Atitit 动态调用java 扩展

目录

[1.1. Ognl只能调用静态 1](#_Toc12716)

[1.2. Spel不错可以，调用动态的 1](#_Toc357)

## Ognl只能调用静态

## Spel不错可以，调用动态的

D:\0wkspc\platform-top-run-coffey\platform-top-service\platform-top-service-finance\src\main\java\org\chwin\firefighting\apiserver\dsl\spelUtil.java

package org.chwin.firefighting.apiserver.dsl;  
  
import org.springframework.expression.Expression;  
import org.springframework.expression.ExpressionParser;  
import org.springframework.expression.spel.standard.SpelExpressionParser;  
  
import javax.sound.midi.Soundbank;  
  
public class spelUtil {  
  
 public static void main(String[] args) {  
 ExpressionParser parser = new SpelExpressionParser();  
  
 Expression exp = parser.parseExpression("new org.chwin.firefighting.apiserver.dsl.lombokTest().m1()");  
  
 System.*out*.println(exp.getValue());  
 // String message = exp.getValue(String.class);  
 // System.out.println(message);  
  
 }  
}

**6.5.9 类型**  
T操作符是一个特殊的操作符、可以同于指定java.lang.Class的实例（类型）。静态方法也可以通过这个操作符调用。  
StandardEvaluationContext使用TypeLocator来查找类型，其中StandardTypeLocator（这个可以被替换使用其他类）默认对java.lang包里的类型可见。也就是说 T()引用java.lang包里面的类型不需要限定包全名，但是其他类型的引用必须要。

|  |  |
| --- | --- |
| 1 | Class dateClass = parser.parseExpression("T(java.util.Date)").getValue(Class.**class**); |

|  |  |
| --- | --- |
| 2 |  |

|  |  |
| --- | --- |
| 3 | Class stringClass = parser.parseExpression("T(String)").getValue(Class.**class**); |

|  |  |
| --- | --- |
| 4 |  |

|  |  |
| --- | --- |
| 5 | **boolean** trueValue = parser.parseExpression( |

|  |  |
| --- | --- |
| 6 | "T(java.math.RoundingMode).CEILING &amp;lt; T(java.math.RoundingMode).FLOOR") |

|  |  |
| --- | --- |
| 7 | .getValue(Boolean.**class**); |

**6.5.10 构造器**  
构造器可以使用new操作符来调用。除了元数据类型和String（比如int,float等可以直接使用）都需要限定类的全名。

[view source](http://ifeve.com/spring-6-spel/" \l "viewSource" \o "view source)[print](http://ifeve.com/spring-6-spel/" \l "printSource" \o "print)[?](http://ifeve.com/spring-6-spel/" \l "about" \o "?)

|  |  |
| --- | --- |
| 1 | nventor einstein = p.parseExpression( |

|  |  |
| --- | --- |
| 2 | "new org.spring.samples.spel.inventor.Inventor('Albert Einstein', 'German')") |

|  |  |
| --- | --- |
| 3 | .getValue(Inventor.**class**); |

|  |  |
| --- | --- |
| 4 |  |

|  |  |
| --- | --- |
| 5 | //create new inventor instance within add method of List |

|  |  |
| --- | --- |
| 6 | p.parseExpression( |

|  |  |
| --- | --- |
| 7 | "Members.add(**new** org.spring.samples.spel.inventor.Inventor( |

|  |  |
| --- | --- |
| 8 | 'Albert Einstein', 'German'))").getValue(societyContext); |

《Spring 5 官方文档》6.Spring表达式语言 \_ 并发编程网 – ifeve.com.html

《Spring 5 官方文档》6.Spring表达式语言 \_ 并发编程网 – ifeve.com.html