

Python之configparser模块详解和使用

目录

- 1 configparser安装
- 2 configparser简介
- 3 表示方法
- 4 configparser详细使用
 - 4.1 对象初始化
 - 4.2 获取所有的sections
 - 4.3 获取所有的sections对应的options
 - 4.4 read方法和get方法，获取指定section下的option值
 - 4.5 items方法，获取指点section所用配置信息
 - 4.6 set和write方法，修改某个option的值
 - 4.7 has_section和has_option方法
 - 4.8 add_section方法，添加section和option
 - 4.9 remove_section和remove_option方法，删除section和option
- 5 configparser常见的异常
- 6 本文所有源码

1 configparser安装

```
1 | pip3 install configparser
```

2 configparser简介

- 用来读取配置文件的python包；
- 一般做自动化测试的时候，会使用到这个模块，用来封装一些常量。比如数据库、邮件、用户名密码、项目常量等等；
- 这个使用根据个人喜好和项目来确定，不一定一定要使用这个模块，也可以使用其它的方法做配置，比如py文件、xml、excel、yaml、json等等。
- configparser源码大约1360行左右，通读源码可有效了解该模块的使用。本文只做简单介绍常用的方法。

3 表示方法

- 新建一个名为 `config.conf` 文件；
- 写入如下数据，格式如下：

```
1 | [mysqldb]
2 | sql_host = 127.0.0.1
3 | sql_port = 3699
4 | sql_user = root
5 | sql_pass = 123456
6 |
7 | [mailinfo]
8 | name = NoamaNelson
9 | passwd = 123456
10 | address = 123456@qq.com
```

4 configparser详细使用

4.1 对象初始化

- 写入如下代码，把对象初始化。

```
1 | # -*- coding:utf-8 -*-
2 | # 作者: NoamaNelson
3 | # 日期: 2021/11/19
4 | # 文件名称: conf.py
5 | # 作用: configparser模块的使用
6 | # 联系: VX(NoamaNelson)
7 | # 博客: https://blog.csdn.net/NoamaNelson
8 |
9 | import configparser
10 | import os
11 |
12 |
13 | class Conf:
14 |     def __init__(self):
15 |
```

```

16         self.conf = configparser.ConfigParser()
17         self.root_path = os.path.dirname(os.path.abspath(__file__))
18         self.f = os.path.join(self.root_path + "/config.conf")
        self.conf.read(self.f)

```

4.2 获取所有的sections

```

1         def read_sections(self):
2             print(f"1、获取所有的sections:{self.conf.sections()}")
3
4         if __name__ == "__main__":
5             aa = Conf()
6             aa.read_sections()

```

- 结果为:

```

1 D:\Python37\python.exe F:/python_study/conf.py
2 1、获取所有的sections:['mysqlldb', 'mailinfo']

```

4.3 获取所有的sections对应的options

```

1         def read_options(self, s1, s2):
2             print(f"2、获取mysqlldb所有的options:{self.conf.options(s1)}")
3             print(f"3、获取mailinfo所有的options:{self.conf.options(s2)}")
4
5         if __name__ == "__main__":
6             aa = Conf()
7             aa.read_sections()
8             aa.read_options("mysqlldb", "mailinfo")

```

- 结果为:

```

1 D:\Python37\python.exe F:/python_study/conf.py
2 1、获取所有的sections:['mysqlldb', 'mailinfo']
3 2、获取mysqlldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass']
4 3、获取mailinfo所有的options:['name', 'passwd', 'address']

```

4.4 read方法和get方法，获取指定section下的option值

- `read` 是读取配置文件，如 `self.conf.read(self.f)` 是读取指定 `config.conf` 文件;
- `get` 是获取具体的值;

```

1         def read_conf(self, m, n):
2             name = self.conf.get(m, n) # 获取指定section的option值
3             print(f"4、获取指定section:{m}下的option: {n}的值为{name}")
4
5         if __name__ == "__main__":
6             aa = Conf()
7             aa.read_sections()
8             aa.read_options("mysqlldb", "mailinfo")
9             aa.read_conf("mysqlldb", "sql_host")

```

- 结果为:

```

1 D:\Python37\python.exe F:/python_study/conf.py
2 1、获取所有的sections:['mysqlldb', 'mailinfo']
3 2、获取mysqlldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass']
4 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 4、获取指定section:mysqlldb下的option: sql_host的值为127.0.0.1

```

4.5 items方法，获取指定section所用配置信息

```

1         def get_items(self, m, n):
2             print(f"5、获取section:{m}下的配置信息为: {self.conf.items(m)}")
3             print(f"6、获取section:{n}下的配置信息为: {self.conf.items(n)}")
4
5         if __name__ == "__main__":
6             aa = Conf()
7             aa.read_sections()
8             aa.read_options("mysqlldb", "mailinfo")
9             aa.read_conf("mysqlldb", "sql_host")
10            aa.get_items("mysqlldb", "mailinfo")

```

- 结果为:

```

1 D:\Python37\python.exe F:/python_study/conf.py
2 1、获取所有的sections:['mysqldb', 'mailinfo']
3 2、获取mysqldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass']
4 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 4、获取指定section:mysqldb下的option: sql_host的值为127.0.0.1
6 5、获取section:mysqldb下的配置信息为: [('sql_host', '127.0.0.1'), ('sql_port', '3699'), ('sql_user', 'root'), ('sql_pass', '123456')]
7 6、获取section:mailinfo下的配置信息为: [('name', 'NoamaNelson'), ('passwd', '123456'), ('address', '123456@qq.com')]

```

4.6 set和write方法, 修改某个option的值

- 如果option不存在则创建它;

```

1 def set_option(self, m, n, s):
2     self.conf.set(m, n, s)
3     self.conf.write(open(self.f, "w"))
4     print(f"7、设置section:{m}下的option:{n}的值为: {s}")
5
6 if __name__ == "__main__":
7     aa = Conf()
8     aa.read_sections()
9     aa.read_options("mysqldb", "mailinfo")
10    aa.read_conf("mysqldb", "sql_host")
11    aa.get_items("mysqldb", "mailinfo")
12    aa.set_option("mysqldb", "sql_name", "游客")

```

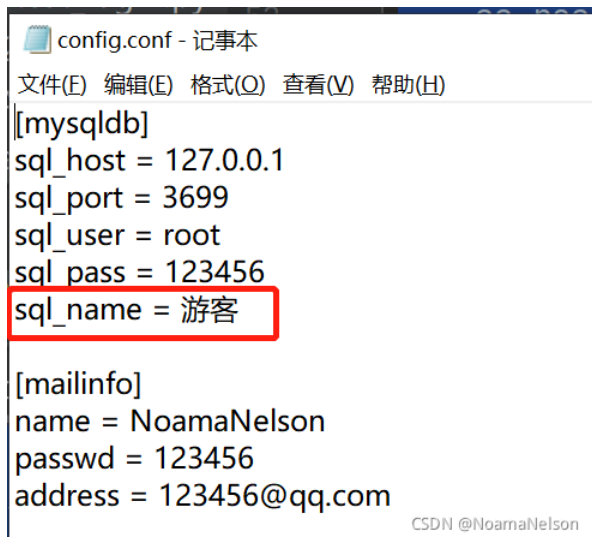
- 结果为:

```

1 D:\Python37\python.exe F:/python_study/conf.py
2 1、获取所有的sections:['mysqldb', 'mailinfo']
3 2、获取mysqldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass']
4 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 4、获取指定section:mysqldb下的option: sql_host的值为127.0.0.1
6 5、获取section:mysqldb下的配置信息为: [('sql_host', '127.0.0.1'), ('sql_port', '3699'), ('sql_user', 'root'), ('sql_pass', '123456')]
7 6、获取section:mailinfo下的配置信息为: [('name', 'NoamaNelson'), ('passwd', '123456'), ('address', '123456@qq.com')]
8 7、设置section:mysqldb下的option:sql_name的值为: 游客

```

- 因为之前的配置文件中没有 sql_name , 所以会新建它, 如果创建的内容存在, 直接修改对应的值;



```

config.conf - 记事本
文件(E) 编辑(E) 格式(O) 查看(V) 帮助(H)

[mysqldb]
sql_host = 127.0.0.1
sql_port = 3699
sql_user = root
sql_pass = 123456
sql_name = 游客

[mailinfo]
name = NoamaNelson
passwd = 123456
address = 123456@qq.com

CSDN @NoamaNelson

```

4.7 has_section和has_option方法

- has_section 方法检查对应的 section 是否存在;
- has_option 方法检查对应的 option 是否存在;

```

1 def has_s_o(self, s, o):
2     print(f"8、检查section: {s}是否存在: {self.conf.has_section(s)}")
3     print(f"9、检查section: {s}下的option: {o}是否存在: {self.conf.has_option(s, o)}")
4
5 if __name__ == "__main__":
6     aa = Conf()

```

```

7 | aa.read_sections()
8 | aa.read_options("mysqldb", "mailinfo")
9 | aa.read_conf("mysqldb", "sql_host")
10 | aa.get_items("mysqldb", "mailinfo")
11 | aa.set_option("mysqldb", "sql_name", "游客")
12 | aa.has_s_o("mysqldb", "sql_name")

```

- 结果为:

```

1 | D:\Python37\python.exe F:/python_study/conf.py
2 | 1、获取所有的sections:['mysqldb', 'mailinfo']
3 | 2、获取mysqldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass', 'sql_name']
4 | 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 | 4、获取指定section:mysqldb下的option: sql_host的值为127.0.0.1
6 | 5、获取section:mysqldb下的配置信息为: [('sql_host', '127.0.0.1'), ('sql_port', '3699'), ('sql_user', 'root'), ('sql_pass', '123456'), ('sql_name', '游客')]
7 | 6、获取section:mailinfo下的配置信息为: [('name', 'NoamaNelson'), ('passwd', '123456'), ('address', '123456@qq.com')]
8 | 7、设置section:mysqldb下的option:sql_name的值为: 游客
9 | 8、检查section: mysqldb是否存在: True
10 | 9、检查section: mysqldb下的option: sql_name是否存在: True

```

4.8 add_section方法, 添加section和option

- add_section: 添加 section;

- 添加前:

```

[mysqldb]
sql_host = 127.0.0.1
sql_port = 3699
sql_user = root
sql_pass = 123456
sql_name = 游客

```

```

[mailinfo]
name = NoamaNelson
passwd = 123456
address = 123456@qq.com

```

CSDN @NoamaNelson

```

1 | def add_s_o(self, s, o, v):
2 |     if not self.conf.has_section(s):
3 |         self.conf.add_section(s)
4 |         print(f"10、添加新的section为{s}")
5 |     else:
6 |         print(f"10、添加新的section为{s}已经存在, 无需添加!")
7 |     if not self.conf.has_option(s, o):
8 |         self.conf.set(s, o, v)
9 |         print(f"11、要添加的option为{o}, 值为{v}")
10 |    else:
11 |        print(f"11、要添加的option为{o}, 值为{v}, 已经存在, 无需添加!")
12 |    self.conf.write(open(self.f, "w"))
13 |
14 | if __name__ == "__main__":
15 |     aa = Conf()
16 |     aa.read_sections()
17 |     aa.read_options("mysqldb", "mailinfo")
18 |     aa.read_conf("mysqldb", "sql_host")
19 |     aa.get_items("mysqldb", "mailinfo")
20 |     aa.set_option("mysqldb", "sql_name", "游客")
21 |     aa.has_s_o("mysqldb", "sql_name")
22 |     aa.add_s_o("login", "name", "root")

```

- 添加后:

```

1 | D:\Python37\python.exe F:/python_study/conf.py
2 | 1、获取所有的sections:['mysqldb', 'mailinfo']
3 | 2、获取mysqldb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass', 'sql_name']
4 | 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 | 4、获取指定section:mysqldb下的option: sql_host的值为127.0.0.1
6 | 5、获取section:mysqldb下的配置信息为: [('sql_host', '127.0.0.1'), ('sql_port', '3699'), ('sql_user', 'root'), ('sql_pass', '123456'), ('sql_name', '游客')]
7 |

```

```

8 | 6、获取section:mailinfo下的配置信息为: [('name', 'NoamaNelson'), ('passwd', '123456'), ('address', '123456@qq.com')]
9 | 7、设置section:mysqlbdb下的option:sql_name的值为: 游客
10 | 8、检查section: mysqlbdb是否存在: True
11 | 9、检查section: mysqlbdb下的option: sql_name是否存在: True
12 | 10、添加新的section为login
    | 11、要添加的option为name, 值为root

```

[mysqlbdb]

sql_host = 127.0.0.1

sql_port = 3699

sql_user = root

sql_pass = 123456

sql_name = 游客

[mailinfo]

name = NoamaNelson

passwd = 123456

address = 123456@qq.com

[login]

name = root

CSDN @NoamaNelson

- 再次运行代码, 会提示已经存在:

```

1 | 10、添加新的section为login已经存在, 无需添加!
2 | 11、要添加的option为name, 值为root, 已经存在, 无需添加!

```

4.9 remove_section和remove_option方法, 删除section和option

```

1 | def remove_s_o(self, s, o):
2 |     if self.conf.has_section(s):
3 |         self.conf.remove_section(s)
4 |         print(f"12、删除section:{s}==OK!")
5 |     else:
6 |         print(f"12、要删除的section:{s}不存在, 不用删除!")
7 |     if self.conf.has_option(s, o):
8 |         self.conf.remove_option(s, o)
9 |         print(f"13、删除section: {s}下的option: {o}==OK!")
10 |    else:
11 |        print(f"13、要删除的section: {s}下的option: {o}不存在, 不用删除!")
12 |
13 | if __name__ == "__main__":
14 |     aa = Conf()
15 |     aa.read_sections()
16 |     aa.read_options("mysqlbdb", "mailinfo")
17 |     aa.read_conf("mysqlbdb", "sql_host")
18 |     aa.get_items("mysqlbdb", "mailinfo")
19 |     aa.set_option("mysqlbdb", "sql_name", "游客")
20 |     aa.has_s_o("mysqlbdb", "sql_name")
21 |     aa.add_s_o("login", "name", "root")
22 |     aa.remove_s_o("login", "name")

```

- 结果为:

```

1 | D:\Python37\python.exe F:/python_study/conf.py
2 | 1、获取所有的sections:['mysqlbdb', 'mailinfo', 'login']
3 | 2、获取mysqlbdb所有的options:['sql_host', 'sql_port', 'sql_user', 'sql_pass', 'sql_name']
4 | 3、获取mailinfo所有的options:['name', 'passwd', 'address']
5 | 4、获取指定section:mysqlbdb下的option: sql_host的值为127.0.0.1
6 | 5、获取section:mysqlbdb下的配置信息为: [('sql_host', '127.0.0.1'), ('sql_port', '3699'), ('sql_user', 'root'), ('sql_pass', '123456'), ('s
7 | 6、获取section:mailinfo下的配置信息为: [('name', 'NoamaNelson'), ('passwd', '123456'), ('address', '123456@qq.com')]
8 | 7、设置section:mysqlbdb下的option:sql_name的值为: 游客
9 | 8、检查section: mysqlbdb是否存在: True
10 | 9、检查section: mysqlbdb下的option: sql_name是否存在: True
11 | 10、添加新的section为login已经存在, 无需添加!
12 | 11、要添加的option为name, 值为root, 已经存在, 无需添加!
13 | 12、删除section:login==OK!
14 | 13、要删除的section: login下的option: name不存在, 不用删除!

```

5 configparser常见的异常

异常	描述
ConfigParser.Error	所有异常的基类
ConfigParser.NoSectionError	指定的section没有找到
ConfigParser.DuplicateSectionError	调用add_section() 时， section名称已经被使用
ConfigParser.NoOptionError	指定的参数没有找到
ConfigParser.InterpolationError	当执行字符串插值时出现问题时， 出现异常的基类
ConfigParser.InterpolationDepthError	当字符串插值无法完成时， 因为迭代次数超过了最大的范围， 所以无法完成。InterpolationError的子类
InterpolationMissingOptionError	当引用的选项不存在时， 会出现异常。InterpolationError的子类
ConfigParser.InterpolationSyntaxError	当产生替换的源文本不符合所需的语法时， 就会出现异常。InterpolationError的子类。
ConfigParser.MissingSectionHeaderError	当试图解析一个没有分段标题的文件时， 会出现异常。
ConfigParser.ParsingError	当试图解析文件时发生错误时， 会出现异常
ConfigParser.MAX_INTERPOLATION_DEPTH	当raw参数为false时， get()的递归插值的最大深度。这只适用于ConfigParser类

6 本文所有源码

```
1  # -*- coding:utf-8 -*-
2  # 作者: NoamaNelson
3  # 日期: 2021/11/19
4  # 文件名称: conf.py
5  # 作用: configparser模块的使用
6  # 联系: VX(NoamaNelson)
7  # 博客: https://blog.csdn.net/NoamaNelson
8
9  import configparser
10 import os
11
12
13 class Conf:
14     def __init__(self):
15         self.conf = configparser.ConfigParser()
16         self.root_path = os.path.dirname(os.path.abspath(__file__))
17         self.f = os.path.join(self.root_path + "/config.conf")
18         self.conf.read(self.f)
19
20     def read_sections(self):
21         print(f"1、获取所有的sections:{self.conf.sections()}")
22
23     def read_options(self, s1, s2):
24         print(f"2、获取mysqlpdb所有的options:{self.conf.options(s1)}")
25         print(f"3、获取mailinfo所有的options:{self.conf.options(s2)}")
26
27     def read_conf(self, m, n):
28         name = self.conf.get(m, n) # 获取指定section的option值
29         print(f"4、获取指定section:{m}下的option: {n}的值为{name}")
30
31     def get_items(self, m, n):
32         print(f"5、获取sectoion:{m}下的配置信息为: {self.conf.items(m)}")
33         print(f"6、获取sectoion:{n}下的配置信息为: {self.conf.items(n)}")
34
35     def set_option(self, m, n, s):
36         self.conf.set(m, n, s)
37         self.conf.write(open(self.f, "w"))
38         print(f"7、设置setion:{m}下的option:{n}的值为: {s}")
39
40     def has_s_o(self, s, o):
41         print(f"8、检查section: {s}是否存在: {self.conf.has_section(s)}")
42         print(f"9、检查section: {s}下的option: {o}是否存在: {self.conf.has_option(s, o)}")
43
44     def add_s_o(self, s, o, v):
45         if not self.conf.has_section(s):
46             self.conf.add_section(s)
47             print(f"10、添加新的section为{s}")
48         else:
49             print(f"10、添加新的section为{s}已经存在，无需添加！")
50         if not self.conf.has_option(s, o):
51             self.conf.set(s, o, v)
52             print(f"11、要添加的option为{o}， 值为{v}")
53         else:
```

```
54         print(f"11、要添加的option为{o}, 值为{v}, 已经存在, 无需添加! ")
55     self.conf.write(open(self.f, "w"))
56
57     def remove_s_o(self, s, o):
58         if self.conf.has_section(s):
59             self.conf.remove_section(s)
60             print(f"12、删除section:{s}==OK!")
61         else:
62             print(f"12、要删除的section:{s}不存在, 不用删除! ")
63         if self.conf.has_option(s, o):
64             self.conf.remove_option(s, o)
65             print(f"13、删除section: {s}下的option: {o}==OK!")
66         else:
67             print(f"13、要删除的section: {s}下的option: {o}不存在, 不用删除! ")
68
69
70 if __name__ == "__main__":
71     aa = Conf()
72     aa.read_sections()
73     aa.read_options("mysqldb", "mailinfo")
74     aa.read_conf("mysqldb", "sql_host")
75     aa.get_items("mysqldb", "mailinfo")
76     aa.set_option("mysqldb", "sql_name", "游客")
77     aa.has_s_o("mysqldb", "sql_name")
78     aa.add_s_o("login", "name", "root")
79     aa.remove_s_o("login", "name")
```