

Projects in 2019-2021

A new unit test application


A sophisticated data parser


Demonstration

Unit test the data parser

Add subfolder to parser project

 unittest

▼  utc1

 utc1_setup.txt

Write unit test 1

Define:

- startup arguments
- settings file
- sample data
- expected outcome

```
#ut1
##inputargs
-t -p ./assignments
```

```
##files
/assignments/new/assignment_20000101120000.txt
parse ./data.txt ./wfcheck/settings.txt
```

```
/wfcheck/settings.txt
#
\##name
@root
...
```

```
/data.txt
1Title: inleiding
      Dit verhaal werd in 2019 geschreven.
<witregel>
2Title: beschrijving van de hoofdpersonen
      De inspecteur
      de verdachte
<witregel>
3Title: de plot
      Er is belangenverstrengeling in het bouwbedrijf
      Er wordt een complot gesmeed.
...
```

```
##correct_result_stdout
0
<witregel>
<witregel>
```

```
##comment
tekst zonder fouten
NB. exitmessage is een blanco regel
```

Write unit test variant

Small change in original unit test

=>

New unit test !

```
#ut1-d
##files
-+
//3Title: de plot
2Title: de plot

##correct_result_stdout
1
ERROR Discontinuity found at lines
                                4 - 8
                                8 - 12

/data.txt:@root

##comment
order_test
    het titelnummer klopt in deze test niet meer, we hebben twee keer 2Title:
```

Creating variants is too easy!

The Flask edit tool makes life even more easy

The screenshot displays the Flask edit tool interface in a web browser. The browser's address bar shows the URL `localhost:5000/home/edit?ut_name=ut1&ut_name=ut1-d`. The interface is divided into several sections:

- Navigation Bar:** Includes tabs for `Main`, `Overview`, `Edit`, and `Run`.
- File List:** A vertical list on the left side of the `Edit` tab, containing entries from `ut0` to `ut5`. Below this list are controls for `Set` (with `Move`, `Delete`, and `Add` options), `Filter` (with `FF off`), and a search bar containing `ut1 ut1-d`.
- Editor:** The main workspace for editing the file `ut1`. It is divided into three panes:
 - inputargs:** Contains the command `-t -p ./assignments`.
 - files:** Contains the file `/assignments/new/assignment_20000101120000.txt` and its contents, which include a `parse` command and a complex set of regex rules for parsing assignment data.
 - stdout:** Displays the output of the command, showing a `correct result` and a list of lines with their corresponding titles and line numbers.
- Diff View:** A side-by-side comparison of the file `ut1` and its previous version `ut1-d`. The `ut1` pane shows the current state, while the `ut1-d` pane shows the previous state, highlighting changes in the file's content.

Time to unit test!

```
ubuntu@ubuntu-VirtualBox:~$ python3 /home/ubuntu/utis/unittestester/unittestester.py utc1 python3 /home/ubuntu/utis/testwf/test_wellformedness.py -sp -ut ut1-a-ut1-d
INFO  Unittestester started

...

INFO  Parse setupfiles from unittestcollections: ['utc1']
INFO  Following unittests will be started: ['ut1-a', 'ut1-b', 'ut1-c', 'ut1-d']

...

INFO  Create unittest environments and run unittests
INFO  unittest ut1-a started with: /home/ubuntu/utis/testwf/test_wellformedness.py -t -p /home/ubuntu/utis/testwf/unittest/utc1/ut1-a/assignments
INFO  unittest ut1-b started with: /home/ubuntu/utis/testwf/test_wellformedness.py -t -p /home/ubuntu/utis/testwf/unittest/utc1/ut1-b/assignments
INFO  unittest ut1-c started with: /home/ubuntu/utis/testwf/test_wellformedness.py -t -p /home/ubuntu/utis/testwf/unittest/utc1/ut1-c/assignments
INFO  unittest ut1-d started with: /home/ubuntu/utis/testwf/test_wellformedness.py -t -p /home/ubuntu/utis/testwf/unittest/utc1/ut1-d/assignments

INFO  === Results for utc1 ===
INFO      Some unittests were not executed
INFO      All other unittests passed
```


What happened?

(1) Test environments were built:



(2) For each environment =>

- Start the parser
- Capture the output
- Compare with expected result

Finally:

How the parser works

1. Identify title lines in the data with a regular expression

2. Perform tests:

- unique test
Are all titles unique?
- title order test
Check chapter numbers:
incorrect: 1, 2, 2, 4, ...

```
/wfcheck/settings.txt
#
##name
@root

##titletypes
@title ^([0-9]{1,})(Title:\s{1,3})([a-zA-Z0-9]{1}[-a-zA-Z0-9_]{1,})$
@todo ^TODO$

##nontitletypes
@wrongtitle ^([0-9]{1,})\s*(title|TITLE|Title)\s*(:){0,1}\s*.*$

##skip_everything_after
@todo

##count
@title,@todo arg1>0 COND NRALLINES arg1>0
@wrongtitle arg1==0

##uninterrupted_matches

##start_end_title
starttitle @title,@todo

##unique
@title 2

##similarity
@title 1

##os
##title_order
test1
@title 0 ASC_CONTINUOUS intraise ^1$
```

Parser latest version

- Lots of new tests
- Data can also be converted and exported