

文件编码识别源码

- 转自：[java自动探测文件的字符编码](#)
- 其中 chardet.jar 包可在 [主页下载](#)，也可在 maven 仓库直接下载
- 识别是通过统计数据得到的，可能不准

```
import org.mozilla.intl.chardet.nsDetector;
import org.mozilla.intl.chardet.nsICharsetDetectionObserver;
import java.io.BufferedInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

public class FileCharsetDetector {
    private boolean found = false;
    private String encoding = null;

    public static void main(String[] argv) throws Exception {
        File file1 = new File("D:\\testg.txt");

        System.out.println("文件编码:" + new
FileCharsetDetector().guessFileEncoding(file1));
    }

    /**
     * 传入一个文件(File)对象，检查文件编码
     *
     * @param file
     *         File对象实例
     * @return 文件编码，若无，则返回null
     * @throws FileNotFoundException
     * @throws IOException
     */
    public String guessFileEncoding(File file) throws FileNotFoundException,
IOException {
        return guessFileEncoding(file, new nsDetector());
    }

    /**
     * <pre>
     * 获取文件的编码
     * @param file
     *         File对象实例
     */
}
```

```

    * @param languageHint
    *         语言提示区域代码 @see #nsPSMDetector ,取值如下:
    *         1 : Japanese
    *         2 : Chinese
    *         3 : Simplified Chinese
    *         4 : Traditional Chinese
    *         5 : Korean
    *         6 : Dont know(default)
    * </pre>
    *
    * @return 文件编码, eg: UTF-8,GBK,GB2312形式(不确定的时候, 返回可能的字符编码序列); 若无, 则
    返回null
    * @throws FileNotFoundException
    * @throws IOException
    */
    public String guessFileEncoding(File file, int languageHint) throws
FileNotFoundException, IOException {
        return guessFileEncoding(file, new nsDetector(languageHint));
    }

    /**
    * 获取文件的编码
    *
    * @param file
    * @param det
    * @return
    * @throws FileNotFoundException
    * @throws IOException
    */
    private String guessFileEncoding(File file, nsDetector det) throws
FileNotFoundException, IOException {
        // Set an observer...
        // The Notify() will be called when a matching charset is found.
        det.Init(new nsICharsetDetectionObserver() {
            public void Notify(String charset) {
                encoding = charset;
                found = true;
            }
        });

        BufferedInputStream imp = new BufferedInputStream(new FileInputStream(file));
        byte[] buf = new byte[1024];
        int len;
        boolean done = false;
        boolean isAscii = false;

        while ((len = imp.read(buf, 0, buf.length)) != -1) {
            // Check if the stream is only ascii.
            isAscii = det.isAscii(buf, len);
            if (isAscii) {
                break;
            }
            // DoIt if non-ascii and not done yet.

```

```

        done = det.DoIt(buf, len, false);
        if (done) {
            break;
        }
    }
    imp.close();
    det.DataEnd();

    if (isAscii) {
        encoding = "ASCII";
        found = true;
    }

    if (!found) {
        String[] prob = det.getProbableCharsets();
        //这里将可能的字符集组合起来返回
        for (int i = 0; i < prob.length; i++) {
            if (i == 0) {
                encoding = prob[i];
            } else {
                encoding += "," + prob[i];
            }
        }

        if (prob.length > 0) {
            // 在没有发现情况下,也可以只取第一个可能的编码,这里返回的是一个可能的序列
            return encoding;
        } else {
            return null;
        }
    }
    return encoding;
}
}

```

编码格式转换源码

- 递归找到目标路径下所有 .java 文件
- 将这些文件格式转换为目标格式

```

public class ChangeEncoding {

    public static void main(String[] args) throws IOException {
        String path = "C:\\Users\\wangk\\Desktop\\多线程并发编程";
        String toEncoding = "utf-8";
        getAllJavaDoc(path, toEncoding);
    }

    private static void getAllJavaDoc(String path, String toEncoding) throws
    IOException {
        File file = new File(path);
        File[] files = file.listFiles();
    }
}

```

```

        for (File file1 : files) {
            if(file1.isDirectory()) {
                getAllJavaDoc(path + "\\\" + file1.getName(), toEncoding);
            } else {
                if(file1.getName().endsWith(".java")) {
                    changeTo(file1,toEncoding);
                }
            }
        }
    }

    private static void changeTo(File file1, String toEncoding) throws IOException {

        String fromEncoding = new FileCharsetDetector().guessFileEncoding(file1);
        if(toEncoding.equalsIgnoreCase(fromEncoding)) return;

        BufferedReader bdf = new BufferedReader(new InputStreamReader(new
FileInputStream(file1),fromEncoding));

        String str=null;
        StringBuilder context = new StringBuilder();
        while((str=bdf.readLine())!=null){
            context.append(str).append("\n");
        }

        BufferedWriter bdw = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream(file1),toEncoding));
        bdw.write(context.toString());
        System.out.println("将" + file1.getPath() + "文件格式从 " + fromEncoding + " 转换
为 " + toEncoding);
        bdw.close();
        bdf.close();
    }
}

```