Apache的Commons Lang和BeanUtils的使用

Apache Commons包估计是Java中使用最广发的工具包了，很多框架都依赖于这组工具包中的一部分，它提供了我们常用的一些编程需要，但是JDK没能提供的机能，最大化的减少重复代码的编写。

# Apache Common Lang的使用

Commons Lang是对JDK中java.lang包的补充，提供了各种各样的Utilities工具类，这里说说最常用的几个工具类。 maven依赖如下：

|  |
| --- |
| <dependency>  <groupId>org.apache.commons</groupId>  <artifactId>commons-lang3</artifactId>  <version>3.4</version>  </dependency> |

## 字符串的空判断

|  |
| --- |
| //isEmpty ：null和空字符串为真  System.out.println(StringUtils.isEmpty(null)); // true  System.out.println(StringUtils.isEmpty("")); // true  System.out.println(StringUtils.isEmpty(" ")); // false  System.out.println(StringUtils.isEmpty("bob")); // false  System.out.println(StringUtils.isEmpty(" bob ")); // false |
| //isBlank ：null和空字符串和空白字符为真  System.out.println(StringUtils.isBlank(null)); // true  System.out.println(StringUtils.isBlank("")); // true  System.out.println(StringUtils.isBlank(" ")); // true  System.out.println(StringUtils.isBlank("bob")); // false  System.out.println(StringUtils.isBlank(" bob ")); // false |

## 字符串的Trim

|  |
| --- |
| //trim  System.out.println(StringUtils.trim(null)); // null  System.out.println(StringUtils.trim("")); // ""  System.out.println(StringUtils.trim(" ")); // ""  System.out.println(StringUtils.trim("abc")); // "abc"  System.out.println(StringUtils.trim(" abc")); // "abc"  System.out.println(StringUtils.trim(" abc ")); // "abc"  System.out.println(StringUtils.trim(" ab c ")); // "ab c" |
| System.out.println(StringUtils.strip(null)); // null  System.out.println(StringUtils.strip("")); // ""  System.out.println(StringUtils.strip(" ")); // ""  System.out.println(StringUtils.strip("abc")); // "abc"  System.out.println(StringUtils.strip(" abc")); // "abc"  System.out.println(StringUtils.strip("abc ")); // "abc"  System.out.println(StringUtils.strip(" abc ")); // "abc"  System.out.println(StringUtils.strip(" ab c ")); // "ab c"    System.out.println(StringUtils.strip(" abcyx", "xyz")); // " abc"    System.out.println(StringUtils.stripStart("yxabcxyz ", "xyz")); // "abcxyz "  System.out.println(StringUtils.stripEnd(" xyzabcyx", "xyz")); // " xyzabc" |

## 字符串的分割

|  |
| --- |
| //默认半角空格分割  String str1 = "aaa bbb ccc";  String[] dim1 = StringUtils.split(str1); // => ["aaa", "bbb", "ccc"]    System.out.println(dim1.length);//3  System.out.println(dim1[0]);//"aaa"  System.out.println(dim1[1]);//"bbb"  System.out.println(dim1[2]);//"ccc"    //指定分隔符  String str2 = "aaa,bbb,ccc";  String[] dim2 = StringUtils.split(str2, ","); // => ["aaa", "bbb", "ccc"]    System.out.println(dim2.length);//3  System.out.println(dim2[0]);//"aaa"  System.out.println(dim2[1]);//"bbb"  System.out.println(dim2[2]);//"ccc"    //去除空字符串  String str3 = "aaa,,bbb";  String[] dim3 = StringUtils.split(str3, ","); // => ["aaa", "bbb"]    System.out.println(dim3.length);//2  System.out.println(dim3[0]);//"aaa"  System.out.println(dim3[1]);//"bbb"    //包含空字符串  String str4 = "aaa,,bbb";  String[] dim4 = StringUtils.splitPreserveAllTokens(str4, ","); // => ["aaa", "", "bbb"]    System.out.println(dim4.length);//3  System.out.println(dim4[0]);//"aaa"  System.out.println(dim4[1]);//""  System.out.println(dim4[2]);//"bbb"    //指定分割后数组个数最大数（超过后不分割）  String str5 = "aaa,bbb,ccc";  String[] dim5 = StringUtils.split(str5, ",", 2); // => ["aaa", "bbb,ccc"]    System.out.println(dim5.length);//2  System.out.println(dim5[0]);//"aaa"  System.out.println(dim5[1]);//"bbb,ccc" |

## 字符串的连接

|  |
| --- |
| //数组元素拼接  String[] array = {"aaa", "bbb", "ccc"};  String result1 = StringUtils.join(array, ",");    System.out.println(result1);//"aaa,bbb,ccc"    //集合元素拼接  List<String> list = new ArrayList<String>();  list.add("aaa");  list.add("bbb");  list.add("ccc");  String result2 = StringUtils.join(list, ",");    System.out.println(result2);//"aaa,bbb,ccc" |

## 字符串的Escape

|  |
| --- |
| //特殊字符转移  System.out.println(StringEscapeUtils.escapeCsv("测试测试哦"));//"测试测试哦"  System.out.println(StringEscapeUtils.escapeCsv("测试,测试哦"));//"\"测试,测试哦\""  System.out.println(StringEscapeUtils.escapeCsv("测试\n测试哦"));//"\"测试\n测试哦\""    System.out.println(StringEscapeUtils.escapeHtml4("测试测试哦  "));//"<p>测试测试哦</p>"  System.out.println(StringEscapeUtils.escapeJava("\"rensaninng\"，欢迎您！"));//"\"rensaninng\"\uFF0C\u6B22\u8FCE\u60A8\uFF01"    System.out.println(StringEscapeUtils.escapeEcmaScript("测试'测试哦"));//"\u6D4B\u8BD5\'\u6D4B\u8BD5\u54E6" |

## 随机数

|  |
| --- |
| // 10位英字  System.out.println(RandomStringUtils.randomAlphabetic(10));    // 10位英数  System.out.println(RandomStringUtils.randomAlphanumeric(10));    // 10位ASCII码  System.out.println(RandomStringUtils.randomAscii(10));    // 指定文字10位  System.out.println(RandomStringUtils.random(10, "abcde"));  //随机的六位整数  System.out.println(RandomStringUtils.randomNumeric(6)); |

## 数组

|  |
| --- |
| // 追加元素到数组尾部  int[] array1 = {1, 2};  array1 = ArrayUtils.add(array1, 3); // => [1, 2, 3]    System.out.println(array1.length);//3  System.out.println(array1[2]);//3    // 删除指定位置的元素  int[] array2 = {1, 2, 3};  array2 = ArrayUtils.remove(array2, 2); // => [1, 2]    System.out.println(array2.length);//2    // 截取部分元素  int[] array3 = {1, 2, 3, 4};  array3 = ArrayUtils.subarray(array3, 1, 3); // => [2, 3]    System.out.println(array3.length);//2    // 数组拷贝  String[] array4 = {"aaa", "bbb", "ccc"};  String[] copied = (String[]) ArrayUtils.clone(array4); // => {"aaa", "bbb", "ccc"}    System.out.println(copied.length);//3    // 判断是否包含某元素  String[] array5 = {"aaa", "bbb", "ccc", "bbb"};  boolean result1 = ArrayUtils.contains(array5, "bbb"); // => true  System.out.println(result1);//true    // 判断某元素在数组中出现的位置（从前往后，没有返回-1）  int result2 = ArrayUtils.indexOf(array5, "bbb"); // => 1  System.out.println(result2);//1    // 判断某元素在数组中出现的位置（从后往前，没有返回-1）  int result3 = ArrayUtils.lastIndexOf(array5, "bbb"); // => 3  System.out.println(result3);//3    // 数组转Map  Map<Object, Object> map = ArrayUtils.toMap(new String[][]{  {"key1", "value1"},  {"key2", "value2"}  });  System.out.println(map.get("key1"));//"value1"  System.out.println(map.get("key2"));//"value2"    // 判断数组是否为空  Object[] array61 = new Object[0];  Object[] array62 = null;  Object[] array63 = new Object[]{"aaa"};    System.out.println(ArrayUtils.isEmpty(array61));//true  System.out.println(ArrayUtils.isEmpty(array62));//true  System.out.println(ArrayUtils.isNotEmpty(array63));//true    // 判断数组长度是否相等  Object[] array71 = new Object[]{"aa", "bb", "cc"};  Object[] array72 = new Object[]{"dd", "ee", "ff"};    System.out.println(ArrayUtils.isSameLength(array71, array72));//true    // 判断数组元素内容是否相等  Object[] array81 = new Object[]{"aa", "bb", "cc"};  Object[] array82 = new Object[]{"aa", "bb", "cc"};    System.out.println(ArrayUtils.isEquals(array81, array82));    // Integer[] 转化为 int[]  Integer[] array9 = new Integer[]{1, 2};  int[] result = ArrayUtils.toPrimitive(array9);    System.out.println(result.length);//2  System.out.println(result[0]);//1    // int[] 转化为 Integer[]  int[] array10 = new int[]{1, 2};  Integer[] result10 = ArrayUtils.toObject(array10);    System.out.println(result.length);//2  System.out.println(result10[0].intValue());//1 |

## 日期

|  |
| --- |
| // 生成Date对象  Date date = DateUtils.parseDate("2010/01/01 11:22:33", new String[]{"yyyy/MM/dd HH:mm:ss"});    // 10天后  Date tenDaysAfter = DateUtils.addDays(date, 10); // => 2010/01/11 11:22:33  System.out.println(DateFormatUtils.format(tenDaysAfter, "yyyy/MM/dd HH:mm:ss"));    // 前一个月  Date prevMonth = DateUtils.addMonths(date, -1); // => 2009/12/01 11:22:33  System.out.println(DateFormatUtils.format(prevMonth, "yyyy/MM/dd HH:mm:ss"));    // 判断是否是同一天  Date date1 = DateUtils.parseDate("2010/01/01 11:22:33", new String[]{"yyyy/MM/dd HH:mm:ss"});  Date date2 = DateUtils.parseDate("2010/01/01 22:33:44", new String[]{"yyyy/MM/dd HH:mm:ss"});  System.out.println(DateUtils.isSameDay(date1, date2));// true    // 日期格式化  System.out.println(DateFormatUtils.format(new Date(), "yyyy/MM/dd HH:mm:ss")); |

# Commons Beanutils的使用

Commons Beanutils是一组专门用于操作Bean的工具类，目前很多流行的框架基本都离不开他。 maven依赖如下：

|  |
| --- |
| <dependency>  <groupId>commons-beanutils</groupId>  <artifactId>commons-beanutils</artifactId>  <version>1.9.2</version>  </dependency> |

## 获取字段值

|  |
| --- |
| SampleBean bean1 = new SampleBean();  bean1.setName("rensanning");  bean1.setAge(31);    String name = BeanUtils.getProperty(bean1, "name");  String age = BeanUtils.getProperty(bean1, "age");    System.out.println(name);  System.out.println(age); |

## 设置字段值

|  |
| --- |
| SampleBean bean2 = new SampleBean();  BeanUtils.setProperty(bean2, "name", "rensanning");  BeanUtils.setProperty(bean2, "age", 31);    System.out.println(bean2.getName());  System.out.println(bean2.getAge()); |

## 赋值Bean

|  |
| --- |
| SampleBean bean3 = new SampleBean();  bean3.setName("rensanning");  bean3.setAge(31);    SampleBean clone = (SampleBean) BeanUtils.cloneBean(bean3);    System.out.println(clone.getName());  System.out.println(clone.getAge()); |

## Bean的describe

|  |
| --- |
| SampleBean bean4 = new SampleBean();  bean4.setName("rensanning");  bean4.setAge(31);    @SuppressWarnings("unchecked")  Map<String, String> map4 = BeanUtils.describe(bean4);    System.out.println(map4.get("name"));  System.out.println(map4.get("age")); |

## Bean的populate

|  |
| --- |
| SampleBean bean5 = new SampleBean();    Map<String, String> map5 = new HashMap<String, String>();  map5.put("name", "rensanning");  map5.put("age", "31");    BeanUtils.populate(bean5, map5);    System.out.println(bean5.getName());  System.out.println(bean5.getAge()); |

## 获取Bean的数组集合字段值

|  |
| --- |
| SampleBean bean6 = new SampleBean();  bean6.setArray(new String[]{"a", "b", "c"});  List<String> list0 = new ArrayList<String>();  list0.add("d");  list0.add("e");  list0.add("f");  bean6.setList(list0);    String[] array = BeanUtils.getArrayProperty(bean6, "array");    System.out.println(array.length);//3  System.out.println(array[0]);//"a"  System.out.println(array[1]);//"b"  System.out.println(array[2]);//"c"    String[] list = BeanUtils.getArrayProperty(bean6, "list");  System.out.println(list.length);//3  System.out.println(list[0]);//"d"  System.out.println(list[1]);//"e"  System.out.println(list[2]);//"f"    System.out.println(BeanUtils.getProperty(bean6, "array[1]"));//"b"  System.out.println(BeanUtils.getIndexedProperty(bean6, "array", 2));//"c" |

## 获取Bean的Map字段值

|  |
| --- |
| SampleBean bean7 = new SampleBean();  Map<String, String> map = new HashMap<String, String>();  map.put("key1", "value1");  map.put("key2", "value2");  bean7.setMap(map);    String value1 = BeanUtils.getMappedProperty(bean7, "map", "key1");  System.out.println(value1);//"value1"    String value2 = BeanUtils.getMappedProperty(bean7, "map", "key2");  System.out.println(value2);//"value2"    System.out.println(BeanUtils.getProperty(bean7, "map.key1"));//"value1"  System.out.println(BeanUtils.getProperty(bean7, "map.key2"));//"value2" |

## 获取Bean的嵌套字段值

|  |
| --- |
| SampleBean bean = new SampleBean();  NestedBean nestedBean = new NestedBean();  nestedBean.setNestedProperty("xxx");  bean.setNestedBean(nestedBean);    String value = BeanUtils.getNestedProperty(bean, "nestedBean.nestedProperty");  System.out.println(value);//"xxx"    System.out.println(BeanUtils.getProperty(bean, "nestedBean.nestedProperty"));//"xxx" |

## URL字段的特殊处理

|  |
| --- |
| SampleBean bean8 = new SampleBean();    BeanUtils.setProperty(bean8, "url", "http://www.google.com/");    URL url = bean8.getUrl();  System.out.println(url.getProtocol());//"http"  System.out.println(url.getHost());//"www.google.com"  System.out.println(url.getPath());//"/" |

## 日期的转化

|  |
| --- |
| SampleBean bean9 = new SampleBean();    DateConverter converter = new DateConverter();  converter.setPattern("yyyy/MM/dd HH:mm:ss");    ConvertUtils.register(converter, Date.class);  ConvertUtils.register(converter, String.class);    BeanUtils.setProperty(bean9, "date", "2010/12/19 23:40:00");    String value9 = BeanUtils.getProperty(bean9, "date");  System.out.println(value9);//"2010/12/19 23:40:00" |