# compress三方件使用说明

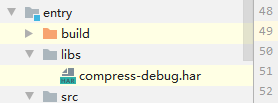
## **关于**

提供了一个轻量级的图像压缩库。将允许您将大照片压缩成小尺寸的照片，图像质量损失或可以忽略不计。

## **compress的依赖添加**

为你的应用添加compress-debug.har。

将compress-debug.har复制到entry\libs目录下即可（由于build.gradle中已经依赖的libs目录下的\*.har，因此不需要在做修改）。



## **使用**

（1）ability\_main.xml设置界面布局

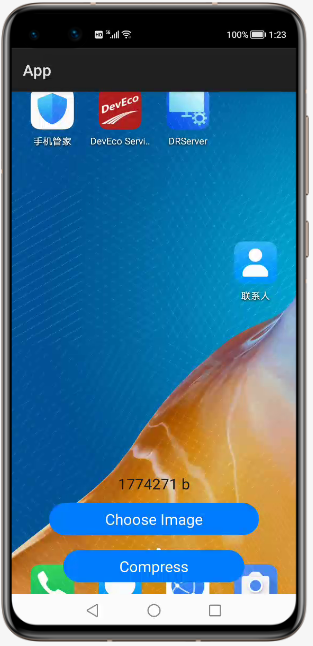
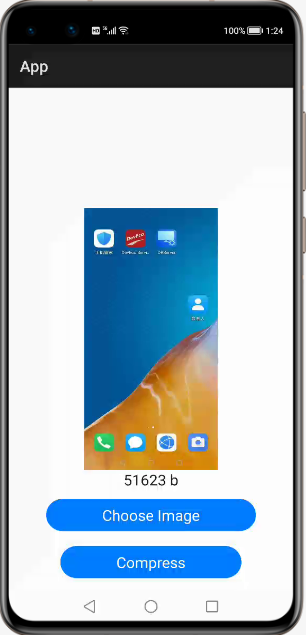
<**DependentLayout  
 xmlns:ohos="http://schemas.huawei.com/res/ohos"  
 ohos:width="match\_parent"  
 ohos:height="match\_parent"  
 ohos:background\_element="#FFFFFF"**>  
 <**Image  
 ohos:id="$+id:image1"  
 ohos:height="match\_parent"  
 ohos:width="match\_parent"  
 ohos:image\_src="$media:dog1.PNG"**/>  
 <**Text  
 ohos:id="$+id:text"  
 ohos:width="match\_content"  
 ohos:height="match\_content"  
 ohos:text=""  
 ohos:text\_size="19fp"  
 ohos:text\_color="#1C1C1C"  
 ohos:top\_padding="8vp"  
 ohos:bottom\_padding="8vp"  
 ohos:right\_padding="70vp"  
 ohos:left\_padding="70vp"  
 ohos:center\_in\_parent="true"  
 ohos:align\_parent\_bottom="true"  
 ohos:bottom\_margin="120vp"**/>  
 <**Button  
 ohos:id="$+id:choose\_button"  
 ohos:width="match\_content"  
 ohos:height="match\_content"  
 ohos:text="Choose Image"  
 ohos:text\_size="19fp"  
 ohos:text\_color="#FFFFFF"  
 ohos:top\_padding="8vp"  
 ohos:bottom\_padding="8vp"  
 ohos:right\_padding="70vp"  
 ohos:left\_padding="70vp"  
 ohos:background\_element="$graphic:background\_button"  
 ohos:center\_in\_parent="true"  
 ohos:align\_parent\_bottom="true"  
 ohos:bottom\_margin="75vp"**/>  
 <**Button  
 ohos:id="$+id:button"  
 ohos:width="match\_content"  
 ohos:height="match\_content"  
 ohos:text="Compress"  
 ohos:text\_size="19fp"  
 ohos:text\_color="#FFFFFF"  
 ohos:top\_padding="8vp"  
 ohos:bottom\_padding="8vp"  
 ohos:right\_padding="70vp"  
 ohos:left\_padding="70vp"  
 ohos:background\_element="$graphic:background\_button"  
 ohos:center\_in\_parent="true"  
 ohos:align\_parent\_bottom="true"  
 ohos:bottom\_margin="15vp"**/>  
</**DependentLayout**>

（2）MainAbilitySlice

获取界面的按钮，并分别向按钮绑定点击事件

**public void** onStart(Intent intent) {  
 **super**.onStart(intent);  
 **super**.setUIContent(ResourceTable.***Layout\_ability\_main***);  
  
 *// 请求文件的读取权限* String[] permissions = {**"ohos.permission.READ\_USER\_STORAGE"**};  
 requestPermissionsFromUser(permissions, 0);  
  
 *// 获取压缩按钮并绑定事件* Button button = (Button) findComponentById(ResourceTable.***Id\_button***);  
 **if** (button != **null**) {  
 *// 为按钮设置点击回调* button.setClickedListener(**new** Component.ClickedListener() {  
 @Override  
 **public void** onClick(Component component) {  
 **try** {  
 File file = **new** File(System.*getProperty*(**"java.io.tmpdir"**) + File.***separator*** + **tmpName**);  
 HiLog.*error*(***LOG\_LABEL***, **"old size..."** + file.length() + **" ...b"**);  
  
 *// 默认压缩  
 // File newFile = Compressor.defaultCompress(file);  
  
 // 自定义压缩* File newFile = Compressor.*customCompress*(getContext(), file, 500, 1000, 60);  
 Text text = (Text) findComponentById(ResourceTable.***Id\_text***);  
 text.setText(**"size: "** + newFile.length() + **" b"**);  
 HiLog.*error*(***LOG\_LABEL***, **"new size..."** + newFile.length() + **" ...b"**);  
 PixelMap newPixelMap = Compressor.*decode*(newFile);  
 Image image = (Image) findComponentById(ResourceTable.***Id\_image1***);  
 image.setPixelMap(newPixelMap);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
  
 *// 获取选择图片按钮并绑定事件* Button chooseButton = (Button) findComponentById(ResourceTable.***Id\_choose\_button***);  
 **if** (chooseButton != **null**) {  
 *// 为按钮设置点击回调* chooseButton.setClickedListener(**new** Component.ClickedListener() {  
 @Override  
 **public void** onClick(Component component) {  
 DataAbilityHelper helper = DataAbilityHelper.*creator*(getContext());  
 **try** {  
 ResultSet resultSet = helper.query(AVStorage.Images.Media.***EXTERNAL\_DATA\_ABILITY\_URI***, **null**, **null**);  
 **while** (resultSet != **null** && resultSet.goToNextRow()) {  
 *// 互殴媒体库的图片* **int** id = resultSet.getInt(resultSet.getColumnIndexForName(AVStorage.Images.Media.***ID***));  
 HiLog.*error*(***LOG\_LABEL***, **"id:..."** + id + **" ..."**);  
 Uri uri = Uri.*appendEncodedPathToUri*(AVStorage.Images.Media.***EXTERNAL\_DATA\_ABILITY\_URI***, **""** + id);  
 *// 根据图片的uri打开文件并保存到临时目录中* FileDescriptor fileDescriptor = helper.openFile(uri, **"r"**);  
 ImageSource.DecodingOptions decodingOpts = **new** ImageSource.DecodingOptions();  
 decodingOpts.**sampleSize** = ImageSource.DecodingOptions.***DEFAULT\_SAMPLE\_SIZE***;  
 ImageSource imageSource = ImageSource.*create*(fileDescriptor, **null**);  
 PixelMap pixelMap = imageSource.createThumbnailPixelmap(decodingOpts, **true**);  
 ImagePacker imagePacker = ImagePacker.*create*();  
 **tmpName** = UUID.*randomUUID*().toString();  
 File file = **new** File(System.*getProperty*(**"java.io.tmpdir"**) + File.***separator*** + **tmpName**);  
 FileOutputStream outputStream = **new** FileOutputStream(file);  
 ImagePacker.PackingOptions packingOptions = **new** ImagePacker.PackingOptions();  
 packingOptions.**quality** = 100;  
 **boolean** result = imagePacker.initializePacking(outputStream, packingOptions);  
 result = imagePacker.addImage(pixelMap);  
 **long** dataSize = imagePacker.finalizePacking();  
 *// 显示图片和图片大小* Text text = (Text) findComponentById(ResourceTable.***Id\_text***);  
 text.setText(**"size: "** + file.length() + **" b"**);  
 Image image = (Image) findComponentById(ResourceTable.***Id\_image1***);  
 image.setPixelMap(pixelMap);  
 }  
 } **catch** (DataAbilityRemoteException | FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
}

1. 结果展示

1. Compressor核心方法

***defaultCompress(Context context, File file)***

输入：用户需要处理的文件

输出：处理后的临时文件

处理流程：传入图片后，先将图片拷贝到临时目录，再按照默认的压缩处理方法，完成后返回处理后图片的临时目录

***customCompress(Context context, File file, int width, int height, int quality)***

输入：用户需要处理的文件，处理后的宽度、高度以及图片质量。

输出：处理后的临时文件

处理流程：传入图片后，先将图片拷贝到临时目录，再按照指定的处理方法进行压缩，完成后返回处理后图片的临时目录

***decode(File file)***

输入：图片的目录

输出：图片的像素矩阵形式

作用：用于界面图片展示

## **License**

Copyright 2015-2019 iSoftStone-dev

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.