APIMicroServiceForWordDocFrequencyCount

By Harsh Verma:

The program is written in python to support both CLI and HTTP API based services. The program uses python 3 or above(3.6).

The program input tested data is present in project directory under data folder with maindata.txt containing the absolute file path of sub directories and other files containing inside data folder having the conversation or usual text data.

1. **WordsCountCLIVersion.py :** File contains the python code and takes input as command line argument of the main file.
2. **WordCountsHTTPAPI.py** : contains HTTP Apis for the program
3. **WordTokanizer.py** : Contain helper class for word map formation
4.Under **test** folder unit test **UnitTest.py** file is present which helps to get trace of some positive/negative test cases

Command Line Execution :

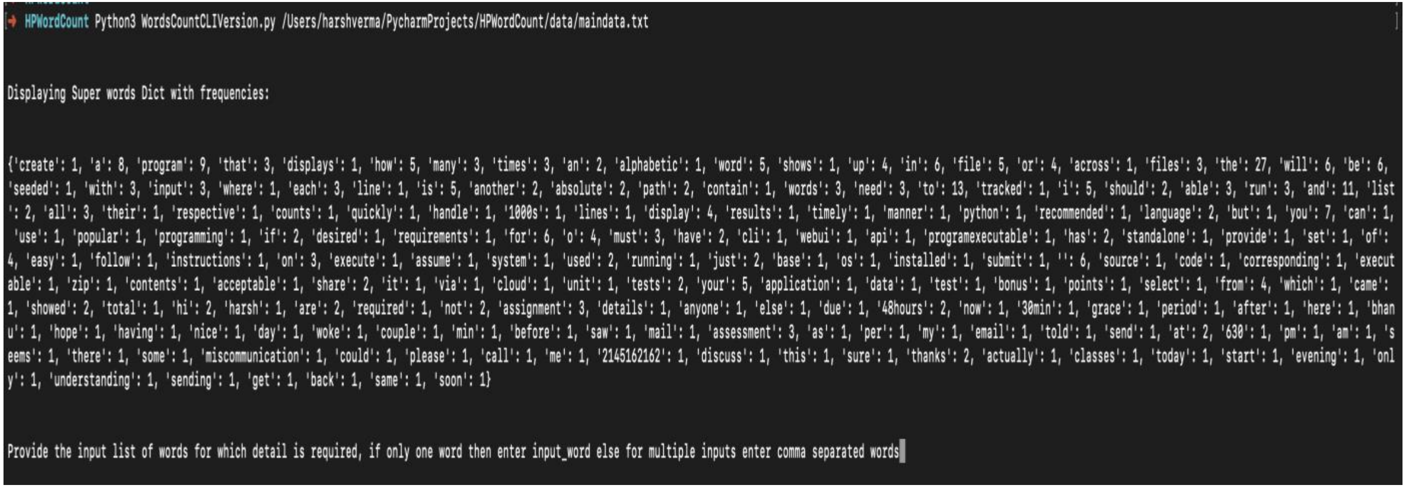
To execute the program pass the main_file_path in the argument :

Eg:

```
Python3 WordsCountCLIVersion.py /Users/harshverma/PycharmProjects/HPWordCount/data/maindata.txt
```

The program will display the output of word list with frequencies across all sub files path which as provided in main file:

Sample output attached :



```
➜ HPWordCount Python3 WordsCountCLIVersion.py /Users/harshverma/PycharmProjects/HPWordCount/data/maindata.txt


Displaying Super words Dict with frequencies:


{'create': 1, 'a': 8, 'program': 9, 'that': 3, 'displays': 1, 'how': 5, 'many': 3, 'times': 3, 'an': 2, 'alphabetic': 1, 'word': 5, 'shows': 1, 'up': 4, 'in': 6, 'file': 5, 'or': 4, 'across': 1, 'files': 3, 'the': 27, 'will': 6, 'be': 6,
'seeded': 1, 'with': 3, 'input': 3, 'where': 1, 'each': 3, 'line': 1, 'is': 5, 'another': 2, 'absolute': 2, 'path': 2, 'contain': 1, 'words': 3, 'need': 3, 'to': 13, 'tracked': 1, 'i': 5, 'should': 2, 'able': 3, 'run': 3, 'and': 11, 'list
': 2, 'all': 3, 'their': 1, 'respective': 1, 'counts': 1, 'quickly': 1, 'handle': 1, '1000s': 1, 'lines': 1, 'display': 4, 'results': 1, 'timely': 1, 'manner': 1, 'python': 1, 'recommended': 1, 'language': 2, 'but': 1, 'you': 7, 'can': 1,
 'use': 1, 'popular': 1, 'programming': 1, 'if': 2, 'desired': 1, 'requirements': 1, 'for': 6, 'o': 4, 'must': 3, 'have': 2, 'cli': 1, 'webui': 1, 'api': 1, 'programexecutable': 1, 'has': 2, 'standalone': 1, 'provide': 1, 'set': 1, 'of':
4, 'easy': 1, 'follow': 1, 'instructions': 1, 'on': 3, 'execute': 1, 'assume': 1, 'system': 1, 'used': 2, 'running': 1, 'just': 2, 'base': 1, 'os': 1, 'installed': 1, 'submit': 1, '': 6, 'source': 1, 'code': 1, 'corresponding': 1, 'execut
able': 1, 'zip': 1, 'contents': 1, 'acceptable': 1, 'share': 2, 'it': 1, 'via': 1, 'cloud': 1, 'unit': 1, 'tests': 2, 'your': 5, 'application': 1, 'data': 1, 'test': 1, 'bonus': 1, 'points': 1, 'select': 1, 'from': 4, 'which': 1, 'came':
1, 'showed': 2, 'total': 1, 'hi': 2, 'harsh': 1, 'are': 2, 'required': 1, 'not': 2, 'assignment': 3, 'details': 1, 'anyone': 1, 'else': 1, 'due': 1, '48hours': 2, 'now': 1, '30min': 1, 'grace': 1, 'period': 1, 'after': 1, 'here': 1, 'bhan
u': 1, 'hope': 1, 'having': 1, 'nice': 1, 'day': 1, 'woke': 1, 'couple': 1, 'min': 1, 'before': 1, 'saw': 1, 'mail': 1, 'assessment': 3, 'as': 1, 'per': 1, 'my': 1, 'email': 1, 'told': 1, 'send': 1, 'at': 2, '630': 1, 'pm': 1, 'am': 1, 's
eems': 1, 'there': 1, 'some': 1, 'miscommunication': 1, 'could': 1, 'please': 1, 'call': 1, 'me': 1, '2145162162': 1, 'discuss': 1, 'this': 1, 'sure': 1, 'thanks': 2, 'actually': 1, 'classes': 1, 'today': 1, 'start': 1, 'evening': 1, 'onl
y': 1, 'understanding': 1, 'sending': 1, 'get': 1, 'back': 1, 'same': 1, 'soon': 1}


Provide the input list of words for which detail is required, if only one word then enter input_word else for multiple inputs enter comma separated words
```

For bonus Task execution in command Line:

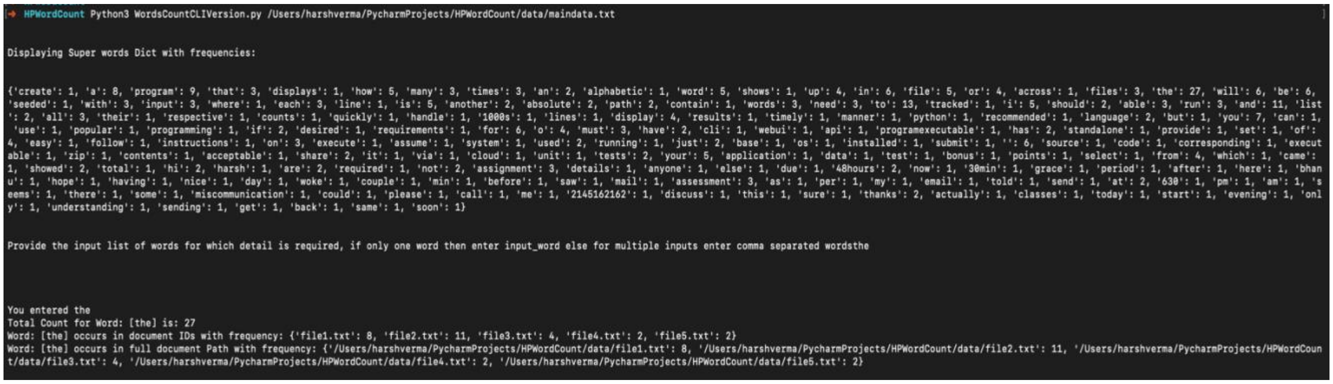
To Execute bonus task pick a word or words from previous word frequencies count list and enter in comma separated input words and Press enter to again start execution of program

It will print required details of the input word/words :
Sample output :
Input provide: the

Example :

Total Count for Word: [the] is: 27
Word: [the] occurs in document IDs with frequency: {'file1.txt': 8, 'file2.txt': 11, 'file3.txt': 4, 'file4.txt': 2, 'file5.txt': 2}
Word: [the] occurs in full document Path with frequency: {'/Users/harshverma/PycharmProjects/HPWordCount/data/file1.txt': 8, '/Users/harshverma/PycharmProjects/HPWordCount/data/file2.txt': 11, '/Users/harshverma/PycharmProjects/HPWordCount/data/file3.txt': 4, '/Users/harshverma/PycharmProjects/HPWordCount/data/file4.txt': 2, '/Users/harshverma/PycharmProjects/HPWordCount/data/file5.txt': 2}

Sample output result image :



```
➜ HPWordCount Python3 WordsCountCLIVersion.py /Users/harshverma/PycharmProjects/HPWordCount/data/maindata.txt


Displaying Super words Dict with frequencies:


{'create': 1, 'a': 8, 'program': 9, 'that': 3, 'displays': 1, 'how': 5, 'many': 3, 'times': 3, 'an': 2, 'alphabetic': 1, 'word': 5, 'shows': 1, 'up': 4, 'in': 6, 'file': 5, 'or': 4, 'across': 1, 'files': 3, 'the': 27, 'will': 6, 'be': 6,
'seeded': 1, 'with': 3, 'input': 3, 'where': 1, 'each': 3, 'line': 1, 'is': 5, 'another': 2, 'absolute': 2, 'path': 2, 'contain': 1, 'words': 3, 'need': 3, 'to': 13, 'tracked': 1, 'i': 5, 'should': 2, 'able': 3, 'run': 3, 'and': 11, 'list
': 2, 'all': 3, 'their': 1, 'respective': 1, 'counts': 1, 'quickly': 1, 'handle': 1, '1000s': 1, 'lines': 1, 'display': 4, 'results': 1, 'timely': 1, 'manner': 1, 'python': 1, 'recommended': 1, 'language': 2, 'but': 1, 'you': 7, 'can': 1,
 'use': 1, 'popular': 1, 'programming': 1, 'if': 2, 'desired': 1, 'requirements': 1, 'for': 6, 'o': 4, 'must': 3, 'have': 2, 'cli': 1, 'webui': 1, 'api': 1, 'programexecutable': 1, 'has': 2, 'standalone': 1, 'provide': 1, 'set': 1, 'of':
4, 'easy': 1, 'follow': 1, 'instructions': 1, 'on': 3, 'execute': 1, 'assume': 1, 'system': 1, 'used': 2, 'running': 1, 'just': 2, 'base': 1, 'os': 1, 'installed': 1, 'submit': 1, '': 6, 'source': 1, 'code': 1, 'corresponding': 1, 'execut
able': 1, 'zip': 1, 'contents': 1, 'acceptable': 1, 'share': 2, 'it': 1, 'via': 1, 'cloud': 1, 'unit': 1, 'tests': 2, 'your': 5, 'application': 1, 'data': 1, 'test': 1, 'bonus': 1, 'points': 1, 'select': 1, 'from': 4, 'which': 1, 'came':
1, 'showed': 2, 'total': 1, 'hi': 2, 'harsh': 1, 'are': 2, 'required': 1, 'not': 2, 'assignment': 3, 'details': 1, 'anyone': 1, 'else': 1, 'due': 1, '48hours': 2, 'now': 1, '30min': 1, 'grace': 1, 'period': 1, 'after': 1, 'here': 1, 'bhan
u': 1, 'hope': 1, 'having': 1, 'nice': 1, 'day': 1, 'woke': 1, 'couple': 1, 'min': 1, 'before': 1, 'saw': 1, 'mail': 1, 'assessment': 3, 'as': 1, 'per': 1, 'my': 1, 'email': 1, 'told': 1, 'send': 1, 'at': 2, '630': 1, 'pm': 1, 'am': 1, 's
eems': 1, 'there': 1, 'some': 1, 'miscommunication': 1, 'could': 1, 'please': 1, 'call': 1, 'me': 1, '2145162162': 1, 'discuss': 1, 'this': 1, 'sure': 1, 'thanks': 2, 'actually': 1, 'classes': 1, 'today': 1, 'start': 1, 'evening': 1, 'onl
y': 1, 'understanding': 1, 'sending': 1, 'get': 1, 'back': 1, 'same': 1, 'soon': 1}


Provide the input list of words for which detail is required, if only one word then enter input_word else for multiple inputs enter comma separated wordsthe



You entered the
Total Count for Word: [the] is: 27
Word: [the] occurs in document IDs with frequency: {'file1.txt': 8, 'file2.txt': 11, 'file3.txt': 4, 'file4.txt': 2, 'file5.txt': 2}
Word: [the] occurs in full document Path with frequency: {'/Users/harshverma/PycharmProjects/HPWordCount/data/file1.txt': 8, '/Users/harshverma/PycharmProjects/HPWordCount/data/file2.txt': 11, '/Users/harshverma/PycharmProjects/HPWordCoun
t/data/file3.txt': 4, '/Users/harshverma/PycharmProjects/HPWordCount/data/file4.txt': 2, '/Users/harshverma/PycharmProjects/HPWordCount/data/file5.txt': 2}
```

2. WordCountsHTTPAPI.py : File contains the code written to support the services via HTTP API. It uses flask to support the HTTP API wrapper.

To start the server execution run file :

python3.6 WordCountsHTTPAPI.py

-> This will start the python flask server and will listen to the request HTTP request in Running on http://0.0.0.0:7000/ (Press CTRL+C to quit) host and port.

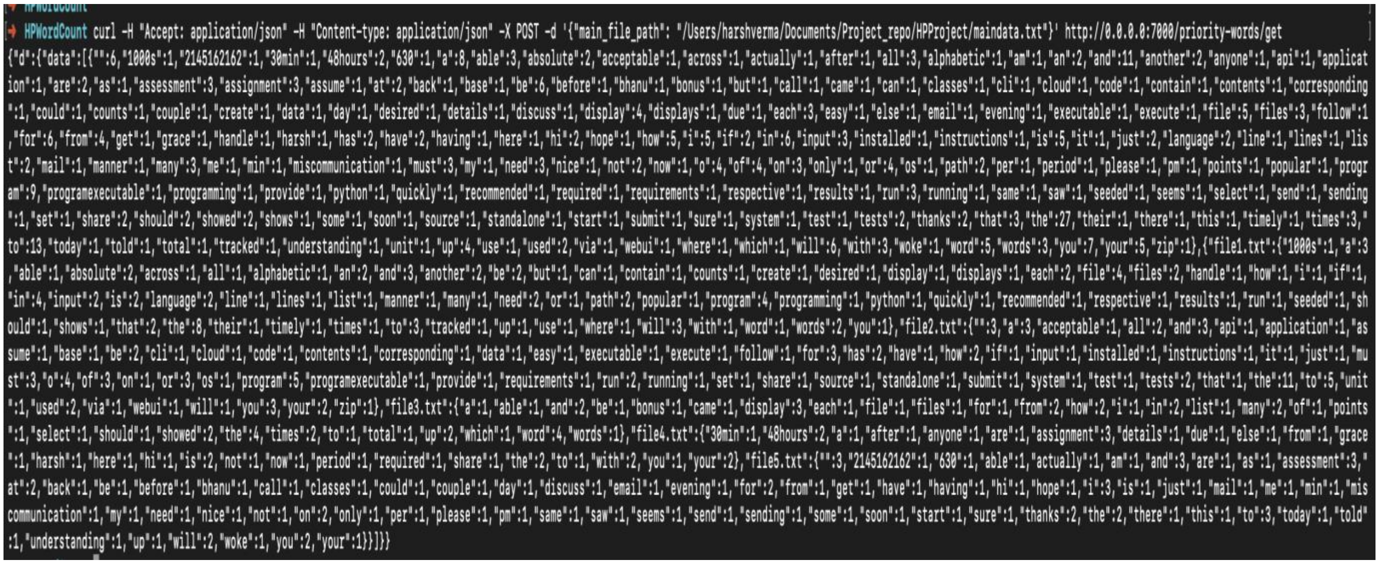
Hit the Curl call using postman or terminal to make a HTTP post call :

1. For Displaying word frequency count :

Example curl for displaying word list with frequencies :
 -> Provide the key parameter as "main_file_path" to give main file full path

curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt"}' http://0.0.0.0:7000/priority-words/get

Sample output result after the curl call to get word frequency:



```
➜ HPWordCount curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt"}' http://0.0.0.0:7000/priority-words/get
{"d":{"data":[{"":6,"1000s":1,"2145162162":1,"30min":1,"48hours":2,"630":1,"a":8,"able":3,"absolute":2,"acceptable":1,"across":1,"actually":1,"after":1,"all":3,"alphabetic":1,"am":1,"an":2,"and":11,"another":2,"anyone":1,"api":1,"applicat
ion":1,"are":2,"as":1,"assessment":3,"assignment":3,"assume":1,"at":2,"back":1,"base":1,"be":6,"before":1,"bhanu":1,"bonus":1,"but":1,"call":1,"came":1,"can":1,"classes":1,"cli":1,"cloud":1,"code":1,"contain":1,"contents":1,"corresponding
":1,"could":1,"counts":1,"couple":1,"create":1,"data":1,"day":1,"desired":1,"details":1,"discuss":1,"display":4,"displays":1,"due":1,"each":3,"easy":1,"else":1,"email":1,"evening":1,"executable":1,"execute":1,"file":5,"files":3,"follow":1
,"for":6,"from":4,"get":1,"grace":1,"handle":1,"harsh":1,"has":2,"have":2,"having":1,"here":1,"hi":2,"hope":1,"how":5,"i":5,"if":2,"in":6,"input":3,"installed":1,"instructions":1,"is":5,"it":1,"just":2,"language":2,"line":1,"lines":1,"lis
t":2,"mail":1,"manner":1,"many":3,"me":1,"min":1,"miscommunication":1,"must":3,"my":1,"need":3,"nice":1,"not":2,"now":1,"o":4,"of":4,"on":3,"only":1,"or":4,"os":1,"path":2,"per":1,"period":1,"please":1,"pm":1,"points":1,"popular":1,"progr
am":9,"programexecutable":1,"programming":1,"provide":1,"python":1,"quickly":1,"recommended":1,"required":1,"requirements":1,"respective":1,"results":1,"run":3,"running":1,"same":1,"saw":1,"seeded":1,"seems":1,"select":1,"send":1,"sending
":1,"set":1,"share":2,"should":2,"showed":2,"shows":1,"some":1,"soon":1,"source":1,"standalone":1,"start":1,"submit":1,"sure":1,"system":1,"test":1,"tests":2,"thanks":2,"that":3,"the":27,"their":1,"there":1,"this":1,"timely":1,"times":3,"
to":13,"today":1,"told":1,"total":1,"tracked":1,"understanding":1,"unit":1,"up":4,"use":1,"used":2,"via":1,"webui":1,"where":1,"which":1,"will":6,"with":3,"woke":1,"word":5,"words":3,"you":7,"your":5,"zip":1},{"file1.txt":{"1000s":1,"a":3
,"able":1,"absolute":2,"across":1,"all":1,"alphabetic":1,"an":2,"and":3,"another":2,"be":2,"but":1,"can":1,"contain":1,"counts":1,"create":1,"desired":1,"display":1,"displays":1,"each":2,"file":4,"files":2,"handle":1,"how":1,"i":1,"if":1,
"in":4,"input":2,"is":2,"language":2,"line":1,"lines":1,"list":1,"manner":1,"many":1,"need":2,"or":1,"path":2,"popular":1,"program":4,"programming":1,"python":1,"quickly":1,"recommended":1,"respective":1,"results":1,"run":1,"seeded":1,"sh
ould":1,"shows":1,"that":2,"the":8,"their":1,"timely":1,"times":1,"to":3,"tracked":1,"up":1,"use":1,"where":1,"will":3,"with":1,"word":1,"words":2,"you":1},"file2.txt":{"":3,"a":3,"acceptable":1,"all":2,"and":3,"api":1,"application":1,"as
sume":1,"base":1,"be":2,"cli":1,"cloud":1,"code":1,"contents":1,"corresponding":1,"data":1,"easy":1,"executable":1,"execute":1,"follow":1,"for":3,"has":2,"have":1,"how":2,"if":1,"input":1,"installed":1,"instructions":1,"it":1,"just":1,"mu
st":3,"o":4,"of":3,"on":1,"or":3,"os":1,"program":5,"programexecutable":1,"provide":1,"requirements":1,"run":2,"running":1,"set":1,"share":1,"source":1,"standalone":1,"submit":1,"system":1,"test":1,"tests":2,"that":1,"the":11,"to":5,"unit
":1,"used":2,"via":1,"webui":1,"will":1,"you":3,"your":2,"zip":1},"file3.txt":{"a":1,"able":1,"and":2,"be":1,"bonus":1,"came":1,"display":3,"each":1,"file":1,"files":1,"for":1,"from":2,"how":2,"i":1,"in":2,"list":1,"many":2,"of":1,"points
":1,"select":1,"should":1,"showed":2,"the":4,"times":2,"to":1,"total":1,"up":2,"which":1,"word":4,"words":1},"file4.txt":{"30min":1,"48hours":2,"a":1,"after":1,"anyone":1,"are":1,"assignment":3,"details":1,"due":1,"else":1,"from":1,"grace
":1,"harsh":1,"here":1,"hi":1,"is":2,"not":1,"now":1,"period":1,"required":1,"share":1,"the":2,"to":1,"with":2,"you":1,"your":2},"file5.txt":{"":3,"2145162162":1,"630":1,"able":1,"actually":1,"am":1,"and":3,"are":1,"as":1,"assessment":3,"
at":2,"back":1,"be":1,"before":1,"bhanu":1,"call":1,"classes":1,"could":1,"couple":1,"day":1,"discuss":1,"email":1,"evening":1,"for":2,"from":1,"get":1,"have":1,"having":1,"hi":1,"hope":1,"i":3,"is":1,"just":1,"mail":1,"me":1,"min":1,"mis
communication":1,"my":1,"need":1,"nice":1,"not":1,"on":2,"only":1,"per":1,"please":1,"pm":1,"same":1,"saw":1,"seems":1,"send":1,"sending":1,"some":1,"soon":1,"start":1,"sure":1,"thanks":2,"the":2,"there":1,"this":1,"to":3,"today":1,"told"
:1,"understanding":1,"up":1,"will":2,"woke":1,"you":2,"your":1}}]}}
```

2. For Displaying details of given word as input(Bonus Task) using HTTP API's:

Maintain the key parameters in curl call :

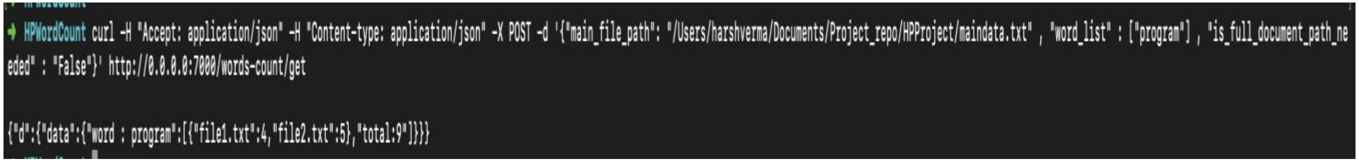
I) main_file_path : Provide the key parameter as "main_file_path" to give main file full path
ii) word_list : provide input word in list to get the details associated with it
iii) If you want to print full path of sub directory file in which the word is stored use key -> is_full_document_path_needed : True

This will provide full paths where the word is present along with its document frequency

Sample curl call :

Curl1 : curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt" , "word_list" : ["program"] , "is_full_document_path_needed" : "False"}' http://0.0.0.0:7000/words-count/get

Sample output :



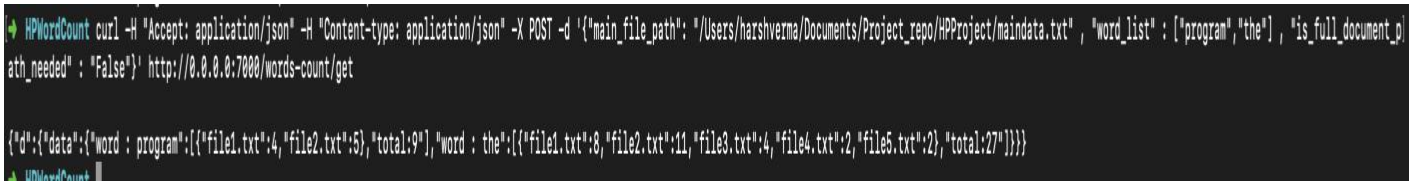
```
➜ HPWordCount curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt" , "word_list" : ["program"] , "is_full_document_path_ne
eded" : "False"}' http://0.0.0.0:7000/words-count/get

{"d":{"data":{"word : program":[{"file1.txt":4,"file2.txt":5},"total:9"]}}}
```

NOTE : For getting information about multiple words in a single curl call: provide input words in list as comma separated under key "word_list"

<u>Sample input and Output :</u>

Curl2 : curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt" , "word_list" : ["program", "true"] , "is_full_document_path_needed" : "False"}' http://0.0.0.0:7000/words-count/get



```
➜ HPWordCount curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d '{"main_file_path": "/Users/harshverma/Documents/Project_repo/HPProject/maindata.txt" , "word_list" : ["program","the"] , "is_full_document_p
ath_needed" : "False"}' http://0.0.0.0:7000/words-count/get

{"d":{"data":{"word : program":[{"file1.txt":4,"file2.txt":5},"total:9"],"word : the":[{"file1.txt":8,"file2.txt":11,"file3.txt":4,"file4.txt":2,"file5.txt":2},"total:27"]}}}
➜ HPWordCount
```

Some Scope of Improvement :

1. Can Also use multithreading to process files in parallel and reduce execution time

Thanks for reading this, Looking forward to connect on same, Please connect to me if any issue occurred while executing program.