目录

[1 概述 1](#_Toc9427478)

[1-1 应用介绍 1](#_Toc9427479)

[1-2 应用模块 1](#_Toc9427480)

[2 数据库设计 2](#_Toc9427481)

[2-1 建立数据库 2](#_Toc9427482)

[2-2 设计数据表 2](#_Toc9427483)

[3 搭建Web服务 3](#_Toc9427484)

[3-1 第三方工具包 3](#_Toc9427485)

[3-2 创建工具包 4](#_Toc9427486)

[3-3 Servlet 7](#_Toc9427487)

[3-4 web.xml配置 73](#_Toc9427488)

[4 创建Android应用 77](#_Toc9427489)

[4-1 环境配置 77](#_Toc9427490)

[4-2 引入资源 79](#_Toc9427491)

[4-3 页面设计 79](#_Toc9427492)

[4-4 对象类 132](#_Toc9427493)

[4-5 工具类 135](#_Toc9427494)

[4-6 页面逻辑处理类 136](#_Toc9427495)

[4-7 项目配置 225](#_Toc9427496)

[5 应用测试 227](#_Toc9427497)

[5-1 “启动”页 227](#_Toc9427498)

[5-2 “主页” 227](#_Toc9427499)

[5-3 “发现”页 228](#_Toc9427500)

[5-4 “详情”页 229](#_Toc9427501)

[5-5 “查找”页 229](#_Toc9427502)

[5-6 “发布”页 230](#_Toc9427503)

[5-7 “我的”页 231](#_Toc9427504)

[6 资源链接 235](#_Toc9427505)

1. 概述

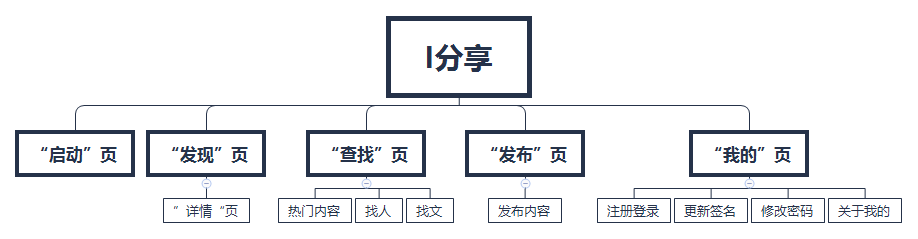
应用介绍

这是一个Android App，可以称之为“i分享”即“爱”分享之意，该应用的开发就是为了给用户提供一个分享交流的地方，在这里用户可以自由的分享、查找、评论、收藏有趣的内容。通过开发这个应用，让我们更加熟悉地掌握和应用Android相关知识。

APP的客户端程序是在Android Studio上开发和运行，服务器端程序是在myeclipse上编写并运行在Tomcat v8.5上。所以，在开发前还需要在计算机上安装Android Studio、myeclipse、mysql以及数据库管理软件——在这里我用的是navicat。CSDN博客上有整个项目的详细开发日志，CSDN地址以及在项目用到的图片、第三方类包等资源在文末会提供链接。

应用模块

“i分享”的主要模块如下



“启动”页

该页面设计是LOGO加上Slogan，作为跳转到应用的一个过渡页。在展示上设计上为淡入效果，逻辑上是判断用户是否开启网络权限，开启则进入，未开启则退出应用。

“发现”页

该页面作为打开APP后看到的第一个页面，主要是以卡片流的形式展示内容表中的内容。点击页面的卡片可以进入到详情页，在详情页可以看到内容的所有信息并可以进行点赞、收藏，而且若是作者打开详情还可以进行删除删除操作。

“查找”页

进入该页面时展示的是按点赞数排行前10的内容，点击列表同样是进入详情页面。在页面顶部是搜索框，可以输入内容标题或简述的关键字找文，也可以输入昵称关键字找人，返回的内容都是以列表形式展现，点击列表则进入详情。

“发布”页

进入发布页，有四个Tab对应可以选择四种发布类型——分享趣事、分享图片、分享音乐、发布日记。不同类型的内容在页面流的展示和详情页的展示都会不一样，这四种类型文章在数据表中用0、1、2、3存储。

“我的”页

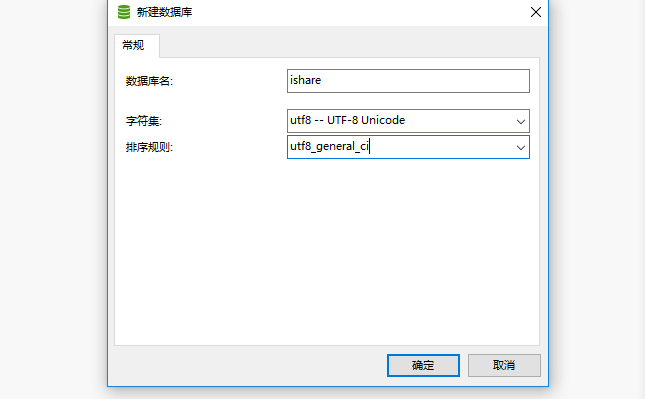
该页面包含四大模块，通过顶部按钮可以弹出注册登录的对话框进行注册和登录操作，另外还可以进入更新签名、修改密码的页面进行更新签名与修改密码，最后就是“关于我的”部分，可以查看我的分享、我的日记、我的收藏以及“关于i分享”的介绍。

1. 数据库设计

建立数据库

创建数据库“ishare”

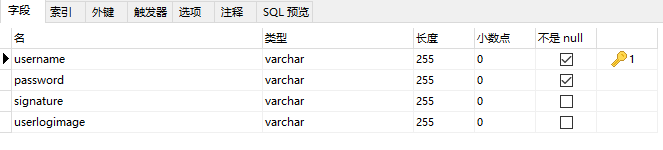
打开navicat新建MYSQL连接，连接到我们计算机的MySQL，在localhost处右击，选择新建数据库，数据库名填写为ishare，字符集选择utf8 -- UTF-8 Unicode，排序规则选择utf8\_general\_ci。



设计数据表

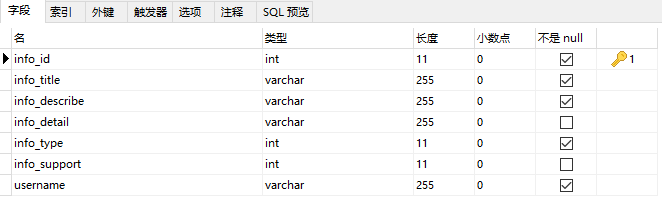
用户表“user”

在建好的数据库ishare下的表上方右击，选择新建表，填入四个字段——username、password、signature、userlogimage，这四个字段分别对应用户名、登录密码、用户签名、用户头像，点击保存，命名为user。在这边没有对字段做太多限制，所以统一设置为varchar类型，字符集设置为utf8，排序规则设置为utf8\_general\_ci。



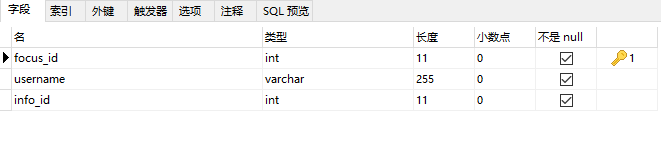
内容表“info”

在建好的数据库ishare下的表上方右击，选择新建表，填入七个字段——info\_id、info\_title、info\_describe、info\_detail、info\_type、info\_support、username，这七个字段分别对应内容ID、内容标题、内容简述、内容详情、内容类型、内容点赞数、内容作者，其中info\_id设置为int类型，且自增。点击保存，命名为info。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci



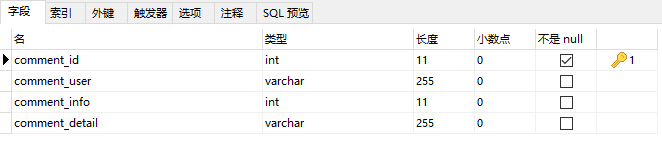
收藏表“focus”

在建好的数据库ishare下的表上方右击，选择新建表，填入三个字段——focus\_id、username、info\_id，这三个字段分别对应收藏ID、收藏用户昵称、内容ID，其中focus\_id设置为int类型，且自增。点击保存，命名为focus。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci



评论表“comment”

在建好的数据库ishare下的表上方右击，选择新建表，填入四个字段——comment\_id、comment\_user、comment\_info、comment\_detail，这四个字段分别对应评论ID、评论用户昵称、评论的内容ID、评论的详情，其中comment\_id设置为int类型，且自增。点击保存，命名为comment。同样，varchar类型的字段，统一将字符集设置为utf8，排序规则设置为utf8\_general\_ci



1. 搭建Web服务

打开myEclipse，新建Workspace，在这我的Wrkspace是在E盘下的myEclipseProject，你也可以直接使用自己已经创建好的Workspace。打开Workspace，在file处右击，选择New下的Web Project，在Project name处填写iShareService，Java EE version选择JavaEE7-Web 3.1，然后点击finish。

第三方工具包

打开刚刚新建的项目iShareService，找到WebRoot>WEB-INF>lib文件夹，将下图中的jar包复制到lib文件夹下，下图的jar包资源可以在文末处的链接找到。引入的jar包可以分为3类，第一个jar包是为了应用rowset，最后一个是连接MySQL数据库，其余6个是封装JSON使用到的相关jar包。



创建工具包

打开iShareService，找到src目录，右击选择New，点击Package，在Name中填写org.utils，点击finish。

JdbcUtils.java

在创建好的org.utils包中右击选择New，点击Class，在name中填写JdbcUtils，点击finish。打开新建好的JdbcUtils.java文件，将下面代码复制到文件中（注意看文件头部中的url、username、password是否和你数据库的连接地址、用户名、密码一致，不一致则需要修改）。这部分的代码主要就是处理与数据库的连接，因为之后要反复使用到所以封装出来比较方便。

**----------------------------- JdbcUtils.java开始 --------------------------**

package org.utils;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class JdbcUtils {

private static String url ="jdbc:mysql://127.0.0.1/ishare?useUnicode=true&characterEncoding=utf8&allowMultiQueries=true";

private static String username="root";

private static String password="";

private static String driverName="com.mysql.jdbc.Driver";

// static静态代码块里面代码，会在类加载到内存的时候立刻执行

static {

//1.注册驱动

try {

Class.forName(driverName);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

/\*\*

\* 获取连接

\* @return

\* @throws SQLException

\*/

public static Connection getConnection() throws SQLException {

//2.获取连接

Connection connection = DriverManager.getConnection(url, username, password);

return connection;

}

public static void close(Connection connection,PreparedStatement prepareStatement,ResultSet resultSet) {

//5.释放资源 connection prepareStatement resultSet

if(resultSet!=null) {

try {

resultSet.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(prepareStatement!=null) {

try {

prepareStatement.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(connection!=null) {

try {

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void statementClose(Connection connection,Statement statement,ResultSet resultSet) {

//5.释放资源 connection prepareStatement resultSet

if(resultSet!=null) {

try {

resultSet.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(statement!=null) {

try {

statement.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(connection!=null) {

try {

connection.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

**----------------------------- JdbcUtils.java 结束----------------------------**

JsonUtils.java

在创建好的org.utils包中右击选择New，点击Class，在name中填写JsonUtils，点击finish。打开新建好的JsonUtils.java文件，将下面代码复制到文件中。这部分的代码就是将List类型的数据封装到JSONArray，在向客户端发送返回信息时会反复用到，所以封装出来。

**-----------------------------JsonUtils.java开始 ----------------------------**

package org.utils;

import net.sf.json.JSONArray;

import net.sf.json.JSONObject;

import java.util.List;

import java.util.Map;

public class JsonUtils {

public static JSONArray formatRsToJsonArray(List<Map<String,Object>> data){

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

for(Map<String,Object> rowItem:data) {

JSONObject json = new JSONObject();

for(Map.Entry<String, Object> entry:rowItem.entrySet()) {

json.put(entry.getKey(), entry.getValue());

}

jsonArray.add(json);

}

return jsonArray;

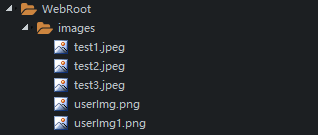
}

}

**-------------------------------- JsonUtils.java结束 --------------------------**

Servlet

打开iShareService，找到src目录，右击选择New，点击Package，在Name中填写myServlet.data，点击finish。打开iShareService> WebRoot>images文件夹，将文末链接资源中的图片复制到images文件夹下。



Register.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写Register，点击finish。打开新建好的Register.java文件，将下面代码复制到文件中。这部分代码主要就是获取到从客户端传过来的参数——用户名、密码、签名，然后设置用户默认头像为userImg.png，再把这些字段插入到表user中，完成用户注册。

**----------------------------- Register.java开始 -----------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

public class Register extends HttpServlet{

/\*\*

\* 注册

\*/

private static final long serialVersionUID = 1L;

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String username =request.getParameter("username");

String password = request.getParameter("password");

String signature =request.getParameter("signature");

String userlogimage = "userImg.png"; /\* 默认头像\*/

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="INSERT INTO user VALUES(?,?,?,?)";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, username);

prepareStatement.setString(2, password);

prepareStatement.setString(3, signature);

prepareStatement.setString(4, userlogimage);

//4.执行sql

prepareStatement.execute();

out.print("注册成功!");

} catch (Exception e) {

e.printStackTrace();

out.print("该用户名已被使用，请更换名字！"+e);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, null);

}

out.flush();

out.close();

}

public void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doGet(request, response);

}

}

**---------------------------- Register.java结束 -----------------------------**

Login.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写Login，点击finish。打开新建好的Login.java文件，将下面代码复制到文件中。这部分代码主要就是获取到从客户端传过来的参数——用户名、密码，然后查询用户表user中是否存在该用户和对应的密码，存在则登录成功，返回查询到的用户昵称、签名、用户头像。

**------------------------------Login.java开始-- -----------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class Login

\*/

@WebServlet("/Login")

public class Login extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

super.init(config);

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String username = request.getParameter("username").trim();

String password = request.getParameter("password").trim();

String signature = "在这里写下你的心情吧！";

String userHeadImg = "";

if(username == null) {

username = "";

}

if(password == null) {

password = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

String backnews="";

boolean boo = false;

boo = username.length()>0&&password.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from user where username=? and password=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, username);

prepareStatement.setString(2, password);

if(boo) {

//4.执行sql

boolean m =false;

rs = prepareStatement.executeQuery();

while(rs.next()) {

signature = rs.getString(3);

userHeadImg = rs.getString(4);

m = true;

}

if(m==true) {

//登录成功

success(request,response,username,userHeadImg,signature,backnews);

}else {

backnews="你输入的用户名或密码有误！";

fail(request,response,username,backnews);

}

}else {

backnews="请输入用户名和密码";

fail(request,response,username,backnews);

}

} catch (Exception e) {

backnews=""+e.toString();

e.printStackTrace();

fail(request,response,username,backnews);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void success(HttpServletRequest request, HttpServletResponse response,String userName,String userLogImage,String signature,String backnews) {

try {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

PrintWriter out = response.getWriter();

Map<String,Object> map = new HashMap<String,Object>();

map.put("userName", userName);

map.put("signature", signature);

map.put("userLogImage", userLogImage);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}catch(IOException e) {

e.printStackTrace();

}

}

public void fail(HttpServletRequest request, HttpServletResponse response,String logname,String backnews) {

response.setContentType("text/html;charset=utf-8");

try {

PrintWriter out = response.getWriter();

out.println(backnews);

}catch(IOException e) {

e.printStackTrace();

}

}

}

**----------------------------- Login.java结束-- ----------------------------**

QueryDiscover.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryDiscover，点击finish。打开新建好的QueryDiscover.java文件，将下面代码复制到文件中。这部分代码主要就是查询内容表info，将查询内容封装到Json中返回给客户端。

**----------------------------QueryDiscover.java开始-- -------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

/\*import java.sql.ResultSetMetaData;\*/

import java.sql.Statement;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

import java.util.List;

import java.util.Map;

import java.util.HashMap;

import java.util.ArrayList;

/\*\*

\* Servlet implementation class QueryDiscover

\*/

@WebServlet("/QueryDiscover")

public class QueryDiscover extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize; //每页加载数量

int pageNum; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryDiscover() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String page = request.getParameter("page");

String count = request.getParameter("count");

if(page == null||page == "") {

page = "1";

}

if(count == null|count == "") {

count = "10";

}

try {

pageNum = Integer.parseInt(page); //第几页

pageSize = Integer.parseInt(count); //每页加载几条

} catch (NumberFormatException e) {

e.printStackTrace();

}

String condition = "select \* from info order by info\_id desc";

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

//ResultSetMetaData metaData = rowSet.getMetaData();

//int columnCount = metaData.getColumnCount(); //返回总列数

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=pageSize&&boo;i++){

int infoId = rowSet.getInt(1); //内容ID

String infoTitle = rowSet.getString(2); //内容标题

String infoDescribe = rowSet.getString(3); //内容简述

String infoDetail = rowSet.getString("info\_detail"); //内容详情

String type = rowSet.getString(5); //类型：0表示日记，1表示趣事

String support = rowSet.getString(6); //点赞数

String infoAuthor = rowSet.getString(7); //作者

Map<String,Object> map = new HashMap<String,Object>();

map.put("infoId", infoId);

map.put("infoTitle", infoTitle);

map.put("infoDescribe", infoDescribe);

map.put("infoDetail", infoDetail);

map.put("infoType", type);

map.put("infoSupport", support);

map.put("infoAuthor", infoAuthor);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**------------------------QueryDiscover.java结束-- -------------------------**

QueryInfoById.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryInfoById，点击finish。打开新建好的QueryInfoById.java文件，将下面代码复制到文件中。这部分代码主要就是根据客户端传来的内容Id去查询内容表info中对应的内容相关信息，将查询结果封装到Json中返回给客户端。

**--------------------------** **QueryInfoById.java开始-- --------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryInfoById

\*/

@WebServlet("/QueryInfoById")

public class QueryInfoById extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize=10; //每页加载数量

int pageNum=1; //第几页

int totalRecord; //总记录数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryInfoById() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String infoId = request.getParameter("infoId");

if(infoId == null||infoId.length()==0) {

return;

}

String condition ="select \* from info where info\_id="+infoId; //按id查找

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

try {

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=totalRecord&&boo;i++){

int id = rowSet.getInt(1); //内容ID

String infoTitle = rowSet.getString(2); //内容标题

String infoDescribe = rowSet.getString(3); //内容简述

String infoDetail = rowSet.getString("info\_detail"); //内容详情

String type = rowSet.getString(5); //类型：0表示日记，1表示趣事

String support = rowSet.getString(6); //点赞数

String infoAuthor = rowSet.getString(7); //作者

Map<String,Object> map = new HashMap<String,Object>();

map.put("infoId", id);

map.put("infoTitle", infoTitle);

map.put("infoDescribe", infoDescribe);

map.put("infoDetail", infoDetail);

map.put("infoType", type);

map.put("infoSupport", support);

map.put("infoAuthor", infoAuthor);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------** **QueryInfoById.java结束-- -------------------------**

QueryHotInfo.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryHotInfo，点击finish。打开新建好的QueryHotInfo.java文件，将下面代码复制到文件中。这部分代码主要就是查询内容表info中点赞数排行前十的内容相关信息，并将将查询结果封装到Json中返回给客户端。

**----------------------------** **QueryHotInfo.java开始-- --------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryHotInfo

\*/

@WebServlet("/QueryHotInfo")

public class QueryHotInfo extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize=10; //每页加载数量

int pageNum=1; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryHotInfo() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String condition = "select \* from info order by info\_support desc"; //按点赞数降序，查询热门

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

//ResultSetMetaData metaData = rowSet.getMetaData();

//int columnCount = metaData.getColumnCount(); //返回总列数

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

/\*out.println("全部记录数"+totalRecord); //全部的记录数\*/

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=pageSize&&boo;i++){

int infoId = rowSet.getInt(1); //内容ID

String infoTitle = rowSet.getString(2); //内容标题

String infoDescribe = rowSet.getString(3); //内容简述

String infoDetail = rowSet.getString("info\_detail"); //内容详情

String type = rowSet.getString(5); //类型：0表示日记，1表示趣事

String support = rowSet.getString(6); //点赞数

String infoAuthor = rowSet.getString(7); //作者

Map<String,Object> map = new HashMap<String,Object>();

map.put("infoId", infoId);

map.put("infoTitle", infoTitle);

map.put("infoDescribe", infoDescribe);

map.put("infoDetail", infoDetail);

map.put("infoType", type);

map.put("infoSupport", support);

map.put("infoAuthor", infoAuthor);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**------------------------** **QueryHotInfo.java结束-- ---------------------------**

QueryInfoByKey.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryInfoByKey，点击finish。打开新建好的QueryInfoByKey.java文件，将下面代码复制到文件中。这部分代码主要就是获取客户端传递的关键词，去info表中的标题和简述查找与关键字相关的内容并封装成Json返回。

**-----------------------** **QueryInfoByKey.java开始- --------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryInfoByKey

\*/

@WebServlet("/QueryInfoByKey")

public class QueryInfoByKey extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize=10; //每页加载数量

int pageNum=1; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryInfoByKey() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String keyWord = request.getParameter("key");

if(keyWord == null||keyWord.length()==0) {

return;

}

String condition ="select \* from info where info\_title like '%"+keyWord+"%' or info\_describe like '%"+keyWord+"%'" ; //按关键词查找

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=totalRecord&&boo;i++){

int infoId = rowSet.getInt(1); //内容ID

String infoTitle = rowSet.getString(2); //内容标题

String infoDescribe = rowSet.getString(3); //内容简述

String infoDetail = rowSet.getString("info\_detail"); //内容详情

String type = rowSet.getString(5); //类型：0表示日记，1表示趣事

String support = rowSet.getString(6); //点赞数

String infoAuthor = rowSet.getString(7); //作者

Map<String,Object> map = new HashMap<String,Object>();

map.put("infoId", infoId);

map.put("infoTitle", infoTitle);

map.put("infoDescribe", infoDescribe);

map.put("infoDetail", infoDetail);

map.put("infoType", type);

map.put("infoSupport", support);

map.put("infoAuthor", infoAuthor);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**------------------------QueryInfoByKey.java结束-- ---------------------------**

QueryPeopleInfoByKey.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryPeopleInfoByKey，点击finish。打开新建好的QueryPeopleInfoByKey.java文件，将下面代码复制到文件中。这部分代码主要就是获取客户端传递的关键词，去user表中的昵称查找与关键字相关的用户信息并封装成Json返回。

**---------------------** **QueryPeopleInfoByKey.java开始-- -------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryPeopleInfo

\*/

@WebServlet("/QueryPeopleInfoByKey")

public class QueryPeopleInfoByKey extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize=10; //每页加载数量

int pageNum=1; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryPeopleInfoByKey() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String keyWord = request.getParameter("nameKey");

if(keyWord == null||keyWord.length()==0) {

return;

}

String condition ="select \* from user where username like '%"+keyWord+"%'" ; //按关键词查找

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=totalRecord&&boo;i++){

String userName = rowSet.getString(1); //用户昵称

//String passWord = rowSet.getString(2); //用户密码

String signature = rowSet.getString(3); //用户签名

String userLogImage = rowSet.getString(4); //用户头像

Map<String,Object> map = new HashMap<String,Object>();

map.put("userName", userName);

/\*map.put("passWord", passWord);\*/

map.put("signature", signature);

map.put("userLogImage", userLogImage);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**----------------------- QueryPeopleInfoByKey.java结束-- -----------------------**

AddInfo.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写AddInfo，点击finish。打开新建好的AddInfo.java文件，将下面代码复制到文件中。这部分代码主要就是获取客户端传递的内容标题、内容简述、内容详情、内容类型、内容发布者，并将点赞数设置为0之后插到内容表info中，完成发布的功能。

**-----------------------------** **AddInfo.java开始-- -----------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.servlet.ServletConfig;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

/\*\*

\* Servlet implementation class AddInfo

\*/

@WebServlet("/AddInfo")

public class AddInfo extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddInfo() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see Servlet#init(ServletConfig)

\*/

public void init(ServletConfig config) throws ServletException {

// TODO Auto-generated method stub

}

//处理中文字符串

public String handleString(String s) {

try {

byte bb[]=s.getBytes("ISO8859-1");

s=new String(bb,"UTF-8");

}catch(Exception e){}

return s;

}

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

Integer infoId =0; //内容ID,默认设置为0

String infoTitle = request.getParameter("infoTitle"); //内容标题

String infoDescribe =request.getParameter("infoDescribe"); //内容简述

String infoDetail = request.getParameter("infoDetail"); //内容详情

String typeStr = request.getParameter("type"); //类型：0表示日记，1表示趣事

Integer type = 0;

if(typeStr == null||typeStr == "") {

typeStr = "0";

}

try {

type = Integer.parseInt(typeStr); //内容类型

} catch (NumberFormatException e) {

e.printStackTrace();

}

Integer support = 0; //点赞数，默认设置为0

String infoAuthor = request.getParameter("infoAuthor"); //作者

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="INSERT INTO info VALUES(?,?,?,?,?,?,?)";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setInt(1, infoId); //自增

prepareStatement.setString(2, infoTitle);

prepareStatement.setString(3, infoDescribe);

prepareStatement.setString(4, infoDetail);

prepareStatement.setInt(5, type);

prepareStatement.setInt(6, support);

prepareStatement.setString(7, infoAuthor);

//4.执行sql

prepareStatement.execute();

out.print("发布成功!");

} catch (Exception e) {

e.printStackTrace();

out.print("发布失败！"+e);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, null);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-----------------------------** **AddInfo.java结束-- ---------------------------**

ChangeSignature.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写ChangeSignature，点击finish。打开新建好的ChangeSignature.java文件，将下面代码复制到文件中。这部分代码主要就是获取客户端传递的用户名、新签名，然后在User表中查找到对应的用户，更新签名。

**--------------------------- ChangeSignature.java开始-- --------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

/\*\*

\* Servlet implementation class ChangeSignature

\*/

@WebServlet("/ChangeSignature")

public class ChangeSignature extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ChangeSignature() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String signature = request.getParameter("signature").trim();

String userName = request.getParameter("username").trim();

if(signature == null) {

signature = "";

}

if(userName == null) {

userName = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

boolean boo = false;

boo = signature.length()>0&&userName.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from user where username=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, userName);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m==true) {

//查询成功

String changeSql = "update user set signature=? where username=?";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setString(1, signature);

prepareStatement2.setString(2, userName);

prepareStatement2.execute();

backnews="修改成功";

}else {

backnews="没有该用户！";

}

}else {

backnews="新签名不能为空";

}

out.print(backnews.toString());

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(backnews);

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

JdbcUtils.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------ChangeSignature.java结束-- -------------------------**

ChangePassword.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写ChangePassword，点击finish。打开新建好的ChangePassword.java文件，将下面代码复制到文件中。这部分代码主要就是获取客户端传递的用户名、旧密码、新密码，然后在User表中查找到对应的用户名和密码的用户，将密码更新。

**-------------------------ChangePassword.java开始-- -----------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

/\*\*

\* Servlet implementation class ChangePassword

\*/

@WebServlet("/ChangePassword")

public class ChangePassword extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public ChangePassword() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String oldPass = request.getParameter("oldPass");

String newPass = request.getParameter("newPass");

String userName = request.getParameter("username").trim();

if(oldPass == null||oldPass.length()==0) {

oldPass="";

}

if(newPass == null||newPass.length()==0) {

newPass="";

}

if(userName == null||userName.length()==0) {

userName = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

boolean boo = false;

boo = oldPass.length()>0&&oldPass.length()>0&&userName.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from user where username=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, userName);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m==true) {

//查询成功

String changeSql = "update user set password=? where username=? and password=?";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setString(1, newPass);

prepareStatement2.setString(2, userName);

prepareStatement2.setString(3, oldPass);

prepareStatement2.execute();

backnews="修改成功";

}else {

backnews="没有该用户！";

}

}else {

backnews="密码不能为空!";

}

out.print(backnews.toString());

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(backnews);

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

JdbcUtils.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**----------------------**--**ChangePassword.java结束-- ------------------------**

QueryAboutMe.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryAboutMe，点击finish。打开新建好的QueryAboutMe.java文件，将下面代码复制到文件中。这部分代码主要就是获取到用户昵称、内容类型，去info表中和focus表中查询“我的分享”、“我的日记”、“我的收藏”，并将数据封装成Json返回。

**--------------------------** **QueryAboutMe.java结束-- -----------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryAboutMe

\*/

@WebServlet("/QueryAboutMe")

public class QueryAboutMe extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize; //每页加载数量

int pageNum; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

int shareType=0;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryAboutMe() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String userName = request.getParameter("username");

String type = request.getParameter("type");

String page = request.getParameter("page");

String count = request.getParameter("count");

if(page == null||page == "") {

page = "1";

}

if(count == null|count == "") {

count = "10";

}

try {

pageNum = Integer.parseInt(page); //第几页

pageSize = Integer.parseInt(count); //每页加载几条

shareType = Integer.parseInt(type);

} catch (NumberFormatException e) {

e.printStackTrace();

}

String condition;

if(shareType==0) {

condition = "select \* from info where (username='"+userName+"') and (info\_type=0 or info\_type=1 or info\_type=2) order by info\_id desc";

}else if(shareType ==1) {

condition = "select \* from info where username='"+userName+"' and info\_type=3 order by info\_id desc";

}else {

condition = "select \* from info left join focus on info.info\_id=focus.info\_id where focus.username='"+userName+"' order by focus.info\_id desc";

}

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=pageSize&&boo;i++){

int infoId = rowSet.getInt(1); //内容ID

String infoTitle = rowSet.getString(2); //内容标题

String infoDescribe = rowSet.getString(3); //内容简述

String infoDetail = rowSet.getString("info\_detail"); //内容详情

String type = rowSet.getString(5); //类型：0表示日记，1表示趣事

String support = rowSet.getString(6); //点赞数

String infoAuthor = rowSet.getString(7); //作者

Map<String,Object> map = new HashMap<String,Object>();

map.put("infoId", infoId);

map.put("infoTitle", infoTitle);

map.put("infoDescribe", infoDescribe);

map.put("infoDetail", infoDetail);

map.put("infoType", type);

map.put("infoSupport", support);

map.put("infoAuthor", infoAuthor);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**----------------------** **QueryAboutMe.java结束-- -------------------------------**

AddSupport.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写AddSupport，点击finish。打开新建好的AddSupport.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id，在内容表info中查找到对应内容，取得点赞数，加一后更新到对应的内容中，完成点赞的功能。

**-----------------------**- **AddSupport.java开始-- ----------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class AddSupport

\*/

@WebServlet("/AddSupport")

public class AddSupport extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddSupport() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String infoId = request.getParameter("InfoId").trim();

if(infoId == null) {

infoId = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

boolean boo = false;

boo = infoId.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from info where info\_id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, infoId);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

int support = rs.getInt(6);

support = support+1;

if(m==true) {

//查询成功

String changeSql = "update info set info\_support=? where info\_id=?";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setInt(1, support);

prepareStatement2.setString(2, infoId);

prepareStatement2.execute();

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", true);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

out.print(backnews.toString());

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(backnews);

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

JdbcUtils.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**------------------------** **AddSupport.java结束-- -------------------------------**

IfAddFocus.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写IfAddFocus，点击finish。打开新建好的IfAddFocus.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id、用户昵称，在收藏表focus中查询是否有对应信息，如果有则表示已经收藏，反之则未收藏。

**-----------------------------** **IfAddFocus.java开始-- --------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class IfAddlike

\*/

@WebServlet("/IfAddFocus")

public class IfAddFocus extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public IfAddFocus() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String username = request.getParameter("username").trim();

String infoId = request.getParameter("infoId").trim();

int focusId = 0;

if(username == null) {

username = "";

}

if(infoId == null) {

infoId = "";

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

boolean boo = false;

boo = username.length()>0&&infoId.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from focus where username=? and info\_id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, username);

prepareStatement.setString(2, infoId);

if(boo) {

//4.执行sql

boolean m =false;

rs = prepareStatement.executeQuery();

while(rs.next()) {

m = true;

focusId = rs.getInt(1);

}

if(m==true) {

success(request,response,focusId);

}else {

fail(request,response);

}

}else {

fail(request,response);

}

} catch (Exception e) {

e.printStackTrace();

fail(request,response);

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void success(HttpServletRequest request, HttpServletResponse response, int focusId) {

try {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

PrintWriter out = response.getWriter();

Map<String,Object> map = new HashMap<String,Object>();

map.put("isFocus", true);

map.put("focusId", focusId);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}catch(IOException e) {

e.printStackTrace();

}

}

public void fail(HttpServletRequest request, HttpServletResponse response) {

response.setContentType("text/html;charset=utf-8");

try {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

PrintWriter out = response.getWriter();

Map<String,Object> map = new HashMap<String,Object>();

map.put("isFocus", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}catch(IOException e) {

e.printStackTrace();

}

}

}

**-------------------------** **IfAddFocus.java结束-- ------------------------**

AddFocus.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写AddFocus，点击finish。打开新建好的AddFocus.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id、用户昵称，将信息插入到收藏表focus中，完成添加收藏功能。

**----------------------------** **AddFocus.java开始-- ---------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class AddFocus

\*/

@WebServlet("/AddFocus")

public class AddFocus extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddFocus() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String infoId = request.getParameter("InfoId").trim();

String username = request.getParameter("username").trim();

if(infoId == null||infoId == "") {

return;

}

if(username == null||username == "") {

return;

}

Connection connection = null;

PreparedStatement prepareStatement = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs = null;

ResultSet rs2 =null;

String backnews="";

boolean boo = false;

boo = infoId.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="select \* from focus where info\_id=? and username=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, infoId);

prepareStatement.setString(2, username);

if(boo) {

//4.执行sql

rs = prepareStatement.executeQuery();

boolean m = rs.next();

if(m!=true) {

String changeSql = "INSERT INTO focus VALUES(?,?,?)";

prepareStatement2 = connection.prepareStatement(changeSql);

prepareStatement2.setInt(1, 0);

prepareStatement2.setString(2, username);

prepareStatement2.setString(3, infoId);

prepareStatement2.execute();

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", true);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

out.print(backnews.toString());

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(backnews);

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

JdbcUtils.close(connection, prepareStatement2, rs2);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**-------------------------** **AddFocus.java结束-- ----------------------------**

RemoveFocus.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写RemoveFocus，点击finish。打开新建好的RemoveFocus.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id、用户昵称，将对应的信息从收藏表focus中删除，完成取消收藏的功能。

**--------------------------** **RemoveFocus.java开始-- ----------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class RemoveFocus

\*/

@WebServlet("/RemoveFocus")

public class RemoveFocus extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public RemoveFocus() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String focusId = request.getParameter("focusId").trim();

if(focusId == null||focusId == "") {

return;

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

boolean boo = false;

boo = focusId.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="delete from focus where focus\_id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, focusId);

if(boo) {

//4.执行sql

prepareStatement.execute();

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("removeOk", true);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("removeOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

} catch (Exception e) {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("removeOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**--------------------------**- **RemoveFocus.java结束-- ----------------------------**

DeleteInfo.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写DeleteInfo，点击finish。打开新建好的DeleteInfo.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id，将对应的信息从内容表info中删除，完成删除内容的功能。

**---------------------------** **DeleteInfo.java结束-- ---------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class DeleteInfo

\*/

@WebServlet("/DeleteInfo")

public class DeleteInfo extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public DeleteInfo() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String infoId = request.getParameter("infoId").trim();

if(infoId == null||infoId == "") {

return;

}

Connection connection = null;

PreparedStatement prepareStatement = null;

ResultSet rs = null;

PreparedStatement prepareStatement2 = null;

ResultSet rs2 =null;

PreparedStatement prepareStatement3 = null;

ResultSet rs3 =null;

String backnews="";

boolean boo = false;

boo = infoId.length()>0;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="delete from focus where info\_id=?";

prepareStatement = connection.prepareStatement(sql);

prepareStatement.setString(1, infoId);

String sql2 ="delete from comment where comment\_info=?";

prepareStatement2 = connection.prepareStatement(sql2);

prepareStatement2.setString(1, infoId);

if(boo) {

//4.执行sql

prepareStatement.execute();

prepareStatement2.execute();

String changeSql = "delete from info where info\_id=?";

prepareStatement3 = connection.prepareStatement(changeSql);

prepareStatement3.setString(1, infoId);

prepareStatement3.execute();

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("deleteOk", true);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("deleteOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

out.print(backnews.toString());

} catch (Exception e) {

backnews="修改失败"+e.toString();

out.print(backnews);

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, rs);

JdbcUtils.close(connection, prepareStatement2, rs2);

JdbcUtils.close(connection, prepareStatement3, rs3);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

}

**------------------------ DeleteInfo.java结束-- --------------------------**

QueryComment.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写QueryComment，点击finish。打开新建好的QueryComment.java文件，将下面代码复制到文件中。这部分代码主要就是获取到内容Id，在评论表comment中查询对应的评论信息，并将数据封装成json返回。

**---------------------------**- **QueryComment.java开始-- --------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.Statement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import com.sun.rowset.CachedRowSetImpl;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class QueryComment

\*/

@WebServlet("/QueryComment")

public class QueryComment extends HttpServlet {

private static final long serialVersionUID = 1L;

CachedRowSetImpl rowSet = null; //存储表中全部记录的行集对象

int pageSize=10; //每页加载数量

int pageNum=1; //第几页

int totalRecord; //总记录数

int totalPage; //总页数

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public QueryComment() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

String infoId = request.getParameter("infoId").trim();

String condition = "select \* from comment where comment\_info='"+infoId+"' order by comment\_id desc";

Connection connection = null;

Statement sql = null;

ResultSet rs = null;

try {

connection = JdbcUtils.getConnection();

sql = connection.createStatement(ResultSet.TYPE\_SCROLL\_SENSITIVE, ResultSet.CONCUR\_READ\_ONLY);

rs = sql.executeQuery(condition);

rowSet = new CachedRowSetImpl(); //创建行集对象

rowSet.populate(rs);

//按查询页数返回结果

returnByPage(request,response,rowSet);

} catch (Exception e) {

out.println(condition+"异常："+e.toString());

e.printStackTrace();

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.statementClose(connection, sql, rs);

}

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

doGet(request, response);

}

public void returnByPage(HttpServletRequest request, HttpServletResponse response,CachedRowSetImpl rowSet)throws ServletException, IOException{

response.setContentType("text/html;charset=utf-8");

response.setCharacterEncoding("utf-8");

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

JSONArray TotaljsonArray = new JSONArray();//存放返回的jsonOjbect数组

//将rowSet的数据提取到Map

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

//将rowSet的数据提取到Map

List<Map<String,Object>> Totaldata = new ArrayList<Map<String,Object>>();

try {

PrintWriter out = response.getWriter();

try {

rowSet.last(); //移到随后一行

totalRecord = rowSet.getRow();

if(totalRecord%pageSize==0){

totalPage = totalRecord/pageSize; //总页数

}else{

totalPage = totalRecord/pageSize+1;

}

int index = (pageNum-1)\*pageSize+1;

rowSet.absolute(index); //查询位置移动到查询页的起始记录位置

boolean boo = true;

for(int i=1; i<=pageSize&&boo;i++){

int commentId = rowSet.getInt(1); //评论内容ID

String commentUser = rowSet.getString(2); //评论者

int commentInfo = rowSet.getInt(3);

String commentDetail = rowSet.getString(4); //评论详情

Map<String,Object> map = new HashMap<String,Object>();

map.put("commentId", commentId);

map.put("commentUser", commentUser);

map.put("commentInfo", commentInfo);

map.put("commentDetail", commentDetail);

data.add(map);

boo = rowSet.next();

}

jsonArray = JsonUtils.formatRsToJsonArray(data);

Map<String,Object> map = new HashMap<String,Object>();

map.put("totalRecord", totalRecord);

map.put("RecordDetail", jsonArray);

Totaldata.add(map);

TotaljsonArray = JsonUtils.formatRsToJsonArray(Totaldata);

out.println(TotaljsonArray.toString()); //返回json

}catch(Exception e) {

out.println("null");

}

}catch(IOException e) {

}

}

}

**-----------------------**- **QueryComment.java结束-- -----------------------------**

AddComment.java

在创建好的myServlet.data包中右击选择New，点击Servlet，在Class name中填写AddComment，点击finish。打开新建好的AddComment.java文件，将下面代码复制到文件中。这部分代码主要就是获取到用户昵称、内容ID，并将信息插入到评论表中，完成评论的功能。

**-------------------------**-- **AddComment.java开始- ---------------------------**

package myServlet.data;

import java.io.IOException;

import java.io.PrintWriter;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.utils.JdbcUtils;

import org.utils.JsonUtils;

import net.sf.json.JSONArray;

/\*\*

\* Servlet implementation class AddComment

\*/

@WebServlet("/AddComment")

public class AddComment extends HttpServlet {

private static final long serialVersionUID = 1L;

/\*\*

\* @see HttpServlet#HttpServlet()

\*/

public AddComment() {

super();

// TODO Auto-generated constructor stub

}

/\*\*

\* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html;charset=utf-8");

request.setCharacterEncoding("utf-8");

response.setCharacterEncoding("utf-8");

PrintWriter out = response.getWriter();

Integer commentId =0; //内容ID,默认设置为0

String infoIdStr = request.getParameter("infoId"); //内容标题

String commentUser =request.getParameter("commentUser"); //内容简述

String commentDetail = request.getParameter("commentDetail"); //内容详情

if(infoIdStr == null||infoIdStr == "") {

infoIdStr = "";

}

if(commentUser == null||commentUser == "") {

commentUser = "";

}

if(commentDetail == null||commentDetail == "") {

commentDetail = "";

}

Boolean m = infoIdStr.length()>0&&commentUser.length()>0&&commentDetail.length()>0;

int infoId = 0;

try {

infoId = Integer.parseInt(infoIdStr); //内容类型

} catch (NumberFormatException e) {

e.printStackTrace();

}

Connection connection = null;

PreparedStatement prepareStatement = null;

try {

connection = JdbcUtils.getConnection();

//3.获取statement

String sql ="INSERT INTO comment VALUES(?,?,?,?)";

prepareStatement = connection.prepareStatement(sql);

if(m) {

prepareStatement.setInt(1, commentId); //自增

prepareStatement.setString(2, commentUser);

prepareStatement.setInt(3, infoId);

prepareStatement.setString(4, commentDetail);

//4.执行sql

prepareStatement.execute();

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", true);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}else {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}

} catch (Exception e) {

JSONArray jsonArray = new JSONArray();//存放返回的jsonOjbect数组

List<Map<String,Object>> data = new ArrayList<Map<String,Object>>();

Map<String,Object> map = new HashMap<String,Object>();

map.put("addOk", false);

data.add(map);

jsonArray = JsonUtils.formatRsToJsonArray(data);

out.println(jsonArray.toString()); //返回json

}finally {

//5.释放资源 connection prepareStatement

JdbcUtils.close(connection, prepareStatement, null);

}

out.flush();

out.close();

}

/\*\*

\* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

// TODO Auto-generated method stub

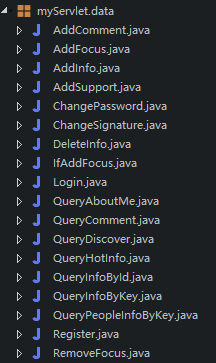
doGet(request, response);

}

}

**---------------------------** **AddComment.java结束-- ----------------------------**

基本上Servlet到这里就已经写完整了，如下图所见一共有18个servlet文件



web.xml配置

打开项目iShareService，找到WebRoot>WEB-INF文件夹，打开文件下的web.xml文件，如果没有该文件就新建一个，然后将下面的代码复制到文件中。这个文件主要就是配置servlet的映射。

**-----------------------------**- **web.xml开始-- --------------------------**

<?xml version="1.0" encoding="UTF-8"?>

<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd" version="3.0">

<servlet>

<description>This is the description of my J2EE component</description>

<display-name>This is the display name of my J2EE component</display-name>

<servlet-name>Register</servlet-name>

<servlet-class>myServlet.data.Register</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Register</servlet-name>

<url-pattern>/servlet/Register</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>Login</servlet-name>

<servlet-class>myServlet.data.Login</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>Login</servlet-name>

<url-pattern>/servlet/Login</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryDiscover</servlet-name>

<servlet-class>myServlet.data.QueryDiscover</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryDiscover</servlet-name>

<url-pattern>/servlet/QueryDiscover</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryHotInfo</servlet-name>

<servlet-class>myServlet.data.QueryHotInfo</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryHotInfo</servlet-name>

<url-pattern>/servlet/QueryHotInfo</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryInfoByKey</servlet-name>

<servlet-class>myServlet.data.QueryInfoByKey</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryInfoByKey</servlet-name>

<url-pattern>/servlet/QueryInfoByKey</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryPeopleInfoByKey</servlet-name>

<servlet-class>myServlet.data.QueryPeopleInfoByKey</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryPeopleInfoByKey</servlet-name>

<url-pattern>/servlet/QueryPeopleInfoByKey</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>AddInfo</servlet-name>

<servlet-class>myServlet.data.AddInfo</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AddInfo</servlet-name>

<url-pattern>/servlet/AddInfo</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>ChangeSignature</servlet-name>

<servlet-class>myServlet.data.ChangeSignature</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ChangeSignature</servlet-name>

<url-pattern>/servlet/ChangeSignature</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryInfoById</servlet-name>

<servlet-class>myServlet.data.QueryInfoById</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryInfoById</servlet-name>

<url-pattern>/servlet/QueryInfoById</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>IfAddFocus</servlet-name>

<servlet-class>myServlet.data.IfAddFocus</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>IfAddFocus</servlet-name>

<url-pattern>/servlet/IfAddFocus</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>AddSupport</servlet-name>

<servlet-class>myServlet.data.AddSupport</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AddSupport</servlet-name>

<url-pattern>/servlet/AddSupport</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>AddFocus</servlet-name>

<servlet-class>myServlet.data.AddFocus</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AddFocus</servlet-name>

<url-pattern>/servlet/AddFocus</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>RemoveFocus</servlet-name>

<servlet-class>myServlet.data.RemoveFocus</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>RemoveFocus</servlet-name>

<url-pattern>/servlet/RemoveFocus</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>DeleteInfo</servlet-name>

<servlet-class>myServlet.data.DeleteInfo</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>DeleteInfo</servlet-name>

<url-pattern>/servlet/DeleteInfo</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>ChangePassword</servlet-name>

<servlet-class>myServlet.data.ChangePassword</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ChangePassword</servlet-name>

<url-pattern>/servlet/ChangePassword</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryAboutMe</servlet-name>

<servlet-class>myServlet.data.QueryAboutMe</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryAboutMe</servlet-name>

<url-pattern>/servlet/QueryAboutMe</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>QueryComment</servlet-name>

<servlet-class>myServlet.data.QueryComment</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>QueryComment</servlet-name>

<url-pattern>/servlet/QueryComment</url-pattern>

</servlet-mapping>

<servlet>

<servlet-name>AddComment</servlet-name>

<servlet-class>myServlet.data.AddComment</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>AddComment</servlet-name>

<url-pattern>/servlet/AddComment</url-pattern>

</servlet-mapping>

</web-app>

**---------------------------------**-- **web.xml结束-- -----------------------------**

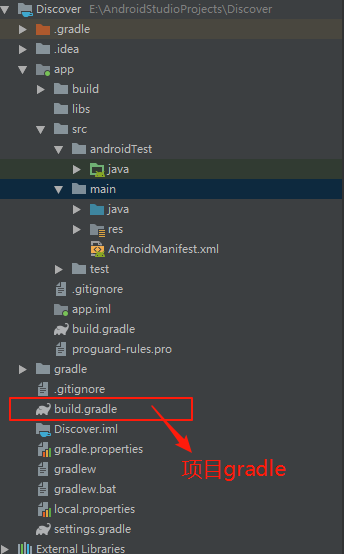
1. 创建Android应用

打开Android Studio，新建Project，在Name中填写Discover，Package name默认为com.example.myapplication，选择项目保存的地址，默认Minimum API level为API 21：Android 5.0(Lollipop)，点击finish。

环境配置

项目下的build.gradle

在Android Studio左上角处将项目显示格式从默认的Android改为Project（个人习惯在这个模式下看代码）。打开创建好的项目Discover，找到build.gradle，将下面的代码复制到文件中。



--------------------------**项目下的build.gradle开始-- ----------------------**

// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {

repositories {

google()

jcenter()

}

dependencies {

classpath 'com.android.tools.build:gradle:3.3.2'

}

}

allprojects {

repositories {

google()

jcenter()

}

}

task clean(type: Delete) {

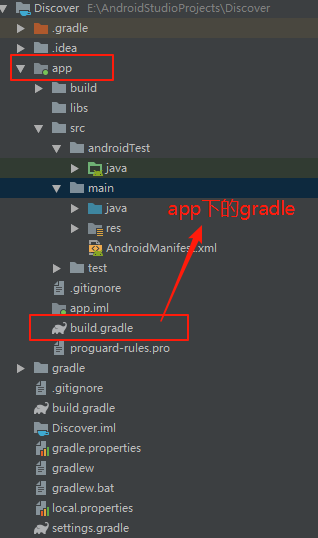
delete rootProject.buildDir

}

--------------------------**项目下的build.gradle结束-- ------------------------**

app下的build.gradle

打开项目Discover，找到app，打开app文件夹，找到build.gradle，将下面的代码复制到文件中。复制完之后，在编辑器上方会出现弹框提示，点击蓝字Sync Now让配置生效。



---------------------------**app下的build.gradle开始-- -------------------------**

apply plugin: 'com.android.application'

android {

compileSdkVersion 28

defaultConfig {

applicationId "com.example.discover"

minSdkVersion 21

targetSdkVersion 28

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

//注解报错的解决方案

javaCompileOptions {

annotationProcessorOptions {

includeCompileClasspath true

}

}

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

implementation fileTree(include: ['\*.jar'], dir: 'libs')

implementation 'com.android.support:appcompat-v7:28.0.0'

implementation 'com.android.support:cardview-v7:28.0.0'

implementation 'com.android.support.constraint:constraint-layout:1.1.3'

implementation 'com.android.support:recyclerview-v7:28.0.0'

implementation 'com.android.support:appcompat-v7:28.0.0'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.2'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

}以及

-----------------------**app下的build.gradle结束-- --------------------------**

引入资源

drawable下的资源

打开项目Discover，找到app>src>main>res>drawable，打开drawable将文末提供资源链接中对应文件夹下的图片以及六个样式xml文件复制到该文件夹下。

values下的资源

打开项目Discover，找到app>src>main>res>values，打开文件夹可以看到默认存在的xml文件。同时打开文末提供资源链接对应的文件夹，将文件夹中的colors.xml、dimens.xml、strings.xml、styles.xml资源复制到该文件夹values下。若文件已经默认存在，则将对应的代码复制进来，覆盖默认存在的代码。

页面设计

“启动”页——splash

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为splash。打开创建的splash.xml，将下面的布局文件代码复制到文件中。该页面是启动页，所以只放了LOGO和slogan，并添加了淡入效果。

--------------------------- **splash .xml开始 ------------------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout android:id="@+id/LinearLayout01"

android:layout\_width="fill\_parent" android:layout\_height="fill\_parent"

xmlns:android="http://schemas.android.com/apk/res/android"

android:gravity="center"

android:background="@drawable/home\_gradient"

android:orientation="vertical">

<ImageView android:layout\_marginTop="-60dip"

android:paddingLeft="20dip"

android:paddingRight="20dip"

android:scaleType="centerInside"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/logo"

android:src="@drawable/logo">

</ImageView>

<!-- android:scaleType是控制图片如何resized/moved来匹对ImageView的size CENTER\_INSIDE

/ centerInside 将图片的内容完整居中显示，通过按比例缩小或原来的size使得图片长/宽等于或小于View的长/宽 -->

<TextView android:text="i Share"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_marginTop="20dip"

android:typeface="serif"

android:shadowDx="0"

android:shadowDy="2"

android:shadowRadius="1"

android:shadowColor="#FFFFFF"

android:textColor="#444444"

android:textSize="35dip"

android:gravity="bottom">

</TextView>

<TextView android:text="爱分享"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:layout\_marginTop="20dip" android:typeface="serif"

android:shadowDx="0" android:shadowDy="2" android:shadowRadius="1"

android:shadowColor="#FFFFFF" android:textColor="#444444"

android:textSize="20dip" android:gravity="bottom">

</TextView>

<TextView android:layout\_width="wrap\_content"

android:text="发现有趣的人和事"

android:layout\_height="wrap\_content" android:layout\_marginTop="20dip"

android:typeface="serif" android:shadowDx="0" android:shadowDy="2"

android:shadowRadius="1" android:shadowColor="#FFFFFF"

android:textColor="#444444" android:textSize="20dip" android:id="@+id/versionNumber"

android:gravity="bottom">

</TextView>

</LinearLayout>

---------------------- **-** **splash .xml结束 -------------------------------------**

“主页”—— main\_tab

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为main\_tab。打开创建的main\_tab.xml，将下面的布局文件代码复制到文件中。该页面是启动页跳转进来的页面，作为首页，它用于显示底部的四个tab以及控制管理对应的Fragment。

------------------------ **-main\_tab .xml开始 --------------------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical" >

<FrameLayout

android:id="@+id/content"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1" >

</FrameLayout>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="2dp"

android:layout\_marginBottom="4dp"

android:background="@color/bg\_gray"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp">

<RelativeLayout

android:id="@+id/discover\_layout"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/discover\_image"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_two\_un\_select" />

<TextView

android:id="@+id/discover\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="发现"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/find\_layout"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/find\_image"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_three\_un\_select" />

<TextView

android:id="@+id/find\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="查找"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/edit\_layout"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/more\_image"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_four\_un\_select" />

<TextView

android:id="@+id/more\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="发布"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/my\_layout"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/my\_image"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_one\_un\_select" />

<TextView

android:id="@+id/my\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="我的"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

</LinearLayout>

</LinearLayout>

------------------------------ **-main\_tab .xml结束 -------------------------**

“发现”页——discover\_tab\_content

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为discover\_tab\_content。打开创建的discover\_tab\_content.xml，将下面的布局文件代码复制到文件中。该页面是首页默认显示的Fragment，包含顶部的bar和一个listView。

-----------------------**discover\_tab\_content .xml开始 ------------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#ebebeb">

<LinearLayout android:gravity="center\_vertical"

android:orientation="horizontal" android:id="@+id/title\_bar"

android:background="@drawable/title\_bg\_night" android:layout\_width="fill\_parent"

android:layout\_height="35.0dip">

<ImageView android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:src="@drawable/app\_icon" />

<LinearLayout android:gravity="center\_vertical"

android:layout\_width="0.0dip" android:layout\_height="wrap\_content"

android:layout\_weight="1.0">

<TextView android:textSize="16.0sp" android:textColor="@color/titleTextColor"

android:singleLine="true" android:id="@+id/myTitle"

android:layout\_width="fill\_parent" android:ellipsize="end"

android:layout\_height="wrap\_content" android:layout\_marginLeft="2.0dip"

android:text="@string/app\_name" />

</LinearLayout>

<ImageButton android:id="@+id/back\_button"

android:layout\_gravity="center" android:background="@color/transparent"

android:layout\_width="60.0dip" android:layout\_height="29.0dip"

android:layout\_marginRight="5.0dip" android:src="@drawable/exit" />

</LinearLayout>

<!-- 这里是最开始用于测试的元素

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical" >

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_two" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:padding="10dp"

android:text="这是发现界面"

android:textSize="20sp" />

</LinearLayout>-->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="15dp"

android:paddingRight="15dp"

android:orientation="vertical"

android:paddingTop="50dp">

<ListView

android:id="@+id/discover\_listView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:dividerHeight="15dp"/>

</LinearLayout>

</RelativeLayout>

------------------------ **discover\_tab\_content .xml结束--------------------**

“内容”列表项——activity\_find\_list\_item

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_find\_list\_item。打开创建的activity\_find\_list\_item.xml，将下面的布局文件代码复制到文件中。这个文件存放的是展示内容的ListView下的item布局，分为标题和内容两部分，其中内容分为三种显示方式，这将根据文章类型确定。

------------------- **activity\_find\_list\_item .xml开始------------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/list\_card\_style"

android:padding="20dp"

android:orientation="vertical">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<TextView

android:id="@+id/cardTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:textColor="#000000"

android:text="我的数据" />

<TextView

android:id="@+id/cardId"

android:visibility="gone"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:textColor="#000000"

android:text="这里存放文章Id" />

<ImageView

android:id="@+id/cardImgTip"

android:layout\_width="20dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:src="@drawable/my\_tab\_list2"

/>

</RelativeLayout>

<View

android:layout\_width="match\_parent"

android:layout\_height="1dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:background="#000000"/>

<TextView

android:id="@+id/cardContent"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18sp"

android:textColor="#000000"

android:text="这里放一些内容，这里放一些图片，这里扩展性很强"/>

<!-- 分享为图片/图书类型则显示-->

<ImageView

android:id="@+id/cardContent\_pic"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/logo"

android:visibility="gone"/>

<!-- 分享为音乐类型则显示-->

<LinearLayout

android:id="@+id/cardContent\_music"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:gravity="center"

android:visibility="gone">

<TextView

android:id="@+id/cardTitle\_music"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:textColor="#000000"

android:text="演员——薛之谦" />

<ImageView

android:id="@+id/cardImg\_musicBac"

android:layout\_width="20dp"

android:layout\_height="wrap\_content"

android:src="@drawable/play\_music"

/>

</LinearLayout>

</LinearLayout>

-------------------- **activity\_find\_list\_item**.**xml结束------------------------**

“加载更多”—— activity\_find\_loadmore

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_find\_loadmore。打开创建的activity\_find\_loadmore.xml，将下面的布局文件代码复制到文件中。这个文件存放的是一个“加载更多”的按钮布局，用于分页加载时使用。

-------------------- **activity\_find\_loadmore**.**xml开始--------------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center">

<Button

android:id="@+id/loadMoreButton"

android:layout\_marginTop="4dp"

android:layout\_marginBottom="4dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="#00000000"

android:text="查看更多..."

/>

</LinearLayout>

--------------------- **activity\_find\_loadmore**.**xml结束------------------------**

“详情”页——activity\_info\_detail

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_info\_detail。打开创建的activity\_info\_detail.xml，将下面的布局文件代码复制到文件中。这个文件是内容详情页，主要用于展示内容详情，包括内容标题、简述、详情、作者、点赞数、收藏、删除、评论这几个部分。其中页面将图片展示框设置为隐藏状态，这部分的显示由文章类型确定。

----------------------- **activity\_info\_detail.xml开始-----------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<FrameLayout

android:id="@+id/share\_detail\_img"

android:visibility="gone"

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:background="@drawable/share\_img\_2">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="35dip"

android:layout\_gravity="bottom"

android:background="#73000000"

android:gravity="center"

android:orientation="vertical">

<TextView

android:id="@+id/share\_img\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="今日分享：分享每一处美景"

android:textColor="@android:color/white" />

</LinearLayout>

</FrameLayout>

<ScrollView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:fadingEdge="none" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:paddingLeft="14dp"

android:paddingRight="14dp"

android:paddingBottom="60dp"

android:orientation="vertical">

<TextView

android:id="@+id/findDetail\_info\_title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="22dp"

android:textStyle="bold"

android:text="这是文章的标题，不长" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp">

<TextView

android:id="@+id/findDetail\_info\_author"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:layout\_weight="1"

android:textColor="#000000"

android:text="作者昵称" />

<ImageView

android:id="@+id/findDetail\_author\_like"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:scaleType="fitEnd"

android:src="@drawable/add\_focus"

/>

</LinearLayout>

<TextView

android:background="@drawable/list\_card\_style"

android:id="@+id/findDetail\_info\_describe"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/margin\_10"

android:padding="10dp"

android:textSize="18sp"

android:textColor="#000000"

android:text="这里放一些内容，主要是对下面详细内容的简述，具体根据文章类型变化"/>

<TextView

android:id="@+id/detail\_info\_detail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/margin\_10"

android:padding="10dp"

android:textSize="18sp"

android:textColor="#000000"

android:singleLine="false"

android:text="这里放一些文章的详细内容，具体会根据文章类型变化

这里放一些文章的详细内容，具体会根据文章类型变化这里放一些文章的详细内容

"/>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp">

<TableLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="1.0dip"

android:shrinkColumns="0"

android:stretchColumns="0" >

<TableRow

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip">

<TextView

android:id="@+id/detail\_info\_support"

android:layout\_width="wrap\_content"

android:layout\_marginRight="200dp"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/not\_support"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="4.0dip"

android:text="点赞数"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageButton

android:id="@+id/detail\_info\_delete"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:background="#00000000"

android:visibility="gone"

android:src="@drawable/delete\_info" />

<ImageButton

android:id="@+id/detail\_info\_focus"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="10.0dip"

android:background="#00000000"

android:src="@drawable/now\_not\_like" />

</TableRow>

</TableLayout>

</LinearLayout>

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="50dp"

android:layout\_marginTop="20dp"

android:background="@drawable/small\_bc">

<!-- 输入的评论信息 -->

<EditText

android:id="@+id/et\_comment\_detail"

android:paddingLeft="20dp"

android:layout\_width="match\_parent"

android:layout\_height="fill\_parent"

android:layout\_weight="4"

android:background="@null"

android:gravity="center\_vertical"

android:hint="输入评论"

android:textColor="#0e0e0e"

android:textColorHint="#b0c6ce"

android:textSize="17sp" />

<!-- 评论 -->

<Button

android:id="@+id/comment\_btn"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_gravity="right|center\_vertical"

android:layout\_margin="24dp"

android:background="@drawable/add\_comment" />

</LinearLayout>

<LinearLayout

android:layout\_below="@+id/title\_bar"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical"

android:paddingTop="10dp">

<View

android:layout\_width="match\_parent"

android:layout\_height="1dp"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:background="#000000"/>

<ListView

android:id="@+id/comment\_listView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:dividerHeight="0dp" />

</LinearLayout>

</LinearLayout>

</ScrollView>

</LinearLayout>

------------------------**activity\_info\_detail.xml结束-------------------------**

“评论”列表项——activity\_comment\_list

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_comment\_list。打开创建的activity\_comment\_list.xml，将下面的布局文件代码复制到文件中。这个文件存放的是展示评论的ListView下的item布局，分为评论者昵称和评论详情两部分。

----------------------- **activity\_comment\_list.xml开始--------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="20dp"

android:orientation="vertical">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center">

<ImageView

android:id="@+id/cardImgTip"

android:layout\_width="20dp"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:src="@drawable/comment\_head"

/>

<TextView

android:id="@+id/comment\_user\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical"

android:paddingLeft="30dp"

android:textSize="18sp"

android:textStyle="bold"

android:textColor="#513861"

android:text="评论者昵称" />

<TextView

android:id="@+id/comment\_id"

android:visibility="gone"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:textColor="#524A47"

android:text="评论的ID" />

</RelativeLayout>

<TextView

android:id="@+id/comment\_detail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18sp"

android:textColor="#000000"

android:text="这里放的是评论内容"/>

</LinearLayout>

---------------------**activity\_comment\_list.xml结束--------------------------**

“查找”页——find\_tab\_content

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为find\_tab\_content。打开创建的find\_tab\_content.xml，将下面的布局文件代码复制到文件中。这个文件用于显示搜索框、热门内容列表、查找内容列表。页面整体分为两部分，上半部分是搜索框，下半部分是内容展示区。

------------------------ **find\_tab\_content.xml开始----------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#ffffff">

<LinearLayout

android:gravity="center\_vertical"

android:orientation="horizontal" android:id="@+id/title\_bar"

android:background="@drawable/title\_bg\_night" android:layout\_width="fill\_parent"

android:layout\_height="35.0dip">

<ImageView android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:src="@drawable/app\_icon" />

<LinearLayout android:gravity="center\_vertical"

android:layout\_width="0.0dip" android:layout\_height="wrap\_content"

android:layout\_weight="1.0">

<TextView android:textSize="16.0sp" android:textColor="@color/titleTextColor"

android:singleLine="true" android:id="@+id/myTitle"

android:layout\_width="fill\_parent" android:ellipsize="end"

android:layout\_height="wrap\_content" android:layout\_marginLeft="2.0dip"

android:text="@string/app\_name" />

</LinearLayout>

<ImageButton android:id="@+id/back\_button2"

android:layout\_gravity="center" android:background="@color/transparent"

android:layout\_width="60.0dip" android:layout\_height="29.0dip"

android:layout\_marginRight="5.0dip" android:src="@drawable/exit" />

</LinearLayout>

<!--搜索框-->

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/find\_search\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="#CCFFFF"

android:paddingBottom="30dp"

android:orientation="vertical"

android:gravity="center"

android:layout\_below="@+id/title\_bar">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="50dp"

android:layout\_marginLeft="20dp"

android:layout\_marginRight="20dp"

android:layout\_marginTop="5dp"

android:background="@drawable/small\_bc"

android:orientation="horizontal">

<Button

android:layout\_width="20dp"

android:layout\_height="20dp"

android:layout\_gravity="right|center\_vertical"

android:layout\_margin="30dp"

android:background="@drawable/search" />

<!-- 输入的搜索信息 -->

<EditText

android:id="@+id/et\_seek\_search"

android:layout\_width="0dp"

android:layout\_height="fill\_parent"

android:layout\_weight="4"

android:background="@null"

android:gravity="center\_vertical"

android:hint="输入想要搜索的关键字"

android:imeOptions="actionSearch"

android:singleLine="true"

android:textColor="#0e0e0e"

android:textColorHint="#b0c6ce"

android:textSize="17sp" />

<!-- 清除查询-->

<Button

android:id="@+id/seek\_bt\_clear"

android:layout\_width="25dp"

android:layout\_height="25dp"

android:layout\_gravity="right|center\_vertical"

android:layout\_margin="30dp"

android:background="@drawable/delete" />

</LinearLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:orientation="horizontal">

<Button

android:id="@+id/find\_search\_by\_name"

android:layout\_width="70dp"

android:layout\_height="30dp"

android:background="@drawable/button\_radius"

android:text="找人"

android:gravity="center"

/>

<Button

android:id="@+id/find\_search\_by\_keyWord"

android:layout\_width="70dp"

android:layout\_height="30dp"

android:layout\_marginLeft="10dp"

android:background="@drawable/button\_radius"

android:text="找文"

android:gravity="center"

/>

</LinearLayout>

</LinearLayout>

<!-- 测试专用

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical" >

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_three" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:padding="10dp"

android:text="这是查找界面"

android:textSize="20sp" />

</LinearLayout>-->

<LinearLayout

android:layout\_below="@+id/title\_bar"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical"

android:paddingTop="140dp">

<TextView

android:id="@+id/seek\_list\_hint\_info"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="6.0dip"

android:gravity="center\_vertical"

android:paddingLeft="10.0dip"

android:text="大家都在看："

android:textColor="#DF3154"

android:textSize="16.0sp" />

<ListView

android:id="@+id/find\_listView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:dividerHeight="0dp" />

</LinearLayout>

</RelativeLayout>

------------------------------ **find\_tab\_content.xml结束--------------------**

“查找内容”列表项——activity\_seek\_list\_item

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_seek\_list\_item。打开创建的activity\_seek\_list\_item.xml，将下面的布局文件代码复制到文件中。这个文件存放的是查找内容、热门内容的ListView下的item布局，文件包含标题和点赞数两部分，其中当查找用户时则显示用户昵称和签名。

**------------------- activity\_seek\_list\_item .xml开始------------------------**

<?xml version="1.0" encoding="UTF-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ffffff">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical"

android:orientation="vertical">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TableLayout

android:id="@+id/seek\_list\_tableLayout"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="1.0dip"

android:shrinkColumns="0"

android:stretchColumns="0"

android:background="#ebebeb">

<TableRow

android:id="@+id/seek\_list\_tableRow"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:background="@drawable/small\_bc"

android:paddingBottom="12.0dip"

android:paddingTop="12.0dip" >

<TextView

android:id="@+id/seek\_list\_textView"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/hot\_key"

android:drawablePadding="16.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:text="热门标题"

android:textColor="#ff333333"

android:textSize="16.0sp"

android:paddingLeft="20dp"/>

<TextView

android:id="@+id/seek\_list\_textView2"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:text="12345"

android:textColor="#424202"

android:textSize="16.0sp"

android:paddingRight="20dp"/>

<!--android:adjustViewBounds="true"图片自适应-->

<TextView

android:id="@+id/seek\_list\_cardId"

android:visibility="gone"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:textColor="#000000"

android:text="" />

<ImageView

android:adjustViewBounds="true"

android:id="@+id/seek\_list\_imgView"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:src="@drawable/next\_right"

android:paddingRight="30dp"/>

</TableRow>

</TableLayout>

</LinearLayout>

</LinearLayout>

</LinearLayout>

--------------------- **activity\_seek\_list\_item .xml结束-----------------------**

“发布”页——edit\_tab\_content

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为edit\_tab\_content。打开创建的edit\_tab\_content.xml，将下面的布局文件代码复制到文件中。这个页面包含发布内容类型选择tab、内容标题输入框、内容简述输入框、内容详情输入框以及一个发布按钮。

------------------------- **edit\_tab\_content .xml开始----------------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<LinearLayout android:gravity="center\_vertical"

android:orientation="horizontal" android:id="@+id/title\_bar"

android:background="@drawable/title\_bg\_night" android:layout\_width="fill\_parent"

android:layout\_height="35.0dip">

<ImageView android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" android:src="@drawable/app\_icon" />

<LinearLayout android:gravity="center\_vertical"

android:layout\_width="0.0dip" android:layout\_height="wrap\_content"

android:layout\_weight="1.0">

<TextView android:textSize="16.0sp" android:textColor="@color/titleTextColor"

android:singleLine="true" android:id="@+id/myTitle"

android:layout\_width="fill\_parent" android:ellipsize="end"

android:layout\_height="wrap\_content" android:layout\_marginLeft="2.0dip"

android:text="@string/app\_name" />

</LinearLayout>

<ImageButton android:id="@+id/back\_button3"

android:layout\_gravity="center" android:background="@color/transparent"

android:layout\_width="60.0dip" android:layout\_height="29.0dip"

android:layout\_marginRight="5.0dip" android:src="@drawable/exit" />

</LinearLayout>

<!--专用测试

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:orientation="vertical" >

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/tab\_icon\_four" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:padding="10dp"

android:text="这是发布界面"

android:textSize="20sp" />

</LinearLayout>-->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:layout\_marginTop="50dp">

<RelativeLayout

android:id="@+id/bt\_share\_funny"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/share\_funny\_image\_bc"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/share\_funny\_select"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="分享趣事"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/bt\_share\_pic"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/share\_pic\_image\_bc"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/share\_pic\_no\_select" />

<TextView

android:id="@+id/find\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="分享图片"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/bt\_share\_music"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/share\_music\_image\_bc"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/share\_music\_no\_select" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="分享音乐"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

<RelativeLayout

android:id="@+id/bt\_share\_diary"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:layout\_weight="1" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:orientation="vertical" >

<ImageView

android:id="@+id/share\_diary\_image\_bc"

android:layout\_width="wrap\_content"

android:layout\_height="40dp"

android:layout\_gravity="center\_horizontal"

android:src="@drawable/share\_diary\_no\_select" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:text="发布日记"

android:textColor="#82858b" />

</LinearLayout>

</RelativeLayout>

</LinearLayout>

<!--分享趣-->

<LinearLayout

android:id="@+id/edit\_base\_info"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_below="@+id/rl\_login\_top"

android:layout\_marginTop="140dp"

android:layout\_marginLeft="26dp"

android:layout\_marginRight="26dp"

android:gravity="center"

>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="标 题："/>

<EditText

android:id="@+id/et\_edit\_title\_info"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:maxLength="10"

android:paddingLeft="10.0dip"

android:singleLine="true"

android:hint="标题请保持在10个字以内"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:maxLength="40"

android:text="简 述："/>

<EditText

android:id="@+id/et\_edit\_describe\_info"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10.0dip"

android:hint="简述你要分享的趣事" />

</LinearLayout>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:layout\_marginBottom="10dp"

android:text="分 享 详 情："/>

<LinearLayout

android:id="@+id/edit\_detail\_info\_layout"

android:layout\_width="match\_parent"

android:layout\_height="250dp"

android:padding="10.0dip"

android:background="@drawable/edit\_layout\_border"

android:orientation="horizontal">

<EditText

android:id="@+id/et\_edit\_detail\_info"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="10.0dip"

android:inputType="textMultiLine"

android:background="@null"

android:hint="在这详细介绍趣事"/>

</LinearLayout>

<Button

android:id="@+id/edit\_sure\_share"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="100dp"

android:layout\_marginRight="100dp"

android:background="@drawable/selector\_loginactivity\_button"

android:text="发布"

android:textColor="#fff"

android:gravity="center"

/>

</LinearLayout>

</RelativeLayout>

----------------------------- **edit\_tab\_content .xml结束----------------------**

“我的”页——my\_tab\_content

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为my\_tab\_content。打开创建的my\_tab\_content.xml，将下面的布局文件代码复制到文件中。这个文件包含注册登录按钮、管理账号、更新签名、修改密码、我的分享、我的日记、我的收藏、关于“i分享”这几个部分。

---------------------------- **my\_tab\_content.xml开始-------------------------**

<?xml version="1.0" encoding="UTF-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/empty\_cart\_view"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ffefebe7" >

<LinearLayout

android:background="@color/white"

android:id="@+id/top\_relative"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ImageView

android:id="@+id/images\_head"

android:layout\_width="80dp"

android:layout\_height="80dp"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:scaleType="fitCenter"

android:src="@drawable/tab\_icon\_one"/>

<LinearLayout

android:id="@+id/login\_top"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="vertical"

android:weightSum="1"

android:visibility="gone">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:layout\_weight="0.3"

android:orientation="horizontal">

<TextView

android:id="@+id/textView8"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:text="昵称："

android:textSize="18sp"/>

<TextView

android:id="@+id/t\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="丫an梅"

android:textSize="18sp"

android:textStyle="bold"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.3"

android:orientation="horizontal">

<TextView

android:id="@+id/textView9"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:text="签名："

android:textSize="18sp"/>

<TextView

android:id="@+id/t\_sign"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.44"

android:text="人生的意义在于尝试和守护"

android:textSize="18sp"

android:textStyle="bold"/>

</LinearLayout>

</LinearLayout>

<LinearLayout

android:id="@+id/un\_login\_top"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="vertical"

android:weightSum="1"

android:visibility="gone">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:layout\_weight="0.3"

android:orientation="horizontal">

<Button

android:id="@+id/pop\_login"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dp"

android:background="@drawable/button\_radius"

android:text="登录"

android:gravity="center"

/>

<Button

android:id="@+id/pop\_register"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:background="@drawable/button\_radius"

android:text="注册"

android:gravity="center"

/>

</LinearLayout>

</LinearLayout>

</LinearLayout>

<ScrollView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/top\_relative"

android:fadingEdge="none" >

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical"

android:orientation="vertical"

android:paddingBottom="60.0dip"

android:paddingTop="10.0dip"

android:layout\_marginTop="10dp">

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10.0dip"

android:layout\_marginTop="10.0dip"

android:gravity="center\_vertical"

android:paddingLeft="10.0dip"

android:text="相关设置"

android:textColor="#ff888888"

android:textSize="16.0sp" />

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TableLayout

android:id="@+id/MorePageTableLayout\_Follow"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="1.0dip"

android:shrinkColumns="0"

android:stretchColumns="0" >

<TableRow

android:id="@+id/more\_page\_row1"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:id="@+id/test\_text\_view"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list1"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="账号管理"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

<TableRow

android:id="@+id/more\_page\_row2"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list2"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="更新签名"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

<TableRow

android:id="@+id/more\_page\_row3"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list3"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="修改密码"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

</TableLayout>

</LinearLayout>

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="10.0dip"

android:layout\_marginTop="10.0dip"

android:gravity="center\_vertical"

android:paddingLeft="10.0dip"

android:text="关于我的"

android:textColor="#ff888888"

android:textSize="16.0sp" />

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" >

<TableLayout

android:id="@+id/MorePageTableLayout\_Client"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="1.0dip"

android:shrinkColumns="0"

android:stretchColumns="0" >

<TableRow

android:id="@+id/more\_page\_row4"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list4"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="我的分享"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

<TableRow

android:id="@+id/more\_page\_row5"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list5"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="我的日记"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

<TableRow

android:id="@+id/more\_page\_row6"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list6"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="我的收藏"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

<TableRow

android:id="@+id/more\_page\_row7"

android:layout\_width="fill\_parent"

android:layout\_marginLeft="2.0dip"

android:layout\_marginRight="2.0dip"

android:background="@drawable/small\_bc"

android:paddingBottom="16.0dip"

android:paddingTop="16.0dip" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:drawableLeft="@drawable/my\_tab\_list7"

android:drawablePadding="10.0dip"

android:gravity="center\_vertical"

android:includeFontPadding="false"

android:paddingLeft="24.0dip"

android:text="关于 i 分享"

android:textColor="#ff333333"

android:textSize="16.0sp" />

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_gravity="right"

android:gravity="center\_vertical"

android:paddingRight="20.0dip"

android:src="@drawable/next\_right" />

</TableRow>

</TableLayout>

</LinearLayout>

</LinearLayout>

</ScrollView>

</RelativeLayout>

------------------------------ **my\_tab\_content.xml结束------------------------**

“注册”页——activity\_register

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_register。打开创建的activity\_register.xml，将下面的布局文件代码复制到文件中。这个文件包含用户名、登录密码、登录密码确认、签名这几个输入框，用于输入用户的注册信息。

----------------------- **activity\_register.xml开始--------------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#eeeeee"

>

<RelativeLayout

android:id="@+id/rl\_top"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:background="@color/color\_minefragment\_top"

>

<TextView

android:id="@+id/tv\_register"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:text="用户注册"

android:textColor="#fff"

android:textSize="20dp"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

/>

</RelativeLayout>

<LinearLayout

android:id="@+id/ll\_body"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_below="@+id/rl\_top"

android:layout\_marginTop="60dp"

android:layout\_marginLeft="50dp"

android:layout\_marginRight="50dp"

>

<!-- 第一个文本编辑框 输入用户名 -->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="用户名："/>

<EditText

android:id="@+id/et\_username"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:hint="请输入用户名"

android:gravity="center\_vertical"

android:layout\_marginLeft="10dp"

/>

</LinearLayout>

<!-- 第二个文本编辑框 输入密码 -->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_password1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="密 码："/>

<EditText

android:id="@+id/et\_password1"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:gravity="center\_vertical"

android:layout\_marginLeft="10dp"

android:inputType="textPassword"

android:hint="请输入密码" />

</LinearLayout>

<!-- 第三个文本编辑框 再次输入密码 -->

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_password2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="密 码："/>

<EditText

android:id="@+id/et\_password2"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:hint="请再次输入密码"

android:gravity="center\_vertical"

android:layout\_marginLeft="10dp"

android:inputType="textPassword"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_signature"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="签 名："/>

<EditText

android:id="@+id/et\_signature"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:hint="输入你的个性签名"

android:gravity="center\_vertical"

android:layout\_marginLeft="10dp"

/>

</LinearLayout>

<!-- 注册按钮 -->

<!-- <Button

android:id="@+id/bt\_register"

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:layout\_marginLeft="80dp"

android:layout\_marginRight="80dp"

android:background="@drawable/selector\_loginactivity\_button"

android:textColor="#fff"

android:text="注册"

android:layout\_marginTop="40dp"

/>-->

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:id="@+id/rl\_bottom"

android:layout\_marginTop="20dp"

android:layout\_marginBottom="30dp"

>

<TextView

android:id="@+id/test1\_view\_info"

android:layout\_centerInParent="true"

android:text="注册即代表同意《用户协议》"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</RelativeLayout>

</LinearLayout>

</RelativeLayout>

------------------- **activity\_register.xml结束--------------------------**

“登录”页——activity\_login

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_login。打开创建的activity\_login.xml，将下面的布局文件代码复制到文件中。这个文件包含用户名、登录密码这两个输入框，用于输入用户的登录信息。

------------------------- **activity\_login.xml结束--------------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#eeeeee">

<RelativeLayout

android:id="@+id/rl\_login\_top"

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:background="@color/color\_minefragment\_top"

>

<TextView

android:id="@+id/tv\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:text="用户登录"

android:textColor="#fff"

android:textSize="20dp"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

/>

</RelativeLayout>

<LinearLayout

android:id="@+id/ll\_login\_two"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_below="@+id/rl\_login\_top"

android:layout\_marginTop="120dp"

android:layout\_marginLeft="50dp"

android:layout\_marginRight="50dp"

>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_login\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="用户名："/>

<EditText

android:id="@+id/et\_login\_username"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="手机号/邮箱/用户名"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_login\_password"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="密 码："/>

<EditText

android:id="@+id/et\_login\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="登录密码"

android:inputType="textPassword"/>

</LinearLayout>

</LinearLayout>

<!--<Button

android:id="@+id/bt\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/ll\_login\_two"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="100dp"

android:layout\_marginRight="100dp"

android:background="@drawable/selector\_loginactivity\_button"

android:text="登录"

android:textColor="#fff"

android:gravity="center"

android:onClick="onClick"

/>-->

<TextView

android:id="@+id/tv\_login\_forget"

android:text="忘记密码?"

android:textColor="#f00"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginVertical="50dp"

android:layout\_below="@+id/ll\_login\_two"

android:layout\_alignLeft="@+id/ll\_login\_two"

/>

<TextView

android:id="@+id/tv\_logina\_check"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="还没有注册？"

android:textColor="#f00"

android:layout\_marginRight="10dp"

android:layout\_marginVertical="50dp"

android:layout\_below="@+id/ll\_login\_two"

android:layout\_alignRight="@+id/ll\_login\_two"

/>

<TextView

android:id="@+id/test2\_view\_info"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/tv\_login\_forget"

android:layout\_centerInParent="true"

android:layout\_marginVertical="30dp"

android:text="测试信息反馈"

android:textColor="#B3B3B3"

android:gravity="center"

/>

</RelativeLayout>

------------------------ **activity\_login. xml结束----------------------------**

“账号管理”页——activity\_account\_manage

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_account\_manage。打开创建的activity\_account\_manage.xml，将下面的布局文件代码复制到文件中。这个文件就实现一个功能——退出登录，在结构上该页面包含显示用户昵称和登录状态的Textview。

----------------------**activity\_account\_manage. xml开始---------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:background="#eeeeee">

<LinearLayout

android:background="@color/white"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<ImageView

android:id="@+id/account\_images\_head"

android:layout\_width="80dp"

android:layout\_height="80dp"

android:layout\_marginBottom="10dp"

android:layout\_marginLeft="10dp"

android:layout\_marginRight="10dp"

android:layout\_marginTop="10dp"

android:scaleType="fitCenter"

android:src="@drawable/tab\_icon\_one"/>

<LinearLayout

android:id="@+id/account\_login\_top"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_weight="1"

android:orientation="vertical"

android:weightSum="1">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:layout\_weight="0.3"

android:orientation="horizontal">

<TextView

android:id="@+id/account\_textView8"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:text="昵称："

android:textSize="18sp"/>

<TextView

android:id="@+id/account\_t\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="丫an梅"

android:textSize="18sp"

android:textStyle="bold"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.3"

android:orientation="horizontal">

<TextView

android:id="@+id/account\_textView9"

android:layout\_width="70dp"

android:layout\_height="wrap\_content"

android:text="状态："

android:textSize="18sp"/>

<TextView

android:id="@+id/account\_t\_status"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.44"

android:text="已登录"

android:textSize="18sp"

android:textStyle="bold"/>

</LinearLayout>

</LinearLayout>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:id="@+id/account\_bt\_out\_login"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="100dp"

android:layout\_marginRight="100dp"

android:background="@drawable/selector\_loginactivity\_button"

android:text="退出登录"

android:textColor="#fff"

android:gravity="center"

/>

</LinearLayout>

</LinearLayout>

--------------------------**activity\_account\_manage. xml结束------------------**

“更新签名”页——activity\_change\_signature

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_change\_signature。打开创建的activity\_change\_signature.xml，将下面的布局文件代码复制到文件中。这个文件包含旧签名的显示框、新签名的输入框和一个确认修改签名的按钮。

----------------------**activity\_change\_signature. xml开始---------------------**

<?xml version="1.0" encoding="utf-8"?>

<!--

这里把整个Activity\_login的布局设置成相对布局，

因为相对布局比较灵活

-->

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#eeeeee">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:background="@color/color\_minefragment\_top"

>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:text="更新签名"

android:textColor="#fff"

android:textSize="20dp"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

/>

</RelativeLayout>

<LinearLayout

android:id="@+id/sign\_change\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_marginTop="120dp"

android:layout\_marginLeft="50dp"

android:layout\_marginRight="50dp"

>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="旧 签 名："

android:textSize="@dimen/textSize\_normal"/>

<TextView

android:id="@+id/old\_user\_sign"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="12dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"

android:text="新 签 名："/>

<EditText

android:id="@+id/et\_login\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"

android:hint="这里填写新签名"/>

</LinearLayout>

</LinearLayout>

<Button

android:id="@+id/bt\_change\_signature"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/sign\_change\_layout"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="150dp"

android:layout\_marginRight="150dp"

android:background="@drawable/selector\_loginactivity\_button"

android:text="修改"

android:textColor="#fff"

android:gravity="center"

android:onClick="onClick"

/>

<TextView

android:id="@+id/changeSign\_view\_info"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginVertical="30dp"

android:text="信息反馈"

android:textColor="#B3B3B3"

android:gravity="center"

/>

</RelativeLayout>

--------------------**activity\_change\_signature. xml结束----------------------**

“更改密码”页——activity\_change\_password

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_change\_password。打开创建的activity\_change\_password.xml，将下面的布局文件代码复制到文件中。这个文件包含旧登录密码输入框和新密码输入框，以及一个确认修改密码的按钮。

-------------------**activity\_change\_password. xml开始----------------------**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#eeeeee">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="70dp"

android:background="@color/color\_minefragment\_top"

>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:text="修改密码"

android:textColor="#fff"

android:textSize="20dp"

android:layout\_centerVertical="true"

android:layout\_marginLeft="20dp"

/>

</RelativeLayout>

<LinearLayout

android:id="@+id/password\_change\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_marginTop="120dp"

android:layout\_marginLeft="50dp"

android:layout\_marginRight="50dp"

>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="旧 密 码："

android:textSize="@dimen/textSize\_normal"/>

<EditText

android:id="@+id/et\_old\_login\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"

android:inputType="textPassword"

android:hint="这里填写旧密码"/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_marginTop="12dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"

android:text="新 密 码："/>

<EditText

android:id="@+id/et\_new\_login\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="@dimen/textSize\_normal"

android:inputType="textPassword"

android:hint="这里填写新密码"/>

</LinearLayout>

</LinearLayout>

<Button

android:id="@+id/bt\_change\_password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/password\_change\_layout"

android:layout\_marginTop="20dp"

android:layout\_marginLeft="150dp"

android:layout\_marginRight="150dp"

android:background="@drawable/selector\_loginactivity\_button"

android:text="修改"

android:textColor="#fff"

android:gravity="center"

android:onClick="onClick"

/>

<TextView

android:id="@+id/bt\_changePassword\_view\_info"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginVertical="30dp"

android:text="信息反馈"

android:textColor="#B3B3B3"

android:gravity="center"

/>

</RelativeLayout>

-------------------- **activity\_change\_password. xml结束--------------------**

“关于我的”详情页——activity\_about\_me

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_about\_me。打开创建的activity\_about\_me.xml，将下面的布局文件代码复制到文件中。这个文件就只是包含一个ListView用于展示查询“我的分享”、“我的日记”、“我的收藏”返回的内容。

------------------- **activity\_about\_me. xml开始-----------------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:paddingLeft="15dp"

android:paddingRight="15dp"

android:orientation="vertical"

android:paddingTop="20dp"

android:paddingBottom="20dp">

<ListView

android:id="@+id/my\_share\_listView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:dividerHeight="15dp" />

</LinearLayout>

</LinearLayout>

----------------------------- **activity\_about\_me. xml结束-----------------------**

“关于i分享”——activity\_about\_ishare

打开项目Discover，找到app>src>main>res>layout，在layout下创建Layout XML File，命名为activity\_about\_ishare。打开创建的activity\_about\_ishare.xml，将下面的布局文件代码复制到文件中。这个页面布局比较简单，就是关于i分享相关信息的介绍。

------------------------- **activity\_about\_ishare. xml开始-----------------**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<ScrollView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:fadingEdge="none" >

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:paddingLeft="14dp"

android:paddingRight="14dp"

android:paddingBottom="60dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="22dp"

android:textStyle="bold"

android:text="关于i分享" />

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:textSize="18sp"

android:layout\_weight="1"

android:textColor="#000000"

android:text="开发者学号：\n

201611671208\n

201611671206\n

201611671221\n

" />

</LinearLayout>

<TextView

android:background="@drawable/list\_card\_style"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/margin\_10"

android:padding="10dp"

android:textSize="18sp"

android:textColor="#000000"

android:text="iShare应用：发现有趣的人和事，在这个平台可以发布自己感兴趣的内容以及查看大家的分享，简历良好的交流氛围"/>

<TextView

android:id="@+id/detail\_info\_detail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/margin\_10"

android:padding="10dp"

android:textSize="18sp"

android:textColor="#000000"

android:singleLine="false"

android:text=" 开发目的：Android课程设计\n\n

开发时间：2019年5月\n\n

素材来源：LOGO是网络资源图，Icon来自www.iconfont.cn，其中用于测试的图片均来自网络\n\n

项目代码：部分结构参考自CSDN、GitHub\n\n

实现功能：起始页、注册、登录、更新签名、更改密码、浏览分享、按标题和简述关键字查找分享的内容、根据用户昵称查找用户、查看分享详情、点赞、收藏、发布四种类型的内容、查看按点赞数排行前十的内容、删除发布的内容、评论、查看我的分享、查看我的日记、查看我的收藏" />

</LinearLayout>

</ScrollView>

</LinearLayout>

------------------------ **activity\_about\_ishare. xml结束----------------------**

对象类

打开项目Discover，找到app>src>main>java右击选择New，点击Package，输入包名com.example.bean，点击ok。

关于内容——FindInfo.java

打开项目Discover，找到app>src>main>java>com.example>bean，在bean中创建Class，命名为FindInfo，将下面的代码复制到该文件中。FindInfo.java应该有内容ID、内容标题、内容简述、内容详情、类型（0表示日记，1表示趣事）、点赞数、作者这几个属性，以及对这些属性的获取和设置的方法。

------------------------------ **FindInfo.java开始---------------------------**

package com.example.bean;

public class FindInfo {

Integer infoId; //内容ID

String infoTitle; //内容标题

String infoDescribe; //内容简述

String infoDetail; //内容详情

Integer type; //类型：0表示日记，1表示趣事

Integer support; //点赞数

String infoAuthor; //作者

public Integer getInfoId() {

return infoId;

}

public void setInfoId(Integer infoId) {

this.infoId = infoId;

}

public String getInfoTitle() {

return infoTitle;

}

public void setInfoTitle(String infoTitle) {

this.infoTitle = infoTitle;

}

public String getInfoDescribe() {

return infoDescribe;

}

public void setInfoDescribe(String infoDescribe) {

this.infoDescribe = infoDescribe;

}

public String getInfoDetail() { return infoDetail; }

public void setInfoDetail(String infoDetail) {

this.infoDetail = infoDetail;

}

public Integer getType() { return type; }

public void setType(Integer type) { this.type = type; }

public Integer getSupport() { return support; }

public void setSupport(Integer support) { this.support = support; }

public String getInfoAuthor() {

return infoAuthor;

}

public void setInfoAuthor(String infoAuthor) { this.infoAuthor = infoAuthor; }

}

------------------------- **FindInfo.java结束------------------------------**

关于评论——CommentInfo.java

打开项目Discover，找到app>src>main>java>com.example>bean，在bean中创建Class，命名为CommentInfo，将下面的代码复制到该文件中。CommentInfo.java应该有评论Id、文章ID、评论者、评论详情这几个属性、评论详情，以及对这些属性的获取和设置的方法。

-------------------------------**CommentInfo.java开始-----------------------**

package com.example.bean;

public class CommentInfo {

Integer commentId; //评论Id

Integer infoId; //文章ID

String commentUser; //评论者

String commentDetail; //评论详情

public Integer getCommentId() {

return commentId;

}

public void setCommentId(Integer commentId) {

this.commentId = commentId;

}

public Integer getInfoId() {

return infoId;

}

public void setInfoId(Integer infoId) {

this.infoId = infoId;

}

public String getCommentUser() {

return commentUser;

}

public void setCommentUser(String commentUser) {

this.commentUser = commentUser;

}

public String getCommentDetail() {

return commentDetail;

}

public void setCommentDetail(String commentDetail) {

this.commentDetail = commentDetail;

}

}

--------------------------**CommentInfo.java结束-------------------------------**

关于用户——FindPeople.java

打开项目Discover，找到app>src>main>java>com.example>bean，在bean中创建Class，命名为FindPeople.，将下面的代码复制到该文件中。FindPeople..java应该有用户昵称、内容详情、用户头像、用户密码这几个属性，以及对这些属性的获取和设置的方法。

------------------------------ **FindPeople.java开始----------------------------**

package com.example.bean;

public class FindPeople {

String userName; //用户昵称

String signature; //用户签名

String userLogImage; //用户头像

// String passWord; //用户密码

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public String getSignature() {

return signature;

}

public void setSignature(String signature) {

this.signature = signature;

}

public String getUserLogImage() {

return userLogImage;

}

public void setUserLogImage(String userLogImage) {

this.userLogImage = userLogImage;

}

/\*public String getPassWord() { return passWord; }

public void setPassWord(String passWord) {

this.passWord = passWord;

}\*/

}

---------------------- **FindPeople.java结束----------------------------------**

工具类

打开项目Discover，找到app>src>main>java右击选择New，点击Package，输入包名com.example.discover，点击ok。

通过链接获取图片资源——LoadImgByNet.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为LoadImgByNet，将下面的代码复制到该文件中。LoadImgByNet.java 实现的方法是通过传过来的图片URL获取到网络的图片资源，并返回Bitmap类型的资源。总的来说这个文件就是为了能在代码中将网络资源图设置为组件背景或着ImageView显示的图片。

--------------------------**LoadImgByNet.java开始----------------------------**

package com.example.discover;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import java.io.IOException;

import java.io.InputStream;

import java.net.HttpURLConnection;

import java.net.URL;

public class LoadImgByNet {

public static Bitmap getBitmap(String path) throws IOException {

URL url = new URL(path);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setConnectTimeout(5000);

conn.setRequestMethod("GET");

if (conn.getResponseCode() == 200){

InputStream inputStream = conn.getInputStream();

Bitmap bitmap = BitmapFactory.decodeStream(inputStream);

return bitmap;

}

return null;

}

}

-------------------------**LoadImgByNet.java结束--------------------------------**

页面逻辑处理类

“启动页”—— SplashActivity.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为SplashActivity，将下面的代码复制到该文件中。SplashActivity.java主要的介绍如下：

* 设置标题栏隐藏，达到启动页全屏的效果
* 判断网络权限是否开启
* 网络可用状态下，设置显示启动页且透明度从0变化到1，达到淡入的效果
* 设置延时两秒后跳转到首页MainActivity .class

------------------------- **SplashActivity.java开始----------------------------**

package com.example.discover;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.net.ConnectivityManager;

import android.net.NetworkInfo;

import android.os.Bundle;

import android.os.Handler;

import android.view.animation.AlphaAnimation;

import android.widget.LinearLayout;

//启动页

public class SplashActivity extends Activity {

private LinearLayout mLinearLayout;

//重写onCreate

protected void onCreate(Bundle savedInstanceState) {

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

//在配置文件设置 android:theme="@android:style/Theme.NoTitleBar"来隐藏标题，避免出现title闪屏的效果

setContentView(R.layout.splash);

mLinearLayout = (LinearLayout)this.findViewById(R.id.LinearLayout01);

//判端当前网络是否可用,可用则进入主页

if(isNetWorkConnetcted()){

//设置进入起始页的动画

AlphaAnimation aa = new AlphaAnimation(0.0f,1.0f); //设置透明度从0变化到1

aa.setDuration(2000); //设置动画时长为2秒

mLinearLayout.setAnimation(aa); //将动画设置到起始页

mLinearLayout.startAnimation(aa); //启动动画

//通过handler设置延时两秒后执行r任务

new Handler().postDelayed(new LoadMainTask(),2000);

}else{

showSetNetworkDialog();

}

}

//启动线程,加载主页

private class LoadMainTask implements Runnable{

@Override

public void run() {

Intent intent = new Intent(SplashActivity.this,MainActivity.class);

startActivity(intent); //打开主页

finish(); //关闭起始页

}

}

/\*判端网络状态\*/

private boolean isNetWorkConnetcted(){

ConnectivityManager manager = (ConnectivityManager)getSystemService(CONNECTIVITY\_SERVICE); //进行类型转换

NetworkInfo info = manager.getActiveNetworkInfo(); //获取所有可用的网络状态信息

//info.isConnected(); 检查网络是否连接

boolean result;

if(info!=null&&info.isConnected()){

result = true;

}else{

result = false;

}

return result;

}

//弹出网络不可用的对话框，课本94页

private void showSetNetworkDialog(){

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("网络设置");

builder.setMessage("网络错误，请检查网络状态");

builder.setPositiveButton("设置网络",new okClick());

builder.setPositiveButton("取消",new cleanClick());

builder.create().show();

}

//普通对话框的确定，点击后展示设置页面

class okClick implements DialogInterface.OnClickListener{

public void onClick(DialogInterface dialog, int which){

//激活系统组件

Intent intent = new Intent();

intent.setClassName("com.android.settings","com.android.settings.WirelessSetting"); //参数：激活activity的包名、类名

startActivity(intent);

finish();

}

}

//关闭

class cleanClick implements DialogInterface.OnClickListener{

public void onClick(DialogInterface dialog, int which){

finish(); //关闭应用

}

}

}

--------------------------- **SplashActivity.java结束---------------------------**

“主页”—— MainActivity.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为MainActivity，将下面的代码复制到该文件中。MainActivity.java主要的介绍如下：

* 显示main\_tab.xml，给布局中的四个Tab添加点击事件，控制四个Fragment显示状态
* 创建方法setTabSelection，控制Tab选中和未被选中时文字与图表的显示状态
* 设置页面启动时选中第0个Tab，即显示“发现页”的内容

----------------------- **MainActivity.java开始-------------------------**

package com.example.discover;

import android.graphics.Color;

import android.os.Bundle;

import android.support.v4.app.FragmentActivity;

import android.support.v4.app.FragmentManager;

import android.support.v4.app.FragmentTransaction;

import android.view.View;

import android.view.Window;

import android.widget.ImageView;

import android.widget.TextView;

public class MainActivity extends FragmentActivity implements View.OnClickListener {

private MyTabFragment myFragment; //用于展示我的Fragment

private DiscoverTabFragment discoverFragment; //用于展示发现的Fragment

private FindTabFragment findFragment; //用于展示查找的Fragment

private EditTabFragment moreFragment; //用于展示发布的Fragment

private View myLayout; //我的布局

private View discoverLayout; //发现界面布局

private View findLayout; //查找界面布局

private View editLayout; //发布界面布局

private ImageView myImage; //在Tab布局上显示我的图标的控件

private ImageView discoverImage; //在Tab布局上显示发现图标的控件

private ImageView findImage; //在Tab布局上显示查找图标的控件

private ImageView moreImage; //在Tab布局上显示更多图标的控件

private TextView myText; //在Tab布局上显示我的标题的控件

private TextView discoverText; //在Tab布局上显示发现标题的控件

private TextView findText; //在Tab布局上显示查找标题的控件

private TextView moreText; //在Tab布局上显示更多标题的控件

/\*\*

\* 用于对Fragment进行管理,事务

\*/

private FragmentTransaction ftr;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

setContentView(R.layout.main\_tab);

// 初始化布局元素

initViews();

// 第一次启动时选中第0个tab

setTabSelection(0);

}

/\*\*

\* 在这里获取到每个需要用到的控件的实例，并给它们设置好必要的点击事件。

\*/

private void initViews() {

myLayout = findViewById(R.id.my\_layout);

discoverLayout = findViewById(R.id.discover\_layout);

findLayout = findViewById(R.id.find\_layout);

editLayout = findViewById(R.id.edit\_layout);

myImage = (ImageView) findViewById(R.id.my\_image);

discoverImage = (ImageView) findViewById(R.id.discover\_image);

findImage = (ImageView) findViewById(R.id.find\_image);

moreImage = (ImageView) findViewById(R.id.more\_image);

myText = (TextView) findViewById(R.id.my\_text);

discoverText = (TextView) findViewById(R.id.discover\_text);

findText = (TextView) findViewById(R.id.find\_text);

moreText = (TextView) findViewById(R.id.more\_text);

discoverLayout.setOnClickListener(this);

findLayout.setOnClickListener(this);

editLayout.setOnClickListener(this);

myLayout.setOnClickListener(this);

}

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.discover\_layout:

// 当点击了发现tab时，选中第1个tab

setTabSelection(0);

break;

case R.id.find\_layout:

// 当点击了查找tab时，选中第2个tab

setTabSelection(1);

break;

case R.id.edit\_layout:

// 当点击了发布tab时，选中第3个tab

setTabSelection(2);

break;

case R.id.my\_layout:

// 当点击了我的tab时，选中第4个tab

setTabSelection(3);

break;

default:

break;

}

}

/\*\*

\* 根据传入的index参数来设置选中的tab页。

\*

\* @param index

\* 每个tab页对应的下标0表示发现，1表示查找，2表示发布，3表示我的。

\*/

private void setTabSelection(int index) {

// 每次选中之前先清楚掉上次的选中状态

clearSelection();

// 开启一个Fragment事务

FragmentManager fm = getSupportFragmentManager();

ftr = fm.beginTransaction();//开启一个事务

// 先隐藏掉所有的Fragment，以防止有多个Fragment显示在界面上的情况

hideFragments(ftr);//自定义一个方法，来隐藏所有的fragment

switch (index) {

case 0:

// 当点击了发现tab时，改变控件的图片和文字颜色

discoverImage.setImageResource(R.drawable.tab\_icon\_two);

discoverText.setTextColor(Color.GREEN);

if (discoverFragment == null) {

// 如果discoverFragment为空，则创建一个并添加到界面上

discoverFragment = new DiscoverTabFragment();

ftr.add(R.id.content, discoverFragment);

} else {

// 如果discoverFragment不为空，则直接将它显示出来

ftr.show(discoverFragment);

}

break;

case 1:

// 当点击了查找tab时，改变控件的图片和文字颜色

findImage.setImageResource(R.drawable.tab\_icon\_three);

findText.setTextColor(Color.GREEN);

if (findFragment == null) {

// 如果findFragment为空，则创建一个并添加到界面上

findFragment = new FindTabFragment();

ftr.add(R.id.content, findFragment);

} else {

// 如果findFragment不为空，则直接将它显示出来

ftr.show(findFragment);

}

break;

case 2:

default:

// 当点击了发布tab时，改变控件的图片和文字颜色

moreImage.setImageResource(R.drawable.tab\_icon\_four);

moreText.setTextColor(Color.GREEN);

if (moreFragment == null) {

// 如果moreFragment为空，则创建一个并添加到界面上

moreFragment = new EditTabFragment();

ftr.add(R.id.content, moreFragment);

} else {

// 如果moreFragment不为空，则直接将它显示出来

ftr.show(moreFragment);

}

break;

case 3:

// 当点击了我的tab时，改变控件的图片和文字颜色

myImage.setImageResource(R.drawable.tab\_icon\_one);

myText.setTextColor(Color.GREEN);

if (myFragment == null) {

// 如果myFragment为空，则创建一个并添加到界面上

myFragment = new MyTabFragment();

ftr.add(R.id.content, myFragment);

} else {

// 如果myFragment不为空，则直接将它显示出来

ftr.show(myFragment);

}

break;

}

ftr.commit();

}

/\*\*

\* 清除掉所有的选中状态。

\*/

private void clearSelection() {

myImage.setImageResource(R.drawable.tab\_icon\_one\_un\_select);

myText.setTextColor(Color.parseColor("#82858b"));

discoverImage.setImageResource(R.drawable.tab\_icon\_two\_un\_select);

discoverText.setTextColor(Color.parseColor("#82858b"));

findImage.setImageResource(R.drawable.tab\_icon\_three\_un\_select);

findText.setTextColor(Color.parseColor("#82858b"));

moreImage.setImageResource(R.drawable.tab\_icon\_four\_un\_select);

moreText.setTextColor(Color.parseColor("#82858b"));

}

/\*\*

\* 将所有的Fragment都置为隐藏状态。

\*

\* @param transaction

\* 用于对Fragment执行操作的事务

\*/

private void hideFragments(FragmentTransaction transaction) {

if (myFragment != null) {

transaction.hide(myFragment);

}

if (discoverFragment != null) {

transaction.hide(discoverFragment);

}

if (findFragment != null) {

transaction.hide(findFragment);

}

if (moreFragment != null) {

transaction.hide(moreFragment);

}

}

}

------------------------- **MainActivity.java结束-----------------------------**

“发现”页——DiscoverTabFragment.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为DiscoverTabFragment，将下面的代码复制到该文件中。DiscoverTabFragment.java主要的介绍如下：

* 创建方法loadDataFromDataBase()，实现从数据表info中获取所有内容
* 创建方法initSetDataToBean解析返回的Json数据，并将内容保存到FindInfo对象中
* 创建类PaginationAdapter实现将一组数据传到ListView等UI显示组件
* 创建方法initializeAdapter()初始化ListView的适配器，即调用上述创建的方法和类达到将数据库数据在页面展示的效果
* 在onActivityCreated中给页面添加“加载更多”的按钮，以及对应的事件；另外给listView添加事件监听，实现item点击跳转的功能

--------------------------- **DiscoverTabFragment.java开始---------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.Handler;

import android.os.StrictMode;

import android.support.v4.app.Fragment;

import android.util.Log;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AbsListView;

import android.widget.AdapterView;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.bean.FindInfo;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import java.util.concurrent.ScheduledExecutorService;

public class DiscoverTabFragment extends Fragment implements AbsListView.OnScrollListener{

Activity mActivity; //存放当前的activity

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

private TextView title;

ImageButton back;

private DiscoverTabFragment.PaginationAdapter adapterAction;

private ScheduledExecutorService scheduledExecutorService;

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View discoverLayout = inflater.inflate(R.layout.discover\_tab\_content, container, false);

return discoverLayout;

}

private ListView listView;

private int page = 1; //请求页数

private int count = 10; //每次请求的数量

private int visibleLastIndex = 0; //最后的可视项索引

private int visibleItemCount; // 当前窗口可见项总数

private int dataSize = 28; //数据集的条数

private DiscoverTabFragment.PaginationAdapter adapter;

private View loadMoreView;

private Button loadMoreButton;

private Handler handler = new Handler();

//返回测试信息

/\*TextView testTxt;\*/

public void refreshDiscover(){

page = 1;

visibleLastIndex = 0;

listView.setAdapter(null); //将ListView控件的内容清空

initializeAdapter();

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();

}

/\*\* Called when the activity is first created. \*/

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

mActivity = this.getActivity();

/\*setContentView(R.layout.main);\*/

loadMoreView = getLayoutInflater().inflate(R.layout.activity\_find\_loadmore, null);

loadMoreButton = (Button)loadMoreView.findViewById(R.id.loadMoreButton);

loadMoreButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

loadMoreButton.setText("正在加载中..."); //设置按钮文字

handler.postDelayed(new Runnable() {

@Override

public void run() {

loadMoreData();

adapter.notifyDataSetChanged();

loadMoreButton.setText("查看更多..."); //恢复按钮文字

}

},2000);

}

});

//返回测试信息

/\* testTxt = (TextView)getActivity().findViewById(R.id.test\_discover\_tab);\*/

back = (ImageButton)getActivity().findViewById(R.id.back\_button);

back.setOnClickListener(new mOnClick());

listView = (ListView)getActivity().findViewById(R.id.discover\_listView);

listView.addFooterView(loadMoreView); //设置列表底部视图

initializeAdapter();

listView.setAdapter(adapter);

listView.setOnScrollListener(this);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

TextView getTvId = (TextView)view.findViewById(R.id.cardId);

/\*Toast.makeText(mActivity,"点击了"+getTvId.getText().toString()+"id", Toast.LENGTH\_SHORT).show();\*/

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

Boolean isLogin = sp.getBoolean("isLogin",false);

if(isLogin){

//点击了进入详情

Intent display\_info\_intent = new Intent();

display\_info\_intent.setClass(mActivity, DisplayDetail.class); //创建intent对象，并制定跳转页面

display\_info\_intent.putExtra("infoId",getTvId.getText().toString());

mActivity.setResult(1, display\_info\_intent); //这里的1就对应到onActivityResult（）方法中的resultCode

startActivity(display\_info\_intent); //跳转到详情页面

}else{

//尚未登录

Toast.makeText(getActivity(),"请先登录", Toast.LENGTH\_LONG).show();

}

}

});

}

class mOnClick implements View.OnClickListener {

@Override

public void onClick(View v) {

mActivity.finish();

}

}

@Override

public void onScrollStateChanged(AbsListView view, int scrollState) {

int itemsLastIndex = adapter.getCount()-1; //数据集最后一项的索引

int lastIndex = itemsLastIndex + 1;

if (scrollState == AbsListView.OnScrollListener.SCROLL\_STATE\_IDLE

&& visibleLastIndex == lastIndex) {

// 如果是自动加载,可以在这里放置异步加载数据的代码

}

}

@Override

public void onHiddenChanged(boolean hidden) {

super.onHiddenChanged(hidden);

if(hidden){

//TODO now visible to user

} else {

refreshDiscover();

}

}

@Override

public void onScroll(AbsListView view, int firstVisibleItem,

int visibleItemCount, int totalItemCount) {

this.visibleItemCount = visibleItemCount;

visibleLastIndex = firstVisibleItem + visibleItemCount - 1; //最后的可视项索引

Log.e("firstVisibleItem = ",firstVisibleItem+"");

Log.e("visibleItemCount = ",visibleItemCount+"");

Log.e("totalItemCount = ",totalItemCount+"");

//如果所有的记录选项等于数据集的条数，则移除列表底部视图

if(totalItemCount == dataSize+1){

listView.removeFooterView(loadMoreView);

Toast.makeText(getActivity(), "数据全部加载完!", Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 查询数据库中的数据

\*/

private JSONArray loadDataFromDataBase(){

//Toast.makeText(getActivity(),"保存",Toast.LENGTH\_SHORT).show();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

try {

String discoverUrl = baseUrl+"QueryDiscover?page="+page+"&count="+count;

URL url = new URL(discoverUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

if(stringBuilder.toString()!="null") {

//返回测试信息

JSONArray jsonArray = new JSONArray(stringBuilder.toString());

/\* testTxt.setText(baseUrl+"QueryDiscover?page="+page+"&count="+count);\*/

//获取到的数据，对Json进行解析

page = page + 1; //一次成功请求后更新请求页面

buffer.close();

return jsonArray;

}else{

return null;

}

}else{

Toast.makeText(getActivity(),"非200", Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(getActivity(), "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

return null;

}

//初始化将详情设置到FindInfo bean中

public List<FindInfo> initSetDataToBean(String detail){

List<FindInfo> findInfo = new ArrayList<FindInfo>();

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindInfo items = new FindInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

Integer infoId = temp.getInt("infoId"); //内容ID

String infoTitle = temp.getString("infoTitle"); //内容标题

String infoDescribe = temp.getString("infoDescribe"); //内容简述

String infoDetail = temp.getString("infoDetail"); //内容详情

Integer type = temp.getInt("infoType"); //类型：0表示日记，1表示趣事

Integer support = temp.getInt("infoSupport"); //点赞数

String infoAuthor = temp.getString("infoAuthor"); //作者

items.setInfoId(infoId);

items.setInfoTitle(infoTitle);

items.setInfoDescribe(infoDescribe);

items.setInfoDetail(infoDetail);

items.setType(type);

items.setSupport(support);

items.setInfoAuthor(infoAuthor);

findInfo.add(items);

}

return findInfo;

}catch (JSONException e){

Toast.makeText(getActivity(), "initSetDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

return null;

}

}

//加载更多将详情设置到FindInfo bean中

public void loadMoreSetDataToBean(String detail){

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindInfo items = new FindInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

Integer infoId = temp.getInt("infoId"); //内容ID

String infoTitle = temp.getString("infoTitle"); //内容标题

String infoDescribe = temp.getString("infoDescribe"); //内容简述

String infoDetail = temp.getString("infoDetail"); //内容详情

Integer type = temp.getInt("infoType"); //类型：0表示日记，1表示趣事

Integer support = temp.getInt("infoSupport"); //点赞数

String infoAuthor = temp.getString("infoAuthor"); //作者

items.setInfoId(infoId);

items.setInfoTitle(infoTitle);

items.setInfoDescribe(infoDescribe);

items.setInfoDetail(infoDetail);

items.setType(type);

items.setSupport(support);

items.setInfoAuthor(infoAuthor);

adapter.addNewsItem(items); //与初始化是有差异的

}

}catch (JSONException e){

Toast.makeText(getActivity(), "loadMoreSetDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 初始化ListView的适配器，即打开页面展示的数据

\*/

private void initializeAdapter(){

// 设置线程策略

setVersion();

JSONArray jsonArray = loadDataFromDataBase();

if(jsonArray!=null){

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail= totalObject.getString("RecordDetail"); //详情

if(initSetDataToBean(detail)!=null) {

adapter = new PaginationAdapter(initSetDataToBean(detail));

}

}catch (JSONException e){

Toast.makeText(getActivity(), "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

listView.removeFooterView(loadMoreView);

Toast.makeText(mActivity, "查询为空" , Toast.LENGTH\_LONG).show();

}

}

//APP如果在主线程中请求网络操作，将会抛出异常，所以需要用线程来操作网络请求

void setVersion() {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()

.detectDiskReads()

.detectDiskWrites()

.detectNetwork()

.penaltyLog()

.build());

StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()

.detectLeakedSqlLiteObjects() //探测SQLite数据库操作

.penaltyLog() //打印logcat

.penaltyDeath()

.build());

}

/\*\*

\* 加载更多数据

\*/

private void loadMoreData(){

JSONArray jsonArray = loadDataFromDataBase(); //从服务器获取资源

try {

JSONObject total = (JSONObject) jsonArray.get(0);

dataSize = total.getInt("totalRecord"); //总记录数

String detail= total.getString("RecordDetail"); //详情

loadMoreSetDataToBean(detail); //将更多的详情设置到bean中

}catch (JSONException e){

Toast.makeText(getActivity(), "loadMoreData()异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 将一组数据传到ListView等UI显示组件

\*/

class PaginationAdapter extends BaseAdapter {

List<FindInfo> newsItems;

public PaginationAdapter(List<FindInfo> newsitems){

this.newsItems = newsitems;

}

@Override

public int getCount() {

return newsItems.size();

}

@Override

public Object getItem(int position) {

return newsItems.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//在这里将Item设置到每个卡片

@Override

public View getView(int position, View view, ViewGroup parent) {

if(view == null){

view = getLayoutInflater().inflate(R.layout.activity\_find\_list\_item, null);

}

//标题

TextView tvTitle = (TextView)view.findViewById(R.id.cardTitle);

tvTitle.setText(newsItems.get(position).getInfoTitle());

//文章ID

TextView tvId = (TextView)view.findViewById(R.id.cardId);

tvId.setText(newsItems.get(position).getInfoId().toString());

int type = newsItems.get(position).getType();

if(type==0||type==3){

ImageView cardImgTip = (ImageView)view.findViewById(R.id.cardImgTip);

if(type==0){

cardImgTip.setImageResource(R.drawable.share\_funny\_select);

}else{

cardImgTip.setImageResource(R.drawable.share\_diary\_select);

}

TextView tvContent = (TextView)view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.VISIBLE); //显示文字框

ImageView ivContent = (ImageView)view.findViewById(R.id.cardContent\_pic);

ivContent.setVisibility(View.GONE); //设置图片显示框隐藏

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

layoutMusic.setVisibility(View.GONE); //设置音乐显示框隐藏

//分享的普通内容

tvContent.setText(newsItems.get(position).getInfoDescribe());

}else if(type==1){

ImageView cardImgTip = (ImageView)view.findViewById(R.id.cardImgTip);

cardImgTip.setImageResource(R.drawable.share\_pic\_select);

//分享图片

TextView tvContent = (TextView)view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.GONE); //隐藏文字框

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

layoutMusic.setVisibility(View.GONE); //设置音乐显示框隐藏

ImageView ivContent = (ImageView)view.findViewById(R.id.cardContent\_pic);

//通过网络链接获取图片

Bitmap one;

String describeUrl = newsItems.get(position).getInfoDescribe().trim();

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+describeUrl); //设置图片

ivContent.setImageBitmap(one);

}catch(IOException e){

e.printStackTrace();

}

ivContent.setVisibility(View.VISIBLE); //设置图片显示框

}else if(type==2){

ImageView cardImgTip = (ImageView)view.findViewById(R.id.cardImgTip);

cardImgTip.setImageResource(R.drawable.share\_music\_select);

//分享的音乐

TextView tvContent = (TextView)view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.GONE); //显示文字框

ImageView ivContent = (ImageView)view.findViewById(R.id.cardContent\_pic);

ivContent.setVisibility(View.GONE); //设置图片显示框隐藏

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

TextView tvMusicContent = (TextView)view.findViewById(R.id.cardTitle\_music);

tvMusicContent.setText(newsItems.get(position).getInfoDescribe());

layoutMusic.setVisibility(View.VISIBLE); //设置音乐显示框显示

}

return view;

}

/\*\*

\* 添加数据列表项

\* @param infoItem

\*/

public void addNewsItem(FindInfo infoItem){

newsItems.add(infoItem);

}

}

}

----------------------**DiscoverTabFragment.java结束-------------------------**

“详情”页—— DisplayDetail.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为DisplayDetail，将下面的代码复制到该文件中。DisplayDetail.java主要的介绍如下：

* 显示activity\_info\_detail.xml，获取到跳转源传递的参数——内容ID，用内容ID请求对应内容的详细信息设置到页面显示
* 给页面的点赞、删除、收藏、发布评论的按钮添加点击事件
* 点赞时则将请求后台给对应内容点赞数加一
* 根据返回的作者和登录用户昵称对比结果，判断登录用户是否为作者，当登录用户为作者时则可以看到删除按钮键并可执行删除文章的功能，反之则不能。
* 根据内容ID请求后台查询用户是否已经收藏文章，收藏则显示已经收藏的按钮状态且可以执行取消收藏的功能；反之未收藏，则显示未收藏的状态且可以执行收藏功能。
* 根据内容ID请求后台对用的评论内容，利用适配器将内容设置到页面显示
* 点击发布按钮时则将用户昵称和评论详情发到服务器，请求成功后刷新适配器，实现评论功能

-------------------------------- **DisplayDetail.java开始-----------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.graphics.drawable.BitmapDrawable;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.view.ViewGroup;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.FrameLayout;

import android.widget.ImageButton;

import android.widget.ListAdapter;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.bean.CommentInfo;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

public class DisplayDetail extends Activity{

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

TextView tv\_info\_author; //内容作者

TextView tv\_info\_title; //内容标题

TextView tv\_info\_describe; //内容简述

TextView tv\_info\_detail; //内容详情

TextView tv\_info\_support; //内容点赞数

ImageButton btn\_info\_like; //内容收藏

ImageButton btn\_info\_delete; //内容删除

EditText et\_comment\_detail; //评论的内容

Button comment\_btn; //评论按钮

TextView share\_img\_title; //图片背景上的文字

FrameLayout share\_detail\_img; //分享的图片背景

String infoId;

String infoTitle; //内容标题

String infoDescribe; //内容简述

String infoDetail; //内容详情

Integer type; //类型：0表示日记，1表示趣事

Integer support; //点赞数

String infoAuthor; //作者

Boolean isFocus = false; //是否收藏

int focusId; //收藏ID

Boolean isMyInfo = false;

private DisplayDetail.CommentAdapter commentAdapter; //存放评论

private int dataSize = 28; //数据集的条数

private ListView listView;

protected void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_info\_detail);

tv\_info\_author = (TextView)findViewById(R.id.findDetail\_info\_author); //文章作者

tv\_info\_title = (TextView)findViewById(R.id.findDetail\_info\_title); //文章标题

tv\_info\_describe = (TextView)findViewById(R.id.findDetail\_info\_describe); //文章简述

tv\_info\_detail = (TextView)findViewById(R.id.detail\_info\_detail); //文章详细内容

tv\_info\_support = (TextView)findViewById(R.id.detail\_info\_support); //文章点赞数

btn\_info\_like = (ImageButton)findViewById(R.id.detail\_info\_focus); //文章收藏

btn\_info\_delete = (ImageButton)findViewById(R.id.detail\_info\_delete); //文章收藏

et\_comment\_detail = (EditText) findViewById(R.id.et\_comment\_detail); //评论的内容

comment\_btn = (Button)findViewById(R.id.comment\_btn); //评论按钮

share\_img\_title = (TextView)findViewById(R.id.share\_img\_title); //图片背景的文字

share\_detail\_img = (FrameLayout)findViewById(R.id.share\_detail\_img); //图片背景的文字

//取得启动该Activity的Intent对象

Intent intent =getIntent();

//取出Intent中附加的数据

if(intent.getStringExtra("infoId")!=null){

infoId =intent.getStringExtra("infoId");

initialize(infoId); //设置页面内容

initializeFocus(infoId); //改变收藏状态

tv\_info\_support.setOnClickListener(new supportClick());

btn\_info\_like.setOnClickListener(new focusClick());

btn\_info\_delete.setOnClickListener(new deleteClick());

listView = (ListView)findViewById(R.id.comment\_listView);

initializeAdapter(); //初始化，填入评论

listView.setAdapter(commentAdapter);

setListViewHeightBasedOnChildren(listView);

comment\_btn.setOnClickListener(new commentClick());

}

}

/\*\*

\* 查询数据库中的数据

\*/

private JSONArray loadData(String QueryInfoUrl){

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

try {

URL url = new URL(QueryInfoUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

//返回测试信息

JSONArray jsonArray = new JSONArray(stringBuilder.toString());

//获取到的数据，对Json进行解析

buffer.close();

return jsonArray;

}else{

Toast.makeText(DisplayDetail.this,"非200", Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(DisplayDetail.this, "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

return null;

}

/\*\*

\* 初始化ListView的适配器，即打开页面展示的评论

\*/

private void initializeAdapter(){

String QueryCommentUrl = baseUrl+"QueryComment?infoId="+infoId;

JSONArray jsonArray = loadData(QueryCommentUrl);

if(jsonArray!=null) {

try {

JSONObject totalObject = (JSONObject) jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail = totalObject.getString("RecordDetail"); //详情

if (initSetDataToBean(detail) != null) {

commentAdapter = new DisplayDetail.CommentAdapter(initSetDataToBean(detail)); //将详情设置到bean中

}

} catch (JSONException e) {

Toast.makeText(DisplayDetail.this, "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(DisplayDetail.this, "查询评论为空" , Toast.LENGTH\_LONG).show();

}

}

/\*刷新适配器\*/

public void refreshDiscover(){

listView.setAdapter(null); //将ListView控件的内容清空

initializeAdapter();

listView.setAdapter(commentAdapter);

commentAdapter.notifyDataSetChanged();

setListViewHeightBasedOnChildren(listView);

}

public void setListViewHeightBasedOnChildren(ListView listView) {

// 获取ListView对应的Adapter

ListAdapter listAdapter = listView.getAdapter();

if (listAdapter == null) {

return;

}

int totalHeight = 0;

for (int i = 0, len = listAdapter.getCount(); i < len; i++) {

// listAdapter.getCount()返回数据项的数目

View listItem = listAdapter.getView(i, null, listView);

// 计算子项View 的宽高

listItem.measure(0, 0);

// 统计所有子项的总高度

totalHeight += listItem.getMeasuredHeight();

}

ViewGroup.LayoutParams params = listView.getLayoutParams();

params.height = totalHeight+ (listView.getDividerHeight() \* (listAdapter.getCount() - 1));

// listView.getDividerHeight()获取子项间分隔符占用的高度

// params.height最后得到整个ListView完整显示需要的高度

listView.setLayoutParams(params);

}

//评论

class commentClick implements View.OnClickListener {

public void onClick(View v) {

SharedPreferences sp = DisplayDetail.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

String username = sp.getString("name","");

String commentDetail = et\_comment\_detail.getText().toString();

String AddSupportUrl = baseUrl+"AddComment?infoId="+infoId+"&commentUser="+username+"&commentDetail="+commentDetail;

JSONArray jsonArray = loadData(AddSupportUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

Boolean addOk = totalObject.getBoolean("addOk"); //是否评论成功

if(addOk){

refreshDiscover(); //刷新适配器

et\_comment\_detail.setText("");

}else{

Toast.makeText(DisplayDetail.this, "评论失败，请稍后再试" , Toast.LENGTH\_LONG).show();

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

//点赞

class supportClick implements View.OnClickListener {

public void onClick(View v) {

support = support+1;

String AddSupportUrl = baseUrl+"AddSupport?InfoId="+infoId;

JSONArray jsonArray = loadData(AddSupportUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

Boolean addOk = totalObject.getBoolean("addOk"); //是否点赞成功

if(addOk){

overSupport(); //改变点赞图片状态为已经点赞

}else{

support = support-1; //没有点赞成功

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

//启动线程,加载主页

private class LoadMainTask implements Runnable{

@Override

public void run() {

Intent intent = new Intent(DisplayDetail.this,MainActivity.class);

startActivity(intent); //打开主页

finish(); //关闭

}

}

//删除文章

class deleteClick implements View.OnClickListener {

public void onClick(View v) {

String deleteInfoUrl = baseUrl+"DeleteInfo?infoId="+infoId;

JSONArray jsonArray = loadData(deleteInfoUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

Boolean deleteOk = totalObject.getBoolean("deleteOk"); //是否点赞成功

if(deleteOk){

Toast.makeText(DisplayDetail.this, "删除成功", Toast.LENGTH\_LONG).show();

//通过handler设置延时1秒后执行r任务，跳转到首页

new Handler().postDelayed(new DisplayDetail.LoadMainTask(),1000);

}else{

Toast.makeText(DisplayDetail.this, "删除失败，请稍后再试", Toast.LENGTH\_LONG).show();

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

//收藏或者取消收藏

class focusClick implements View.OnClickListener {

public void onClick(View v) {

if(isFocus){

//取消收藏

String removeFocusUrl = baseUrl+"RemoveFocus?focusId="+focusId;

JSONArray jsonArray = loadData(removeFocusUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

Boolean removeOk = totalObject.getBoolean("removeOk"); //是否取消收藏

if(removeOk){

nowNotLike(); //改变收藏状态为未收藏

}else{

Toast.makeText(DisplayDetail.this, "取消收藏失败，请稍后再试.", Toast.LENGTH\_LONG).show();

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

//添加到我的收藏

SharedPreferences sp = DisplayDetail.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

String username = sp.getString("name","");

String addFocusUrl = baseUrl+"AddFocus?InfoId="+infoId+"&username="+username;

JSONArray jsonArray = loadData(addFocusUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

Boolean addOk = totalObject.getBoolean("addOk"); //是否收藏成功

if(addOk){

overFocus(); //改变收藏状态为已经收藏

Toast.makeText(DisplayDetail.this, "收藏成功!", Toast.LENGTH\_LONG).show();

}else{

Toast.makeText(DisplayDetail.this, "收藏失败，请稍后操作", Toast.LENGTH\_LONG).show();

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

}

/\*\*

\* 初始化，即打开页面展示的数据

\*/

private void initialize(String info\_id){

String QueryInfoUrl = baseUrl+"QueryInfoById?infoId="+info\_id;

JSONArray jsonArray = loadData(QueryInfoUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

infoTitle = totalObject.getString("infoTitle"); //内容标题

/\* Toast.makeText(DisplayDetail.this,totalObject.toString(), Toast.LENGTH\_LONG).show();\*/

infoDescribe = totalObject.getString("infoDescribe"); //内容简述

/\* infoDetail = totalObject.getString("infoDetail"); //内容详情\*/

String str = totalObject.getString("infoDetail");

infoDetail =str.replaceAll("<br>","\n"); //获取回车符

infoAuthor = totalObject.getString("infoAuthor"); //内容作者

type = totalObject.getInt("infoType"); //内容类型

support = totalObject.getInt("infoSupport"); //点赞数

tv\_info\_title.setText(infoTitle);

tv\_info\_author.setText(infoAuthor);

tv\_info\_support.setText(support.toString());

SharedPreferences sp = DisplayDetail.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

String userName = sp.getString("name","");

if(infoAuthor.equals(userName)){

displayDelete(); //显示删除按钮

}else{

hideDelete(); //隐藏删除按钮

}

if(type==0||type==3){

setArticlesDiaries();

}else if(type==1){

//分享图片

setImg();

}else if(type==2){

//分享的音乐

setArticlesDiaries();

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 初始化，是否收藏

\*/

private void initializeFocus(String info\_id){

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = DisplayDetail.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

String username = sp.getString("name","");

String QueryIfFocusUrl = baseUrl+"IfAddFocus?username="+username+"&infoId="+info\_id;

JSONArray jsonArray = loadData(QueryIfFocusUrl);

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

isFocus = totalObject.getBoolean("isFocus"); //是否收藏

//Toast.makeText(DisplayDetail.this, "是否收藏" + isFocus.toString(), Toast.LENGTH\_LONG).show();

if(isFocus){

overFocus(); //改变收藏图片状态为已经收藏

focusId = totalObject.getInt("focusId"); //设置收藏Id

}

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initialize异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 改变为已点赞

\*/

public void overSupport(){

Drawable drawableLeft = getResources().getDrawable(

R.drawable.over\_support);

tv\_info\_support.setCompoundDrawablesWithIntrinsicBounds(drawableLeft,

null, null, null);

tv\_info\_support.setCompoundDrawablePadding(4);

tv\_info\_support.setText(support.toString()); //更新点赞数

}

/\*\*

\* 改变为已收藏

\*/

public void overFocus(){

btn\_info\_like.setImageResource(getResources().getIdentifier(

"over\_focus", "drawable",

DisplayDetail.this.getPackageName()));

isFocus = true;

}

/\*\*

\* 改变为未收藏

\*/

public void nowNotLike(){

btn\_info\_like.setImageResource(getResources().getIdentifier(

"now\_not\_like", "drawable",

DisplayDetail.this.getPackageName()));

isFocus = false;

}

/\*\*

\* 隐藏文章简述

\*/

public void hideDelete(){

btn\_info\_delete.setVisibility(View.GONE); //隐藏删除

}

/\*\*

\* 显示文章简述

\*/

public void displayDelete(){

btn\_info\_delete.setVisibility(View.VISIBLE); //显示删除

}

/\*\*

\* 隐藏文章简述

\*/

public void hideDescribe(){

tv\_info\_describe.setVisibility(View.GONE); //隐藏文章简述

}

/\*\*

\* 显示文章简述

\*/

public void displayDescribe(){

tv\_info\_describe.setVisibility(View.VISIBLE); //显示文章简述

}

/\*\*

\* 隐藏图片

\*/

public void hideImg(){

share\_detail\_img.setVisibility(View.GONE); //隐藏文章简述

}

/\*\*

\* 显示图片

\*/

public void displayImg(){

share\_detail\_img.setVisibility(View.VISIBLE); //显示图片

}

/\*\*

\* 对于文章和日记类的文章

\*/

public void setArticlesDiaries(){

tv\_info\_describe.setText(infoDescribe);

tv\_info\_detail.setText(infoDetail);

}

/\*\*

\* 对于文章和日记类的文章

\*/

public void setImg(){

share\_img\_title.setText(infoTitle);

tv\_info\_detail.setText(infoDetail);

Bitmap one;

//通过网络链接获取图片

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+infoDescribe.trim());

share\_detail\_img.setBackground(new BitmapDrawable(getResources(),one)); //以背景的方式设置图片

}catch(IOException e){

e.printStackTrace();

}

hideDescribe(); //隐藏文章简述

displayImg(); //显示图片

}

//初始化将详情设置到FindInfo bean中

public List<CommentInfo> initSetDataToBean(String detail){

List<CommentInfo> CommentInfo = new ArrayList<CommentInfo>();

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

CommentInfo items = new CommentInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

int commentId = temp.getInt("commentId"); //评论的文章ID

String commentUser = temp.getString("commentUser"); //用户昵称

int commentInfo = temp.getInt("commentInfo"); //评论的文章ID

String commentDetail = temp.getString("commentDetail"); //评论详情

items.setCommentId(commentId);

items.setInfoId(commentInfo);

items.setCommentUser(commentUser);

items.setCommentDetail(commentDetail);

CommentInfo.add(items);

}

return CommentInfo;

}catch (JSONException e){

Toast.makeText(DisplayDetail.this, "initSetPeopleDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

return null;

}

}

/\*\*

\* 根据返回的用户设置卡片的样式

\*/

class CommentAdapter extends BaseAdapter {

List<CommentInfo> newsItems;

public CommentAdapter(List<CommentInfo> newsitems){

this.newsItems = newsitems;

}

@Override

public int getCount() {

return newsItems.size();

}

@Override

public Object getItem(int position) {

return newsItems.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//在这里将Item设置到每个卡片

@Override

public View getView(int position, View view, ViewGroup parent) {

if(view == null){

view = getLayoutInflater().inflate(R.layout.activity\_comment\_list, null);

}

//用户昵称

TextView tvTitle = (TextView)view.findViewById(R.id.comment\_user\_name);

tvTitle.setText(newsItems.get(position).getCommentUser());

//用户ID

TextView getTvId = (TextView)view.findViewById(R.id.comment\_id);

getTvId.setText(newsItems.get(position).getInfoId().toString());

//评论内容

TextView tvComment = (TextView)view.findViewById(R.id.comment\_detail);

tvComment.setText(newsItems.get(position).getCommentDetail());

return view;

}

/\*\*

\* 添加数据列表项

\* @param infoItem

\*/

public void addNewsItem(CommentInfo infoItem){

newsItems.add(infoItem);

}

}

}

---------------------- **DisplayDetail.java结束--------------------------**

“查找”页——FindTabFragment.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为FindTabFragment，将下面的代码复制到该文件中。FindTabFragment.java主要的介绍如下：

* 显示find\_tab\_content.xml，从服务器端获取按点赞数排名前十的内容，将返回的数据设置到页面中
* 给顶部找人、找文按钮添加点击事件，点击时获取输入内容并请求服务器，获取与关键字相似的内容并以列表形式显示
* 若查找类型为找文则需要给listView添加事件监听，达到点击列表跳转到详情的效果
* 给顶部的搜索框清空按钮添加事件，清空时清除搜索框内容，且刷新适配器显示热门内容列表

--------------------------- **FindTabFragment.java开始---------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.Handler;

import android.os.StrictMode;

import android.support.v4.app.Fragment;

import android.util.DisplayMetrics;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.view.WindowManager;

import android.widget.AbsListView;

import android.widget.AdapterView;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.bean.FindInfo;

import com.example.bean.FindPeople;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

public class FindTabFragment extends Fragment implements AbsListView.OnScrollListener{

Activity mActivity; //存放当前的activity

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View findLayout = inflater.inflate(R.layout.find\_tab\_content, container, false);

return findLayout;

}

private ListView listView;

private EditText searchView; //搜索内容框

private Button clearInfo; //找文按钮

private Button seekPeople; //找人按钮

private Button seekInfo; //找文按钮

private TextView hintTv; //提示信息

ImageButton back; //退出应用

private String keyWord; //搜索框的文

private Boolean isHot = true; //用于判断加载热文还是查找的内容

private int page = 1; //请求页数

private int visibleLastIndex = 0; //最后的可视项索引

private int visibleItemCount; // 当前窗口可见项总数

private int dataSize = 28; //数据集的条数

private FindTabFragment.PaginationAdapter adapter; //存放热门和查文信息

private FindTabFragment.PeopleAdapter peopleAdapter; //存放查人

private Handler handler = new Handler();

//返回测试信息

/\*TextView testTxt;\*/

/\*\* Called when the activity is first created. \*/

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

mActivity = this.getActivity();

back = (ImageButton)getActivity().findViewById(R.id.back\_button2);

back.setOnClickListener(new FindTabFragment.mOnClick());

//返回测试信息

/\* testTxt = (TextView)getActivity().findViewById(R.id.test\_discover\_tab);\*/

listView = (ListView)getActivity().findViewById(R.id.find\_listView);

initializeAdapter(); //初始化，填入热门

listView.setAdapter(adapter);

//点击进入详情

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

TextView getTvId = (TextView)view.findViewById(R.id.seek\_list\_cardId);

if(getTvId.length()!=0){

/\*Toast.makeText(mActivity,"点击了"+getTvId.getText().toString()+"id", Toast.LENGTH\_SHORT).show();\*/

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

Boolean isLogin = sp.getBoolean("isLogin",false);

if(isLogin){

//点击了进入详情

Intent display\_info\_intent = new Intent();

display\_info\_intent.setClass(mActivity, DisplayDetail.class); //创建intent对象，并制定跳转页面

display\_info\_intent.putExtra("infoId",getTvId.getText().toString());

mActivity.setResult(1, display\_info\_intent); //这里的1就对应到onActivityResult（）方法中的resultCode

startActivity(display\_info\_intent); //跳转到详情页面

}else{

//尚未登录

Toast.makeText(getActivity(),"请先登录", Toast.LENGTH\_LONG).show();

}

}

}

});

//输入框

searchView = (EditText)getActivity().findViewById(R.id.et\_seek\_search);

//找人、找文按钮、清空文本框内容的button

seekPeople = (Button) getActivity().findViewById(R.id.find\_search\_by\_name);

seekInfo = (Button) getActivity().findViewById(R.id.find\_search\_by\_keyWord);

clearInfo = (Button) getActivity().findViewById(R.id.seek\_bt\_clear);

//提示信息

hintTv = (TextView) getActivity().findViewById(R.id.seek\_list\_hint\_info);

seekPeople.setOnClickListener(new mClick());

seekInfo.setOnClickListener(new mClick());

clearInfo.setOnClickListener(new mClick());

}

class mOnClick implements View.OnClickListener {

@Override

public void onClick(View v) {

mActivity.finish();

}

}

class mClick implements View.OnClickListener {

public void onClick(View v) {

keyWord = searchView.getText().toString();

if(keyWord == ""){

Toast.makeText(getActivity(), "查找内容不能为空！", Toast.LENGTH\_LONG).show();

return;

}

if (v == seekPeople) {

//点击了找人

hintTv.setText("查询结果：");

listView.setAdapter(null); //将ListView控件的内容清空

initSearchPeopleAdapter();

listView.setAdapter(peopleAdapter); //将查询结果插入页面

peopleAdapter.notifyDataSetChanged();

}else if(v == seekInfo){

//点击了找文

isHot = false;

hintTv.setText("查询结果：");

listView.setAdapter(null);

initSearchInfoAdapter();

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();

}else if(v == clearInfo){

//点击了清空

isHot = true;

hintTv.setText("大家都在看：");

listView.setAdapter(null);

initializeAdapter(); //初始化，填入热门

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();

searchView.setText("");

}

}

}

@Override

public void onScrollStateChanged(AbsListView view, int scrollState) {

int itemsLastIndex = adapter.getCount()-1; //数据集最后一项的索引

int lastIndex = itemsLastIndex + 1;

if (scrollState == AbsListView.OnScrollListener.SCROLL\_STATE\_IDLE

&& visibleLastIndex == lastIndex) {

// 如果是自动加载,可以在这里放置异步加载数据的代码

}

}

@Override

public void onScroll(AbsListView view, int firstVisibleItem,

int visibleItemCount, int totalItemCount) {

this.visibleItemCount = visibleItemCount;

visibleLastIndex = firstVisibleItem + visibleItemCount - 1; //最后的可视项索引

//如果所有的记录选项等于数据集的条数，则移除列表底部视图

if(totalItemCount == dataSize+1){

Toast.makeText(getActivity(), "数据全部加载完!", Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 查询数据库中的数据

\*/

private JSONArray loadDataFromDataBase(String QueryInfoUrl){

//Toast.makeText(getActivity(),"保存",Toast.LENGTH\_SHORT).show();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

try {

URL url = new URL(QueryInfoUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

if(stringBuilder.toString()!="null"){

//返回测试信息

JSONArray jsonArray = new JSONArray(stringBuilder.toString());

/\* testTxt.setText(baseUrl+"QueryDiscover?page="+page+"&count="+count);\*/

//获取到的数据，对Json进行解析

page = page+1; //一次成功请求后更新请求页面

buffer.close();

return jsonArray;

}else{

return null;

}

}else{

Toast.makeText(getActivity(),"非200", Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(getActivity(), "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

return null;

}

//初始化将详情设置到FindInfo bean中

public List<FindInfo> initSetDataToBean(String detail){

List<FindInfo> findInfo = new ArrayList<FindInfo>();

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindInfo items = new FindInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

Integer infoId = temp.getInt("infoId"); //内容ID

String infoTitle = temp.getString("infoTitle"); //内容标题

String infoDescribe = temp.getString("infoDescribe"); //内容简述

String infoDetail = temp.getString("infoDetail"); //内容详情

Integer type = temp.getInt("infoType"); //类型：0表示日记，1表示趣事

Integer support = temp.getInt("infoSupport"); //点赞数

String infoAuthor = temp.getString("infoAuthor"); //作者

items.setInfoId(infoId);

items.setInfoTitle(infoTitle);

items.setInfoDescribe(infoDescribe);

items.setInfoDetail(infoDetail);

items.setType(type);

items.setSupport(support);

items.setInfoAuthor(infoAuthor);

findInfo.add(items);

}

return findInfo;

}catch (JSONException e){

Toast.makeText(getActivity(), "initSetDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

return null;

}

}

//初始化将详情设置到FindInfo bean中

public List<FindPeople> initSetPeopleDataToBean(String detail){

List<FindPeople> findPeople = new ArrayList<FindPeople>();

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindPeople items = new FindPeople();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

String userName = temp.getString("userName"); //用户昵称

String signature = temp.getString("signature"); //用户签名

String userLogImage = temp.getString("userLogImage"); //用户头像

items.setUserName(userName);

items.setSignature(signature);

items.setUserLogImage(userLogImage);

findPeople.add(items);

}

return findPeople;

}catch (JSONException e){

Toast.makeText(getActivity(), "initSetPeopleDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

return null;

}

}

/\*\*

\* 初始化ListView的适配器，即打开页面展示的数据

\*/

private void initializeAdapter(){

// 设置线程策略

setVersion();

String QueryHotInfoUrl = baseUrl+"QueryHotInfo";

JSONArray jsonArray = loadDataFromDataBase(QueryHotInfoUrl);

if(jsonArray!=null) {

try {

JSONObject totalObject = (JSONObject) jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail = totalObject.getString("RecordDetail"); //详情

if (initSetDataToBean(detail) != null) {

adapter = new PaginationAdapter(initSetDataToBean(detail)); //将详情设置到bean中

}

} catch (JSONException e) {

Toast.makeText(getActivity(), "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(getActivity(), "查询为空" , Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 初始化ListView的适配器，将查找文结果显示出来

\*/

private void initSearchInfoAdapter(){

// 设置线程策略

setVersion();

String QueryHotInfoUrl = baseUrl+"QueryInfoByKey?key="+keyWord;

JSONArray jsonArray = loadDataFromDataBase(QueryHotInfoUrl);

if(jsonArray!=null) {

try {

JSONObject totalObject = (JSONObject) jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail = totalObject.getString("RecordDetail"); //详情

if (initSetDataToBean(detail) != null) {

adapter = new PaginationAdapter(initSetDataToBean(detail)); //将详情设置到bean中

}

} catch (JSONException e) {

Toast.makeText(getActivity(), "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(getActivity(), "查询为空" , Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 初始化ListView的适配器，将查找人结果显示出来

\*/

private void initSearchPeopleAdapter(){

// 设置线程策略

setVersion();

String QueryPeopleInfoUrl = baseUrl+"QueryPeopleInfoByKey?nameKey="+keyWord;

JSONArray jsonArray = loadDataFromDataBase(QueryPeopleInfoUrl);

if(jsonArray!=null) {

try {

JSONObject totalObject = (JSONObject) jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail = totalObject.getString("RecordDetail"); //详情

if (initSetPeopleDataToBean(detail) != null) {

peopleAdapter = new PeopleAdapter(initSetPeopleDataToBean(detail)); //将详情设置到bean中

}

} catch (JSONException e) {

Toast.makeText(getActivity(), "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(getActivity(), "查询为空" , Toast.LENGTH\_LONG).show();

}

}

//APP如果在主线程中请求网络操作，将会抛出异常，所以需要用线程来操作网络请求

void setVersion() {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()

.detectDiskReads()

.detectDiskWrites()

.detectNetwork()

.penaltyLog()

.build());

StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()

.detectLeakedSqlLiteObjects() //探测SQLite数据库操作

.penaltyLog() //打印logcat

.penaltyDeath()

.build());

}

/\*\*

\* 将Int转为String

\*/

private String IntToString(Integer num){

try {

String str = num.toString(); //第几页

return str;

} catch (NumberFormatException e) {

e.printStackTrace();

}

return null;

}

/\*\*

\* 根据返回的文章类型设置卡片的样式

\*/

class PaginationAdapter extends BaseAdapter {

List<FindInfo> newsItems;

public PaginationAdapter(List<FindInfo> newsitems){

this.newsItems = newsitems;

}

@Override

public int getCount() {

return newsItems.size();

}

@Override

public Object getItem(int position) {

return newsItems.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//在这里将Item设置到每个卡片

@Override

public View getView(int position, View view, ViewGroup parent) {

if(view == null){

view = getLayoutInflater().inflate(R.layout.activity\_seek\_list\_item, null);

}

//将热门标识去掉

if(!isHot){

TextView textViewImgBc = (TextView) view.findViewById(R.id.seek\_list\_textView);

textViewImgBc.setCompoundDrawables(null, null, null, null);

}

//内容标题

TextView tvTitle = (TextView)view.findViewById(R.id.seek\_list\_textView);

tvTitle.setText(newsItems.get(position).getInfoTitle());

//内容id

TextView getTvId = (TextView)view.findViewById(R.id.seek\_list\_cardId);

getTvId.setText(newsItems.get(position).getInfoId().toString());

//内容点赞数

TextView tvSupport = (TextView)view.findViewById(R.id.seek\_list\_textView2);

String num = IntToString(newsItems.get(position).getSupport());

tvSupport.setText(num);

return view;

}

/\*\*

\* 添加数据列表项

\* @param infoItem

\*/

public void addNewsItem(FindInfo infoItem){

newsItems.add(infoItem);

}

}

/\*\*

\* 根据返回的用户设置卡片的样式

\*/

class PeopleAdapter extends BaseAdapter {

List<FindPeople> newsItems;

public PeopleAdapter(List<FindPeople> newsitems){

this.newsItems = newsitems;

}

@Override

public int getCount() {

return newsItems.size();

}

@Override

public Object getItem(int position) {

return newsItems.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//在这里将Item设置到每个卡片

@Override

public View getView(int position, View view, ViewGroup parent) {

if(view == null){

view = getLayoutInflater().inflate(R.layout.activity\_seek\_list\_item, null);

}

//用户昵称

TextView tvTitle = (TextView)view.findViewById(R.id.seek\_list\_textView);

tvTitle.setText(newsItems.get(position).getUserName());

//这是用户，内容id为空

TextView getTvId = (TextView)view.findViewById(R.id.seek\_list\_cardId);

getTvId.setText("");

//用户签名

TextView tvSupport = (TextView)view.findViewById(R.id.seek\_list\_textView2);

tvSupport.setText(newsItems.get(position).getSignature());

//用户头像

//通过网络链接获取图片

String imgUrl = newsItems.get(position).getUserLogImage();

Bitmap one;

ImageView userImg = (ImageView) view.findViewById(R.id.seek\_list\_imgView);

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+imgUrl);

userImg.setImageBitmap(one);

}catch(IOException e){

e.printStackTrace();

}

TextView textViewImgBc = (TextView) view.findViewById(R.id.seek\_list\_textView);

textViewImgBc.setCompoundDrawables(null, null, null, null);

/\*TableRow userImg = (TableRow) view.findViewById(R.id.seek\_list\_tableRow);

String imgUrl = newsItems.get(position).getUserLogImage();

Bitmap one;

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+imgUrl);

\*//\*userImg.setBackground(new BitmapDrawable(getResources(),one)); //以背景的方式设置头像\*//\*

Drawable drawable = new BitmapDrawable(getResources(),one);

// 这一步必须要做,否则不会显示.

drawable.setBounds(drawable.getMinimumWidth(), 0,0 , drawable.getMinimumHeight());

textViewImgBc.setCompoundDrawables(drawable, null, null, null);

}catch(IOException e){

e.printStackTrace();

}\*/

return view;

}

/\*\*

\* 添加数据列表项

\* @param infoItem

\*/

public void addNewsItem(FindPeople infoItem){

newsItems.add(infoItem);

}

}

}

----------------------- **FindTabFragment.java结束-------------------------**

“发布”页——EditTabFragment.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为EditTabFragment，将下面的代码复制到该文件中。EditTabFragment.java主要的介绍如下：

* 显示edit\_tab\_content.xml，给布局中顶部四个Tab添加点击事件，切换不同分享类型时改变输入框的提示文字
* 给发布按钮添加事件，点击时获取到用户选择的类型、输入的内容标题、内容简述、内容详情，并将数据发送到后台服务，实现发布内容的功能
* 发布成功后，清空用户的输入内容

--------------------------- **EditTabFragment.java开始-------------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.os.StrictMode;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class EditTabFragment extends Fragment {

RelativeLayout bt\_share\_funny,bt\_share\_pic,bt\_share\_music,bt\_share\_diary; //四种类型的内容分享

LinearLayout edit\_detail\_info\_layout;

EditText et\_edit\_title\_info,et\_edit\_describe\_info; //标题和简述

EditText et\_edit\_detail\_info; //分享的详情

Button edit\_sure\_share; //发布按钮\_

ImageButton back; //退出应用

ImageView share\_funny\_image\_bc,share\_pic\_image\_bc,share\_music\_image\_bc,share\_diary\_image\_bc;

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

Integer infoType = 0; //默认是分享趣事

Activity mActivity; //存放当前的activity

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

View moreLayout = inflater.inflate(R.layout.edit\_tab\_content, container, false);

return moreLayout;

}

//APP如果在主线程中请求网络操作，将会抛出异常，所以需要用线程来操作网络请求

void setVersion() {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()

.detectDiskReads()

.detectDiskWrites()

.detectNetwork()

.penaltyLog()

.build());

StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()

.detectLeakedSqlLiteObjects() //探测SQLite数据库操作

.penaltyLog() //打印logcat

.penaltyDeath()

.build());

}

//绑定点击事件

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

mActivity = this.getActivity();

bt\_share\_funny = (RelativeLayout) getActivity().findViewById(R.id.bt\_share\_funny); //分享趣事

bt\_share\_pic = (RelativeLayout) getActivity().findViewById(R.id.bt\_share\_pic); //分享图片

bt\_share\_music = (RelativeLayout) getActivity().findViewById(R.id.bt\_share\_music); //分享音乐

bt\_share\_diary = (RelativeLayout) getActivity().findViewById(R.id.bt\_share\_diary); //发布日记

edit\_detail\_info\_layout = (LinearLayout) getActivity().findViewById(R.id.edit\_detail\_info\_layout);// 详情框

et\_edit\_title\_info = (EditText) getActivity().findViewById(R.id.et\_edit\_title\_info); //分享标题

et\_edit\_describe\_info = (EditText) getActivity().findViewById(R.id.et\_edit\_describe\_info); //分享简述

et\_edit\_detail\_info = (EditText) getActivity().findViewById(R.id.et\_edit\_detail\_info); //分享简述

edit\_sure\_share = (Button) getActivity().findViewById(R.id.edit\_sure\_share); //确定分享

share\_funny\_image\_bc = (ImageView) getActivity().findViewById(R.id.share\_funny\_image\_bc);

share\_pic\_image\_bc = (ImageView) getActivity().findViewById(R.id.share\_pic\_image\_bc);

share\_music\_image\_bc = (ImageView) getActivity().findViewById(R.id.share\_music\_image\_bc);

share\_diary\_image\_bc = (ImageView) getActivity().findViewById(R.id.share\_diary\_image\_bc);

//默认显示分享趣事

changeEditDisplay(infoType);

bt\_share\_funny.setOnClickListener(new btnClick());

bt\_share\_pic.setOnClickListener(new btnClick());

bt\_share\_music.setOnClickListener(new btnClick());

bt\_share\_diary.setOnClickListener(new btnClick());

edit\_detail\_info\_layout.setOnClickListener(new btnClick());

edit\_sure\_share.setOnClickListener(new okClick());

back = (ImageButton)getActivity().findViewById(R.id.back\_button3);

back.setOnClickListener(new EditTabFragment.mOnClick());

}

class mOnClick implements View.OnClickListener {

@Override

public void onClick(View v) {

mActivity.finish();

}

}

//改变UI

public void changeEditDisplay(int selectType) {

if(selectType==0){

setAllBc();

setAllNone();

share\_funny\_image\_bc.setImageResource(R.drawable.share\_funny\_select);

et\_edit\_title\_info.setHint("在这里填写标题");

et\_edit\_describe\_info.setHint("简述你要分享的趣事");

et\_edit\_detail\_info.setHint("在这详细介绍趣事");

edit\_sure\_share.setText("分享趣事");

}else if(selectType==1){

setAllBc();

setAllNone();

share\_pic\_image\_bc.setImageResource(R.drawable.share\_pic\_select);

et\_edit\_title\_info.setHint("在这里填写标题");

et\_edit\_describe\_info.setHint("在这里填写图片链接：test1.jpeg");

et\_edit\_detail\_info.setHint("在这里填写图片详细介绍");

edit\_sure\_share.setText("分享图片");

}else if(selectType==2){

setAllBc();

setAllNone();

share\_music\_image\_bc.setImageResource(R.drawable.share\_music\_select);

et\_edit\_title\_info.setHint("在这里填写标题");

et\_edit\_describe\_info.setHint("在这里填写歌名");

et\_edit\_detail\_info.setHint("在这里填写音乐链接");

edit\_sure\_share.setText("分享音乐");

}else if(selectType==3){

setAllBc();

setAllNone();

share\_diary\_image\_bc.setImageResource(R.drawable.share\_diary\_select);

et\_edit\_title\_info.setHint("在这里填写标题");

et\_edit\_describe\_info.setHint("在这里填写今天的心情");

et\_edit\_detail\_info.setHint("在这里记录今天的事");

edit\_sure\_share.setText("发布日记");

}

}

//让所有的按钮变成灰色

public void setAllBc(){

share\_funny\_image\_bc.setImageResource(R.drawable.share\_funny\_no\_select);

share\_pic\_image\_bc.setImageResource(R.drawable.share\_pic\_no\_select);

share\_music\_image\_bc.setImageResource(R.drawable.share\_music\_no\_select);

share\_diary\_image\_bc.setImageResource(R.drawable.share\_diary\_no\_select);

}

//清空所有文本输入框的值

public void setAllNone(){

et\_edit\_title\_info.setText("");

et\_edit\_describe\_info.setText("");

et\_edit\_detail\_info.setText("");

}

//点击不同分享内容的时候改变UI

class btnClick implements View.OnClickListener {

public void onClick(View v) {

if (v == bt\_share\_funny) {

infoType = 0;

changeEditDisplay(infoType);

/\*Toast.makeText(getActivity(), textView.getText(), Toast.LENGTH\_LONG).show();\*/

}else if(v == bt\_share\_pic){

infoType = 1;

changeEditDisplay(infoType);

}else if(v == bt\_share\_music){

infoType = 2;

changeEditDisplay(infoType);

}else if(v == bt\_share\_diary){

infoType = 3;

changeEditDisplay(infoType);

}else if(v == edit\_detail\_info\_layout){

et\_edit\_detail\_info.setFocusable(true);

et\_edit\_detail\_info.setFocusableInTouchMode(true);

}

}

}

//点击不同分享内容的时候改变UI

class okClick implements View.OnClickListener {

public void onClick(View v) {

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

Boolean isLogin = sp.getBoolean("isLogin",false);

if(isLogin){

setVersion();

String userName = sp.getString("name","");

String title = et\_edit\_title\_info.getText().toString();

String describe = et\_edit\_describe\_info.getText().toString();

String str = et\_edit\_detail\_info.getText().toString();

String detail=str.replaceAll("\n", "<br>");

Boolean boo = title.length()>0&&describe.length()>0&&detail.length()>0;

if(boo){

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

try {

String registerUrl = baseUrl+"AddInfo?infoTitle="+title+"&infoDescribe="+describe+"&infoDetail="+detail +"&type="+infoType+"&infoAuthor="+userName;

URL url = new URL(registerUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

setAllNone();

Toast.makeText(getActivity(), stringBuilder, Toast.LENGTH\_LONG).show();

//Toast.makeText(getActivity(), detail, Toast.LENGTH\_LONG).show();

buffer.close();

}else{

Toast.makeText(getActivity(), "非200.."+connGET.getResponseCode(), Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(getActivity(), "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(getActivity(), "清将信息填写完整!", Toast.LENGTH\_LONG).show();

}

}else{

Toast.makeText(getActivity(), "清先登录!", Toast.LENGTH\_LONG).show();

}

}

}

}

---------------------------- **EditTabFragment.java结束-----------------------**

“我的”页——MyTabFragment. java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为MyTabFragment，将下面的代码复制到该文件中。MyTabFragment.java主要的介绍如下：

* 显示my\_tab\_content.xml，给布局中的注册、登录绑定点击事件，点击时创建注册、登录对话框，实现注册登录功能
* 给账号管理、更新签名、修改密码、我的分享、我的日记、我的收藏、关于i分享这几个组件添加点击跳转页面的事件
* 在登录成功之后改变页面UI，显示用户昵称和签名

---------------------------- **MyTabFragment.java开始------------------------**

package com.example.discover;

import android.app.Activity;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.StrictMode;

import android.support.v4.app.Fragment;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.widget.TableRow;

import android.widget.TextView;

import android.widget.Toast;

import org.json.JSONArray;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class MyTabFragment extends Fragment{

TableRow account\_set\_bt,signature\_set\_bt,password\_set\_bt,share\_list\_bt,diary\_list\_bt,focus\_list\_bt,about\_iShare; //相关功能

View myLayout; //“我的”页面

Button pop\_register,pop\_login; //弹出对话框的按钮

ImageView my\_head\_img; //用户头像

TextView my\_name,my\_sign; //用户昵称和心情

/\*Button go\_login,go\_register;\*/

String userName,passWord,signature,headImg; //存放用户基本信息

RelativeLayout login,register; //注册、登录页面

LinearLayout login\_top,un\_login\_top; //“我的”页面顶部两种状态下的布局

Activity mActivity; //存放当前的activity

/\*TextView test3; //用于返回测试信息\*/

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

boolean isLogin;

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

myLayout = inflater.inflate(R.layout.my\_tab\_content, container, false);

return myLayout;

}

//APP如果在主线程中请求网络操作，将会抛出异常，所以需要用线程来操作网络请求

void setVersion() {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()

.detectDiskReads()

.detectDiskWrites()

.detectNetwork()

.penaltyLog()

.build());

StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()

.detectLeakedSqlLiteObjects() //探测SQLite数据库操作

.penaltyLog() //打印logcat

.penaltyDeath()

.build());

}

//绑定点击事件

@Override

public void onActivityCreated(Bundle savedInstanceState) {

super.onActivityCreated(savedInstanceState);

mActivity = this.getActivity();

// 设置线程策略

setVersion();

login\_top = ( LinearLayout)getActivity().findViewById(R.id.login\_top); //“我的”页面顶部登录后显示的用户信息

un\_login\_top = (LinearLayout)getActivity().findViewById(R.id.un\_login\_top); //“我的”页面未登录时显示的登录、注册按钮

my\_head\_img = ( ImageView) getActivity().findViewById(R.id.images\_head); //我的头像

my\_name = (TextView) getActivity().findViewById(R.id.t\_name); //我的昵称

my\_sign = (TextView) getActivity().findViewById(R.id.t\_sign); //我的签名

account\_set\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row1); //"我的"界面第一个功能，账号管理

account\_set\_bt.setOnClickListener(new mClick());

signature\_set\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row2); //"我的"界面第二个功能，修改签名

signature\_set\_bt.setOnClickListener(new mClick());

password\_set\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row3); //"我的"界面第三个功能，修改密码

password\_set\_bt.setOnClickListener(new mClick());

share\_list\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row4); //"我的"界面第四个功能，我的分享

share\_list\_bt.setOnClickListener(new mClick());

diary\_list\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row5); //"我的"界面第五个功能，我的日记

diary\_list\_bt.setOnClickListener(new mClick());

focus\_list\_bt =(TableRow)getActivity().findViewById(R.id.more\_page\_row6); //"我的"界面第六个功能，我的日记

focus\_list\_bt.setOnClickListener(new mClick());

about\_iShare = (TableRow)getActivity().findViewById(R.id.more\_page\_row7); //"我的"界面第七个功能，关于iShare

about\_iShare.setOnClickListener(new mClick());

pop\_register = (Button)getActivity().findViewById(R.id.pop\_register); //弹出注册对话框的按钮

pop\_login = (Button)getActivity().findViewById(R.id.pop\_login); //弹出登录对话框的按钮

/\* test3 = (TextView) getActivity().findViewById(R.id.test3\_info); //测试\*/

//根据登录状态显示不同的UI ，用SharePreferences判断用户是否登录

changeStatus();

pop\_register.setOnClickListener(new popClick()); //弹出注册登录对话框事件

pop\_login.setOnClickListener(new popClick());

}

//“我的”页面中相关点击事件

class mClick implements View.OnClickListener {

public void onClick(View v) {

if(isLogin) {

if (v == account\_set\_bt) {

//点击了账号管理

Intent account\_manage\_intent = new Intent();

account\_manage\_intent.setClass(mActivity, AccountManagement.class); //创建intent对象，并制定跳转页面

startActivity(account\_manage\_intent); //跳转到账号管理页面

/\*TextView textView = (TextView) getActivity().findViewById(R.id.test\_text\_view); //用于放测试的提示信息

Toast.makeText(getActivity(), textView.getText(), Toast.LENGTH\_LONG).show();\*/

}else if(v == signature\_set\_bt){

//点击了更改签名

Intent signature\_manage\_intent = new Intent();

signature\_manage\_intent.setClass(mActivity, ChangeSignature.class); //创建intent对象，并指定跳转页面

//startActivity(signature\_manage\_intent); //跳转到更新签名页面

startActivityForResult(signature\_manage\_intent,1); //需要返回该页面

}else if(v == password\_set\_bt){

//点击了更改密码

Intent password\_manage\_intent = new Intent();

password\_manage\_intent.setClass(mActivity, ChangePassword.class); //创建intent对象，并指定跳转页面

/\*startActivity(password\_manage\_intent); \*/

startActivityForResult(password\_manage\_intent,2); // 跳转到修改密码页面

}else if(v == share\_list\_bt){

//点击了我的分享

Intent intent = new Intent();

intent.setClass(mActivity, MyShareList.class); //创建intent对象，并指定跳转页面

intent.putExtra("requestType",0);

mActivity.setResult(4, intent); //这里的4就对应到onActivityResult（）方法中的resultCode

startActivity(intent); //跳转到list

}else if(v==diary\_list\_bt){

//点击了我的日记

Intent intent = new Intent();

intent.setClass(mActivity, MyShareList.class); //创建intent对象，并指定跳转页面

intent.putExtra("requestType",1);

mActivity.setResult(5, intent); //这里的4就对应到onActivityResult（）方法中的resultCode

startActivity(intent); //跳转到list

}else if(v==focus\_list\_bt){

//点击了我的收藏

Intent intent = new Intent();

intent.setClass(mActivity, MyShareList.class); //创建intent对象，并指定跳转页面

intent.putExtra("requestType",2);

mActivity.setResult(6, intent); //这里的4就对应到onActivityResult（）方法中的resultCode

startActivity(intent); //跳转到list

} else if(v==about\_iShare){

//点击了关于iShare

Intent about\_iShare\_intent = new Intent();

about\_iShare\_intent.setClass(mActivity, AboutIShare.class); //创建intent对象，并指定跳转页面

startActivity(about\_iShare\_intent); //跳转到iShare介绍页面

}

}else{

Toast.makeText(getActivity(), "请先登录！", Toast.LENGTH\_LONG).show();

}

}

}

//登录、注册弹框

class popClick implements View.OnClickListener {

AlertDialog.Builder builder = new AlertDialog.Builder(getContext());

public void onClick(View v) {

if (v == pop\_register) {

register= (RelativeLayout) getLayoutInflater().inflate(R.layout.activity\_register, null); //从另外的布局关联组件

builder.setTitle("用户注册");

builder.setTitle("i 分享").setView(register);

builder.setNegativeButton("取消", new exitClick());

/\*dialog.setPositiveButton("注册", new registerClick());\*/

builder.setPositiveButton("注册", null); //不监听，防止被关闭

builder.setIcon(R.drawable.app\_icon);

final AlertDialog dialog = builder.create();

dialog.show();

//不会自动关闭对话框，可手动关闭

dialog.getButton(AlertDialog.BUTTON\_POSITIVE).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//Toast.makeText(getActivity(),"保存",Toast.LENGTH\_SHORT).show();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

EditText txtUserName,txtUserPass1,txtUserPass2,txtSignature;

TextView test1;

String passWord2;

txtUserName = (EditText)register.findViewById(R.id.et\_username);

txtUserPass1 = (EditText)register.findViewById(R.id.et\_password1);

txtUserPass2 = (EditText)register.findViewById(R.id.et\_password2);

txtSignature= (EditText)register.findViewById(R.id.et\_signature);

test1 = (TextView)register.findViewById(R.id.test1\_view\_info);

userName = txtUserName.getText().toString();

passWord = txtUserPass1.getText().toString();

passWord2 = txtUserPass2.getText().toString();

signature = txtSignature.getText().toString();

//从密码框中取值比较

if(!passWord.equals(passWord2)){

test1.setText("两次密码不一致！");

Toast.makeText(getActivity(), "两次密码不一致", Toast.LENGTH\_LONG).show();

return;

}

try {

String registerUrl = baseUrl+"Register?username="+userName+"&password="+passWord+"&signature="+signature;

URL url = new URL(registerUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

test1.setText(stringBuilder);

dialog.dismiss(); //注册成功，手动关闭对话框

buffer.close();

}else{

test1.setText("非200.."+connGET.getResponseCode() );

}

} catch (Exception e) {

e.printStackTrace();

test1.setText("get 提交 err.." + e.toString());

}

Toast.makeText(getActivity(), test1.getText(), Toast.LENGTH\_LONG).show();

}

});

} else if (v == pop\_login) {

login = (RelativeLayout) getLayoutInflater().inflate(R.layout.activity\_login, null); //从另外的布局关联组件

builder.setTitle("i 分享").setView(login);

builder.setNegativeButton("取消", new exitClick());

builder.setPositiveButton("登录", null);

builder.setIcon(R.drawable.app\_icon);

final AlertDialog dialog2 = builder.create();

dialog2.show();

//不会自动关闭对话框，可手动关闭

dialog2.getButton(AlertDialog.BUTTON\_POSITIVE).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//Toast.makeText(getActivity(),"保存",Toast.LENGTH\_SHORT).show();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

EditText txtUserName2,txtUserPass2;

TextView test2;

txtUserName2 = (EditText)login.findViewById(R.id.et\_login\_username);

txtUserPass2 = (EditText)login.findViewById(R.id.et\_login\_password);

test2 = (TextView)login.findViewById(R.id.test2\_view\_info);

userName = txtUserName2.getText().toString();

passWord = txtUserPass2.getText().toString();

//从密码框中取值比较

if(userName.length()==0||passWord.length()==0){

test2.setText("用户名和密码不能为空!");

Toast.makeText(getActivity(), "用户名和密码不能为空!", Toast.LENGTH\_LONG).show();

return;

}

try {

String LoginUrl = baseUrl+"Login?username="+userName+"&password="+passWord;

URL url = new URL(LoginUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

JSONArray jsonArray = new JSONArray(stringBuilder.toString());

JSONObject userObject = (JSONObject)jsonArray.get(0);

userName = userObject.getString("userName");

signature = userObject.getString("signature");

headImg = userObject.getString("userLogImage");

loginSuccess(userName,signature,headImg);

dialog2.dismiss(); //登录成功，手动关闭对话框

buffer.close();

}else{

test2.setText("登录失败"+connGET.getResponseCode());

Toast.makeText(getActivity(), "非200.."+connGET.getResponseCode(), Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

test2.setText("登录失败.." + e.toString());

Toast.makeText(getActivity(), "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

//dialog2.dismiss()

}

});

}

}

}

//登录成功后的界面以及状态更改

public void loginSuccess(String name,String sign,String headImg){

//登录成功后更改登录状态为已登录

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sp.edit();

editor.putBoolean("isLogin", true);

editor.putString("name", name);

editor.putString("sign", sign);

editor.putString("headImg", headImg);

editor.commit();

Toast.makeText(getActivity(), "登录成功", Toast.LENGTH\_LONG).show();

//UI更新

changeStatus();

}

//根据状态显示或隐藏“我的”页面中头部信息

public void changeStatus(){

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

isLogin = sp.getBoolean("isLogin",false);

if(isLogin){

Bitmap one;

userName = sp.getString("name","");

signature = sp.getString("sign","");

headImg = sp.getString("headImg","");

//设置用户昵称、头像、心情

my\_name.setText(userName);

my\_sign.setText(signature);

//通过网络链接获取图片

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+headImg);

my\_head\_img.setImageBitmap(one);

}catch(IOException e){

e.printStackTrace();

}

//已经登录时设置用户相关信息为显示状态

//username = this.getActivity().getSharedPreferences("config", Context.MODE\_PRIVATE).getString("fname", "");

login\_top.setVisibility(View.VISIBLE);

un\_login\_top.setVisibility(View.GONE);

}else{

//尚未登录，设置登录、注册按钮为显示状态

un\_login\_top.setVisibility(View.VISIBLE);

login\_top.setVisibility(View.GONE);

}

}

//注册对话框的“注册”按钮

/\*class registerClick implements DialogInterface.OnClickListener{

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

EditText txtUserName,txtUserPass1,txtUserPass2,txtSignature;

TextView test1;

String passWord2;

public void onClick(DialogInterface dialog,int which){

txtUserName = (EditText)register.findViewById(R.id.et\_username);

txtUserPass1 = (EditText)register.findViewById(R.id.et\_password1);

txtUserPass2 = (EditText)register.findViewById(R.id.et\_password2);

txtSignature= (EditText)register.findViewById(R.id.et\_signature);

test1 = (TextView)register.findViewById(R.id.test1\_view\_info);

userName = txtUserName.getText().toString();

passWord = txtUserPass1.getText().toString();

passWord2 = txtUserPass2.getText().toString();

signature = txtSignature.getText().toString();

//从密码框中取值比较

if(!passWord.equals(passWord2)){

test1.setText("两次密码不一致！");

Toast.makeText(getActivity(), "两次密码不一致", Toast.LENGTH\_LONG).show();

return;

}

try {

String registerUrl = baseUrl+"Register?username="+userName+"&password="+passWord+"&signature="+signatstringbuilder用法ure;

URL url = new URL(registerUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

test1.setText(stringBuilder);

buffer.close();

}else{

test1.setText("非200.."+connGET.getResponseCode() );

}

} catch (Exception e) {

e.printStackTrace();

test1.setText("get 提交 err.." + e.toString());

}

Toast.makeText(getActivity(), test1.getText(), Toast.LENGTH\_LONG).show();

//dialog.dismiss();

test3.setText(userName+"——"+passWord+"——"+passWord2+"——"+signature+"——"+baseUrl+"Register?username="+userName+"&password="+passWord+"&signature="+signature+"\_"+test1.getText());

}

}\*/

//输入对话框的“退出”按钮事件

class exitClick implements DialogInterface.OnClickListener{

public void onClick(DialogInterface dialog,int which){

dialog.cancel();

}

}

public void onActivityResult(int requestCode,int resultCode,Intent data){

super.onActivityResult(requestCode,resultCode,data);

switch (resultCode){

//requestCode：就是之前在主界面的.startActivityForResult的第二个参数值，通过判断requestCode就可以知道是哪张页面返回回来的

//resultCode： 根据它来判断返回页面所需要的不同操作

case 1:

changeStatus();

//("新签名为："+data.getStringExtra("newSignature")); //是从修改签名页面跳转回来的

break;

case 2:

break;

default:

break;

}

}

}

---------------------- **MyTabFragment.java结束----------------------------**

“关于我的”——MyShareList.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为MyShareList，将下面的代码复制到该文件中。MyShareList java主要的介绍如下：

* 显示activity\_about\_me.xml，根据跳转源的参数确定要想查询内容的类型，再从服务器获取对应的数据并显示到页面中
* 给listView添加监听事件，实现点击列表跳转内容详情的功能

----------------------------- **MyShareList.java开始----------------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.Handler;

import android.os.StrictMode;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AbsListView;

import android.widget.AdapterView;

import android.widget.BaseAdapter;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

import com.example.bean.FindInfo;

import org.json.JSONArray;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import java.util.concurrent.ScheduledExecutorService;

public class MyShareList extends Activity implements AbsListView.OnScrollListener{

Activity mActivity; //存放当前的activity

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

int requestType=0; //请求类型，0：我的分享、1：我的日记；2：我的收藏

String userName="";

String requestUrl="";

private TextView title;

private MyShareList.PaginationAdapter adapterAction;

private ScheduledExecutorService scheduledExecutorService;

private ListView listView;

private int page = 1; //请求页数

private int count = 10; //每次请求的数量

private int visibleLastIndex = 0; //最后的可视项索引

private int visibleItemCount; // 当前窗口可见项总数

private int dataSize = 28; //数据集的条数

private MyShareList.PaginationAdapter adapter;

private View loadMoreView;

private Button loadMoreButton;

private Handler handler = new Handler();

//返回测试信息

/\*TextView testTxt;\*/

/\*public void refreshDiscover(){

page = 1;

visibleLastIndex = 0;

listView.setAdapter(null); //将ListView控件的内容清空

initializeAdapter();

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();

}\*/

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about\_me);

mActivity = MyShareList.this;

//取得启动该Activity的Intent对象

Intent intent =getIntent();

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = MyShareList.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

userName = sp.getString("name","");

requestType =intent.getIntExtra("requestType",0);

//取出Intent中附加的数据

if(requestType==0){

requestUrl = baseUrl+"QueryAboutMe?username="+userName+"&type=0&page="+page+"&count="+count;

}else if(requestType==1){

requestUrl = baseUrl+"QueryAboutMe?username="+userName+"&type=1&page="+page+"&count="+count;

}else{

requestUrl = baseUrl+"QueryAboutMe?username="+userName+"&type=2&page="+page+"&count="+count;

}

loadMoreView = getLayoutInflater().inflate(R.layout.activity\_find\_loadmore, null);

loadMoreButton = (Button)loadMoreView.findViewById(R.id.loadMoreButton);

loadMoreButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

loadMoreButton.setText("正在加载中..."); //设置按钮文字

handler.postDelayed(new Runnable() {

@Override

public void run() {

loadMoreData();

adapter.notifyDataSetChanged();

loadMoreButton.setText("查看更多..."); //恢复按钮文字

}

},2000);

}

});

//返回测试信息

/\* testTxt = (TextView)getActivity().findViewById(R.id.test\_discover\_tab);\*/

listView = (ListView)findViewById(R.id.my\_share\_listView);

listView.addFooterView(loadMoreView); //设置列表底部视图

initializeAdapter();

listView.setAdapter(adapter);

listView.setOnScrollListener(this);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

TextView getTvId = (TextView)view.findViewById(R.id.cardId);

/\*Toast.makeText(mActivity,"点击了"+getTvId.getText().toString()+"id", Toast.LENGTH\_SHORT).show();\*/

SharedPreferences sp = mActivity.getSharedPreferences("save", Context.MODE\_PRIVATE);

Boolean isLogin = sp.getBoolean("isLogin",false);

if(isLogin){

//点击了进入详情

Intent display\_info\_intent = new Intent();

display\_info\_intent.setClass(mActivity, DisplayDetail.class); //创建intent对象，并制定跳转页面

display\_info\_intent.putExtra("infoId",getTvId.getText().toString());

mActivity.setResult(1, display\_info\_intent); //这里的1就对应到onActivityResult（）方法中的resultCode

startActivity(display\_info\_intent); //跳转到详情页面

}else{

//尚未登录

Toast.makeText(MyShareList.this,"请先登录", Toast.LENGTH\_LONG).show();

}

}

});

}

@Override

public void onScrollStateChanged(AbsListView view, int scrollState) {

int itemsLastIndex = adapter.getCount()-1; //数据集最后一项的索引

int lastIndex = itemsLastIndex + 1;

if (scrollState == AbsListView.OnScrollListener.SCROLL\_STATE\_IDLE

&& visibleLastIndex == lastIndex) {

// 如果是自动加载,可以在这里放置异步加载数据的代码

}

}

@Override

public void onScroll(AbsListView view, int firstVisibleItem,

int visibleItemCount, int totalItemCount) {

this.visibleItemCount = visibleItemCount;

visibleLastIndex = firstVisibleItem + visibleItemCount - 1; //最后的可视项索引

//如果所有的记录选项等于数据集的条数，则移除列表底部视图

if(totalItemCount == dataSize+1){

listView.removeFooterView(loadMoreView);

Toast.makeText(MyShareList.this, "数据全部加载完!", Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 查询数据库中的数据

\*/

private JSONArray loadDataFromDataBase(String loadDataUrl){

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

try {

URL url = new URL(loadDataUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

if(stringBuilder.toString()!="null"){

//返回测试信息

JSONArray jsonArray = new JSONArray(stringBuilder.toString());

//获取到的数据，对Json进行解析

page = page+1; //一次成功请求后更新请求页面

buffer.close();

return jsonArray;

}else{

return null;

}

}else{

Toast.makeText(MyShareList.this,"非200", Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(MyShareList.this, "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

return null;

}

//初始化将详情设置到FindInfo bean中

public List<FindInfo> initSetDataToBean(String detail){

List<FindInfo> findInfo = new ArrayList<FindInfo>();

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindInfo items = new FindInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

Integer infoId = temp.getInt("infoId"); //内容ID

String infoTitle = temp.getString("infoTitle"); //内容标题

String infoDescribe = temp.getString("infoDescribe"); //内容简述

String infoDetail = temp.getString("infoDetail"); //内容详情

Integer type = temp.getInt("infoType"); //类型：0表示日记，1表示趣事

Integer support = temp.getInt("infoSupport"); //点赞数

String infoAuthor = temp.getString("infoAuthor"); //作者

items.setInfoId(infoId);

items.setInfoTitle(infoTitle);

items.setInfoDescribe(infoDescribe);

items.setInfoDetail(infoDetail);

items.setType(type);

items.setSupport(support);

items.setInfoAuthor(infoAuthor);

findInfo.add(items);

}

return findInfo;

}catch (JSONException e){

Toast.makeText(MyShareList.this, "initSetDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

return null;

}

}

//加载更多将详情设置到FindInfo bean中

public void loadMoreSetDataToBean(String detail){

try {

JSONArray detailJsonArray = new JSONArray(detail);

for (int i = 0; i < detailJsonArray.length(); i++) {

FindInfo items = new FindInfo();

JSONObject temp = (JSONObject) detailJsonArray.get(i);

Integer infoId = temp.getInt("infoId"); //内容ID

String infoTitle = temp.getString("infoTitle"); //内容标题

String infoDescribe = temp.getString("infoDescribe"); //内容简述

String infoDetail = temp.getString("infoDetail"); //内容详情

Integer type = temp.getInt("infoType"); //类型：0表示日记，1表示趣事

Integer support = temp.getInt("infoSupport"); //点赞数

String infoAuthor = temp.getString("infoAuthor"); //作者

items.setInfoId(infoId);

items.setInfoTitle(infoTitle);

items.setInfoDescribe(infoDescribe);

items.setInfoDetail(infoDetail);

items.setType(type);

items.setSupport(support);

items.setInfoAuthor(infoAuthor);

adapter.addNewsItem(items); //与初始化是有差异的

}

}catch (JSONException e){

Toast.makeText(MyShareList.this, "loadMoreSetDataToBean异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 初始化ListView的适配器，即打开页面展示的数据

\*/

private void initializeAdapter(){

// 设置线程策略

setVersion();

JSONArray jsonArray = loadDataFromDataBase(requestUrl);

if(jsonArray!=null){

try {

JSONObject totalObject = (JSONObject)jsonArray.get(0);

dataSize = totalObject.getInt("totalRecord"); //总记录数

String detail= totalObject.getString("RecordDetail"); //详情

if(initSetDataToBean(detail)!=null) {

adapter = new MyShareList.PaginationAdapter(initSetDataToBean(detail)); //将详情设置到bean中

}

}catch (JSONException e){

Toast.makeText(MyShareList.this, "initializeAdapter异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}else{

listView.removeFooterView(loadMoreView);

Toast.makeText(MyShareList.this, "查询为空" , Toast.LENGTH\_LONG).show();

}

}

//APP如果在主线程中请求网络操作，将会抛出异常，所以需要用线程来操作网络请求

void setVersion() {

StrictMode.setThreadPolicy(new StrictMode.ThreadPolicy.Builder()

.detectDiskReads()

.detectDiskWrites()

.detectNetwork()

.penaltyLog()

.build());

StrictMode.setVmPolicy(new StrictMode.VmPolicy.Builder()

.detectLeakedSqlLiteObjects() //探测SQLite数据库操作

.penaltyLog() //打印logcat

.penaltyDeath()

.build());

}

/\*\*

\* 加载更多数据

\*/

private void loadMoreData(){

JSONArray jsonArray = loadDataFromDataBase(requestUrl);

try {

JSONObject total = (JSONObject) jsonArray.get(0);

dataSize = total.getInt("totalRecord"); //总记录数

String detail= total.getString("RecordDetail"); //详情

loadMoreSetDataToBean(detail); //将更多的详情设置到bean中

}catch (JSONException e){

Toast.makeText(MyShareList.this, "loadMoreData()异常 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

/\*\*

\* 将一组数据传到ListView等UI显示组件

\*/

class PaginationAdapter extends BaseAdapter {

List<FindInfo> newsItems;

public PaginationAdapter(List<FindInfo> newsitems) {

this.newsItems = newsitems;

}

@Override

public int getCount() {

return newsItems.size();

}

@Override

public Object getItem(int position) {

return newsItems.get(position);

}

@Override

public long getItemId(int position) {

return position;

}

//在这里将Item设置到每个卡片

@Override

public View getView(int position, View view, ViewGroup parent) {

if (view == null) {

view = getLayoutInflater().inflate(R.layout.activity\_find\_list\_item, null);

}

//标题

TextView tvTitle = (TextView) view.findViewById(R.id.cardTitle);

tvTitle.setText(newsItems.get(position).getInfoTitle());

//文章ID

TextView tvId = (TextView) view.findViewById(R.id.cardId);

tvId.setText(newsItems.get(position).getInfoId().toString());

int type = newsItems.get(position).getType();

if (type == 0 || type == 3) {

ImageView cardImgTip = (ImageView) view.findViewById(R.id.cardImgTip);

if (type == 0) {

cardImgTip.setImageResource(R.drawable.share\_funny\_select);

} else {

cardImgTip.setImageResource(R.drawable.share\_diary\_select);

}

TextView tvContent = (TextView) view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.VISIBLE); //显示文字框

ImageView ivContent = (ImageView) view.findViewById(R.id.cardContent\_pic);

ivContent.setVisibility(View.GONE); //设置图片显示框隐藏

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

layoutMusic.setVisibility(View.GONE); //设置音乐显示框隐藏

//分享的普通内容

tvContent.setText(newsItems.get(position).getInfoDescribe());

} else if (type == 1) {

ImageView cardImgTip = (ImageView) view.findViewById(R.id.cardImgTip);

cardImgTip.setImageResource(R.drawable.share\_pic\_select);

//分享图片

TextView tvContent = (TextView) view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.GONE); //隐藏文字框

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

layoutMusic.setVisibility(View.GONE); //设置音乐显示框隐藏

ImageView ivContent = (ImageView) view.findViewById(R.id.cardContent\_pic);

//通过网络链接获取图片

Bitmap one;

String describeUrl = newsItems.get(position).getInfoDescribe().trim();

try {

one = LoadImgByNet.getBitmap(imgBaseUrl + describeUrl); //设置图片

ivContent.setImageBitmap(one);

} catch (IOException e) {

e.printStackTrace();

}

ivContent.setVisibility(View.VISIBLE); //设置图片显示框

} else if (type == 2) {

ImageView cardImgTip = (ImageView) view.findViewById(R.id.cardImgTip);

cardImgTip.setImageResource(R.drawable.share\_music\_select);

//分享的音乐

TextView tvContent = (TextView) view.findViewById(R.id.cardContent);

tvContent.setVisibility(View.GONE); //显示文字框

ImageView ivContent = (ImageView) view.findViewById(R.id.cardContent\_pic);

ivContent.setVisibility(View.GONE); //设置图片显示框隐藏

LinearLayout layoutMusic = (LinearLayout) view.findViewById(R.id.cardContent\_music);

TextView tvMusicContent = (TextView) view.findViewById(R.id.cardTitle\_music);

tvMusicContent.setText(newsItems.get(position).getInfoDescribe());

layoutMusic.setVisibility(View.VISIBLE); //设置音乐显示框显示

}

return view;

}

/\*\*

\* 添加数据列表项

\*

\* @param infoItem

\*/

public void addNewsItem(FindInfo infoItem) {

newsItems.add(infoItem);

}

}

}

------------------------------- **MyShareList.java结束-----------------------**

“更新签名”页——ChangeSignature.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为ChangeSignature，将下面的代码复制到该文件中。ChangeSignature.java主要的介绍如下：

* 显示activity\_change\_signature.xml，从跳转源获得用户的旧签名，设置到页面
* 给布局中的修改按钮添加事件，当点击时获取到用户输入的新签名，并提交到后台，完成更新签名的功能

------------------------- **ChangeSignature.java开始---------------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class ChangeSignature extends Activity {

TextView old\_sign\_tv; //旧签名

EditText new\_sign\_et; //新签名

Button sure\_change\_sign; //确定修改

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

String userName,sign; //存放用户基本信息

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_change\_signature);

old\_sign\_tv = (TextView)findViewById(R.id.old\_user\_sign);

new\_sign\_et = (EditText) findViewById(R.id.et\_login\_password); //新签名

sure\_change\_sign = (Button)findViewById(R.id.bt\_change\_signature) ;//确定修改签名的按钮

setBaseInfo();

sure\_change\_sign.setOnClickListener(new onClick());

}

class onClick implements View.OnClickListener {

@Override

public void onClick(View v) {

String sign = new\_sign\_et.getText().toString();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

if(sign.length()==0){

Toast.makeText(ChangeSignature.this, "签名不能为空", Toast.LENGTH\_LONG).show();

return;

}

try {

String registerUrl = baseUrl+"ChangeSignature?signature="+sign+"&username="+userName;

URL url = new URL(registerUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

changeSuccess(sign);

Toast.makeText(ChangeSignature.this, stringBuilder.toString(), Toast.LENGTH\_LONG).show();

//通过handler设置延时1秒后执行r任务

new Handler().postDelayed(new ChangeSignature.LoadMainTask(),500);

buffer.close();

}else{

Toast.makeText(ChangeSignature.this,"非200.."+connGET.getResponseCode() , Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(ChangeSignature.this, "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

//启动线程,加载我的页面

private class LoadMainTask implements Runnable{

@Override

public void run() {

Intent intent = new Intent(ChangeSignature.this,MyTabFragment.class);

//startActivity(intent);

intent.putExtra("newSignature",sign);

setResult(1, intent); //这里的1就对应到onActivityResult（）方法中的resultCode

finish();

}

}

//修改签名后进行用户签名修改

public void changeSuccess(String sign){

SharedPreferences sp = ChangeSignature.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sp.edit();

editor.putString("sign", sign);

editor.commit();

}

//设置修改签名页面用户相关信息

public void setBaseInfo(){

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = ChangeSignature.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

userName = sp.getString("name","");

sign = sp.getString("sign","");

//设置旧签名

old\_sign\_tv.setText(sign);

}

}

--------------------------- **ChangeSignature.java结束-------------------------**

“更改密码”页——ChangePassword.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为ChangePassword，将下面的代码复制到该文件中。ChangePassword.java主要的介绍如下：

* 显示activity\_change\_password.xml，给布局中的修改按钮添加点击事件，点击时获取到用户的昵称、用户输入的旧密码和新密码，发送到后台实现修改密码的功能
* 密码修改成功时返回“我的”页面

------------------------- **ChangePassword.java开始-------------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

public class ChangePassword extends Activity {

EditText old\_pass\_et; //旧密码

EditText new\_pass\_et; //新密码

Button sure\_change\_pass; //确定修改

String baseUrl = "http://10.0.2.2:8080/iShareService/servlet/"; //web服务器的地址

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

String userName,oldPass,newPass; //存放用户基本信息

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_change\_password);

old\_pass\_et = (EditText) findViewById(R.id.et\_old\_login\_password); //新密码

new\_pass\_et = (EditText) findViewById(R.id.et\_new\_login\_password); //新密码

sure\_change\_pass = (Button)findViewById(R.id.bt\_change\_password) ;//确定修改密码的按钮

setBaseInfo();

sure\_change\_pass.setOnClickListener(new ChangePassword.onClick());

}

class onClick implements View.OnClickListener {

@Override

public void onClick(View v) {

oldPass = old\_pass\_et.getText().toString();

newPass = new\_pass\_et.getText().toString();

StringBuilder stringBuilder = new StringBuilder();

BufferedReader buffer = null;

HttpURLConnection connGET = null;

if(oldPass.length()==0||newPass.length()==0){

Toast.makeText(ChangePassword.this, "密码不能为空", Toast.LENGTH\_LONG).show();

return;

}

try {

String registerUrl = baseUrl+"ChangePassword?oldPass="+oldPass+"&newPass="+newPass+"&username="+userName;

URL url = new URL(registerUrl);

connGET = (HttpURLConnection) url.openConnection();

connGET.setConnectTimeout(5000);

connGET.setRequestMethod("GET");

if (connGET.getResponseCode() == 200) {

buffer = new BufferedReader(new InputStreamReader(connGET.getInputStream()));

for (String s = buffer.readLine(); s != null; s = buffer.readLine()) {

stringBuilder.append(s);

}

Toast.makeText(ChangePassword.this, stringBuilder.toString(), Toast.LENGTH\_LONG).show();

//通过handler设置延时1秒后执行r任务，跳转到我的页面

new Handler().postDelayed(new ChangePassword.LoadMainTask(),500);

buffer.close();

}else{

Toast.makeText(ChangePassword.this,"非200.."+connGET.getResponseCode() , Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

e.printStackTrace();

Toast.makeText(ChangePassword.this, "get 提交 err.." + e.toString(), Toast.LENGTH\_LONG).show();

}

}

}

//启动线程,加载我的页面

private class LoadMainTask implements Runnable{

@Override

public void run() {

Intent intent = new Intent(ChangePassword.this,MyTabFragment.class);

/\*startActivity(intent);\*/

setResult(2, intent); //这里的2就对应到onActivityResult（）方法中的resultCode

finish();

}

}

//获取用户信息

public void setBaseInfo(){

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = ChangePassword.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

userName = sp.getString("name","");

}

}

---------------------------- **ChangePassword.java结束--------------------------**

“账号管理”页——AccountManagement.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为AccountManagement，将下面的代码复制到该文件中。AccountManagement.java主要的介绍如下：

* 显示activity\_account\_manage.xml，给布局中的退出登录按钮添加点击事件，点击时清空本地存储的用户信息
* 退出成功时，返回到“我的”页面并改变页面UI，显示登录注册按钮

--------------------- **AccountManagement.java开始--------------------------**

package com.example.discover;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.SharedPreferences;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.TextView;

import android.widget.Toast;

import java.io.IOException;

public class AccountManagement extends Activity {

ImageView my\_head\_img; //用户头像

TextView my\_name,status; //用户昵称和心情

Button out\_login; //退出登录

String imgBaseUrl = "http://10.0.2.2:8080/iShareService/images/"; //图片资源

String userName,headImg; //存放用户基本信息

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_account\_manage);

my\_head\_img = (ImageView)findViewById(R.id.account\_images\_head); //我的头像

my\_name = (TextView)findViewById(R.id.account\_t\_name); //我的昵称

status = (TextView)findViewById(R.id.account\_t\_status); //登录状态

out\_login = (Button)findViewById(R.id.account\_bt\_out\_login) ;//退出登录的按钮

setBaseInfo();

out\_login.setOnClickListener(new onClick());

}

class onClick implements View.OnClickListener {

@Override

public void onClick(View v) {

exitSuccess();

Toast.makeText(AccountManagement.this, "退出成功！", Toast.LENGTH\_LONG).show();

//通过handler设置延时1秒后执行r任务，跳转到首页

new Handler().postDelayed(new AccountManagement.LoadMainTask(),1000);

}

}

//启动线程,加载主页

private class LoadMainTask implements Runnable{

@Override

public void run() {

Intent intent = new Intent(AccountManagement.this,MainActivity.class);

startActivity(intent); //打开主页

finish(); //关闭

}

}

//退出登录后进行用户登录状态修改

public void exitSuccess(){

//退出登录后更改登录状态

SharedPreferences sp = AccountManagement.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

SharedPreferences.Editor editor = sp.edit();

editor.putBoolean("isLogin", false);

editor.putString("name", "");

editor.putString("sign", "");

editor.putString("img", "");

editor.commit();

}

//设置账号管理页面用户相关信息

public void setBaseInfo(){

//获取最新的登录状态，SharePreferences为永久存储，需要手动退出

SharedPreferences sp = AccountManagement.this.getSharedPreferences("save", Context.MODE\_PRIVATE);

//boolean isLogin = sp.getBoolean("isLogin",false); //第一版本不做登录判断，在跳转前要求登录状态才能够跳转到这个页面

status.setText("已登录");

Bitmap one; //用于获取网络图片

userName = sp.getString("name","");

headImg = sp.getString("headImg","");

//设置用户昵称、头像

my\_name.setText(userName);

//通过网络链接获取图片

try {

one= LoadImgByNet.getBitmap(imgBaseUrl+headImg);

my\_head\_img.setImageBitmap(one); //设置用户头像

}catch(IOException e){

e.printStackTrace();

}

}

}

----------------------- **AccountManagement.java结束--------------------------**

“关于i分享”—— AboutIShare.java

打开项目Discover，找到app>src>main>java>com.example> discover，在discover中创建Class，命名为AboutIShare，将下面的代码复制到该文件中。AboutIShare.java主要的介绍如下：

* 显示activity\_about\_ishare.xml，这部分只是简单地显示应用相关信息

------------------------------ **AboutIShare.java开始---------------------------**

package com.example.discover;

import android.app.Activity;

import android.os.Bundle;

public class AboutIShare extends Activity {

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about\_ishare);

}

}

-------------------------- **AboutIShare.java结束---------------------**

项目配置

AndroidManifest.xml

打开项目Discover，找到app>src>main，打开文件夹main，找到AndroidManifest.xml，将下面的配置代码复制到该文件中。需要跳转到、或者需要用到组件都需要在这个文件内进行配置，否则会因为找不到而报错。

---------------------------**-AndroidManifest.xml开始-------------------------**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.discover">

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme"

android:usesCleartextTraffic="true">

<activity

android:name=".SplashActivity"

android:label="@string/app\_name"

android:theme="@android:style/Theme.NoTitleBar">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name=".MainActivity"

android:label="i分享">

</activity>

<activity android:name=".AccountManagement"

android:label="账号管理">

</activity>

<activity android:name=".ChangeSignature"

android:label="更新签名">

</activity>

<activity android:name=".ChangePassword"

android:label="更改密码">

</activity>

<activity android:name=".MyShareList"

android:label="关于我的">

</activity>

<activity android:name=".DisplayDetail"

android:label="查看详情">

</activity>

<activity android:name=".AboutIShare"

android:label="iShare介绍">

</activity>

</application>

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.INTERNET"/>

</manifest>

------------------------**-AndroidManifest.xml结束------------------------------**

1. 应用测试

应用测试之前，需要将myEclipse中的项目iShareService上传到Tomcat服务器，并确保启动服务，另外还需要net start mysql启动mysql数据库。

“启动”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“启动页”，在Android端需要好确保准备好以下几个部分：

* splash.xml创建完成（4-3-1）
* SplashActivity.java创建完成（4-6-1）
* main\_tab .xml创建完成（4-3-2）
* AndroidManifest.xml（4-7-1）对应SplashActivity、MainActivity配置完成
* MainActivity.java（4-6-2）创建完成，且只需要保留setContentView就可以看到启动页效果。



“主页”

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“主页”，在Android端需要好确保准备好以下几个部分：

* main\_tab.xml创建完成，具体看4-3-1
* discover\_tab\_content.xml（4-3-3）、find\_tab\_content.xml（4-3-8）、edit\_tab\_content.xml（4-3-10）、my\_tab\_content.xml（4-3-11）创建完成
* MainActivity.java（4-6-2）创建完成
* DiscoverTabFragment.java（4-6-3）、FindTabFragment.java（4-6-5）、EditTabFragment.java（4-6-6）、 MyTabFragment. java（4-6-7）创建完成，且代码只需包含onCreateView



“发现”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“发现页”，在Android端需要好确保准备好以下几个部分：

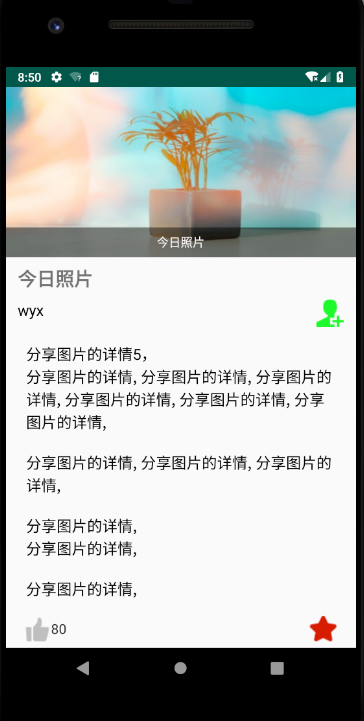
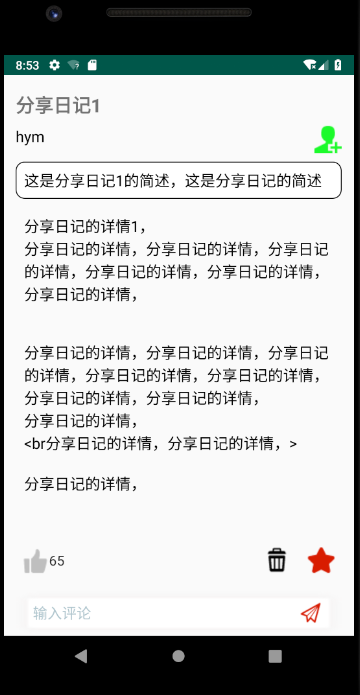
* 主页已经实现（5-2）
* discover\_tab\_content.xml（4-3-3）、activity\_find\_list\_item.xml（4-3-4）、activity\_find\_loadmore.xml（4-3-5）、activity\_info\_detail.xml（4-3-6）创建完成
* DiscoverTabFragment.java创建完成（4-6-3）
* DisplayDetail.java（4-6-4）创建完成，且只需要保留setContentView
* AndroidManifest.xml（4-7-1）对应DisplayDetail配置完成



“详情”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“详情页”，在Android端需要好确保准备好以下几个部分：

* “发现页”已经实现（5-3）
* activity\_comment\_list.xml（4-3-7）创建完成
* DisplayDetail.java创建完成（4-6-3）

“查找”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“查找页”，在Android端需要好确保准备好以下几个部分：

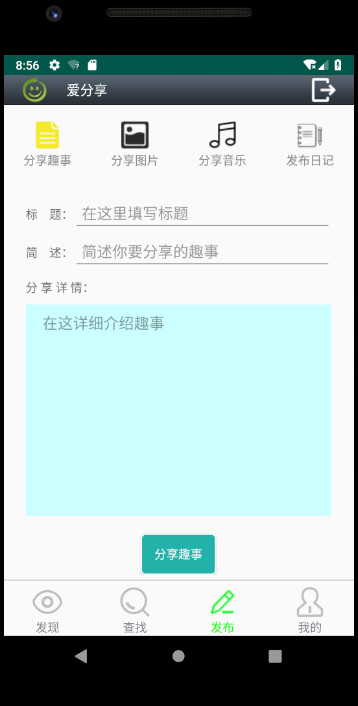
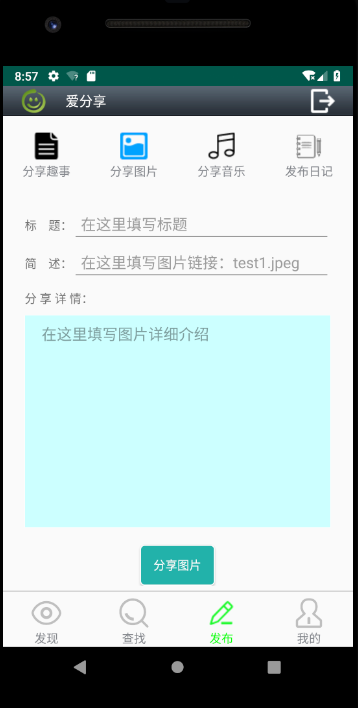
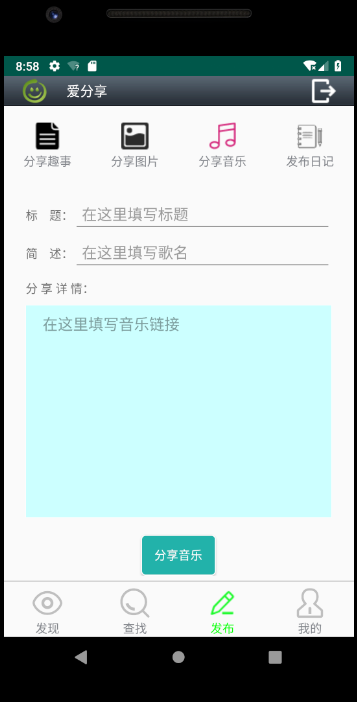
* 主页已经实现（5-2）
* activity\_seek\_list\_item.xml（4-3-4）、activity\_info\_detail.xml（4-3-6）、find\_tab\_content.xml（4-3-8）创建完成
* FindTabFragment.java（4-6-5）创建完成
* DisplayDetail.java（4-6-4）创建完成，且只需要保留setContentView
* AndroidManifest.xml（4-7-1）对应DisplayDetail配置完成

“发布”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“发布页”，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* edit\_tab\_content.xml（4-3-10）创建完成
* EditTabFragment.java（4-6-6）创建完成

“我的”页

在数据库设计完整、服务器环境搭建好、Android环境配置完成、完整引入资源的情况下运行“我的”页，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* my\_tab\_content.xml（4-3-11）创建完成
* MyTabFragment.java（4-6-7）创建完成，且只需要保留setContentView

“注册”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* activity\_register.xml（4-3-12）创建完成
* MyTabFragment.java（4-6-7）添加注册按钮事件



“登录”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

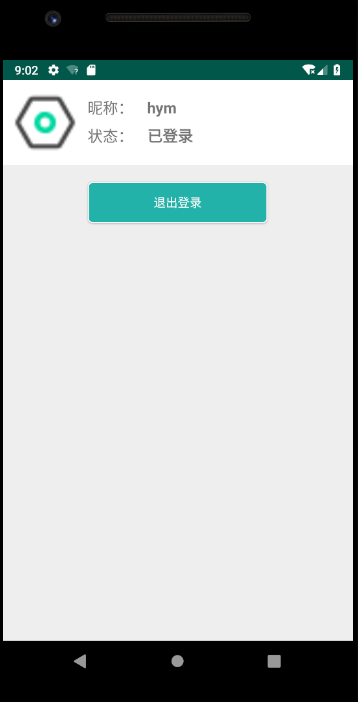
* 主页已经实现（5-2）
* activity\_login.xml（4-3-13）创建完成
* MyTabFragment.java（4-6-7）添加登录按钮事件



“账号管理”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

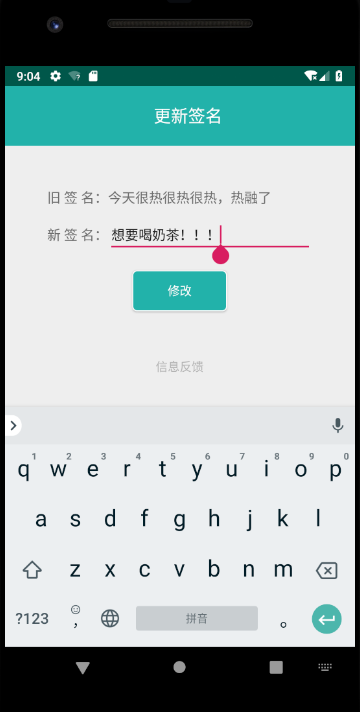
* 主页已经实现（5-2）
* activity\_account\_manage.xml（4-3-14）创建完成
* MyTabFragment.java（4-6-7）添加“账号管理”按钮事件
* AccountManagement.java（4-6-11）创建完成



“更新签名”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* activity\_change\_signature.xml（4-3-15）创建完成
* MyTabFragment.java（4-6-7）添加“更新签名”按钮事件
* ChangeSignature.java（4-6-9）创建完成
* AndroidManifest.xml（4-7-1）对应ChangeSignature配置完成

“更改密码”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* activity\_change\_password.xml（4-3-16）创建完成
* MyTabFragment.java（4-6-7）添加“修改密码”按钮事件
* ChangePassword.java（4-6-10）创建完成
* AndroidManifest.xml（4-7-1）对应ChangePassword配置完成



“关于我的”页

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* activity\_about\_me.xml（4-3-17）创建完成
* MyTabFragment.java（4-6-7）添加“关于我的”下四个按钮的事件
* MyShareList.java（4-6-8）创建完成
* AndroidManifest.xml（4-7-1）对应MyShareList配置完成

“关于i分享”

在 “我的”页已经运行成功之后，在Android端需要好确保准备好以下几个部分：

* 主页已经实现（5-2）
* activity\_about\_ishare.xml（4-3-18）创建完成
* MyTabFragment.java（4-6-7）添加“关于i分享”按钮事件
* AboutIShare.java（4-6-12）创建完成
* AndroidManifest.xml（4-7-1）对应AboutIShare配置完成



1. 资源链接

百度网盘：<https://pan.baidu.com/s/1yJHbLxlqFnS2jD_8-T_cow>

提取码：29y1

CSDN博客链接：<https://blog.csdn.net/YanMeiHe>