链滴

Im4java+GraphicsMagick 图片处理工具

作者：Sysecho
原文链接：https://ld246.com/article/1498744026872
来源网站：链滴

```
package com.jeeplus.modules.isp.utils;

import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;

import org.apache.commons.lang3.SystemUtils;
import org.im4java.core.ConvertCmd;
import org.im4java.core.IM4JavaException;
import org.im4java.core.IMOperation;
import org.im4java.process.Pipe;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import com.jeeplus.modules.isp.service.impl.MongoFileServiceImp;

/**
 * 图片处理工具<br>
 * 代码实现类将图片装换压缩成固定的大小格式的图片<br>
 * 使用工具为im4java+GraphicsMagick-1.3.24-Q8<br>
 * 参考: <a href="http://im4java.sourceforge.net/">im4java</a><br>
 * GraphicsMagick: <a href="ftp://ftp.graphicsmagick.org/pub/GraphicsMagick/windows/">
载</a><br>
 *
 * @author xiaofei.xian
 * @version
 *         1.0, 2016年8月8日 下午2:53:20
 */
public class GraphicsMagicUtil {

    private static Logger logger = LoggerFactory.getLogger(MongoFileServiceImp.class);

    private static String GRAPHICS_MAGICK_PATH;

    private static boolean IS_WINDOWS;

    /**
     * 缩放图片大小
     *
     * @throws IM4JavaException
     * @throws InterruptedException
     * @throws IOException
     * @return
     */
    public static OutputStream zoomPic(OutputStream os, InputStream is, String contentType,
nteger width, Integer height)
            throws IOException, InterruptedException, IM4JavaException {
        IMOperation op = buildIMOperation(contentType, width, height);

        Pipe pipeIn = new Pipe(is, null);
        Pipe pipeOut = new Pipe(null, os);
```

```
        ConvertCmd cmd = new ConvertCmd(true);
        if (IS_WINDOWS) {
            // linux下不要设置此值，不然会报错
            cmd.setSearchPath(GRAPHICS_MAGICK_PATH);
        }
        cmd.setInputProvider(pipeIn);
        cmd.setOutputConsumer(pipeOut);
        cmd.run(op);
        return os;
    }

    /**
     * 压缩图片，返回输入流
     *
     * @param is
     * @param contentType
     * @param width
     * @param height
     * @return
     */
    public static InputStream convertThumbnailImage(InputStream is, String contentType, dou
le width, double height) {
        try {
            IMOperation op = buildIMOperation(contentType, width, height);

            Pipe pipeIn = new Pipe(is, null);
            ByteArrayOutputStream os = new ByteArrayOutputStream();
            Pipe pipeOut = new Pipe(null, os);

            ConvertCmd cmd = new ConvertCmd(true);
            if (IS_WINDOWS) {
                // linux下不要设置此值，不然会报错
                cmd.setSearchPath(GRAPHICS_MAGICK_PATH);
            }
            cmd.setInputProvider(pipeIn);
            cmd.setOutputConsumer(pipeOut);
            cmd.run(op);
            return new ByteArrayInputStream(os.toByteArray());
        } catch (Exception e) {
            if (logger.isInfoEnabled()) {
                logger.info("Failed to convert image {}", e.getMessage());
            }
            return null;
        }
    }

    /**
     * @param contentType
     * @param width
     * @param height
     * @return
     */
    private static IMOperation buildIMOperation(String contentType, Number width, Number
eight) {
```

```
        IMOperation op = new IMOperation();

        String widHeight = width + "x" + height;
        op.addImage("-"); // 命令：处输入流中读取图片
        op.addRawArgs("-scale", widHeight);// 按照给定比例缩放图片
        op.addRawArgs("-gravity", "center"); // 缩放参考位置 对图像进行定位
        op.addRawArgs("-extent", width + "x" + height); // 限制JPEG文件的最大尺寸
        op.addRawArgs("+profile", "*");// 去除Exif信息

        // 设置图片压缩格式
        op.addImage(contentType.substring(contentType.indexOf("/") + 1) + ":-");
        return op;
    }

    public static void setGraphicsMagickPath(String graphicsMagickPath) {
        GraphicsMagicUtil.GRAPHICS_MAGICK_PATH = graphicsMagickPath;
        IS_WINDOWS = SystemUtils.IS_OS_WINDOWS;
    }

}
```