链滴

40 Commons Lang 常用类

作者：Gao-Eason

原文链接：https://ld246.com/article/1649669380597

来源网站：链滴

40 Commons Lang常用类

StringUtils

- **StringUtils 是提供字符串操作的工具类，提供的方法如下：**
- boolean isEmpty(String str)：**如果参数 str 为 NULL 或者 str.length() == 0 返回 true**
- boolean isNotEmpty(String str)：**判断给定参数是否不为空**
- boolean isBlank(String str)：**如果参数 str 为 NULL 或者其长度等于 0，又或者其由空格组成，么此方法都返回 true。**
- boolean isNotBlank(String str)
- boolean isAnyBlank(CharSequence... css)
- boolean isNoneBlank(CharSequence... css)
- boolean isAnyEmpty(CharSequence... css)
- boolean isNotEmpty(final CharSequence cs)
- String trim(String str)：**去除字符串开头和结尾处的空格字符（如果参数 str 为 null，则返回 null)**
- String stripStart(String str, String stripChars)：**去掉 str 前端的在 stripChars 中的字符**
- String stripEnd(String str, String stripChars)：**去掉 str 末端的在 stripChars 中的字符**
- int ordinalIndexOf(String str, String searchStr, int ordinal)**：返回字符串 search 在字符串 str 中第 ordinal 次出现的位置（如果 str=null 或 searchStr=null 或 ordinal<=0 则返回-1）**

StringEscapeUtils

- **StringEscapeUtils 这个类里提供了很多转义的方法，比如可以转成 json、xml、html 等格式**

- **已被 org.apache.commons.text.StringEscapeUtils 取代**
- String escapeJava(String str)：**使用 java String rules 转义给定字符串 str（转为 unicode 编码）**
- String escapeJavaScript(String str)：**与 escapeJava() 方法的唯一区别是，在 javaScript 中，引号必须被转义**
- String escapeHtml4(String input)

ArrayUtils

- **全局静态常量：EMPTY*** __ARRAY（根据***** 处的类型，返回对应的长度为 0 的数组）、INDEX NOT_FOUND（-1，表示数组下标未找到）**
- String toString(Object array, String stringIfNull)：**如果为空，返回 stringIfNull**
- int hashCode(Object array)：**使用 HashCodeBuilder 返回数组的 hashcode**
- boolean isEquals(Object array1, Object array2)：**用 EqualsBuilder 返回两个数组比较的结果**
- Map<Object, Object> toMap(Object[] array)：**Converts the given array into a Map. 作为数的数组有两个选择：一是成员为 Map.Entry，然后通过遍历该数组，把 Map.Entry 拆分并分别放新生成 Map 的 Key 和 Value 中；二是成员为长度大于等于 2 的数组，位置 0 的元素作为 key，位 1 的元素作为 value**
- xxx[] clone(xxx[] array)：**如果不为空，则使用参数自己的 clone 方法处理**
- Object[] subarray(Object[] array, int startIndexInclusive, int endIndexExclusive)：**数组拷贝**
- void reverse(Object[] array, int startIndexInclusive, int endIndexExclusive)：**数组元素反转（数组前后两个坐标 i, j 开始，交换数据，并向中间移动，当 i>j 时停止）**
- int indexOf(Object[] array, Object objectToFind, int startIndex)：**查找数组元素位置，return -1 if not found or null array input**
- boolean contains(Object[] array, Object objectToFind)：**查找元素是否存在数组中**
- boolean isEmpty(Object[] array)：**判断是否为空，length=0 或 null 都属于空**
- Xxx[] toObject(xxx[] array)：**将基本类型数组转为包装类型数组**
- Object[] addAll(Object[] array1, Object... array2)：**并集操作，合并数组**

DateUtils

- **全局静态常量：MILLIS_PER_SECOND、MILLIS_PER_SECOND、MILLIS_PER_MINUTE、MILLS_PER_HOUR、MILLIS_PER_DAY**
- **日期比较**boolean isSameDay(Date date1, Date date2)``boolean isSameDay(Calendar cal1, Calendar cal2)：**比较日期是否相同，忽略 time（通过比较 Calendar.ERA、YEAR、DAY_OF_YEAR 个属性判断给定日期是否相同）**
- **时间比较**boolean isSameInstant(Date date1, Date date2)``boolean isSameInstant(Calendar cal1, Calendar cal2)：**比较时间是否相同（通过 Date 类中的 getTime() 方法）**
- **add 族**Date addYears(Date date, int amount)：**在给定日期 date 的基础上添加 amount 年，回新的对象**Date addMonths(Date date, int amount)：**添加月**Date addWeeks(Date date, int amount)：**添加周**Date addDays(Date date, int amount)：**添加日**Date addHours(Date date, int amont)：**添加小时**Date addMinutes(Date date, int amount)：**添加分钟**Date addSeconds(Date date, nt amount)：**添加秒**Date addMilliseconds(Date date, int amount)：**添加毫秒**
- **set 族**Date setYears(Date date, int amount)：**为 date 设置新的年份信息，并返回一个新的对象对象 date 未发生改变**Date setMonths(Date date, int amount)：**设置月份**Date setDays(Date date

int amount)：**设置日期**Date setHours(Date date, int amount)：**设置小时**Date setMinutes(Date ate, int amount)：**设置分钟**Date setSeconds(Date date, int amount)：**设置秒**Date setMilliseconds(Date date, int amount)：**设置毫秒**

- **round 族、truncate 族、ceil 族：日期取整（日期精度调节，如调节至秒/分等）**Date round(Dat date, int field)：**相当于数学中的四舍五入法取整**Date truncate(Date date, int field)：**相当于去余取整**Date ceiling(Date date, int field)：**相当于向上取整**

DateFormatUtils

- **与 JDK 的 SimpleDateFormat 相比，其主要优点是：线程安全，常用的方法 ** String format(ate date, String pattern)
- **DateFormatUtils 定义了很多内置的固定日期格式，均为 FastDateFormat 类型，如 ISO_DATE_ ORMAT**
- **使用 FastDateFormat 的 format() 方法可以直接将日期格式化为内置的固定格式 ** String for at (Date date)

NumberUtils

- **为 JDK 中的 Number 类提供额外的功能**
- **提供可复用的值为 0, 1 的数值型的包装类，包括 Long、Integer、Short、Byte、Double、Float**
- boolean isParsable(String str)：**判断字符串是否可以转换为 number**
- boolean isDigits(String str)：**判断字符串是否是只包含数字字符**
- Xxx toXxx(String str, xxx defaultValue)：**将给定的字符串装换成 xxx 类型的数值类型，包括 Lon \Integer\Short\Byte\Double\Float，如果指定了默认值，那么当 String 为空，或是转换发生异，则返回指定的默认值 xxx**
- Xxx createXxx(String str)：**用给定的字符串创建 Xxx 数值类型的对象，包括 Float\Double\Inte er\Long\BigInteger\BigDecimal**
- Xxx min(Xxx[] array)：**从类型为 Xxx 的数组中找出最小的，Xxx 可为 long\float\double\byte\ hort\int，时间复杂度是 O(n)**
- Xxx max(Xxx[] array)：**从类型为 Xxx 的数组中找出最大的**

RandomUtils

- **随机数据生成类，包括浮点，双精，布尔，整形，长整在内的随机数生成**
- int nextInt(int startInclusive, int endExclusive) ，**类似方法：nextLong()、nextBoolean()、n xtFloat()、nextDouble()**

RandomStringUtils

StopWatch

Pair\、Triple

- **Pair\ implements Map.Entry\，表示由两个元素组成的一个配对，实现类：ImmutablePair、**

utablePair

- **Triple\，表示由 3 个元素组成的一个组，实现类：ImmutableTriple、MutableTriple**
- **类方法**Pair<L, R> of(final L left, final R right)``Triple<L, M, R> of(final L left, final M middle, final R right)
- **实例方法：**L getLeft()、M getMiddle()、R getRight()

反射相关

FieldUtils

- **通过反射技术来操作成员变量**
- **带有 Declared 的仅考虑当前类的字段，其它情况会考虑当前类实现的接口以及其父类的字段**
- **获取字段：getField()、getDeclaredField()**
- **获取字段的值：readStaticField()、readDeclaredStaticField()、readField()、readDeclaredField()**
- **设置字段的值：writeStaticField()、writeDeclaredStaticField()、writeField()、writeDeclareField()**

ClassUtils

ConstructorUtils

MemberUtils

- **调用方法：invokeMethod()、invokeStaticMethod()**

CharUtils

- boolean isAscii(char ch)：**判断是否为 ASCII 字符，**ch < 128
- boolean isAsciiAlpha(char ch)：**判断是否为 ASCII 字母，**(ch >= 'A'&& ch <= 'Z') || (ch >= 'a' & CH <= 'z')
- Boolean isAsciiAlphaLower(char ch)：**判断是否为 ASCII 小写字母，即 a 到 z**
- boolean isAsciiAlphaUpper(char ch)：**判断是否为 ASCII 大写字母，即 A 到 Z**
- boolean isAsciiAlphanumeric(char ch)：**判断是否为 ASCII 字符数字，**(ch >= 'A'&& cn <= 'Z') | (ch >= 'a'&& ch <= 'z') || (ch >= '0'&& ch <= '9')
- boolean isAsciiNumeric(char ch)：**判断是否为数字**
- Boolean isAsciiControl(char ch)：**判断是否为控制字符，**ch < 32 || ch = 127
- Boolean isAsciiPrintable(char ch)：**判断是否可打印出得 ASCII 字符，**ch >= 32 && ch < 127
- int toIntValue(char ch)：**数字转换，**ch – 48
- String unicodeEscaped(char ch)：**将 ch 转换为 unicode 表示的字符串形式**

BooleanUtils

- negate(Boolean bool)：否定指定的 boolean 值
- isTrue(Boolean bool)：检查一个 boolean 值是否为 true，如果参数为 null，返回 false
- isNotTrue(Boolean bool)：检查一个 boolean 值是否为 false，如果参数为 null，返回 true
- isFalse(Boolean bool)：检查一个 boolean 值是否为 false，如果是返回 true，如果检查的值为 true 或 null 返回 false.
- isNotFalse(Boolean bool)：检查一个 boolean 值是否不为 false，如果是返回 true
- toBoolean(Boolean bool)：转换一个为 null 的 boolean 值，返回一个 false.
- toBooleanDefaultIfNull(Boolean bool, boolean valueIfNull)：转换一个为 null 的 boolean 值返回后面参数给定的 boolean 值.
- toBoolean(int value)：当参数为 0 是返回 false，其它都返回 true.
- toBooleanObject(int value)：当参数为 0 是返回 Boolean.FALSE 对象，其它都返回 Boolean.TUE.
- toBooleanObject(Integer value)：当参数为 0 是返回 Boolean.FALSE 对象，为 null 返回 null
- toBoolean(int value, int trueValue, int falseValue)：如果第一个参数和第二个参数相等返回 true

ExceptionUtils

- 对异常的常见操作，获得堆栈，异常抛出方法名，错误链中对象数
- Throwable getCauseUsingMethodName(Throwable throwable, String methodName)` 获取导致 Throwable 的 Throwable，可以设置自己制定的方法名称
- Throwable getRootCause(Throwable throwable)`：获取导致 Throwable 的 Root Throwable
- int getThrowableCount(Throwable throwable)`：统计异常链上的 Throwable 对象数量