**以下接口均在RequestAddress文件中**

1. **页面登录模块**

**1.1、StartActivity为启动页面,用来显示App启动动画5秒。**

①、在initData（）方法中，通过UserUtils工具类中的getIsFirstLoad()方法来从sharepreference中获取ISFIRST值，若是第一次安装登陆，则跳转至SplashActivity界面，该活动包含一个ViewPaper控件，用来展示介绍App的三张本地图片。若不是第一次登陆，通过StartRequest的start方法来请求广告页图片。

②StartRequest继承自ProjectRequest，ProjectRequest中包含RequestBuilder对象，通过startRequest的start方法对RequestBuilder对象的公共请求参数，请求地址，监听器参数赋值，ProjectRequest的构造方法来实现对RequestBuilder对象的请求方法、请求地址各个参数赋值，start方法中通过create()方法创建BaseRequest对象，通过其execute()方法来执行请求图片的操作。

接口：

③在onsuccess的回调方法中，通过Gson解析类型为StartBean返回的response数据，startbean对象中的checkversionbean对象如果不为空，则给该页面中的ImageView控件加载图片，若返回的错误码为401，则tokenLost为true，用户未登录。

④在广告5秒后或者点击跳过按钮，则会调用gotoLogin()方法，在该方法中，通过判断tokenLost值，若为true则用户未登录，跳转至LoginActivity，否则用户已登录，通过UserUtils工具类的getUser()方法来从sharepreference中获取字段为USER的用户信息，并通过Gson解析为UserBean对象，若该对象为空或者isTourist()，则userbean对象赋值为null,否则如果userbean对象的getClub()不为空则直接跳转至HomeUI主界面，否则跳转至LoginActivity界面。

**1.2、LoginActivity为登录页面（监听器及内容如何传入的）**

①对电话框et\_tel和验证码框et\_code进行监听，addTextChangedListener(new TextWatcher)，TextWatcher的afterTextChanged()中，将电话框中的×号按钮显示，其点击事件将编辑框清空；当电话框 长度为11并且验证码长度大于4时，并将登录框setEnable（true），并改变背景资源为黄色，否则为false，将其背景改为灰色，且不可按动。

②点击获取验证码后，当点击获取验证码时间和当前时间相差60秒时，

通过GetCodeRequest对象的getCode(tel)方法来获取验证码，否则弹出在1分钟后操作的弹框。

接口：

③GetCodeRequest继承自ProjectRequest类，其构造方法对ProjectRequest的url、监听器参数进行赋值，getCode方法传入params参数，GET请求，通过create方法创建RequestBuilder对象，通过该对象的execute方法来执行网络请求。在LoginAct0ivity的success回调方法中，当requestCode为CODE\_GETCODE时，Gson解析格式为BaseBean的response返回值，toast该basebean对象的getMessage()信息。

④点击登录按钮tv\_login，当输入的手机号长度为11且验证码不为空长度不超过6时，调用LoginRequest对象的login方法，传入手机号、验证码、设备ID来进行验证。该PushID通过UserUtils类的getPushToken方法从sharepreference中读取PUSHTOKEN。

LoginActivity通过LoginRequest接口来获取用户名、token、验证码，在LoginAcitiviy中进行验证登录，验证成功后跳转HomeUI

1. 获取验证码：

LoginActivity中创建GetCodeRequest对象，通过其getCode(tel)方法来获取验证码，GetCodeRequest对象继承自ProjectRequest,通过requestbuilder对象的setParams()setType()setUrl()等方法来设置信息，获取验证码，这些都是写好的。具体的实现在BaseRequest()的execute()方法中。

4、

**2、主页面模块**

**2.1、HomeUI为主页**

布局内含有FragmentLayout容器和自定义布局BottomBar。

HomeUI extends BaseUI,

①initView()

在initView()中，通过框架Fragmentation的loadMultipleRootFragment（）来加载主页，宗亲汇、我的三个fragment，通过BottomBar控件的addItem(new BottomBarTab())方法来添加底部导航栏控件，并通过setOnTabSelectedListener()注册监听器，通过onTableSelected()来监听按钮的切换，在onTabSelected()回调方法中，通过showHideFragment()方法来进行三个碎片界面的切换。当切换第二个消息界面时，通过UserUtils工具类的getUser方法从sharepreference中获取UserBean对象，通过isTourst()方法来判断是否为游客，当是游客时，通过MsgDialog的showLoginHint()方法弹出还未登陆的dialog对话框，并setOnCancelClickListener()建立监听，在onCancleClick()回调方法中，隐藏该dialog框，并将当前界面转至firstfragment。在onTabReselected()方法中：？？？

②initData()

首先通过getIntent()方法获取到从LoginActivity中传来的msg积分值，然后通过showIntegralHint(msg)方法来显示获取积分的dialog框。

通过IMUserP对象的login()方法来实现腾讯云的登陆，在imLoginSuccess()回调方法中，通过PushUtil的getInstance()方法实现初始化后台消息推送，在imLoginFail()的回调方法中会直接弹出MsgDialog框显示版本更新需要重新登录，点击后通过调用logout()方法跳转至LoginActivity界面。

通过StartRequest对象的start()方法来获取启动页信息，接口：URL\_START 在success()回调方法中，解析数据类型为StartBean的response字符串，通过startbean的getData()方法来获取CheckVersionBean对象，通过UserUtils工具类从sharepreference中获取本地的CheckVersionBean对象，通过StringUtils工具的compareVersion()方法来进行比较，若返回值为1，？？通过showUpdateDialog()方法来显示软件更新对话框，在该方法中调动downLoadApk()方法来下载，downloadApk()方法中，通过DownloadManagerUtils工具类的downloadApk()方法来下载，并注册广播监听器，当下载完成后，在DownloadReceiver中，同DownloadManagerUtils的installApk()方法来进行安装。

以下三个Fragment均是采用MVP模式，Model层完成具体的操作，Presenter调用model层，View层即Activity调用P层来获取数据后展现

**2.2、FirstFragment为首页界面 BannerBean**

页面布局包含一个RecycleView控件，其外层包裹一个SmartRefreshLayout布局，用来实现下来刷新和上拉加载的功能。

①initData()

调用initRecyclerView()方法，来初始化头部布局，包含轮播图、同宗活动、寄思先祖、企业秀、TextView同宗动态。

通过FirstP对象调用requestToBanner()方法请求轮播图，具体实现是FirstModel中的getBanner()方法来请求，接口：URL\_GETBANNER，请求在FirstP中回调方法bannerSuccess()内给bannerList赋值，同时调用IFirstFM接口中的showbanner()方法来展图片。showbanner()方法的具体实现是在FirstFragment中。

调用其requestToListRefresh()方法请求同宗动态数据。具体实现是在FirstModel中的getListRefresh()方法，接口：URL\_GETHOMEDYNAMIC，请求回调结果在FirstP中的回调方法listSuccess()中，通过loadMore()方法给dynamics同宗动态集合赋值，在FirstFragment里为适配器初始化时通过firstP的getListData()方法来得到这一集合。

②initRecycleView()

创建DynamicAdapter()同宗动态信息填充的适配器，item子布局为item\_dynamic，数据通过FirstP对象的 getListData()方法获取。

通过适配器的.addHeaderView()方法来添加头部视图，头部视图通过getHeadView()方法来加载，其调用initHeadView()方法，在该方法中并为其中的“同宗活动”、“寄思先祖”、“企业秀”三个ImageButton注册监听器。并设立setOnPicItemClickListener()点击事件，当点击图片的时候，调用openImag()方法来查看大图，该方法是通过跳转至ImageActivity活动来展示的。（具体的ImageActivity待解析...DynamicAdater适配器讲解）

③onClickBanner() 轮播图点击事件

通过firstP对象的getBanner().get(position)方法来获取当前位置轮播图的BannerBean.DataBean 信息，通过getType()判读其实跳转内部链接还是网页，为1时通过databean的getUrl()方法获取跳转链接，然后跳转至WebLoadActivity活动，通过其webview来加载跳转链接。

④onClick() 控件点击事件

点击rl\_image 轮播图时，调用onClickBanner()方法

点击ib\_back 通知（铃铛按键）时，跳转至MessageActivity

点击ib\_huodong 同宗活动时，跳转至ActionListUI

点击ib\_xiangzu 寄思先祖时，跳转至AncestorActivity

点击ib\_qiye 企业时，跳转至ShowListUI

点击tv\_right 发布时，跳转至IssueDynamicActivity

⑤onActivityResult() 进入同宗动态详情页，当点赞或评论后返回则给该FirstFragment返回数据，通过FirstP的upDate()方法来更新点赞数和评论数，FirstP中的upData()？？？

⑥onItemChildClick() 同宗动态列表点击事件

点击昵称 进入FriendDetailsUI活动

点击点赞 ？

点击评论 进入到DynamicInfoActivity活动

点击图片 调用openImage方法，其进入到ImageActivity活动

点击视频 进入到VideoPlayActivity活动

⑦stopRefresh() 停止刷新并刷新视图

若为true则调用SmartRefreshLayout的finishRefresh()方法，若原适配器不为空则更新适配器

若为false则只调用finishRefresh()方法

⑧stopLoadMore() 停止加载更多并刷新视图

若为true则调用SmartRefreshLayout的finishLoadMore()方法，若原适配器不为空则更新适配器

若为false则只调用finishLoadMore()方法

⑨loadMoreFinish() 加载更多完毕

调用SmartRefreshLayout的finishLoadMoreWithNoMoreData()方法

⑩onRefresh() currentPage=1 刷新

调用FirstP的requestToList(Contants.TAG\_REFRESH)方法

⑪onLoadMore() currentPage++

调用FirstP的requestToList(Contatnts.TAG\_LOAD)方法

⑫returnToTop()

⑬dynamicLike

⑭hintHetDialog

**2.3、SecondFragment extends BaseFM**

1、布局内包含FrameLayout通过loadRootFragment()方法来加载ViewPaperFragment 碎片

**2.3.1、ViewPaperFragment**

1、布局通过ViewPaper和TabLayout通过setupWithViewPaper()方法联动进行加载消息、好友、群聊三个页面

2、tablayout通过addTab(tablayout.newTab())方法加入三个标题栏

3、IMPagerAdapter extends FragmentStatePagerAdapter

在getItem()方法中根据位置分别加载ConversationFM消息碎片、FriendFM好友碎片、GroupFM群聊碎片

在getPageTitle()方法中为tablayout设立标题

4、通过ChatRoomM对象的getRoomList()方法来获取聊天室列表，在roomListSuccess()的回调方法里分别为省、市、国控件赋值

接口：URL\_CHATROOMLIST

5、onClick()**???还未写**

**2.3.1.1、ConversationFM 消息列表**

1、布局就是一个recyclerview

2、initData()

①新建ConversationPresenter对象，通过其getConversation()方法来获取聊天信息列表，在initView()回调方法中，将获取到的TIMConversation集合内数据群组会话的元素添加到conversationList中，通过getPeer()方法将元素中的账户或群组id信息加入到groupList()集合中，通过notifyDataSetChanged()方法更新聊天信息列表。ConversationAdapters传入的集合为conversationList

在updateMessage()回调方法中，

新建GroupManagerPresenter和ConversationPresenter对象，在initView()方法中分别通过getGroupManageLastMessage()和getFriendshipLastMessage()来获取群管理最新一条消息和未读消息数，在回调方法onGetGroupManageLastMessage()里，将获取到的消息添加到conversationList中，然后通过Collections.sort()方法对其排序，最后调用refresh()方法更新ConversationAdapter。onGetFriendshipMessage()回调方法里什么也没做。

②...？？其他的貌似没有了

**2.3.1.2、FriendFM 好友列表**

1、具体内容都在fragment-friend包下

2、initData()

①新建FriendP对象，通过其requestFriendList()方法来获取好友列表，具体的实现是在该方法内通过FriendM的getFriendList()方法获取的（使用OkGo框架）

接口：URL\_CHATFRIENDLIST

②通过setUnReadNewFriend()方法从sharepreference中获取为添加的好友的个数

3、initRecyclerView()

新建FriendListAdapter适配器，通过FriendP的getFriendList()方法获取好友列表集合，给适配器填充数据。

4、onItemClick()

点击子项的时候，跳转至FriendDetailsUI界面，并携带好友姓名的bundle

**FriendDetailsUI好友详情界面**

**ApplyFriendActivity同意好友申请界面**

1、InitData()

新建GetApplyListRequest对象，通过其getApplyFriendList()方法来获取申请加好友的列表信息。

接口：URL\_APPLYFRIENDS

2、

**2.3.1.3、GroupFM 群聊列表**

1、

**2.4、ThirdFragment extends BaseFM 为我的界面**

①setPerson()

通过UserUtils的getUser()方法从sharepreference中获取UserBean对象，然后将这些信息给相应的控件填充信息。

②onClick()

点击：

lin\_person 跳转至PersonInfoActivity fromType 0

rel\_integral 若不是游客则跳转至IntegralUI我的积分界面

rel\_dynamic 跳转至MyDynamicUI我的动态界面 fromType 0

rel\_collect 跳转至MyDynamicUI我的收藏 fromType 1

rel\_enterprise 跳转至ShowListUI我的企业秀界面 fromType 1

rel\_action 跳转至MyActionUI我的同宗活动界面

rel\_set 跳转至MineSetActivity我的设置界面

onActivityResult()

会在更改完信息返回我的界面时重新更新我的信息，从本地获取。

**2.4.1、 PersonInfoActivity 个人详情信息**

1、initView()

通过UserUtils的getUser()方法从sharepreference获取个人信息UserBean对象，通过其getHeadimg()方法获取头像地址，然后通过Glide框架来加载头像，其他控件值也是在此进行赋值。 isUP设为false来判断是否修改

1. initData()

GetPersonInfoReuquest的getPersonInfo()方法通过用户名获取用户的信息

接口：URL\_GETPERSONINFO

建立LoadAccessoryRequest附件上传对象

建立UoDataPersonInfoRequest更新用户信息对象

为姓名（名）、行业、家乡、特长、兴趣属性框建立TextListener,监听其是否发生改变，在afterTextChanged()回调方法中和getPersonInfo()获取到的personInfo.databean进行对比，若发生改变，isUP设为true

1. initListener()

为男女的radiobutton设置选择监听器，在onCheckedChanged()回调方法中，选择女为gender赋值为“0”，男为“1”，均为字符串

1. success()为网络请求的回调方法

当requestCode为CODE\_ACCESSORYLOAD时，sendEmptyMessage()

当requestCode为CODE\_UPDATAERSONINFO时，更新用户信息（头像地址、工作、名、性别），并通过setResult()传值到上一级，即thirdFragment

当requestCode为CODE\_GETPERSONINFO时，更新用户信息（日期、行业、户籍、特长、兴趣、男女），并将信息保存到本地sharepreference中

1. Handler的handleMessage()方法

从accessorys附件信息中获取头像的地址，然后通过Glide来加载头像。

6、当点击返回按钮时，调用showMyDiagram方法，通过MSGDialog弹出是否要放弃编辑的对话框，在onEnsureClick()回调方法里dismiss()结束弹窗显示

7、点击日期按钮

调用showDateDialog(time)方法，参数通过DateUtil工具的getDateForString(DataUtil.getday())方法获取日期。（将日期分割为2018-09-20的格式）

showDateDialog()方法内部建立DatePickerDialog类，该类继承自Dialog

8、点击保存按钮

判断idUP是否修改、isChoose日期是否修改

若accessorys附件为空，判断日期是否为空，为空设birthday设为“”，不为空通过DataUtil.formatDateStr()方法转换日期，完成后通过upDataPersonInfoRequest对象的upDate方法来上传

若accessory附件不为空，通过accessory对象的.get(0).getPath()方法来获取图片地址进行上传。

9、点击头像

调用PhotoSelUtil工具类的sel()方法进行选择图片，sel()方法内使用的是知乎的Matisse框架

10、onActivityResult()的回调方法中，当requestCode为PHOTO\_REQUEST\_RESTURE时，为选择图片后的回调，在里面调用startActivityForResult(invokeSystemCrop())打开系统照片的裁剪功能，修改编辑头像的选择模式。invokeSystemCrop()方法

**2.4.2、CodeUI 我的二维码**

1、进入CodeUI时，由ThirdFragment给其传bundle,其包含：username、headimg、姓+名三个值。

2、initData()

通过getIntent()将从打开它的活动或fragment中传来的bundle给名字头像控件加载数据

调用createChineseQRCode()方法

3、createChinestQRCode() 生成二维码方法

若userId不为空（手机号），通过DESUtil工具类的encryptDES()方法对字符串“姓\_电话号”进行DES加密

若groupId不为空，同样的方法对“姓：groupId”进行加密

在AsyncTask中对加密的字符串通过第三方框架QRCodeEncoder的syncEncodeQRCode()方法生成二维码图片。

4、onClick()

点击保存：先通过PermissionUtil工具类的verifyReadAndWritePermissions()方法判断是否含有存储权限，然后通过saveImg()方法进行保存

点击分享：将bitmap格式图像通过compress()方法转化为png图像，然后跳转至IssueDynamicActivity活动

5、saveImg() 保存二维码图片

通过BitmapUtil工具类的saveImageToGallery()方法进行保存

**2.4.3、IntegralUI 我的积分**

1、setTitle()

从sharepreference中获取userbean对象，再根据此对各个控件进行数据填充

1. initRecyclerView() 初始化获取积分方式的数据

通过requestData()方法来请求获取积分的数据

3、requestData()

通过IntegralM对象的getTaskList()方法来获取任务列表，请求方式是使用OkGo的网络请求框架，在getTaskListSuccess的回调方法中，将信息填入tasks集合内后更新recyclerview

接口：URL\_GETTASKLIST

通过IntegralM对象的getIntegral()方法来获取积分和宗币信息，在getIntegralSuccess回调方法中，更新recyclerview。调用saveUser()方法，更新本地的userbean中的积分和宗币信息

接口：URL\_GETMYINTEGRAL

4、onItemChildClick()

完善信息： 跳转至PersonInfoActivity界面

登录： 源代码未做任何操作

发布动态： 跳转至IssueDynamicActivity界面

祭思先祖： 跳转至AncestorActivity界面

评论： 跳转至HomeUI首页

发布同宗活动： 跳转至ActionListUI界面

发布企业秀： 跳转至ShowListUI界面

邀请好友： 调用showSharPop()方法

该方法是创建继承自Dialog的ShareDialog对象，传入分享的地址、图 片、标题，通过show()方法显示

接口：URL\_TZH

收藏动态： 跳转至HomeUI首页

分享活动： 跳转至ActionListUI界面

**2.4.4、MyDynamicUI 我的动态** 内部含有一个viewpaper控件

当传来的fromtype为0时是我的动态

通过SerachViewPaperAdapter适配器填充MyDynamicFragment来加载我的动态界面

**2.4.4.1、MyDynamicFragment 我的动态FM**

1、initData()

若getIntent()方法得到的fromType为0，则展示个人动态界面，否则为我的收藏界面，二者的item及适配器不同。

2、setData()方法

MyDynamicFragment适配器：

创建**GetUserDynamicRequest**对象，调用getXinwenFromNet()方法，其内部通过getUserDynamic()方法来获取个人动态信息，在success()回调方法里，将返回的response字符串通过Gson解析为HomeDynamicBean对象，其中的DataBean填入list，并更新dynamicAdapter。

接口：URL\_GETUSERDYNAMIC

创建**DeleteDynamicRequest**对象，在dialog弹出对话框的回调onEnsureClick()方法内通过deleteDynamic()方法来删除

接口：CODE\_DELETEDYNAMIC

创建**LikeRequest**对象，在onItemChildClick()方法中，点赞后通过like()方法进行请求

接口：URL\_LIKE

创建**CancelLikeRequest**对象，在onItemChildClick()方法中，通过cancelLike()方法进行取消点赞的请求

接口：URL\_CANCELLIKE

3、success() 网络请求回调方法

当request分别为

CODE\_GETUSERDAYNAMIC 个人动态信息获取: 获取到信息后，将listbean信息填充到dynamics集合中，通过notifyDataSetChanged()改变adapter的值

CODE\_LIKE 点赞: 点赞后，更改原listbean的prsise和praisecount值，然后notifyItemChanged()即可更新点赞状况

CODE\_CANCELLIKE 取消点赞:与点赞类似

CODE\_DELETEDYNAMIC: 删除动态： 通过适配器的remove()方法删除相应位置的信息，并notifyDataSetChanged()更改视图

**2.4.5、MyDynamic 我的收藏**

当传来的fromType非零时为我的收藏

通过SerachViewPaperAdapter适配器填充CollectionFragment来加载我的收藏界面

**2.4.5.1、CollectionFragment 我的收藏FM（未看懂？）**

1、setData()

创建GetCollectDynamicRequest对象，调用getXinwenFromNet()方法，在其内通过getCollectDynamic()方法来获取我的收藏数据

接口：URL\_GETCOLLECTDYNAMIC

创建CancelCollectionRequest对象，在onEnsureClick()方法中，通过cancelCollent()方法来取消我的收藏

接口：URL\_CANCLECOLLECT

创建LikeRequest对象

创建CancelLikeRequest对象 以上两个未用到，点击评论或点赞都会掉转至详细信息 界面

1. setListeners()

？？删除的逻辑有些混乱

**2.4.6、我的企业秀**

**2.4.7、我的同宗活动**

**2.4.8、MineSetActivity 设置**

1、initData()

①从sharepreference中获取userbean，通过getUsername()获取电话给tv\_tel赋值

②initThird()方法内从userbean中获取thirdUsers数据，并对QQ和微信是否绑定进行判断

③通过IsAuthtypeRequest对象的isAuthtype()来获取是否实名认证信息，

接口：URL\_ISAUTHTYPE 此接口网站上没写？？？

在success回调方法中，将获取到的认证信息先保存到sharepreference中后再通过initAuth()方法将认证状态显示出来。

④通过DataCleanManager.getTotalCacheSize()方法获取缓存的大小并显示

⑤建立广播监听器，在onReceive()方法中通过thirdBindRequest对象的thirdBind()方法来进行绑定

接口：URL\_THIRDBIND

⑥通过initMsg()方法来设置消息通知，通过UserUtils的getMsgInfo()方法从本地获取获取。

2、onClick() 点击事件

①rel\_cache 清除缓存

弹出MsgDialog框，在onEnsureClick()的回调方法中，通过DataCleanManager.cleanApplicationData()方法来清除缓存。

②tv\_unlogin 退出登录

通过UnLoginRequest()对象的unLogin()方法来退出登录

接口：URL\_UNLOGIN

通过IMUserP对象的logout()方法来进行TIM的退出登录操作？应该是和聊天相关的退出

③rel\_tel 修改手机号

跳转至UpTelActivity活动

UptelActivity：先通过GetCodeRequest对象的getCodeRequest方法来获取新手机号的验证码

接口：URL\_GETCODE

然后再通过UpTelRequest对象的upTel()方法来更新手机号，在onsuccess()的回调方法中，将信息保存至本地。？？没有返回basebean吗

接口：URL\_UPTEL

④rel\_msg 消息通知

跳转至MsgInformActivity，在该活动中更改sharepreference中的MSGINFO值

⑤rel\_qq QQ解绑与绑定

是否绑定的信息在initThird()方法里已经获取了

若已绑定，调用showCancelHint()方法来弹出是否解绑通知框，解绑通过CancelThirdBindRequest对象的cancelThirdBind()方法来进行解绑

接口：URL\_CANCELTHIRDBIND

若未绑定，调用Tencent对象的login()方法进行绑定，在onComplete()回调方法里，通过ThirdBindRequest对象的thirdBind()方法来进行绑定

接口：URL\_THIRDBIND

⑥rel\_weixin 微信解绑与绑定

若已绑定，调用showCancelHint()方法来弹出是否解绑通知框，解绑通过CancelThirdBindRequest对象的cancelThirdBind()方法来进行解绑

接口：URL\_THIRDBIND

⑦实名认证

rel\_autonym 跳转至ActionInfoUI活动

布局内包含一个BridgeWebView控件加载h5页面，注册号具体的回调函数和h5进行交互

⑧用户协议

rel\_protocol 跳转至ProtocolActivity活动

通过BridgeWebView加载h5页面

⑨介绍我们

rel\_introduce 跳转至ProtocolActivity活动

通过BridgeWebView加载h5页面

⑩当前版本

tv\_version

调用StartRequest的start()方法，在success()回调方法中，通过StringUtils工具类的compareVersion()方法来比较版本号，若为1则调用showUpdateDialog()方法，在该方法里通过downloadApk()方法来下载APK,downloadAok()方法里通过queryDownloadManager()查询下载？？？

接口：URL\_START

secondFragment的页面布局内为FragmentLayout容器，通过loadRootFragment方法来加载根布局ViewPaperFragment，根布局也是一个容器，布局内通过TabLayout控件和viewpaper控件，使用IMPagerAdapter适配器的getItem方法来实现消息、好友、群聊三个fragment的切换。

消息界面ConversationFM信息的获取通过ConversationPresenter的getConversation方法来获取消息集合

好友列表界面FriendFM信息的获取通过FriendP的requestFriendtList方法来获取好友集合

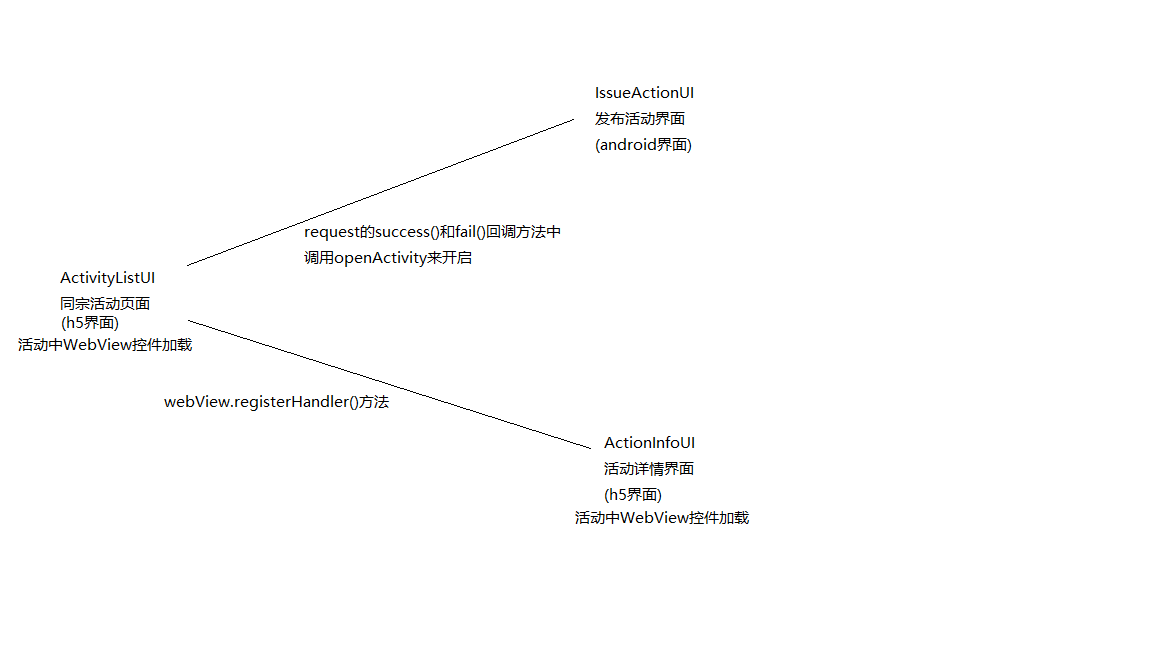
群组列表界面GroupFM信息的获取通过GroupP的requestGroupList方法来获取群组集合

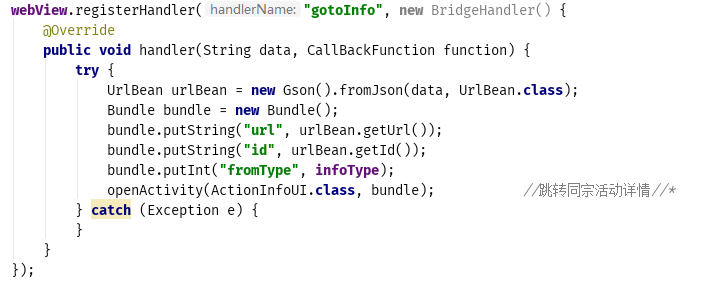
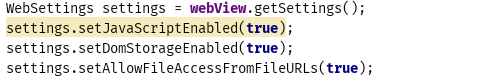
以上三个获取信息的方式都是通过MVP架构，通过Model层来具体实现网络请求方法，Presenter层来实现调用Model层的方法，在View层调用Presenter层的方法来加载数据。Model层均是通过OKGo网络请求框架，传入RequestAddress类内不同的url地址来获取相应的信息。

七、QQ和微信的初始化

1、在LoginActivity活动中需要初始化操作，因为可能会第三方登录

2、在分享时，在ShareDialog中调用ShareUtil工具类中的regToWX和reToQQ方法来进行微信和QQ终端的注册

八、同宗活动逻辑



将webview设置后，通过registerHandler方法写一个叫gotoInfo的方法，这样JavaScript就能通过调用gotoInfo方法，返回一个data，通过解析h5返回的信息，进行下一步操作

十一、发布动态

1、拍照或摄像功能

IssueDynamicActivity继承自ProjectBaseActivity，当点击拍摄rel\_camera按钮,会调用父类ProjectBaseActivity中的takePhoto()方法，在takePhoto()方法中，当通过PermissionUtil工具检查开启读写和拍照录音权限后，跳转至VideoRecordActivity进行拍照或摄像。

VideoRecordActivity的jcameraview控件的监听事件中，当captureSuccess(Bitmap)时，拍照成功，将bitmap通过FileUtil工具类的writeFile()写入手机，并setResult()将地址传出，在IssueDynamicActivity的onActivityResult中接受照片的地址；当recordSuccess时，接受视频的地址 。

在IssueDynamicActivity中，通过videoDispose(videoUrl)方法上传视频。该方法中，通过BitmapUtil的getVideoImg()方法来获取视频第一帧图片，获取失败则无法上传，成功则通过BitmaoUtil的saveBitmap方法将其存储，再通过File的getAbsolutePath()方法来获取其绝对路径，path里存储图片地址和视频地址，然后调用loadAccessoryRequest对象的load(paths)方法，进行上传…

布局文件：activity\_video\_secord.xml包含一个JCameraView即可

2、选择图片的操作

ImageActivity查看图片的操作

十二、

常用的BaseBean:

**一、登录时用户相关的**

1. **LoginBean** 验证短信验证码时的返回类型包含字段

**private boolean isError**;   
**private** UserBean **data**; 用户信息  
**private boolean isOk**; true  
**private** String **message**; “登录成功”  
**private** String **status**; “200”

1. **UserBean**为LoginBean中的data数据类型

包含字段：

**private** String **address**; “山西省，太原市，小店区”  
**private** String **headimg**; 头像地址

**private** String **gender**; 性别 “0”女“1”男  
**private** AreadressBean **areadress**; ？啥地址

**private** String **token**; 登陆成功后会返回用户token值, app需要保存到本地, 以后请求用户相关信息都用token值获取

**private** String **realname**; 名 磊

**private int teamid**; 群组名？

**private** String **nickname**; 李磊全名  
**private** ClubBean **club**; 姓氏数据

**private** String **id**; ？

**private** String **job**; 工作  
**private** String **username**; 电话

**private** List<ThirdUsersBean> **thirdUsers**; 第三方登录  
------------------------------下面的接口文档未写？  
**private** String **usersig**;  
**private** String **openid**;  
**private int authtype**;  
**private int integral**;  
**private int currency**;  
**private int msg**;  
**private boolean isTourist**;

1. **AreadressBean** 为UserBean中的areadress数据类型

包含字段：

**private int areaid**; 140105地区id  
**private** String **created**; 创建时间  
**private** String **id**; ？  
**private int cityid**; 城市id  
**private** String **userid**; ？貌似和外层LoginBean中的id是一样的  
**private int provid**; 省id  
**private** String **updated**; 更新时间

1. **ClubBean** 为UserBean中的club数据类型

包含字段：

**private** String **club**; 姓 李  
**private** String **id**; ？  
**private** String **club\_qp**; ？  
**private** String **club\_sx**; ？  
**private** String **updated**; ？  
**private** String **created**; ？

1. **ThirdUsersBean**为LoginBean中的thirdUsers数据类型

包含字段：

**private** String **openid**;   
**private int usertype**; 2代表QQ，3代表微信

1. **HomeUI相关的**

1、**StartBean** 启动页获取数据的类型

包含字段：

**private boolean isError**; false  
**private** CheckVersionBean **data**; 版本信息相关  
**private boolean isOk**; true  
**private** String **message**; “启动成功”  
**private** String **status**; “200”

2、**CheckVersionBean** 为StartBean中data的数据类型

包含字段：

**private** String **appShowInfo**; 不同启动信息对应展示信息不同

**private int startFlag**; 0更新1不更新-1强制更新  
**private** String **startImgUrl**; 启动页图片

**private** Object **startImgGoUrl**; 启动页跳转路径

**private** String **newsid**; 消息ID

**private** String **downloadPath**; apk下载地址

**private int appType**; 1为IOS，0为Android

**private** String **created**;

**private** String **serverIp**;  
**private** String **modified**;

**private int id**;

**private** String **version**; ？？  
**private** String **resversion**; 资源版本号  
**private int resupdateFlag**; 资源是否需要更新0否1是  
**private int msg**; ？？积分  
  
  
  
  
  
  
  
三、FirstFragment相关

1、**BannerBean** 轮播图获取数据的类型

包含字段：

**private boolean isError**;   
**private boolean isOk**;  
**private** String **message**; “获取Banner图成功”  
**private** String **status**; “200”  
**private** List<DataBean> **data**;

**DataBean** 轮播图地址相关BannerBean中的data数据类型

包含字段：

**private** String **adtitle**;   
**private** String **created**;  
**private** String **endtime**;  
**private** Object **gotoid**;  
**private** String **starttime**;  
**private int type**; ？应该是用来判断是跳转内部页面还是网

站1是网站，其他是内部页面  
**private** String **adtext**; 广告内容  
**private** String **url**; 跳转链接  
**private** String **imgurl**; 广告图片地址  
**private** String **adname**; 广告标题  
**private** String **club\_id**;   
**private** String **androidurl**; 跳转内部页面1-寄思先祖 2-积分页面  
**private** String **club**;  
**private** String **id**;  
**private** String **adconfigid**;  
**private** String **iosurl**;  
**private** String **updated**;  
**private** String **status**;

2、**HomeDynamicBean**

**四、PersonInfoActivity**

1、**PersonInfoBean**

**private boolean isError**;  
**private** DataBean **data**;  
**private boolean isOk**;  
**private** String **message**;  
**private** String **status**;

3、**DataBean**

**private** String **birthday**; 日期  
**private** String **speciality**; 特长  
**private** String **headimg**; 头像url  
**private** String **gender**; 性别0女1男  
**private** String **interest**; 兴趣

**private boolean isfriend**; ？  
**private** String **nickname**;   
**private** String **job**; 工作  
**private** String **register**; 籍贯

---------------以下接口文档未写？？（通过用户id和手机号获取个人信息有，通过用户名获取用户信息没有）

**private** String **userid**;  
**private** String **realname**; 名  
**private int authtype**;  
**private int teamid**;   
4、**LoadAccessoryBean**

**private boolean isError**;  
**private boolean isOk**;  
**private** String **message**;  
**private** String **status**;  
**private** List<DataBean> **data**;

**DataBean**

**private** String **path**; 附件路径  
**private** String **cdnpath**; 附件cdn路径  
**private** String **id**; 附件id  
**private boolean notNet**; ？？接口文档中没有

**五、IntegralUI.Activity**

**1、TaskBean.DataBean**

**private boolean isError**;  
**private boolean isOk**;  
**private** String **message**;  
**private** String **status**;  
**private** List<DataBean> **data**;

**DataBean**

**private int iscomplete**; 任务是否完成0未完成1已完成  
**private** String **createtime**;  
**private** String **task**; 任务描述  
**private** String **integral**; 积分  
**private** String **start**; ？？0代表说  
**private** String **id**;  
**private int task\_type**; 0-一次任务，1没日重复任务  
**private int type**; 任务类型  
**private** String **updatetime**;

**六、ApplyFriendsActivity 对应好友申请列表接口**

**FriendBean** 好友列表实体

**private boolean isError**;

**private** DataBean **data**;

**private boolean isOk**;  
**private** String **message**;  
**private** String **status**;

**DataBean**

**private int totalRow**;  
**private int pageNumber**;  
**private boolean firstPage**;  
**private boolean lastPage**;  
**private int totalPage**;  
**private int pageSize**;  
**private** List<FriendListBean> **list**;

**FriendListBean** 好友信息实体

**private** String **headimg**; 头像地址

**private** String **created**;   
**private** String **remark**; 申请备注  
**private** String **remarkName**;  
**private** String **friendUserid**;  
**private** String **id**;  
**private** String **meUserid**;  
**private** String **updated**;  
**private int status**; 0待处理1接受2拒绝  
**private** String **username**; 手机号  
**private int teamid**;  
**private boolean istop** = **false**;  
*//这个是申请的新好友列表加入的相关字段***private** String **gender**; 性别  
**private** String **tuserid**;  
**private** String **nickname**;   
**private** String **fuserid**; ？？  
*//这个是群详情里面的好友消息***private** String **realname**;  
*//建群那儿要个东西***private boolean isChoose** = **false**; *//是不是选中了***private boolean isInvisible** = **false**; *//是不是不显示*

**接口文件：RequestAddress，内部包含所有接口**