**HOW TO RUN THE BACKEND CODE**

For window machine (if you’re using linux then you don’t have to do step 1)

Note: This tutorial will only help you to run the backend. If you wish to edit and write code, you will need to install IDE like Visual Studio Code and install the necessary extension. Please look for external tutorial for that. Highly recommend doing so.

**Step1: Do this tutorial to enable wsl and ubuntu on your window machine**

<https://www.toptechskills.com/windows-10-tutorials-courses/how-to-install-enable-windows-subsystem-for-linux-wsl-windows-10/>

Note: You can choose the latest version of ubuntu. The tutorial one is 18.04, the new one as I writing this tutorial is 20.04.

**Step 2: Run “sudo apt-get update”**

Text

Description automatically generated

**Step 3: Run “sudo apt-get upgrade”**

Text

Description automatically generated

Type “Y” if “Do you want to continue” is prompt

**Step 4: Install Java using “sudo apt install openjdk-11-jre-headless”**

Text

Description automatically generated

Again type “Y” if “Do you want to continue” is prompt

**Step 5: Install maven using “sudo apt install maven”**

Text

Description automatically generated

Again type “Y” if “Do you want to continue” is prompt (Well actually I don’t remember if this step has a prompt, just put it here just in case there is one lol)

**Step 6: Cloning the project using “git clone** [**https://github.com/s3783616/P107IAQ/**](https://github.com/s3783616/P107IAQ/)**” (or whichever git repository URL the code is reside)**

Text

Description automatically generated

**Step 7: Type “cd P107IAQ/BackEnd/data\_services/” to go into the microservice directory and type “mvn spring-boot:run”**

Text

Description automatically generated

It will take a couple minute for maven to install then necessary dependency but when it’s finished you will be greeted with this.

Graphical user interface

Description automatically generated

And that is it. Now there are two microservices in the BackEnd, data\_services and user\_services. The one run in the example is data\_services. If you want to run the user\_service, simply “cd” then “cd P107IAQ/BackEnd/user\_services” then “mvn spring-boot:run” and the user\_services will be up

If you wish to run both microservices, you will need to open two Ubuntu terminal (simply by looking for the Ubuntu icon in the window search bar)

**Step 8 (Final step): Press “Ctrl+C” to stop the microservices**

Graphical user interface, text

Description automatically generated

**API documentation**

Note: please learn how to use postman it’s a simple and effective tool to test the backend api.

Postman tutorial: <https://www.youtube.com/watch?v=VywxIQ2ZXw4&t=508s>. You only need to reach lesson 11 to be able to use postman for the purpose of testing this project backend. Probably take 1 hours.

**Data services**

**Getting device list:**

Endpoint : localhost:8081/api/data/list

Example:

Graphical user interface, text, application, email

Description automatically generated

**Getting data:**

Endpoint: http://localhost:8081/api/data/getData?deviceID=deviceID&dateFrom=dateFrom&dateTo=dateTo&dataType=dataType

Attribute table:

|  |  |
| --- | --- |
| deviceID | Type: int  Can be retrieve using the get device list api |
| dateFrom | Type: date time  Format: yyyy-mm-ddThh:mm:00 |
| dateTo | Type: date time  Format: yyyy-mm-ddThh:mm:00 |
| dataType | Type: string  Possible data type :[raw,5-min-avg,15-min-avg] |

Example of raw data:

Graphical user interface, text, application, email

Description automatically generated

Example of 5-minute average:

Graphical user interface, text, application, email

Description automatically generated

Example of 15-min average:

Graphical user interface, text, application, email

Description automatically generated

**Getting 1 minute average data:**

Endpoint: <http://localhost:8081/api/data/1minInterval?deviceID=deviceID&dateFrom=dateFrom&dateTo=dateTo>

Attribute table: same as the previous table except there is no datatype

Example:

Graphical user interface, text, application

Description automatically generated

**USER SERVICES**

**Getting list of all users:**

Endpoint: localhost:8080/api/users/

Example: Graphical user interface, text, application, email

Description automatically generated

**Registering:**

Endpoint: localhost:8080/api/users/register

Example:

Graphical user interface, text, application

Description automatically generated

**Login:**

Endpoint: localhost:8080/api/users/login

Example:

Graphical user interface, application

Description automatically generated

**POSTSCRIPTS**

To future developer, after the project ended, I would have terminated the AWS Relation Database Service required for user services. If you want to test the user services ,you will need to create a new database. It can be either local or on any cloud platform.

**THANK YOU FOR READING THIS TUTORIAL**