Atitit.java expression fsm 表达式分词fsm引擎

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",

")"

]

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;

}

}