

昵称：凯-子

园龄：1个月

粉丝：40

关注：14

+加关注

< 2016年2月 >

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	1	2	3	4	5
6	7	8	9	10	11	12

搜索

找找看

谷歌搜索

- 常用链接
- 我的随笔

我的评论

我的参与

最新评论

我的标签

博客园 首页 新随笔 联系 管理 订阅 XML

python之haproxy配置文件操作（第三天）

作业：

对haproxy配置文件进行操作

要求：

对haproxy配置文件中backend下的server实现增删改查的功能

一、这个程序有二个版本

- 1、python2.7版本见haproxy_python27.py
- 2、python3.4版本见haproxy_python34.py

二、具体实现了如下功能：

- 1、输入1，进入backend菜单，查询server信息
- 2、输入2，进入backend菜单，添加server条目
- 3、输入3，进入backend菜单，选择server条目，进入修改环节
- 4、输入4，进入backend菜单，选择server条目，进入删除环节
- 5、输入5，退出程序

[更多链接](#)[最新随笔](#)

1. python学习笔记（第五天）
2. python学习笔记（第四天）
3. python之计算器（第四天）
4. Git中pull对比fetch和merge
5. git操作笔记
6. python之haproxy配置文件操作（第三天）
7. python学习笔记（第三天）
8. python之三层菜单递归
9. python之编写购物车（第二天）
10. window7下安装第三方包报错及解决

[我的标签](#)[python笔记\(5\)](#)[python作业\(5\)](#)[python\(4\)](#)[git\(2\)](#)[python问题\(1\)](#)[随笔分类\(14\)](#)[python\(7\)](#)[python作业\(5\)](#)[shell](#)[其他\(2\)](#)[随笔档案\(15\)](#)[2016年1月 \(8\)](#)[2015年12月 \(7\)](#)[文章分类\(4\)](#)[第一天\(4\)](#)

三、haproxy配置文件如下：

```
global
    log 127.0.0.1 local2
    daemon
    maxconn 256
    log 127.0.0.1 local2 info
defaults
    log global
    mode http
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms
    option dontlognull

listen stats :8888
    stats enable
    stats uri /admin
    stats auth admin:1234

frontend oldboy
    bind 0.0.0.0:80
    option httplog
    option httpclose
    option forwardfor
    log global
    acl www hdr_reg(host) -i www.oldboy.org
    use_backend www.oldboy.org if www

backend www.oldboy.org
    server 100.1.7.9 100.1.7.9 weight 20 maxconn 3000
    server 100.1.7.10 100.1.7.10 weight 10 maxconn 2000

frontend mysql
    bind *:3306
    mode tcp
    log global
    default_backend mysqlserver

backend mysqlserver
```

文章档案(6)

2016年1月 (4)

2015年12月 (2)

链接收藏

docker学习

凯哥的博客

廖雪峰的官方网站

南非波波(github)

女神博客

田埂上的梦

曾春云(github)

张晓宇(github)

积分与排名

积分 - 2180

排名 - 40590

最新评论

1. Re:python之编写购物车（第二天）

牛逼 哥你是怎么学的啊？

--知其然，知其所以然

2. Re:python之编写购物车（第二天）

两个字：NB

--揉碎的青春

3. Re:python之编写购物车（第二天）

哥们几期的

--bdslinux

4. Re:python之编写登陆接口（第一天）

@南非波波那就按部就班的关闭，当然用with更好。...

--凯-子

5. Re:python之编写登陆接口（第一

```
server mysql1 10.1.1.110:3306 weight 20 maxconn 300
server mysql2 10.1.1.120:3306 weight 10 maxconn 200
```



四、流程图如下：

天)

@凯-子文件不会自动关闭啊亲，除非你使用with关键字。否则就得按部就班的操作文件open write/read close...

--南非波波

6. Re:python之编写购物车（第二天）

小伙子 挺牛逼的。看来我也要加油啊

--奔腾的狮子

7. Re:python之编写购物车（第二天）

@daivlin是用processon画的。在线画图工具...

--凯-子

8. Re:python之编写购物车（第二天）

流程图是用什么画的啊？

--daivlin

9. Re:python之编写登陆接口（第一天）

@南非波波是不会执行到最后那个语句，程序关闭，文件也就自动关闭了...

--凯-子

10. Re:python之编写登陆接口（第一天）

很好。不过我能提一个疑问吗？就是你的文件为什么不在你将用户加锁退出程序的时候也加个关闭的操作，因为这个时候不会执行到程序的最后那个语句

--南非波波

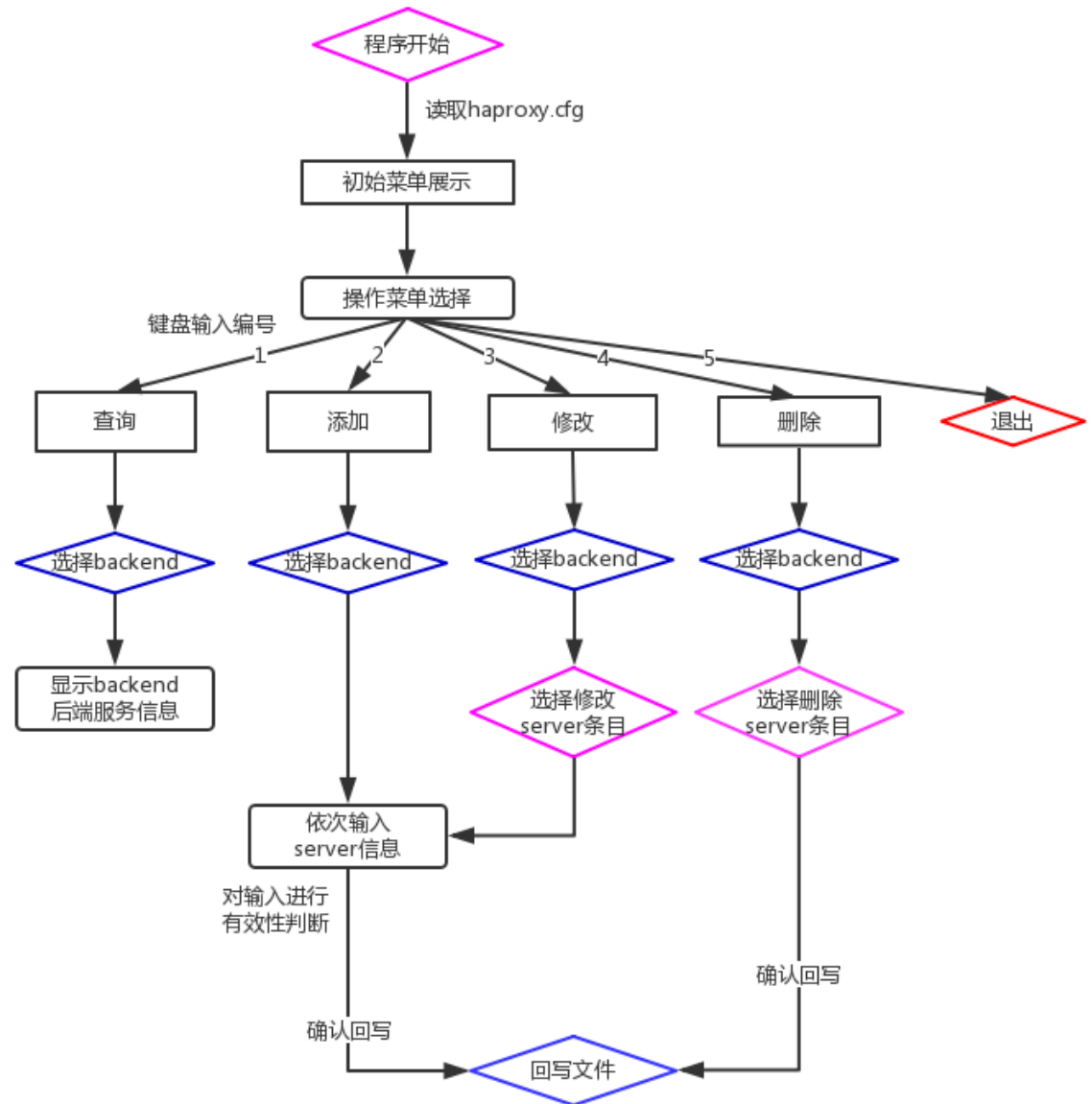
阅读排行榜

1. python之编写登陆接口（第一天）(405)
2. python之编写三层菜单（第一天）(325)
3. python之编写购物车（第二天）(305)
4. python学习笔记（第一天）(215)
5. python学习笔记（第二天）(150)
6. python之三层菜单递归(121)

haproxy配置文件修改流程图：

备注：
1、8个函数
2、5层菜单

- 输入字符类型转化函数
- 名称或变量名输入判断函数
- IP地址及端口输入判断函数
- 数字输入判断函数
- backend输入判断函数
- backend列表展示函数
- 显示backend后端服务函数
- 回写配置文件函数



五、python2.7代码：

7. python学习笔记（第三天）(119)
8. python代码中碰到的问题及解决(60)
9. python之haproxy配置文件操作（第三天）(56)
10. python之计算器（第四天）(40)

评论排行榜

1. python之编写购物车（第二天）(7)
2. python之编写登陆接口（第一天）(4)
3. python之编写三层菜单（第一天）(2)
4. python学习笔记（第二天）(1)

推荐排行榜

1. python之编写三层菜单（第一天）(3)
2. python之haproxy配置文件操作（第三天）(1)
3. python之编写购物车（第二天）(1)



```
#!/usr/bin/python27
#_*_ coding=utf-8 *_*
'''
Created on 2016年1月17日
@author: 王凯
'''

import os,sys,re,time
from collections import defaultdict,OrderedDict

#####raw_input输入字符类型转化函数#####
def input_handle(s):
    if str.isdigit(s):
        s = int(s)
    return s

#####名称或变量名输入判断函数#####
def name_decide():
    (name,name_flag) = ('',True)

    while name_flag:
        name_input = raw_input('\033[32m请输入新服务名:(退出输入q)\033[0m')
        if len(name_input) == 0:
            continue
        elif name_input == 'q':
            name_flag = False
        elif re.match('[0-9a-zA-Z\_]+',name_input):
            name = name_input
            name_flag = False
        else:
            print('\033[31m名称输入错误,请重新输入!!!\033[0m')
    return(name)

#####IP地址及端口输入判断函数#####
def ipaddress_decide():
    (address,address_flag) = ('',True)

    while address_flag:
        address_input = raw_input('\033[32m请输入新地址(IP哦):(退出输入q)\033[0m')
```

```

if len(address_input) == 0:
    continue
elif address_input == 'q':
    address_flag = False
    #####匹配输入是否是ip:port的格式#####
    elif re.match('(25[0-5]|2[0-4]\d|[0-1]?[0-9]?[0-9])\.((25[0-5]|2[0-4]\d|[0-1]?[0-9]?[0-9])?)\{3\}(\:\d{1,5})?'
$,address_input):
    address = address_input
    address_flag = False
    else:
        print('\033[31m地址输入错误, 请重新输入!!!\033[0m')
    return(address)

#####数字输入判断函数#####
def number_decide(name):
    (number,number_flag) = ('',True)

    while number_flag:
        number_input = raw_input('\033[32m请输入 %s (数字哦):(退出输入q)\033[0m' % name)
        if len(number_input) == 0:
            continue
        elif number_input == 'q':
            number_flag = False
        else:
            try:
                int(number_input)
            except:
                print('\033[31m %s 输入错误, 请重新输入!!!\033[0m' % name)
            else:
                number = number_input
                number_flag = False

    return(number)

#####backend输入判断函数#####
def backend_input_if(input_index,input_dict):
    (input_name,input_flag) = ('',True)

    #####如果输入非空, 对输入进行判断并转化类型#####

```

```

if len(input_index) != 0:
    input_index = input_handle(input_index)
if input_index == 'b':
    input_flag = False
elif input_index in input_dict.keys():
    input_name = input_dict[input_index]
elif input_index in input_dict.values():
    input_name = input_index
else:
    input_name = ''

return(input_name,input_flag)

#####backend列表展示函数#####
def backend_read(file):
    backend_list = []
    show_dict = {}
    backend_name_dict = defaultdict(list)

    server_flag = False

    with open(file,'r') as ha:
        for line in ha:
            server_dict = OrderedDict()
            line = line.strip('\n')

            if re.match('backend',line):
                backend_name = re.split('\s+',line)[1]
                backend_list.append(backend_name)

                server_flag = True
            elif server_flag and re.match('\s+server',line):
                server_info = re.split('\s+',line)

                server_dict['name'] = server_info[2]
                server_dict['address'] = server_info[3]
                server_dict['weight'] = server_info[5]

```

###如果输入为b,则退出程序###

###如果输入为数字编号,则从字典中获取具体backend名称

###如果输入为backend名称,则直接获取###

###输入其他字符,赋值变量为空###

###返回输入的结果和循环标志位###

###初始化backend的列表###

###初始化要返回的显示字典###

###定义一个value为列表的字典,用来存放server###

###初始化server判断标志位###

###打开haproxy配置文件###

###定义一个有序字典###

###匹配配置文件以backend开头的行###

###将配置到的backend的名称插入到列表###

###赋值标志位为真,用来与server关联###

###匹配配置文件以server开头的行###

###对server进行分隔###

###对server的具体信息进行字典赋值###

```

server_dict['maxconn'] = server_info[7]

        backend_name_dict[backend_name].append(server_dict)                                ###将server字典与backend的名称进行关联###
    else:
        server_flag = False                                                                ###当server没匹配到，赋值标志位为假，结束关联###

for k,v in enumerate(backend_list,1):
    show_dict[k] = v                                                                      ###对backend名称进行新字典赋值，方便查询和展示

###
    print('%d . %s' % (k,v))                                                            ###输出backend列表###

return(show_dict,backend_name_dict)                                                       ###返回显示的字典和backend-server字典###

#####显示backend后端服务器函数#####
def backend_server_show(backend_show_value,server_show_dict):

    #####对backend名称进行遍历并加上数字编号#####
    print('\n=====')
    print('后端服务器(%s)信息如下 : ' % backend_show_value)
    print('%-5s  %-20s  %-20s  %-20s  %-20s' % ('id','name','address','weight','maxconn'))

    server_list = server_show_dict[backend_show_value]
    for k,v in enumerate(server_list,1):                                                  ###用enumerate进行server展示###
        print '%-5s' % k + ' ',
        for kk,vv in v.items():
            print '%-20s' % vv + ' ',
        print
    print('\n=====')

#####backend后端服务操作并回写配置文件函数#####
def backend_server_handle(file,handle_dict):
    newfile = '%s.new' % file                                                            ###定义回写的新文件###
    server_flag = False

    with open(file,'r') as read_file,open(newfile,'w') as write_file:                    ###同时打开二文件，一个读，一个写###
        for line in read_file:
            if re.match('backend',line):                                                  ###匹配到backend行时进行server信息插入###
                write_file.write(line)

```



```

backend_name = re.split('\s+',line)[1]
for server_dict in handle_dict[backend_name]:          ###对backend-server字典进行遍历###
    server_line = '\tserver {name} {address} weight {weight} maxconn {maxconn}\n'
    write_file.write(server_line.format(**server_dict))    ###将指定的backend下的server条目插入文件###

    server_flag = True                                     ###指定标志位为真，方便server判断###
elif server_flag and re.match('\s+server',line):         ###匹配server开头的行，跳过不做任何操作###
    pass
else:
    write_file.write(line)                                ###其他的行，直接插入新文件###
    server_flag = False
print('\033[33m server更新成功\033[0m')
os.system('mv %s %s.bak' % (file,file))                  ###对源配置文件进行备份###
os.system('mv %s %s' % (newfile,file))                    ###对新生成的配置文件进行改名###

#####主程序开始#####

if __name__ == '__main__':
    flag = True
    haproxy_file = 'haproxy.cfg'                          ###指定haproxy配置文件###

    backend_name = ''
    haproxy_name_show_dict = {}                            ###初始化backend显示字典###
    haproxy_server_show_dict = {}                          ###初始化server显示字典###
    while flag:
        os.system('clear')                                  ###清屏###
        print('\n=====')
        print('\033[33m 欢迎访问haproxy配置文件管理平台： \033[0m')
        print('\nbackend列表信息如下： ')
        (haproxy_name_show_dict,haproxy_server_show_dict) = backend_read(haproxy_file)
        print('\n=====')
        print '''
\033[31m 1. 查询后端服务\033[0m
\033[32m 2. 添加后端服务\033[0m
\033[33m 3. 修改后端服务\033[0m
\033[35m 4. 删除后端服务\033[0m
\033[36m 5. 退出\033[0m '''
        print('\n=====')

        select_num = raw_input('\033[33m请选择操作条目： \033[0m')

```

```

#####1. 查询后端服务#####
if select_num == '1':
    query_flag = True                                ###初始化查询循环标志###
    while query_flag:
        backend_index = raw_input('\033[32;1m请输入查询的backend编号或名称 (返回上层菜单, 请输入b): \033[0m')

        #####对输入值进行判断#####
        (backend_name, query_flag) = backend_input_if(backend_index, haproxy_name_show_dict)
        if backend_name:                                ###显示对应backend下的server列表###
            backend_server_show(backend_name, haproxy_server_show_dict)

#####2. 添加后端服务#####
if select_num == '2':
    add_flag = True                                    ###初始化添加循环标志###
    while add_flag:
        backend_index = raw_input('\033[32;1m请输入添加的backend的编号或名称 (返回上层菜单, 请输入b): \033[0m')
        #####对输入值进行判断#####
        (backend_name, add_flag) = backend_input_if(backend_index, haproxy_name_show_dict)

        if backend_name:                                ###显示对应backend下的server列表###
            backend_server_show(backend_name, haproxy_server_show_dict)

        add_server_dict = OrderedDict()
        print('\033[34;1m请依次输入后端服务信息 (name, address, weight, maxconn): \033[0m')
        add_server_dict['name'] = name_decide()          ###对输入的name有效性进行判断###
        add_server_dict['address'] = ipaddress_decide()  ###对输入的IP有效性进行判断###
        add_server_dict['weight'] = number_decide('权重值')  ###对输入的权重有效性进行判断###
        add_server_dict['maxconn'] = number_decide('最大连接数')  ###对输入的连接数有效性进行判断###

print(add_server_dict['name'], add_server_dict['address'], add_server_dict['weight'], add_server_dict['maxconn'])

#####对输入四个服务信息是否成功判断#####
if add_server_dict['name'] and add_server_dict['address'] and add_server_dict['weight'] and
add_server_dict['maxconn']:
    add_commit = raw_input('请确认是否添加此条目 (y | n): ')
    if add_commit == 'y':                                ###确认添加服务条目, 并回写配置文件###
        haproxy_server_show_dict[backend_name].append(add_server_dict)
        backend_server_handle(haproxy_file, haproxy_server_show_dict)

```

```

else:
    add_flag = False                                     ###否则退出本次循环###

else:
    print('\033[31m server输入信息有误, 请重新输入!!!\033[0m')

#####3. 修改后端服务#####
if select_num == '3':
    backend_modify_flag = True                           ###初始化修改循环标志###
    while backend_modify_flag:
        backend_index = raw_input('\033[32;1m请输入修改的backend的编号或名称(返回上层菜单, 请输入b): \033[0m')
        #####对输入值进行判断#####
        (backend_name, backend_modify_flag) = backend_input_if(backend_index, haproxy_name_show_dict)

        if backend_name:                                ###显示对应backend下的server列表###
            backend_server_show(backend_name, haproxy_server_show_dict)

            server_modify_flag = True                    ###初始化server条目修改标志###
            while server_modify_flag:

                server_index = raw_input('\033[32;1m请输入修改的server的编号(返回上层菜单, 请输入b): \033[0m')
                if len(server_index) != 0:
                    server_index = input_handle(server_index)
                if server_index == 'b':                    ###输入b, 返还上一层###
                    server_modify_flag = False
                #####指定具体的server编号进行判断#####
                elif server_index >= 1 and server_index <= len(haproxy_server_show_dict[backend_name]):
                    modify_server_dict = OrderedDict()
                    print('\033[34;1m请依次输入后端服务信息 (name,address,weight,maxconn):\033[0m')
                    modify_server_dict['name'] = name_decide()          ###对输入的四个信息有效性进行判断###
                    modify_server_dict['address'] = ipaddress_decide()
                    modify_server_dict['weight'] = number_decide('权重值')
                    modify_server_dict['maxconn'] = number_decide('最大连接数')

print(modify_server_dict['name'], modify_server_dict['address'], modify_server_dict['weight'], modify_server_dict['maxconn'])

#####对输入的四个服务信息是否成功判断#####
if modify_server_dict['name'] and modify_server_dict['address'] and
modify_server_dict['weight'] and modify_server_dict['maxconn']:

```

```

        modify_commit = raw_input('请确认是否修改此条目 (y | n) : ')
        if modify_commit == 'y':
            ##确认修改服务条目，并回写配置文件##
            haproxy_server_show_dict[backend_name][server_index - 1] = modify_server_dict
            backend_server_handle(haproxy_file,haproxy_server_show_dict)
            modify_server_flag = False

        else:
            ##否则退出本次循环##
            modify_server_flag = False
    else:
        print('\033[31m server输入信息有误，请重新输入!!!\033[0m')
    else:
        print('\033[31m server编号输入错误，请重新输入!!!\033[0m')

#####4. 删除后端服务#####
if select_num == '4':
    backend_delete_flag = True
    ##初始化删除循环标志##
    while backend_delete_flag:
        backend_index = raw_input('\033[32;1m请输入删除的backend下条目编号或名称 (返回上层菜单, 请输入b): \033[0m')
        #####对输入值进行判断#####
        (backend_name,backend_delete_flag) = backend_input_if(backend_index,haproxy_name_show_dict)

        if backend_name:
            ##显示对应backend下的server列表##
            backend_server_show(backend_name,haproxy_server_show_dict)

            server_delete_flag = True
            ##初始化server条目删除标志位##
            while server_delete_flag:

                server_index = raw_input('\033[32;1m请输入删除的server的编号 (返回上层菜单, 请输入b): \033[0m')
                if len(server_index) != 0:
                    server_index = input_handle(server_index)
                if server_index == 'b':
                    ##输入b，返还上一层##
                    server_delete_flag = False
                #####指定具体的server编号进行判断#####
                elif server_index >= 1 and server_index <= len(haproxy_server_show_dict[backend_name]):

                    print(haproxy_server_show_dict[backend_name][server_index - 1])
                    delete_commit = raw_input('请确认是否删除此条目 (y | n) : ')
                    if delete_commit == 'y':
                        ##确认删除服务条目，并回写配置文件##
                        del haproxy_server_show_dict[backend_name][server_index - 1]
                        backend_server_handle(haproxy_file,haproxy_server_show_dict)

```

```

        delete_server_flag = False

    else:
        delete_server_flag = False
    else:
        print('\033[31m server编号输入错误，请重新输入!!!\033[0m')

#####5. 退出程序#####
if select_num == '5':
    sys.exit('\033[32m 退出程序，欢迎下次光临，谢谢\033[0m')

```



六、python3.4代码：



```

import time,os,datetime,sys,MySQLdb,logging

logging.basicConfig(level=logging.DEBUG,
                    format='%(asctime)s %(filename)s[line:%(lineno)d] %(levelname)s %(message)s',
                    datefmt='%Y %m %d %H:%M:%S',
                    # filename=filename,
                    filename='/home/logs/svncommit.log',
                    filemode='a')

g_DB_CONFIG={
    'host':'127.0.0.1',
    'user':'root',
    'passwd':'',
    'db':''
}

def GetCurPathInfo():
    today_time=datetime.datetime.now()
    today_0_time = datetime.datetime(today_time.year, today_time.month, today_time.day)
    yestoday_time=datetime.datetime.now()-datetime.timedelta(days=0)
    yestoday_0_time = datetime.datetime(yestoday_time.year, yestoday_time.month, yestoday_time.day)
    todayUnix = int(time.mktime(today_0_time.timetuple()))
    yesterdayUnix = int(time.mktime(yestoday_0_time.timetuple()))

```

```
logging.info('yestertoday : date %s, unix_time %s' %(yestoday_0_time,yesterdayUnix))
logging.info('today : date %s, unix_time %s' %(today_0_time,todayUnix))
CurPath='/home/svnroot'
ChildrenList = os.listdir(CurPath)
# logging.info(ChildrenList)
appidList=[]
index = 0
for name in ChildrenList:
    index = index + 1
    if index%500==0:
        logging.info (index)
    if len(name) != 14:
        continue
    dbName = os.path.join(CurPath,name,'db')
    if not os.path.exists(dbName):
        logging.info('db is not exists, appId is '+name)
        continue
    # logging.info(dbName)
    appFileStats = os.stat(os.path.join(CurPath,name))
    appModifyTime = appFileStats.st_mtime
    fileStats = os.stat(dbName)
    modifyTime = fileStats.st_mtime
    if modifyTime > yesterdayUnix :
        appidList.append(name)
        logging.info ('today create appid submit '+name)
logging.info (appidList)
logging.info ('len is :'+str(len(appidList)))

yestoday_timeStr='%d-%02d-%02d' % (yestoday_time.year, yestoday_time.month,yestoday_time.day)
# logging.info(yestoday_timeStr)
values=[]
svn_ip='10.124.156.225'
for appid in appidList:
    values.append((appid,yestoday_timeStr,svn_ip))
# logging.info(values)
# try:
#
conn=MySQLdb.connect(host=g_DB_CONFIG['host'],user=g_DB_CONFIG['user'],passwd=g_DB_CONFIG['passwd'],port=3306,db=g_DB_CONFIG['db'])
# cur=conn.cursor()
```

```
# cur.executemany('insert ignore into statistic_svncommit(appId,reportDate,svn_ip) values(%s,%s,%s)',values)
# conn.commit()
# cur.close()
# conn.close()
except MySQLdb.Error,e:
    logging.info( "Mysql Error %d: %s" % (e.args[0], e.args[1]))
    raise e
return

if __name__ == '__main__':
    try:
        logging.info('-'*60 + 'begin Statistic_Svncommit'+'-'*60)
        GetCurPathInfo()
        logging.info('-'*60+'end Statistic_Svncommit'+'-'*60)
    except:
        info = sys.exc_info()
        for file, lineno, function, text in traceback.extract_tb(info[2]):
            logging.error(text)
    else:
        logging.info('SUCCESS')
```



七、效果图：

1、 初始菜单直接显示backend列表，并列出选择菜单：

```
=====
欢迎访问haproxy配置文件管理平台：
```

```
backend列表信息如下：
```

```
1 . www.oldboy.org
2 . mysqlserver
```

```
=====

1. 查询后端服务
2. 添加后端服务
3. 修改后端服务
4. 删除后端服务
5. 退出
```

```
=====
请选择操作条目：█
```

2、 输入1，选择查询菜单，并输入backend的名称或编号，均可，并展示对应名称下的server信息：


```

=====
欢迎访问haproxy配置文件管理平台：

backend列表信息如下：
1 . www.oldboy.org
2 . mysqlserver

=====

1. 查询后端服务
2. 添加后端服务
3. 修改后端服务
4. 删除后端服务
5. 退出

=====
请选择操作条目：1
请输入查询的backend编号或名称(返回上层菜单,请输入b): mysqlserver

=====
后端服务器(mysqlserver)信息如下：
id      name      address      weight      maxconn
1       mysql1    10.1.1.110:3306  20          300
2       mysql2    10.1.1.120:3306  30          200
3       wangkai   12.12.12.120:3306  30          40
4       kevin     10.10.10.10:2222  30          4000
5       sky       192.168.100.33:8888  30          20000

=====
请输入查询的backend编号或名称(返回上层菜单,请输入b): █

```

3、 输入b可以返回上层菜单，输入2进入添加server条目：

```
=====
欢迎访问haproxy配置文件管理平台：

backend列表信息如下：
1 . www.oldboy.org
2 . mysqlserver

=====

1. 查询后端服务
2. 添加后端服务
3. 修改后端服务
4. 删除后端服务
5. 退出

=====
请选择操作条目：2
请输入添加的backend的编号或名称(返回上层菜单,请输入b): 1

=====
后端服务器(www.oldboy.org)信息如下：
id      name      address      weight      maxconn
1       100.1.7.9     100.1.7.9     20          3000
2       100.1.7.12    100.1.7.12:80 30          4000
3       sky           100.1.7.15:8000 40          50000

=====
请依次输入后端服务信息(name,address,weight,maxconn):
请输入新服务名:(退出输入q)
```

4、 输入server对应的name, address, weight, maxconn, 并对有效性进行判断:

```
=====
请选择操作条目: 2
```

```
请输入添加的backend的编号或名称(返回上层菜单,请输入b): 2
```

```
=====
后端服务器(mysqlserver)信息如下:
```

id	name	address	weight	maxconn
1	mysql1	10.1.1.110:3306	20	300
2	mysql2	10.1.1.120:3306	30	200
3	wangkai	12.12.12.120:3306	30	40
4	kevin	10.10.10.10:2222	30	4000
5	sky	192.168.100.33:88888	30	20000

```
=====
请依次输入后端服务信息(name,address,weight,maxconn):
```

```
请输入新服务名:(退出输入q)!aa
```

```
名称输入错误,请重新输入!!!
```

```
请输入新服务名:(退出输入q)test
```

```
请输入新地址(IP哦):(退出输入q)192.168.100.3aa
```

```
地址输入错误,请重新输入!!!
```

```
请输入新地址(IP哦):(退出输入q)192.168.100.3:8080
```

```
请输入 权重值 (数字哦):(退出输入q)aa
```

```
权重值 输入错误,请重新输入!!!
```

```
请输入 权重值 (数字哦):(退出输入q)30
```

```
请输入 最大连接数 (数字哦):(退出输入q)aa
```

```
最大连接数 输入错误,请重新输入!!!
```

```
请输入 最大连接数 (数字哦):(退出输入q)3000
```

```
('test', '192.168.100.3:8080', '30', '3000')
```

```
请确认是否添加此条目 (y | n): y
```

```
server更新成功
```

```
请输入添加的backend的编号或名称(返回上层菜单,请输入b):
```

添加成功后查看结果:

```
=====
请输入添加的backend的编号或名称(返回上层菜单,请输入b): 2
```

```
=====
后端服务器(mysqlserver)信息如下:
```

id	name	address	weight	maxconn
1	mysql1	10.1.1.110:3306	20	300
2	mysql2	10.1.1.120:3306	30	200
3	wangkai	12.12.12.120:3306	30	40
4	kevin	10.10.10.10:2222	30	4000
5	sky	192.168.100.33:88888	30	20000
6	test	192.168.100.3:8080	30	3000

注:

name以数字、字母、下划线开头（其中包含数字的原因是可以输入**ip**地址）

address可以是单独的**ip**地址，也可以是**ip**加端口的形式，例：**192.168.100.3**或**192.168.100.3:8080**

weight和**maxconn**必须输入数字，大小暂时没有限制

5、输入b可以返回上层菜单，输入3进入修改server条目，并对**www.oldboy.org**下的**sky**条目进行修改：

```
=====
欢迎访问haproxy配置文件管理平台：

backend列表信息如下：
1 . www.oldboy.org
2 . mysqlserver

=====

1. 查询后端服务
2. 添加后端服务
3. 修改后端服务
4. 删除后端服务
5. 退出

=====
请选择操作条目：3
请输入修改的backend的编号或名称(返回上层菜单,请输入b)：1

=====
后端服务器(www.oldboy.org)信息如下：
id      name      address      weight      maxconn
1       100.1.7.9    100.1.7.9    20          3000
2       100.1.7.12   100.1.7.12:80 30          4000
3       sky          100.1.7.15:8000 40          50000

=====
请输入修改的server的编号(返回上层菜单,请输入b)：3
请依次输入后端服务信息(name,address,weight,maxconn):
请输入新服务名:(退出输入q)
```

同样对输入的server有效性进行判断，最后确认回写配置文件，查看修改结果：

=====

后端服务器(www.oldboy.org)信息如下：

id	name	address	weight	maxconn
1	100.1.7.9	100.1.7.9	20	3000
2	100.1.7.12	100.1.7.12:80	30	4000
3	sky	100.1.7.15:8000	40	50000

=====

请输入修改的server的编号(返回上层菜单,请输入b): 3

请依次输入后端服务信息(name,address,weight,maxconn):

请输入新服务名:(退出输入q)100.1.7.13

请输入新地址(IP哦):(退出输入q)100.1.7.13

请输入 权重值 (数字哦):(退出输入q)40

请输入 最大连接数 (数字哦):(退出输入q)4000

('100.1.7.13', '100.1.7.13', '40', '4000')

请确认是否修改此条目 (y | n): y

server更新成功

请输入修改的server的编号(返回上层菜单,请输入b): b

请输入修改的backend的编号或名称(返回上层菜单,请输入b): 1

=====

后端服务器(www.oldboy.org)信息如下：

id	name	address	weight	maxconn
1	100.1.7.9	100.1.7.9	20	3000
2	100.1.7.12	100.1.7.12:80	30	4000
3	100.1.7.13	100.1.7.13	40	4000

=====

请输入修改的server的编号(返回上层菜单,请输入b):

6、 输入b可以返回上层菜单，输入4进入删除server条目，并对指定条目进行修改：

```

=====
请选择操作条目: 4
请输入删除的backend下条目编号或名称(返回上层菜单,请输入b): 2
=====

后端服务器(mysqlserver)信息如下:
id      name      address      weight      maxconn
1      mysql1      10.1.1.110:3306      20      300
2      mysql2      10.1.1.120:3306      30      200
3      wangkai      12.12.12.120:3306      30      40
4      kevin      10.10.10.10:2222      30      4000
5      sky      192.168.100.33:8888      30      20000
6      test      192.168.100.3:8080      30      3000

=====
请输入删除的server的编号(返回上层菜单,请输入b): 6
OrderedDict([('name', 'test'), ('address', '192.168.100.3:8080'), ('weight', '30'), ('maxconn', '3000')])
请确认是否删除此条目 (y | n): y
server更新成功
请输入删除的server的编号(返回上层菜单,请输入b): b
请输入删除的backend下条目编号或名称(返回上层菜单,请输入b): 2
=====

后端服务器(mysqlserver)信息如下:
id      name      address      weight      maxconn
1      mysql1      10.1.1.110:3306      20      300
2      mysql2      10.1.1.120:3306      30      200
3      wangkai      12.12.12.120:3306      30      40
4      kevin      10.10.10.10:2222      30      4000
5      sky      192.168.100.33:8888      30      20000

=====
请输入删除的server的编号(返回上层菜单,请输入b): 

```

7、输入b可以返回上层菜单，输入5，即可退出程序。

八、针对python2.7和python3.4的代码区别：

- 1、print的使用。
- 2、python2.7下使用raw_input，python3.4下使用input。

做人一定要靠自己

分类：python作业

标签: [python作业](#)

好文要顶

关注我

收藏该文



凯-子

关注 - 14

粉丝 - 40

+加关注

1

0

(请您对文章做出评价)

« 上一篇: [python学习笔记（第三天）](#)

» 下一篇: [git操作笔记](#)

posted @ 2016-01-22 00:16 [凯-子](#) 阅读(56) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

昵称:

评论内容:



提交评论

[注销](#) [订阅评论](#)

[Ctrl+Enter快捷键提交]

【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【阿里云SSD云盘】速度行业领先



最新IT新闻:

- 我13岁那年是如何对程序员失去信任的
 - 三星发布邀请函，Galaxy S7开箱见客就在2/21
 - 用4天时间欺骗全世界：一款虚构出来的约架App为何能红遍网络？
 - 软件行业的先锋者Ed Yourdon逝世
 - 霍金最新理论"黑洞软毛发"引争议：专家意见分歧
- » 更多新闻...

最新知识库文章:

- 消息队列（Message Queue）基本概念
 - 编程每一天（Write Code Every Day）
 - Docker简介
 - Docker简明教程
 - Git协作流程
- » 更多知识库文章...

Copyright ©2016 凯-子