**Python Homework**

**Connect to below MongoDB and implement below APIs through FastAPI.**

* You can add log you need in your functions.
* Documentation for each API
  + You can access fastAPI swagger/docs by http://{ip:port}/docs#/
  + You have to explain the purpose of each API and provide examples with below information.
    1. title: The title of the API
    2. summary: A short summary of the API
    3. description: A short description of the API
    4. request example and response example of the API
  + **Reference and hint:**
    1. <https://fastapi.tiangolo.com/tutorial/metadata/#metadata-and-docs-urls>
    2. <https://fastapi.tiangolo.com/tutorial/schema-extra-example/>
    3. <https://fastapi.tiangolo.com/tutorial/response-model/>

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DB Information**  **URL** : 172.18.212.131:27019,172.18.212.132:27019  **Database** : training  **User/password** : test\_user/pega#1234  **Collection**: UserInfo  **Example Data in UserInfo Collection**   |  |  | | --- | --- | | username | wanyu\_lee | | email | Wanyu\_Lee@pegatroncorp.com | | ext | 36714 | | active | true | | createAt | ISODate("2023-10-04T14:56:24.276+08:00") | |

1. **Add New User : Create a user and insert into DB**

|  |  |
| --- | --- |
| **URL** | */user* |
| **Method** | POST |
| **Content Type** | application/json |
| **Data Parameter** | |  |  |  |  | | --- | --- | --- | --- | | **Key** | **Type** | **Description** | **Required** | | username | string | User name | Y | | email | string | User’s email | Y | | ext | string | User’s extension | Y | |
| **Success Response:** | **Status Code** : 201 **Content :**  {“msg”:” Add new user succeed.”,  } |
| **Error Response:** | **Scenario:** User didn’t input data parameter or the input data type wrong (also need to check the email format) **Status Code: 400/422 (FastAPI default)**  **Content:** {“msg”:”Wrong input parameters.”}  **Scenario:** User already exists (Check active user only)  **Status Code: 409**  **Content:** {“msg”:”User already exists.”}  **Scenario:** Add failed.  **Status Code: 500**  **Content:** {“msg”:”Add new user failed.”} |
| **Sample Call** | curl -X POST  -d ‘{“username”:”wanyu\_lee”,”email”:[Wanyu\_Lee@pegatroncorp.com](mailto:Wanyu_Lee@pegatroncorp.com),”ext”:”36714”} ’  -H "Content-Type: application/json"  /user  **Response :**  {“msg”:” Add new user succeed.”} |

* **Hint: FastAPI BaseModel**

1. **Get User Information**

|  |  |
| --- | --- |
| **URL** | */user/{user\_name}* |
| **Method** | GET |
| **URL Params:** | **N/A** |
| **Success Response:** | **Status Code** : 200 **Content :**  {  “username”: “{user\_name}”,  “email”:”{user\_email}”.  “ext”:”{user\_ext}”  } |
| **Error Response:** | **Scenario:** User did not input user name **Status Code: 400/422 (FastAPI default)**  **Content:** {“msg”:” Wrong input parameters.”}  **Scenario:** User not exist  **Status Code: 404**  **Content:** {“msg”:”User does not exist.”}  **Scenario:** Get user info failed  **Status Code: 500**  **Content:** {“msg”:” Get {user\_name}’s data failed.”} |
| **Sample Call** | curl -X GET /user/wanyu\_lee  **Response :**  {"username":"wanyu\_lee",  "email":”[Wanyu\_Lee@pegatroncorp.com](mailto:Wanyu_Lee@pegatroncorp.com).”  “ext”: ”36714”  } |

1. **Update User Data**

|  |  |
| --- | --- |
| **URL** | */user/{user\_name}* |
| **Method** | PUT |
| **Content Type** | application/json |
| **Data Parameter** | |  |  |  |  | | --- | --- | --- | --- | | **Key** | **Type** | **Description** | **Required** | | email | string | User’s email | N | | ext | string | User’s extension | N | |
| **Success Response:** | **Status Code** : 200 **Content :**  {“msg”:” Update {user\_name}’s data succeed.”} |
| **Error Response:** | **Scenaro:** User didn’t input data parameter **Status Code: 400/422**  **Content:** {“msg”:”Wrong input parameters.”}  **Scenario:** User not exist  **Status Code: 404**  **Content:** {“msg”:”User does not exist.”}  **Scenario:** Update failed.  **Status Code: 500**  **Content:** {“msg”:” Update {user\_name}’s data failed.”} |
| **Sample Call** | curl -X PUT  -d ‘{“ext”:”36715” } ’  -H "Content-Type: application/json"  /user/wanyu\_lee  Response: {“msg”:” Update wanyu\_lee’s data succeed.”} |

1. **Delete User**

|  |  |
| --- | --- |
| **URL** | */user/{user\_name}* |
| **Method** | DELETE |
| **Success Response:** | **Status Code** : 200 **Content :**  {“msg”:” Delete user{user\_name} succeed.”} |
| **Error Response:** | **Scenario:** User didn’t input data parameter **Status Code: 400/422**  **Content:** {“msg”:”Wrong input parameters.”}  **Scenario:** User not exist  **Status Code: 404**  **Content:** {“msg”:”User does not exist.”}  **Scenario:** Delete failed.  **Status Code: 500**  **Content:** {“msg”:” Delete {user\_name} Failed.”} |
| **Sample Call** | curl -X DELETE /user/wanyu\_lee  **Response:**  {“msg”:” Delete wanyu\_lee succeed.”} |

* **Do not delete collection data directly. Mark user’s active field as false.**

**Docker Homework**

Dockerized your FastAPI project above.

1. **Write a Dockerfile**
   * Add all packages you need and FastAPI project above
   * Timezone : Asia/Taipei
   * Export a port not used on testnode05.
2. **Build Image**
   * Image name: ctrdc-dev-dockerhub.pegatroncorp.com/training\_intern/{your OA name}:latest
3. **Push image to Harbor**
   * docker login ctrdc-dev-dockerhub.pegatroncorp.com
   * docker push ctrdc-dev-dockerhub.pegatroncorp.com/training\_intern/{your OA name}:latest
4. **Write a docker-compose file to run your image**
   * Container name: {your name}
   * network\_mode: bridge
   * Expose port
   * Setup cpu and memory **limit** for your container.
5. Try to revise the dockerfile with **multi stage** way and check is there any change on docker **image size.**