

Problem#3 Design API endpoint

1) Create task

- Description: สร้าง Task โดยมีการรับค่า Input ทั้งหมด 6 ค่า คือ id (int), password (str), firstName (str), lastName (str), salary (int), role (str) นำไปเก็บไว้ใน task เพื่อเป็นจุดเริ่มต้นในการเข้าสู่ระบบเพื่อใช้งานฟังก์ชันอื่นต่อไป
- Path: sgcu/task
- Method: POST
- Headers: ไม่มีเพิ่มเติม
- Request Body: JSON

```
{
  "id": xxxxx,
  "password": "84000cells",
  "firstName": "Prayut",
  "lastName": "Ungkan-o-cha",
  "role": "HR",
  "salary": 84000
}
```

```
{
  "id": xxxxx,
  "password": "84000cells",
  "firstName": "Pareena",
  "lastName": "Narkupt",
  "role": "employee",
  "salary": 15000
}
```
- Request Params: ไม่มี
- Request Query: ไม่มี
- Responses

1. Successful task creation

- Response Body: JSON

```
{  
  "No": 1,  
  "id": xxxxx,  
  "password": "84000cells"  
  "firstName": "Prayut",  
  "lastName": "Ungkan-o-cha",  
  "role": "HR",  
  "salary": 84000  
}  
  
{  
  "No": 2,  
  "id": xxxxx,  
  "password": "84000cells",  
  "firstName": "Pareena",  
  "lastName": "Nearkupt",  
  "role": "employee",  
  "salary": 15000  
}
```

- Status Code: 200 OK

2. Incorrect request body

- Response Body: JSON

```
{  
  "message": "some data do not exist."  
}
```

- Status Code: 400 Bad Request

3. Body exists

- Response Body: JSON

```
{
```

```
    "message": "id exist"
  }
}
```

- Status Code: 404 Not Found

2) Login

- Description: เข้าสู่ระบบโดยรับ input 2 ค่า คือ id และ password ซึ่ง id ถ้าของ HR จะเป็น string ส่วนของ employee จะเป็น integer

- Path: sgcu/login

- Method: POST

- Headers: Bearer Token Authorization if {"role" = "HR"}

```
{
  "HR_key": {{auth-key}}
}
```

#มันคือการที่ถ้าจับ role ได้ว่าคน ๆ นั้นเป็น HR แล้วเราให้ token กับ id ที่เป็น HR เพื่อที่จะยืนยันตัวตนและสามารถเข้าสู่ฟังก์ชันที่ทำได้เฉพาะ HR

- Request Body: JSON

```
{
  "id": xxxxx
  "password": "xxxxxxxxx"
}
```

- Request Params: int id

- Request Query: น่าจะต้องมี

- Responses

1. Login Success

- Response Body: JSON

```
{
  "message": "Login Success"
}
```

- Status Code: 200 OK

2. Incorrect request body (ID or Password Incorrect)

- Response Body: JSON

```
{
  "message": "Your ID or Password Incorrect."
}
```
- Status Code: 400 Bad Request

3) Logout

- Description: ออกจากระบบ
- Path: sgcU/logout
- Method: Post
- Headers: ไม่มี
- Request Body: ไม่มี
- Request Params: "id"
- Request Query: ไม่มี
- Responses
 - Response Body: JSON

```
{
  "message": "Logout Complete!"
}
```
 - Status Code: 200 OK

4) Change password

- Description: เปลี่ยน password รับค่า input ทั้งหมด 3 ค่า
- Path: sgcU/user/{firstName+lastName}/change_password or
sgcU/admin/{firstName+lastName}/change_password
- Method: PUT
- Headers: ไม่มี

- Request Body: JSON

```
{  
  "old_password": "xxxxxxx"  
  "new_password": "xxxxxxxxx"  
  "password_agin": "xxxxxxxxx"  
}
```

- Request Params: int id

- Request Query: น่าจะต้องมี

- Responses

1. Change password success

- Response Body: JSON

```
{  
  "error": false,  
  "message": "success"  
}
```

- Status Code: 200 OK

2. Old password incorrect or new passcode again incorrect

- Response Body: JSON

1. Old password incorrect

```
{  
  "error": true,  
  "message": "Old password incorrect"  
}
```

2. New passcode again incorrect

```
{  
  "error": true,  
  "message": "Old password incorrect"  
}
```

- Status Code: 400 Bad Request

5) Check self-information

- Description: เช็คข้อมูลปัจจุบันของตัวเอง
- Path: sgcuser/user/user_info or sgcuser/admin/admin_info
- Method: GET
- Headers: ไม่มี
- Request Body: ไม่มี
- Request Params: ไม่มี
- Request Query: ไม่มี
- Responses:
 - Responses Body

```
{
    "firstName": "Prayut",
    "lastName": "Ungkan-o-cha",
    "role": "HR",
    "salary": 84000
}
```
 - Status Code: 200 OK

6) Add people

- Description: รับค่า Input ทั้งหมด 6 ค่า คือ id (int), password (str), firstName (str), lastName (str), salary (int), role (str) เพื่อนำไปเก็บไว้ใน task ที่จัดเตรียมไว้
- Path: sgcuser/admin/{firstName+lastName}/add_people
- Method: POST
- Headers: Bearer Token Authorization
- Request Body:

```
{
    "id": xxxxx,
    "password": "No112",
```

```
    "firstName": "Nakarin",  
    "lastName": "Rujieiei",  
    "role": "HR",  
    "salary": 120000  
  }
```

- Request Params: int id
- Request Query: น่าจะต้องมี
- Responses

1. Add success

- Response Body: JSON

```
{  
  "message": "Add people Complete!"  
  "Confirm_info": {  
    "No": 3,  
    "id": xxxxx,  
    "password": "No112",  
    "firstName": "Nakarin",  
    "lastName": "Rujieiei",  
    "role": "HR",  
    "salary": 120000  }  
}
```

- Status Code: 200 OK

2. Incorrect request body

- Response Body: JSON

```
{  
  "message": "some input data do not exist."  
}
```
- Status Code: 400 Bad Request

3. Body exists

- Response Body: JSON

```
{  
  "message": "id exist"  
}
```

- Status Code: 404 Not Found

7) Check employee's information

- Description: รับค่า Input 1-3 ค่า คือ firstName (str), lastName (str), role (str) เพื่อนำไปค้นหาว่าบุคคลนี้อยู่ในบริษัทหรือไม่ และมีข้อมูลส่วนตัวว่าอะไร
- Path: sgcu/admin/{firstName+lastName}/check_user/user/{firstName+lastName}/
- Method: POST
- Headers: Bearer Token Authorization
- Request Body:

```
{  
  "firstName": "Nakarin",  
  "lastName": "Rujieiei",  
  "role": "HR"  
}
```

#Optional สามารถเขียนได้อย่างใดอย่างหนึ่ง 2 อย่าง หรือ 3 อย่างได้

- Request Params: int id
- Request Query: น่าจะต้องมี
- Responses

1. Search success

- Response Body: JSON

```
{  
  "No": 3,  
  "id": xxxxx,  
  "password": "No112",  
  "firstName": "Nakarin",  
  "lastName": "Rujieiei",
```



```
    "role": "HR",  
    "salary": 120000  
  }
```

- Status Code: 200 OK

2. Incorrect request body

- Response Body: JSON

```
{  
  "message": "some input data do not exist."  
}
```

- Status Code: 400 Bad Request

8) Fix employee's information

- Description: รับค่า Input 3ค่า คือ firstName (str) และ lastName (str), New data ใหม่ที่ต้องการแก้ไข โดยใช้รูปแบบที่กำหนดไว้ ดังRequest Body ด้านล่าง เพื่อนำไปค้นหาว่าบุคคลนี้อยู่ในบริษัทหรือไม่ ถ้าอยู่ก็นำข้อมูลใหม่ไปแก้ไข
- Path: sgcu/admin/check_user/fix_user
- Method: PATCH
- Headers: Bearer Token Authorization
- Request Body:

```
{  
  "firstName": "Prayut",  
  "lastName": "Ungkan-o-cha",  
  "new_data": {"lastName": "Put-o-cha",  
               "salary": 84001  
            }  
}
```

#Optional สามารถเขียนได้อย่างใดอย่างหนึ่ง 2อย่าง หรือ 3 อย่างได้

- Request Params: int id
- Request Query: น่าจะต้องมี
- Responses

1. Fixable Success

- Response Body: JSON

```
{  
  "Message": "Fixable success"  
  "New_info": { "firstName": "Prayut",  
                "lastName": "Put-o-cha",  
                "role": "HR",  
                "salary": 84001}  
}
```

- Status Code: 200 OK

2. Incorrect request body

- Response Body: JSON

```
{  
  "message": "some input data do not exist."  
}
```

- Status Code: 400 Bad Request

9) Delete employees

- Description: รับค่า Input 2ค่า คือ firstName (str) และ lastName (str) ว่าบุคคลนี้อยู่ในบริษัทหรือไม่ ถ้าอยู่ก็ลบออก
- Path: sgcu/admin/check_user/detele
- Method: DELETE
- Headers: Bearer Token Authorization
- Request Body:

```
{  
  "firstName": "Prayut",  
  "lastName": "Ungkan-o-cha",  
}
```
- Request Params: int id

- Request Query: น่าจะต้องมี
- Responses
 1. Delete Success
 - Response Body: JSON


```
{
    "Message": "Delete success"
}
```
 - Status Code: 200 OK
 2. Incorrect request body
 - Response Body: JSON


```
{
    "message": "some input data do not exist."
}
```
 - Status Code: 400 Bad Request

10) Read employee's information (All people)

- Description: สามารถอ่านข้อมูลของพนักงานทุกคนในบริษัทได้ เรียงตามเลขที่ตามที่อยู่ใน task
- Path: sgcu/admin/{firstName+lastName}/employee_info
- Method: READ
- Headers: Bearer Token Authorization
- Request Body: ไม่มี
- Request Params: ไม่มี
- Request Query: ไม่มี
- Responses
 - Response Body: JSON


```
{
    "employee_information":
      [{"No": 1,
        "firstName": "Prayut",
```

```
        "lastName": "Ungkan-o-cha",
        "role": "HR",
        "salary": 84000},
    {"No": 2
      "firstName": "Pareena",
      "lastName": "Nearkupt",
      "role": "employee",
      "salary": 15000},
    {"No": 3,
      "firstName": "Nakarin",
      "lastName": "Rujieiei",
      "role": "HR",
      "salary": 120000}
  }
```

- Status Code: 200 OK

**ทางผู้สมัครได้ศึกษาในส่วนของ Request Parameters และ Request Query อย่างละเอียดและถี่ถ้วนแล้ว แต่ไม่ค่อยเข้าใจมากนัก จึงต้องขอภัยใน ณ ที่นี้ด้วย ซึ่งที่ผู้สมัครเข้าใจมีดังนี้

1. Request Parameters จะเป็นมันคือเราใช้เพื่อที่จะแยกออกจากค่าของURI ซึ่งจะมีลักษณะ เช่น
sgcu/admin/check_user/fix_user?id=xxxxx ซึ่งส่วนที่ไฮไลต์จะเป็นRequest params
2. Request Query จะเป็นส่วนหนึ่งใน URL ซึ่งมีลักษณะเหมือน Request params แต่จะอยู่ใน URL