Criando um log simples.

Neste Post vamos criar uma classe Log, criei essa classe pra quem quiser adicionar um log a mais além do log do Behave, é bem simple usa a classe logging do Python. Criamos uma classe pra definir as configuras pré-definidas do nosso log, adicionamos a data no formato brasileiro.

 O FileHandler grava todas as mensagens no arquivo. Isso pode ser um único arquivo ou um conjunto de arquivos girados. Se os arquivos girados forem usados, cada arquivo será preenchido até um determinado limite de tamanho, após o qual um novo arquivo será criado.

**1. Primeiro criamos o arquivo logging.py;**

1. import logging
3. from os import path, getcwd
4. class Logger(object):
5. def \_\_init\_\_(self, name="mylog.log"):
6. # pega o caminho do projeto.
7. path\_app = getcwd()
8. # remove o .log que jah eh gerado pela funcao.
9. name = name.replace('.log','')
10. # log\_namespace can be replaced with your namespace - 'log\_namespace.%s' % name
11. logger = logging.getLogger(name)
12. # Defina o nível de log. Nível deve ser um int ou um str.
13. logger.setLevel(logging.DEBUG)
14. if not logger.handlers:
15. file\_name = path.join(path\_app, 'logs','%s.log' % name)
16. print(file\_name)
17. handler = logging.FileHandler(file\_name)
18. # Define uma formatacao para o log adicionando a data no formato %d/%m/%Y %I:%M:%S %p.
19. # Voce pode remover a data e o Nivel e deixar somente a %(message)s
20. formatter = logging.Formatter('%(asctime)s %(levelname)s: %(message)s',
21. datefmt='%d/%m/%Y %I:%M:%S %p')
22. handler.setFormatter(formatter)
23. handler.setLevel(logging.DEBUG)
24. logger.addHandler(handler)
25. self.\_logger = logger
26. def get(self):
27. return self.\_logger

**2. No arquivo de teste adicionamos os comandos abaixo:**

# Importando a Classe Logger.

from logging import Logger

**# No inicio do Teste Instanciamos a classe Logger passando o nome do Log, fiz dessa forma para que quiser passar um novo log, senão passar nada ele vai definir por padrão  name="mylog.log".**

1. logger = Logger().get()
2. logger.info('Test Propert')

ou

1. logger = Logger("mylog3.log").get()
2. logger.info('Test Propert')

**# Apos o Assert podemos adicionar o resultado do Teste:**

logger.info('\n01 - Passed - Propert\_test.py')

**3. Por fim criamos a pasta /logs no nosso projeto.**