Atitit prgrmlan topic--express lan QL query lan表达式语言

目录

[1. 通用表达语言（CEL） 1](#_Toc20932)

[1.1. 8.2 功能概述 2](#_Toc22894)

[2. SpEL支持如下表达式： 3](#_Toc25682)

[2.1. 基本表达式：字面量表达式、关系，逻辑与算数运算表达式、 3](#_Toc20780)

[2.2. 类相关表达式：方法调用 变量定义及引用 3](#_Toc18556)

[2.3. 集合相关表达式： 3](#_Toc13737)

[2.4. 四、其他表达式：模板表达式。 3](#_Toc3769)

[3. Ongl （调用java等扩展，集合投影选择等） 4](#_Toc5107)

[3.1. Spel （调用java等扩展，集合投影选择等） 4](#_Toc15474)

[3.2. 正则表达式 re语言 5](#_Toc7709)

[4. Other 5](#_Toc7316)

[4.1. 表达式语言（Expression Language），或称EL表达式， 5](#_Toc22523)

[4.2. JUEL是统一表达语言（EL）的开源实现 6](#_Toc24017)

[4.3. Mvel （不常用 6](#_Toc20189)

[4.4. Java表达式语言（JEXL） 6](#_Toc13308)

[4.5. Commons EL项目。 apche 7](#_Toc23560)

[4.6. HTML 模板语言(HTL) 由 Adobe Experience Manager (AEM) 提供支持， 7](#_Toc7450)

[5. ref 7](#_Toc12278)

# 通用表达语言（CEL）

ExpressionLanguage Component

The ExpressionLanguage component provides an engine that can compile and evaluate expressions. An expression is a one-liner that returns a value (mostly, but not limited to, Booleans).

通用表达语言（CEL）是非图灵完整语言，旨在实现简单性，速度，安全性和可移植性。

## **8.2 功能概述**

表达式语言支持以下功能

* 文字表达式
* 布尔和关系运算符
* 正则表达式
* 类表达式
* 访问 properties, arrays, lists, maps
* 方法调用
* 关系运算符
* 参数
* 调用构造函数
* Bean引用
* 构造Array
* 内嵌lists
* 内嵌maps
* 三元运算符
* 变量
* 用户定义的函数
* 集合投影
* 集合筛选
* 模板表达式

　表达式语言一般是用最简单的形式完成最主要的工作，减少我们的工作量。

# SpEL支持如下表达式：

## ****基本表达式：****字面量表达式、关系，逻辑与算数运算表达式、

1. 字符串连接及截取表达式、三目运算及Elivis表达式、正则表达式、括号优先级表达式；

## ****类相关表达式：****方法调用 变量定义及引用

1. 类类型表达式、类实例化、instanceof表达式、变量定义及引用、赋值表达式、自定义函数、对象属性存取及安全导航表达式、对象方法调用、Bean引用；

## ****集合相关表达式：****

1. 内联List、内联数组、集合，字典访问、列表，字典，数组修改、集合投影、集合选择；不支持多维内联数组初始化；不支持内联字典定义；

## ****四、其他表达式****：模板表达式。字符串模板 变量插入

****注：SpEL表达式中的关键字是不区分大小写的。****

# Ongl （调用java等扩展，集合投影选择等）

* [OGNL](https://en.wikipedia.org/wiki/OGNL" \o "OGNL) – [WebWork](https://en.wikipedia.org/wiki/WebWork" \o "网络工作)（和[Struts](https://en.wikipedia.org/wiki/Apache_Struts" \o "Apache Struts) 2）使用的开源EL 。
* [MVEL](https://en.wikipedia.org/wiki/MVEL" \o "MVEL) –在许多基于Java的项目中使用的开源EL。
* [SpEL](http://docs.spring.io/spring/docs/current/spring-framework-reference/html/expressions.html) – Spring Expression Language，是[Spring Framework的](https://en.wikipedia.org/wiki/Spring_Framework" \o "春季框架)一部分的开源EL 。它主要用于Spring投资组合项目，但由于它与技术无关，因此可能会用于其他项目。
* [Ant-Flaka-](http://ant-flaka.bitio.org/)旨在借助EL 简化[Ant](https://en.wikipedia.org/wiki/Apache_Ant" \o "阿帕奇蚂蚁)构建脚本。
* [CEL](https://en.wikipedia.org/w/index.php?title=Common_Expression_Language&action=edit&redlink=1" \o "通用表达语言（页面不存在）) –由Google开发的开源EL。

## Spel （调用java等扩展，集合投影选择等）

Spring表达式语言（简称SpEl）是一个支持查询和操作运行时对象导航图功能的强大的表达式语言. 它的语法类似于传统EL，但提供额外的功能，最出色的就是函数调用和简单字符串的模板函数。

尽管有其他可选的 Java 表达式语言，如 OGNL, MVEL,JBoss EL 等等，但 Spel 创建的初衷是了给 Spring 社区提供一种简单而高效的表达式语言，一种可贯穿整个 Spring 产品组的语言。这种语言的特性应基于 Spring 产品的需求而设计。

虽然SpEL引擎作为Spring 组合里的表达式解析的基础 ，但它不直接依赖于Spring,可独立使用。为了整合，许多在本章使用SpEL例子就好像它是一个独立的表达式语言。这就需要创建一些引导 如解析器这样的基础构造类。大多数Spring用户将不再需要处理这些基础构建，而是仅将作者表达的字符串进行解析。一个传统的使用例子是集成SpEL去创建XML或者定义Bean的注解,可以选择这里看到 [表达式支持定义b](http://itmyhome.com/spring/expressions.html" \l "expressions-beandef" \o "8.4 定义bean的beandef表达支持)

## 正则表达式 re语言

# Other

## **表达式语言**（Expression Language），或称**EL表达式**，

简称**EL**，是[Java](https://zh.wikipedia.org/wiki/Java" \o "Java)中的一种特殊的通用编程语言，借鉴于[JavaScript](https://zh.wikipedia.org/wiki/JavaScript" \o "JavaScript)和[XPath](https://zh.wikipedia.org/wiki/XPath" \o "XPath)。主要作用是在Java Web应用程序嵌入到网页（如[JSP](https://zh.wikipedia.org/wiki/JSP" \o "JSP)）中，用以访问页面的上下文以及不同作用域中的[对象](https://zh.wikipedia.org/wiki/%E5%AF%B9%E8%B1%A1_(%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A7%91%E5%AD%A6)" \o "对象 (计算机科学))，取得对象属性的值，或执行简单的运算或判断操作。EL在得到某个[数据](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE" \o "数据)时，会自动进行[数据类型](https://zh.wikipedia.org/wiki/%E6%95%B8%E6%93%9A%E9%A1%9E%E5%9E%8B" \o "数据类型)的转换。

Package javax.el

Provides the API for **Jakarta Expression Language 3.0**

See: [Description](https://jakarta.ee/specifications/expression-language/3.0/apidocs/javax/el/package-summary.html" \l "package.description)

## [JUEL](http://juel.sourceforge.net/)是统一表达语言（EL）[的开源](https://en.wikipedia.org/wiki/Open-source_software" \o "开源软件)实现

* [Java Expression Language（JEXL）](http://commons.apache.org/proper/commons-jexl/)是一个旨在促进用Java编写的应用程序和框架中动态和脚本功能的实现的库。最新版本，版本：3.1,2017年4月14日。
* [JUEL](http://juel.sourceforge.net/)是统一表达语言（EL）[的开源](https://en.wikipedia.org/wiki/Open-source_software" \o "开源软件)实现，被指定为JSP 2.1标准（JSR-245）的一部分。它被认为是稳定且功能完善的，并根据[Apache License 2.0](https://en.wikipedia.org/wiki/Apache_License" \o "Apache许可证)获得[许可](https://en.wikipedia.org/wiki/Apache_License" \o "Apache许可证)。JUEL还适用于非JSP应用程序。最新版本2.2.7，2014年2月6日。
* [Apache Commons EL](https://commons.apache.org/dormant/commons-el/)是[Apache](https://commons.apache.org/dormant/commons-el/)的JSP 2.0 EL解释器。最新版本，版本1.0，2003年6月20日。指向源和二进制文件的下载链接已断开。

## Mvel （不常用

## **Java表达式语言（JEXL）**

JEXL是一个旨在促进在用Java编写的应用程序和框架中实现动态和脚本功能的库。

JEXL基于对JSTL表达式语言的一些扩展来实现一种表达式语言，该扩展支持在shell脚本或ECMAScript中看到的大多数构造。  
它的目标是公开技术人员或与企业平台合作的顾问可以使用的脚本功能。

该库公开了一个占用空间小的API- [核心功能](https://commons.apache.org/proper/commons-jexl/apidocs/org/apache/commons/jexl3/package-summary.html" \l "usage)适合3类和10种方法-可在各种条件下使用：

* 脚本功能：
  + 您的应用程序允许（高级）用户评估或定义一些简单的表达式，例如计算公式。
* 模块或组件配置：
  + 您的应用程序具有最终用户模块使用的配置文件（最终由设计模块生成），这些配置文件将从变量和表达式中受益。
  + 使用IOC方便但总体复杂性不需要（或不能依赖）成熟的库（Spring，Guice ...）时。
* 接口和实现的松散耦合或鸭式键入：
  + 您有一些可选类，您的代码不能将其视为编译依赖项。
  + 您必须集成并调用“旧版”代码，或者使用您不想强烈依赖的组件。
* 简单的模板功能：
  + 您的应用程序具有基本模板要求，并且JSP或Velocity可能会过大或太不方便部署。

JEXL名称代表Java EXpression Language，这是一种简单的表达语言，最初受Apache Velocity和JavaServer Pages标准标记库版本1.1（JSTL）和JavaServer Pages 2.0（JSP）中定义的表达语言的启发。JEXL 2.0增加了受[Unified EL](https://en.wikipedia.org/wiki/Unified_Expression_Language)启发的功能 。现在，语法接近于ECMAScript和“ shell脚本”的混合体，使技术人员或顾问可以轻松掌握。尽管暴露的对象及其行为显然需要记录在案...

API和表达式语言通过内省来利用Java Bean命名模式，以公开属性获取器和设置器。它还将公共类字段视为属性，并允许调用任何可访问的方法。

JEXL试图向Velocity社区学习一些关于表达语言的经验，以吸引更多的读者。 [Commons Jelly](http://commons.apache.org/jelly)需要使用Velocity的方法访问权限，它只需要拥有它即可。

必须注意，JEXL **不是** JSTL 1.1（JSR-052）或JSP 2.0（JSR-152）中定义的EL的兼容实现。有关这些规范的兼容实现，请参见[Commons EL](http://commons.apache.org/el)项目。

## [Commons EL](http://commons.apache.org/el)项目。 apche

**概述**

## HTML 模板语言(HTL) 由 Adobe Experience Manager (AEM) 提供支持，

其目的是提供可增加安全性的高效企业级 Web 框架，并让不了解 Java 的 HTML 开发者更好地参与 AEM 项目。

AEM 6.0 中引入了 HTML 模板语言，它取代了 JSP (JavaServer Pages)，成为 HTML 的首选和推荐的服务器端模板系统。 对于需要构建强大企业网站的 Web 开发者来说，HTML 模板语言有助于提高安全性和开发效率

# ref

Atitit QL查询语言总结

Lucene表达式语言 - elasticsearch中文文档.html

8. Spring 表达式语言 (SpEL).html