java日期操作类：

commons-lang的api，

<http://commons.apache.org/proper/commons-lang/javadocs/api-release/index.html>

总：

使用1，4和需要的3，并自己写基于这些的公共时间类；

///////////////////////////////////////////////////////////////////////////////////////

1，当前时间：

主要：

·获得当前时间；

·日期比较；

date类， ---**import** java.util.date

//类中设置当前时间； ---一般实体类中就如此使用

Date currentTime = new Date();

获得系统时间： ---毫秒表示，long类型

Long time = System.*currentTimeMillis*() //可以以如此形式保存在数据库；

//转换为日期格式；

**new** Date(System.*currentTimeMillis*());

**new** Date().getTime() 等于 System.*currentTimeMillis*();

日期比较：

//转换为毫秒比较，

**if**(date11.getTime() > date2.getTime()){

2，---废弃，不使用；

~~SimpleDateFormat类：---~~**~~import~~** ~~java.text.SimpleDateFormat;~~

~~·继承DateFormat类，所以一般使用这个，不使用DateFormat；~~

~~日期：~~

~~SimpleDateFormat formatter =~~ **~~new~~** ~~SimpleDateFormat("yyyy-MM-dd");~~

~~String dateString = formatter.format(currentTime);~~

~~---~~

~~时间：~~

~~SimpleDateFormat formatter =~~ **~~new~~** ~~SimpleDateFormat("yyyy-MM-dd HH:mm:ss");~~

~~时分秒：~~

~~SimpleDateFormat formatter =~~ **~~new~~** ~~SimpleDateFormat("HH:mm:ss");~~

~~汉字格式，~~

~~SimpleDateFormat myFmt =~~ **~~new~~** ~~SimpleDateFormat("yyyy年MM月dd日 HH时mm分ss秒");~~

~~~~~

~~更多参数，见官方文档，~~

~~<http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html>~~

~~w 一年第多少周，E 星期几，......~~

~~SimpleDateFormat myFmt =~~ **~~new~~** ~~SimpleDateFormat("yyyy年MM月dd日 HH时mm分ss秒 一年第w周 E");~~

~~---~~

~~将字符串转换为时间：~~

~~时间字符串strDate；~~

~~String strDate = "2000-02-28";~~

~~SimpleDateFormat formatter =~~ **~~new~~** ~~SimpleDateFormat("yyyy-MM-dd");~~

~~ParsePosition pos =~~ **~~new~~** ~~ParsePosition(0);~~

~~Date strtodate = formatter.parse(strDate, pos);~~

3，日历时间，

Calendar类， ---**import** java.util.Calendar

主要：

·取年月日等；

·加减年月日等；

·和标准日期的转换；

//当前日期，

Calendar calendar = **new** GregorianCalendar();

System.***out***.println("年: " + calendar.get(Calendar.***YEAR***));

System.***out***.println("当月第几天: " + calendar.get(Calendar.***DAY\_OF\_MONTH***));

System.***out***.println("当周第几天: " + calendar.get(Calendar.***DAY\_OF\_WEEK***));

System.***out***.println("当年第几周: " + calendar.get(Calendar.***WEEK\_OF\_YEAR***));

//放入日期，---可以放入上面的strtodate，

Calendar calendar2 = Calendar.*getInstance*(); //也可以获得当前日期；

Date trialTime = **new** Date();

calendar2.setTime(trialTime);

//可以将Calendar.***YEAR****写为*calendar2.*YEAR，一样；*

System.***out***.println("年: " + calendar2.get(Calendar.***YEAR***));

System.***out***.println("时间: " + calendar2.getTime().toString());

//加入计算，add方法，

calendar2.add(Calendar.***YEAR***, -3);// 减去年份

System.***out***.println(calendar2.getTime().toString());// 得到日期

//滚轮所设置的日期变量，其它不变； | 12月1号，回滚1天，则为31号；| 月数不变，还是12；

calendar2.roll(Calendar.***DATE***, -1);

//闰年相关，会自动进行闰年相关处理

calendar2.add(Calendar.***MONTH***, -1);// 减去月份

System.***out***.println(calendar2.getTime().toString());// 得到日期

//计算闰年

**int** year2 = calendar2.get(Calendar.*YEAR*);

**boolean** b = (**new** GregorianCalendar()).isLeapYear(year2);

**if**(b){

System.*out*.println("是閏年");

}**else**{

System.*out*.println("不是閏年");

}

-----------------------------------

4，

commons lang3 ---java公共类：

**import** org.apache.commons.lang3.time.DateFormatUtils;

主要：

·格式化日期；

·字符串转换日期；

·加年月日等；

·日期间隔；---相差天数；

//String格式标准当前时间，

DateFormatUtils.*format*(System.*currentTimeMillis*(),"yyyy-MM-dd HH:mm:ss")

或，

DateFormatUtils.format(**new** Date(), "yyyy-MM-dd HH:mm:ss");

获得年：---笨办法，

DateFormatUtils.format(**new** Date(), "yyyy");

//设定时间显示样式，

System.***out***.println(DateFormatUtils.*format*(System.*currentTimeMillis*(),

"yyyy年MM月dd日"));

//将string格式的转换为date，

·公共类方法中的，*parseDate方法；*

*·或直接使用，*

String dateStr = "2014-12-05 23:13:45";

Date date1 = DateUtils.*parseDate*(dateStr, "yyyy-MM-dd HH:mm:ss");

System.***out***.println(date1);

**import** org.apache.commons.lang3.time.DateUtils; //源码实现使用Calendar；

//加天

Date trialTime = **new** Date();

System.***out***.println(DateUtils.*addDays*(trialTime, 1));

//月，年

addMonths addYears

Ceiling---向上取整；

如：*ceiling*(**new** Date(), Calendar.***DATE***)) //日期向上，则为下一天的0:00:00；

//设定日期中的某个值，| 本例：天为12；

System.***out***.println(DateUtils.*setDays*(**new** Date(), 12));

//指定位置取1，年月日0、月日1、日2， | 本例：当月第一天；

System.***out***.println(DateUtils.*truncate*(**new** Date(), 2));

//比较同一天，

DateUtils.*isSameDay*(**new** Date(time), **new** Date())

////////////////////////////////////////////////////////////////////////////////////////////////

写一个公共类：MyDateUtils.java | 继承commons lang3；

·简化转换写法；---隐藏代码实现；

·一些复杂逻辑的处理方法；

使用：

//获得当前日期

System.***out***.println(MyDateUtils.*getDate*());

//日期格式化

Date trialTime = **new** Date();

System.***out***.println(MyDateUtils.*getDateTime*(trialTime));

//字符串转换为日期

String strDate = "2000/02/28";

System.***out***.println(MyDateUtils.*parseDate*(strDate).toString());

//获取过去的天数

Date oldDate = MyDateUtils.*parseDate*(strDate);

System.***out***.println(MyDateUtils.*pastDays*(oldDate));

//获得两个日期的相隔天数 ---是按照标准的24小时间隔算一天的；

String strDate = "2000/03/01";

Date oldDate1 = MyDateUtils.*parseDate*(strDate);

String strDate2 = "2000/02/27";

Date oldDate2 = MyDateUtils.*parseDate*(strDate2);

System.***out***.println(MyDateUtils.*distanceDays*(oldDate1, oldDate2));

~

如果要按日来计算相隔，则需要先全部取年月日，再计算；

//日期时间转换为年月日

String dateStr = "2014-12-05";

Date date1 = DateUtils.*parseDate*(dateStr, "yyyy-MM-dd");

System.***out***.println(date1);

String dateStr2 = DateFormatUtils.*format*(**new** Date(), "yyyy-MM-dd");

Date date2 = DateUtils.*parseDate*(dateStr2, "yyyy-MM-dd");

System.***out***.println(date2);

**int** dayNum = MyDateUtils.*distanceDays*(strtodate,nowtoDate);

//两个日期的相隔：

相隔月数，简单，

(年相减)\*12+月相减；

~

相隔年月日，太麻烦，不计算；

//月开始时间；

1 0:00:00

//返回date，

System.***out***.println(DateUtils.*truncate*(**new** Date(), 2));

~

方法2，---时间相同；

Date trialTime = **new** Date();

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(trialTime);

calendar.set(Calendar.***DATE***, 1);

System.***out***.println(calendar.getTime());

//月结束时间；

23:59:59

方法1，//月开始时间+1月-1秒；

Date firstTime = DateUtils.*truncate*(**new** Date(), 2);

Date nextTime = DateUtils.*addMonths*(firstTime, 1);

Date endTime = DateUtils.*addSeconds*(nextTime, -1);

System.***out***.println(endTime);

方法2，---时间相同；

//日期转换为，---可以放入任意日期，

Date trialTime = **new** Date();

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(trialTime);

calendar.set(Calendar.***DATE***, calendar.getActualMaximum(Calendar.***DATE***));

System.***out***.println(calendar.getTime());

//天开始时间 ---返回date

Calendar currentDate = **new** GregorianCalendar();

currentDate.set(Calendar.***HOUR\_OF\_DAY***, 0);

currentDate.set(Calendar.***MINUTE***, 0);

currentDate.set(Calendar.***SECOND***, 0);

System.***out***.println(currentDate.getTime());

//天结束时间

Calendar currentDate = **new** GregorianCalendar();

currentDate.set(Calendar.***HOUR\_OF\_DAY***, 23);

currentDate.set(Calendar.***MINUTE***, 59);

currentDate.set(Calendar.***SECOND***, 59);

System.***out***.println(currentDate.getTime());

//月的天数，

//date加入Calendar 中，

Calendar calendar = Calendar.*getInstance*();

calendar.setTime(date1);

System.***out***.println(calendar.getActualMaximum(Calendar.***DATE***));

/////////////////////////////////////////

实体类中，转化页面时间显示样式：

**private** Date tAddtime; //原数据表对应的实体类的时间变量

**private** String tAddtimeStr; //声明一个String变量；

//使用org.apache.commons.lang3.time.DateFormatUtils | 或写成公共方法，调用；

**public** String gettAddtimeStr() {

**if** (gettAddtime() != **null**) {

**return** DateFormatUtils.*format*(gettAddtime(), "yyyy-MM-dd HH:mm:ss")

}

**return** tAddtimeStr;

}