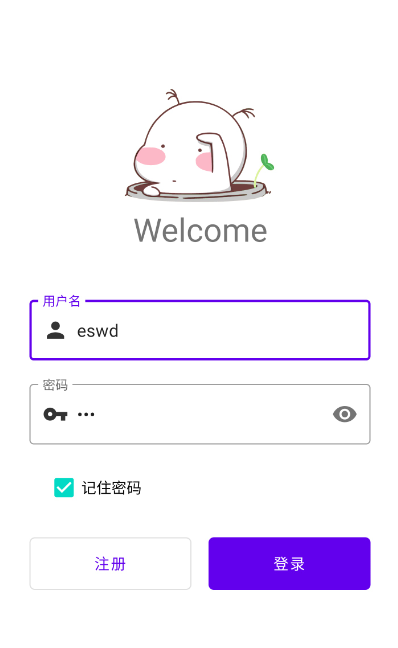
 

图3.1 登陆界面 图3.2 注册界面

（3）主界面

主页面是视频，底部栏有三个按钮，分别是首页、上传视频和个人中心。主界面视图如图3.3所示。

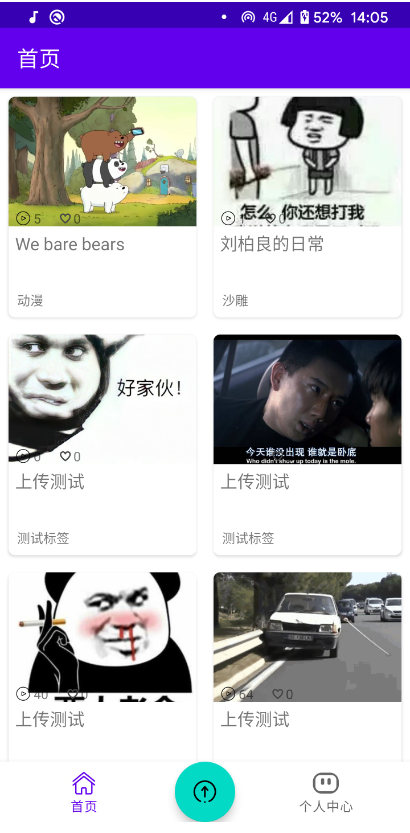


图3.3 主界面

（4）视频内容

视频内容里简介和评论，简介里面可以进行点赞和收藏，评论区可以发表评论。视频简介如图3.4所示，视频评论如图3.5所示。

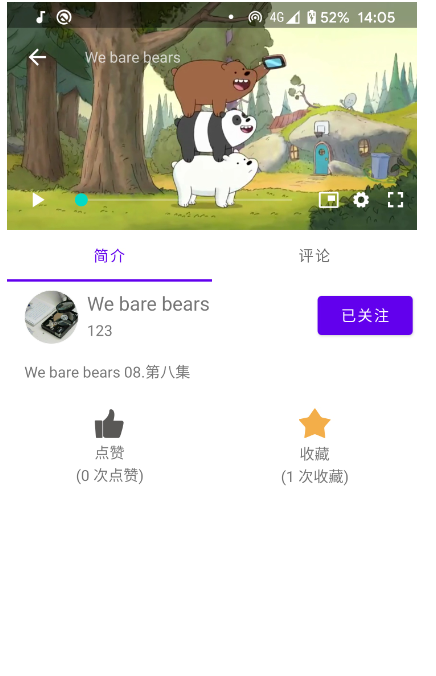
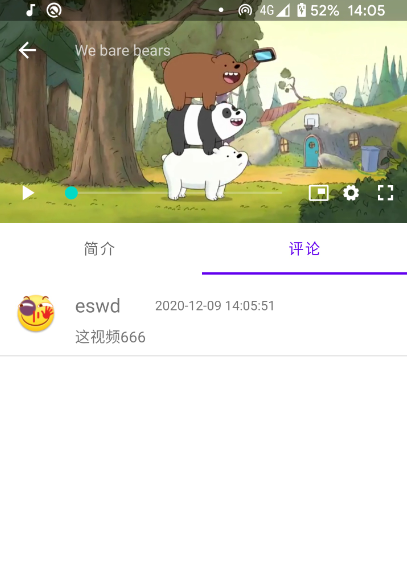
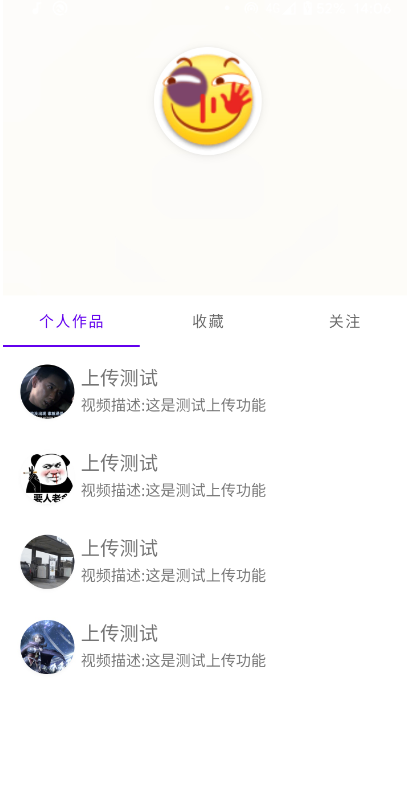
 

图3.4视频简介 图3.5视频评论

（5）上传视频功能

< -- 功能介绍-- >

上传功能如图3.6所示，上传视频如图3.7所示。

3.5上传功能 3.6上传视频

（6）个人中心

个人中心有三个碎片，分别是个人作品、收藏、关注。个人作品如图3.7所示，收藏如图3.8所示，关注如图3.9所示。

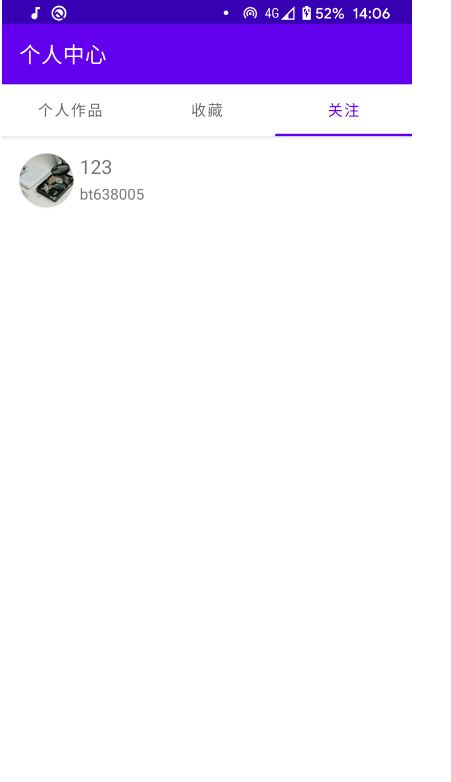
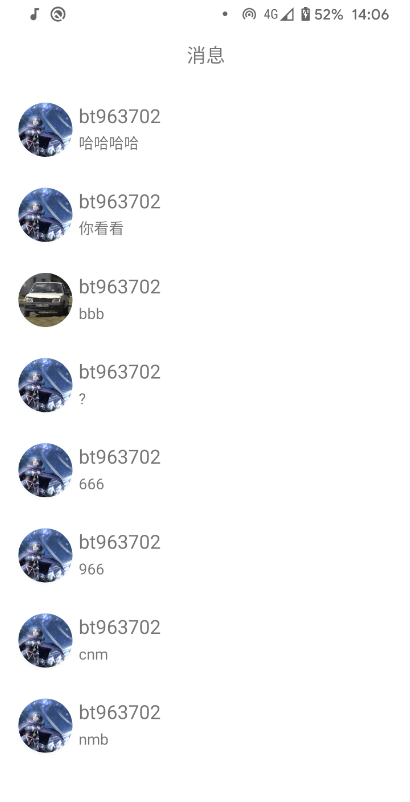
  

图3.7 个人作品 图3.8 收藏 图3.9 关注

（7）消息内容

< --消息功能 -- >

消息功能如图3.10所示



3.10 消息功能

图3.11视频详情

## 3.2后端部分

#### 3.2.1开发运行环境

1.后端语言：Spring boot

3.开发平台：Intellij IDEA

4.数据库： mysql

5.操作系统: 64位windows10操作系统环境。

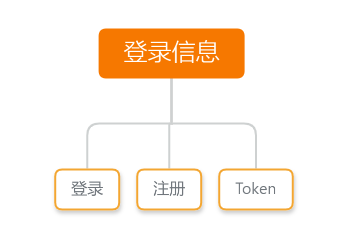
#### 3.2.2数据库设计

在该系统中，需要对账号信息，用户信息，视频信息，收藏信息，点赞信息进行管理。所以需要为这些信息创建相应的数据库表。

本系统一共设计规划出5个实体类，分别是账号信息，用户信息，视频信息，收藏信息，点赞信息，且通过在模型中建立模型间的关联，实现建立表与表之间的关联，以降低数据库的冗余。

1、账号信息实体类

账号信息实体类用来保存用户的账号，密码，token三个属性。账号信息实体类如图3.2.2.1所示：



3.2.2.1账号信息实体类

2、个人信息实体类

个人信息实体类用来保存用户的个人作品、收藏、关注的信息。用户信息实体类如图3.2.2.2所示：



图3.2.2.2个人信息实体类

3、视频详情实体类

视频详情实体类有点赞、收藏、评论、关注四个属性。视频信息实体类如图3.2.2.3所示：

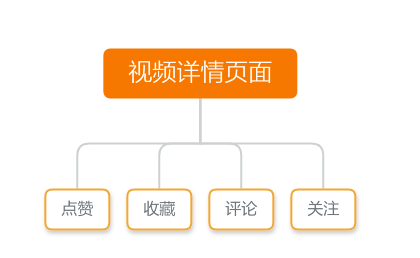


图3.2.2.3视频信息实体类

4、个人中心类

个人中心类有关注、个人信息、设置、充值、个人ID、个人头像、收藏视频、历史记录、已上传视频、上传头像、修改个人资料、粉丝数、消息十三类如图3.2.2.4所示：

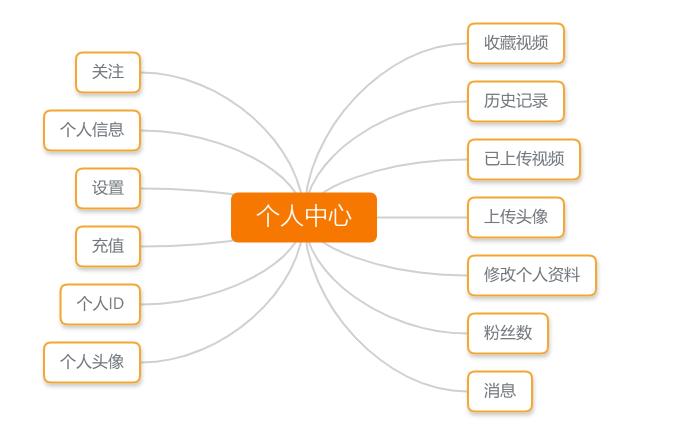


图3.2.2.4收藏信息实体类

5、视频列表类

视频列表有热门视频播放、下拉刷新、上拉刷新三个类。视频列表类如图3.2.2.5所示：

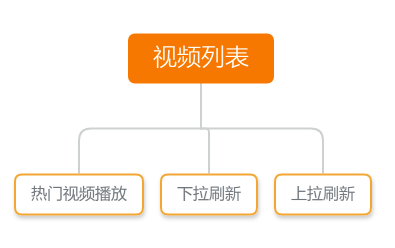


图3.2.2.5点赞信息实体类

#### 3.2.3接口文档

1. 用户注册接口：如图3.2.3.1所示。

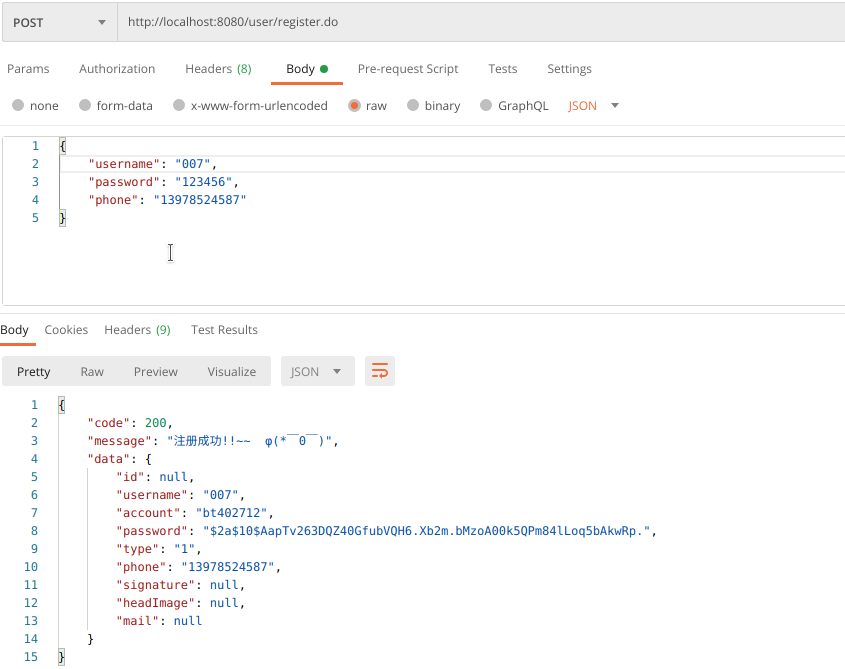


图3.2.3.1接口文档

1. 注册失败接口：如图3.2.3.2所示。

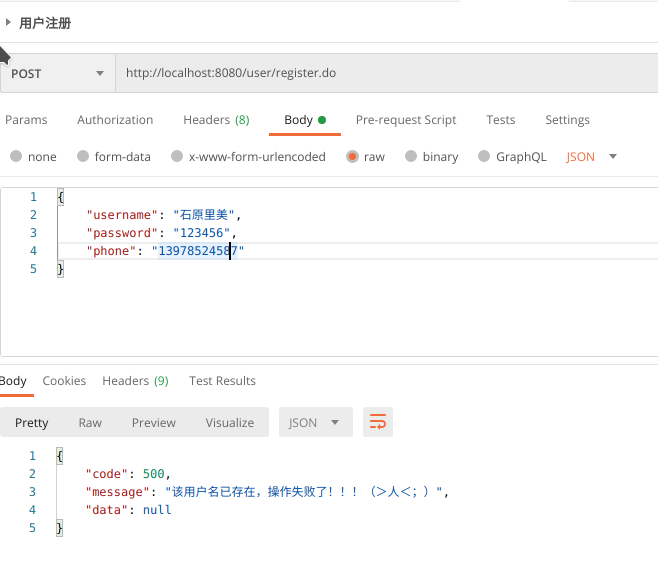


图3.2.3.2注册失败接口

1. 登录成功接口：如图3.2.3.3所示。

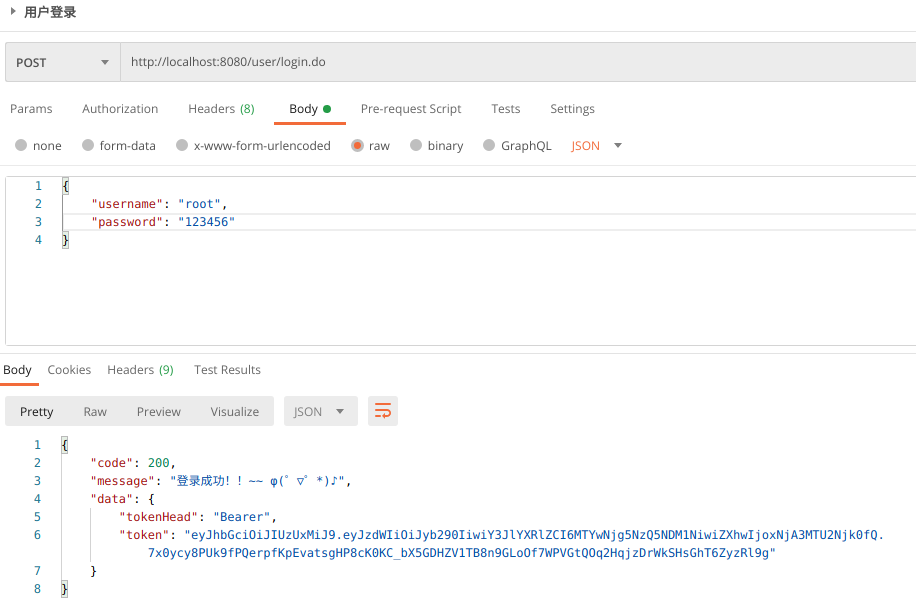


图3.2.3.3登陆成功

1. 登录失败接口：如图3.2.3.4所示。

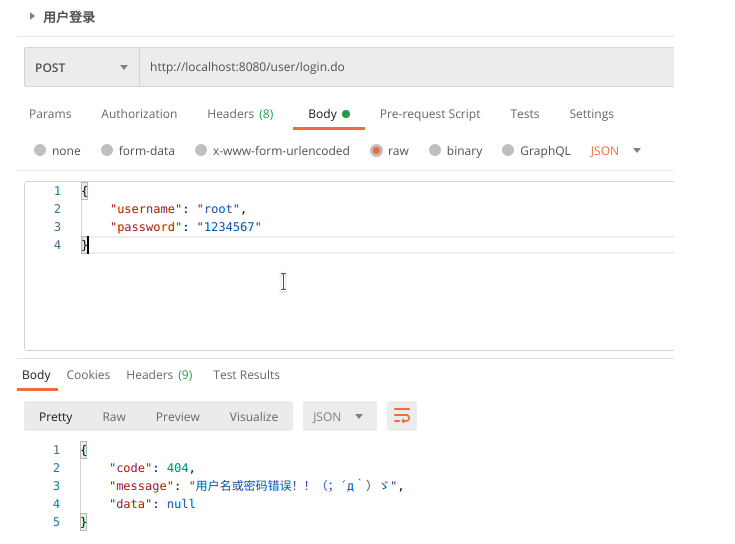


图3.2.3.4登录失败接口

1. 查询视频列表接口：如图3.2.3.5所示。

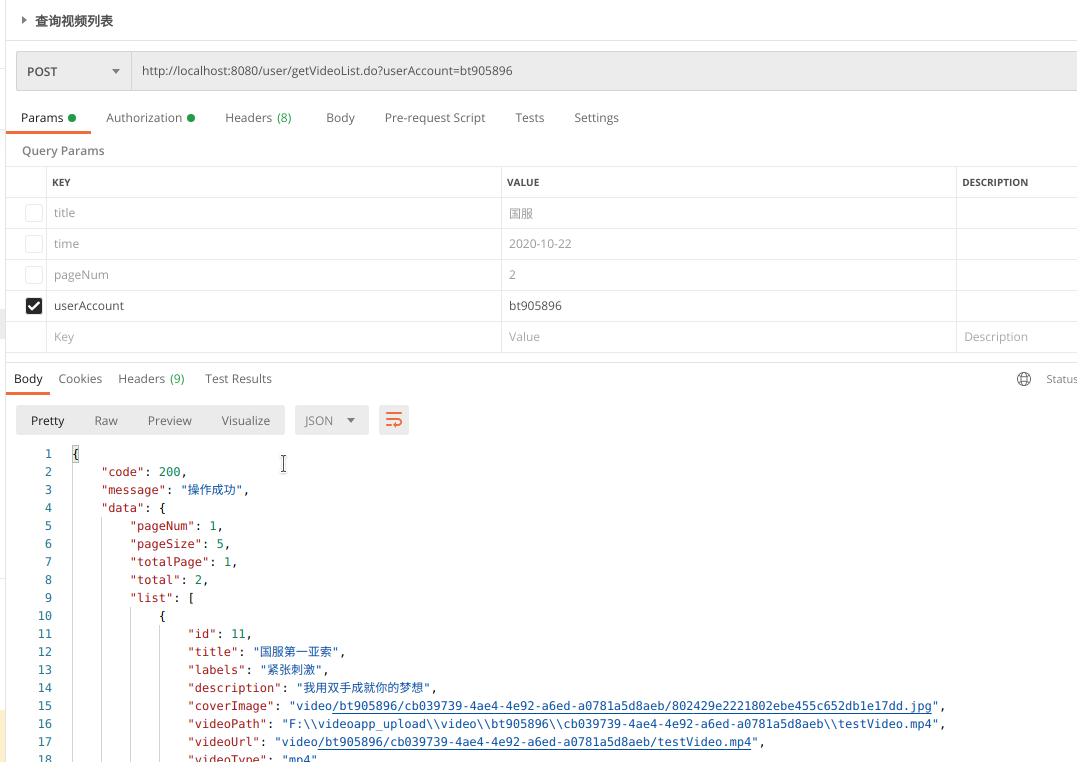


图3.2.3.5

1. 查询用户资料：如图3.2.3.6所示。

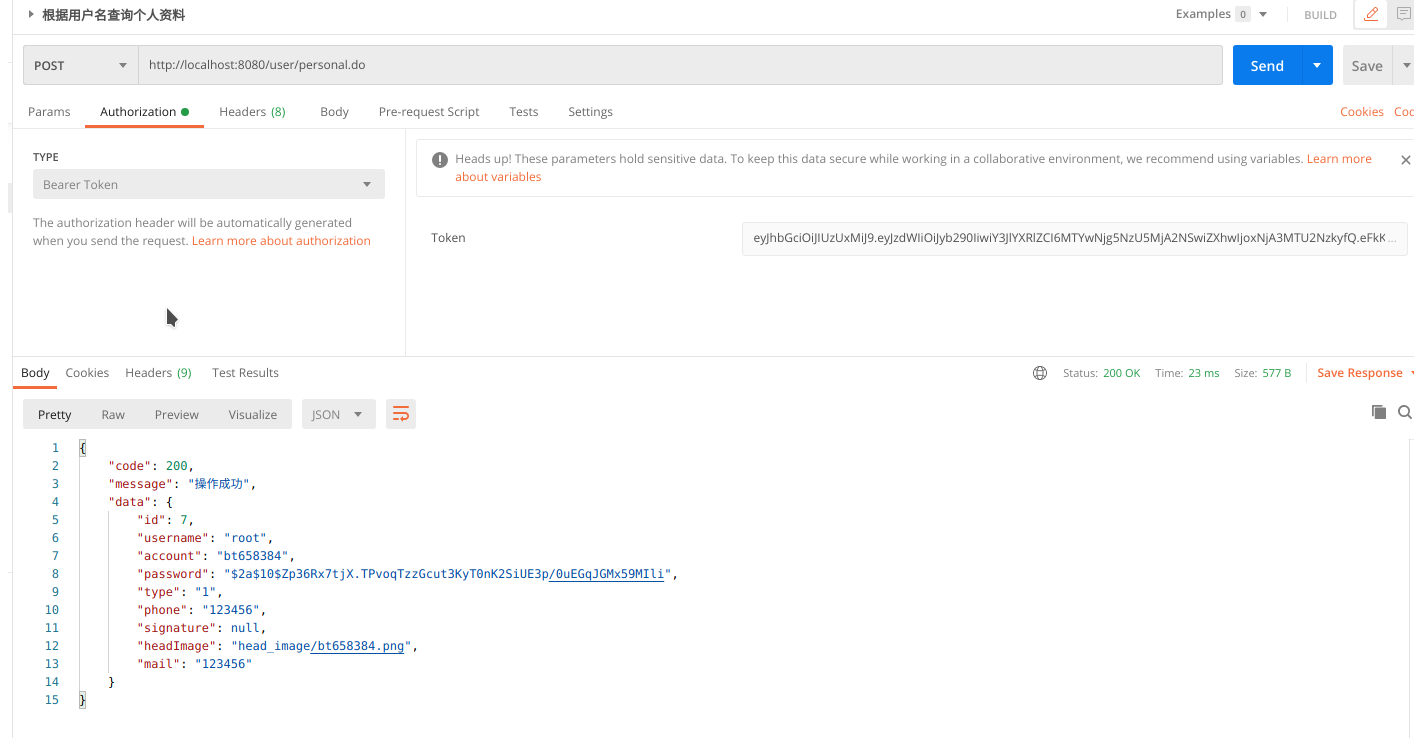


图3.2.3.6查询用户资料

1. 当前视频评论接口：如图3.2.3.7所示。

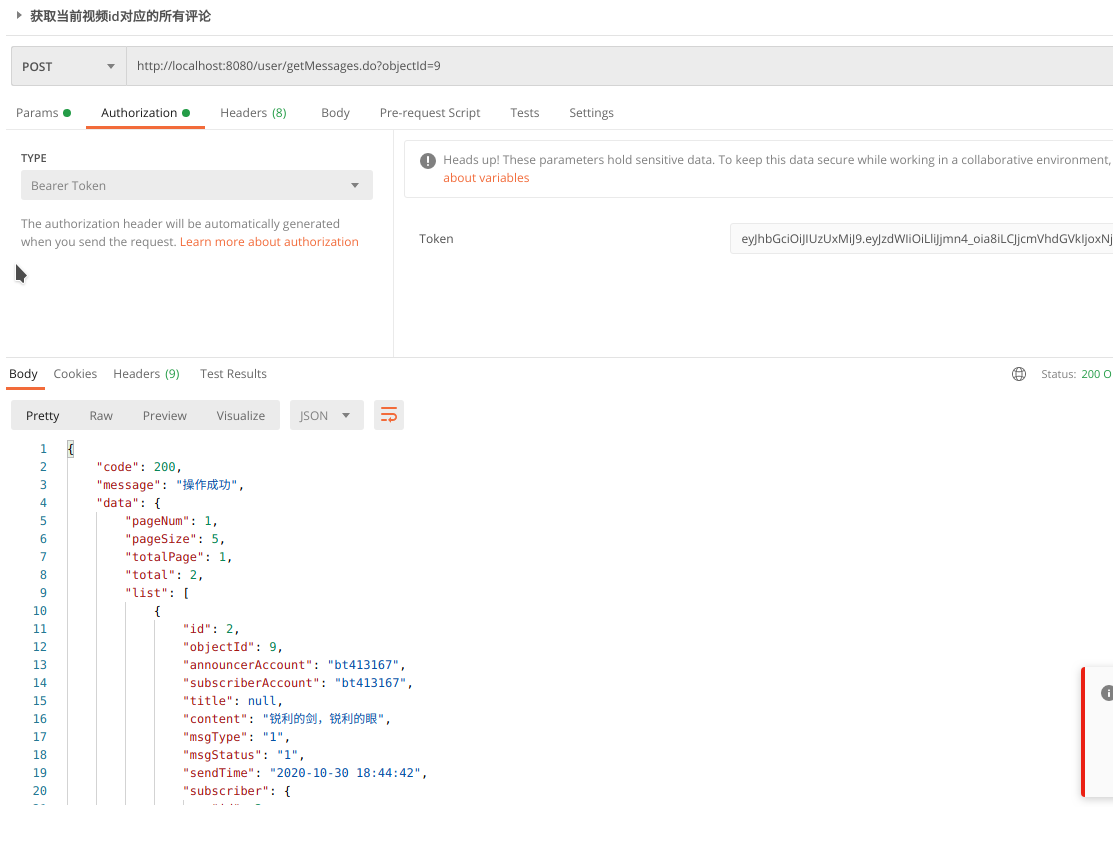


图3.2.3.7当前视频评论接口

1. 点赞视频接口：如图3.2.3.8所示。

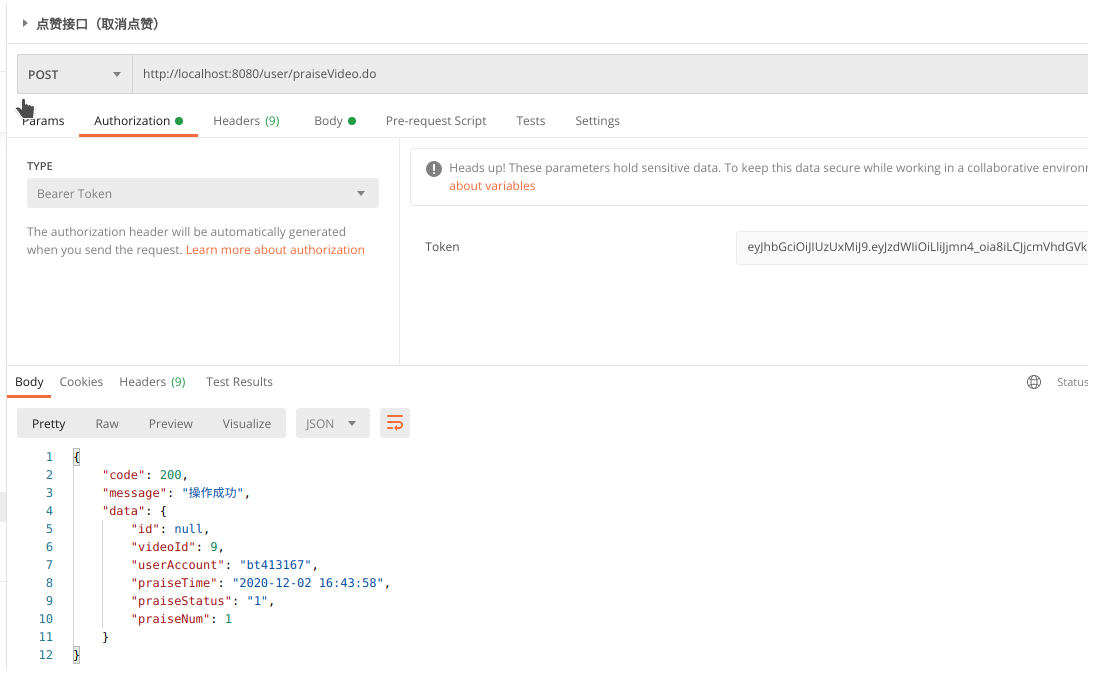


图3.2.3.8点赞视频

1. 上传视频接口：如图3.2.3.9所示。

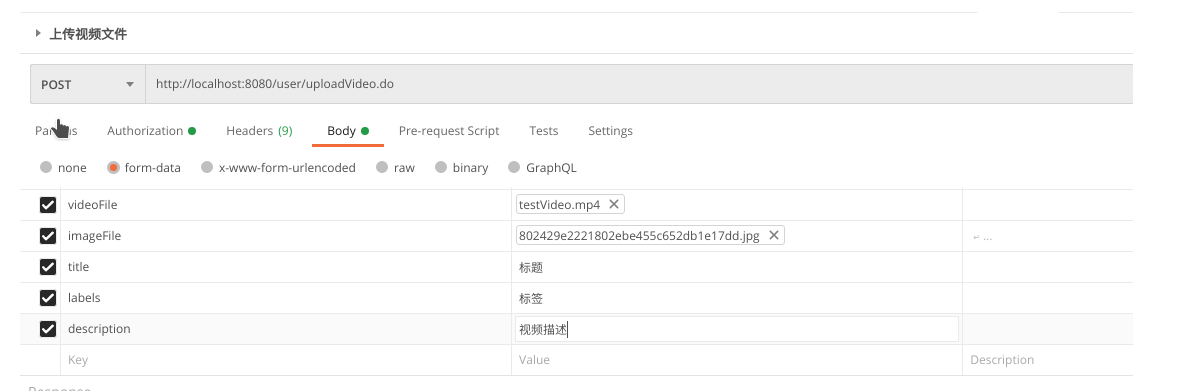


图3.2.3.9上传视频接口

1. 上传头像接口：如图3.2.3.10所示。

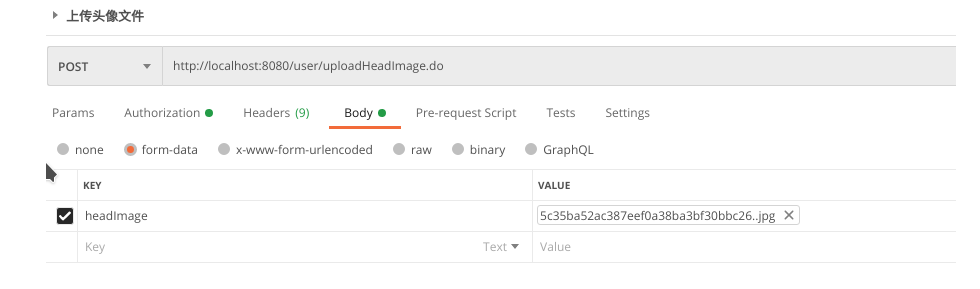


图3.2.3.10上传头像接口

1. 身份过期接口：如图3.2.3.11所示。

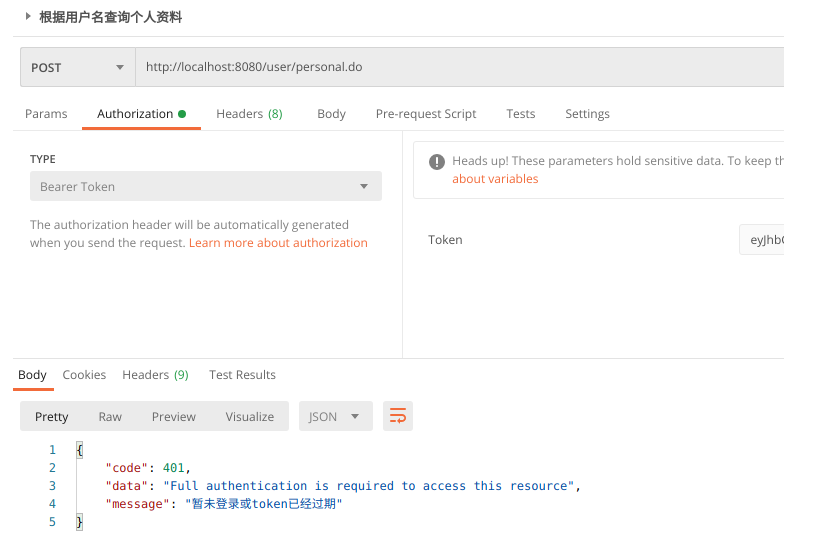


图3.2.3.11身份过期接口

1. 视频详情页接口：如图3.2.3.12所示。

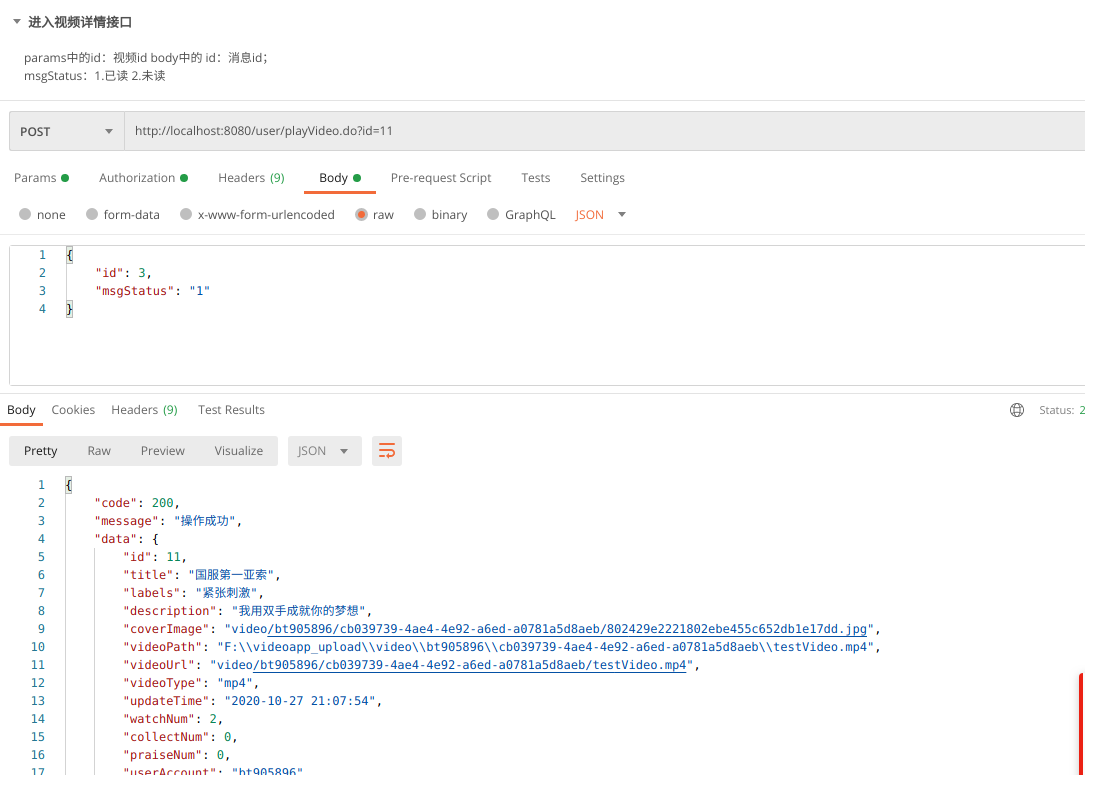


图3.2.3.12视频详情页接口

1. 收藏视频接口：如图3.2.3.13所示。

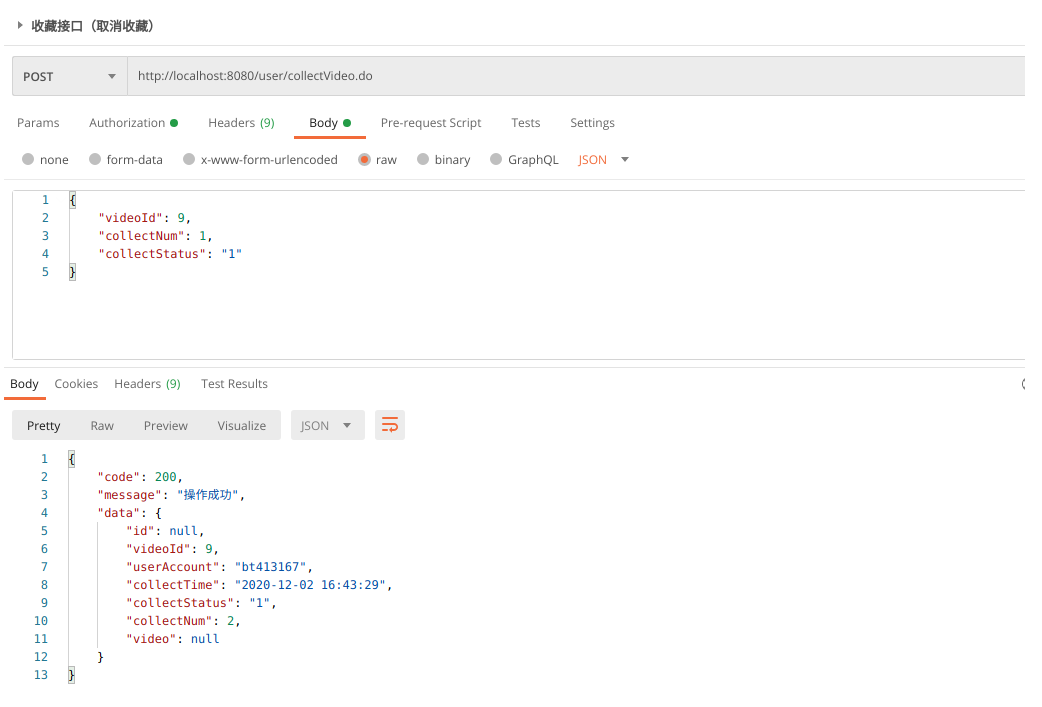


图3.2.3.13收藏视频接口

1. 获取消息接口：如图3.2.3.14所示。

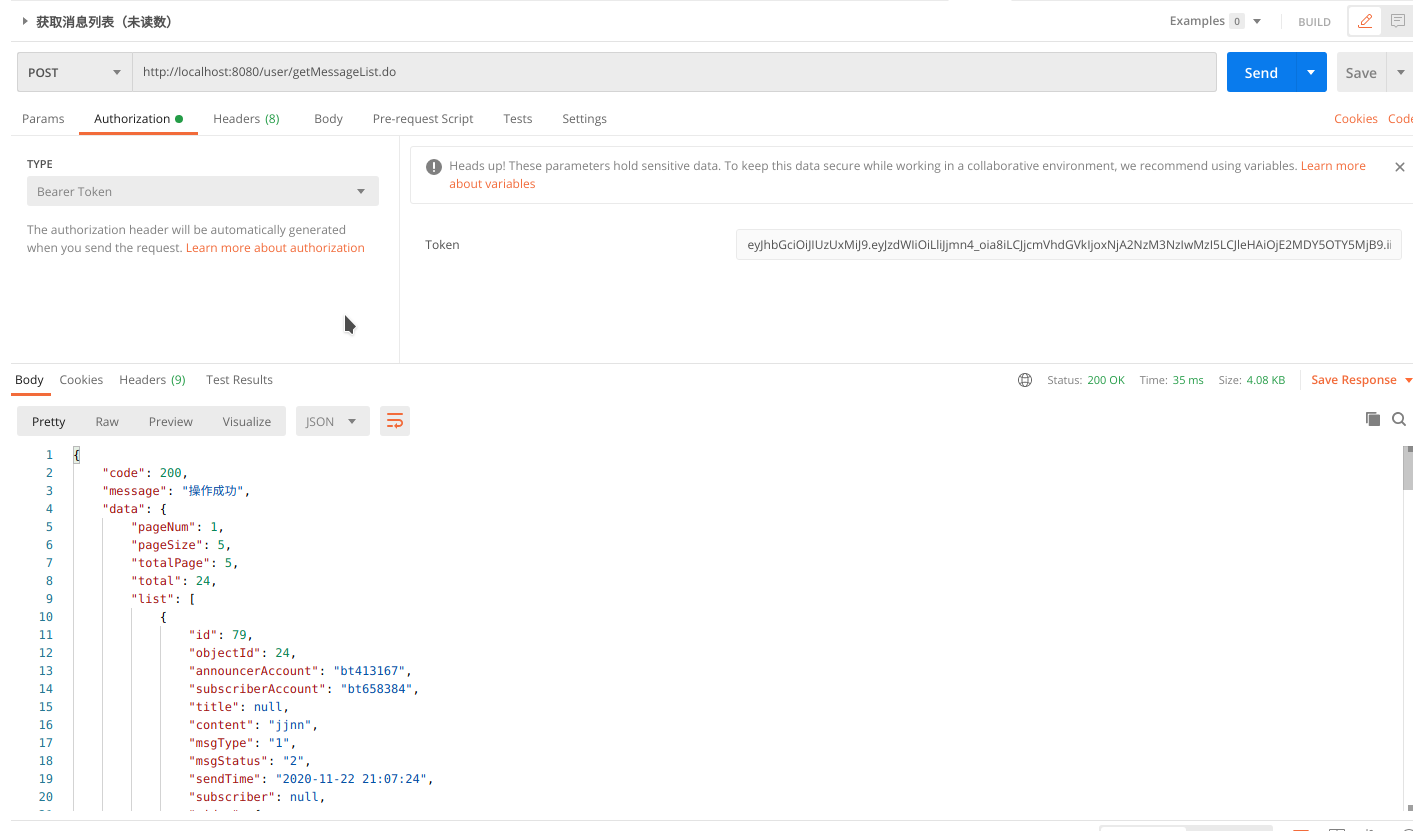


图3.2.3.14获取消息接口

1. 修改个人资料接口：图3.2.3.15所示。

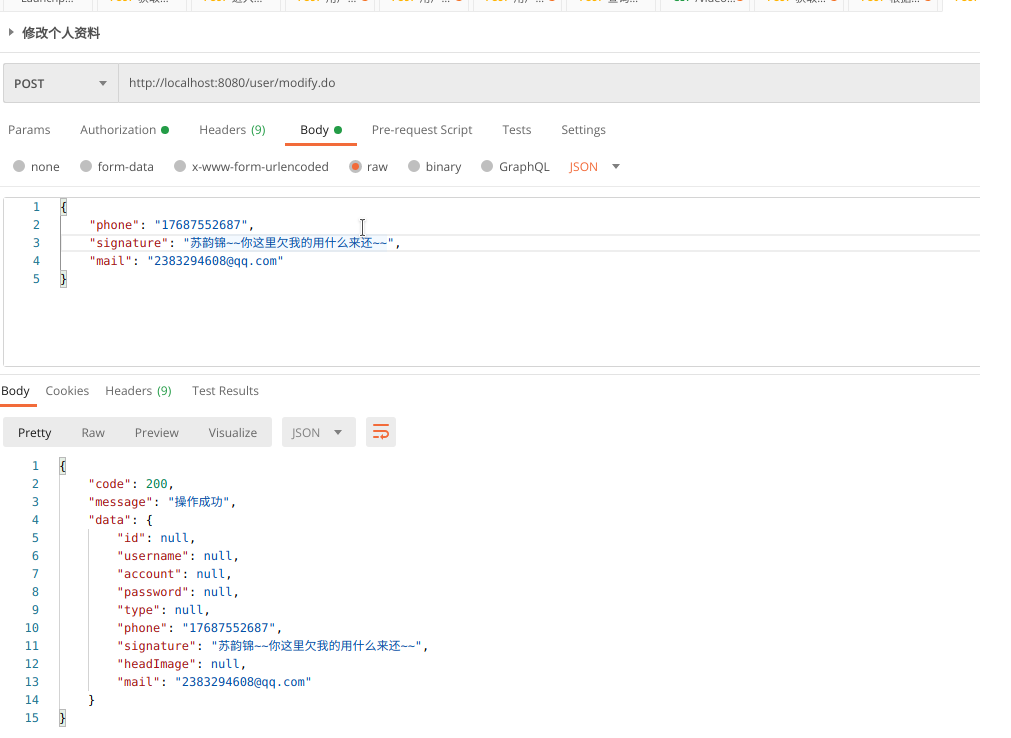


图3.2.3.15修改个人资料接口

16、用户发送评论接口：如图3.2.3.16所示。

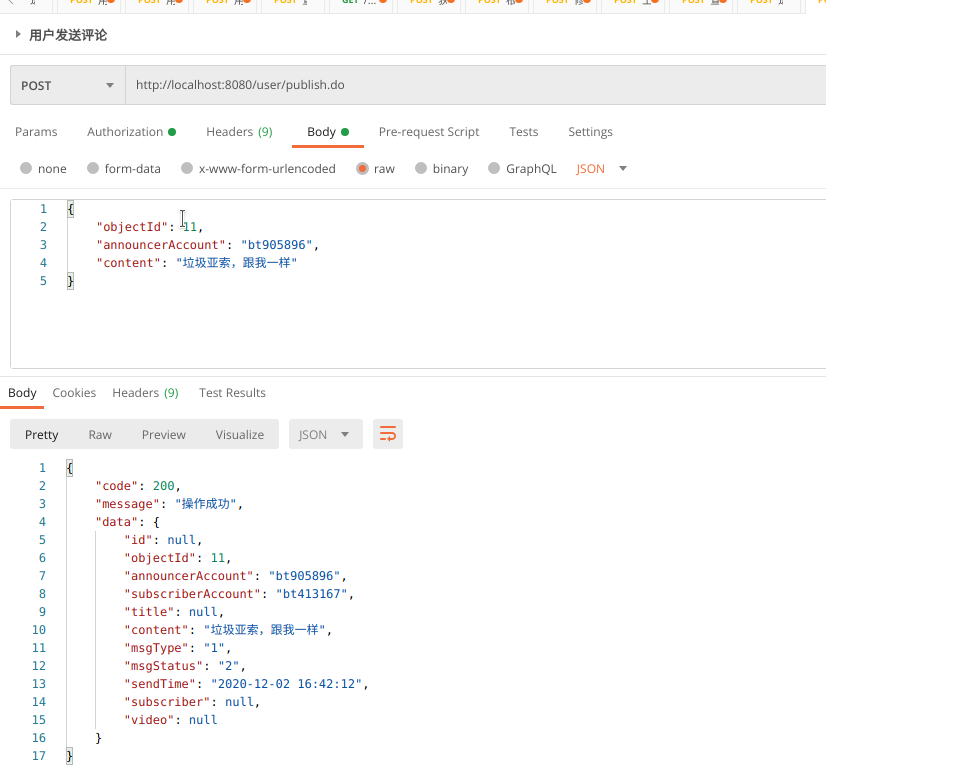


图3.2.3.16 用户发送评论接口

17、用户藏列表接口：如图3.2.3.17所示。

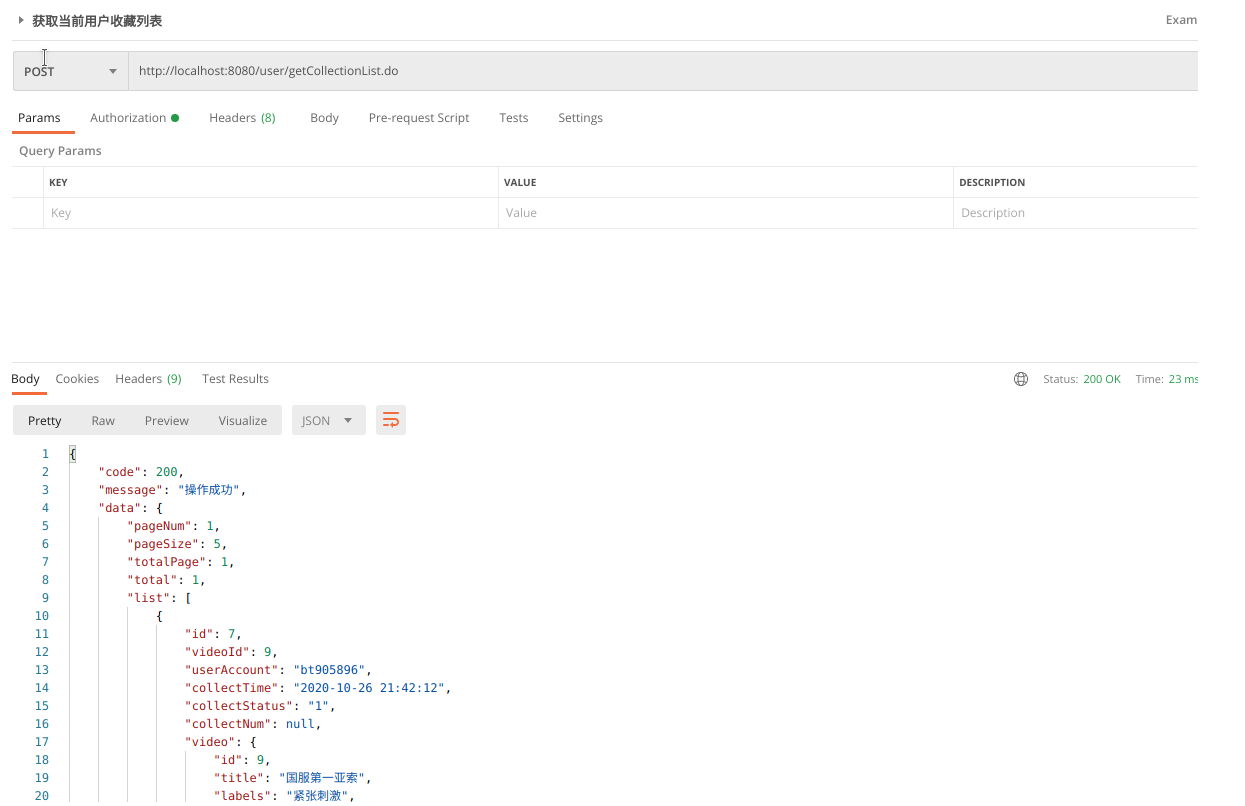


图3.2.3.17 用户收藏列表接口

**5．参考文献：**

1. 李刚. 疯狂Android讲义（第三版）[M]. 北京: 电子工业出版社, 2015.
2. 郭霖. 第一行代码（第二版）[M]. 北京:人民邮电出版社,2016.Asdasd
3. 郭志宏. Android应用开发详解[M].电子工业出版社. 2010.
4. 李安渝. Web Services技术与实现[M]. 北京：国防工业出版社，2003

# 4．心得体会

## 4.1综合学习心得

本次实验是一次综合性的实验，讲究团队合作，每个团队都有自己的分工。黄振倚负责前端的代码编写以及报告的编写、黄中翔负责后端的处理、黄锦泰负责接口的编写，每个人都分工有序。经过了前期的学习，已经初步了解了Android是如何实现的，再通过后期的学习，以及网上的查阅资料，我们确定了这个视频类型的题目。

因为有人Android学习了比较久，所以我们都听他的分工，黄振倚前端是模仿网上大部分安卓视频设计的，图标是从阿里库拿的。后端是找Android的课本以及网上的开源代码来实现的。接口是自己设计。前端的交互确实不是容易实现，对于每一个点击的功能，都要设计一个对应的实现代码。对于图片的展示格式，也要设置对应的代码。

这次的经历使我们对系统有了一个更深层次的理解，从应用端到服务器再到底层的实现都深深的刻在脑海中，将会对今后的学习和项目的开发有着巨大的帮助。本次的实验收获的并不仅仅是一个项目完成的成就感，更有着团队协作，为完成一个共同的目标，一起解决问题，共同进步。

**4.2存在问题及分析**

**< -- 需要加内容的可以加内容-- >**

1. 由于安卓版本的不断更新，安卓手机在安全性方面的不断加强，从6.0开始就有了动态权限申请。由于本次实验使用的是Android 10 版本的手机进行调试。所以出现了一个尽管已经对相应的权限进行动态申请仍然不能对外部文件读取的问题。这个问题当时查了好久。在各大博客找寻对应的接口都未能找到。期间也尝试过使用第三方库来解决，甚至使用了先截取视频保存到手机内部路径的方法，但是由于效果不够好，还是没有采用。最后在官方找到了对应的代码，才得以解决问题。

1. 在与服务器对接时，由于在服务器某些文件夹存在着读取权限的问题，导致最后我们发送的视频无法上传成功，期间也是经过了很多次的测试才发现这个问题，在上传视频过程中发生无法读写文件夹的错误，后来我们修改了uploads上传文件夹的读写权限后解决了此问题。
2. 在完成个人资料修改功能时，我们碰见了更新头像失败，但是其他资料可以更新成功的问题，在经过多次对接口测试后，我们发现了后端存在一个默认头像覆盖更新头像的Bug，我们通过修改模型的获取器，解决了这个问题。

5.**项目分工**

黄振倚：负责前端代码的编写以及文档编写。

黄锦泰：负责接口代码的实现以及文档编写。

黄中翔：负责后端代码的实现以及文档的编写。

**< -- 附录要改-- >**

# 附录

代码1.1 视频上传

|  |
| --- |
| final AlertDialog dialog= show\_dialog();  final Handler handler = new Handler() {  //在主线程中处理从子线程发送过来的消息  public void handleMessage(Message msg) {  //获得消息中携带的数据  String data = (String) msg.obj;  dialog.cancel();  //使用吐司对话框提示用户登录成功还是登录失败  Toast.makeText(PicChooseActivity.this, data, Toast.LENGTH\_SHORT).show();  }  };  new Thread(new Runnable() {  @Override  public void run() {  int success=0;  Message msg;  client=new OkHttpClient();  MultipartBody.Builder requestBody = new MultipartBody.Builder().setType(MultipartBody.FORM);  requestBody.addFormDataPart("token",MainActivity.token)  .addFormDataPart("title",title);  if(f != null){  // MediaType.parse() 里面是上传的文件类型。  RequestBody body = RequestBody.create(MediaType.parse("image/\*"), f);  String filename = f.getName();  // 参数分别为， 请求key ，文件名称 ， RequestBody  requestBody.addFormDataPart("file", filename, body);  }else{  RequestBody body = RequestBody.create(MediaType.parse("image/\*"), f);  // 参数分别为， 请求key ，文件名称 ， RequestBody  requestBody.addFormDataPart("file", null, body);  }  Request request= new Request.Builder()  .url("http://119.29.24.11/index/Pictures/sharePicture")  .post(requestBody.build())  .build();  try {  Response response=client.newCall(request).execute();  String responsedata = response.body().string();  JSONObject jsonObject = new JSONObject(responsedata);  success=jsonObject .optInt("code");  } catch (Exception e) {  e.printStackTrace();  }  if(success==200) {  //视频分享成功  msg = handler.obtainMessage();  //设置消息对象携带的数据  msg.obj = "视频分享成功";  handler.sendMessage(msg);  finish();  }else{  //视频分享失败  msg = handler.obtainMessage();  //设置消息对象携带的数据  msg.obj = "视频分享失败";  handler.sendMessage(msg);  }  }  }).start(); |

代码1.2 初始化主界面（所有用户分享视频）

|  |
| --- |
| private void init(){  //final AlertDialog dialog= show\_dialog();  final Handler handler = new Handler() {  //在主线程中处理从子线程发送过来的消息  public void handleMessage(Message msg) {  //获得消息中携带的数据  String data = (String) msg.obj;  // dialog.cancel();  homePageAdapter.notifyDataSetChanged();  if(swipe.isRefreshing()){  swipe.setRefreshing(false);  }  // Toast.makeText(getActivity(), data, Toast.LENGTH\_SHORT).show();  }  };  new Thread(new Runnable() {  @Override  public void run() {  int success=0;  Message msg;  client=new OkHttpClient();  MultipartBody.Builder requestBody =  new MultipartBody.Builder().setType(MultipartBody.FORM);  requestBody.addFormDataPart("token",MainActivity.token);  Request request= new Request.Builder()  .url("http://119.29.24.11/index/index/dataList")  .post(requestBody.build())  .build();  try {  Response response=client.newCall(request).execute();  String responsedata = response.body().string();  JSONObject jsonObject = new JSONObject(responsedata);  success=jsonObject.optInt("code");  JSONArray jsonArray = jsonObject.optJSONArray("data");  if(jsonArray.length()>0)  success=200;  parseJson(jsonArray);  } catch (Exception e) {  e.printStackTrace();  }  if(success==200) {  //刷新数据成功  msg = handler.obtainMessage();  //设置消息对象携带的数据  msg.obj = "刷新成功";  handler.sendMessage(msg);  }else{  //刷新数据失败  msg = handler.obtainMessage();  //设置消息对象携带的数据  msg.obj = "刷新失败 "+success;  handler.sendMessage(msg);  }  }  }).start();  public void parseJson(JSONArray jsonArray)  {  homeList.clear();  try {  for(int i=0;i<jsonArray.length();i++) {  JSONObject jsonObject = jsonArray.getJSONObject(i);  Issue issue = new Issue();  String time=format.format(gc.getTime());  issue.setId(jsonObject.optString("id"));  issue.setLike(jsonObject.optString("praise"));  issue.setNickName(jsonObject.optString("nickname"));  issue.setLove(jsonObject.optString("collection"));  issue.setTitle(jsonObject.optString( "title"));  issue.setHot(jsonObject.optString("heat"));  issue.setImage(Uri.parse(jsonObject.optString("img")));  issue.setHeadPic(Uri.parse(jsonObject.optString("userImg")));  homeList.add(issue);  }  } catch (Exception e) {  e.printStackTrace();  }  } |

代码1.3修改状态栏颜色

|  |
| --- |
| public static final void changeStatusBarTextImgColor( boolean isBlack, Activity activity) {  if (isBlack) {  //设置状态栏黑色字体  activity.getWindow().getDecorView().  setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR);  } else {  //恢复状态栏白色字体  activity.getWindow().getDecorView().  setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_VISIBLE);  }  } |

代码1.4 视频下载

|  |
| --- |
| private void downloadPic(){  final AlertDialog dialog= show\_dialog();  final Handler handler = new Handler() {  //在主线程中处理从子线程发送过来的消息  public void handleMessage(Message msg) {  //获得消息中携带的数据  String data = (String) msg.obj;  dialog.cancel();  //使用吐司对话框提示用户登录成功还是登录失败  Toast.makeText(PicDetailActivity.this, data, Toast.LENGTH\_SHORT).show();  }  };  new Thread(new Runnable() {  @Override  public void run() {  Message msg;  client=new OkHttpClient();  MultipartBody.Builder requestBody =  new MultipartBody.Builder().setType(MultipartBody.FORM);  requestBody.addFormDataPart("token",MainActivity.token)  .addFormDataPart("id",Id);  Request request= new Request.Builder()  .url(issue.getImage().toString())  .build();  try {  Response response=client.newCall(request).execute();  byte[] bytes = response.body().bytes();  bytesToImageFile(bytes);  } catch (Exception e) {  e.printStackTrace();  Log.d("run: ", "run: "+e.getMessage());  }  msg = handler.obtainMessage();  //设置消息对象携带的数据  msg.obj = "视频已经下载到 "+file\_path+" 目录下";  handler.sendMessage(msg);  }  }).start();  }  private void bytesToImageFile(byte[] bytes) {  try {  String sdCardDir = Environment.getExternalStorageDirectory() + "/DCIM/";  File appDir = new File(sdCardDir, "myImage");  if (!appDir.exists()) {  appDir.mkdir();  }  String str = issue.getImage().toString();  String fileName = str.substring(str.lastIndexOf("/")+1);  file\_path=appDir.getPath();  File f = new File(appDir, fileName);  FileOutputStream fos = new FileOutputStream(f);  fos.write(bytes, 0, bytes.length);  fos.flush();  fos.close();  Intent intent = new Intent(Intent.ACTION\_MEDIA\_SCANNER\_SCAN\_FILE);  Uri uri = Uri.fromFile(f);  intent.setData(uri);  PicDetailActivity.this.sendBroadcast(intent);  //这个广播的目的就是更新图库，发了这个广播进入相册就可以找到保存的视频了  } catch (Exception e) {  e.printStackTrace();  }  } |