Java 解析stunnel文件，动态增删改

下面的是 一个stunnel.conf文件

; Sample stunnel configuration file for Win32 by Michal Trojnara 2002-2012

; Some options used here may be inadequate for your particular configuration

; This sample file does \*not\* represent stunnel.conf defaults

; Please consult the manual for detailed description of available options

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \* Global options \*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Debugging stuff (may useful for troubleshooting)

;debug = 7

;output = stunnel.log

; Disable FIPS mode to allow non-approved protocols and algorithms

;fips = no

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \* Service defaults may also be specified in individual service sections \*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Certificate/key is needed in server mode and optional in client mode

cert = stunnel.pem

;key = stunnel.pem

; Authentication stuff needs to be configured to prevent MITM attacks

; It is not enabled by default!

;verify = 2

; Don't forget to c\_rehash CApath

;CApath = certs

; It's often easier to use CAfile

;CAfile = certs.pem

; Don't forget to c\_rehash CRLpath

;CRLpath = crls

; Alternatively CRLfile can be used

;CRLfile = crls.pem

; Disable support for insecure SSLv2 protocol

options = NO\_SSLv2

; Workaround for Eudora bug

;options = DONT\_INSERT\_EMPTY\_FRAGMENTS

; These options provide additional security at some performance degradation

;options = SINGLE\_ECDH\_USE

;options = SINGLE\_DH\_USE

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \* Service definitions (at least one service has to be defined) \*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Example SSL server mode services

[pop3s]

accept = 995

connect = 110

[imaps]

accept = 993

connect = 143

[ssmtp]

accept = 465

connect = 25

; Example SSL client mode services

;[gmail-pop3]

;client = yes

;accept = 127.0.0.1:110

;connect = pop.gmail.com:995

;[gmail-imap]

;client = yes

;accept = 127.0.0.1:143

;connect = imap.gmail.com:993

;[gmail-smtp]

;client = yes

;accept = 127.0.0.1:25

;connect = smtp.gmail.com:465

; Example SSL front-end to a web server

;[https]

;accept = 9999

;connect = 192.168.160.94:8888

; "TIMEOUTclose = 0" is a workaround for a design flaw in Microsoft SSL

; Microsoft implementations do not use SSL close-notify alert and thus

; they are vulnerable to truncation attacks

;TIMEOUTclose = 0

; vim:ft=dosini

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

public class ModifyConfFile

{

// 默认节点

private static String GLOBLE\_CONFIG = "\_\_globle";

// 默认注释符号

private String[] comments = {"#",";"};

// 判断此行是否是注释行

private boolean isComment(String line)

{

for (String comment : comments)

{

if (line.startsWith(comment))

{

return true;

}

}

return false;

}

// 行开始是否用[]框起来的

private boolean isSection(String line)

{

return line.startsWith("[") && line.indexOf("]") > 0;

}

public static void main(String[] args) throws IOException

{

String path = "F:/stunnel222.conf";

FileReader fr = new FileReader(path);

BufferedReader br = new BufferedReader(fr,20480);

//FileWriter fw = new FileWriter(path,false);

//BufferedWriter bw = new BufferedWriter(fw,20480);

String s = null;

String content = "";

ModifyConfFile m = new ModifyConfFile() ;

// 用来存放配置文件中的内容

Map<String, Map<String, String>> configMap = new HashMap<String, Map<String, String>>();

Map<String, String> inMap = new HashMap<String, String>();

Map<String, String> inMap2 = new HashMap<String, String>();

configMap.put(GLOBLE\_CONFIG, inMap);

// 按行读取，将内容存入map中

while((s = br.readLine()) != null){

if(m.isComment(s)){

continue;

}else if(m.isSection(s)){

String section = s.substring(1, s.indexOf("]"));

if (configMap.containsKey(section))

{

inMap = configMap.get(section);

}

else

{

inMap = new HashMap<String, String>();

configMap.put(section, inMap);

}

}else{

int index = s.indexOf("=");

String key = s;

String val = null;

if (index != -1)

{

key = s.substring(0, index);

val = s.substring(index + 1);

}

if("".equals(s) || s == null){

inMap = configMap.get(GLOBLE\_CONFIG);

}

if(!"".equals(s) && s != null){

inMap.put(key == null ? key : key.trim(), val == null ? val : val.trim());

}

}

content += s +"\r\n";

}

System.out.println(content);

System.out.println("=============================================================");

// 获取[luweiwei]的节点下的map

Map<String, String> tempMap = configMap.get("luweiwei");

// 修改[luweiwei]的节点下的map中的connect对应的value

tempMap.put("connect", "192.168.4.185:8081");

configMap.put("luweiwei", tempMap);

String newContent ="";

for (Map.Entry<String, Map<String, String>> entry : configMap.entrySet()) {

if(GLOBLE\_CONFIG.equals(entry.getKey())){

for (Map.Entry<String, String> entry2 : entry.getValue().entrySet()) {

newContent = newContent + entry2.getKey() + "=" + entry2.getValue() + "\r\n";

}

newContent += "\r\n";

}else{

newContent += "["+entry.getKey()+"]"+"\r\n";

for (Map.Entry<String, String> entry2 : entry.getValue().entrySet()) {

newContent = newContent + entry2.getKey() + "=" + entry2.getValue() + "\r\n";

}

newContent += "\r\n";

}

}

System.out.println(newContent);

System.out.println("=============================================================");

// 写进stunnel.conf文件中去

FileWriter fw = new FileWriter(path,false);

BufferedWriter bw = new BufferedWriter(fw,20480);

bw.write(newContent);

bw.flush();

// 打印查看修改后的内容

content = "";

String path2 = "F:/stunnel222.conf";

FileReader fr2 = new FileReader(path2);

BufferedReader br2 = new BufferedReader(fr2,20480);

while((s = br2.readLine()) != null){

content += s +"\r\n";

}

System.out.println(content);

}

}