# TV（原始代码）--交接文档

## 一、开发模式：

采用MVC的开发模式；

git地址：https://git.gldev.cn:31010/xxmskt/xxmskt-tv-android.git

## 二、开发环境:

Android studio版本: 3.0.1;

gradle版本:3.0.1;

jdk版本:1.8；

## 三、上线使用：

打包使用的签名文件：在项目中的此项目是原始代码，可以用于开发江西广电或者其他地方项目，签名文件在局方那边，比如江西广电，就是到了那边之后，打包好APK给他们签名即可;

## 四、项目使用到的第三方库：

1、Recyclerview：使用tvrecyclerview;

2、网络请求：使用okhttputils；

3、查找控件：使用butterknife；

4、图片加载框架：使用Glide；

5、Json解析：使用Gson；

6、Base64编码：使用commons-codec；

## 第五、项目的目录结构：

1、model：存放网络请求，实体bean等，其中的bean：存放网络请求等的bean；

2、ui.activity：存放activity;

3、ui.adapter：存放Recyclerview、ListView、ViewPager、GridView和Tablayout等的adapter;

4、ui.Fragment:存放Fragment；

6、ui.view：存放自定义view；

7、utils：存放各种工具类；

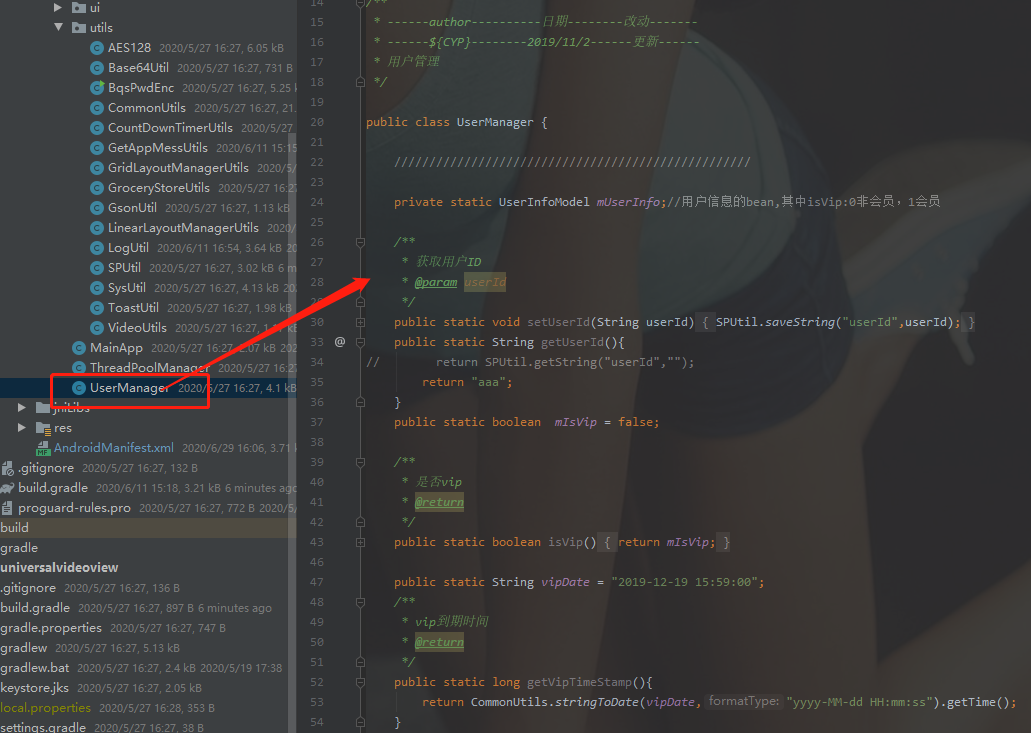
8、ui.event：通讯信息

9、model：实体类和Adpater

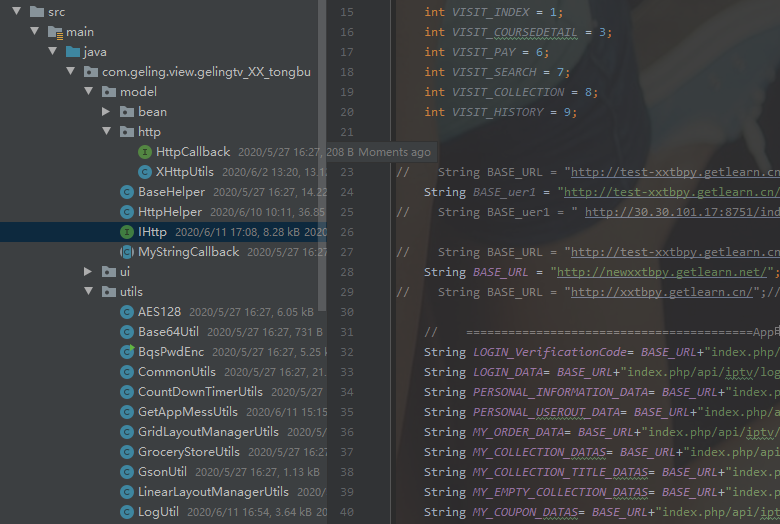
### 网络请求和调用

加解密使用的是AES128算法utils.AES128类

数据保存使用SharedPreferences方式 所需要存储的信息时使用utils.PSUtils类 ，如用户信息类是在UserManager类 所有有关用户信息的请保存在此类如下列子



网络请求：是使用okhttp3网络框架，解析是使用GSON解析网络请求在一下目录

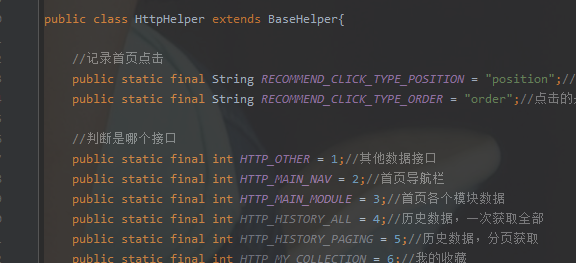


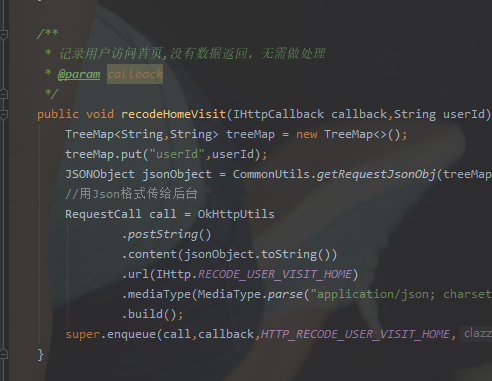
IHttp是域名和请求接口

BaseHelper是okhttp网络请求的基类，请求成功后可在onResponse中定义状态码判断

@Override  
 public void onResponse(String response, int id) {  
 try {  
 LogUtil.*e*("\_\_basehelper\_enqueue\_onResponse",response.toString());  
 JSONObject jsonObject1111 = new JSONObject(response.toString());  
// LogUtil.e("\_\_BaseHelper\_response\_" + reqType,response.toString() + "\_\_");  
 //转化成JSONObject，对象，取出data  
 String dataSrc = jsonObject1111.getString("content");  
 LogUtil.*e*("\_\_basehelper\_response\_content" + reqType,dataSrc);  
 //解密获取data  
 String data = AES128.*Decrypt*(dataSrc, CommonUtils.*PY\_KEY*);  
 LogUtil.*e*("\_\_basehelper\_response\_Decrypt" + reqType,data);  
 JSONObject jsonObject = new JSONObject(data);  
// LogUtil.e("\_\_BaseHelper\_response\_data\_" + reqType,data + "\_\_");  
 int status = jsonObject.getInt("status");  
 int error = jsonObject.getInt("error");  
 //请求成功  
 if (status == 200) {  
 if (error == 10000){//请求成功  
 Gson gson = new Gson();  
 T bean = null;  
 if (clazz != null){  
 bean = gson.fromJson(data, clazz);  
 Message msg = new Message();  
 msg.what = what;//之前传进来的what放进去  
 msg.arg1 = arg1;//之前传进来的arg1放进去  
 msg.obj = bean;//把解密之后的数据放进去  
// msg.arg1 = code;//把获取到的code放进去  
 if (callback != null){  
 callback.onHttpSuccess(reqType,msg);  
 }

HttpHelper是网络请求的协助类，定义接口的变量名，此类开始请求网络发起





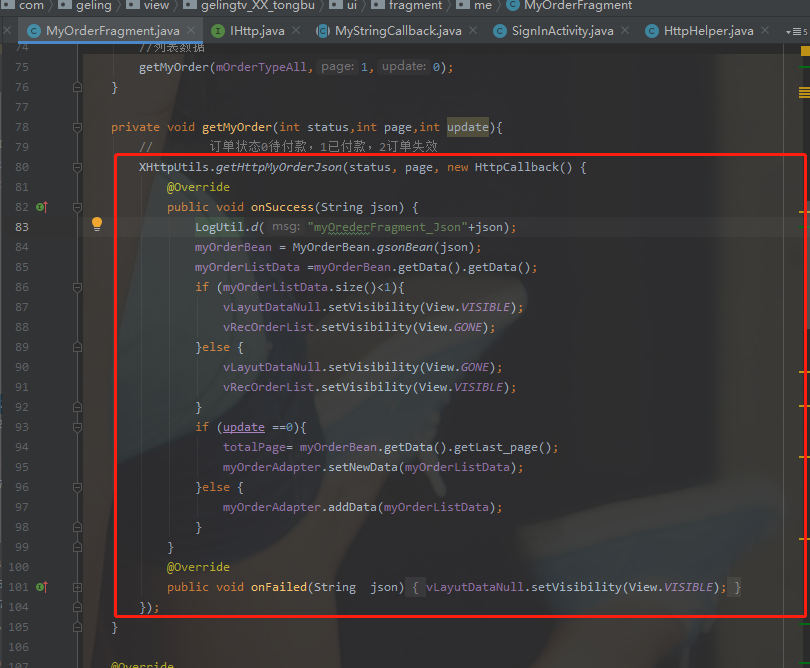
HttpCallback定义网络请求的的返回接口 onSuccess:成功，onFailed:失败

public interface HttpCallback {  
 void onSuccess(String json);  
  
 void onFailed(String json);  
}

XHttpUtils测类定义以请求接口的传参 例如：

//我的订单  
public static void getHttpMyOrderJson(int status ,int page,HttpCallback httpCallback){  
 //订单状态0待付款，1已付款，2订单失效  
 TreeMap<String, String> treeMap = new TreeMap<>();  
 treeMap.put("timestamp", String.*valueOf*(CommonUtils.*getCurrentTimeStampInt*()));  
 treeMap.put("limit","6");  
 treeMap.put("page", String.*valueOf*(page));  
 if (status!=99){  
 treeMap.put("status", String.*valueOf*(status));  
 }  
 *mapSign*(0, UserManager.*getToken*(),IHttp.*MY\_ORDER\_DATA*,treeMap,httpCallback);  
}

然后请求成功后用json解析 在所需要的活动中调用此方法 调用方式如下



### 基类

活动：BaseActivity

大杂烩：GroceryStoreUtils(此类获取手机产商 时间 Glide使用图片等方法)

Event:这个主要是对不同页面发送消息 实现内部通信

TAG:类型

Masg：内容

### 日志打印

//开关  
private static boolean *debug* = true;