Atitit.减少http请求数方案

[1. 原理与方法 1](#_Toc20188)

[1.1. -------jsp 1](#_Toc12533)

[1.2. "index/js.txt"; 2](#_Toc27998)

[1.3. connReduceDync 2](#_Toc21237)

# 原理与方法

界面调用js.jsp css.js， 如果存在缓存文件就返回缓存文件，否则

读取js css列表文件

读取文件，拼接成一个大的js css文件，然后缓存起来

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

## -------jsp

<%@ page language="java" import="java.util.\*,com.csmy.my.center.util.CTUtils" pageEncoding="UTF-8"%>

<%@ page import="com.attilax.util.connReduceDync"%>

<%@ page import="com.attilax.io.pathx"%>

<%

connReduceDync c=new connReduceDync();

// c.res="com.attilax,core/import.js,core/core.js,core/yaml.js|com.atilax.frmwk,/jquery-1.8.3.min.js ";

c.resfile="index/js.txt";

c.resfileUse=true;

// c.base = "" //def is approot

c.output="index/jobus\_index.js"; //index/jobus.js

c.recreateMode=true; //else use append mode

// System.out.println(c.joinNout()); //trest mode

// System.out.println(); /

// return;

%>

<%=c.readCacheOrjoinNout()%>

## "index/js.txt";

index/img.js

com.atilax.frmwk/jquery-1.8.3.min.js

com.attilax/HRE.js

com.attilax/jsbridge/jsb\_v8q414.js

com.attilax/templete/jquery.tmpl.js

com.atilax.frmwk/unslider.min.js

com.attilax/core/coreV3.js

cmsPhone4jobusImp/list.js

com.attilax/log/logV2.js

## connReduceDync

package com.attilax.util;

import java.io.File;

import java.util.List;

import java.util.Map;

import com.attilax.coll.ListX;

import com.attilax.collection.listUtil;

import com.attilax.io.filex;

import com.attilax.io.pathx;

import com.google.common.collect.Lists;

import com.google.common.collect.Maps;

/\*\*

\* servlet

\* @author Administrator

\*

\* com.attilax.util.connReduceDync

\*

\*/

public class connReduceDync extends ConnReducerV2 {

public String resfile;

public boolean resfileUse=false;

public static void main(String[] args) {

connReduceDync c=new connReduceDync();

c.res="com.attilax,core/import.js,core/core.js,core/yaml.js|com.atilax.frmwk,/jquery-1.8.3.min.js ";

c.resfile="list/list\_jss.txt";

c.resfileUse=true;

// c.base = pathx.webAppPath();

c.output="list/jobus\_list.js"; //index/jobus.js

// c.output\_css=c.base+"/list/jobus\_list.css";

// c.isGeneCss=false;

c.recreateMode=true; //else use append mode

c.genefile=true;

System.out.println(c.joinNout()); //trest mode

// System.out.println(c.readCacheOrjoinNout()); //product mode

// Map m=Maps.newLinkedHashMap();

// m.put("res",s);

System.out.println("--f");

}

public String readCacheOrjoinNout() {

output= pathx.fillFullpath(output);

if(new File(this.output).exists())

return filex.read(this.output);

return joinNout();

}

public String joinNout() {

output= pathx.fillFullpath(output);

gene() ;

return this.resCache;

}

private void gene() {

resfile=pathx.webAppPath()+"/"+resfile;

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

if(resfileUse==false)

li=parseLi(base);

if(resfileUse)

li=listUtil.trimLine( filex.read2list(this.resfile) );

for (String line : li) {

line = line.trim();

if (line.length() == 0)

continue;

importx(line);

}

reduce();

}

private List<String> parseLi(String base) {

List<String> li=Lists.newArrayList();

String reses=res;

String[] rows=reses.split("\\|");

for (String row : rows) {

String[] cols=CsvUtil.toCols(row);

String dir=cols[0];

for (int i=1;i<cols.length;i++) {

String f = cols[i].trim();

if(f.trim().startsWith("/"))

f=f.substring(1);

String full\_file=dir+"/"+f;

li.add(full\_file);

}

}

return li;

}

}