



链滴

# Java 时间处理总结, 以及时间工具类的封装

作者: [zhangzeshan](#)

原文链接: <https://ld246.com/article/1602747438135>

来源网站: 链滴

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



## 前景

作为一名刚学java不久的小菜鸟,在写项目的过程中,对时间的处理很不熟练,于是在做完项目之后,决定已封装一个关于时间的工具类,今后在开发过程中就可以拿来就用! 我将挨个记录,并在文章的最后贴上个类的完整代码!

### 获取yyyyMMddHHmmss 格式的时间

```
public static final String YYYYMMDDHHMMSS = "yyyyMMddHHmmss";
private final static SimpleDateFormat sdfTimes = new SimpleDateFormat(YYYYMMDDHHMMSS);
/**
 * 获取yyyyMMddHHmmss格式
 *
 * @return
 */
public static String getSdfTimes() {
    return sdfTimes.format(new Date());
}
```

### 获取YYYY 格式的时间

```
public static final String YYYY = "yyyy";
private final static SimpleDateFormat sdfYear = new SimpleDateFormat(YYYY);
/**
 * 获取YYYY格式(年)
 *
 * @return
 */
public static String getYear() {
```

```
    return sdfYear.format(new Date());
}
```

## 获取MM 格式的时间(月)

```
public static final String MM = "MM";
private final static SimpleDateFormat sdfMM = new SimpleDateFormat(MM);
```

```
/**
 * 获取MM格式
 *
 * @return
 */
public static String getMM() {
    return sdfMM.format(new Date());
}
```

## 获取dd 格式的时间(日)

```
public static final String DD = "dd";
private final static SimpleDateFormat sdfDd = new SimpleDateFormat(DD);
```

```
/**
 * 获取dd格式
 *
 * @return
 */
public static String getDd() {
    return sdfDd.format(new Date());
}
```

## 获取yyyy-MM-dd 格式的时间

```
public static final String YYYY_MM_DD = "yyyy-MM-dd";
private final static SimpleDateFormat sdfDay = new SimpleDateFormat(YYYY_MM_DD);
```

```
/**
 * 获取yyyy-MM-dd格式
 *
 * @return
 */
public static String getDay() {
    return sdfDay.format(new Date());
}
```

## 获取yyyyMMdd 格式的时间

```
public static final String YYYYMMdd = "yyyyMMdd";
private final static SimpleDateFormat sdfDays = new SimpleDateFormat(YYYYMMdd);
```

```
/**
 * 获取yyyyMMdd格式
 *
 * @return
 */
```

```

public static String getDays() {
    return sdfDays.format(new Date());
}

```

## 获取YYYY-MM-DD HH:mm:ss 格式的时间

```

public static final String YYYY_MM_DD_HH_MM_SS = "yyyy-MM-dd HH:mm:ss";
private final static SimpleDateFormat sdfTime = new SimpleDateFormat(YYYY_MM_DD_HH_
M_SS);
/**
 * 获取YYYY-MM-DD HH:mm:ss格式
 *
 * @return
 */
public static String getTime() {
    return sdfTime.format(new Date());
}

```

## 日期比较 ,如果s > = e 返回true 否则返回false

```

public static boolean compareDate(String s, String e) {
    if (fomatDate(s) == null || fomatDate(e) == null) {
        return false;
    }
    return fomatDate(s).getTime() >= fomatDate(e).getTime();
}

```

## String类型格式化成yyyy-MM-dd的Date类型

```

public static Date fomatDate(String date) {
    DateFormat fmt = new SimpleDateFormat(YYYY_MM_DD);
    try {
        return fmt.parse(date);
    } catch (ParseException e) {
        e.printStackTrace();
        public static final String YYYY_MM_DD_HH_MM_SS = "yyyy-MM-dd HH:
m:ss";Trace();
        return null;
    }
}

```

## String类型格式化成yyyy-MM-dd HH:mm:ss的Date类型

```

public static Date fomatDateToYyyyMMddHHmmss(String date) {
    DateFormat fmt = new SimpleDateFormat(YYYY_MM_DD_HH_MM_SS);
    try {
        return fmt.parse(date);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

```

## 校验日期是否合法

```
public static boolean isValidDate(String s) {
    DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");
    try {
        fmt.parse(s);
        return true;
    } catch (Exception e) {
        return false; // 如果throw java.text.ParseException或者NullPointerException, 就说明格
    }
}
不对
}
```

## 时间相减得到天数

```
public static long getDaySub(String beginDateStr, String endDateStr) {
    long day = 0;
    SimpleDateFormat format = new SimpleDateFormat("YYYY_MM_DD");
    Date beginDate = null;
    Date endDate = null;
    try {
        beginDate = format.parse(beginDateStr);
        endDate = format.parse(endDateStr);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    day = (endDate.getTime() - beginDate.getTime()) / (24 * 60 * 60 * 1000);
    //System.out.println("相隔的天数="+day);
    return day;
}
```

## 得到n天之后的日期

```
public static String getAfterDayDate(String days) {
    int daysInt = Integer.parseInt(days);
    Calendar canlendar = Calendar.getInstance(); // java.util包
    canlendar.add(Calendar.DATE, daysInt); // 日期减 如果不够减会将月变动
    Date date = canlendar.getTime();
    SimpleDateFormat sdfd = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String dateStr = sdfd.format(date);
    return dateStr;
}
```

## 得到n天之后是周几

```
public static String getAfterDayWeek(String days) {
    int daysInt = Integer.parseInt(days);
    Calendar canlendar = Calendar.getInstance(); // java.util包
    canlendar.add(Calendar.DATE, daysInt); // 日期减 如果不够减会将月变动
    Date date = canlendar.getTime();
    SimpleDateFormat sdf = new SimpleDateFormat("E");
    String dateStr = sdf.format(date);
}
```

```
    return dateStr;
}
```

### 按照yyyy-MM-dd HH:mm:ss的格式，日期转字符串

```
public static String date2Str(Date date) {
    return date2Str(date, "yyyy-MM-dd HH:mm:ss");
}
```

### 按照yyyy-MM-dd HH:mm:ss的格式，字符串转日期

```
public static Date str2Date(String date) {
    if (Tools.notEmpty(date)) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        try {
            return sdf.parse(date);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return new Date();
    } else {
        return null;
    }
}
```

### 按照yyyy-MM-dd的格式，字符串转日期

```
public static Date strToDate(String date) {
    if (Tools.notEmpty(date)) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        try {
            return sdf.parse(date);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return new Date();
    } else {
        return null;
    }
}
```

### 把时间根据时、分、秒转换为时间段

```
public static String getTimes(String StrDate) {
    String resultTimes = "";
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    Date now;
    try {
        now = new Date();
        Date date = df.parse(StrDate);
        long times = now.getTime() - date.getTime();
        long day = times / (24 * 60 * 60 * 1000);
    }
}
```

```

    long hour = (times / (60 * 60 * 1000) - day * 24);
    long min = ((times / (60 * 1000)) - day * 24 * 60 - hour * 60);
    long sec = (times / 1000 - day * 24 * 60 * 60 - hour * 60 * 60 - min * 60);

    StringBuffer sb = new StringBuffer();
    //sb.append("发表于: ");
    if (hour > 0) {
        sb.append(hour + "小时前");
    } else if (min > 0) {
        sb.append(min + "分钟前");
    } else {
        sb.append(sec + "秒前");
    }
    resultTimes = sb.toString();
} catch (ParseException e) {
    e.printStackTrace();
}
return resultTimes;
}

```

## 按照参数format的格式，日期转字符串

```

public static String date2Str(Date date, String format) {
    if (date != null) {
        SimpleDateFormat sdf = new SimpleDateFormat(format);
        return sdf.format(date);
    } else {
        return "";
    }
}

```

好了,现在贴上这个工具类的整个代码:

```

package com.center.util;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;

/**
 * 说明：日期处理
 */
public class DateUtil {

    public static final String YYYY = "yyyy";
    public static final String MM = "MM";
    public static final String DD = "dd";
    public static final String YYYY_MM_DD = "yyyy-MM-dd";
    public static final String YYYY_MM_DD_HH_MM_SS = "yyyy-MM-dd HH:mm:ss";
    public static final String YYYYMMdd = "yyyyMMdd";
    public static final String YYYYMMDDHHMMSS = "yyyyMMddHHmmss";
}

```

```
private final static SimpleDateFormat sdfYear = new SimpleDateFormat(YYYY);
private final static SimpleDateFormat sdfMM = new SimpleDateFormat(MM);
private final static SimpleDateFormat sdfDd = new SimpleDateFormat(DD);
private final static SimpleDateFormat sdfDay = new SimpleDateFormat(YYYY_MM_DD);
private final static SimpleDateFormat sdfDays = new SimpleDateFormat(YYYYMMdd);
private final static SimpleDateFormat sdfTime = new SimpleDateFormat(YYYY_MM_DD_HH
MM_SS);
private final static SimpleDateFormat sdfTimes = new SimpleDateFormat(YYYYMMDDHH
MSS);
```

```
/**
 * 获取yyyyMMddHHmmss格式
 *
 * @return
 */
public static String getSdfTimes() {
    return sdfTimes.format(new Date());
}
```

```
/**
 * 获取YYYY格式
 *
 * @return
 */
public static String getYear() {
    return sdfYear.format(new Date());
}
```

```
/**
 * 获取MM格式
 *
 * @return
 */
public static String getMM() {
    return sdfMM.format(new Date());
}
```

```
/**
 * 获取dd格式
 *
 * @return
 */
public static String getDd() {
    return sdfDd.format(new Date());
}
```

```
/**
 * 获取yyyy-MM-dd格式
 *
 * @return
 */
public static String getDay() {
```



```

    return sdfDay.format(new Date());
}

/**
 * 获取yyyyMMdd格式
 *
 * @return
 */
public static String getDays() {
    return sdfDays.format(new Date());
}

/**
 * 获取YYYY-MM-DD HH:mm:ss格式
 *
 * @return
 */
public static String getTime() {
    return sdfTime.format(new Date());
}

/**
 * @param s
 * @param e
 * @return boolean
 * @throws
 * @Title: compareDate
 * @Description: (日期比较, 如果s >= e 返回true 否则返回false)
 * @author fh
 */
public static boolean compareDate(String s, String e) {
    if (fomatDate(s) == null || fomatDate(e) == null) {
        return false;
    }
    return fomatDate(s).getTime() >= fomatDate(e).getTime();
}

/**
 * String类型格式化成yyyy-MM-dd的Date类型
 *
 * @return
 */
public static Date fomatDate(String date) {
    DateFormat fmt = new SimpleDateFormat("YYYY_MM_DD");
    try {
        return fmt.parse(date);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

/**
 * String类型格式化成yyyy-MM-dd HH:mm:ss的Date类型

```

```

*
* @return
*/
public static Date formatDateToYyyyMMddHHmmss(String date) {
    DateFormat fmt = new SimpleDateFormat("YYYY_MM_DD_HH_MM_SS");
    try {
        return fmt.parse(date);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

/**
 * 校验日期是否合法
 *
 * @return
 */
public static boolean isValidDate(String s) {
    DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");
    try {
        fmt.parse(s);
        return true;
    } catch (Exception e) {
        return false; // 如果throw java.text.ParseException或者NullPointerException, 就说明格
        不对
    }
}

/**
 * @param startTime
 * @param endTime
 * @return
 */
public static int getDiffYear(String startTime, String endTime) {
    DateFormat fmt = new SimpleDateFormat("yyyy-MM-dd");
    try {
        int years = (int) (((fmt.parse(endTime).getTime() - fmt.parse(startTime).getTime()) / (10
        0 * 60 * 60 * 24)) / 365);
        return years;
    } catch (Exception e) {
        return 0; // 如果throw java.text.ParseException或者NullPointerException, 就说明格
        不对
    }
}

/**
 * <li>功能描述: 时间相减得到天数
 *
 * @param beginDateStr
 * @param endDateStr
 * @return long
 * @author Administrator
 */

```

```

public static long getDaySub(String beginDateStr, String endDateStr) {
    long day = 0;
    SimpleDateFormat format = new SimpleDateFormat("YYYY-MM-DD");
    Date beginDate = null;
    Date endDate = null;
    try {
        beginDate = format.parse(beginDateStr);
        endDate = format.parse(endDateStr);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    day = (endDate.getTime() - beginDate.getTime()) / (24 * 60 * 60 * 1000);
    //System.out.println("相隔的天数="+day);
    return day;
}

/**
 * 得到n天之后的日期
 *
 * @param days
 * @return
 */
public static String getAfterDayDate(String days) {
    int daysInt = Integer.parseInt(days);
    Calendar canlendar = Calendar.getInstance(); // java.util包
    canlendar.add(Calendar.DATE, daysInt); // 日期减 如果不够减会将月变动
    Date date = canlendar.getTime();
    SimpleDateFormat sdfd = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String dateStr = sdfd.format(date);
    return dateStr;
}

/**
 * 得到n天之后是周几
 *
 * @param days
 * @return
 */
public static String getAfterDayWeek(String days) {
    int daysInt = Integer.parseInt(days);
    Calendar canlendar = Calendar.getInstance(); // java.util包
    canlendar.add(Calendar.DATE, daysInt); // 日期减 如果不够减会将月变动
    Date date = canlendar.getTime();
    SimpleDateFormat sdf = new SimpleDateFormat("E");
    String dateStr = sdf.format(date);
    return dateStr;
}

/**
 * 按照yyyy-MM-dd HH:mm:ss的格式，日期转字符串
 *
 * @param date
 * @return yyyy-MM-dd HH:mm:ss
 */

```

```

public static String date2Str(Date date) {
    return date2Str(date, "yyyy-MM-dd HH:mm:ss");
}

/**
 * 按照yyyy-MM-dd HH:mm:ss的格式，字符串转日期
 *
 * @param date
 * @return
 */
public static Date str2Date(String date) {
    if (Tools.isEmpty(date)) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        try {
            return sdf.parse(date);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return new Date();
    } else {
        return null;
    }
}

/**
 * 按照yyyy-MM-dd的格式，字符串转日期
 *
 * @param date
 * @return
 */
public static Date strToDate(String date) {
    if (Tools.isEmpty(date)) {
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        try {
            return sdf.parse(date);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return new Date();
    } else {
        return null;
    }
}

/**
 * 把时间根据时、分、秒转换为时间段
 *
 * @param StrDate
 */
public static String getTimes(String StrDate) {
    String resultTimes = "";
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    Date now;
    try {

```

```

    now = new Date();
    Date date = df.parse(StrDate);
    long times = now.getTime() - date.getTime();
    long day = times / (24 * 60 * 60 * 1000);
    long hour = (times / (60 * 60 * 1000) - day * 24);
    long min = ((times / (60 * 1000)) - day * 24 * 60 - hour * 60);
    long sec = (times / 1000 - day * 24 * 60 * 60 - hour * 60 * 60 - min * 60);

    StringBuffer sb = new StringBuffer();
    //sb.append("发表于: ");
    if (hour > 0) {
        sb.append(hour + "小时前");
    } else if (min > 0) {
        sb.append(min + "分钟前");
    } else {
        sb.append(sec + "秒前");
    }
    resultTimes = sb.toString();
} catch (ParseException e) {
    e.printStackTrace();
}
}
return resultTimes;
}

/**
 * 按照参数format的格式, 日期转字符串
 *
 * @param date
 * @param format
 * @return
 */
public static String date2Str(Date date, String format) {
    if (date != null) {
        SimpleDateFormat sdf = new SimpleDateFormat(format);
        return sdf.format(date);
    } else {
        return "";
    }
}

/**
 * 获取月份第一天
 */
public static String getFirstDayDateOfMonth(Date date) {
    Calendar cal = Calendar.getInstance();
    cal.setTime(date);
    int last = cal.getActualMinimum(Calendar.DAY_OF_MONTH);
    cal.set(Calendar.DAY_OF_MONTH, last);
    return date2Str(cal.getTime(), "yyyy-MM-dd");
}

public static Date getFirstDayDateOfMonth() {
    Calendar cal = Calendar.getInstance();

```

```

    cal.setTime(new Date());
    int last = cal.getActualMinimum(Calendar.DAY_OF_MONTH);
    cal.set(Calendar.DAY_OF_MONTH, last);
    return fomateDate(date2Str(cal.getTime(), "yyyy-MM-dd"));
}

public static Date getSubFirstDayDateOfMonth() {
    Calendar calendar1 = Calendar.getInstance();
    calendar1.add(Calendar.MONTH, -1);
    calendar1.set(Calendar.DAY_OF_MONTH, 1);
    return fomateDate(date2Str(calendar1.getTime(), "yyyy-MM-dd"));
}

public static String getFirstDayDateOfWeek(Date date) {
    Calendar cal = Calendar.getInstance();
    cal.setTime(date);
    int last = cal.getActualMinimum(Calendar.DAY_OF_WEEK);
    cal.set(Calendar.DAY_OF_WEEK, last);
    return date2Str(cal.getTime(), "yyyy-MM-dd");
}

public static List<Date> findDates(Date dBegin, Date dEnd) {
    List lDate = new ArrayList();
    lDate.add(dBegin);
    Calendar calBegin = Calendar.getInstance();
    // 使用给定的 Date 设置此 Calendar 的时间
    calBegin.setTime(dBegin);
    Calendar calEnd = Calendar.getInstance();
    // 使用给定的 Date 设置此 Calendar 的时间
    calEnd.setTime(dEnd);
    // 测试此日期是否在指定日期之后
    while (dEnd.after(calBegin.getTime())) {
        // 根据日历的规则, 为给定的日历字段添加或减去指定的时间量
        calBegin.add(Calendar.DAY_OF_MONTH, 1);
        lDate.add(calBegin.getTime());
    }
    return lDate;
}

public static Date addOneDay(Date date) {
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(date);
    calendar.add(Calendar.DATE, 1);
    return calendar.getTime();
}

public static Date subDay(Integer sub) {
    Date date = new Date();
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(date);
    calendar.add(Calendar.DATE, sub);
    return fomateDate(date2Str(calendar.getTime(), "yyyy-MM-dd"));
}

```

```

public static String getTimeByCalendar(Date date) {
    Calendar cal = Calendar.getInstance();
    cal.setTime(date);
    int year = cal.get(Calendar.YEAR);//获取年份
    int month = cal.get(Calendar.MONTH);//获取月份
    int day = cal.get(Calendar.DATE);//获取日
    int hour = cal.get(Calendar.HOUR);//小时
    int minute = cal.get(Calendar.MINUTE);//分
    int second = cal.get(Calendar.SECOND);//秒
    int WeekOfYear = cal.get(Calendar.DAY_OF_WEEK);//一周的第几天
    return "" + month + "月" + day + "日" + hour + "时" + minute + "分" + second + "秒";
}

public static Date getToDay() {
    return fomateDate(getDay());
}

/**
 * 获取分
 *
 * @return
 */
public static Long getMinute() {
    return System.currentTimeMillis() / (1000 * 60);
}

/**
 * 获取秒
 *
 * @return
 */
public static Long getSecond() {
    return System.currentTimeMillis() / 1000;
}

/**
 * 获取毫秒
 *
 * @return
 */
public static Long getMillisecond() {
    return System.currentTimeMillis();
}

/**
 * 传入具体日期，返回具体日期。
 * @param date 日期(2017-04-13)
 */
/**
 * 年月日减/加天数得到年月日
 */

```

```

* @param date
* @return
* @throws ParseException
*/
public static String stringSubDay(String date, int num) throws ParseException {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Date temp = sdf.parse(date);
    Calendar rightNow = Calendar.getInstance();
    rightNow.setTime(temp);
    rightNow.add(Calendar.DATE, num);
    return sdf.format(rightNow.getTime());
}

```

```

/**
 * 获取某个月份
 *
 * @param i
 * @return
 */
public static String getLast12Months(int i) {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar c = Calendar.getInstance();
    c.setTime(new Date());
    c.add(Calendar.MONTH, -i);
    Date m = c.getTime();
    return sdf.format(m);
}

```

```

/**
 * 返回之后n天
 * @param time
 * @param i
 * @return
 * @throws ParseException
 */
public static String getNextTime(String time , int i) throws ParseException {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar cal = Calendar.getInstance();
    cal.setTime(sdf.parse(time));
    cal.add(Calendar.DATE, i);
    return sdf.format(cal.getTime());
}

```

```

/**
 * 获取之前n天
 * @param time
 * @param i
 * @return
 * @throws ParseException
 */
public static String getPrevTime(String time , int i) throws ParseException {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

```



```

    Calendar cal = Calendar.getInstance();
    cal.setTime(sdf.parse(time));
    cal.add(Calendar.DATE, -i);
    return sdf.format(cal.getTime());
}

/**
 * 返回之后n个月
 * @return
 */
public static String getNextMonth(String time,int i) throws ParseException {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar cal = Calendar.getInstance();
    cal.setTime(sdf.parse(time));
    cal.add(Calendar.MONTH, i);
    return sdf.format(cal.getTime());
}

/**
 * 返回之前n个月
 * @return
 */
public static String getPrevMonth(String time,int i) throws ParseException {
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar cal = Calendar.getInstance();
    cal.setTime(sdf.parse(time));
    cal.add(Calendar.MONTH, -i);
    return sdf.format(cal.getTime());
}

public static void main(String[] args) {
    System.out.println(getSecond());
}
}

```