

# Dr Stocks Backend Setup Guide

---

Prerequisite to run this project

1. WAMPServer
2. MySQL
3. Python
4. Docker
5. Postman

How to setup the backend services?

1. Unzip the zip file into a folder on your desktop
2. Open the backend folder and navigate to the "dr\_stocks-backend/database" folder
3. Start the WAMPServer
4. Open all the SQL files using MySQL (via phpmyadmin or not) and execute/run all the files
5. Next, Open the "docker-compose.yml" file and change the docker id for the images to your own docker id.

```
... - kongadata: /app/kongadata

#####
# stock_info: The stock_info microservice
#####
stock_info:
  build:
    context: ./
    dockerfile: stock_info.Dockerfile
  image: markytan/stock_info:esd
  restart: always
  networks:
    - kong-net
  environment:
    dbURL: mysql+mysqlconnector://is213@host.docker.internal:3306/stock_infoDB
    PYTHONUNBUFFERED: 1
  ports:
    - "5001:5001"

#####
# stock_pref: The stock_pref microservice
#####
```

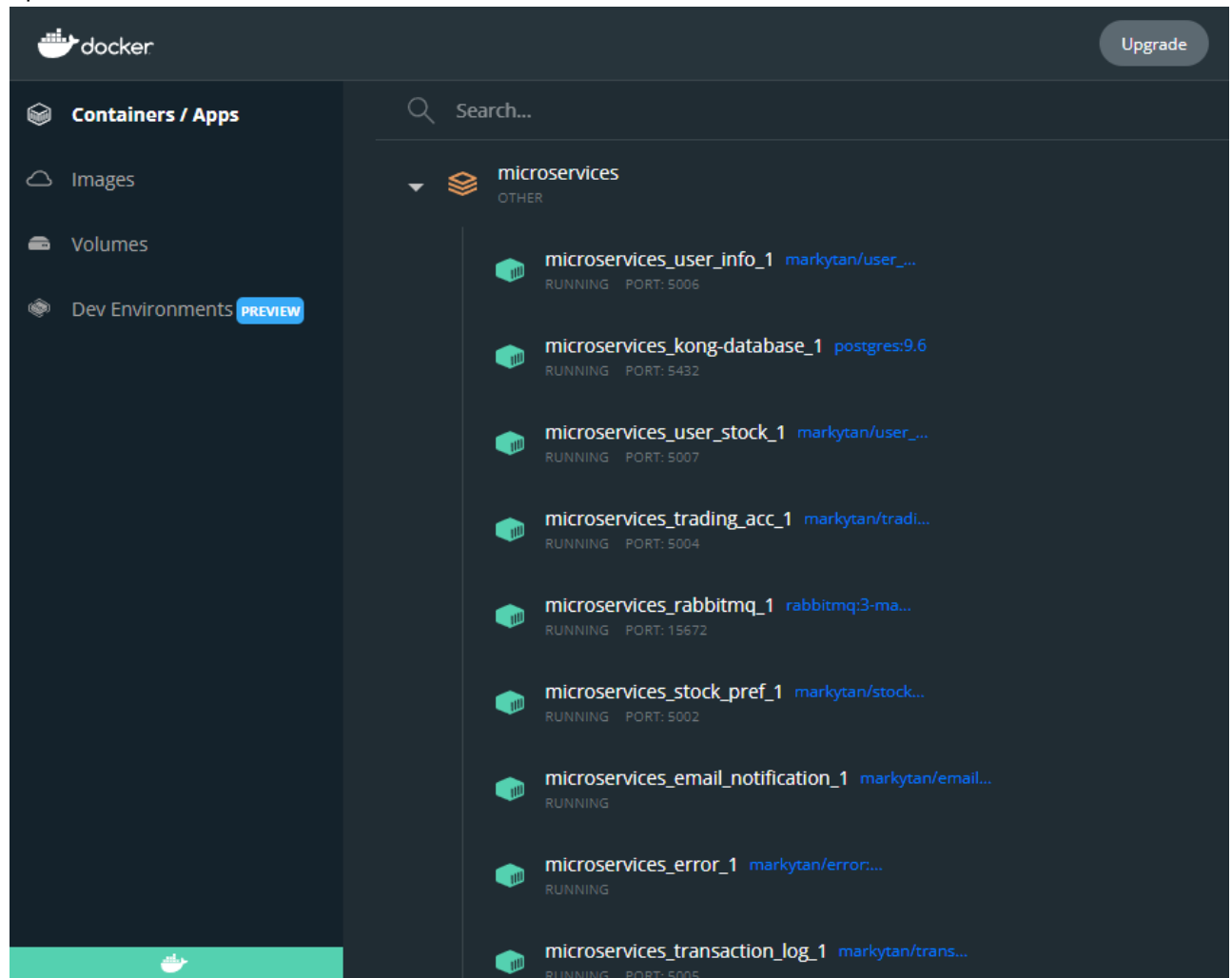
6. Make sure you have run the Docker application.
7. Open a cmd terminal and cd to the "dr\_stocks-backend/microservices" folder.
8. After, proceed to type this command "docker-compose up -d" and hit enter to run it in the terminal. This will set up all the required services.

```
Microsoft Windows [Version 10.0.19042.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ezeki\Documents\GitHub\dr_stocks-backend>cd microservices

C:\Users\ezeki\Documents\GitHub\dr_stocks-backend\microservices>docker-compose up -d
```

9. Wait for about 30 seconds after the containers have been created and started to let the services start up.



10. Next, type this URL "localhost:1337" in your browser to access the KONGA portal.

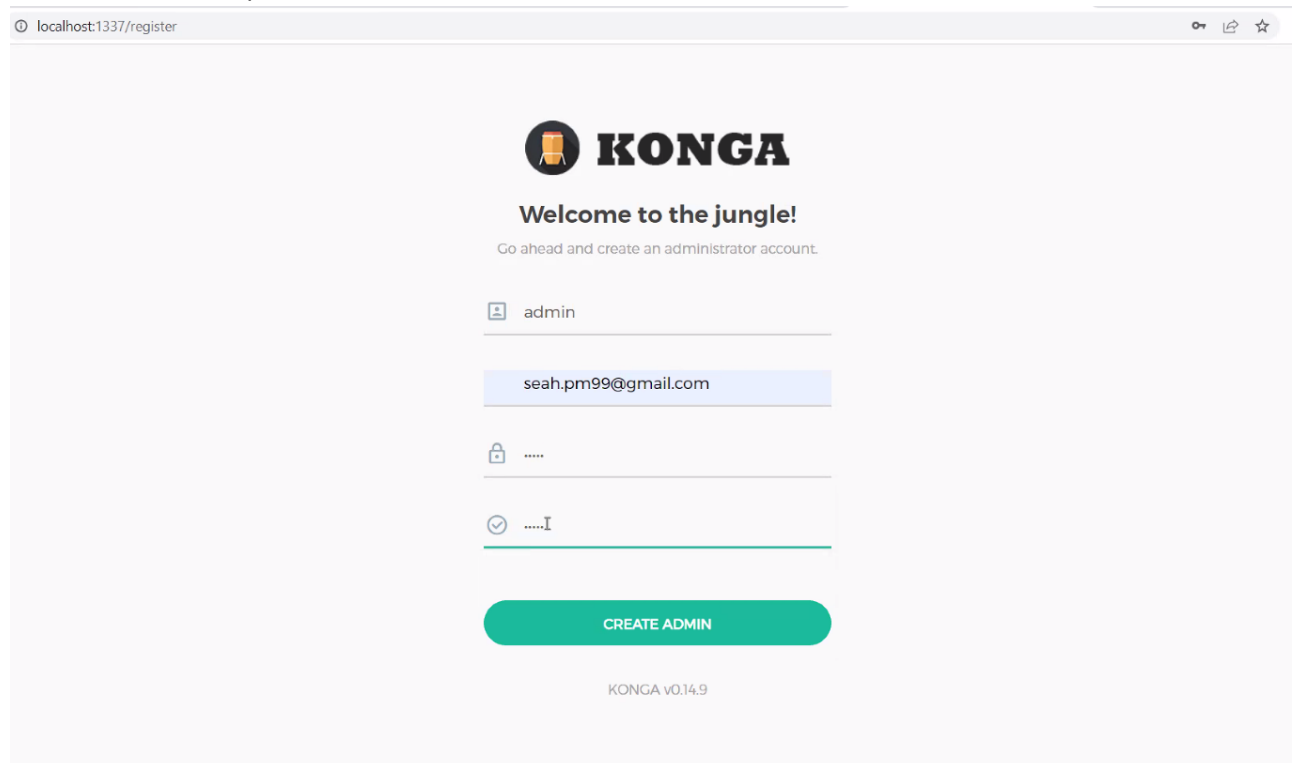
11. Create an admin account:

Name: admin

Email: Any email

Password: password

Confirm Password: password

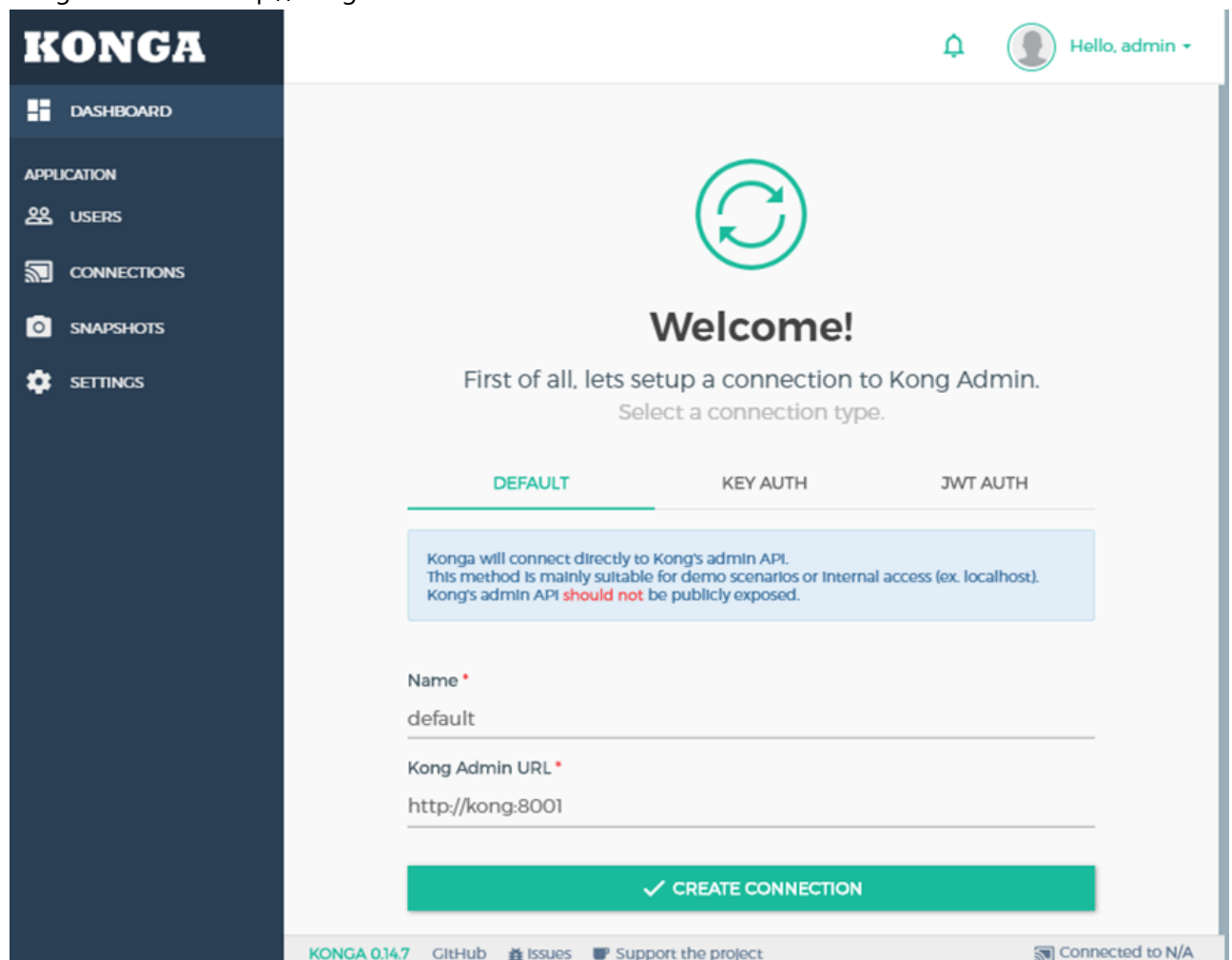


The image shows the Kong Admin registration page. At the top, it says "Confirm Password: password". Below that is a browser address bar showing "localhost:1337/register". The main content area has the Kong logo and the text "Welcome to the jungle! Go ahead and create an administrator account." There are four input fields: "admin" for the username, "seah.pm99@gmail.com" for the email, a password field with a lock icon and ".....", and a confirm password field with a checkmark icon and ".....I". A green "CREATE ADMIN" button is at the bottom. The footer says "KONGA v0.14.9".

12. and set the default connection as this:

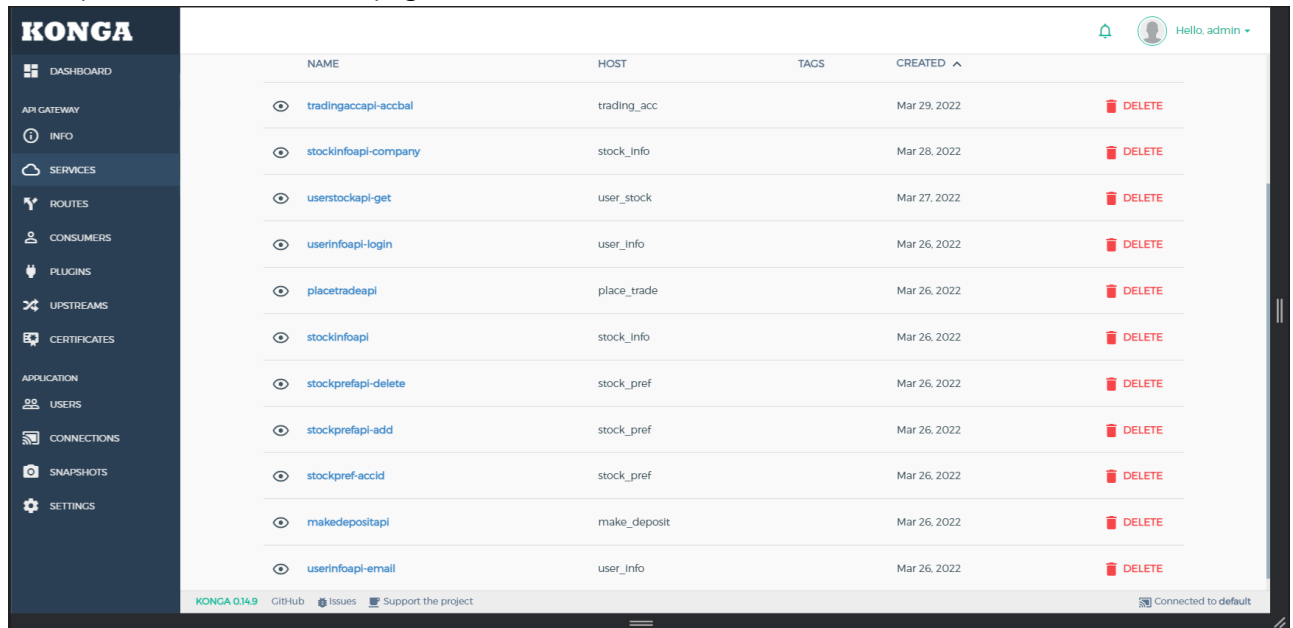
Name: default

Kong Admin URL: http://kong:8001



The image shows the Kong Admin dashboard. On the left is a dark blue sidebar with the Kong logo and a menu: DASHBOARD, APPLICATION, USERS, CONNECTIONS, SNAPSHOTS, and SETTINGS. The main content area has a green circular refresh icon and the text "Welcome! First of all, let's setup a connection to Kong Admin. Select a connection type." There are three tabs: "DEFAULT" (selected), "KEY AUTH", and "JWT AUTH". Under the "DEFAULT" tab, there is a blue box with text: "Kong will connect directly to Kong's admin API. This method is mainly suitable for demo scenarios or internal access (ex. localhost). Kong's admin API **should not** be publicly exposed." Below this are two input fields: "Name" with the value "default" and "Kong Admin URL" with the value "http://kong:8001". A green "CREATE CONNECTION" button is at the bottom. The footer shows "KONGA 0.14.7", "GitHub", "Issues", "Support the project", and "Connected to N/A".

13. Next, proceed to the services page.

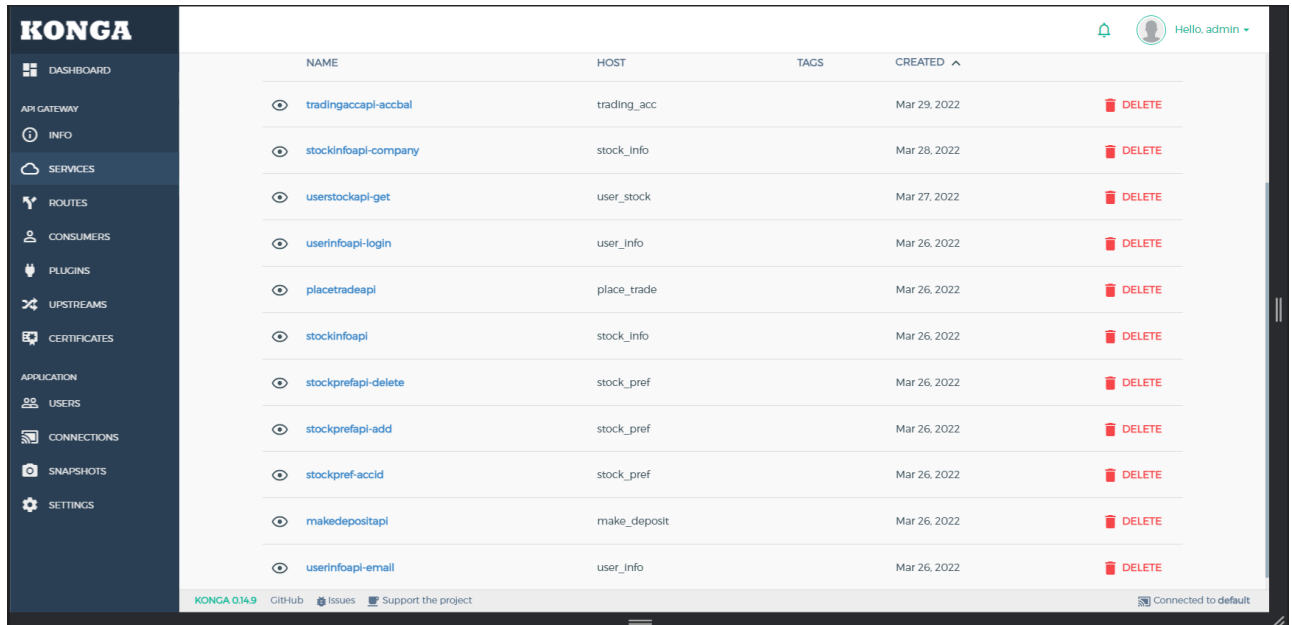


\*It will be blank since no services has been added

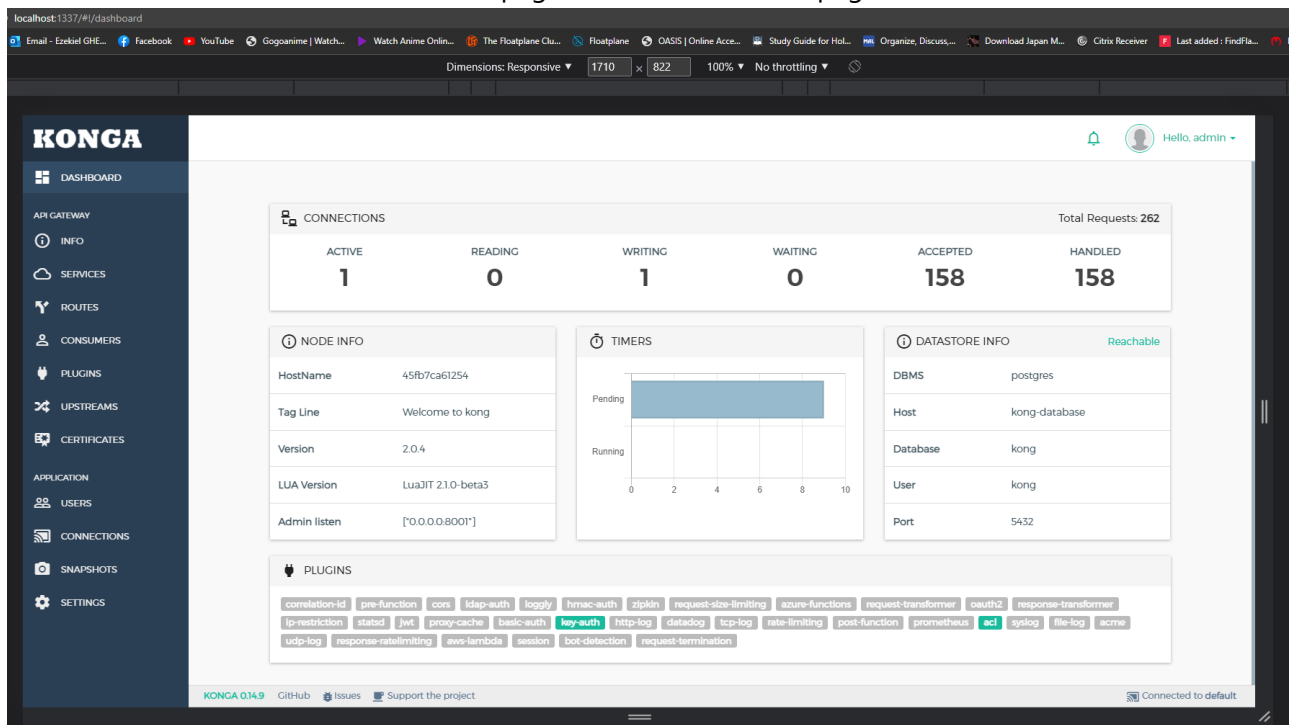
14. Create all this services and routes as stated in the table

(can refer to the included word document: "KONG-Setup.docx"):

Service Name	Service Details	Route Details	Has Key Auth?
userinfoapi-email	URL: http://user_info:5006/account/em ail/	Path: /api/v1/account/email/ Methods: GET	<b>Yes</b>
makedepositapi	URL: http://make_deposit:5101/make_d eposit	Path: /api/v1/make_deposit Methods: POST	<b>Yes</b>
stockpref-accid	URL: http://stock_pref:5002/stock_pref/	Path: /api/v1/stock_pref/ Methods: GET	<b>Yes</b>
stockprefapi-add	URL: http://stock_pref:5002/stock_pref/ add/	Path: /api/v1/stock_pref/add/ Methods: POST	<b>Yes</b>
stockprefapi-delete	URL: http://stock_pref:5002/stock_pref/ remove/	Path: /api/v1/stock_pref/remove/ Methods: POST	<b>Yes</b>
stockinfoapi	URL: http://stock_info:5001/stock_info/	Path: /api/v1/stock_info/ Methods: GET	<b>Yes</b>
placetradeapi	URL: http://place_trade:5100/place_tra de	Path: /api/v1/place_trade Methods: POST	<b>Yes</b>
userinfoapi-login	URL: http://user_info:5006/account/logi n	Path: /api/v1/login Methods: GET	<b>No</b>
userstockapi-get	URL: http://user_stock:5007/user_stock /	Path: /api/v1/user_stock/ Methods: GET	<b>Yes</b>
stockinfoapi-company	URL: http://stock_info:5001/stock_info/ profile2/	Path: /api/v1/stock_info/profile2/ Methods: GET	<b>Yes</b>
tradingaccapi-accbal	URL: http://trading_acc:5004/trading_a cc/	Path: /api/v1/trading_acc/ Methods: GET	<b>Yes</b>

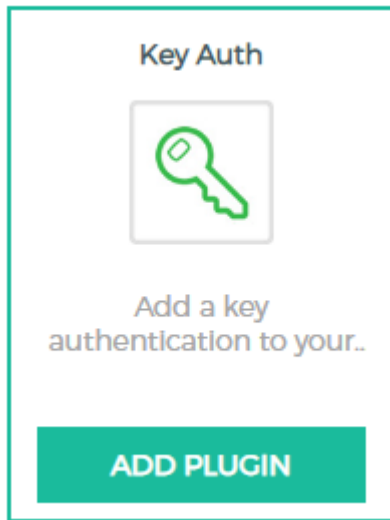


You will be redirected to the dashboard page which is the default page:



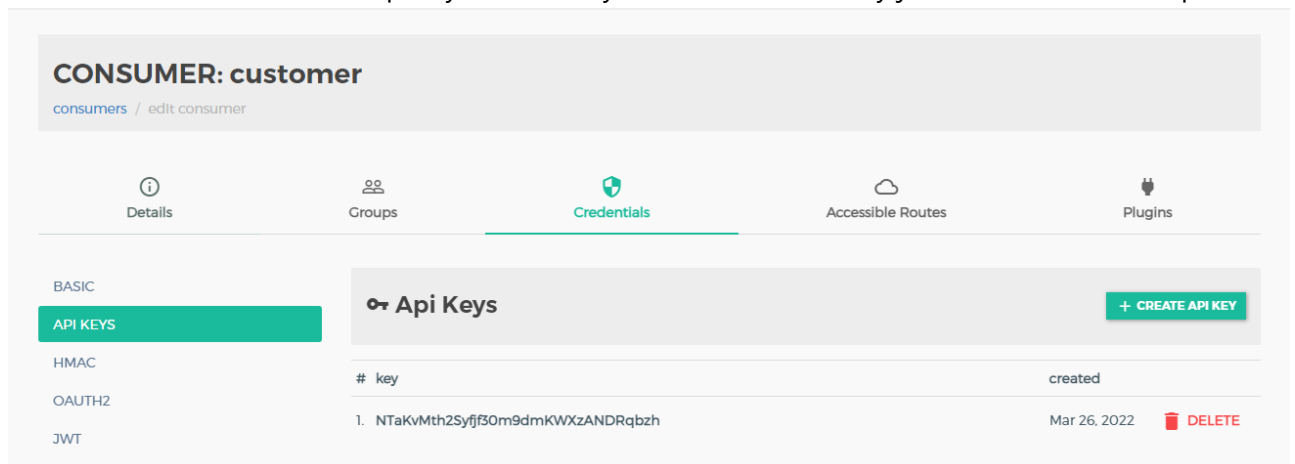
15. Next, go to the plugin tab for each service and add the Key Auth plugin for all services

EXCEPT LOGIN (userinfoapi-login).

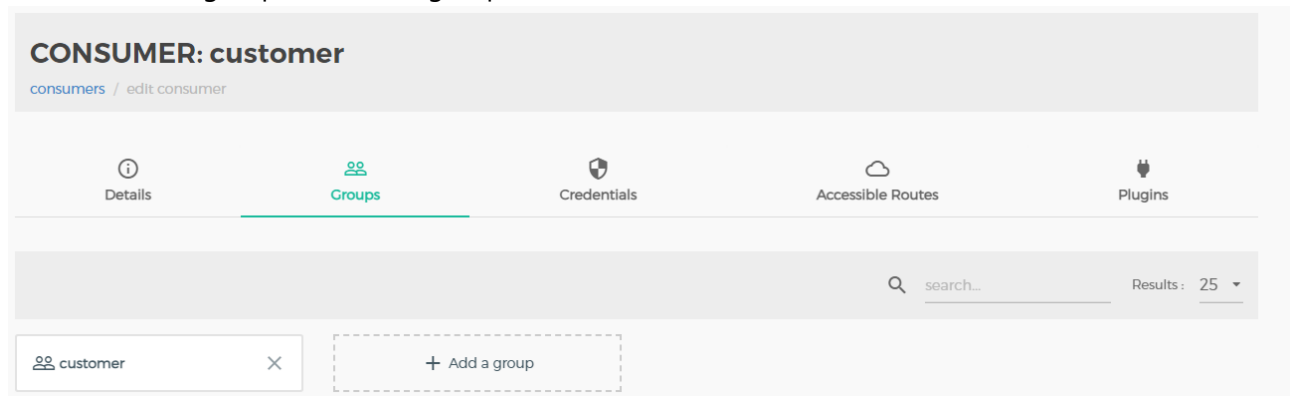


For each Key Auth, under key names, add a key called "apikey" and hit enter and click add plugin at the bottom.

16. Next, navigate to the consumer tab (left nav bar) and create a consumer called "customer" and under the credentials tab, create an api key with the key as this "NTaKvMth2Syjf30m9dmKWxzANDRqbzh".



17. Also, under the groups tab, add a group called "customer".



18. With this, you can open Postman and test the connection using this URL:  
`http://localhost:8000/api/v1/login?email=maryesther@gmail.com&password=dGVtcDE=`

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `http://localhost:8000/api/v1/login?email=maryesther@gmail.com&password=dGVtcDE=`
- Params:** email (maryesther@gmail.com), password (dGVtcDE=)
- Status:** 200 OK, Time: 37 ms, Size: 569 B
- Response Body (JSON):**

```

{
  "code": 200,
  "data": {
    "accid": 1000002,
    "apikey": "TlRhS3ZNdGgyU3lmanYzMG05ZG1LV1h6QU5EUUnFiemg=",
    "birthdate": "Sat, 03 Dec 1994 00:00:00 GMT",
    "email": "maryesther@gmail.com",
    "name": "Mary Esther",
    "password": "dGVtcDE=",
    "trade_accid": 4000002
  }
}

```

19. Lastly, once all has been configured or if you face any issues, please stop the all the microservices docker containers and start them again and wait for about 1 minute for all the services to complete starting up.

The screenshot shows the Docker Desktop interface with the following details:

- Containers / Apps:** A list of running containers under the 'microservices' namespace.
- Containers:**
  - microservices\_user\_info\_1** (markytan/user\_...): RUNNING, PORT: 5006
  - microservices\_kong-database\_1** (postgres:9.6): RUNNING, PORT: 5432
  - microservices\_user\_stock\_1** (markytan/user\_...): RUNNING, PORT: 5007
  - microservices\_trading\_acc\_1** (markytan/tradi...): RUNNING, PORT: 5004
  - microservices\_rabbitmq\_1** (rabbitmq:3-ma...): RUNNING, PORT: 15672
  - microservices\_stock\_pref\_1** (markytan/stock...): RUNNING, PORT: 5002
  - microservices\_email\_notification\_1** (markytan/email...): RUNNING
  - microservices\_error\_1** (markytan/error...): RUNNING
  - microservices\_transaction\_log\_1** (markytan/trans...): RUNNING, PORT: 5005

With this, you have successfully configured the backend services! You can proceed to setup the frontend services if you have not done so.

