ugen.py

Client: PANTHEON.tech
Author: Marek Paulik
Date: 2024-06-16 16:26

Command line tool for generating usernames. Input data are provided from one or more formatted input files. Processed data are formatted and stored in specified output file.

Disclaimer: For using test.py, please set *usecase* variable to "test" in *FormatCli* class (ugen.py: line 18) for correct formatting of CLI output in this report file. For use in production, set the *usecase* variable to "production". If set to "test" in production, script will work as intended, only CLI output will be less fancy. :)

Usage: Input files must be named *input_file1.txt, input_file2.txt, … input_fileN.txt* encoded in utf-8. Format of the data is *ID, Forename, Middle Name (optional), Surname, Department* separated by colon.

Example input data line:
```
1234:Jozef:Miloslav:Hurban:Legal
```

Script is validating input data. If *ID* is not an integer, if *Forename* or *Surname* is missing, it will skip the line and report it after running.
If *Department* is missing, script will process the line, but also report it after running.

To run the script you have to provide output file using *-o* or *- -output* flag and input file(s) using positional arguments.

Example CLI command:
```
python3 ugen.py -o output_file.txt input_file1.txt input_file2.txt input_file3.txt
```

Running script with *-h* or *- -help* flag will display help.
If required arguments are missing or incorrect (wrong filepath, missing input file…) it will display error and help and exit.

Case 1

Running script without arguments.

CLI command:

```
python3 ugen.py
```

```
!!! ERROR !!!
the following arguments are required: INPUT FILE(S), -o/--output
!!! ERROR !!!


usage: ugen.py [-h] -o OUTPUT FILE INPUT FILE(S) [INPUT FILE(S) ...]

Generate list of usernames in single output file based on user info from
one
or more input files.

positional arguments:
  INPUT FILE(S)          filename(s) of user info file(s)

options:
  -h, --help             show this help message and exit
  -o OUTPUT FILE, --output OUTPUT FILE
                         specify the name of the output file
```

Case 2

Running script with valid arguments and correctly formatted input files.

Input file content:
```
1234:Jozef:Miloslav:Hurban:Legal
4567:Milan:Rastislav:Stefanik:Defence
4563:Jozef::Murgas:Development
```

CLI command:
```
python3 ugen.py -o test/test_output.txt test/test_input.txt
```

CLI Output:
```
3 lines processed.
0 lines skipped.
```

Output file content:
```
1234:jmhurban:Jozef:Miloslav:Hurban:Legal
4567:mrstefan:Milan:Rastislav:Stefanik:Defence
4563:jmurgas:Jozef::Murgas:Development
```

Case 3

Script is handling length of usernames (8 chars) and duplicate usernames (adding number to username).

Input file content:
```
4567:Milan:Rastislav:Stefanik:Defence
1111:Pista::Hufnagel:Sales
4563:Pista::Hufnagel:Sales
```

CLI command:
```
python3 ugen.py -o test/test_output.txt test/test_input.txt
```

CLI Output:
```
3 lines processed.
0 lines skipped.
```

Output file content:
```
4567:mrstefan:Milan:Rastislav:Stefanik:Defence
1111:phufnage:Pista::Hufnagel:Sales
4563:phufnage1:Pista::Hufnagel:Sales
```

Case 4

Providing ill-formatted data will result in only partial processing.
Possible format errors:
(Lines in input file are corresponding to list below.)
- wrong number of values in line
- using different separator
- ID is not a number
- missing Forename
- missing Surname
- missing Department
(Last line is correctly formatted)

Input file content:
```
1234:Miloslav:Hurban:Legal
7890,Milan,Rastislav,Stefanik,Defence
abcd:Milan:Rastislav:Stefanik:Defence
4563:::Murgas:Development
1111:Pista:::Sales
4563:Pista::Hufnagel:
7890:Milan:Rastislav:Stefanik:Defence
```

CLI command:
```
python3 ugen.py -o test/test_output.txt test/test_input.txt
```

CLI Output:

```
2 lines processed.
5 lines skipped.

ERRORS AND WARNINGS:
ERR: test/test_input.txt, line 1: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 2: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 3: ID is not a number. Line skipped!
ERR: test/test_input.txt, line 4: Forename or Surname is empty. Line
skipped!
ERR: test/test_input.txt, line 5: Forename or Surname is empty. Line
skipped!
WARN: test/test_input.txt, line 6: Department is not specified.
```

Output file content:
```
4563:phufnage:Pista::Hufnagel:
7890:mrstefan:Milan:Rastislav:Stefanik:Defence
```

Case 5

If incorrect data is provided, nothing is processed.

Input file content:
```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse
auctor enim ac velit ultrices euismod. Praesent felis nisi, tempor ac
posuere eu, interdum a magna. Quisque gravida at felis ut viverra.
Vestibulum eget dignissim lorem,
vel viverra nisi. Proin est risus, blandit at arcu vel, rutrum
pellentesque nulla. Aenean accumsan elit libero. Donec ut auctor nisl.
Aenean faucibus nisi sapien, in placerat nulla cursus eget. Nullam euismod
massa vitae dapibus tincidunt.
Sed porttitor
fermentum
blandit.
Praesent aliquet mi vulputate augue pharetra commodo. Maecenas lacus
tortor, tristique sit amet pretium in, fermentum quis augue. Proin
efficitur venenatis ligula ac mattis.
```

CLI command:
```
python3 ugen.py -o test/test_output.txt test/test_input.txt
```

CLI Output:
```
0 lines processed.
6 lines skipped.

ERRORS AND WARNINGS:
ERR: test/test_input.txt, line 1: Wrong format of data. Line skipped!
```

```
ERR: test/test_input.txt, line 2: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 3: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 4: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 5: Wrong format of data. Line skipped!
ERR: test/test_input.txt, line 6: Wrong format of data. Line skipped!
```

Output file content: