API for Bank using Python

Web APIs are tools for making information and application functionality accessible over the web server. In other words it is a programmatic way of interfacing third party systems and communication between different domains.

API Creation:

This API is designed for the following purpose:

1. Your data set is large, making download via FTP is not the best way.
2. Your users will need to access your data in real time, such as for display on another website or as part of an application.
3. Your users only need access to a part of the data at any one time.

The API that we created can be used to get

1.All the customer details of the bank

2.Customer details of a particular branch

2.All the branches of the bank

The data for our API is fetched from the database customerData.db which is already loaded with the customer_details.csv.

The csv has more than 26000 records and it has columns named Person,Branches,Employee Annual Salary which are respectively the names of the customer,the branch in which they have the account and their annual salary.

Prerequisites:

Sqlite3 db

python.The python we used is python 3.6.5

How to run the API

Step 1:

Download all the files from the location

https://github.com/priku577/API/

Step 2:

Once downloaded naviagate to this location through the terminal and run the follwing command

pip install -r requirements.txt

This will make the life easier beacuse all the libraries required to run **app.py** will be automatically installed while running this single line .Installing each library will be a tedious taks.

Since the database is already loaded with the csv that step can also be avoided.

Step 3:

Now run the command

python app.py.

Once the server is started click the below links

http://localhost:5000/Branches/

http://localhost:5000/StockholmBranch/

http://localhost:5000/KirunaBranch/

http://localhost:5000/completeData/

API files description

The API that we created has the following files:

1.customer_details.csv

This is a dummy data created by us with more than 26000 records .The csv has the name,branch and salary of the customer.

2.customerData.db

This is the database our API uses.The database is loaded with the csv as an initial step.

3.requirements.txt

This file has all the libraries need to run the app.py.The librabries will be installed when we run the requirements.txt.

4.app.py

This has the api code .When we run the app.py the API starts running on the localhost,port 5000.Now when we want to get the data of all the distinct branches where the customer has the account we just click the url

http://localhost:5000/Branches/

and what happens in the background is using the api.add_resource in app.py it find the function that is connected with the url and inside the function we have written the select query. This gets executed and the result is displayed in json format.The **flask_restful** classes help to map the function with the API URL .It is the **sqlalchemy** that helps to connect to the databse to do the select operations.

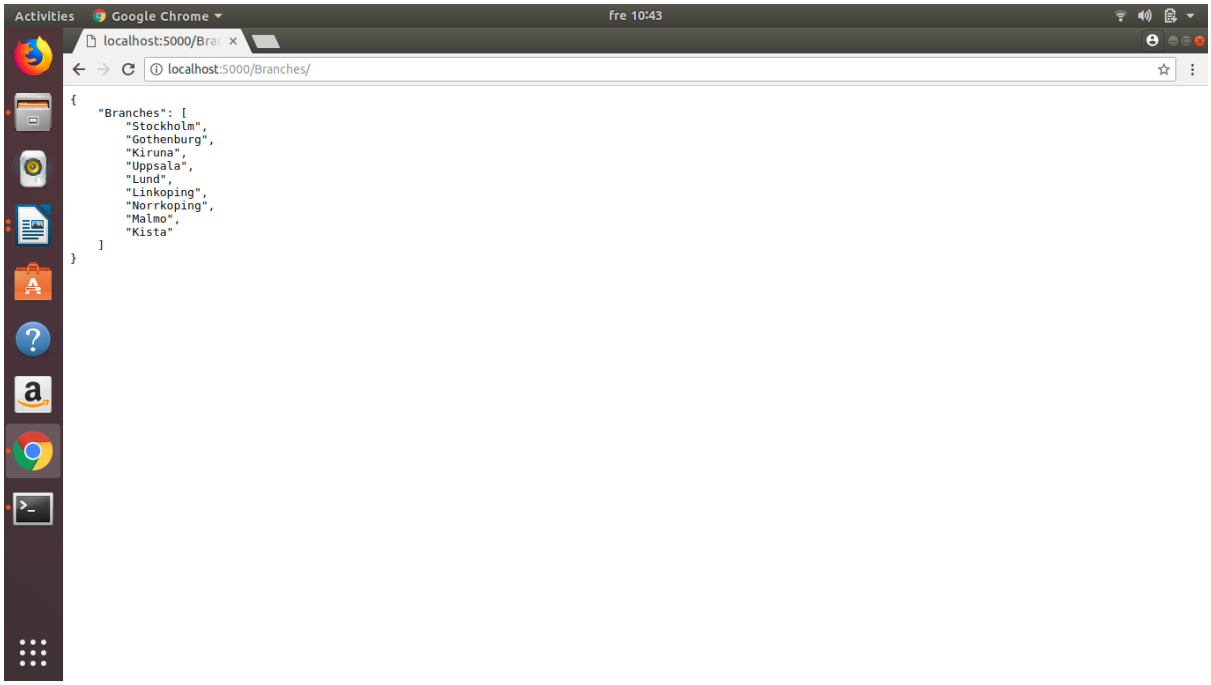
The output we created is in json format.

Note:

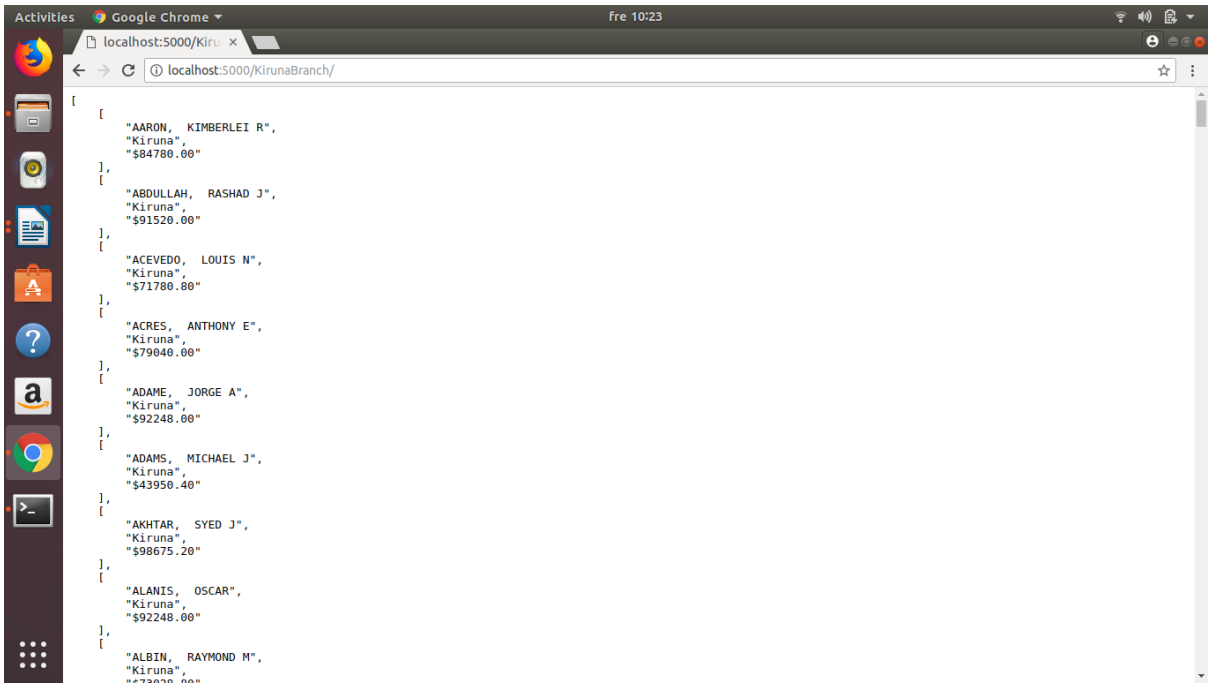
We have not worked with the front end as the idea was to create a succesfull API.When the end user is not a programmer , we need to put effort for front end which will be done in the next work.

Results:

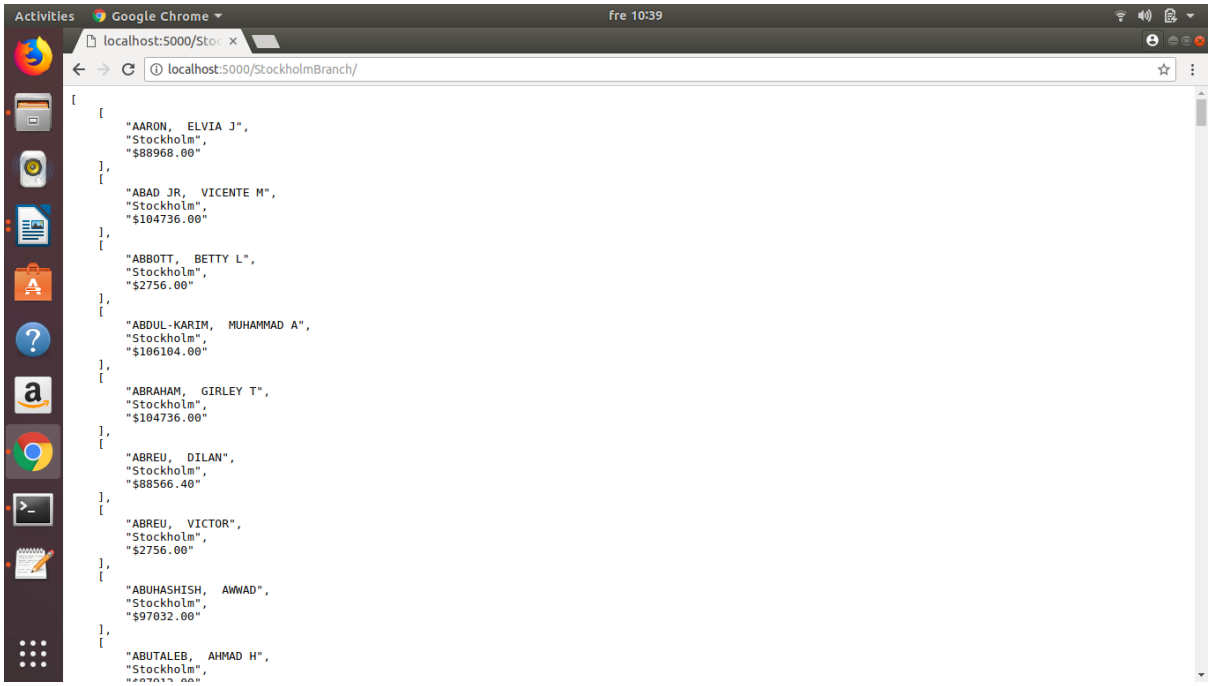
http://localhost:5000/Branches/



http://localhost:5000/KirunaBranch/

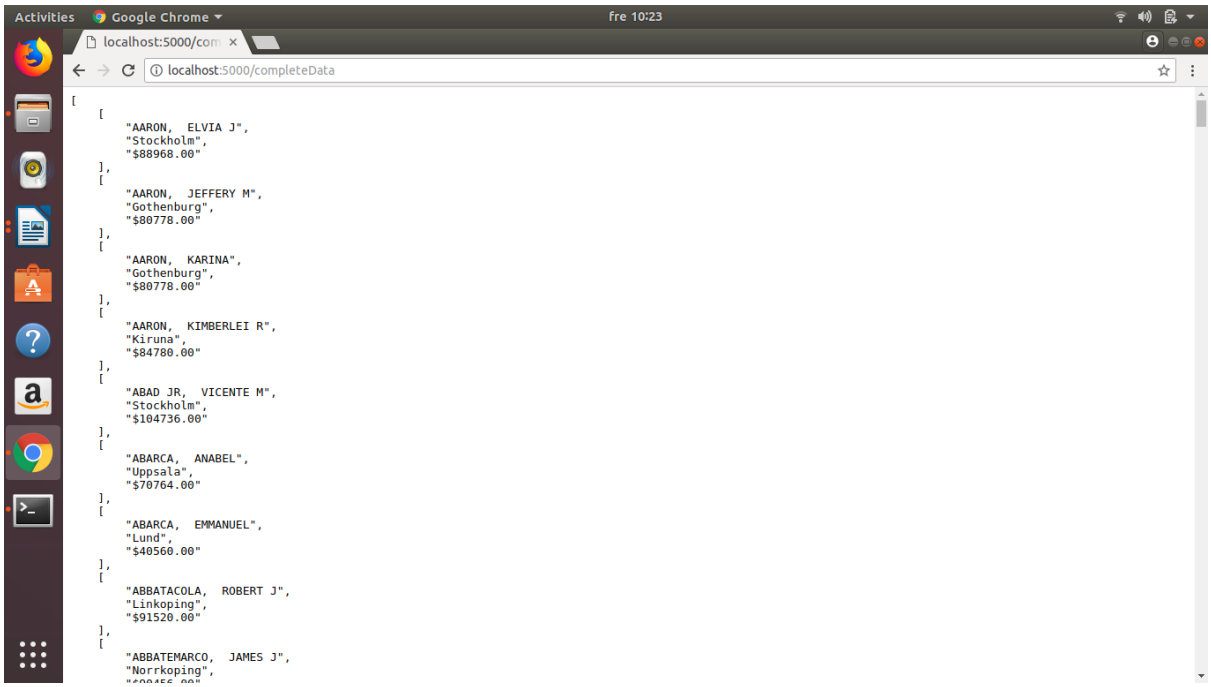


http://localhost:5000/StockholmBranch/




```
[
    [
        "AARON,  ELVIA J",
        "Stockholm",
        "$88968.00"
    ],
    [
        "ABAD JR,  VICENTE M",
        "Stockholm",
        "$104736.00"
    ],
    [
        "ABBOTT,  BETTY L",
        "Stockholm",
        "$2756.00"
    ],
    [
        "ABDUL-KARIM,  MUHAMMAD A",
        "Stockholm",
        "$106104.00"
    ],
    [
        "ABRAHAM,  GIRLEY T",
        "Stockholm",
        "$104736.00"
    ],
    [
        "ABREU,  DILAN",
        "Stockholm",
        "$88566.40"
    ],
    [
        "ABREU,  VICTOR",
        "Stockholm",
        "$2756.00"
    ],
    [
        "ABUHASHISH,  AWWAD",
        "Stockholm",
        "$97032.00"
    ],
    [
        "ABUTALEB,  AHMAD H",
        "Stockholm",
```


http://localhost:5000/completeData/




```
[
    [
        "AARON,  ELVIA J",
        "Stockholm",
        "$88968.00"
    ],
    [
        "AARON,  JEFFERY M",
        "Gothenburg",
        "$80778.00"
    ],
    [
        "AARON,  KARINA",
        "Gothenburg",
        "$80778.00"
    ],
    [
        "AARON,  KIMBERLEI R",
        "Kiruna",
        "$84780.00"
    ],
    [
        "ABAD JR,  VICENTE M",
        "Stockholm",
        "$104736.00"
    ],
    [
        "ABARCA,  ANABEL",
        "Uppsala",
        "$70764.00"
    ],
    [
        "ABARCA,  EMMANUEL",
        "Lund",
        "$40560.00"
    ],
    [
        "ABBATACOLA,  ROBERT J",
        "Linkoping",
        "$91520.00"
    ],
    [
        "ABBATEMARCO,  JAMES J",
        "Norrkoping",
```