转自：<http://tsunzhang.iteye.com/>

|  |
| --- |
| import java.io.BufferedReader;  import java.io.File;  import java.io.FileOutputStream;  import java.io.IOException;  import java.io.InputStreamReader;  import java.io.OutputStream;  import java.io.OutputStreamWriter;  import java.text.DateFormat;  import java.text.SimpleDateFormat;  import java.util.Date;  public class JavaShellUtil {  //基本路径  private static final String basePath = "/tmp/";  //记录Shell执行状况的日志文件的位置(绝对路径)  private static final String executeShellLogFile = basePath + "executeShell.log";  //发送文件到Kondor系统的Shell的文件名(绝对路径)  private static final String sendKondorShellName = basePath + "sendKondorFile.sh";  public int executeShell(String shellCommand) throws IOException {  int success = 0;  StringBuffer stringBuffer = new StringBuffer();  BufferedReader bufferedReader = null;  //格式化日期时间，记录日志时使用  DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:SS ");  try {  stringBuffer.append(dateFormat.format(new Date())).append("准备执行Shell命令 ").append(shellCommand).append(" \r\n");  Process pid = null;  String[] cmd = {"/bin/sh", "-c", shellCommand};  //执行Shell命令  pid = Runtime.getRuntime().exec(cmd);  if (pid != null) {  stringBuffer.append("进程号：").append(pid.toString()).append("\r\n");  //bufferedReader用于读取Shell的输出内容 bufferedReader = new BufferedReader(new InputStreamReader(pid.getInputStream()), 1024);  pid.waitFor();  } else {  stringBuffer.append("没有pid\r\n");  }  stringBuffer.append(dateFormat.format(new Date())).append("Shell命令执行完毕\r\n执行结果为：\r\n");  String line = null;  //读取Shell的输出内容，并添加到stringBuffer中  while (bufferedReader != null &  &  (line = bufferedReader.readLine()) != null) {  stringBuffer.append(line).append("\r\n");  }  } catch (Exception ioe) {  stringBuffer.append("执行Shell命令时发生异常：\r\n").append(ioe.getMessage()).append("\r\n");  } finally {  if (bufferedReader != null) {  OutputStreamWriter outputStreamWriter = null;  try {  bufferedReader.close();  //将Shell的执行情况输出到日志文件中  OutputStream outputStream = new FileOutputStream(executeShellLogFile);  outputStreamWriter = new OutputStreamWriter(outputStream, "UTF-8");  outputStreamWriter.write(stringBuffer.toString());  } catch (Exception e) {  e.printStackTrace();  } finally {  outputStreamWriter.close();  }  }  success = 1;  }  return success;  }  } |