

LINTOL README

Linux Set Up

This Challenge will require the use of python3, with pip3 package installer within a pipenv Virtual Environment

Installing python3

Open your Terminal

Check your current Python Version

```
$ python3 --version
```

If it does not recognise this command, then you are missing python3, in order to install it

```
$ sudo apt-get update
```

```
$ sudo apt-get install python3.7
```

Now repeat the python3 --version command and it should return python3.7

Installing pip3

Open your Terminal

Check your current Python Version

```
$ pip3 --version
```

If it does not recognise this command, then you are missing pip3, in order to install it

```
$ sudo apt-get update
```

```
$ sudo apt-get install python3-pip
```

Now repeat the pip3 --version command and it should return pip version and storage path

Installing pipenv

Open your Terminal

Check if you have pipenv installed

```
$ pipenv
```

If it does not recognise this command, then you are missing pipenv, in order to install it

```
$ pip3 install --user pipenv
```

Now repeat the pipenv command and it should return the command options

Challenge Set Up

Go to GitLab Repo and Fork Project to your own Workspace

<https://gitlab.com/lintol/coding-challenge-2021-exemplar-1/-/tree/master>

Open Terminal

Create a new directory locally on your machine

\$ mkdir "foldername"

```
stephen@debian:~/lintoldemo$ mkdir lintolsetup
stephen@debian:~/lintoldemo$
```

Clone the Forked Repo from Your Workspace into your new local directory

\$ cd "foldername"

\$ git clone "forked repo clone url" .

```
stephen@debian:~/lintoldemo$ mkdir lintolsetup
stephen@debian:~/lintoldemo$ ls
lintolsetup
stephen@debian:~/lintoldemo$ cd lintolsetup
stephen@debian:~/lintoldemo/lintolsetup$

stephen@debian: ~/lintoldemo/lintolsetup
stephen@debian: ~/lintoldemo$ mkdir lintolsetup
stephen@debian: ~/lintoldemo$ ls
lintolsetup
stephen@debian:~/lintoldemo$ cd lintolsetup
stephen@debian:~/lintoldemo/lintolsetup$ git clone https://gitlab.com/username/coding-challenge-2021-exemplar-1.git .
```

Your local directory should now contain the challenge files

\$ ls

```
stephen@debian: ~/lintoldemo
stephen@debian: ~/lintoldemo/lintolsetup$ ls
_init_.py LICENSE out-example-2021-02-01-hansard-plenary.txt output.html output.png processor.py README.
```

When you have the challenge files within your directory, it is time to install the dependencies, to do this we must start a new virtual environment

\$ python3 -m pipenv shell

```
stephen@debian: ~/lintoldemo/lintolsetup$ python3 -m pipenv shell
```

This will build the virtual environment and if successful update the terminal path
(foldername) \$

```
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$
```

Once inside the virtual environment, you can now install the necessary dependencies (any errors rerun command)

(foldername) \$ pip3 install -r requirements.txt

```
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ pip3 install -r requirements.txt
```

To test if you have set up the challenge enviroment successfully, you can now check by running the following command which will output the current demo processor results in the terminal

(foldername) \$

python3 processor.py ./sample_transcripts/out-example-2021-02-01-hansard-plenary.txt

```
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ python3 processor.py ./sample_transcripts/out-example-2021-02-01-hansard-plenary.txt
{"supplementary": [], "item-count": 20, "error-count": 20, "counts": {"errors": 0, "warnings": 2}, [{"processor": "lintol-code-challenge-city-finder:1", "code": "city-cropped-up", "to": {"line": 158, "character": 283}}, {"definition": {"aspect": "annotated", "plaintext": "here was disgraceful graffiti in south Belfast referring to the same thing -- a noose -- and the Belfast", "ranges": [{"start": "/", "end": "/", "startOffset": 154, "endOffset": 161}], "tags": [
```

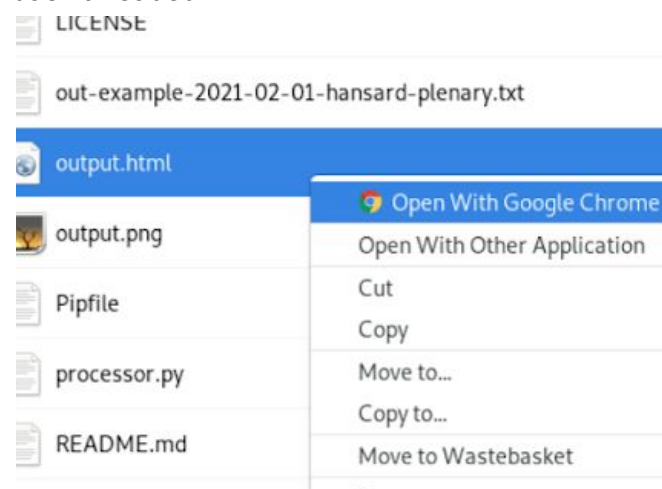
If you would like to see an html output of the processor results, you can run the following command which will generate an output.html file in local directory with challenge files, which you can open in chrome

(foldername) \$

ltdoorstep -o html --output-file output.html process

./sample_transcripts/out-example-2021-02-01-hansard-plenary.txt processor.py -e

dask.threaded



98%

ActivitiesGoogle Chrome

Doorstep Report

File | /home/stephen/lintoldemo/lintolsetup/output.html

Warnings

Processor	Location	Issue	Description
lintol-code-challenge-city-finder:1	FROM: { 'line': 158, 'character': 0 } TO: { 'line': 158, 'character': 283 }	city-cropped-up	Found belfast On a point of order, Mr Speaker. Mr [GDPRREDACT] [GDPRREDACT], [GDPRREDACT]. Is that language
lintol-code-challenge-city-finder:1	FROM: { 'line': 1472, 'character': 0 } TO: { 'line': 1472, 'character': 925 }	city-cropped-up	Found belfast Nevertheless, the changes are welcome now, but there is which is welcome, but politics is not just about what has conducive to their or their constituents time. We need to others who are independents, such as [GDPRREDACT] to be looked at in the long term.

Carrying out the Challenge

Open your favourite Text Editor and add your local directory with the challenge files to your workspace

The magic of the processor is held within the processor.py file which is where you are free to edit and update to implement your new creative ideas

For example, a City Finder function has already been created for you to show off some of the possibilities, however we could make a quick example such as a Month Finder?

This will follow a similar structure to the City Finder function, instead we will be searching through a Months Array, rather than Cities.

```
18 import logging
19 from dask.threaded import get
20
21 from ltldoorstep.processor import DoorstepProcessor
22 from ltldoorstep.aspect import AnnotatedTextAspect
23 from ltldoorstep.report import combine_reports
24 from ltldoorstep.document_utils import load_text, split_into_paragraphs
25
26 # We name some cities - we will do all our comparisons in lower case
27 CITIES = ['armagh', 'belfast', 'derry', 'lisburn', 'newry', 'dulyn']
28
29 MONTHS = ['january', 'february', 'march', 'april', 'may', 'june']
30
31 def city_finder(text, rpvt):
32     """
33     Add report items to indicate where cities appear, and how often
34     """
35
36     # This doorstep utility splits a big text into paragraphs,
```

```
top > processor.py > MONTHS
def month_finder(text, rpvt):
    """
    Add report items to indicate where months appear, and how often
    """

    # This doorstep utility splits a big text into paragraphs, and
    # the punctuation. We do this splitting so that it is easier to
    # than scrolling through highlighted lines in one big document
    paragraphs = split_into_paragraphs(text)

    # This is our counter for months - we initialize every month's
    month_counts = {month: 0 for month in MONTHS}

    # Now we loop through the paragraphs - 'enumerate' gives us a
    for para_n, (paragraph, line_number) in enumerate(paragraphs):

        # As mentioned above, we will search entirely in lowercase
        # a case insensitive search (others welcome too!)
        paragraph_lower = paragraph.lower()

        # We check this paragraph for month names...
        for month in MONTHS:
            # This gets a total count for the month
            month_occurrences = paragraph_lower.count(month)
```

Once we have completed the Month Finder function, we will have to call the updated function in the Class and define it in the output section

```
class CityFinderProcessor(DoorstepProcessor):
    """
    This class wraps some of the Lintol magic under the hood, that lets us plug
    our city finder into the online version, and create reports mixing and matching
    from various processors.
    """

    # This is the type of report we create - it could be tabular (e.g. CSV), geospatial
    # (e.g. GeoJSON) or document, as in this case.
    preset = 'document'

    # This is a unique code and version to identity the processor. The code should be
    # hyphenated, lowercase, and start with lintol-code-challenge
    code = 'lintol-code-challenge-city-finder:1'

    # This is a short phrase or description explaining the processor.
    description = "City Finder for Lintol Coding Challenge"

    # Some of our processors get very complex, so this lets us build up execution graphs
    # However, for the coding challenge, you probably only want one or more steps.
    # To add two more, create functions like city_finder called town_finder and country_finder,
    # then uncomment the code in this function (and remove the extra parenthesis in the 'output' line)
    def get_workflow(self, filename, metadata={}):
        workflow = {
            'load-text': (load_text, filename),
            'get-report': (self.make_report,),
            # 'step-A': (city_finder, 'load-text', 'get-report'),
            'step-B': (month_finder, 'load-text', 'get-report'),
            # 'step-C': (country_finder, 'load-text', 'get-report'),
            'output': (workflow_condense, ['step-B'] #, ['step-B', 'step-C'])
        }
```

Then when we have updated the code, to use the processor in our own creative way, we can view the output by running the html command as before and refreshing the google chrome page

```
(foldername) $ ltldoorstep -o html --output-file output.html process
sample_transcripts/out-example-2021-02-01-hansard-plenary.txt processor.py -e
dask.threaded
```

s organised by the Barnish Action Group and signed by 1,867 people. All of them are asking the Educa
ntrim. The threat of school closure has hung over the rural community of [GDPRREDACT] for a num
lment targets, bureaucratic decisions were made on the basis of funding, transport and nursery provisio
cision in November 2019 to close the school in August 2020 was met with a vociferous local campaig

inties brought about by the COVID health pandemic, there was a sigh of relief when August 2020 can

The challenge is to see how creative you can get with the processor.py file, the html output is simply meant to be a way to view the results and the presentation of this will not be counted.

Repeat these processes in order to keep checking how your updates are coming along and until you are happy with what you have. When you have added in your updates, viewed the output in the html file and are happy to submit, you can then push to git lab

Submitting the Challenge

When you are happy with your testing locally and want to submit some code to see if it runs as you expect on gitlab pages simply git push to your new forked repo

Make sure you are on the master branch

```
$ git status
```

Then Add all of your changes

```
$ git add .
```

Write a commit message to help understand what is different about this commit

```
$ git commit -m "message"
```

Then push to your Repo

```
$ git push -u origin master
```

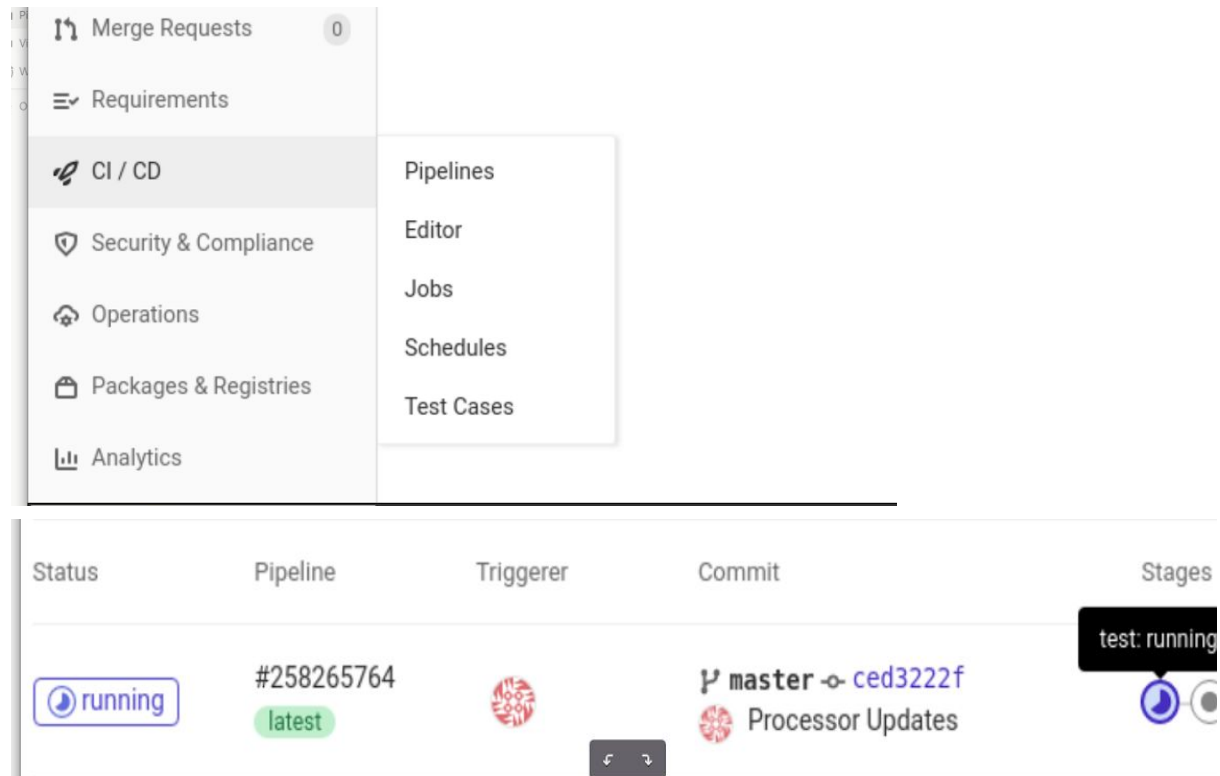
```
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

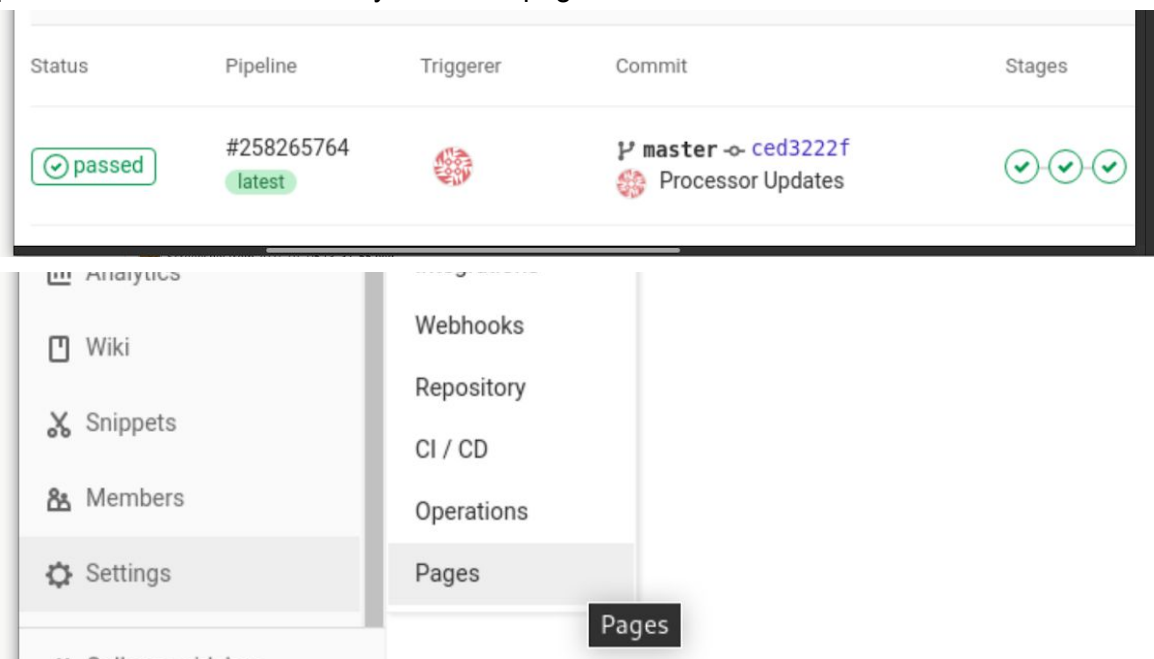
        modified:   output.html
        modified:   processor.py

no changes added to commit (use "git add" and/or "git commit -a")
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ git add .
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ git commit -m "Processor Updates"
[master ced3222] Processor Updates
 2 files changed, 902 insertions(+), 190 deletions(-)
 rewrite output.html (67%)
(lintolsetup) stephen@debian:~/lintoldemo/lintolsetup$ git push -u origin master
```

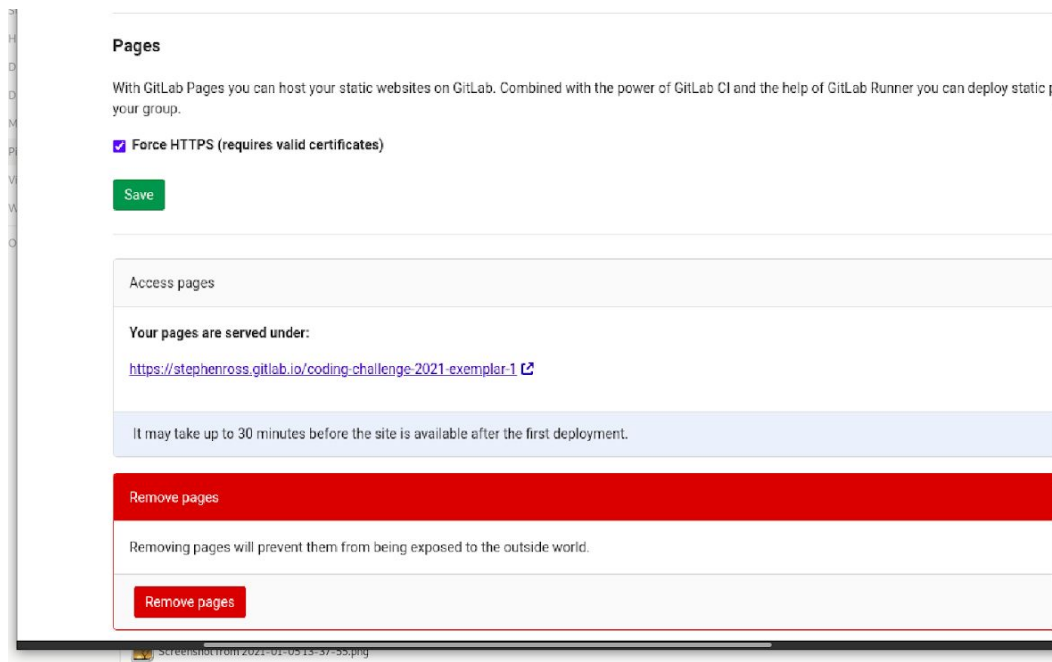
Navigate to your GitLab repo where you pushed the changes to and go to the CI/CD Tab in the Menu and the Pipelines tab in the submenu - here you can see if your push has been processed successfully by looking through each job stage



If all has ran successssfully and all stages show green ticks, it has been sucessffly processed and should be on your Gitlab pages



To view your Gitlab Pages, go to the Settings Tab in the Menu and the Pages tab in the submenu and there should be an Access Pages link to a server where you can view the output.html of your process



Warnings			
Processor	Location	Issue	Description
lintol-code-challenge-city-finder:1	FROM: {'line': 724, 'character': 0} TO: {'line': 724, 'character': 902}	month-cropped-up	Found february Six thematic areas are proposed in the new PEACE PLUS programme. Theme 3, which is "Emp the learning together programme, which will provide direct, sustained contact between school-a programme, which will enhance the capacity of children and young people to form positive and youth mental health and well-being, which will support cross-community and cross-border activ will commence in <u>February</u> 2021.
Info			
Processor	Location	Issue	Description
lintol-code-challenge-	FROM: {'line': 38, 'character': 0} TO: {'line': 38, 'character': 902}	month-cropped-	Found may Mr Speaker, thank you for the opportunity to speak to this Matter of the Day. It is a very import Member who secured the Matter of the Day said. It was wrong and unnecessary, and I think tha

Repeat these processes in order to keep checking how your updates are coming along and until you are happy with what you have