middleware0.4运行说明

1. 启动方物服务器：#swift-init main restart

启动方物服务器以开启存储服务。

2. 启动中间件服务器：#python mwserver.py

客户端与中间件通过socket通信，中间件服务器在本机，在守护进程下运行，关机前不需要再次启动。

注：需要安装加密库，安装方法见4.(7)。

3. 客户端通过import mwclient模块，调用sendmsg\_mw(request)函数来与中间件通信，request的具体形式即为以下的服务器接口。

其中testclient.py是一个客户端调用中间件模块发送消息的示例。

中间件测试时，服务器ip是10.131.1.35，湖大需修改服务器ip(也可将服务器放在本地，为localhost)，但端口443不可改变。

sendmsg\_mw(request)中request参数的具体格式如下：

(1).关于用户容器的三个接口。

用户可以自行创建和删除容器，本说明中所有的文件和文件夹都在test容器中作为示例，用户可自行创建并修改容器。

AUTH\_glfs用户默认有四个基础容器：包括user 存放用户文件和目录，recycle 回收站功能，segments 存放临时文件，versions 存放用户文件历史。

1).列出所有容器及其信息

request = '-X GET -H "X-Auth-Token: " http://10.131.1.35:443/v1/AUTH\_glfs'

列出AUTH\_glfs用户的所有容器。

2).创建容器

request = '-X PUT -H "X-Auth-Token: " http://10.131.1.35:443/v1/AUTH\_glfs/test'

创建一个test容器。

3).删除容器

request = '-i -X DELETE -H "X-Auth-Token: " http://10.131.1.35:443/v1/AUTH\_glfs/test'

删除test容器，返回http的head信息。

(2).桌面客户端所需要实现的接口功能如下：

1).上传文件

request = '-X PUT -H "X-Auth-Token: " -T /home/user/mytest/test.py "http://10.131.250.153:443/v1/AUTH\_glfs/test/dir5/test.py?fype=f&type=NORMAL"'

其中'/home/user/mytest/test.py'是待上传文件的绝对路径，'/test/dir5/test.py'是上传至服务器的路径(注意test.py后面的空格)，test是容器，dir5是目录，test.py是上传的文件名。

由于目前接口不完整，所以默认是通过加密压缩上传。压缩和加密的临时文件存放在/tmp目录。

2).下载文件和右击文件打开

request = '-X GET -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/test/test.py?fype=f&type=NORMAL" -o /home/user/mytest/test.py'

其中'/test/test.py'服务器上待下载的文件路径，'/home/user/mytest/test.py'是客户端需要保存至本地的绝对路径。

中间件会处理下载后的文件，将解密和解压缩之后的文件放到客户端指定路径。临时文件存放在/tmp目录。

3).查看容器中的文件和文件夹

request = '-X GET -H "X-Auth-Token: " http://10.131.250.153:443/v1/AUTH\_glfs/test'

其中test是创建的容器，返回是json格式的字符串。客户端需要在登录时调用该接口以列出全部文件。

4).右击获取文件属性

request = '-i -X HEAD -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/test/test.py"'

其中'/test/test.py'是服务器上的文件路径，可以查看test.py的属性，以http head的形式返回。

5).删除单个文件或文件夹

request = '-i -X DELETE -H "X-Auth-Token: " http://10.131.250.153:443/v1/AUTH\_glfs/test/test.py'

其中'/test/test.py'是所需要删除的文件在服务器上的路径。

该接口也可删除文件夹(如request = '-i -X DELETE -H "X-Auth-Token: " http://10.131.250.153:443/v1/AUTH\_glfs/test/dir2')

6).批量删除文件和文件夹

request = '-i -X POST -d \'{"list":[{"path":"/test/version2.txt","ftype":"f"},{"path":"/test/dir5","ftype":"d"}]}\' -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/batch?op=DELETE&type=NORMAL"'

实现了删除文件'/test/version2.txt'和目录'/test/dir5'的功能。

7).创建目录

request = '-i -X PUT -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/test/dir5/subdir5?op=MKDIRS&ftype=d&type=NORMAL"'

实现了在'/test/dir5'目录中创建subdir5子目录的功能。

8).重命名目录

request = '-i -X PUT -H "X-Auth-Token: " -H "Destination: /test/dir6" "http://10.131.250.153:443/v1/AUTH\_glfs/test/dir5?op=RENAME&ftype=d&type=NORMAL"'

实现了将'/test/dir5'目录重命名为'/test/dir6'的功能。

9).查询正在传输文件的状态信息

i. 通过import mwstatus模块，调用其中的get\_status(logpath)函数来获取json格式的文件传输状态。

ii. 其中，返回信息主要包括八个并且以json的形式给出：

TotalPercent //当前下载文件的百分比

Current Speed //当前下载文件的速度

Total //下载文件的大小

Received //接收到文件的大小

AverageDload //平均下载速度

SpeedUpload //平均上传速度

Time Spent //总的时间花费

Time Left //下载剩余时间

返回示例：（下载一个69.4M完成时的进度信息反馈）

{"TotalPercent": "100","Current Speed":"11.2M","Total":"69.4M","Received":"69.4M ","AverageDload":"11.0M","SpeedUpload":"0","Time Spent":"0:00:06 ","Time Left":"--:--:--"}

iii.get\_status(logpath)函数中的logpath参数是由该正在传输的文件绝对路径经过取md5值获得。

如:正在传输的文件绝对路径 path='/home/user/mytest/test.py'

则 logpath = '/tmp/' + hashlib.md5(path).hexdigest() + '.log'

注：需要import hashlib模块

4. 注意：

(1).request参数中"X-Auth-Token: "，冒号后面只要不为空即可，目前软件所还没有提供验证功能，本说明中所有的’X-Auth-Token’的值为空格。

(2).request参数中’AUTH\_glfs’代表token\_id为glfs，目前方物还没有提供验证功能，默认所有token\_id为glfs。

(3).由于该版本是中间件单线程版本，request一次只能发送一条指令，后续将会对此进行改进。

(4). request参数中部分含’-i’的解释：对于那些有返回json的服务器接口(如上传文件接口)中间件直接去掉了'-i'开关，这样中间件可以直接将json返回给客户端，而对于那些除了http-head信息再没有返回信息的服务器接口(如获取文件属性、删除文件接口)，中间件开启'-i'开关，把http的head信息返回给客户端。

(5).如果上传文件时没有返回json，只返回了’<’，即上传文件失败。原因可能有二：一是上传的文件太大(服务器默认一个用户有1G空间)；二是空间不足，此时需要清空回收站。

(6).清空回收站以解决空间不足的问题

1).清空回收站的文件

request = '-i -X PUT -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/recycle/user?op=RESET&ftype=d"'

2).清空回收站的meta

request = '-i -X PUT -H "X-Auth-Token: " "http://10.131.250.153:443/v1/AUTH\_glfs/recycle/meta?op=RESET&ftype=d"'

(7).加密库的安装(python支持)   
 1).安装支持库：#pip install rsa

2).安装加密库：#dpkg -i libsecrypto.deb

3).确认库以添加：#ls /usr/lib/libsecrypto.so