# 系统分享

## 分享图片

Toast.*makeText*(DisplayImageActivity\_2.**this**, **"分享图片"**, Toast.***LENGTH\_SHORT***).show();  
Intent sendIntent = **new** Intent();  
sendIntent.setAction(Intent.***ACTION\_SEND***);  
sendIntent.putExtra(Intent.***EXTRA\_STREAM***,**messageInfo**.getSavePath());  
sendIntent.setType(**"image/\*"**);  
startActivity(Intent.*createChooser*(sendIntent,**"分享到"**));

# 聊天框的图片展示

## 图片展开关闭

1. 点击图片，打开要展示图片的activity。（将当前略缩图的尺寸位置传递过去）

*/\*\*  
 \* 点击图片显示  
 \** ***@param v*** *\*/***private void** onImageClick(View v,MessageInfo msgInfo,**boolean** isMine) {  
 **int** location[] = **new int**[2];  
 v.getLocationOnScreen(location);  
 FullImageInfo fullImageInfo = **new** FullImageInfo();  
 fullImageInfo.setLocationX(location[0]);  
 fullImageInfo.setLocationY(location[1]);  
 fullImageInfo.setWidth(v.getWidth());  
 fullImageInfo.setHeight(v.getHeight());  
 msgInfo.setFullImageInfo(fullImageInfo);  
  
 Intent intent = **new** Intent(**context**, DisplayImageActivity\_2.**class**);  
 Bundle bundle = **new** Bundle();  
 bundle.putSerializable(SysConstant.***CUR\_MESSAGE***, msgInfo);  
 bundle.putBoolean(**"ISMINE"**, isMine);  
 intent.putExtras(bundle);  
 **context**.startActivity(intent);  
 **context**.overridePendingTransition(0,0);  
}

1. 加载图片。

**final int** left = **mFullImageInfo**.getLocationX();  
**final int** top = **mFullImageInfo**.getLocationY();  
**final int** width = **mFullImageInfo**.getWidth();  
**final int** height = **mFullImageInfo**.getHeight();  
**mFullImageView**.getViewTreeObserver().addOnPreDrawListener(**new** ViewTreeObserver.OnPreDrawListener() {  
 @Override  
 **public boolean** onPreDraw() {  
 **mFullImageView**.getViewTreeObserver().removeOnPreDrawListener(**this**);  
 **int** location[] = **new int**[2];  
 **mFullImageView**.getLocationOnScreen(location);  
 **mLeft** = left - location[0];  
 **mTop** = top - location[1];  
 **mScaleX** = width \* 1.0f / **mFullImageView**.getWidth();  
 **mScaleY** = height \* 1.0f / **mFullImageView**.getHeight();  
 activityEnterAnim();  
 **return true**;  
 }  
});

*/\*\*  
 \* 展开动画  
 \*/***private void** activityEnterAnim() {  
 **mFullImageView**.setPivotX(0);  
 **mFullImageView**.setPivotY(0);  
 **mFullImageView**.setScaleX(**mScaleX**);  
 **mFullImageView**.setScaleY(**mScaleY**);  
 **mFullImageView**.setTranslationX(**mLeft**);  
 **mFullImageView**.setTranslationY(**mTop**);  
 **mFullImageView**.animate().scaleX(1).scaleY(1).translationX(0).translationY(0).  
 setDuration(500).setInterpolator(**new** DecelerateInterpolator()).start();  
 ObjectAnimator objectAnimator = ObjectAnimator.*ofInt*(**mBackground**, **"alpha"**, 0, 255);  
 objectAnimator.setInterpolator(**new** DecelerateInterpolator());  
 objectAnimator.setDuration(500);  
 objectAnimator.start();  
}

1. 退出。

*/\*\*  
 \* 关闭页面（图片缩小）  
 \*/***private void** exitActivity(){  
 activityExitAnim(**new** Runnable() {  
 @Override  
 **public void** run() {  
 finish();  
 overridePendingTransition(0, 0);  
 }  
 });  
}  
@TargetApi(Build.VERSION\_CODES.***JELLY\_BEAN***)  
**private void** activityExitAnim(Runnable runnable) {  
 **mFullImageView**.setPivotX(0);  
 **mFullImageView**.setPivotY(0);  
 **mFullImageView**.animate().scaleX(**mScaleX**).scaleY(**mScaleY**).translationX(**mLeft**).translationY(**mTop**).  
 withEndAction(runnable).  
 setDuration(500).setInterpolator(**new** DecelerateInterpolator()).start();  
 ObjectAnimator objectAnimator = ObjectAnimator.*ofInt*(**mBackground**, **"alpha"**, 255, 0);  
 objectAnimator.setInterpolator(**new** DecelerateInterpolator());  
 objectAnimator.setDuration(500);  
 objectAnimator.start();  
}

1. activity主题透明

**android:theme="@android:style/Theme.Translucent"**

# Json数据转换

## 组装json.

*/\*\*  
 \* 组装json字符串  
 \** ***@return*** *\*/***public** String toJsonString(){  
 JSONObject rootJson = **new** JSONObject();*//根json对象* **try** {  
 JSONArray arryJson = **new** JSONArray();*//json数组* JSONObject object = **new** JSONObject();*//json对象* object.put(**"text"**, **"额额额"**);*//json对象添加json元素* object.put(**"type"**,0);  
 arryJson.put(object);*//json数组添加json对象* rootJson.put(**"content"**,object);*//json对象添加json元素* } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 **return** rootJson.toString();  
}

{  
 **"content"**: [  
 {  
 **"text"**:**"额额额"**,  
 **"type"**:0  
 }  
 ]  
}

## 解析json.

*/\*\*  
 \* 获取json字段  
 \*  
 \** ***@param sJson*** *\*/***public void** getJsonFiled(String sJson) {  
 **try** {  
 JSONObject jsonObject = **new** JSONObject(sJson);  
 JSONArray content = jsonObject.getJSONArray(**"content"**);  
 JSONObject object = (JSONObject) content.get(0);  
 String text = object.getString(**"text"**);  
 **int** type = object.getInt(**"type"**);  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
}

# 文件缓存

## 内存缓存（LruCache）

*//设置内存缓存***int** maxMemory = (**int**) Runtime.*getRuntime*().maxMemory();  
**int** cacheSize = maxMemory / 8;  
**mMemoryCache** = **new** LruCache<String, Bitmap>(cacheSize) {  
 @Override  
 **protected int** sizeOf(String key, Bitmap value) {  
 **return** value.getByteCount();  
 }  
};

## 硬盘缓存（DiskLruCache）

1. 初始化

mDiskLruCache = DiskLruCache.*open*(diskCacheDir,getAppVersion(context),1,10\*1024\*1024);

1. 添加对象

DiskLruCache.Editor edit = **mDiskLruCache**.edit(key);  
**if**(edit!=**null**){  
 OutputStream outputStream = edit.newOutputStream(0);  
 *//下载文件* **if**(downloadUrlToStream(**imageUrl**,outputStream)){  
 edit.commit();  
 }**else**{  
 edit.abort();  
 }  
}

1. 取出对象

**try** {  
 DiskLruCache.Snapshot snapshot = **mDiskLruCache**.get(key);  
 **if**(snapshot!=**null**){  
 InputStream inputStream = snapshot.getInputStream(0);  
 Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream);  
 }  
} **catch** (IOException e) {  
 e.printStackTrace();  
}

## 图片墙示例

见picturecachedemo；