RESTful Web Services with Python Flask

In this assignment exercise I have created a in memory JSON DB to store and manipulate a simple employee database and develop RESTful APIs to perform CRUD operations using GET, POST, PUT methods.

I have developed the below APIs -

* GET /empdb/employees' – Retrieve all the list of employees from the DB
* GET //empdb/employee/<empId>'– Retrieve the details of given employee Id
* POST //empdb/employee/create'– Create a record in the employee DB, where as the employee details are sent in the request as a JSON object
* POST /rolesDB/Roles/create -Create a record in the Roles DB, where as the Roles details are sent in the request as a JSON object
* GET /empdb/Roles -Retrieve the details of all Roles
* PUT //empdb/employee/assignrole/<empId>' Update the Associate a Role to a employee user with the given details of employee in the data part as a JSON object

# Below are Installing packages to install before start testing the utility

Python 3.8.2

Flask==2.1.1

Jinja2==3.0.3

unitest==1.3.8

urllib3==1.26.9

After installing above package verify flask server is running successfully.

Execute *Python app.py*

"D:\OneDrive - Infosys Limited\2022\python-rest-api\Scripts\python.exe" C:/Users/pritesh.modi01/Desktop/UBS\_Assignment/Employee\_db\_api/app.py

\* Running on http://127.0.0.1:5000 (Press CTRL+C to quit)

\* Serving Flask app 'app' (lazy loading)

\* Environment: production

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

\* Debug mode: off these commands, verify application by opening a browser and navigating to http://127.0.0.1:5000/ or by issuing curl http://127.0.0.1:5000/.

All the integration tests are developed and unit tested.

Execute the testcases by running following command

*Python C:/Users/pritesh.modi01/Desktop/UBS\_Assignment/Employee\_db\_api/test.py*

As an API User,

Given the User Service

When I call the URL mapping, and pass a new User JSON, it’s created on the Service

def test\_employee\_create(self)

As an API User,

Given the User Service

When I call the URL mapping, and pass a new Role in JSON, it’s created on the Service

def test\_role\_create(self)

As an API User,

Given the User Service

When I call the third URL mapping, I get a list of Roles

def test\_all\_roles(self):

As an API User,

Given the User Service

When I call the URL mapping, I get a list of employee Users

def test\_all\_employee(self)

As an API User,

Given the User Service

When I call the URL mapping, and pass a new Role  JSON, it’s created on the Service

def test\_assign\_role(self)

As an API User,

Given the User Service

When I call the fourth URL mapping, when I search for specific employee details, I get a details of employee

def test\_employee\_details(self):