MANUAL TEST PLAN

By Tianli Ding

➔

PREREQUISITES:

In order to use the web scraper and test the code, you could download a text editor, for example, Visual Studio Code. (Fig.1), or PyCharm CE. (Fig.2)

Fig.1

Fig.2

Eclipse Download Link:

https://code.visualstudio.com/download

PyCharm Download Link:

https://www.jetbrains.com/student/

Below we take PyCharm as an example.

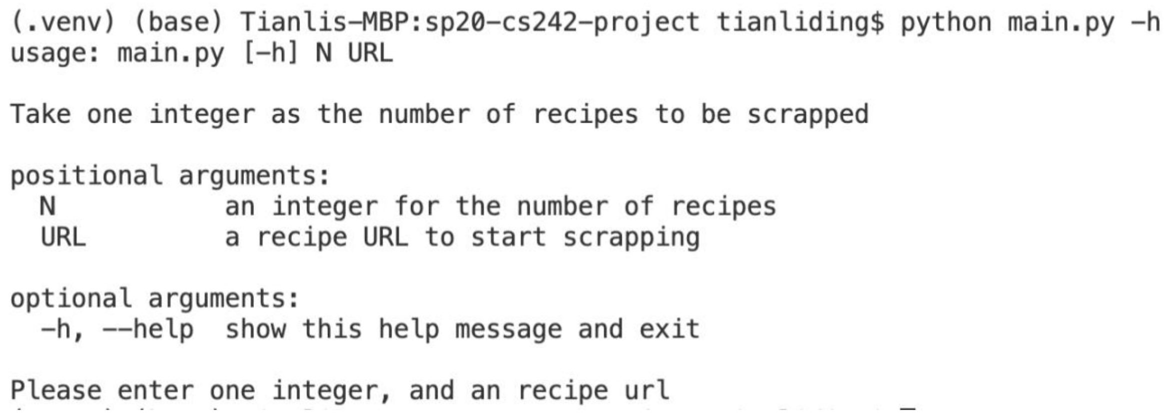
➔

Operations and the Results (screen capture with description):

1. START

To run the web scraper program, you should put in “python main.py -h” in terminal to start.

Then the usage and help information will show up in your terminal as follows:



```
(.venv) (base) Tianlis-MBP:sp20-cs242-project tianliding$ python main.py -h
usage: main.py [-h] N URL

Take one integer as the number of recipes to be scrapped

positional arguments:
  N           an integer for the number of recipes
  URL         a recipe URL to start scrapping

optional arguments:
  -h, --help  show this help message and exit

Please enter one integer, and an recipe url
```

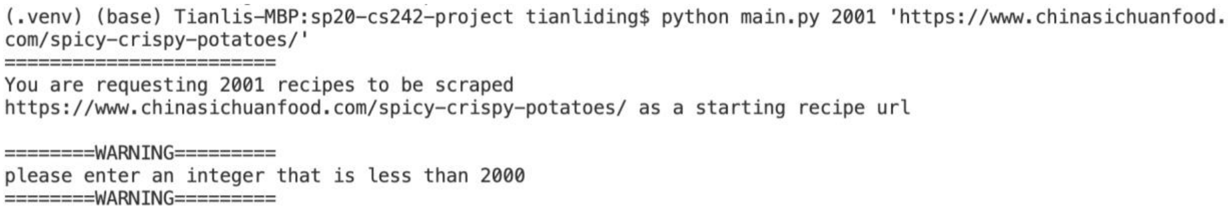
Therefore, the main.py takes one integer and a recipe URL from www.chinasichuanfood.com

The first integer is the number of books to be scrapped, which should be less than 2000.

The URL is a recipe URL as the start page of scrapping.

2. WARNINGS

If you have entered an integer that is larger than 2000, a warning will show up:



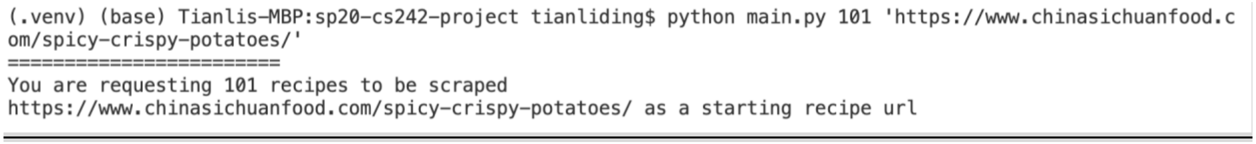
```
(.venv) (base) Tianlis-MBP:sp20-cs242-project tianliding$ python main.py 2001 'https://www.chinasichuanfood.
com/spicy-crispy-potatoes/'
========================
You are requesting 2001 recipes to be scraped
https://www.chinasichuanfood.com/spicy-crispy-potatoes/ as a starting recipe url

========WARNING=========
please enter an integer that is less than 2000
========WARNING=========
```

Therefore, please make sure to enter integer that is smaller than 2000.

3. VALIDITY

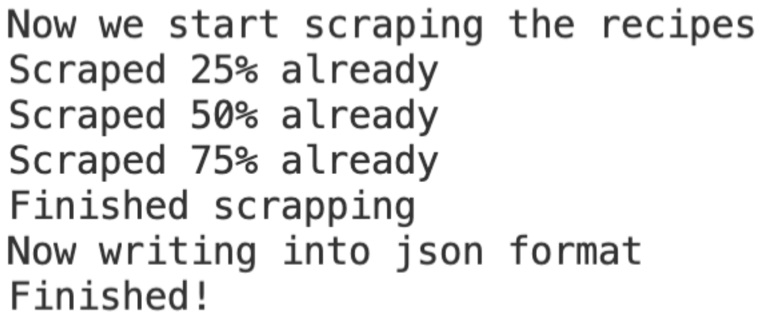
If you have entered a valid number for recipes, and a valid URL, then you will see a message as below:



```
(.venv) (base) Tianlis-MBP:sp20-cs242-project tianliding$ python main.py 101 'https://www.chinasichuanfood.c
om/spicy-crispy-potatoes/'
========================
You are requesting 101 recipes to be scraped
https://www.chinasichuanfood.com/spicy-crispy-potatoes/ as a starting recipe url
```

4. PROCESS

If the scraper start to scape, the process will be shown in the terminal, , as the scrapping going on.

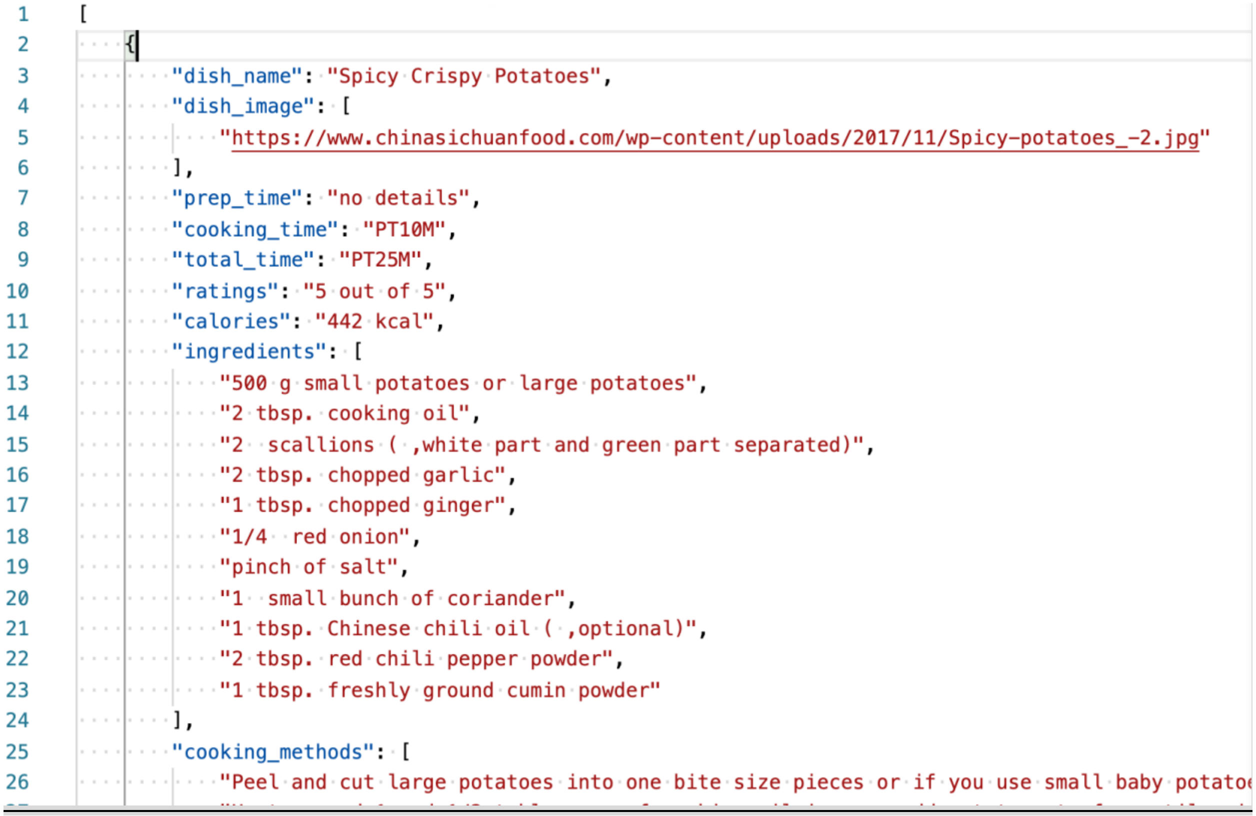


```
Now we start scraping the recipes
Scraped 25% already
Scraped 50% already
Scraped 75% already
Finished scrapping
Now writing into json format
Finished!
```

It may take some time if you have requested a large number of recipes.

5. <u>JSON FILE</u>

The result of the scrape will be saved into json files, namely “recipes.json”.



```
[
    {
        "dish_name": "Spicy Crispy Potatoes",
        "dish_image": [
            "https://www.chinasichuanfood.com/wp-content/uploads/2017/11/Spicy-potatoes_-2.jpg"
        ],
        "prep_time": "no details",
        "cooking_time": "PT10M",
        "total_time": "PT25M",
        "ratings": "5 out of 5",
        "calories": "442 kcal",
        "ingredients": [
            "500 g small potatoes or large potatoes",
            "2 tbsp. cooking oil",
            "2  scallions ( ,white part and green part separated)",
            "2 tbsp. chopped garlic",
            "1 tbsp. chopped ginger",
            "1/4  red onion",
            "pinch of salt",
            "1  small bunch of coriander",
            "1 tbsp. Chinese chili oil ( ,optional)",
            "2 tbsp. red chili pepper powder",
            "1 tbsp. freshly ground cumin powder"
        ],
        "cooking_methods": [
            "Peel and cut large potatoes into one bite size pieces or if you use small baby potatoe
```

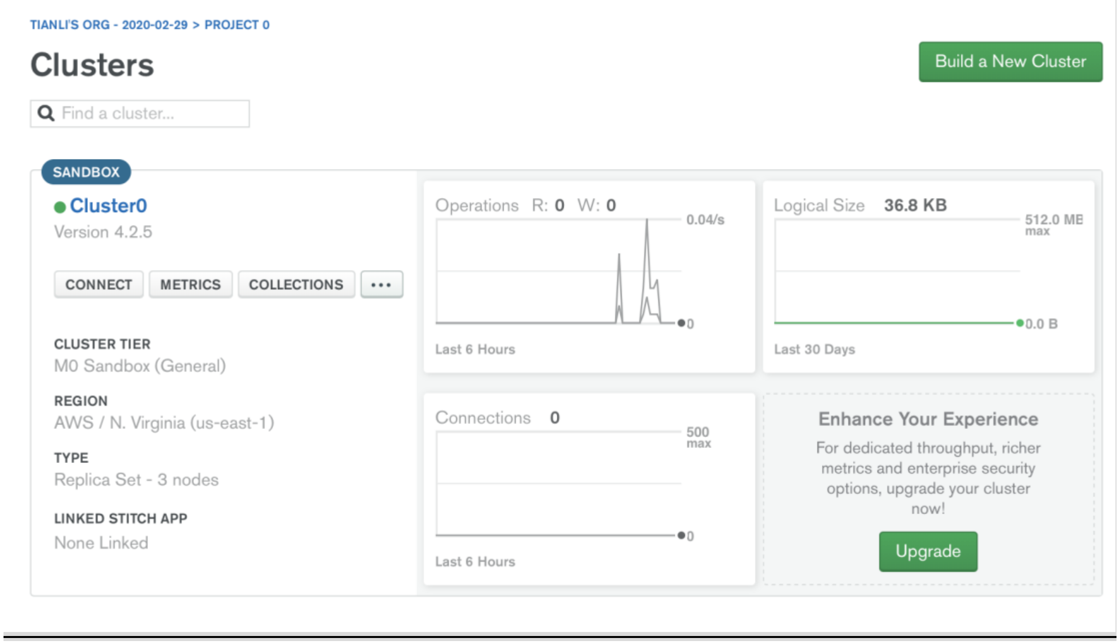
➔ <u>INFORMATION INCLUDED</u>

The information included in the json file are:

- Dish name
- Dish image
- Preparation time
- Cooking time
- Total time
- Ratings
- Calories
- Ingredients
- Cooking methods
- Recommendations

6. DATABASE

For this project, we use MongoDB to store the user information we scraped.



Cluster0
Version 4.2.5

CLUSTER TIER
M0 Sandbox (General)

REGION
AWS / N. Virginia (us-east-1)

TYPE
Replica Set - 3 nodes

LINKED STITCH APP
None Linked

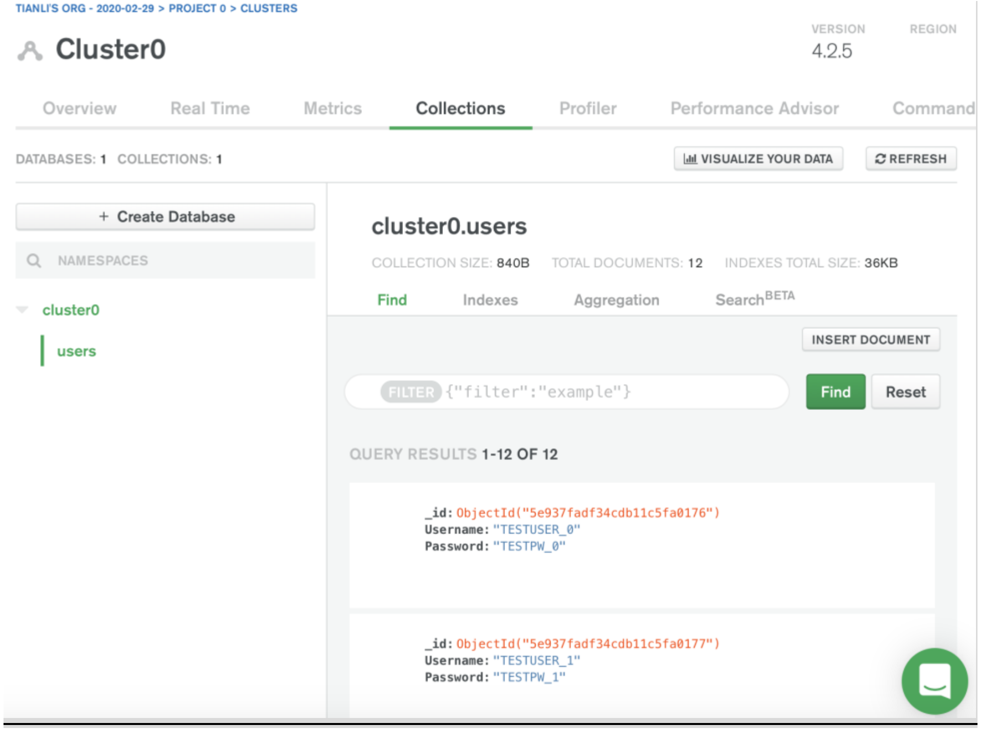
Operations R: 0 W: 0
Last 6 Hours

Logical Size 36.8 KB
Last 30 Days

Connections 0
Last 6 Hours

Enhance Your Experience
For dedicated throughput, richer metrics and enterprise security options, upgrade your cluster now!

In the page below, the cluster0 contains one collection of users, storing user information.



VERSION 4.2.5
REGION

DATABASES: 1 COLLECTIONS: 1

cluster0.users

COLLECTION SIZE: 840B TOTAL DOCUMENTS: 12 INDEXES TOTAL SIZE: 36KB

```
{"filter":"example"}
```

QUERY RESULTS 1-12 OF 12

```
_id: ObjectId("5e937fadf34cdb11c5fa0176")
Username: "TESTUSER_0"
Password: "TESTPW_0"
```

```
_id: ObjectId("5e937fadf34cdb11c5fa0177")
Username: "TESTUSER_1"
Password: "TESTPW_1"
```

7. UNITTEST

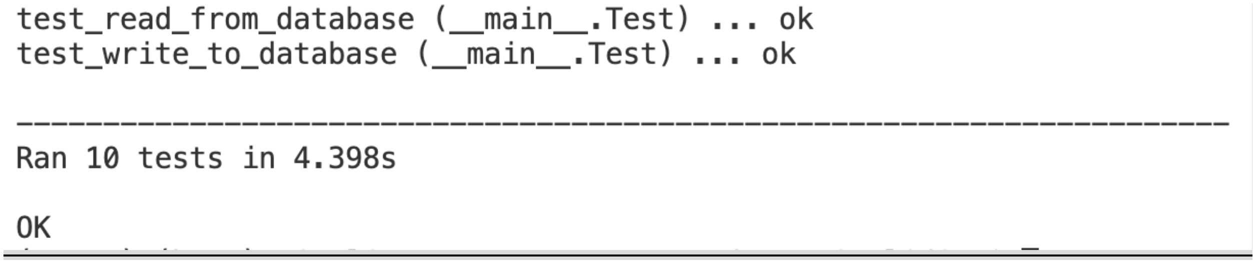
To further test the functionality of the scraper, there is unit test designed for the scraper class and user class.

In “logic_test.py”, you could see all the test for each function as well as tests for json file and database.

To run the logic_test.py, you could type “python logic_test.py” to view the overall test result.

In order to view the details of each test, you can type “python logic_test.py -v”.

If you see “OK” for each of the test, then the scraper and user class should be function fine (see below).



```
test_read_from_database (__main__.Test) ... ok
test_write_to_database (__main__.Test) ... ok

----------------------------------------------------------------------
Ran 10 tests in 4.398s

OK
```