



Lab: Using a Catchup Subscription

Overview

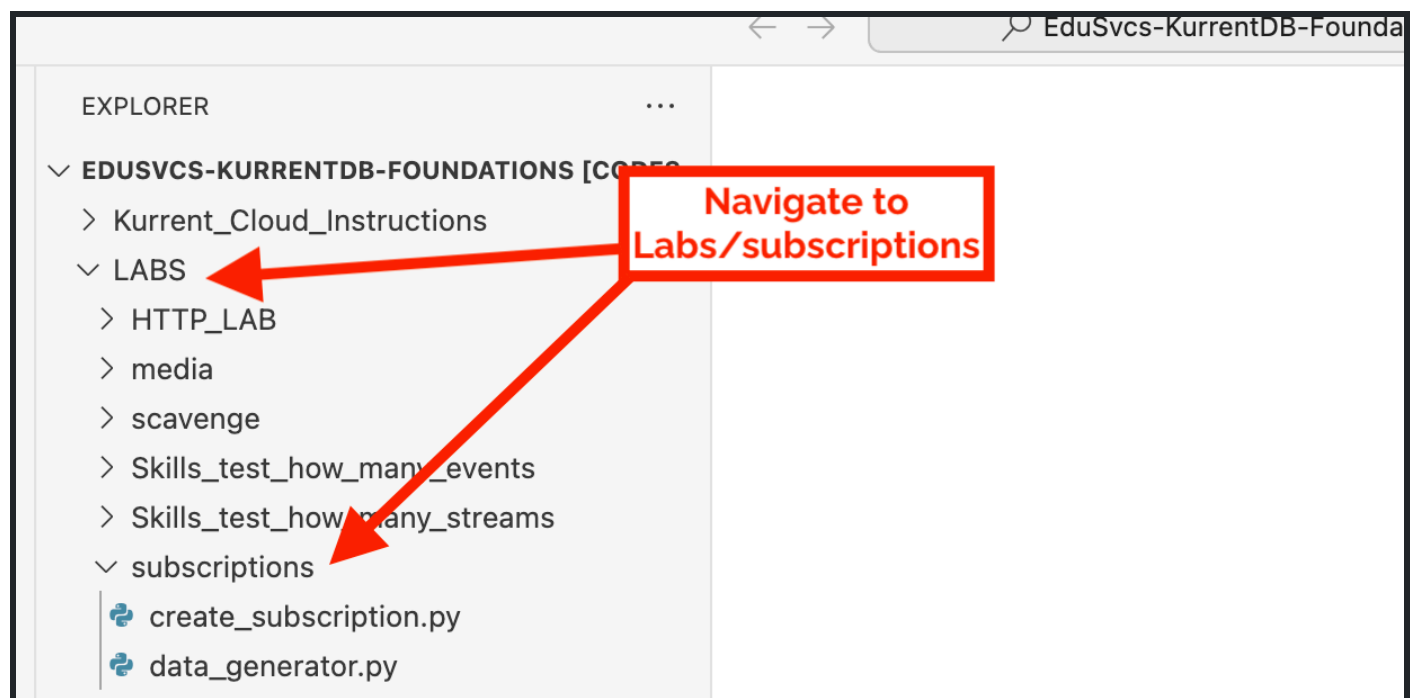
In this lab, you will create a catchup subscription to a stream. The subscription will be used to maintain a list of users.

Notes:

This Lab requires nothing more than a running KurrentDB database and python. If attempted outside of the context of this course, any KurrentDB installation, including Kurrent cloud would work. The only modification would be to change the connection string. For python create a virtual environment and run `pip install -r requirements.txt` to install the needed libraries.

Step 1: Navigate to the Labs

Open the `LABS/subscriptions` folder.



Step 2: Review the Contents

There are two python programs.

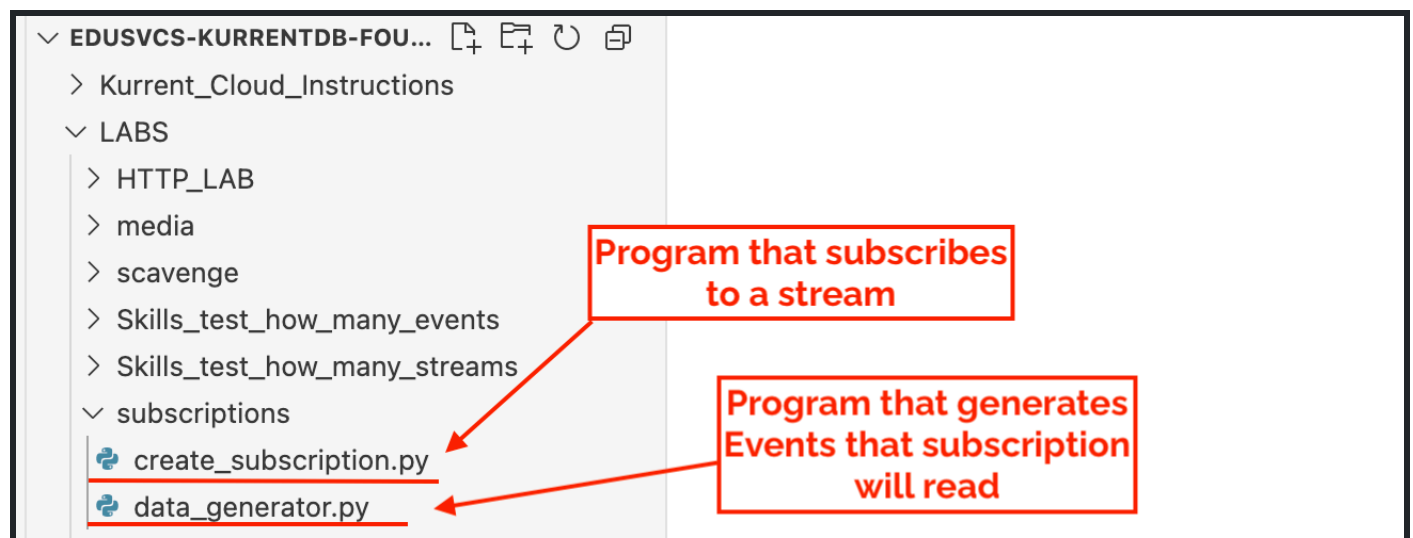
`create_subscription.py`

and

`data_generator.py`

`data_generator.py` writes events into KurrentDB.

`create_subscription.py` creates a subscription that will read the events written by `data_generator.py` extract the value of the name field and write that to file to maintain a list of users in all streams that start with the string `CAN`.



Step 3: Review the Data Generator Code

The data generator appends events with user information.

```
{
  "name": "Richard Simmons",
  "address": "742 Hall Fork\nNicoleland, KS 33562",
  "email": "jluna@example.com",
  "job": "Theatre manager"
}
```

The events are appended to streams named

`(COUNTRY_CODE)-(0-9)`

Where `COUNTRY_CODE` is a random choice of "CAN,USA,MEX" and (0-9) is a random integer between 0-9.

`python data_generator.py` will append 100 events.

`python data_generator.py -n 99` will append 99 events.

`python data_generator.py -n 99 -f log.txt` will append 99 events and record the time it took to execute in the file `log.txt`

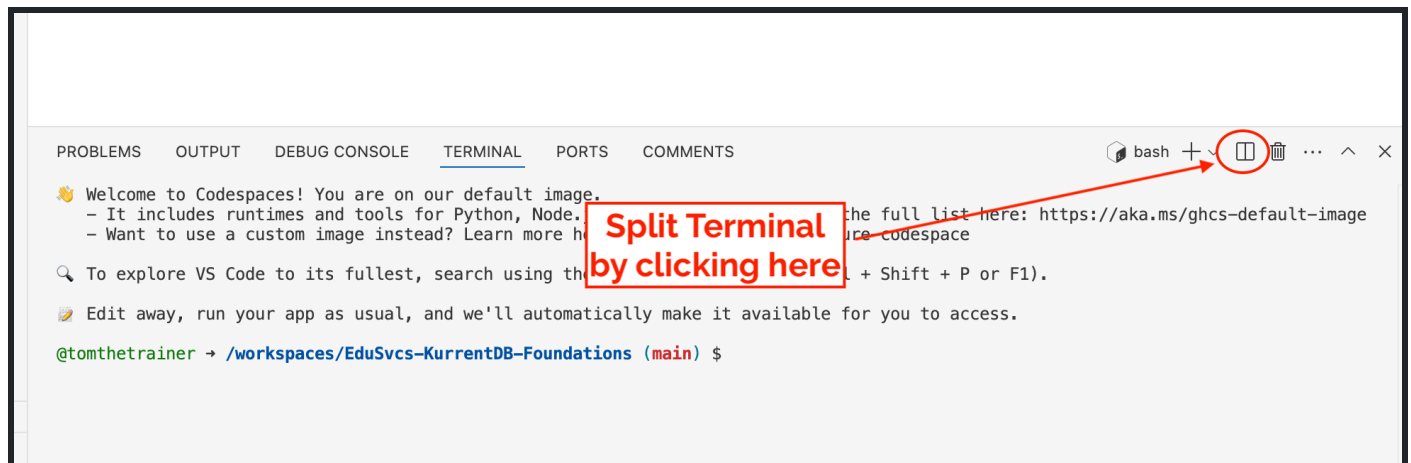
There is no need to run the program at this point, but if you do that is alright.

Step 4: Review the Content of the Create Subscription Code

The code is documented with comments, read through and review each section.

Step 5: Split the Terminal Window

Since the subscription code will continue running and waiting for events, it will occupy a terminal while it executes. In order to generate events, and view the subscription receive the events two terminal windows are needed.



Step 6: Run Each Program

Make sure you change to the `LABS/subscriptions` folder. The subscription code creates a file, and if ran in another directory it will be difficult to find the file.

In the terminal enter `python create_subscription.py`

If you want to stop the program, enter `ctrl-C` twice.

Restart by running `python create_subscription.py`

In the second terminal run the data generator by entering

`python data_generator.py`

With each run of the data generator, you should see more events in the terminal of the subscription.

Step 7: Review the Contents of the Data File [🔗](#)

Open the file `Canada_userlist.txt`.

You should see something similar to this.

```
1,John Ford
2,Bobby Hutchinson
3,Stephen Callahan
4,Julie Owens
5,Bryan Kent
6,Christine Watson MD
7,Daniel Weber
8,Cynthia Snyder
9,Richard Freeman
10,Sara Smith
11,Toni Pineda
12,Ann Thomas
```

Optional Steps:

Delete the userslist file, and rerun the subscriber.

What happens?

Restart the database by running `start_db.sh`

Rerun the data generator and the subscription.

What happens?

Did you delete the userslist? If not, delete it and restart the subscription.

What happens?

Congratulations !!!

You now know more about creating and using catchup subscriptions in KurrentDB.