我们在java开发时，使用Thumbnails工具类能帮助我们对图片进行很好的处理，Thumbnails对图片的操作进行了很好的封装，往往很复杂的步骤能用一行代码就完成。

Thumbnails支持：

指定大小进行缩放

按照比例进行缩放

不按照比例，指定大小进行缩放

旋转

水印

裁剪

转化图像格式

输出到OutputStream

输出到BufferedImage

使用步骤

导入架包

<dependency>

<groupId>net.coobird</groupId>

<artifactId>thumbnailator</artifactId>

<version>0.4.8</version>

</dependency>12345

2.具体使用方法

/\*\*

\* 指定大小进行缩放

\*

\* @throws IOException

\*/

private void test1() throws IOException {

/\*

\* size(width,height) 若图片横比200小，高比300小，不变

\* 若图片横比200小，高比300大，高缩小到300，图片比例不变 若图片横比200大，高比300小，横缩小到200，图片比例不变

\* 若图片横比200大，高比300大，图片按比例缩小，横为200或高为300

\*/

Thumbnails.of("images/test.jpg").size(200, 300).toFile("C:/image\_200x300.jpg");

Thumbnails.of("images/test.jpg").size(2560, 2048).toFile("C:/image\_2560x2048.jpg");

}

/\*\*

\* 按照比例进行缩放

\*

\* @throws IOException

\*/

private void test2() throws IOException {

/\*\*

\* scale(比例)

\*/

Thumbnails.of("images/test.jpg").scale(0.25f).toFile("C:/image\_25%.jpg");

Thumbnails.of("images/test.jpg").scale(1.10f).toFile("C:/image\_110%.jpg");

}

/\*\*

\* 不按照比例，指定大小进行缩放

\*

\* @throws IOException

\*/

private void test3() throws IOException {

/\*\*

\* keepAspectRatio(false) 默认是按照比例缩放的

\*/

Thumbnails.of("images/test.jpg").size(120, 120).keepAspectRatio(false).toFile("C:/image\_120x120.jpg");

}

/\*\*

\* 旋转

\*

\* @throws IOException

\*/

private void test4() throws IOException {

/\*\*

\* rotate(角度),正数：顺时针 负数：逆时针

\*/

Thumbnails.of("images/test.jpg").size(1280, 1024).rotate(90).toFile("C:/image+90.jpg");

Thumbnails.of("images/test.jpg").size(1280, 1024).rotate(-90).toFile("C:/iamge-90.jpg");

}

/\*\*

\* 水印

\*

\* @throws IOException

\*/

private void test5() throws IOException {

/\*\*

\* watermark(位置，水印图，透明度)

\*/

Thumbnails.of("images/test.jpg").size(1280, 1024).watermark(Positions.BOTTOM\_RIGHT, ImageIO.read(new File("images/watermark.png")), 0.5f)

.outputQuality(0.8f).toFile("C:/image\_watermark\_bottom\_right.jpg");

Thumbnails.of("images/test.jpg").size(1280, 1024).watermark(Positions.CENTER, ImageIO.read(new File("images/watermark.png")), 0.5f)

.outputQuality(0.8f).toFile("C:/image\_watermark\_center.jpg");

}

/\*\*

\* 裁剪

\*

\* @throws IOException

\*/

private void test6() throws IOException {

/\*\*

\* 图片中心400\*400的区域

\*/

Thumbnails.of("images/test.jpg").sourceRegion(Positions.CENTER, 400, 400).size(200, 200).keepAspectRatio(false)

.toFile("C:/image\_region\_center.jpg");

/\*\*

\* 图片右下400\*400的区域

\*/

Thumbnails.of("images/test.jpg").sourceRegion(Positions.BOTTOM\_RIGHT, 400, 400).size(200, 200).keepAspectRatio(false)

.toFile("C:/image\_region\_bootom\_right.jpg");

/\*\*

\* 指定坐标

\*/

Thumbnails.of("images/test.jpg").sourceRegion(600, 500, 400, 400).size(200, 200).keepAspectRatio(false).toFile("C:/image\_region\_coord.jpg");

}

/\*\*

\* 转化图像格式

\*

\* @throws IOException

\*/

private void test7() throws IOException {

/\*\*

\* outputFormat(图像格式)

\*/

Thumbnails.of("images/test.jpg").size(1280, 1024).outputFormat("png").toFile("C:/image\_1280x1024.png");

Thumbnails.of("images/test.jpg").size(1280, 1024).outputFormat("gif").toFile("C:/image\_1280x1024.gif");

}

/\*\*

\* 输出到OutputStream

\*

\* @throws IOException

\*/

private void test8() throws IOException {

/\*\*

\* toOutputStream(流对象)

\*/

OutputStream os = new FileOutputStream("C:/image\_1280x1024\_OutputStream.png");

Thumbnails.of("images/test.jpg").size(1280, 1024).toOutputStream(os);

}

/\*\*

\* 输出到BufferedImage

\*

\* @throws IOException

\*/

private void test9() throws IOException {

/\*\*

\* asBufferedImage() 返回BufferedImage

\*/

BufferedImage thumbnail = Thumbnails.of("images/test.jpg").size(1280, 1024).asBufferedImage();

ImageIO.write(thumbnail, "jpg", new File("C:/image\_1280x1024\_BufferedImage.jpg"));

}

---------------------