D2 Testing Documentation report

Describe the testing processes used in the development of API, referring to the data and scripts included in Phase_1 folder.

There are mainly two methods we used to test our API, AWS API Gateway ->Method Execution Method Test interface and python with Pytest and requests library. To make it clearly, these two term “**aws method test**” and “**pytest**” is used below to represent this two test method.

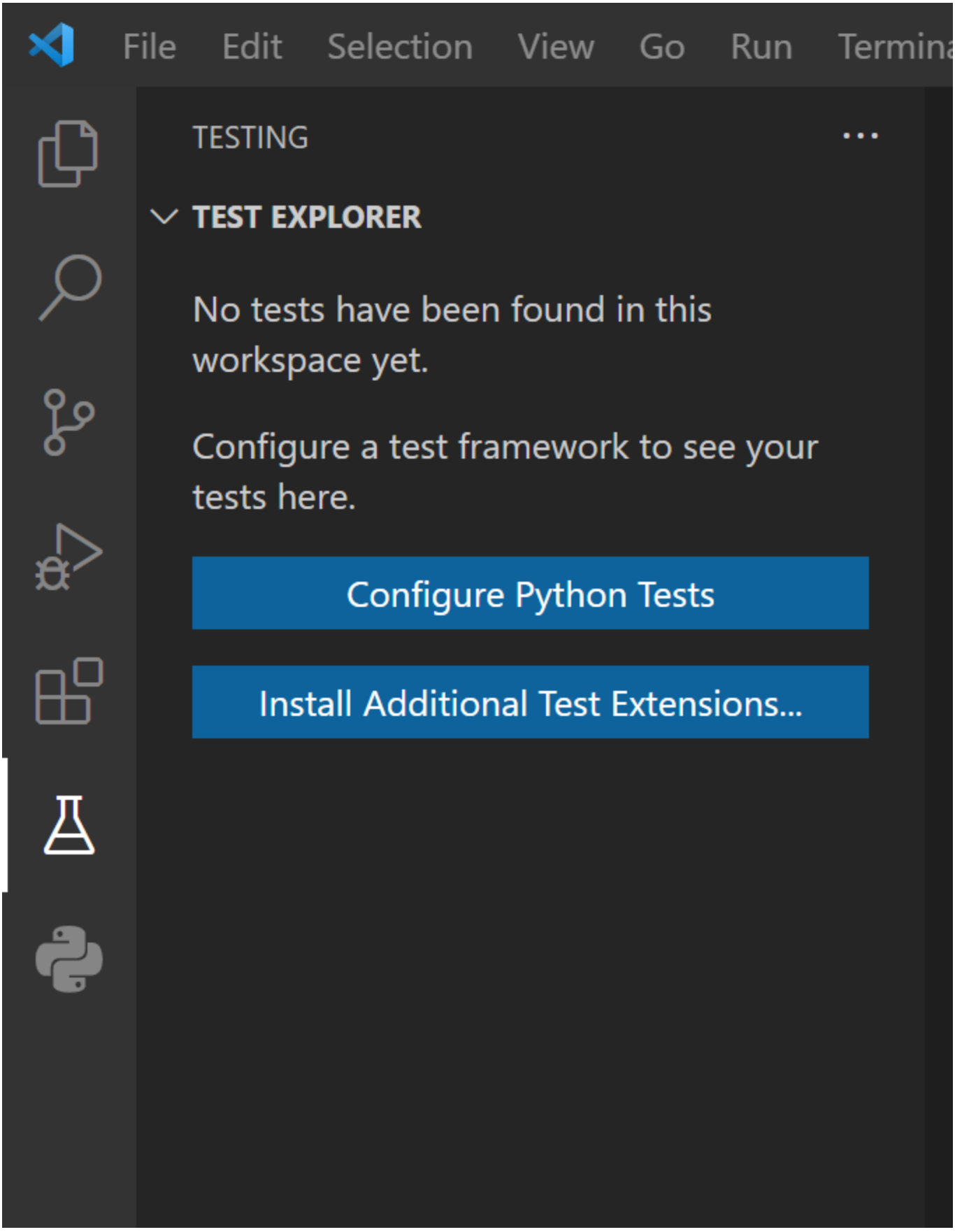
Test Usage

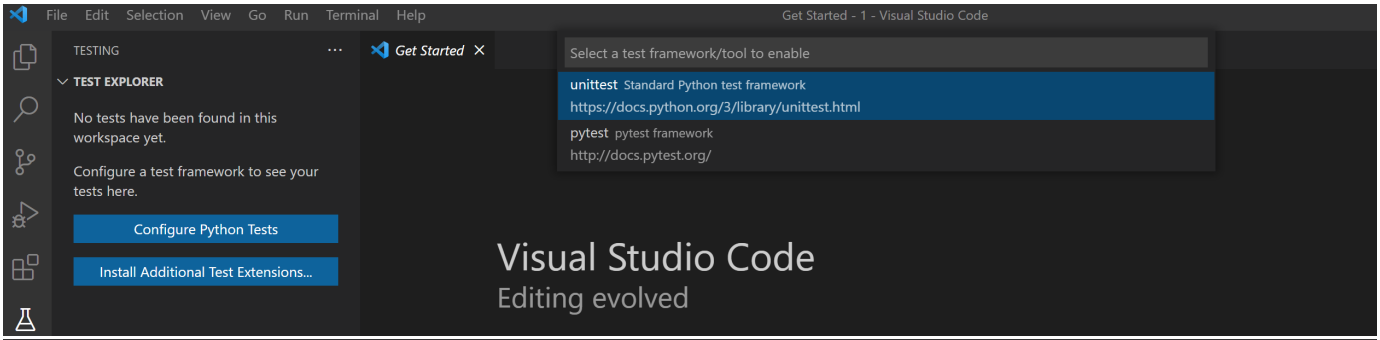
To use **aws method test**, login to AWS is required, the steps of how to login AWS is in D2 API design details.

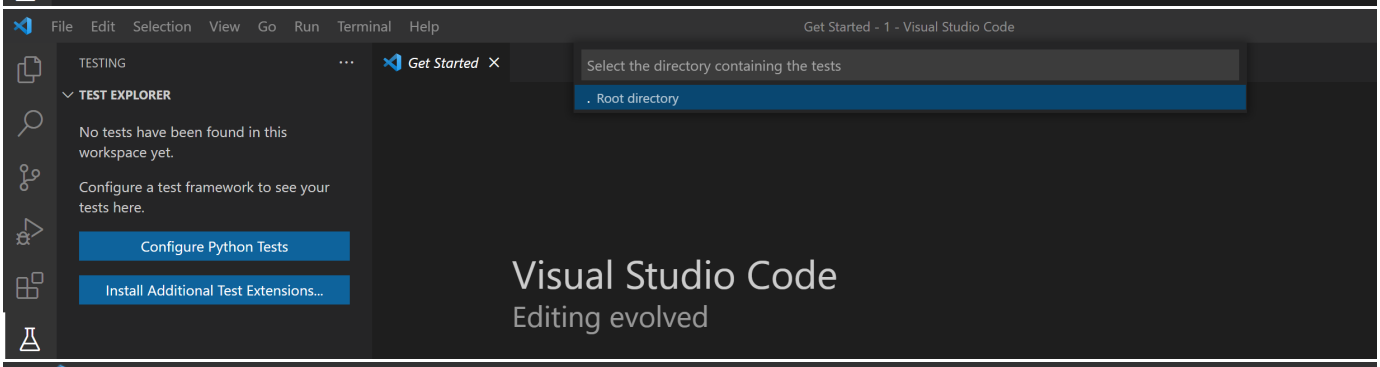
Type API gateway in search bar and click to access AWS API Gateway interface. Go to Resources {The method you want to test} Test. Then put query in the **{index}** input bar and click Test.

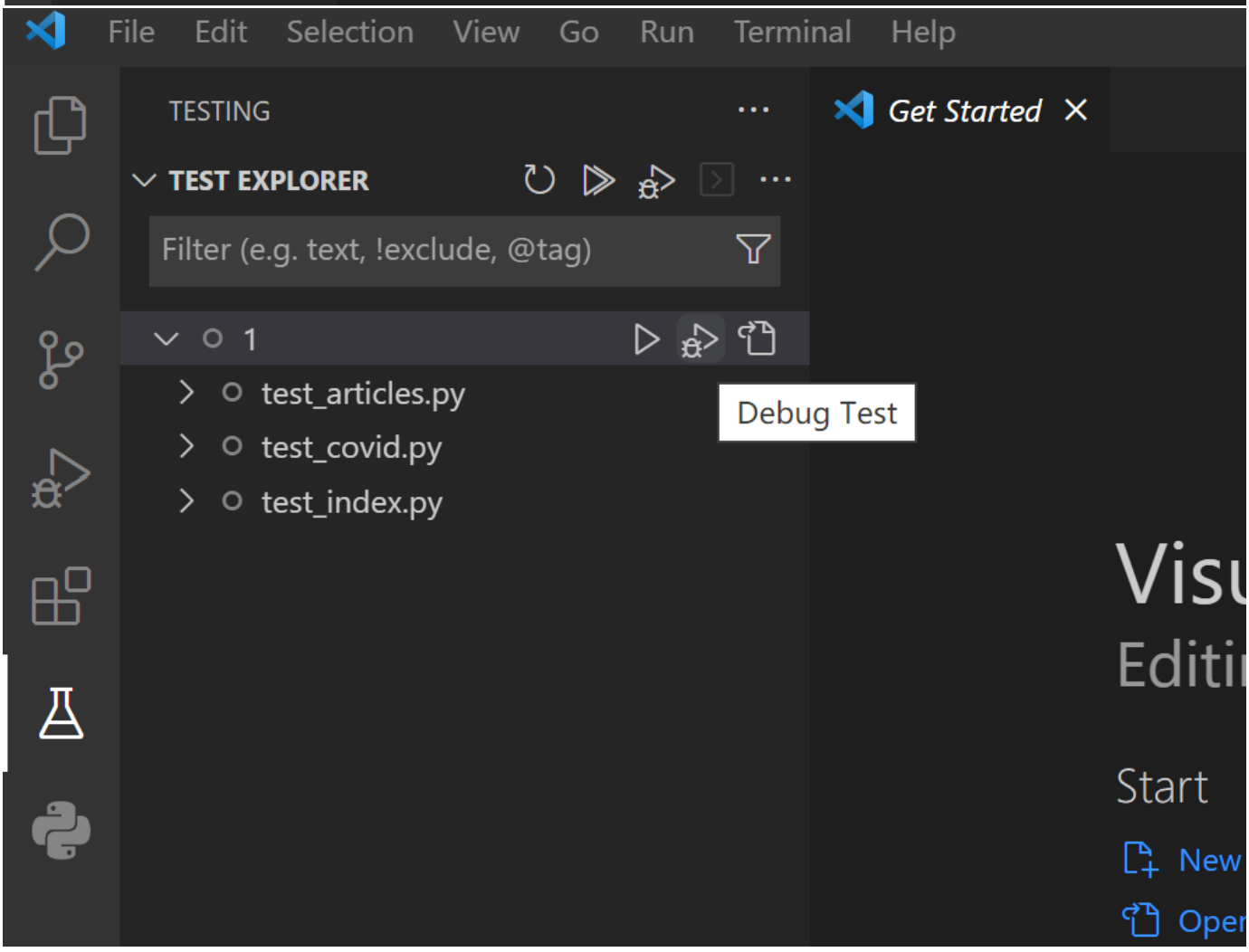
To use **pytest** to test the test scripts, here are the steps of using pytest:

Go to github, find the directory ‘**SENG3011_GroupName/**PHASE_1/TestScripts/', download the folder ‘**test script D2**’, then open this folder with VSCode, go to the left-hand-side column, find the ‘testing icon’. If you are using this for first time, you probably need to configure Python Tests, then select ‘pytest’, then ‘root directory ’. Then you can run the 'Debug Test' for pytesting. The debug console will show the details if some tests went wrong.


File Edit Selection View Go Run Termina
TESTING
TEST EXPLORER
No tests have been found in this workspace yet.
Configure a test framework to see your tests here.
Configure Python Tests
Install Additional Test Extensions...


File Edit Selection View Go Run Terminal Help
Get Started - 1 - Visual Studio Code
TESTING
TEST EXPLORER
No tests have been found in this workspace yet.
Configure a test framework to see your tests here.
Configure Python Tests
Install Additional Test Extensions...
Get Started
Select a test framework/tool to enable
unittest Standard Python test framework
https://docs.python.org/3/library/unittest.html
pytest pytest framework
http://docs.pytest.org/
Visual Studio Code
Editing evolved


File Edit Selection View Go Run Terminal Help
Get Started - 1 - Visual Studio Code
TESTING
TEST EXPLORER
No tests have been found in this workspace yet.
Configure a test framework to see your tests here.
Configure Python Tests
Install Additional Test Extensions...
Get Started
Select the directory containing the tests
. Root directory
Visual Studio Code
Editing evolved


File Edit Selection View Go Run Terminal Help
TESTING
Get Started
TEST EXPLORER
Filter (e.g. text, !exclude, @tag)
1
test_articles.py
test_covid.py
test_index.py
Debug Test
Start
New
Oper

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

============================= test session starts ==============================
platform win32 -- Python 3.7.9, pytest-6.1.1, py-1.9.0, pluggy-0.13.1
rootdir: c:\Users\fando\Desktop\1
collected 39 items

test_articles.py ..........                                              [ 25%]
test_covid.py
```

This two methods actually represent our development stage, we use aws method test more often in the relatively early stage of development. This is because aws method test gives more detailed error message and full request log when something went wrong, make it easier to debug the program.

Here is a screen shot of our text file used to store some queries for aws method test. And it's position on github repo.

SENG3011_GroupName/PHASE_1/TestScripts/AWS_Method_Test.txt

```
covid:
    200:
        country=US&date=02-02-2020
        country=Mainland_China&date=02-02-2020
        state=New_South_Wales&country=Australia&date=02-02-2020
    400:
        &country=US&date=2020-02-02
    204:
        state=Chicago,_IL&country=US&date=02-02-2018


articles:
    200:
        location=New_York&key_term=measles&period_of_interest=15-10-01T08:45:10&period_of_interest=22-11-01T19:37:12&limit=0
    400:
        location=Beni&key_term=ebola&period_of_interest=qqq&period_of_interest=qqq


index:
    200:
        location=Beni&key_term=ebola&period_of_interest=00-03-17T20:33:40&period_of_interest=22-04-18T21:23:52
    400:
        location=Beni&key_term=ebola&period_of_interest=qqq&period_of_interest=qqq
```

Here is the position of pytest files.

SENG3011_GroupName/PHASE_1/TestScripts/test script D2/test_covid.py

SENG3011_GroupName/PHASE_1/TestScripts/test script D2/test_articles.py

SENG3011_GroupName/PHASE_1/TestScripts/test script D2/test_index.py

and the position of test input file

SENG3011_GroupName/PHASE_1/TestScripts/test script D2/input_file_main_text_articles.json

SENG3011_GroupName/PHASE_1/TestScripts/test script D2/input_file_main_text_index.json

We use pytest when development is close to finish, when there is less possibility to occur bug. This is because pytest provides great automated testing function which is much more efficient than other methods.

So the usual testing process in later development stage is to use about 20 pytest functions to check one endpoint. When a test function failed, trun to aws method test to try again then give request and response log file to the development team to fix the bug.

Describe your testing environment and/or tools used, and limitations (e.g. things that are not tested).

As we stated before, we mainly use two method pytest , and aws method test. Beside that, we also use swagger UI “try it out“ a few times. The limitation of all these methods is described below.

When we use aws method test, we have to manually input the query even we prepare a txt file to store the query, it is still way too slow. And it also hide one specific error which is “Missing Authentication Token”, this error only occur when the API gateway is not deployed (our API will be undeployed when we change API structure). The reason is that use aws method test doesn’t require API to be deployed, and this error could be easily identified by all other test methods.

We also use swagger UI “try it out“ a few times just to check it works on “try it out“ and check the query input by “try it out“ is the same format. It needs manually input the query and no error message which is the reason why we don’t use it a lot.

Pytest provide some good aotomated testing features, but when use pytest to test API endponit we can’t see the coverage of our program, unlike testing backend program. This requires us to spend more time to find edge cases and can’t be sure about if we chacked all edge cases. And when we use Pytest to test our API and something goes wrong, only the clinet side error message will be shown (the development team have to go to AWS cloud watch and search for that error to see the full log file). Make the debug phase takes more time.

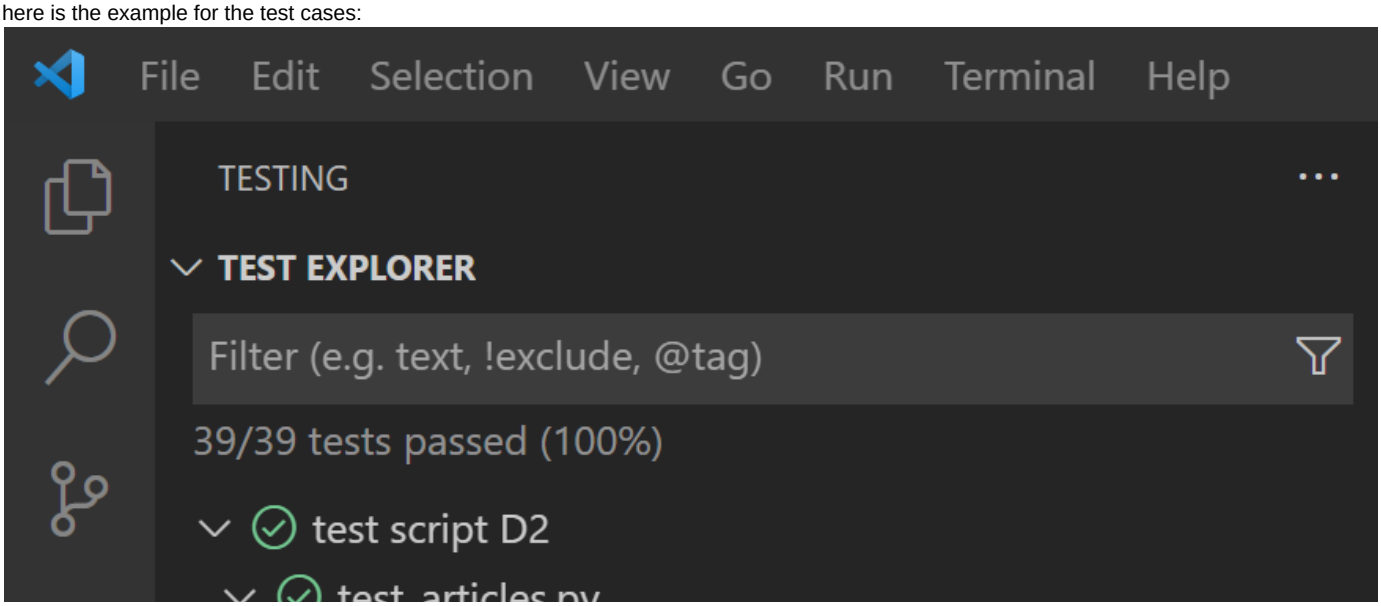
Describe your testing process i.e. how your team conducts testing using the test data (e.g. in which order) and an overview of test cases, testing data and testing results.

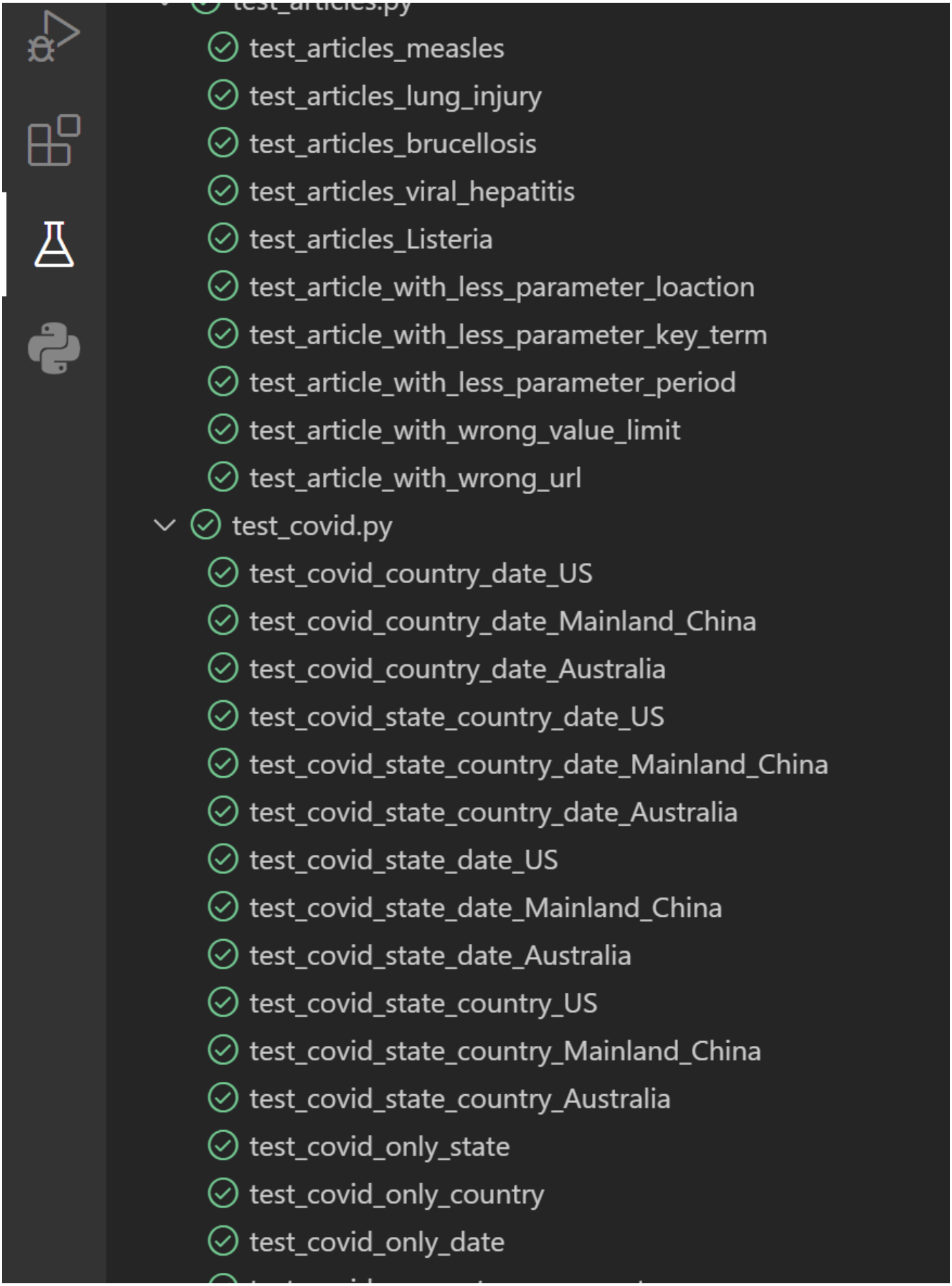
For the testing process, the team decided to do the important tests first such as the ‘200 response', and then do the edge cases such as ‘400 response’ to make sure the API can work in basic usage. There are three endpoints, so the team decide split them into three test files as ‘test_articles.py’, ‘test_covid.py’, and ‘test_index.py’ which make the tests more clear and organized. Then in each file, we first test the request that can work, which is 200 response, then test the 400 response, 204 response.

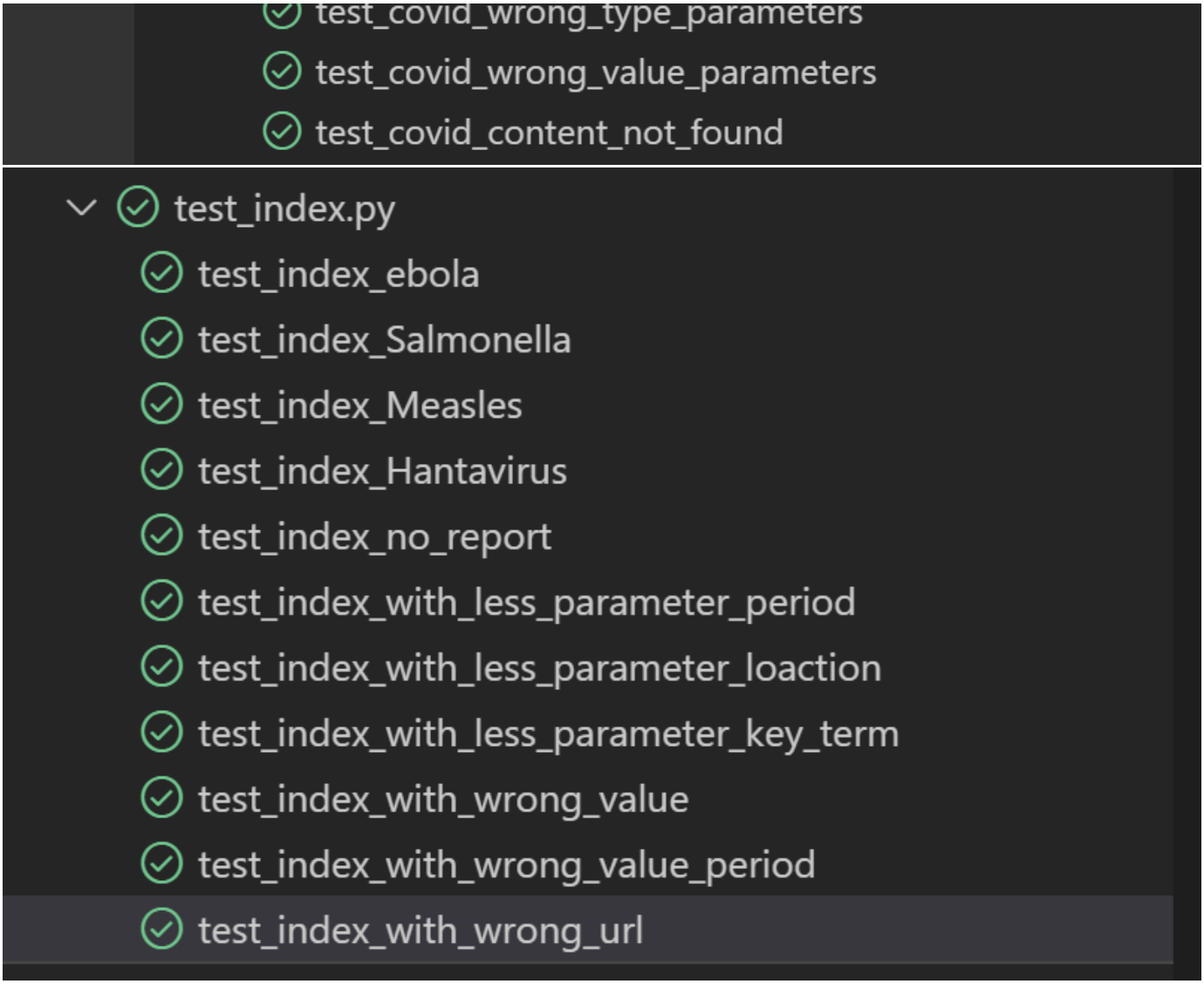
Here is an example in 'test_articles.py', where response is the result data from request and the right-hand-side data is from the CDC website. Additionally, as the access_time in API is always changing, so it is hard to test properly. Then we decide to remove it from response first and then check it with the current_time when testing. Therefore, we only need to compare the difference of these time which is more easy. The details of how to do that is in the test files.

```
assert True == access_time_status
assert response == [
                    {
                        "team_name": "SENG3011_GroupName",
                        "url": "https://www.cdc.gov//measles/cases-outbreaks.html",
                        "date_of_publication": "Announced January 2019",
                        "header": "Measles Cases and Outbreaks",
                        "main_text": main_text["measles"],
                        "reports": [
                        {
                            "event_date": "January 2019",
                            "locations": [
                            {
                                "country": "Have not been implantment",
                                "location": "New York"
                            },
                            {
                                "country": "Have not been implantment",
                                "location": "Columbia"
                            }],
                            "diseases":
                                [{"name": "Measles (Rubeola)"}],
                            "syndromes":
                                [{"name": "did not find any"}],
                            "cases": 49,
                            "deaths": 0,
                            "hospitalizations": 0
                        }
                        ]
                    }
                   ]
```

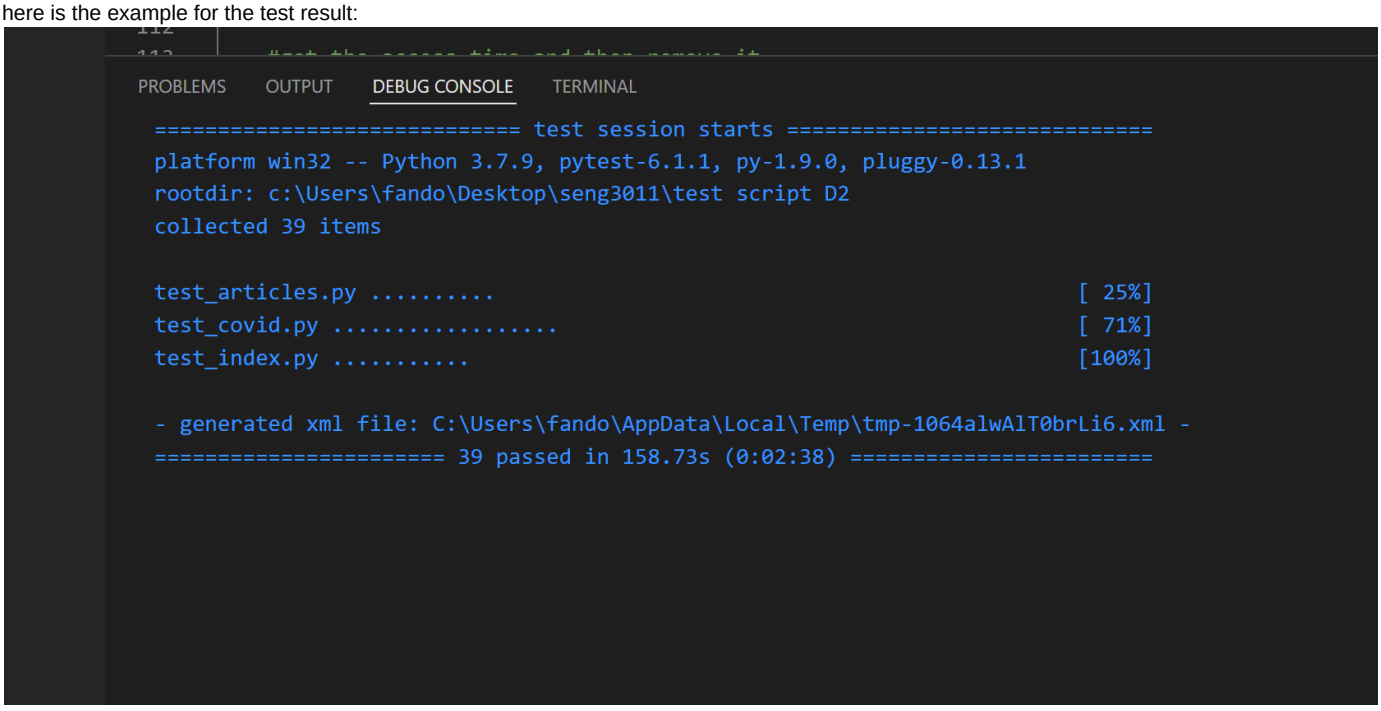
here is the example for the test cases:




test_articles_measles
test_articles_lung_injury
test_articles_brucellosis
test_articles_viral_hepatitis
test_articles_Listeria
test_article_with_less_parameter_loaction
test_article_with_less_parameter_key_term
test_article_with_less_parameter_period
test_article_with_wrong_value_limit
test_article_with_wrong_url
test_covid.py
test_covid_country_date_US
test_covid_country_date_Mainland_China
test_covid_country_date_Australia
test_covid_state_country_date_US
test_covid_state_country_date_Mainland_China
test_covid_state_country_date_Australia
test_covid_state_date_US
test_covid_state_date_Mainland_China
test_covid_state_date_Australia
test_covid_state_country_US
test_covid_state_country_Mainland_China
test_covid_state_country_Australia
test_covid_only_state
test_covid_only_country
test_covid_only_date



here is the example for the test result:



Describe the output of testing and what actions you took to improve the test results.

When pytest is used, it will output the correctness of each test case like the images above, and only show client side error message when test failed.

When aws method test is used, it will output the full request and response (with no indicator of right or wrong result).

In order to improve the test results, we include more test cases in our Pytest script and consider more edge cases and different positive cases.

The aim of our test script is to at least one positive test for each founction of our API e.g. each unique key_term of /index and /articles and with ot without “state” parameter of /covid. And at least one edge case test for each type of non 200 response.