

Assignment

AltoTech

SET UP PROJECT

จากโจทย์ให้ทำ property management API tool ที่ใช้จะมีดังนี้ครับ

- **Django** – ใช้ทำ API ทั้งระบบและ manage เรื่อง user และ permission
- **MongoDb** - ใช้เก็บข้อมูล work order เพื่อให้ง่ายต่อการแก้ไขและเปลี่ยนแปลงจึงเลือกใช้ mongo Db ครับ
- **OpenAi** - ใช้ integrate กับ ai ตาม assignment ในข้อ 2 ครับ

1. Clone project มาไว้ในเครื่อง
2. Install MongoDB ก่อนและสร้าง database และ สร้าง collection
3. Run mongoDb
4. ไปที่ file project/hotels/hotels/settings.py
5. แก้ไข connection string และ collection name ให้ตรงกับที่สร้างไว้

```
MONGO_DB_CONNECTION_STRING = "mongodb://localhost:27017"  
MONGO_DB_NAME = "hotels"
```

6. แก้ไข openai api key

```
CHAT_GPT_KEY = "chat-gpt-key"
```

7. Set up project ด้วยการ run file setup.bat จะ install lib ที่ต้องการและลงข้อมูลพื้นฐานตัว permission
8. สามารถ run server ได้ที่ file run-server.bat หาก run สำเร็จจะเป็นดังรูป

```
C:\Users\Work\Documents\altoTech\project>call .\env\Scripts\activate.bat

(env) C:\Users\Work\Documents\altoTech\project>cd hotels

(env) C:\Users\Work\Documents\altoTech\project\hotels>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 20, 2023 - 11:30:15
Django version 4.2.5, using settings 'hotels.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

9. Run unit test ได้ที่ file run-unittest.bat หาก run สำเร็จจะเป็นดังรูป

```
C:\Users\Work\Documents\altoTech\project>call .\env\Scripts\activate.bat

(env) C:\Users\Work\Documents\altoTech\project>cd hotels

(env) C:\Users\Work\Documents\altoTech\project\hotels>python manage.py test
Found 17 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.....
-----
Ran 17 tests in 7.590s

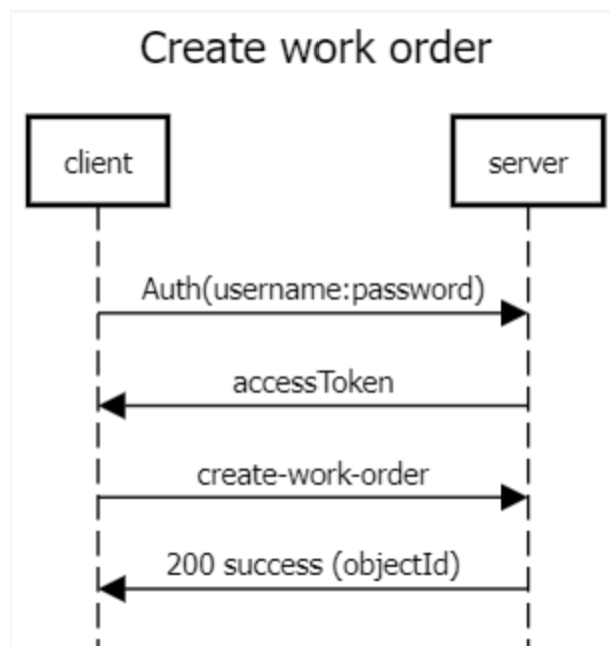
OK
Destroying test database for alias 'default'...

(env) C:\Users\Work\Documents\altoTech\project\hotels>cmd /k
(env) C:\Users\Work\Documents\altoTech\project\hotels>
```

10. มี file สำหรับทดสอบ request ของ postman ชื่อว่า hotels-module.postman_collection

ASSIGNMENT I

จากโจทย์ที่ได้รับผมออกแบบให้ manage เรื่อง permission ในการ create work order แต่ละ type แต่ละ user type ผ่านทาง token จากการ authentication นะครับ เพราะเห็นว่ามีกรเก็บ user จึงคิดว่าน่าจะต้องมีการเข้าระบบของ user อยู่แล้ว และทำให้ API create work order ไม่ต้อง access database เพื่อเช็ค permission ทุกครั้ง เพราะได้รับ permission มาตั้งแต่ตอน authenticate แล้ว และข้อดีอีกอย่างคือ เราสามารถแก้ไข permission ของ user ได้เลยว่าจะให้ permission ใดกับ user type ใดบ้าง ผ่านทาง database เมื่อ user login ใหม่ ก็จะได้รับ permission ชุดใหม่ แทนที่ไม่ต้องมาแก้ที่ code flow ในการ test จะเป็นแบบนี้ครับ



เริ่มต้นทดสอบ

1. create user สำหรับการทดสอบโดยส่ง parameter ดังนี้

```
{
  "username": "testMaidSupervisor",
  "password": "OdiskeP",
  "user_type": 1
}
```

ASSIGNMENT

AltoTech

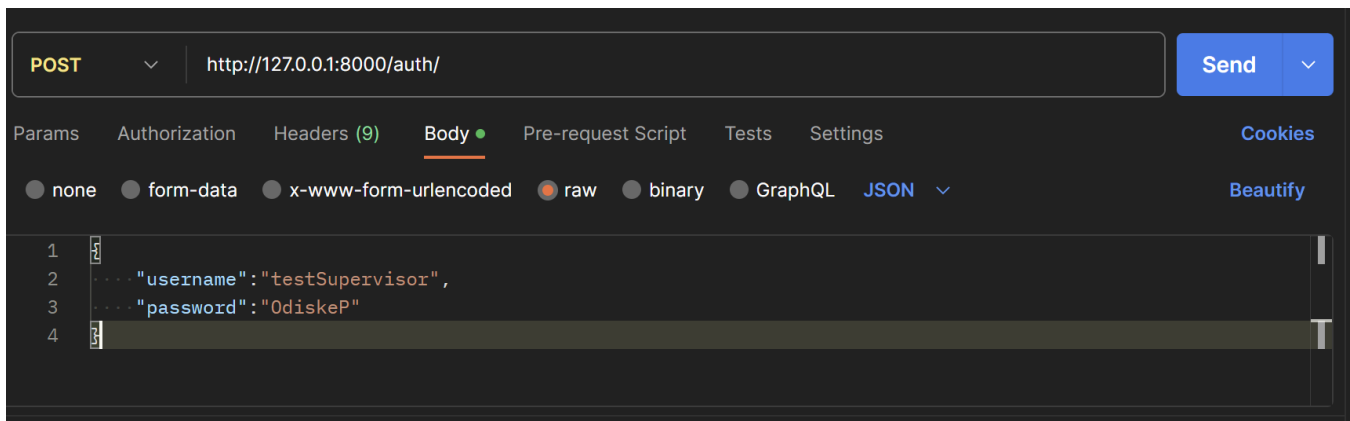
user_type เป็น enum value ตามนี้ครับ

```
ADMIN = 0
MAID_SUPERVISOR = 1
SUPERVISOR = 2
GUEST = 3
```

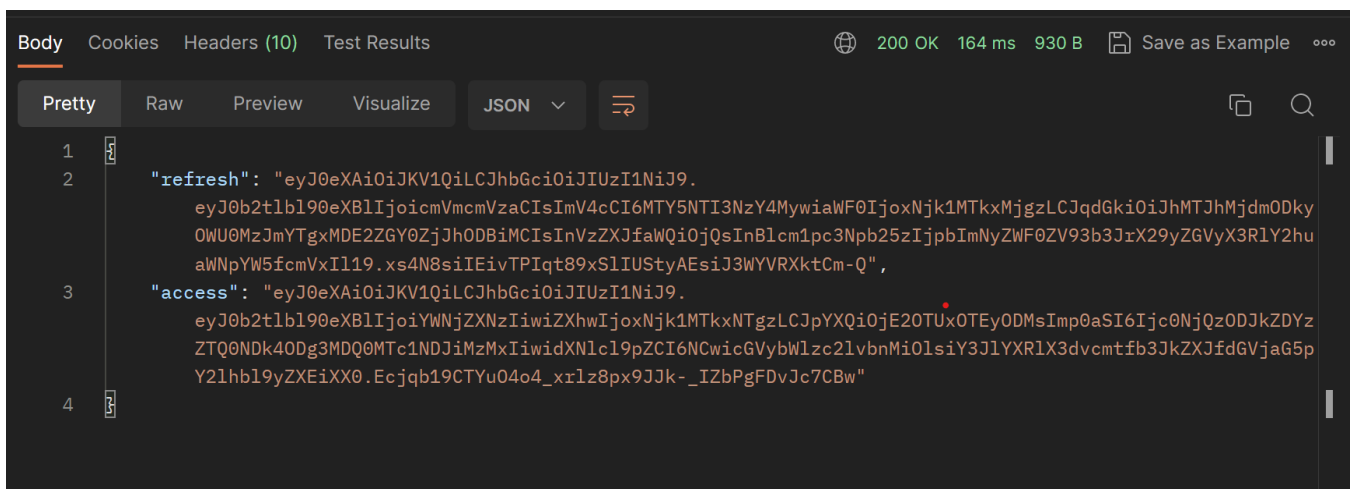
ส่ง request ไปที่ url: <http://127.0.0.1:8000/users/>

หากสำเร็จก็จะได้ response กลับมาเป็น 201 Created

2. Auth ส่ง username และ password เพื่อ ขอ access token ไปที่ url: <http://127.0.0.1:8000/auth/>



สำเร็จจะได้ token กลับมา



*เนื่องจากมี หลายส่วนที่ผมไม่แน่ใจ บาง field จะเป็น optional บาง field ก็จะถูกระบบใส่ให้เองนะครับ

Created By (User) - จะถูกใส่จากใน server เองจากที่ auth ไว้ทำให้ไม่ต้องส่งใน parameter นี้

Room - required

Started At – จะถูกใส่อัตโนมัติเมื่อเปลี่ยน status == IN_PROGRESS

Finished At – จะถูกใส่อัตโนมัติเมื่อเปลี่ยน status == Done

Type (Cleaning, Maid Request, Technician Request, Amenity Request) - required

Status (Created, Assigned, In Progress, Done, Cancel) - ตอน create จะถูกใส่เป็น

CREATED โดยอัตโนมัติ

POST

http://127.0.0.1:8000/work-orders/

Send

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

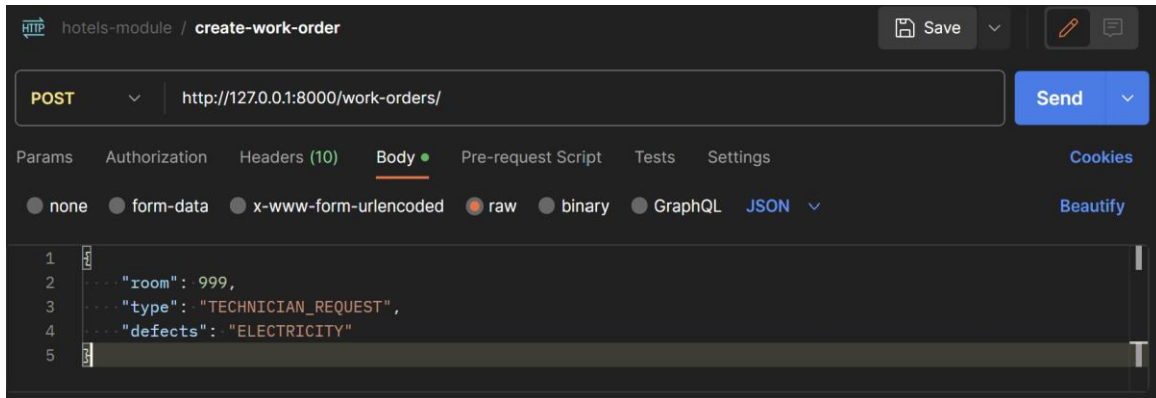
Cookies

Headers

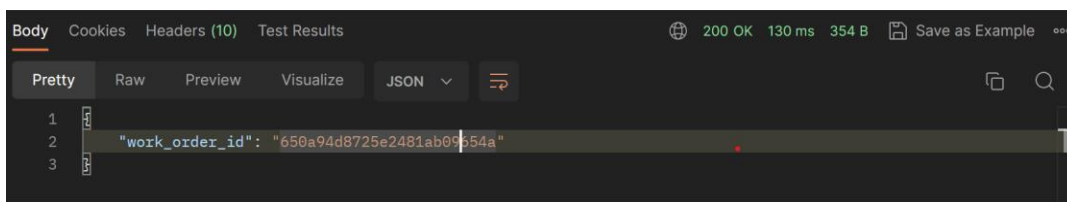
8 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ0b2t1bi90eXBlljoiYWlnZnNziwIZXhwlgxNjk1MTkyNTYyLmFkbG9yZWQ1ZS50OTUyNjlsmp0aSI6IyYyZmM4MDhjNjJiYzQ2NDFlmRjYjZkZWlznThlMGNIiwidNlcl9pZC1				
<input checked="" type="checkbox"/>	Content-Type					
	Key		Description			

ใส่ parameter



สำเร็จจะได้ id มาเพื่อไว้อ้างอิงในการ update



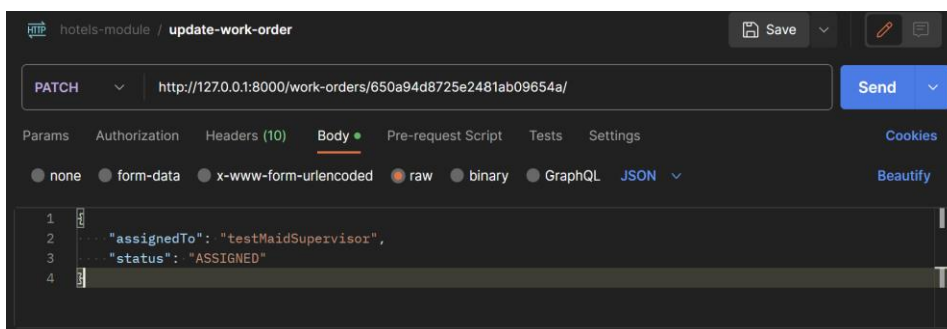
ตรวจสอบใน MongoDB Compass

```
_id: ObjectId('650a94d8725e2481ab09654a')
status: "CREATED"
room: 999
type: "TECHNICIAN_REQUEST"
defects: "ELECTRICITY"
createdBy: "testSupervisor"
```

4. Update work order

*ในส่วนนี้ผมไม่ทราบ permission ว่า user ไหน สามารถ update work order type ไหนหรือ status ไหนได้บ้าง หรือมี sequence ของ status work order ไหม จึงไม่ได้ใส่ logic ในส่วนนี้ไว้จนครบ

Request patch method ไปที่ url: <http://127.0.0.1:8000/work-orders/work-order-id/> โดยใส่ work_order_id ใน path parameter และใส่ access token ใน header ดังข้อที่แล้ว

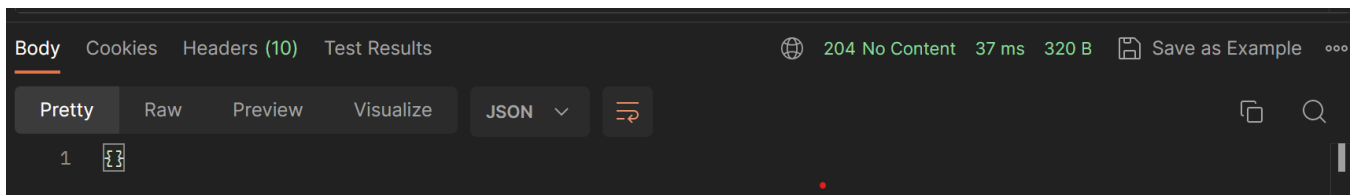


ASSIGNMENT

AltoTech

*assignedTo parameter ต้องเป็น username ที่ create ขึ้นจริงๆด้วยนะครับ

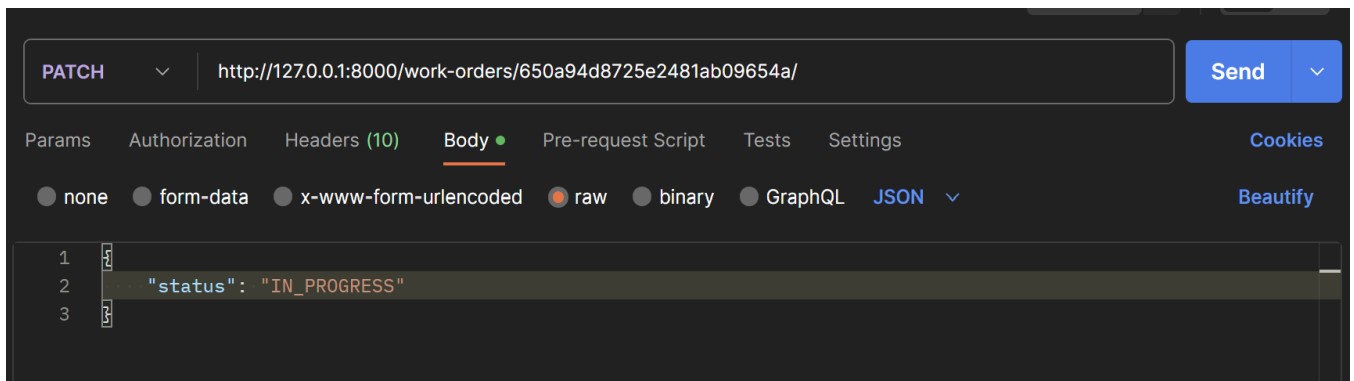
หากสำเร็จจะ return code 204



ตรวจสอบใน mongo db

```
_id: ObjectId('650a94d8725e2481ab09654a')
status: "ASSIGNED"
room: 999
type: "TECHNICIAN_REQUEST"
defects: "ELECTRICITY"
createdBy: "testSupervisor"
assignedTo: "testMaidSupervisor"
```

ลอง update status เป็น IN_PROGRESS



ตรวจสอบใน mongo db จะเห็น field startedAt เพิ่มมา

```
_id: ObjectId('650a94d8725e2481ab09654a')
status: "IN_PROGRESS"
room: 999
type: "TECHNICIAN_REQUEST"
defects: "ELECTRICITY"
createdBy: "testSupervisor"
assignedTo: "testMaidSupervisor"
startedAt: 2023-09-20T14:00:43.693+00:00
```

ASSIGNMENT 2

1. ChatGPT Integration ผมทำหน้าไว้ทดสอบที่ url: <http://127.0.0.1:8000/chatGPT/> มีช่องให้ใส่ chat และปรับ temperature

A.I WEB ASSISTANT

New Chat +

System: You are now chatting with a user, provide them with comprehensive, short and concise answers.

Temperature:

0.1

GENERATE

2. ทดสอบเรียกดู work order โดยการส่ง post request ไปที่ url: <http://127.0.0.1:8000/work-orders-openai/> โดยส่ง parameter เป็น command เช่น เรียกดูข้อมูลที่ต้องการ

ASSIGNMENT

AltoTech

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/work-orders-openai/
- Body (JSON):**

```
{  "command": "I want to get data that room number = 999 and status = CREATED"}
```
- Response (JSON):**

```
{  "_id": "65070019a5906618c8a9fa41",  "status": "CREATED",  "room": 999,  "type": "TECHNICIAN_REQUEST",  "defects": "ELECTRICITY,AIR_CON"}
```
- Status:** 200 OK, 1601 ms, 437 B

หรือ รวมข้อมูล

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://127.0.0.1:8000/work-orders-openai/
- Body (JSON):**

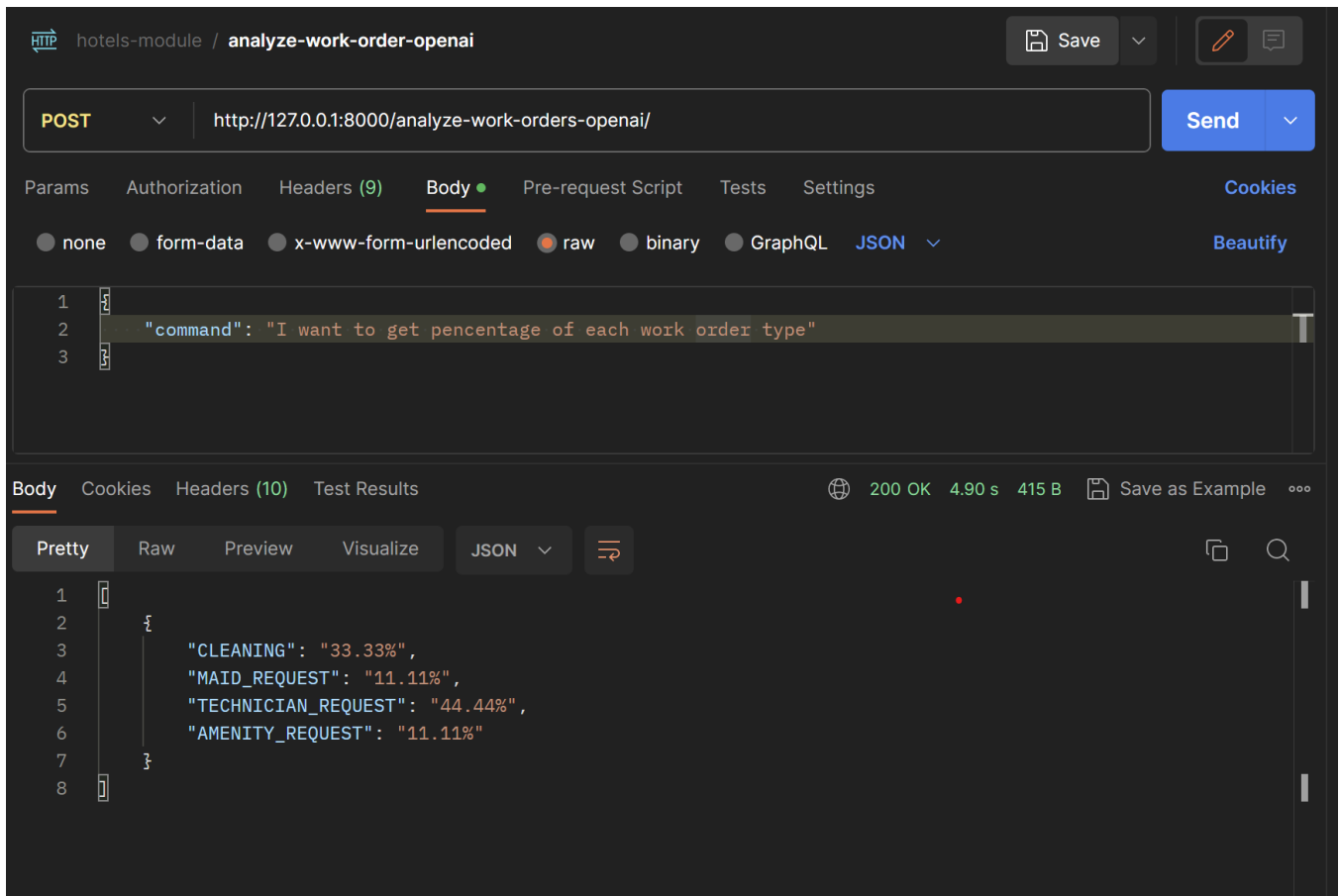
```
{  "command": "I want to get sum data of work order that room number = 999"}
```
- Response (JSON):**

```
[  {    "total": 5  }]
```
- Status:** 200 OK, 3.50 s, 323 B

ASSIGNMENT

AltoTech

3. Analyze ข้อมูล ทดสอบโดยการส่ง post request ไปที่ url: <http://127.0.0.1:8000/analyze-work-orders-openai/> โดยส่ง parameter เป็น command เช่น



ASSIGNMENT 3

จากโจทย์ที่แบ่งจาก service เดียวตอนนี้เป็น microservice ณ ตอนนี้ก็จะประกอบไปด้วย 4 service

- Gateway ทำหน้าที่ generate token, ตรวจสอบ permission, circuit breaker โดย service นี้จะ
ไม่มีการ access database ทำหน้าที่รับ request และหน้าที่ด้าน security อย่างเดียว
- User ทำหน้าที่ manage user สามารถ access database users ได้เพียง service เดียว
- WorkOrder ทำหน้าที่ manage work order สามารถ access database workOrder ได้เพียง
service เดียว
- OpenAi ทำหน้าที่ service ที่เกี่ยวข้องกับ ai

ASSIGNMENT

AltoTech

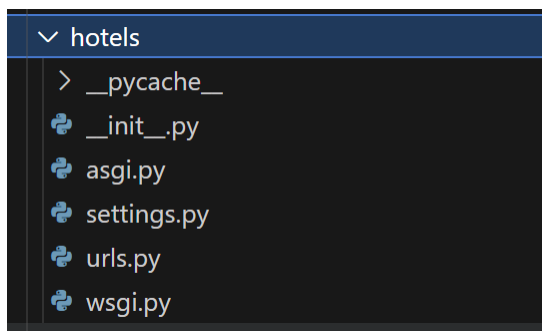
*** จริงๆตั้งใจจะทำเป็นแบบนี้ตอนแรกด้วยครับ แต่ด้วยเงื่อนไขของเวลาที่ต้องใส่ security ระหว่างแต่ละ service คงไม่ทัน

โดย set up ระบบบน cloud amazon แล้วใช้ kubernetes ในการ scaling pod ของแต่ละ services ตาม load ที่ได้รับ

CODE EXPLAIN

ส่วนนี้ขออธิบาย file แต่ละ file นะครับ จะได้สามารถเข้าไปอ่านได้ถูก

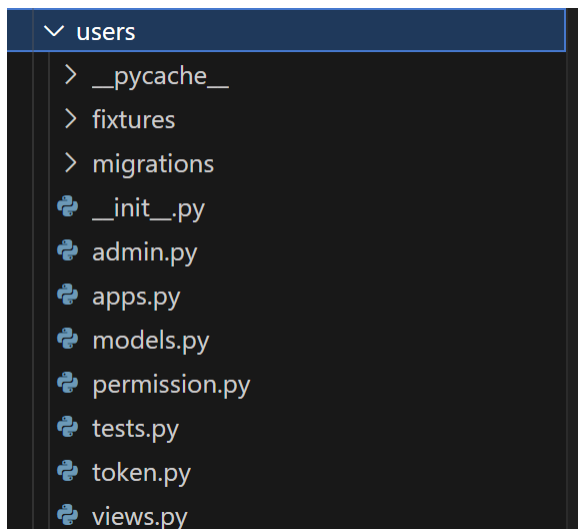
- hotels



ที่สำคัญตรงนี้มี 2 file นะครับ

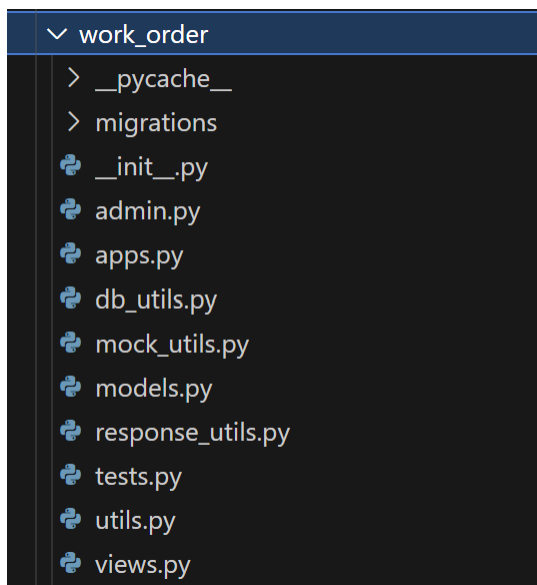
- settings.py ส่วนนี้จะเป็น config ค่า settings ของทั้ง project
- urls.py ส่วนนี้เป็นตัวกำหนดว่า path url ไหนเรียกไปที่ views ไหน

- users



- fixtures เก็บ file ข้อมูล initial data base ใน format json ที่นี้คือเก็บว่า user type ไหน มี permission ไหนบ้าง
- migrations เก็บ file migration ของ database ใน project นี้ใช้เก็บ table CustomUser และ UserPermission ในส่วนนี้ใช้ sqlite3 ซึ่งเป็น build in ของ Django ครับ
- admin.py ไว้ register หน้า backend เพื่อให้ admin สามารถ edit ข้อมูลใน Database ได้โดยตรงใน project นี้คือ table CustomUser และ UserPermission
- models.py ไว้สร้างโครงสร้างของ database แต่ละ table
- permission.py เป็น utility สำหรับตรวจสอบ permission
- token.py ไว้ตรวจสอบ authenticate จาก request และ ดึง permission จาก payload
- views.py logic ของ request API app นี้คือ create-user และ auth

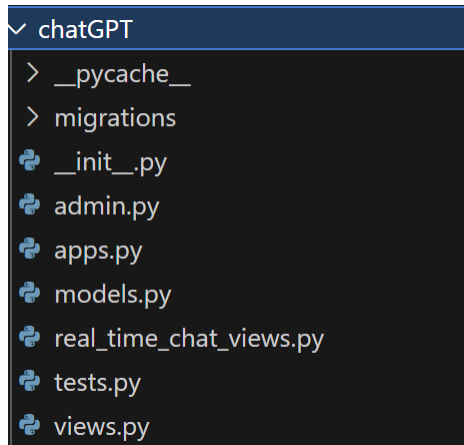
- work_order



- db_utils.py file utility สำหรับ เชื่อมต่อกับ mongo db , สำหรับ create, update, find ข้อมูล work_order
- mock_utils.py เก็บ function สำหรับ mock function ที่มีการติดต่อกับ service ภายนอก ให้ return ค่าที่เหมาะสมสำหรับ unit test
- reponse_utils.py utility สำหรับ return response code ต่างๆ ตามความเหมาะสม

- tests.py เขียน test case สำหรับ run unit test
- utils.py จัดการ enum สำหรับ work order เช่น WORK_ORDER_STATUS และ WORK_ORDER_TYPE
- views.py logic ของ request API app นี้คือ create-work-order และ update-work-order

- ChatGPT



- real_time_chat_views.py logic ของหน้า realtime chat
- views.py logic ของ request API app นี้คือ query-work-order-openai และ analyze-work-order-openai