## 活动详细信息的实现

1. 新建activityInformation.java类，用来存放一个活动的所有信息，并且这个类一定要继承Serializable这个接口

|  |
| --- |
| **public class** activityInformation **implements** Serializable {  String **credit**;*//学分* String **name**;*//标题* String **createTime**;*//发布时间* String **starttime**;*//活动开始时间* String **endtime**;*//活动结束时间* String **looknum**;*//点击量* String **maxnum**;*//限量预约* String **reservationnum**;*//已预约人数* String **address**;*//地点* String **introduce**;*//活动精度* String **tea\_pic**;*//导师照片* String **tea\_name**;*//导师名字* String **ste\_introcedouce**;*//地点* String **picture\_url**;*//活动图片* String **code**;*//活动进行情况*  }  全部都要生成get set方法 |

1. 给轮播图添加点击事件

|  |  |
| --- | --- |
| 在HomeFragment.java添加变量：   |  | | --- | | String **idd**;*//活动的id* String **imgsrc**;*//活动的图片url* HomeClass **homeClass**=**new** HomeClass(**this**); activityInformation **activityInformation**;*//一个活动的详细信息* |   在public View onCreateView这个方法里实现轮播图的点击事件  **banner**.setOnBannerListener(**new** OnBannerListener() {  @Override  **public void** OnBannerClick(**int** position) {  **homeClass**.show(**numList**.get(position)+**""**);//这个是通过活动的id找到他的信息信息  **imgsrc**=**imageList**.get(position);//得到活动的图片链接  **idd**=**numList**.get(position)+**""**;//得到活动id  } }); |

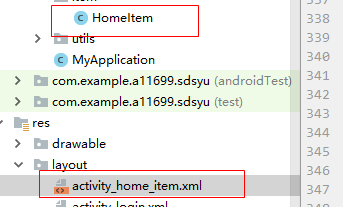
1. 新建类HomeClass.java

|  |
| --- |
| 定义变量：  activityInformation **activityInformation** = **new** activityInformation(); HomeFragment **homeFragment**;  **public** HomeClass(HomeFragment homeFragment) {  **this**.**homeFragment** = homeFragment; }  写show(String aa\_id)方法  *//访问网络拿到数据* **public void** show(String aa\_id) {  **final** Handler handlger=**new** Handler(){  @Override  **public void** handleMessage(Message msg) {  **super**.handleMessage(msg);  **if**(msg.**what**==0x01){  Log.*i*(**"zjc"**,**"我执行了"**);  //执行HomeFragment里的回调函数  **homeFragment**.shoee(**activityInformation**);  }  }  };  **final** String url = Net.***S\_ACTIVITY*** + **"?act\_id="** + aa\_id + **"&s\_id="** + MyApplication.*getS\_id*();  Log.*i*(**"zjc"**, url);  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  OkHttpClient okHttpClient = **new** OkHttpClient();  **final** Request request = **new** Request.Builder().url(url).build();*//访问不同的url在这改* Call call = okHttpClient.newCall(request);  call.enqueue(**new** Callback() {  @Override  **public void** onFailure(Call call, IOException e) {   Log.*i*(**"zjc"**, **"里面的：网络请求失败"**);  }   @Override  **public void** onResponse(Call call, Response response) **throws** IOException {  Log.*i*(**"zjc"**, **"网络请求好了"**);  String responseData = response.body().string();  **try** {  JSONObject jsonObject = **new** JSONObject(responseData);   **activityInformation**.setCreateTime(jsonObject.getString(**"createtime"**));  **activityInformation**.setAddress(jsonObject.getString(**"address"**));  **activityInformation**.setEndtime(jsonObject.getString(**"endtime"**));  **activityInformation**.setCredit(jsonObject.getString(**"credit"**));  **activityInformation**.setIntroduce(jsonObject.getString(**"introduce"**));  **activityInformation**.setLooknum(jsonObject.getString(**"looknum"**));  **activityInformation**.setName(jsonObject.getString(**"name"**));  **activityInformation**.setMaxnum(jsonObject.getString(**"maxnum"**));  **activityInformation**.setReservationnum(jsonObject.getString(**"reservationnum"**));  **activityInformation**.setTea\_pic(jsonObject.getString(**"tea\_pic"**));  **activityInformation**.setTea\_name(jsonObject.getString(**"tea\_name"**));  **activityInformation**.setStarttime(jsonObject.getString(**"starttime"**));  **activityInformation**.setAddress(jsonObject.getString(**"address"**));  **activityInformation**.setPicture\_url(jsonObject.getString(**"picture\_url"**));  **activityInformation**.setCode(jsonObject.getString(**"code"**));  handlger.sendEmptyMessage(0x01);  Log.*i*(**"zjc"**, **"我出来了"**);  } **catch** (JSONException e) {  e.printStackTrace();  }  }  });  }  }).start(); } |

1. 在HomeFragment添加回调shoee(activityInformation)回调函数

|  |
| --- |
| **public void** shoee(activityInformation activityInformation){  **this**.**activityInformation**=activityInformation;  Intent intent=**new** Intent(getActivity(), HomeItem.**class**);  intent.putExtra(**"act\_id"**,**idd**);//传递活动的id  intent.putExtra(**"ac"**,activityInformation);//传递整个活动信息  intent.putExtra(**"imgsrc"**,**imgsrc**);//传递活动的图片url  startActivity(intent); }; |

1. 新建活动HomeItem



|  |
| --- |
| 写方法ss(activityInformation activityInfomation)  **public void** ss(activityInformation activityInformation) {  **this**.**activityInformationn**=activityInformation;  **event\_sfl**.setRefreshing(**false**);  **event\_name\_tv**.setText(**activityInformationn**.getName());  **event\_createtime\_tv**.setText(**activityInformationn**.getCreateTime());  **event\_looknum\_tv**.setText(**activityInformationn**.getLooknum());  **event\_starttime\_tv**.setText(**activityInformationn**.getStarttime());  **event\_endtime\_tv**.setText(**activityInformationn**.getEndtime());  **event\_address\_tv**.setText(**activityInformationn**.getAddress());  **event\_reservationnum\_tv**.setText(**activityInformationn**.getReservationnum());  **event\_maxnum\_tv**.setText(**activityInformationn**.getMaxnum());  **event\_credit\_tv**.setText(**activityInformationn**.getCredit());  **event\_tea\_name\_tv**.setText(**activityInformationn**.getTea\_name());  **event\_tea\_introduce\_tv**.setText(**activityInformationn**.getIntroduce());  **event\_introduce\_tv**.setText(**activityInformationn**.getSte\_introcedouce());  String status=**null**;  **switch** (activityInformation.getCode()){  **case "1"**:status=**"取消预约"**;**break**;  **case "2"**:status=**"预约活动"**;**break**;  **case "3"**:status=**"不可取消"**;**break**;  **case "4"**:status=**"不可预约"**;**break**;  **case "5"**:status=**"已经签到"**;**break**;  **case "6"**:status=**"已经开始，扫码进行签到"**;**break**;  **case "7"**:status=**"等待老师给分"**;**break**;  **case "8"**:status=**"已经给分"**;**break**;  **case "9"**:status=**"活动已开始"**;**break**;  **case "10"**:status=**"活动已结束"**;**break**;  **case "11"**:status=**"第一次活动结束，第二次还未开始"**;**break**;  **default**:**break**;  }  **event\_appoint\_tv**.setText(status);  List<String> imglist=**new** ArrayList<>();  imglist.add(**imgsrc**);  **event\_banner**.setBannerAnimation(Transformer.*Default*);  **event\_banner**.setImages(imglist);  **event\_banner**.setImageLoader(**new** GlideImageLoader());  **event\_banner**.start();  **event\_banner**.setOnBannerListener(**new** OnBannerListener() {  @Override  **public void** OnBannerClick(**int** position) {  Log.*i*(**"zjc"**,**"我被点了"**);  }  });image(activityInformation.getTea\_pic());  Log.*i*(**"zjc"**,**"活动图片"**+activityInformation.getPicture\_url()+**""**);  Log.*i*(**"zjc"**,**"kl老师图片"**+activityInformation.getTea\_pic()+**""**); }  加载网络图片的方法：  **public void** image(String url){  OkHttpClient okHttpClient=**new** OkHttpClient();  Request request=**new** Request.Builder().url(url).build();  Call call=okHttpClient.newCall(request);  call.enqueue(**new** Callback() {  @Override  **public void** onFailure(Call call, IOException e) {   }  @Override  **public void** onResponse(Call call, Response response) **throws** IOException {  **byte**[] Picture\_bt=response.body().bytes();  Log.*i*(**"zjc"**,Picture\_bt.toString());  Message message=**handler**.obtainMessage();  message.**obj**=Picture\_bt;  message.**what**=1;   **handler**.sendMessage(message);  }  }); }  Handler **handler**=**new** Handler(){  @Override  **public void** handleMessage(Message msg) {  **super**.handleMessage(msg);  **if**(msg.**what**==1){  **byte** []Pi=(**byte**[])msg.**obj**;  Bitmap bitmap=BitmapFactory.*decodeByteArray*(Pi,0,Pi.**length**);  **event\_tea\_pic\_iv**.setImageBitmap(bitmap);  }  } };  添加下拉刷新功能  **event\_sfl**.setOnRefreshListener(**new** SwipeRefreshLayout.OnRefreshListener() {  @Override  **public void** onRefresh() {  showinfor(a\_id);   } });  写方法showinfor(String a\_id);  **public void** showinfor(String aa\_id){  **final** String url = Net.***S\_ACTIVITY*** + **"?act\_id="** + aa\_id + **"&s\_id="** + MyApplication.*getS\_id*();  Log.*i*(**"zjc"**, url);  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  OkHttpClient okHttpClient = **new** OkHttpClient();  **final** Request request = **new** Request.Builder().url(url).build();*//访问不同的url在这改* Call call = okHttpClient.newCall(request);  call.enqueue(**new** Callback() {  @Override  **public void** onFailure(Call call, IOException e) {   Log.*i*(**"zjc"**, **"里面的：网络请求失败"**);  }  @Override  **public void** onResponse(Call call, Response response) **throws** IOException {  Log.*i*(**"zjc"**, **"网络请求好了"**);  String responseData = response.body().string();  **try** {  JSONObject jsonObject = **new** JSONObject(responseData);   **activityInformationn**.setCreateTime(jsonObject.getString(**"createtime"**));  **activityInformationn**.setAddress(jsonObject.getString(**"address"**));  **activityInformationn**.setEndtime(jsonObject.getString(**"endtime"**));  **activityInformationn**.setCredit(jsonObject.getString(**"credit"**));  **activityInformationn**.setIntroduce(jsonObject.getString(**"introduce"**));  **activityInformationn**.setLooknum(jsonObject.getString(**"looknum"**));  **activityInformationn**.setName(jsonObject.getString(**"name"**));  **activityInformationn**.setMaxnum(jsonObject.getString(**"maxnum"**));  **activityInformationn**.setReservationnum(jsonObject.getString(**"reservationnum"**));  **activityInformationn**.setTea\_pic(jsonObject.getString(**"tea\_pic"**));  **activityInformationn**.setTea\_name(jsonObject.getString(**"tea\_name"**));  **activityInformationn**.setStarttime(jsonObject.getString(**"starttime"**));  **activityInformationn**.setAddress(jsonObject.getString(**"address"**));  **activityInformationn**.setPicture\_url(jsonObject.getString(**"picture\_url"**));  *// activityInformation.setSte\_introcedouce(jsonObject.getString("introcedouce"));  //handler.sendEmptyMessage(12);* Log.*i*(**"zjc"**, **"我出来了"**);  } **catch** (JSONException e) {  e.printStackTrace();  }  }  });  }  }).start();  ss(**activityInformationn**); } |