Atitit. camel分词器 分词引擎 camel拆分 的实现设计

[1. camel分词器 1](#_Toc18153)

[1.1. 实现的界定符号大写字母小写字母数字 1](#_Toc24177)

[1.2. 特殊处理 对于JSONObject 多个大写字母连写的处理 1](#_Toc19298)

[1.3. camel分词器code 1](#_Toc3937)

[2. 代码实现 1](#_Toc29346)

[2.1. public class CamelScannerQ1 extends Scanner { 2](#_Toc3088)

[2.2. public class iniState extends State { 3](#_Toc17249)

[2.3. public class BigCaseLetterState extends State { 4](#_Toc27404)

[2.4. public class LowerLetterState extends State { 5](#_Toc16866)

[2.5. public class NumState extends State { 7](#_Toc28621)

# camel分词器

## 实现的界定符号大写字母小写字母数字

## 特殊处理 对于JSONObject 多个大写字母连写的处理

## camel分词器code

作者:: 老哇的爪子 Attilax 艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

# 代码实现

package com.attilax.fsm.camelScanner;

import java.util.ArrayList;

import java.util.List;

import com.attilax.fsm.Context;

import com.attilax.fsm.FinishState;

import com.attilax.fsm.Scanner;

import com.attilax.fsm.Token;

import com.attilax.json.JSONArray;

import com.attilax.json.JSONObject;

## public class CamelScannerQ1 extends Scanner {

public static void main(String[] args) {

String s = "fld1=1,fld2='at''t,lax',fld3='val3'";

s = "aaaaCamelJSONObject123forMac";

s="com.sun.jna.platform.win32.Variant$VARIANT$\_VARIANT$\_\_VARIANT$BRECORD$ByReference.class";

s="";

// s="a";

@SuppressWarnings("rawtypes")

List tokenList = new CamelScannerQ1().getTokenList(s);

System.out.println(JSONArray.fromObject(tokenList).toString(2));

}

public List<String> getTokenList\_retStr(String s)

{

if(s.length()==0)

return new ArrayList<String>();

List<String> li\_r=new ArrayList<String>();

List<Token> li=getTokenList(s);

for (Token token : li) {

li\_r.add(token.value);

}

return li\_r;

}

public List getTokenList(String s) {

// DslPaserContext context = new DslPaserContext();

Context context = new Context();

context.setState(new iniState());

int n = 0;

while (!(context.state instanceof FinishState)) {

// System.out.println(n);

// ����

context.request(s);

n++;

if (n > 200)

break;

}

for (Token tk : context.tokenList) {

// if(tk.value.trim().length()>0)

System.out.println(tk.value + "===");

}

return (List) context.tokenList;

}

}

package com.attilax.fsm.camelScanner;

import java.util.List;

import com.attilax.fsm.Context;

import com.attilax.fsm.State;

import com.attilax.lang.text.strUtil;

import com.attilax.util.numUtil;

## public class iniState extends State {

@Override

public void handle(String sampleParameter, Context context) {

context.curcharIndex=0;

char curChar=context.sa[context.curcharIndex];

if(strUtil.isBigLetter(curChar))

{

newToken(context, curChar);

context.state=new BigCaseLetterState();

}

if(strUtil.isLowerLetter(curChar))

{

newToken(context, curChar);

context.state=new LowerLetterState();

}

if(numUtil.isNum(curChar))

{

newToken(context, curChar);

context.state=new NumState();

}

}

}

package com.attilax.fsm.camelScanner;

import com.attilax.fsm.Context;

import com.attilax.fsm.FinishState;

import com.attilax.fsm.State;

import com.attilax.fsm.Token;

import com.attilax.lang.text.strUtil;

import com.attilax.util.numUtil;

## public class BigCaseLetterState extends State {

@Override

public void handle(String sampleParameter, Context context) {

if(MoveNextisEnd(context) ) //or next cur move

return;

char curChar=context.sa[context.curcharIndex];

if(strUtil.isBigLetter(curChar))

{

context.curToken.value=context.curToken.value+String.valueOf(curChar);

context.state=new BigCaseLetterState();

}

if(strUtil.isLowerLetter(curChar))

{

context.curToken.value=context.curToken.value+String.valueOf(curChar);

context.state=new LowerLetterState();

}

if(numUtil.isNum(curChar))

{

context.tokenList.add(context.curToken);

context.curToken=new Token();

context.curToken.value=context.curToken.value+String.valueOf(curChar);

context.state=new NumState();

}

}

}

package com.attilax.fsm.camelScanner;

import com.attilax.fsm.Context;

import com.attilax.fsm.FinishState;

import com.attilax.fsm.State;

import com.attilax.fsm.Token;

import com.attilax.lang.text.strUtil;

import com.attilax.util.numUtil;

## public class LowerLetterState extends State {

@Override

public void handle(String sampleParameter, Context context) {

// TODO Auto-generated method stub

if(MoveNextisEnd(context) )

return;

String trim = context.curToken.value.trim();

String allUpperLetter =trim.substring(0,trim.length()-1);

if(allUpperLetter.startsWith("J"))

System.out.println("");

if(strUtil.isBigLetter(allUpperLetter) && allUpperLetter.length()>1) //multi upperLetter

{

final String word=allUpperLetter.substring(0,trim.length()-2);

Token t=new Token(){

{

this.value=word;

}

};

context.tokenList.add(t);

Token t2=new Token();

t2.value=trim.substring( trim.length()-2);

context.curToken=t2;

// context.tokenList.add(t2);

}

char curChar=context.sa[context.curcharIndex];

if(strUtil.isBigLetter(curChar))

{

addCurTokenNnewToken(context, curChar);

context.state=new BigCaseLetterState();

}

if(strUtil.isLowerLetter(curChar))

{

context.curToken.value=context.curToken.value+String.valueOf(curChar);

// context.state=new LowerLetterState(); //state not change

}

if(numUtil.isNum(curChar))

{

addCurTokenNnewToken(context, curChar);

context.state=new NumState();

}

}

package com.attilax.fsm.camelScanner;

import com.attilax.fsm.Context;

import com.attilax.fsm.FinishState;

import com.attilax.fsm.State;

import com.attilax.fsm.Token;

import com.attilax.lang.text.strUtil;

import com.attilax.util.numUtil;

## public class NumState extends State {

@Override

public void handle(String sampleParameter, Context context) {

// TODO Auto-generated method stub

if(MoveNextisEnd(context) )

return;

char curChar=context.sa[context.curcharIndex];

if(strUtil.isBigLetter(curChar))

{

addCurTokenNnewToken(context, curChar);

context.state=new BigCaseLetterState();

}

if(strUtil.isLowerLetter(curChar))

{

addCurTokenNnewToken(context, curChar);

context.state=new LowerLetterState();

}

if(numUtil.isNum(curChar))

{

context.curToken.value=context.curToken.value+String.valueOf(curChar);

// context.state=new NumState();

}

}

}

}