



## python通过TimedRotatingFileHandler按时间切割日志

### 通过TimedRotatingFileHandler按时间切割日志

线上跑了一个定时脚本，每天生成的日志文件都写在了一个文件中。但是日志信息不可能输出到单一的一个文件中。

原因有二：1.日志文件越来越大会影响系统的性能。2.日志文件格式不够清晰，比如我想看今天的日志，不太方便找到的今天的日志信息(即使对日志输出做了时间提示)

通过设置 `TimedRotatingFileHandler` 进行日志按周(W)、天(D)、时(H)、分(M)、秒(S)切割。

先看一个简单例子：

```
import time
import logging
import os
from logging import handlers
```

```
def _logging(**kwargs):
    level = kwargs.pop('level', None)
    filename = kwargs.pop('filename', None)
    datefmt = kwargs.pop('datefmt', None)
    format = kwargs.pop('format', None)
    if level is None:
        level = logging.DEBUG
    if filename is None:
        filename = 'default.log'
    if datefmt is None:
        datefmt = '%Y-%m-%d %H:%M:%S'
    if format is None:
        format = '%(asctime)s [% (module)s] %(levelname)s [% (lineno)d] %(message)s'
```

[Copy](#)

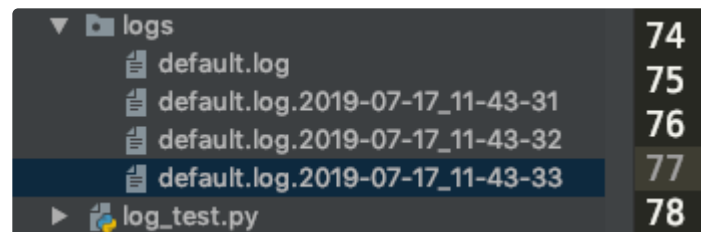


```
log = logging.getLogger(filename)
format_str = logging.Formatter(format, datefmt)
# backupCount 保存日志的数量，过期自动删除
# when 按什么日期格式切分(这里方便测试使用的秒)
th = handlers.TimedRotatingFileHandler(filename=filename, when='S', backupCount=3, encoding='utf-8')
th.setFormatter(format_str)
th.setLevel(logging.INFO)
log.addHandler(th)
log.setLevel(level)
return log

os.makedirs("./logs", exist_ok=True)
logger = _logging(filename='./logs/default.log')

if __name__ == '__main__':
    while True:
        time.sleep(0.1)
        logger.info('哈哈')
```

结果如下:



上述代码可以正常运行，而且也可以生成固定的日志个数，但是有一个问题，生成的日志文件格式是你的 文件名+时间 的格式，没有时间的话默认设置到了秒(这里是按秒切割)

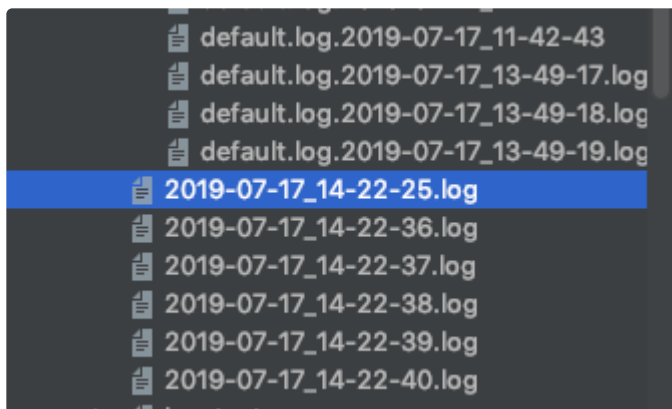
修改日志格式后缀名称:



# 在上述代码中加入

```
def namer(filename):  
    return filename.split('default.')[1]  
th.namer = namer  
# 设置为S, 默认的suffix为 Y-%m-%d_%H-%M-%S  
th.suffix = "%Y-%m-%d_%H-%M-%S.log"  
  
# 为了看的更视觉效果, 可以显示在控制台答应  
cmd = logging.StreamHandler()  
cmd.setFormatter(format_str)  
cmd.setLevel(level)  
log.addHandler(cmd)
```

运行结果:



名字好像可以了, 但是日志好像没有起到自动删除的目的啊, 而且也没在之前的log文件夹了。  
来看看源码:

```
def getFilesToDelete(self):  
    """  
    Determine the files to delete when rolling over.  
    """
```



```
More specific than the earlier method, which just used glob.glob().
"""
dirName, baseName = os.path.split(self.baseFilename)
fileNames = os.listdir(dirName)
result = []
prefix = baseName + "."
plen = len(prefix)
for fileName in fileNames:
    if fileName[:plen] == prefix:
        suffix = fileName[plen:]
        if self.extMatch.match(suffix):
            result.append(os.path.join(dirName, fileName))
if len(result) < self.backupCount:
    result = []
else:
    result.sort()
    result = result[:len(result) - self.backupCount]
return result
```

这是它的删除逻辑，关键是通过 `.` 前面的字段判断是否重复，当有特定的重复数后开始删除。所以问题来了，要么自己去重写源码，要么就只能用 `default.日期.log` 这种格式了。附上平时使用的日志代码

```
import logging
import os
from logging import handlers

def _logging(**kwargs):
    level = kwargs.pop('level', None)
    filename = kwargs.pop('filename', None)
    datefmt = kwargs.pop('datefmt', None)
```

Copy



```
format = kwargs.pop('format', None)
if level is None:
    level = logging.DEBUG
if filename is None:
    filename = 'default.log'
if datefmt is None:
    datefmt = '%Y-%m-%d %H:%M:%S'
if format is None:
    format = '%(asctime)s [%(module)s] %(levelname)s [%(lineno)d] %(message)s'

log = logging.getLogger(filename)
format_str = logging.Formatter(format, datefmt)

def namer(filename):
    return filename.split('default.')[1]

# cmd = logging.StreamHandler()
# cmd.setFormatter(format_str)
# cmd.setLevel(level)
# log.addHandler(cmd)

os.makedirs("./debug/logs", exist_ok=True)
th_debug = handlers.TimedRotatingFileHandler(filename="./debug/" + filename, when='D', backupCount=3,
                                             encoding='utf-8')

# th_debug.namer = namer
th_debug.suffix = "%Y-%m-%d.log"
th_debug.setFormatter(format_str)
th_debug.setLevel(logging.DEBUG)
log.addHandler(th_debug)

th = handlers.TimedRotatingFileHandler(filename=filename, when='D', backupCount=3, encoding='utf-8')
# th.namer = namer
th.suffix = "%Y-%m-%d.log"
th.setFormatter(format_str)
th.setLevel(logging.INFO)
```



```
log.addHandler(th)
log.setLevel(level)
return log
```

```
os.makedirs('./logs', exist_ok=True)
logger = _logging(filename='./logs/default')
```

作者：学一点也是好

出处：<https://www.cnblogs.com/mangM/p/11200987.html>

本站使用「CC BY 4.0」创作共享协议，转载请在文章明显位置注明作者及出处。

分类：[python学习](#)

推荐 0

赞赏

收藏

反对 0

« 上一篇：[Java面向对象入门](#)

» 下一篇：[Python重试模块retrying](#)

posted @ 2019-07-17 14:47 学一点也是好 阅读(1770) 评论(0)

藏

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

