目录

[Utils -工具 3](#_Toc461113862)

[L:打印日志 3](#_Toc461113863)

[SharePreUtil:使用SharePreprence存储信息 3](#_Toc461113864)

[PreferenceHelper ——SharePreprence存储信息 3](#_Toc461113865)

[Permission:6.0权限申请工具包 3](#_Toc461113866)

[CacheUtils -应用缓存处理 4](#_Toc461113867)

[DensityUtil -度量单位转换 4](#_Toc461113868)

[DialogUtils -等待加载Dialog 4](#_Toc461113869)

[MD5 -MD5加密工具 4](#_Toc461113870)

[PhoneInfoUtils -获取手机信息 4](#_Toc461113871)

[StringUtils -字符串处理类 4](#_Toc461113872)

[TimeUtils -时间处理类 4](#_Toc461113873)

[ToastUtils -吐司类 4](#_Toc461113874)

[NetWorkHelper ——网络监测类， 4](#_Toc461113875)

[FileTool ——文件工具 5](#_Toc461113876)

[Widget -控件 5](#_Toc461113877)

[Pulltorefresh -上拉加载，下拉刷新 5](#_Toc461113878)

[AppUtils ——获取应用内部信息 5](#_Toc461113879)

[EncryptionUtils ——加密方法 5](#_Toc461113880)

[HomeGridView -包裹gridView高度 5](#_Toc461113881)

[WrapContentHeightViewPager -包裹ViewPager高度 5](#_Toc461113882)

[WrapList -包裹ListView高度 5](#_Toc461113883)

[ProgressWebView-带进度条的WebView控件 5](#_Toc461113884)

[App:项目全局配置文件 5](#_Toc461113885)

[AppConfig -对整个项目控制 5](#_Toc461113886)

[AppContext -集成自Application的项目全局类 6](#_Toc461113887)

[CircleTextView ——环形围绕文字 6](#_Toc461113888)

[Base :各种通用父类 6](#_Toc461113889)

[BaseActivity -继承自AppCompatActivity的Activity父类 6](#_Toc461113890)

[BaseAdapter –试用于普通的GrideView和ListView 6](#_Toc461113891)

[BaseFragment -普通的Fragment父类 6](#_Toc461113892)

[BaseTabViewPagerAdapter -Fragment+ViewPager+TabLayout的适配器 6](#_Toc461113893)

[LasyFragment ——包含懒加载的Fragment 7](#_Toc461113894)

[Service -服务类 7](#_Toc461113895)

[PlayerMusic -音乐播放 7](#_Toc461113896)

[View ——保存View视图类 8](#_Toc461113897)

[AlertDialog ——选项弹窗 8](#_Toc461113898)

[Wheelview -滚轮管理库 9](#_Toc461113899)

[TextPopupWindow ——文字滚轮 9](#_Toc461113900)

[Refresh ——上拉加载，下拉刷新库 10](#_Toc461113901)

[使用方法： 10](#_Toc461113902)

[控件布局： 10](#_Toc461113903)

[Threutils -第三方jar包或者类库的工具类 11](#_Toc461113904)

[NetWorkRequest ——xutils和gson的工具类 11](#_Toc461113905)

[ImageLoad ——glide的工具类，从网络下载图片并显示 11](#_Toc461113906)

[GlideCircleTransform ——配合glide工具类，可显示圆形图片 11](#_Toc461113907)

[Callback -所有的接口 11](#_Toc461113908)

# Utils -工具

## LogUtils:打印日志

拥有一个统一的打印控制参数，可统一关闭日志

-拥有打印警告，信息，调试，错误日志的功能

-打印堆栈信息

-打印日志并存储为文件

-打印时间日志

## SharePreUtil:使用SharePreprence存储信息

## PreferenceHelper ——SharePreprence存储信息

## Permission:6.0权限申请工具包

使用方法参加：http://git.oschina.net/xmqian/bucsmge40p8tnwf7d9iq3.code.git

@OnClick({R.id.btn\_contact, R.id.btn\_camera})

public void open(View view){

switch (view.getId()){

case R.id.btn\_contact:

//多个权限申请

PermissionGen.with(MainActivity.this)

.addRequestCode(100)

.permissions(

Manifest.permission.READ\_CONTACTS,

Manifest.permission.RECEIVE\_SMS,

Manifest.permission.WRITE\_CONTACTS)

.request();

break;

case R.id.btn\_camera:

PermissionGen.needPermission(this, 200, Manifest.permission.CAMERA);

break;

}

}

@PermissionSuccess(requestCode = 100)

public void test(){//申请成功执行的方法，方法自定义，必须要注解

startActivity(new Intent(this, ContactActivity.class));

}

@PermissionFail(requestCode = 100)

private void test2() {//申请失败执行的方法，方法自定义，必须要注解

Dlog.debug("contact fail");

}

@PermissionSuccess(requestCode = 200)

public void openCamera(){

Dlog.debug("open camera success");

}

@PermissionFail(requestCode = 200)

public void failOpenCamera(){

Toast.makeText(this, "Camera permission is not granted", Toast.LENGTH\_SHORT).show();

}

//权限申请返回方法，

@Override public void onRequestPermissionsResult(int requestCode, String[] permissions,

int[] grantResults) {

PermissionGen.onRequestPermissionsResult(this, requestCode, permissions, grantResults);

}

}

## CacheUtils -应用缓存处理

DataCacheManager – 更详细的缓存处理类

使用方法：

读取缓存大小：

**DataCacheManager**.*getInstance*().readCacheSize(**PersonalCenterDialogActivity**.**this**, **new *DataCacheManager.CacheSizeCallBack***() {  
 **@Override  
 public void** onSize(**String** sizeString) {  
 **tvDataSpace**.setText(sizeString);  
 }  
  
 **@Override  
 public void** readFail() {  
  
 }  
});

清理缓存：（非文件）

**DataCacheManager**.*getInstance*().cleanCache(**PersonalCenterDialogActivity**.**this**,  
 **new *DataCacheManager.CacheCleanCallBack***() {  
 **@Override  
 public void** cleanFinished() {  
 **dialogUtil**.dismiss();  
 **ToastUtils**.*showMsg*(**PersonalCenterDialogActivity**.**this**, **"清除成功"**);  
 }  
  
 **@Override  
 public void** cleanFail() {  
  
 }  
 });

## DensityUtil -度量单位转换

系统默认为dp

## DialogUtils -等待加载Dialog

## MD5 -MD5加密工具

## PhoneInfoUtils -获取手机信息

## StringUtils -字符串处理类

## TimeUtils -时间处理类

## ToastUtils -吐司类

不会重复显示的吐司，

使用需要在Applicaton中初始化。

**private void** initToast() {  
 **ToastUtils**.*init*(**this**);  
}

## NetWorkHelper ——网络监测类，

判断是否有网络，是否是wifi

## FileTool ——文件工具

保存图片的工具类

**try** {  
 *//保存图片到本地并--给系统发送广播刷新相册数据  
 sendBroadCastToSystem*(**FileTool**.*saveFile*(resource,***PHOTO\_POTH*** ), context);  
 *// img.setImageBitmap(resource);*} **catch** (**IOException** e) {  
 e.printStackTrace();  
}

## UpdateManager

用于升级应用，内部进行下载升级，需要自行修改。

# Widget -控件

## Pulltorefresh -上拉加载，下拉刷新

RefreshViewUtils为控件的封装工具类

## AppUtils ——获取应用内部信息

* 获取当前版本号

## EncryptionUtils ——加密方法

* 内置MD5加密

## HomeGridView -包裹gridView高度

## WrapContentHeightViewPager -包裹ViewPager高度

## WrapList -包裹ListView高度

## ProgressWebView-带进度条的WebView控件

相关文件： res\drawable\progress\_webview\_bar\_states.xml

效果：

## ContainsEmojiEditText –限制表情输入

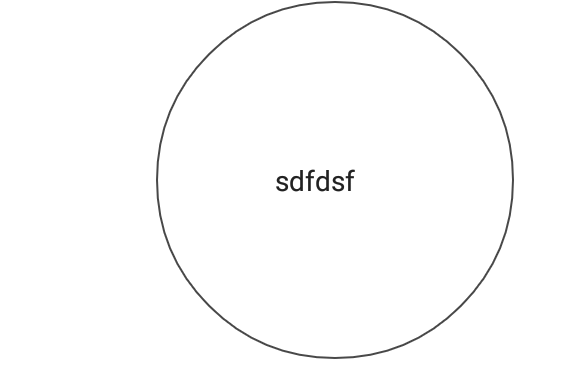
可以限制emoji表情的输入，当输入到表情时，提示不支持并还原输入表情前的文字。

# App:项目全局配置文件

## AppConfig -对整个项目控制

## AppContext -集成自Application的项目全局类

## CircleTextView ——环形围绕文字



# Base :各种通用父类

## BaseActivity -继承自AppCompatActivity的Activity父类

Manifest配置时需要加上主题

## BaseAdapter –试用于普通的GrideView和ListView

## BaseFragment -普通的Fragment父类

## BaseTabViewPagerAdapter -Fragment+ViewPager+TabLayout的适配器

相关资源：/res/layout/layout\_tabviewpager.xml

资源为布局文件，

使用方法：

**private void** initTabViewPager() {  
 **String**[] tabTitles = {**"往期回顾"**, **"报名参加"**};//标题名称  
 List<**Fragment**> list = **new** ArrayList<>();  
 list.add(**new OldtimeyFragment**());  
 list.add(**new SignUpForFragment**());//显示Fragment  
 **AlumniEventsVpAdapter** adapter = **new** AlumniEventsVpAdapter(getSupportFragmentManager(), list, tabTitles);//适配器，1,传入管理器，2，传入显示，3，传入标题  
 **viewPager**.setAdapter(adapter);  
 **tabLayout**.setupWithViewPager(**viewPager**);  
 **tabLayout**.setTabMode(**TabLayout**.***MODE\_FIXED***);  
 **tabLayout**.setTabsFromPagerAdapter(adapter);  
}

## LasyFragment ——包含懒加载的Fragment

**@Override  
protected void** lazyLoad() {  
 **if** (!**isVisible** || !**isPrepare**) {//是否显示，是否准备就绪  
 **Logger**.*e*(**"懒加载执行"**, **this**);  
 **Logger**.*e*(**"isVisible:"** + **isVisible** + **" isPrepare:"** + **isPrepare**);  
 **isPrepare** = **true**;  
 }  
 **Logger**.*e*(**"懒加载未执行"**, **this**);  
 **Logger**.*e*(**"isVisible:"** + **isVisible** + **" isPrepare:"** + **isPrepare**);  
}

# Service -服务类

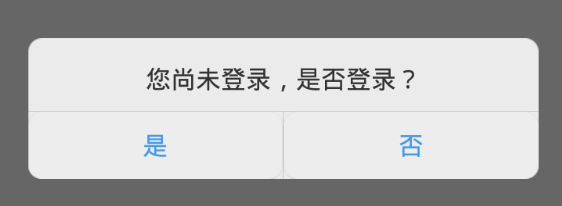
## PlayerMusic -音乐播放

启动服务的方法：

**Intent** startIntent = **new** Intent(getActivity(), **PlayerMusic**.**class**);  
startIntent.putExtra(**"url"**, musicPath);  
startIntent.putExtra(**"MSG"**, 0);  
Context.startService(startIntent);//启动服务  
Context.stopService(startIntent);//停止服务

# View ——保存View视图类

## AlertDialog ——选项弹窗



**AlertDialog** alertDialog = **new** AlertDialog(**this**);  
alertDialog.builder()  
 .setMsg(**"您尚未登录，是否登录？"**)  
 .setNegativeButton(**"是"**,**null**)  
 .setPositiveButton(**"否"**,**null**)  
 .show();

不写监听默认只有一个选项。

相关资源：drawable:

alertdialog\_left\_selector.xml

alertdialog\_right\_selector.xml

layout：

alertdialog\_view.xml

drawable-hdpi:

alert\_bg.9.png

alert\_btn\_left\_pressed.9.png

alert\_btn\_right\_pressed.9.png

style:

|  |
| --- |
| <!--dialog--> <style name="AlertDialogStyle" parent="@android:style/Theme.Dialog">  <item name="android:windowBackground">@android:color/transparent</item>  <item name="android:windowContentOverlay">@null</item>  <item name="android:windowIsFloating">true</item>  <item name="android:windowFrame">@null</item>  <item name="android:backgroundDimEnabled">true</item>  <item name="android:windowNoTitle">true</item>  <item name="android:windowIsTranslucent">true</item> </style> |

# Wheelview -滚轮管理库

## TextPopupWindow ——文字滚轮



# **Refresh ——上拉加载，下拉刷新库**

<http://note.youdao.com/yws/public/redirect/share?id=30648aad9160e63677d2d465b5a0ebe1&type=false>

## 使用方法：

*/\*\*  
 \* 刷新初始化  
 \*/***private void** initRefresh() {  
 **refresh**.autoRefresh();*//自动刷新* **refresh**.setLoadMore(**true**);  
 **refresh**.setMaterialRefreshListener(**new *MaterialRefreshListener***() {  
 **@Override  
 public void** onRefresh(**MaterialRefreshLayout** materialRefreshLayout) {  
 **RefreshUtils**.*counDown*(**refresh**);  
 addListData(**true**);  
 }  
  
 **@Override  
 public void** onRefreshLoadMore(**MaterialRefreshLayout** materialRefreshLayout) {  
 **RefreshUtils**.*counDown*(**refresh**);  
 addListData(**false**);  
 }  
 });  
}

## 控件布局：

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:**android**="http://schemas.android.com/apk/res/android"  
 xmlns:**app**="http://schemas.android.com/apk/res-auto"  
 **android**:layout\_width="match\_parent"  
 **android**:layout\_height="match\_parent">  
  
 <com.cjj.MaterialRefreshLayout  
 **android**:id="@+id/refresh"  
 **android**:layout\_width="match\_parent"  
 **android**:layout\_height="match\_parent"  
 **app**:overlay="true"  
 **app**:progress\_colors="@array/material\_colors"  
 **app**:wave\_show="false">  
  
 <ListView  
 **android**:id="@+id/lv\_list"  
 **android**:layout\_width="match\_parent"  
 **android**:layout\_height="match\_parent"  
 **android**:paddingLeft="10dp"  
 **android**:paddingRight="10dp"  
 **android**:scrollbars="none">  
  
 </ListView>  
  
 </com.cjj.MaterialRefreshLayout>  
</LinearLayout>

# **Threutils -第三方jar包或者类库的工具类**

## **NetWorkRequest ——xutils和gson的工具类**

配合xutils和gson实现。

配制方法：

compile **'org.xutils:xutils:3.3.34'**compile **'com.google.code.gson:gson:2.6.2'**

并在application里面初始化

**x.Ext**.*init*(**this**);

配合类：

接口：callback/NetWorkCallBack

## ImageLoad ——glide的工具类，从网络下载图片并显示

配合glide.jar可以使用网络下载图片并显示

配置方法：

compile **'com.github.bumptech.glide:glide:3.7.0'**

配合类：GlideCircleTranform可以生成圆形图片

## GlideCircleTransform ——配合glide工具类，可显示圆形图片

# **Callback -所有的接口**