

Website Input - Task #2487

Task # 2484 (Closed): Python 3 support

Fix search commands

10/31/2019 06:22 AM - Luke Murphey

Status:	Closed	Start date:	10/31/2019
Priority:	Normal	Due date:	
Assignee:	Luke Murphey	% Done:	100%
Category:	Search Command	Estimated time:	0.00 hour
Target version:	4.5.8		
Description			

Associated revisions

Revision 704 - 11/01/2019 06:55 AM - lmurphey

Making search command work

Reference #2487

Revision 705 - 11/01/2019 06:56 AM - lmurphey

Flxing import

Reference #2487

History

#1 - 11/01/2019 06:33 AM - Luke Murphey

Getting error "command="webscrape", global name 'splunk' is not defined "

Observations:

- This works on 7.31

Questions:

- Does Network Toolkit work?
 - Yes, it does.
 - from splunk.util import normalizeBoolean is in traceroute but it works

#2 - 11/01/2019 06:48 AM - Luke Murphey

```
"""  
This script provides a search command that allows you to perform web-scrapes from the Splunk  
command-line.
```

```
This wires up the WebInput modular input code to a search command so that you can execute  
web-scrape as the search command..
```

```
"""
```

```

import os
import sys

from splunk.util import normalizeBoolean

from web_input import WebInput
from website_input_app.search_command import SearchCommand
from website_input_app.web_scraper import WebScraper

path_to_mod_input_lib = os.path.join(os.path.dirname(os.path.abspath(__file__)), 'modular_input.zip')
if path_to_mod_input_lib not in sys.path:
    sys.path.insert(0, path_to_mod_input_lib)
from modular_input import ModularInput

class WebScraperSearchCommand(SearchCommand):
    """
    The search command takes the arguments provided by the command-line and sends it to the
    modular input functions so that you could you run the input manually.
    """

    def __init__(self, url=None, selector=None, username=None, password=None, timeout=30,
                 name_attributes=None, output_as_mv=True, output_matches_as_mv=None,
                 output_matches_as_separate_fields=False, use_element_name=False, page_limit=1,
                 depth_limit=50, url_filter=None, text_separator=" ", raw_content=False,
                 include_raw_content=None, browser=None, match_prefix=None, user_agent=None,
                 empty_matches=False, empty_value='NULL', authentication_url=None,
                 username_field=None, password_field=None):

        # Note: output_matches_as_mv and include_raw_content are supported for legacy purposes

        SearchCommand.__init__(self, run_in_preview=True, logger_name="web_scrape")

        self.logger.info("Web scraper running against url=%s", url)

    def handle_results(self, results, session_key, in_preview):
        # Output the results
        self.output_results([
            {'see': normalizeBoolean('true')}
        ])

if __name__ == '__main__':
    WebScraperSearchCommand.execute()

```

#3 - 11/01/2019 06:59 AM - Luke Murphey

- *Status changed from New to Closed*

- *% Done changed from 0 to 100*