

[SCA] 图片选择上传组件介绍

1. 背景

在部分项目中，会要求上传图片，包含拍照或选择设备中的图片，还包含预览和上传功能。功能和界面都很类似，可以做成组件，通过线上集成到各个项目，方便使用。

2. 功能

功能点	功能描述
权限控制	使用功能时，要求提供存储权限和拍照权限。
图片选择	支持调用照相机功能和图片选择功能。
图片预览	缩略图排列可自动换行，图片放大预览，左右滑动可切换图片。
图片上传	由于图片上传接口未统一，目前仅支持将图片Uri提供给业务方。
图片压缩	可设置压缩图片的品质和最小宽度分辨率阈值。

3. 使用介绍

3.1 SDK 接入

```
implementation 'com.shopee.sc:picture-picker:{lastReleaseVersion}'
```

3.2 代码使用

注意：使用该控件的Activity必须为ComponentActivity及其子类。

3.2.1 初始化

由于控件涉及权限申请等功能，用到了Activity Result API (androidx.activity) ，需要在Activity的onCreate调用ImagesLayoutManager的initOrGetManager方法，在onDestroy调用release方法。

3.2.2 入口控件ShowImageLayout

在xml布局文件中直接使用：

```
<com.shopee.android.picturepicker.widget.thumbnailview.ShowImageLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:picturePickerType="1"
    app:picturePickerMaxItem="4"
    app:picturePickerItemWidth="90dp"
    app:picturePickerItemHeight="90dp"
    app:picturePickerItemSpace="6dp"
    app:picturePickerItemSpanCount="4"
    app:picturePickerChildAddDrawable="@drawable/ic_add_grey_24dp"
    app:picturePickerChildBackground="@color/colorSplitLine"/>
```

picturePickerType表示选择本地图片的方式，0为系统自带的图片选择页面，1为SPX目前使用的自定义相册图片选择页面。

picturePickerMaxItem表示设置最多可选择图片数量。

picturePickerItemWidth和picturePickerItemHeight分别表示设置缩略图的长宽值。

picturePickerItemSpace表示设置缩略图之间的间隔。

picturePickerChildAddDrawable表示设置添加新图片的样式中的图片资源，如WMS为一个加号图标，SPX为一个照相机图标。

picturePickerChildBackground表示设置item样式的背景。

picturePickerItemSpanCount表示列数/行数

```
<declare-styleable name="ShowImagesLayout" tools:ignore="ResourceName">
    <!-- -->
    <attr name="picturePickerType" format="integer">
        <!-- -->
        <enum name="system_picker" value="0" />
        <!-- SPX -->
        <enum name="custom_picker" value="1" />
    </attr>
    <!-- -->
    <attr name="picturePickerShowMode" format="integer">
        <!-- -->
        <enum name="edit_mode" value="0" />
        <!-- -->
        <enum name="show_mode" value="1" />
    </attr>
    <!-- -->
    <attr name="picturePickerMaxItem" format="integer" />
    <!-- item -->
    <attr name="picturePickerItemWidth" format="dimension">
        <enum name="fill_parent" value="-1" />
        <enum name="match_parent" value="-1" />
        <enum name="wrap_content" value="-2" />
    </attr>
    <!-- item -->
    <attr name="picturePickerItemHeight" format="dimension">
        <enum name="fill_parent" value="-1" />
        <enum name="match_parent" value="-1" />
        <enum name="wrap_content" value="-2" />
    </attr>
    <!-- item/ -->
    <attr name="picturePickerItemSpanCount" format="integer"/>
    <!-- item -->
    <attr name="picturePickerItemOrientation" format="integer">
        <!-- -->
        <enum name="horizontal" value="0" />
        <!-- -->
        <enum name="vertical" value="1" />
    </attr>
    <!-- item -->
    <attr name="picturePickerItemSpace" format="dimension"/>
    <!-- item -->
    <attr name="picturePickerChildBackground" format="reference|color" />
    <!-- item -->
    <attr name="picturePickerChildAddDrawable" format="reference|color" />
</declare-styleable>
```

3.2.2 图片控制

单张图片信息结构体ShowImageLayout.ImageInfo，包含信息：

```
String uri; // uri+isUrl=trueURL
String compressedPath; // +
boolean isUrl; //
boolean needCompress; // false
```

本地选择的图片以及拍照图片，都不是网络图片，uri不是绝对路径，并且都默认需要压缩。

设置图片的方法setImages(List<ImageInfo>), 可重设所有选择的图片。

添加图片的方法addImage(ImageInfo)和addImages(List<ImageInfo>), 可在末尾添加一张或多张图片。

删除图片的方法removeImage(String uri)和removeImages(List<ImageInfo>), 可删除现有的一张或多张图片。

3.2.3 获取信息

获取已选择图片的uri列表getFileUriStrings(), 获取已选择图片的ImageInfo信息列表getImageInfos()。

监听ImageListener, 可以监测选择图片的变化以及点击事件。

```
public interface ImageListener {
    /**
     *
     */
    int TYPE_CHANGE = 0;
    /**
     *
     */
    int TYPE_ADD = 1;
    /**
     *
     */
    int TYPE_REMOVE = 2;

    /**
     *
     * @param wholeInfos
     * @param changeInfos
     * @param changeType
     */
    void onImageChange(List<ShowImagesLayout.ImageInfo> wholeInfos, List<ShowImagesLayout.ImageInfo> changeInfos,
        int changeType);

    /**
     *
     * @param position
     * @param command {@link OnItemClickListener#COMMAND_ADD_NEW_PICTURE}
     * @param command {@link OnItemClickListener#COMMAND_PREVIEW_PICTURE}
     * @param command {@link OnItemClickListener#COMMAND_REMOVE_PICTURE}
     */
    void onImageClick(int position, int command);
}
```

通过setImageChangeListener(ImageListener)方法设置监听。

3.2.4 图片压缩功能

压缩功能的参数: 压缩的品质 (1 ~ 100), 需要压缩的阈值 (图片短边长度px超过阈值则压缩)。

压缩功能可以设置压缩回调, 在回调中获取压缩结果, 也可以直接在子线程中调用方法获取压缩的结果。

压缩回调接口ImageCompressCallback:

```
public interface ImageCompressCallback {
    /**
     * {@link ShowImagesLayout#processCompress()}
     */
    @param files
    @param imageInfos info
    * {@link ShowImagesLayout.ImageInfo#compressedPath}
    */
    void onCompressComplete(List<File> files, List<ShowImagesLayout.ImageInfo> imageInfos);
}
```

通过setCompressCallback方法设置回调，通过processCompress()方法开启压缩，该方法也有设置参数的重载方法。终止压缩流程的方法为shutdownCompress()，以及页面销毁时要记得调用removeCompressCallback()方法移除回调对象。

直接可以在子线程中调用的获取压缩结果的方法为，getCompressedFiles()获取成功压缩的File对象列表，getHandledImageInfo()获取处理完成的图片信息列表，这两个方法各自有设置压缩参数的重载方法。

3.2.5 其他

在上传图片成功后，可使用clearCameraCache(long time)方法清除拍照图片的缓存，使用clearCacheAll()方法同时删除压缩文件缓存和照片缓存。