bigapple之加载网络图片

1、问题需求

有一种需求是这样的，需要从网络上加载图片。例如，加载头像，加载广告图片等。如果从头自己写，需要考虑很多问题啊，例如：

（1）缓存问题，你不可能每次都去网络加载，用户那点可怜的流量被你这么白白浪费了不砍死你。

（2）说到缓存，你用什么缓存呢。据说一个应用安卓只分配给你16M内存，有些是24M根据不同手机而定。而现在的一张图片就十几M，两个图片

（3）在加载多张图片快速切换时，一般的会出现卡顿现象，这就需要特殊处理。

2、解决方案

俗话说的好，有问题有需求，就会有解决方法。这不,bigapple1.1.2之后加入了一个强大的加载网络图片。你完全不用考虑上面种种问题，使用

他的基本原理是同时使用了两种缓存。内存缓存和磁盘缓存。每次都会先从内存缓存中取，没有再从磁盘中，最后才去网络地址上加载。当然你

不过建议是两个同时使用。内存缓存快但容量小，一般只会分配个8M都嫌多了。磁盘缓存相对多点，但是访问比内存慢，但是比起网络加载，那

（1）默认最简单的使用例子

```
public static void loadBitmapDefault(Context context, ImageView imageView) {
    AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context);
    anBitmapUtils.configDefaultShowOriginal(true);
    anBitmapUtils.display(imageView, "http://img7.9158.com/200709/01/11/53/200709018758949.jpg");
}
```

上面的context你懂的，就是activity的上下文。imageView就是加载图片的控件对象。anBitmapUtils.configDefaultShowOriginal(true);表示

（2）自定义一些参数使用例子。上面display还有一个重载方法。可以传入一个BitmapDisplayConfig对象，进行自定义显示配置。上面没有传

```
public static void loadBitmapCustom(Context context, ImageView imageView) {
    BitmapDisplayConfig displayConfig = new BitmapDisplayConfig(context);
    displayConfig.setShowOriginal(true);
    displayConfig.setLoadingBitmap(onLoadingBitmap);
    displayConfig.setLoadFailedBitmap(failBitmap);

    AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context);
    anBitmapUtils.display(imageView, "http://img7.9158.com/200709/01/11/53/200709018758949.jpg",
loadingBitmap, failBitmap);
}
```

displayConfig.setShowOriginal(true);表示设置原图显示。displayConfig.setLoadingBitmap(onLoadingBitmap);表示设置图片正在加载中的

```
AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context, diskCachePath, memoryCacheSize);
AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context, diskCachePath, memoryCacheSize,
diskCacheSize);
AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context, diskCachePath,
memoryCachePercent);AnBitmapUtils anBitmapUtils = new AnBitmapUtils(context, diskCachePath,
memoryCachePercent, diskCacheSize);
```

diskCachePath表示磁盘缓存的路径，默认是/Android/data/程序包名/cache/目录下。当应用被卸载时，这个目录会自动被清理。如果自己设置

memoryCacheSize表示内存缓存大小，默认分配了8M。

diskCacheSize表示分配给磁盘的缓存。默认分配了50M。对了无论是内存缓存还是磁盘缓存，都使用了LRU算法。所以当缓存到最大值时，会换

memoryCachePercent表示分配给磁盘缓存的百分比。例如你的应用有可用内存24M，你设置了0.3，那么就分配给内存缓存大小是24*0.3M

当让。还有在生成AnBitmapUtils对象之后，可以使用AnBitmapUtils对象的config开头的方法，再细致的配置自定义参数。

```
anBitmapUtils.configDefaultCacheExpiry(1000L * 60 * 60 * 24 * 3);
anBitmapUtils.configThreadPoolSize(5);
anBitmapUtils.configMemoryCacheEnabled(true);
anBitmapUtils.configDiskCacheEnabled(true);
```

configDefaultCacheExpiry表示缓存3天后自动过期，默认是30天。
configThreadPoolSize表示设置加载图片的线程池里的线程数。

configMemoryCacheEnabled表示是否开启内存缓存，默认开启。

configDiskCacheEnabled同理，是否开启磁盘缓存，默认开启。

当然这些配置还有很多，这个就需要你慢慢的去发现。一般默认设置就可以了。只有在特地需求出现时，才会去配置。

（3）有缓存么，当然要刷一下。调用AnBitmapUtils对象的clearCache方法就可以刷新所有的缓存了。当然你还可以刷新指定缓存。例如下面这

```
anBitmapUtils.clearCache();
anBitmapUtils.clearMemoryCache();anBitmapUtils.clearDiskCache();anBitmapUtils.clearCache(uri,null);
```

clearCache这个方法就刷新所有缓存，包括缓存在内存中和磁盘中的。

clearMemoryCache表示刷新所有内存缓存。

clearDiskCache表示刷新所有磁盘缓存

clearCache（uri,null）表示刷新特定缓存对象。因为我们的bitmap是用加载地址uri来做key的。如果你使用默认BitmapDisplayConfig对象，

还有一个重要监听接口，AfterClearCacheListener。有时候，我们需要在刷新缓存之后做一些操作，例如更新UI或者通知用户。使用这个接口

```
anBitmapUtils.configAfterClearCacheListener(new AfterClearCacheListener() {
                        @Override
                        public void afterClearCache(int type) {
                            notifyDataSetChanged4baseAdapters();
                        }
                    });
```

当然要注意的是，这个接口操作被调用一次后会被设置null，就像消息一样，每次，你清理的缓存后的操作是不一样的，所以每次要拦截操作时

3、写在最后

这个模块还在不断的优化进化当中。里面还有很多用法，你可以慢慢发掘。不过最基本的都写的很清楚了。如有使用过程中出现了什么bug，欢