

06/14/17 05:37:10 /home/zz/Documents/TIPE/src/Logger/FileLogger.py

```
1  #!/usr/bin/python3.4
2  # -*-coding:Utf-8 -*
3
4  from .AbstractLogger import AbstractLogger
5  from lib.inherit_docstring import inherit_docstring
6  from src.meta.ABCInheritableDocstringsMeta import ABCInheritableDocstringsMeta
7  from src.Writer.PathManager import PathManager
8
9
10 class FileLogger(AbstractLogger, metaclass=ABCInheritableDocstringsMeta):
11     """Log Generator events into a file """
12
13
14     def onProcessusResume(self, event):
15         self.processus_id = event.processus_id
16         super().onProcessusResume(event)
17
18     def onProcessusStart(self, event):
19         self.processus_id = event.processus_id
20         super().onProcessusStart(event)
21
22
23     @inherit_docstring
24     def write(self, msg):
25         with PathManager.getPath(self.processus_id).with_name('log').open('a') as f:
26             f.write(msg + '\n')
27
28
29     @inherit_docstring
30     def overwrite(self, msg):
31         with PathManager.getPath(self.processus_id).with_name('log').open('r') as f:
32             lines = f.readlines()
33         with PathManager.getPath(self.processus_id).with_name('log').open('w') as f:
34             f.writelines([item for item in lines[:-1]])
35             f.write(msg + '\n')
```