org.apache.commons包介绍

# org.apache.commons包

## 功能介绍

**Apache Commons**是一个非常有用的工具包，解决各种实际的通用问题。

官网：<http://commons.apache.org/index.html>

## Maven添加依赖或下载jar包

<http://search.maven.org/> 搜索 org.apache.commons

<http://mvnrepository.com/>

Maven依赖添加：

org.apache.commons是一个**groupId名字**，其包括很多artifactId包，其中commons-lang3就是其中一个，提供了很多工具类，如StringUtils就是一个用于字符串的工具类。

org.apache.commons这个groupId中包括了很多artifactId，如commons-langs、commons-collections4、commons-compress、commons-math3、commons-io等。

## 列表简单介绍

**BeanUtils** Commons-BeanUtils 提供对 Java 反射和自省API的包装；

**Betwixt** Betwixt提供将 JavaBean 映射至 XML 文档，以及相反映射的服务；

**Chain** Chain 提供实现组织复杂的处理流程的“责任链模式”；

**CLI** CLI 提供针对命令行参数，选项，选项组，强制选项等的简单API;

**Codec** Codec 包含一些通用的编码解码算法。包括一些语音编码器， Hex, Base64, 以及URL encoder.

**Collections** Commons-Collections 提供一个类包来扩展和增加标准的 Java Collection框架

**Configuration** Commons-Configuration 工具对各种各式的配置和参考文件提供读取帮助.

**Daemon** 一种 unix-daemon-like java 代码的替代机制

**DBCP** Commons-DBCP 提供数据库连接池服务

**DbUtils** DbUtils 是一个 JDBC helper 类库，完成数据库任务的简单的资源清除代码.

**Digester** Commons-Digester 是一个 XML-Java对象的映射工具，用于解析 XML配置文件.

**Discovery** Commons-Discovery 提供工具来定位资源 (包括类) ，通过使用各种模式来映射服务/引用名称和资源名称。.

**EL** Commons-EL 提供在JSP2.0规范中定义的EL表达式的解释器.

**FileUpload** FileUpload 使得在你可以在应用和Servlet中容易的加入强大和高性能的文件上传能力

**HttpClient** Commons-HttpClient 提供了可以工作于HTTP协议客户端的一个框架.

**IO** IO 是一个 I/O 工具集

**Jelly** Jelly是一个基于 XML 的脚本和处理引擎。 Jelly 借鉴了 JSP 定指标签，Velocity, Cocoon和Xdoclet中的脚本引擎的许多优点。Jelly 可以用在命令行， Ant 或者 Servlet之中。

**Jexl**Jexl是一个表达式语言，通过借鉴来自于Velocity的经验扩展了JSTL定义的表达式语言。.

**JXPath** Commons-JXPath 提供了使用Xpath语法操纵符合Java类命名规范的 JavaBeans的工具。也支持 maps, DOM 和其他对象模型。.

**Lang** Commons-Lang 提供了许多许多通用的工具类集，提供了一些java.lang中类的扩展功能;

**Latka** Commons-Latka 是一个HTTP 功能测试包，用于自动化的QA,验收和衰减测试.

**Launcher** Launcher 组件是一个交叉平台的Java 应用载入器。Commons-launcher 消除了需要批处理或者Shell脚本来载入Java 类。.原始的 Java 类来自于Jakarta Tomcat 4.0 项目

**Logging** Commons-Logging 是一个各种 logging API实现的包裹类.

**Math** Math 是一个轻量的，自包含的数学和统计组件，解决了许多非常通用但没有及时出现在Java标准语言中的实践问题.

**Modeler** Commons-Modeler 提供了建模兼容JMX规范的Mbean的机制.

**Net** Net 是一个网络工具集，基于 NetComponents 代码，包括 FTP 客户端等等。

**Pool** Commons-Pool 提供了通用对象池接口，一个用于创建模块化对象池的工具包，以及通常的对象池实现.

**Primitives** Commons-Primitives提供了一个更小，更快和更易使用的对Java基本类型的支持。当前主要是针对基本类型的 collection。.

**Validator** The commons-validator提供了一个简单的，可扩展的框架来在一个XML文件中定义校验器 (校验方法)和校验规则。支持校验规则的和错误消息的国际化。

# Commons Lang包 commons-lang3包

## 官网及下载依赖

Home Page: <http://commons.apache.org/proper/commons-lang/>

Download: <http://commons.apache.org/proper/commons-lang/download_lang.cgi>

或者：jar包： <http://search.maven.org/> 搜索commons-lang3

**doc文档**下载：

Maven :

<!-- <https://mvnrepository.com/artifact/org.apache.commons/commons-lang3> -->

<dependency>

<groupId>**org.apache.commons**</groupId>

<artifactId>**commons-lang3**</artifactId>

<version>3.7</version>

</dependency>

## 功能介绍

**Apache Commons Lang**, a package of Java utility classes for the classes that are in java.lang's hierarchy, or are considered to be so standard **as to justify existence in java.lang**.

跟**java.lang**这个包的作用类似，Commons Lang这一组API也是**提供一些基础的、通用的操作和处理**，如自动生成toString()的结果、自动实现hashCode()和equals()方法、数组操作、枚举、日期和时间的处理等等。

lang包主要是一些可以高度重用的Util类；lang.builder包包含了一组用于产生每个Java类中都常使用到的toString()、hashCode()、equals()、compareTo()等等方法的构造器；lang.enums包顾名思义用于处理枚举；lang.exception包用于处理Java标准API中的exception，为1.4之前版本提供Nested Exception功能；lang.math包用于处理数字；lang.mutable用于包装值型变量；lang.time包提供处理日期和时间的功能。

Commons Proper **is dedicated to** one principal goal: creating and maintaining reusable Java components. The Commons Proper is a place for collaboration and sharing, where developers from throughout the Apache community can work together on projects to be shared by the Apache projects and Apache users.

Commons developers will make an effort to ensure that their components have minimal dependencies on other libraries, so that these components can be deployed easily. In addition, Commons components will keep their interfaces as stable as possible, so that Apache users (including other Apache projects) can implement these components without having to worry about changes in the future.

This article gives an overview of (some of) the components which can be found here.

## 常用的类

首先来看org.apache.commons.lang包，这个包提供了一些有用的包含static方法的Util类。除了6个Exception类和2个已经deprecated的数字类之外，commons.lang包共包含了17个实用的类：

**ArrayUtils** – 用于对数组的操作，如添加、查找、删除、子数组、倒序、元素类型转换等；

BitField – 用于操作位元，提供了一些方便而安全的方法；

BooleanUtils – 用于操作和转换 boolean 或者 Boolean 及相应的数组；

CharEncoding – 包含了 Java 环境支持的字符编码，提供是否支持某种编码的判断；

CharRange – 用于设定字符范围并做相应检查；

CharSet – 用于设定一组字符作为范围并做相应检查；

CharSetUtils – 用于操作 CharSet ；

CharUtils – 用于操作 char 值和 Character 对象；

ClassUtils – 用于对 Java 类的操作，不使用反射；

ObjectUtils – 用于操作 Java 对象，提供 null 安全的访问和其他一些功能；

RandomStringUtils – 用于生成随机的字符串；

SerializationUtils – 用于处理对象序列化，提供比一般 Java 序列化更高级的处理能力；

StringEscapeUtils – 用于正确处理转义字符，产生正确的 Java 、 JavaScript 、 HTML 、 XML 和 SQL 代码；

**StringUtils** – 处理 String 的核心类，提供了相当多的功能；

SystemUtils – 在 java.lang.System 基础上提供更方便的访问，如用户路径、 Java 版本、时区、操作系统等判断；

Validate – 提供验证的操作，有点类似 assert 断言；

WordUtils – 用于处理单词大小写、换行等。

接下来我准备用两个例子来分别说明ArrayUtils和StringUtils的常见用法。

# ArrayUtils类

数组是我们经常需要使用到的一种数据结构，但是由于Java本身并没有提供很好的API支持，使得很多操作实际上做起来相当繁琐，以至于我们实际编码中甚至会不惜牺牲性能去使用Collections API，用Collection当然能够很方便的解决我们的问题，但是我们一定要以性能为代价吗？ArrayUtils帮我们解决了处理类似情况的大部分问题。

可以利用ArrayUtils类帮我们完成**数组的打印、查找、克隆、倒序、以及值型/对象数组之间的转换等操作**。如果想了解更多，请参考Javadoc。

# StringUtils工具类

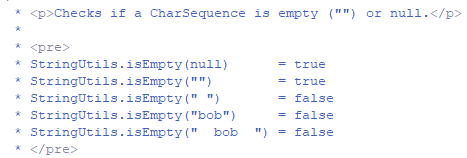
处理文本对Java应用来说应该算是家常便饭了，在1.4出现之前，Java自身提供的API非常有限，如String、StringTokenizer、StringBuffer，操作也比较单一。无非就是查找substring、分解、合并等等。到1.4的出现可以说Java的文字处理上了一个台阶，因为它支持regular expression了。这可是个重量级而方便的东东啊，缺点是太复杂，学习起来有一定难度。相较而言，Jakarta Commons提供的StringUtils和WordUtils至今还维持着那种简洁而强大的美，使用起来也很顺手。

大致了解到这个StringUtils类简单而强大的处理能力，从检查空串（对null的情况处理很得体），到分割子串，到生成格式化的字符串，使用都很简洁，也很直截了当。

**org.apache.commons.lang.StringUtils中方法的操作对象是java.lang.String类型的对象**，是JDK提供的String类型操作方法的补充，并且是**null安全的**(即如果输入参数String为null则不会抛出NullPointerException，而是做了相应处理，例如，如果输入为null则返回也是null等，具体可以查看源代码)。

除了构造器，**StringUtils中一共有130多个方法，并且都是static的**，所以可以这样调用**StringUtils.xxx()**。

## isEmpty方法

public static boolean isEmpty(final CharSequence cs)

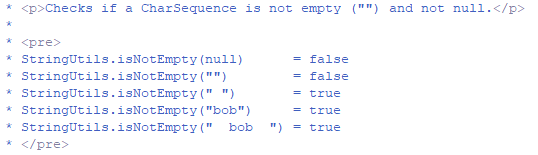
**public** **static** **boolean** isEmpty(**final** CharSequence cs) {

**return** **cs == null || cs.length() == 0;**

}

## isNotEmpty:与isEmpty相反

!isEmpty(cs);



**public** **static** **boolean** isNotEmpty(**final** CharSequence cs) {

**return** **!*isEmpty*(cs);**

}

## isBlank等等

## 示例

判断字符串是否为空,可以使用StringUtils.isEmpty(str)，避免了空指针异常。

因为字符串对象有可能为null，出现空指针异常，所以需要判断null != str && str.length != 0