



链滴

# Hutool - 小而全的 Java 开发工具库推荐

作者: [valarchie](#)

原文链接: <https://ld246.com/article/1585819004832>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Hutool简介

Hutool是一款小而全的开源开发工具类库，在github上拥有将近一万九的star，基本上你能想到的开当中常需要用到的小轮子，基本上都有具备。学习一下hutool工具包可以避免我们在平常开发中重复轮子。这款hutool开源库，更新频率快，jar包小仅1.5Mb。对比其他同款类型的基础工具类库来说是一大优势。因为其他同款类型的基础工具类库多多少少都有基于apache commons做了一些封装。

对于很多小型公司来说，公司内部并没有完善的基础工具类库，使用hutool可以节省了开发人员对项中公用类和公用工具方法的封装时间，使开发专注于业务同时可以最大限度的避免封装不完善带来的bug。

[hutool github地址](#)

## Hutool主要组件

模块	介绍
hutool-aop 面支持	JDK动态代理封装，提供非IOC下的
hutool-bloomFilter 法的布隆过滤	布隆过滤，提供一些Hash
hutool-cache	简单缓存实现
hutool-core il等	核心，包括Bean操作、日期、各种U
hutool-cron 式的定时任务	定时任务模块，提供类Crontab表
hutool-crypto 和摘要算法封装	加密解密模块，提供对称、非对
hutool-db ecord思想	JDBC封装后的数据操作，基于Active
hutool-dfa	基于DFA模型的多关键字查找
hutool-extra 、邮件、Servlet、二维码、Emoji、FTP、分词等)	扩展模块，对第三方封装（模板引
hutool-http 端封装	基于URLConnection的Http客
hutool-log	自动识别日志实现的日志门面
hutool-script	脚本执行封装，例如Javascript
hutool-setting operties封装	功能更强大的Setting配置文件和P
hutool-system	系统参数调用封装（JVM信息等）
hutool-json	JSON实现
hutool-captcha	图片验证码实现
hutool-poi	针对POI中Excel的封装
hutool-socket 装	基于Java的NIO和AIO的Socket

hutool库的工具类非常全，一篇文章难以概括。以下就列举开发中常用到的工具类。

## 常用工具类介绍

工具类内包含的方法都比较齐全，我们就不一一列举，仅列举比较有代表性的方法。

### DateUtil

字符串转日期

DateUtil.parse方法会**自动**识别一些常用格式，包括：

```
yyyy-MM-dd HH:mm:ss  
yyyy-MM-dd  
HH:mm:ss  
yyyy-MM-dd HH:mm  
yyyy-MM-dd HH:mm:ss.SSS
```

```
String dateStr = "2017-03-01";  
Date date = DateUtil.parse(dateStr);
```

我们也可以使用自定义日期格式转化：

```
String dateStr = "2017-03-01";  
Date date = DateUtil.parse(dateStr, "yyyy-MM-dd");
```

计时器（常用于计算代码执行时间）

```
TimeInterval timer = DateUtil.timer();  
  
//-----  
//-----这是执行过程  
//-----  
  
timer.interval();//花费毫秒数  
timer.intervalRestart();//返回花费时间，并重置开始时间  
timer.intervalMinute();//花费分钟数
```

### IoUtil

拷贝:流的读写可以总结为从输入流读取，从输出流写出，这个过程我们定义为拷贝。这个是一个基本程，也是文件、流操作的基础。

以文件流拷贝为例：

```
BufferedInputStream in = FileUtil.getInputStream("d:/test.txt");  
BufferedOutputStream out = FileUtil.getOutputStream("d:/test2.txt");  
long copySize = IoUtil.copy(in, out, IoUtil.DEFAULT_BUFFER_SIZE);
```

### FileUtil

FileUtil内包含以下方法：

- 文件操作：包括文件目录的新建、删除、复制、移动、改名等
- 文件判断：判断文件或目录是否非空，是否为目录，是否为文件等等。
- 绝对路径：针对ClassPath中的文件转换为绝对路径文件。
- 文件名：主文件名，扩展名的获取
- 读操作：包括类似IoUtil中的getReader、readXXX操作
- 写操作：包括getWriter和writeXXX操作

## EnumUtil

首先我们定义一个枚举对象：

```
//定义枚举
public enum TestEnum{
    TEST1("type1"), TEST2("type2"), TEST3("type3");

    private TestEnum(String type) {
        this.type = type;
    }

    private String type;

    public String getType() {
        return this.type;
    }
}
```

getNames

获取枚举类中所有枚举对象的name列表。栗子：

```
//定义枚举
public enum TestEnum {
    TEST1, TEST2, TEST3;
}

List<String> names = EnumUtil.getNames(TestEnum.class);
//结果： [TEST1, TEST2, TEST3]
```

## ImgUtil

scale 缩放图片

提供两种重载方法：其中一个是按照长宽缩放，另一种是按照比例缩放。

```
ImgUtil.scale(
    FileUtil.file("d:/face.jpg"),
    FileUtil.file("d:/face_result.jpg"),
    0.5f//缩放比例
);
```

cut 剪裁图片

```
ImgUtil.cut(  
    FileUtil.file("d:/face.jpg"),  
    FileUtil.file("d:/face_result.jpg"),  
    new Rectangle(200, 200, 100, 100)//裁剪的矩形区域  
);
```

## NetUtil

网络工具类主要包含以下方法

1. longToIpv4 根据long值获取ip v4地址
2. ipv4ToLong 根据ip地址计算出long型的数据
3. isUsableLocalPort 检测本地端口可用性
4. isValidPort 是否为有效的端口
5. isInnerIP 判定是否为内网IP
6. localIpv4s 获得本机的IP地址列表
7. toAbsolutePath 相对URL转换为绝对URL
8. hideIpPart 隐藏掉IP地址的最后一部分为 \* 代替
9. buildInetSocketAddress 构建InetSocketAddress
10. getIpByHost 通过域名得到IP
11. isInner 指定IP的long是否在指定范围内

## SecureUtil

加密工具类主要针对常用加密算法构建快捷方式，还有提供一些密钥生成的快捷工具方法。

对称加密

```
SecureUtil.aes  
SecureUtil.des
```

摘要算法

```
SecureUtil.md5  
SecureUtil.sha1  
SecureUtil.hmac  
SecureUtil.hmacMd5  
SecureUtil.hmacSha1
```

非对称加密

```
SecureUtil.rsa  
SecureUtil.dsa
```

## HtmlUtil

比如我们在使用爬虫爬取HTML页面后，需要对返回页面的HTML内容做一定处理，比如去掉指定标（例如广告栏等）、去除JS、去掉样式等等，这些操作都可以使用HtmlUtil完成。

HtmlUtil.removeHtmlTag 清除指定HTML标签和被标签包围的内容

```
String str = "pre<img src=\"xxx/dfdsfds/test.jpg\">";  
// 结果为: pre  
String result = HtmlUtil.removeHtmlTag(str, "img");
```

HtmlUtil.filter 过滤HTML文本，防止XSS攻击

```
String html = "<alert></alert>";  
// 结果为: ""  
String filter = HtmlUtil.filter(html);
```

HtmlUtil.removeAllHtmlAttr 去除指定标签的所有属性

```
String html = "<div class=\"test_div\" width=\"120\"></div>";  
// 结果为: <div></div>  
String result = HtmlUtil.removeAllHtmlAttr(html, "div");
```

## QrCodeUtil

二维码工具类仅需一行代码

```
// 生成指定url对应的二维码到文件，宽和高都是300像素  
QrCodeUtil.generate("https://hutool.cn/", 300, 300, FileUtil.file("d:/qrcode.jpg"));
```

## TokenizerUtil

中文分词工具类

解析文本并分词

```
//自动根据用户引入的分词库的jar来自动选择使用的引擎  
TokenizerEngine engine = TokenizerUtil.createEngine();  
  
//解析文本  
String text = "这两个方法的区别在于返回值";  
Result result = engine.parse(text);  
//输出: 这两个方法的区别在于返回值  
String resultStr = CollUtil.join((Iterator<Word>)result, " ");
```

## ExcelUtil

Excel操作工具封装

使用

从文件中读取Excel为ExcelReader

```
ExcelReader reader = ExcelUtil.getReader(FileUtil.file("test.xlsx"));
```

从流中读取Excel为ExcelReader（比如从ClassPath中读取Excel文件）

```
ExcelReader reader = ExcelUtil.getReader(ResourceUtil.getStream("aaa.xlsx"));
```

读取指定的sheet

```
ExcelReader reader;
```

```
//通过sheet编号获取
```

```
reader = ExcelUtil.getReader(FileUtil.file("test.xlsx"), 0);
```

```
//通过sheet名获取
```

```
reader = ExcelUtil.getReader(FileUtil.file("test.xlsx"), "sheet1");
```

读取大数据量的Excel

```
private RowHandler createRowHandler() {  
    return new RowHandler() {  
        @Override  
        public void handle(int sheetIndex, int rowIndex, List<Object> rowlist) {  
            Console.log("{} {} {}", sheetIndex, rowIndex, rowlist);  
        }  
    };  
}
```

```
ExcelUtil.readBySax("aaa.xlsx", 0, createRowHandler());
```

更多内容大家有时间的话，可以参阅hutool的官方文档学习。如果在开发过程中需要写轮子的时候不上Hutool文档看看，说不定已经有现成的轮子可以使用。大大节省我们的开发效率。

[Hutool官方文档](#)