

[SCA] 图片压缩框架接入文档

项目痛点：

随着公司业务体量逐渐拓展迭代，当前s3服务器集群使用率已经超过80%，集群存储空间不足，其中spx项目占49%;App端某些签收图片和签名图片大小过大，占用内存过多;因此需要将签收图片、签名图片尽量压缩到较小程度，同时业务可接受范围；

框架功能说明

功能点	功能描述
图片大小	支持从图片质量和图片宽高缩放的多个角度对图片进行压缩
图片存放路径和文件名	支持图片存放的路径和文件名自定义
自定义压缩条件	设置压缩阈值和宽高缩放阈值
多种压缩格式	可设置目标文件格式: String Uri File三者其一
支持重命名和压缩路径	可设置压缩后的存放路径及存放文件名
压缩进度	可监听压缩进度， onStart, onSuccess, onError

框架接入方法：

1. 依赖引入

```
implementation "com.shopee.sc:lbcompressor:1.0.2-SNAPSHOT"
```

方法列表说明

方法	描述
load	传入原图，支持String Uri File三种格式
filter	设置开启压缩条件
setCompressStandard	发生压缩的阈值，单位为KB
setCropShortSideSize	发生宽高缩放的阈值，单位为px
setFocusAlpha	设置是否保留透明通道
setTargetDirPath	设置压缩文件路径
setTargetFileName	设置压缩文件名称
setCompressListener	压缩回调接口
setRenameListener	压缩文件重命名接口

2. 异步调用

- 框架内部采用 IO 线程进行图片压缩，外部调用只需设置好结果监听即可，下面是以Listener形式调用：

```

//sync invoke with listener
private void withLs(final List<File> photos) {
    Luban.with(this)
        .load(photos)//load string uri file
        .setTargetDirPath(getPath())//
        .setTargetPathName("fileName")//
        .setFocusAlpha(false)//alpha channel
        .filter(path -> !(TextUtils.isEmpty(path) || path.toLowerCase().endsWith(".gif")))//
        .setCompressListener(new OnCompressListener() {
            @Override
            public void onStart() {
            }

            @Override
            public void onSuccess(File file) {
                Log.i(TAG, file.getAbsolutePath());
                mIvTest1.setImageBitmap(getBitmapFromFile(file));
            }

            @Override
            public void onError(Throwable e) {
            }
        }).launch();
}

```

- 以rxjava形式调用

```

//sync invoke with rxJava
private void withRx(final List<File> photos) {
    Disposable disposable = Single.fromCallable(() -> Luban.with(CompressorTestActivity.this)
        .setTargetDirPath(getPath())//
        .setTargetFileName("fileName")//
        .filter(path -> !(TextUtils.isEmpty(path) || path.toLowerCase().endsWith(".gif")))//
        .setCompressStandard(100)//
        .setCropShortSideSize(800)//
        .setRenameListener(filePath -> filePath + "renameFile")//
        .load(photos).get())//load string uri file
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .onErrorReturn(throwable -> {
            Log.d(TAG, throwable.toString());
            return null;
        }).subscribe(list -> {
            for (File file : list) {
                if (list.size() == 0) {
                    return;
                }
                Log.i(TAG, file.getAbsolutePath());

                mIvTest1.setImageBitmap(getBitmapFromFile(file));
            }
        });
}

```

3.同步调用

```

//async invoke
private void withAsync(final List<File> photos) throws IOException {
    Luban.with(CompressorTestActivity.this)
        .setTargetDirPath(getPath())//
        .setTargetFileName("fileName")//
        .filter(path -> !(TextUtils.isEmpty(path) || path.toLowerCase().endsWith(".gif")))//
        .setCompressStandard(100)//
        .setCropShortSideSize(800)//
        .load(photos).get();
}

```