Restful API And Microservices with Python

Day 9

Day 9 - Overview

- JWT Optional
- Create an optional JWT API to search for user - Task
  - Create an User Search API that takes email as request parameter
  - User with token can access complete details
  - User without token can only access name and email
- Customizing Flask-JWT-Extended callbacks and responses
- API logging

Prerequisite

- VM with windows OS
- Python 3.8 or >
- Visual Studio Code - Code Editor
- Postman
- GIT

https://github.com/saurav-samantray/flask-microservices-training/blob/main/slides/Setup%20GIT%20in%20your%20Local%20system.pdf

- Docker - Not Mandatory for current training

Sync your fork for Day 9 activities

- Follow the below document to sync your fork and update local repository.

  https://github.com/saurav-samantray/flask-microservices-training/blob/main/slides/Setup%20GIT%20in%20your%20Local%20system.pdf

Local Setup for Day 9

- Navigate to the below folder

  C:\workspace\flask-microservices-training\day9\user-management-service

- Create a virtual environment and activate it

  python -m venv venv

  .\venv\Scripts\activate

- Install the dependencies, initialize DB and start server

  pip install -r requirements.txt

  python init_db.py

  python server.py

JWT Optional

- Some APIs can be partially protected with the below decorator

  jwt_required(optional=True)

- This functionality can be used to create APIs that can be accessed by both authenticated and Anonymous users.
- What information is exposed to each can be determined based on the logic

User Search API - Task

- Create a new API in the file api/users_search_api.py
- Update the GET call to be accessible by both authenticated and non authenticated users
- Provide a URL mapping for the new API
- The GET call should accept a mandatory request parameter called email. If not found in the request reject return the call with a 400 error.
    - **email = request.args.get('email')**
- If the email request param is found, query the database for the user.
- If not found in database return not found error
- If found
    - For authenticated user give all details available in database
    - For Anonymous user only return name and email

Flask JWT Callbacks

- Callbacks are post processors. Any action that we need to take after something has happened.
- We will use the Flask JWT Extended libraries callback methods to customize the error messages

Flask Logging

- Managing logging in a production grade application is very important
- Excessive logging, fewer logging or unformatted logging are all biggest bottlenecks when debugging an application
- Understanding various log level
    - DEBUG
    - INFO
    - WARNING
    - ERROR
    - CRITICAL
- How to configure the log level for flask application

Q and A