Atitit.java expression fsm 表达式分词fsm引擎

[1. 分词 1](#_Toc24116)

[1.1. 分词主要流程 1](#_Toc26022)

[1.2. 界符：“；”分号,“{}”大括号,单引号，双引号 1](#_Toc30619)

[1.3. 分词的属性如下表antrl.token 1](#_Toc18891)

[2. V2版本 1](#_Toc30787)

[3. New()版本 2](#_Toc1587)

[4. 参考 5](#_Toc14087)

# 分词

## 分词主要流程

字符切分，对比字典词典得到分词，并查阅词典，标注词性。完善其他属性，输出

## 词型

共分五类: 保留字、标识符var、常数、运算符、和界符

常数就是一个数,常量是一个不能被改变的变量

## 界符：“；”分号,“{}”大括号,单引号，双引号

界符是不出现在token list 里面儿的..

## 分词的属性如下表antrl.token

|  |  |  |
| --- | --- | --- |
| Col | 列数 |  |
| Filename | 所属文件名 |  |
| Line | 行数 |  |
| Text | 分词 |  |
| Type | 词性 |  |

# V2版本

Add l java common fmt new class() fmt..

V1 only new() ver

Add stataic method invoke

# New()版本

C:\0workspace\AtiPlatf\_cms\src\com\attilax\fsm\JavaExpFsm.java

**String** **code** = "new(com.attilax.util.connReduceDync).set\_resfile(uc\_js.txt).joinNout() "

.trim();

$code=new(com.attilax.user.AgentService).login(admin,admin)

[

"new",

"(",

"com.attilax.user.AgentService",

")",

".",

"login",

"(",

"admin",

"admin",

")"

]

作者::  ★(attilax)>>>  绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

package com.attilax.fsm;

import java.util.List;

import com.google.common.collect.Lists;

public class JavaExpFsm {

List<String> tokens = Lists.newLinkedList();

String curToken = "";

// String splitors = "(),\"";

String curStat = "ini";

private String code;

public char[] code\_char\_arr;

public JavaExpFsm(String code) {

this.code = code;

}

/\*\*

\* http://localhost/new(com.attilax.util.connReduceDync).set\_resfile(uc\_js.txt).joinNout()

\* http://localhost/wrmiServlet?code=new(com.attilax.util.connReduceDync).set\_resfile(uc\_js.txt).joinNout()

\* @param args

\*/

public static void main(String[] args) {

String code = "new(com.attilax.util.connReduceDync).set\_resfile(uc\_js.txt).joinNout() "

.trim();

code= " new(com.attilax.agent.AgentRechargeService).getSubMemTotalsRecycleByAgentId(\"promoter:$pid$,fld2:v2\")";

List li = new JavaExpFsm(code).getTokens();

for (Object object : li) {

System.out.println(object);

}

System.out.println(li);

}

public List getTokens() {

code\_char\_arr = code.toCharArray();

for (char c : code\_char\_arr) {

// get next char,,then change stat

// jude cur char and cur stat...then if or not chage stat

if (c == '(' && !this.curStat.equals("strStart")) { //&& cur stta=ini

this.curStat = "brkStart";

tokens.add(this.curToken);

tokens.add("(");

this.curToken = "";

continue;

}

// if (c == '.' && this.curStat.equals("brkStart")) {

//

// curToken = curToken + String.valueOf(c);

// continue;

//

// // this.curStat.equals("brkEnd"))

//

// }

//

if (c == ')' && !this.curStat.equals("strStart") ) { //&& cur stat =brk start

this.curStat = "brkEnd";

if(this.curToken.length()>0)

tokens.add(this.curToken);

tokens.add(")");

this.curToken = "";

continue;

}

if (c == '.' && this.curStat.equals("brkEnd")) {

tokens.add(".");

curToken = "";

continue;

}

if(c=='\"' && this.curStat.equals("brkStart"))

{

this.curStat = "strStart";

// tokens.add(c);

this.curToken = "";

continue;

}

if(c=='\"' && this.curStat.equals("strStart"))

{

this.curStat = "strEnd";

tokens.add(this.curToken);

this.curToken = "";

continue;

}

if(c==',' && this.curStat.equals("brkStart"))

{

//this.curStat = "strEnd";

tokens.add(this.curToken);

this.curToken = "";

continue;

}

// if (this.curStat.equals("ini"))

curToken = curToken + String.valueOf(c);

}

return tokens;

}

}

# 参考

atitit.词法分析的实现token attilax总结 - attilax的专栏 - 博客频道 - CSDN.NET.html