Projects in 2019-2021

A new unit test application A sophisticated data parser

Demonstration

Unit test the data parser

Add subfolder to parser project



Write unittest 1

Define:

- startup arguments
- settingsfile
- 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
<witregel>
<witregel>
##comment
tekst zonder fouten
NB. exitmessage is een blanco regel
```

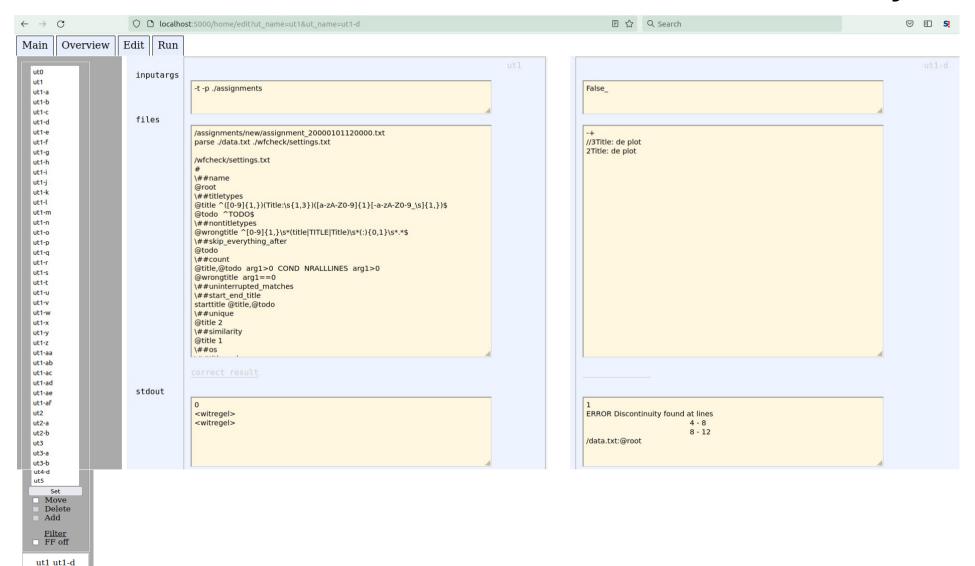
Write unittest variant

Small change in original unittest =>
New unittest!



Creating variants is too easy!

The Flask edit tool makes life even more easy



Time to unittest!

```
ubuntu@ubuntu-VirtualBox:-$ python3 /home/ubuntu/utils/unittester/unittester.py utc1 python3 /home/ubuntu/utils/testwf/test_wellformedness.py -sp -ut ut1-a-ut1-d
INFO Unittester started

...

INFO Parse setupfiles from unittestcollections: ['utc1']
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/utils/testwf/test_wellformedness.py -t -p /home/ubuntu/utils/testwf/unittest/utc1/ut1-a/assignments
INFO unittest ut1-b started with: /home/ubuntu/utils/testwf/test_wellformedness.py -t -p /home/ubuntu/utils/testwf/unittest/utc1/ut1-b/assignments
INFO unittest ut1-c started with: /home/ubuntu/utils/testwf/test_wellformedness.py -t -p /home/ubuntu/utils/testwf/unittest/utc1/ut1-c/assignments
INFO unittest ut1-d started with: /home/ubuntu/utils/testwf/test_wellformedness.py -t -p /home/ubuntu/utils/testwf/unittest/utc1/ut1-d/assignments
INFO == Results for utc1 ===
INFO Some unittests were not executed
All other unittests passed
```

What happened?

(1) test environments were build:



- (2) For each environment =>
 - Start the parser
 - Capture the output
 - Compare with expected result

Finally: How the parser works

- 1. Identify titlelines in 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_\s]{1,})$
##nontitletypes
@wrongtitle ^[0-9]{1,}\s*(title|TITLE|Title)\s*(:){0,1}\s*.*$
##skip everything after
@todo
##count
@title,@todo arg1>0 COND NRALLLINES arg1>0
@wrongtitle arg1==0
##uninterrupted matches
##start end title
starttitle @title,@todo
##unique
Atitle 2
##similarity
@title 1
##os
##title_order
@title 0 ASC_CONTINUOUS intraise ^1$
```

Parser latest version

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