

End to End Test Explanations

e2e_test_view_flights.py:

This script will test the end-to-end process of a user starting their web browser, navigating to the website, and clicking the link that allows users to view their flights. There are comments in the script itself to explain the process, but the general idea is creating a chrome instance, navigating to the website hosted on localhost at port 5000, finding the link to View Flights, clicking it, and checking that some expected data exists in a table element on that page. The following is a screenshot of the output of this test:

```
(venv) alexandermuglia@Alexanders-MBP Group-37-CH % python3 app/e2e_test_view_flights.py
/Users/alexandermuglia/assignments/qa/Group-37-CH/venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
Opened site home page.
Clicked link to get to Flight Viewing page
Table loaded in the View Flight page
YYZ is in the page source
(venv) alexandermuglia@Alexanders-MBP Group-37-CH %
```

We see that the page navigation goes as expected, and we find the table element with the YYZ airport inside of it.

e2e_test_flight_search.py:

This script will test the end-to-end process of a user starting their web browser, navigating to the website, and clicking the link that allows users to search for flights. There are comments in the script itself to explain the process, but the general idea is creating a chrome instance, navigating to the website hosted on localhost at port 5000, finding the link to Flight Search, clicking it, and checking that some expected data exists on that page (in this case, a dropdown to select departure airport code and a dropdown to select arrival airport code). The following is a screenshot of the output of this test:

```
(venv) (base) MacBookPro:Group-37-CH nasreen$ python3 app/e2e_test_flight_search.py
Opened site home page.
Clicked link to get to Flight Search page
Departure dropdown is loaded in the Flight Search page.
Arrival dropdown is loaded in the Flight Search page.
Select Departure Airport Code is in the page source
Select Arrival Airport Code is in the page source
(venv) (base) MacBookPro:Group-37-CH nasreen$
```

Page navigation goes as expected, and we find the text element with Select Departure/Arrival Airport Code exists.

Running the End to End Tests

The following is for linux / macos systems. If you are on Windows, the process should be the same but you will have to use the Powershell equivalent of all of the commands. Navigate to the root directory of the project in a terminal to begin.

1. If you have a venv already created, skip to step 2. If you don't have a venv yet, in the root directory of the project run
python3 -m venv venv

This will create a venv in the root directory.

2. Start the venv. If you created one in the root directory as shown above, run
`source venv/bin/activate`
3. Install the requirements from the given requirements.txt file with
`pip3 install -r requirements.txt`
4. Start the server for the website with
`python3 app/app.py`
5. If you do not yet have chrome installed, install the chrome web browser. This is required to do the end to end tests. Note that the executable file for chrome must be in the default location, i.e.:

OS	Expected Location of Chrome
Linux	/usr/bin/google-chrome ¹
Mac	/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome
Windows XP	%HOMEPATH%\Local Settings\Application Data\Google\Chrome\Application\chrome.exe
Windows Vista and newer	C:\Users%USERNAME%\AppData\Local\Google\Chrome\Application\chrome.exe

6. In another terminal, navigate to the root project directory again. Make sure to start the venv again to ensure you are using the correct packages (see Step 2). Once you have started the venv in this new terminal window, run the tests with
`python3 app/<test-name>`
For example, to run the `e2e_test_view_flights` test, use the command
`python3 app/e2e_test_view_flights.py`
7. Repeat step 5 for the other end to end test(s)
8. Once you are finished, you can deactivate the venv with
`deactivate`