



ฝ่ายพัฒนาระบบสารสนเทศ
องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย (อบจ.)

แบบทดสอบทักษะพื้นฐาน สำหรับผู้สมัครตำแหน่ง

Backend Developer

ฝ่ายพัฒนาระบบสารสนเทศ

องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2564

คำชี้แจง

1. แบบทดสอบนี้ถูกออกแบบมาให้ทำเสร็จภายใน **72 ชั่วโมง** อย่างไรก็ตามคณะกรรมการฯ จะให้เวลาในการทำทั้งหมด **120 ชั่วโมง** หลังจากที่ผู้สมัครได้รับบททดสอบนี้แล้ว เนื่องจากผู้สมัครบางท่านอาจจะมีช่วงเวลาที่สะดวกแตกต่างกันไป
2. ให้ผู้สมัครทำการ fork repository [isd-sgcu/sgcu64-backend-recruitment](https://github.com/isd-sgcu/sgcu64-backend-recruitment) โดยใช้ account ของผู้สมัครเอง
3. กรุณาอ่านรายละเอียดโจทย์และทำความเข้าใจให้เรียบร้อยก่อนทำโจทย์ หากมีข้อสงสัยประการใดผู้สมัครสามารถส่งอีเมลไปที่ isd.sgcu64@gmail.com เพื่อสอบถามได้ตลอดเวลา
4. พึงระลึกไว้เสมอว่าสิ่งเหล่านี้มีความสำคัญในการพัฒนาซอฟต์แวร์จริง ๆ ซึ่งมีผลต่อการพิจารณาผู้สมัคร ดังนี้
 - Code Readability – โค้ดอ่านรู้เรื่อง กระชับ เข้าใจง่าย
 - Commit Message Readability – ข้อความใน commit ต้องมีความหมาย
5. หากมีไอเดียเพิ่มเติมจากที่โจทย์ระบุไว้ที่ดีกว่าจากที่โจทย์กำหนด ผู้สมัครสามารถระบุไอดีเดีย้นั้นลงไปในส่วนของคำตอบ โดยเขียนด้วยรูปแบบดังนี้
Additional Idea: <ไอดีเพิ่มเติมของผู้สมัคร>
6. หากมีสมมติฐานเพิ่มเติมที่โจทย์ไม่ได้ให้ข้อมูลไว้ ผู้สมัครสามารถระบุสมมติฐานนั้นลงไปในส่วนของคำตอบ โดยเขียนด้วยรูปแบบดังนี้
Assumption: <สมมติฐานของผู้สมัคร>
7. การส่ง assignment ขอให้ผู้สมัครอัปโหลด source code ทั้งหมดลง repository ดังกล่าว โดยแยกออกเป็น folder ละ 1 ข้อ เช่น “Problem #1” โดยสามารถเพิ่มไฟล์ README.md ไว้ในแต่ละข้อได้ หากต้องการอธิบายเกี่ยวกับโค้ดที่ส่งมาเพิ่มเติม
8. การตรวจ assignment จะยึดจาก **commit สุดท้ายใน branch master** ก่อนเวลาสิ้นสุดแบบทดสอบที่ระบุไว้ในอีเมลที่ผู้สมัครได้รับเท่านั้น โปรดพึงระลึกไว้เสมอว่า การตรงต่อเวลา เป็นคุณสมบัติที่สำคัญอย่างหนึ่งต่อการเป็นวิศวกรซอฟต์แวร์ที่ดี
9. ขอให้ทยอยส่ง source code เมื่อทำเสร็จแต่ละข้อ เพื่อที่จะได้ไม่มีปัญหาการส่งเมื่อเวลาใกล้หมด **ไม่จำเป็นต้องเสร็จทุกข้อแล้วค่อยส่ง**
10. ผู้สมัครสามารถค้นหาข้อมูลเพิ่มเติมจากช่องทางใดก็ได้ ในระหว่างการทำแบบทดสอบ
11. การตัดสินใจของทีมคัดเลือกถือเป็นที่สุด

โจทย์ข้อที่ 1 Seven Segments

จุดประสงค์

- เพื่อทดสอบความรู้ความเข้าใจในการเขียนโปรแกรมเบื้องต้น
- เพื่อประเมินความสามารถในการออกแบบวิธีการแก้ปัญหาอย่างง่าย

รายละเอียดโจทย์

เมื่อโลกถูกปกคลุมไปด้วยความมืดมิด ทวีตทุกสารทิศมีเพียงความหวาดผวและความกลัว โลกจึงต้องการผู้กอบกู้เพื่อเป็นแสงสว่างแห่งอนาคต จึงได้มีการจัดกิจกรรมโอลิมปิกผู้กอบกู้ เพื่อเฟ้นหายอดมนุษย์ไปจัดการกับกองทัพจอมมารผู้ชั่วร้าย ซึ่งสุดท้ายแล้วชาวโลกก็สามารถหาผู้กอบกู้ได้ทั้งหมด 7 คนด้วยกัน ซึ่งผู้กอบกู้ทั้ง 7 ก็ได้ช่วยกันฝ่าฝืนอุปสรรคจนมาถึงหน้าปราสาทของจอมมารได้แล้ว

เนื่องจากจอมมารเป็นคนที่มีความละเอียดรอบคอบจึงได้วางแผนสำรองหากตัวเองไม่สามารถปราบผู้กอบกู้ได้ นั่นก็คือจอมมารจะติดตั้งเครื่องวัดชีพจรไว้ที่ตัวเอง และหากชีพจรของจอมมารหยุดนิ่งเมื่อไร "ระเบิดเวลา" จะเริ่มนับถอยหลังแล้วแยกผืนแผ่นดินออกเป็น 7 ส่วนออกจากกันเพื่อไม่ให้ผู้กอบกู้กลับบ้านเกิดของตนเองได้ แต่ทั้งนี้ทั้งนั้น "จอนับเวลาลอยหลัง" ของจอมมารที่ชื่อมาจาก Shopee มีระบบแสดงผลเป็นแบบ "7 ส่วน" [จอมมารชื่อมาจากลิงก์นี้](#)

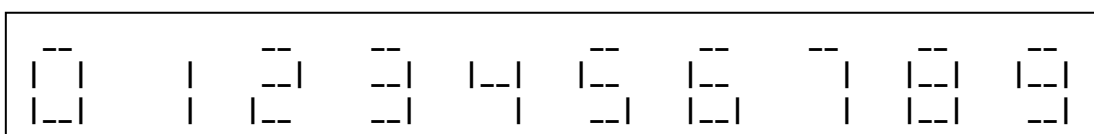
โดยในปราสาทจอมมารมีเพียงแค่คุณคนเดียวที่สามารถเขียนระบบแสดงผล 7 ส่วน ได้ ดังนั้นจอมมารจึงมอบหมายหน้าที่อันยิ่งใหญ่ให้กับคุณ ในการเขียนระบบแสดงผล 7 ส่วนให้เสร็จสมบูรณ์ให้ได้

Features ที่ต้องมี

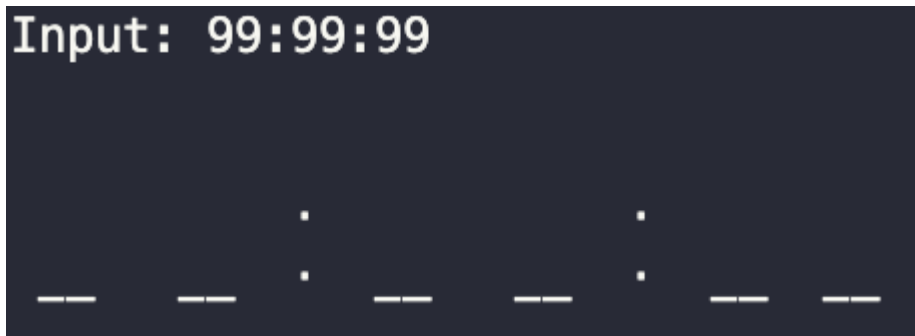
1. สามารถรับ input ใน format xx:xx:xx ได้ เมื่อ x คือเลขจำนวนโดดฐานสิบ โดยแสดงผล
2. สามารถแสดงผล output เป็นรูปแบบ seven segment โดย output จะประกอบไปด้วยตัวอักษร _ หรือ | หรือ · เท่านั้น เช่น



3. ตัวอย่างการแสดงผลแต่ละตัวเลข



4. สามารถบอกได้ว่า input ที่ส่งมาเป็นเวลาที่ถูกต้องหรือไม่ โดยเวลาในหลักนาที่และวินาทีอยู่ได้แค่ในช่วง 0 – 59 แต่หลักชั่วโมงจะเป็นเท่าไรก็ได้ หาก input ไม่ถูกต้องให้แสดงผลเป็น __:__:__ ดังนี้



5. **ไม่จำเป็น** ต้องให้เวลานับถอยหลังได้จริง ๆ แต่ถ้าอยากโชว์ของก็จัดมา
6. ผู้สมัครสามารถเขียนโปรแกรมโดยใช้ภาษาใดก็ได้ โดยขอให้ระบุวิธีการรันโปรแกรกดังกล่าวไว้ใน README.md ของ folder “Problem #1”
- หากเป็นภาษาที่สามารถ compile ออกมาเป็นไฟล์ executable ได้ (เช่น C/C++, etc.) ขอให้อัปไฟล์ executable นั้นขึ้นมาด้วย
 - หากเป็นภาษาที่ต้องใช้ interpreter ในการทำงาน (เช่น Python, Node.js, etc.) ขอให้ระบุ command ที่ใช้ในการรันโปรแกรมด้วย เช่น “node index.js”
7. ผู้สมัครจะเขียนอธิบายอะไรเพิ่มใน README.md หรือ comment ในไฟล์ก็ได้ตามอัธยาศัย

Tips

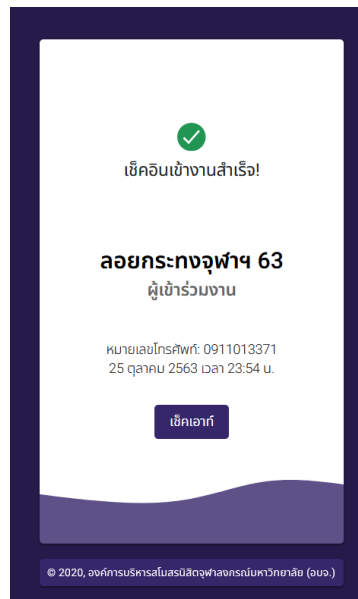
1. **สำหรับโจทย์ข้อนี้** ผู้สมัครไม่จำเป็นต้องกังวลเรื่อง Time and Memory Complexity (เวลาและ memory ที่ใช้) ขอเพียงให้โปรแกรมทำงานถูกต้องก็พอ

โจทย์ข้อที่ 2 Chula Chana

จุดประสงค์

- เพื่อทดสอบความรู้ความเข้าใจในการเขียนโปรแกรมเบื้องต้น
- เพื่อทดสอบความสามารถในการจัดการข้อมูลที่ได้รับมา ตั้งแต่การจัดเก็บไปจนถึงการค้นข้อมูล ได้อย่างถูกต้องและแม่นยำ

รายละเอียดโจทย์



(ภาพหน้าเว็บ CU Check In)

CU Check In เป็น Web Application ที่ทาง ISD SGCU63 ทำขึ้นเพื่อทำการ Track จำนวนคนในพื้นที่ ในช่วงอีเวนต์ลอยกระทงจุฬาฯ ในปีที่ผ่านมา ซึ่งใน app ผู้ใช้จะต้องสแกน QR Code ที่หน้างานลอยกระทง และกรอกเบอร์โทรศัพท์เพื่อ check in แล้วจึงจะสามารถเข้างานได้ และสามารถสแกน QR Code เพื่อ check out เมื่อจะออกจากงาน

ใน 1 ปีถัดมา ตัวคุณเองพบว่า Web Application ดังกล่าวของ ISD SGCU63 มี User Experience ที่ใช้งานยาก จึงได้ทำการเขียนโปรแกรม Chula Chana (จุฬาฯ ชนะ) เพื่อแข่งขันกับ CU Check In และ ไทยชนะ เพื่อที่จะก้าวขึ้นเป็นที่หนึ่งของ Tracking Application ของประเทศไทยให้ได้เลย

```
Welcome to Chula Chana!!!
Available commands:
  1. Check in user
  2. Check out user
  3. Print people count
Please input any number: 1
-----

Check In
Enter phone number: 0855555555
  1. Mahamakut Building
  2. Sara Phra Keaw
  3. CU Sport Complex
  4. Sanam Juub
  5. Samyan Mittr Town
Select the place: 3
Checking in 0855555555 into CU Sport Complex
-----

Welcome to Chula Chana!!!
Available commands:
  1. Check in user
  2. Check out user
  3. Print people count
Please input any number: 3
-----

Current Population
  1. Mahamakut Building: 20
  2. Sara Phra Keaw: 9
  3. CU Sport Complex: 35
  4. Sanam Juub: 2
  5. Samyan Mittr Town: 17
-----

Welcome to Chula Chana!!!
Available commands:
  1. Check in user
  2. Check out user
  3. Print people count
Please input any number:
```

(ตัวอย่าง UI อันสวยงามของโปรแกรมจุฬาฯ ชนะ)

โดยจุฬาฯ ชนะ ที่คุณจะสร้างขึ้นนั้น เป็นโปรแกรมง่าย ๆ บนคอมพิวเตอร์ ที่สามารถรับข้อมูลเป็นเบอร์โทรศัพท์ เพื่อ check in หรือ check out ในพื้นที่ไหนก็ได้ที่ตั้งค่าไว้ พร้อมทั้งสามารถดูจำนวนคนที่อยู่ในพื้นที่ต่างๆ ได้ และแสดงผลออกมาทาง standard output ดังเช่นตามในรูปข้างต้น

ทั้งนี้ รูปแบบการรับข้อมูลเข้าหรือการแสดงผลนั้น ผู้สมัครสามารถออกแบบเป็นอย่างไรก็ได้ตามความเหมาะสม ไม่มีความจำเป็นจะต้องเขียนเหมือนแบบในตัวอย่างข้างต้น เพียงแต่จะมี feature บางประการที่จำเป็นจะต้องมีอยู่ในโปรแกรมของคุณ ดังนี้

Features ที่ต้องมี

1. Check In

- input มีสองอย่าง คือ เบอร์โทรศัพท์ (สมมติว่าผู้ใช้กรอกถูก format ได้เลย) พื้นที่ที่ต้องการ Check in (สามารถ Hard Code ลงไปใน source code ได้)
- ไม่จำเป็นต้องมี output แต่อาจมีได้ เพื่อการใช้งานที่ง่ายขึ้น

- ในกรณีที่เบอร์โทรศัพท์นั้น เคย check in อยู่ในพื้นที่ที่ check in ไว้อยู่แล้ว แล้วจะต้องทราบได้ว่าเป็นคนเดียวกัน และไม่ส่งผลกระทบต่อจำนวนคนในพื้นที่นั้น
- ในกรณีที่เบอร์โทรศัพท์นั้น เคย check in อยู่ในพื้นที่อื่น ไม่ใช่พื้นที่ที่กำลังจะ check in ขอให้เข้าใจว่าผู้ใช้เดิม checkout พื้นที่เดิม จะต้องนำเบอร์โทรศัพท์นั้น ออกจากพื้นที่เก่า แล้วนำมาใส่พื้นที่ใหม่ (แสดงว่าจำนวนคนในพื้นที่เก่าจะต้องลดลง และจำนวนคนในพื้นที่ใหม่จะต้องมากขึ้น)
- ในสองกรณีข้างต้น สามารถแจ้งเตือนผู้ใช้ให้รู้ตัวหรือไม่ก็ได้ แล้วแต่ออกแบบ

2. Check out

- Input มีอย่างเดียว คือ เบอร์โทรศัพท์
- ไม่จำเป็นต้องมีข้อมูลส่งออก แต่อาจมีได้ เพื่อการใช้งานที่ง่ายขึ้น
- การ check out จะต้องตรวจสอบว่า เบอร์โทรศัพท์นั้นๆ อยู่ในพื้นที่ใด แล้วจึง check out ออกจากพื้นที่นั้นให้โดยอัตโนมัติ (แสดงว่าจำนวนคนในพื้นที่ที่ผู้ใช้เคยอยู่ต้องลดลง)
- ในกรณีที่เบอร์โทรศัพท์นั้นไม่ได้ถูก Check in อยู่พื้นที่ใด ไม่จำเป็นต้องทำอะไรใด ๆ สามารถแจ้งเตือนผู้ใช้ให้รู้ตัวหรือไม่ก็ได้ แล้วแต่ออกแบบ

3. แสดงผลจำนวนคนในแต่ละพื้นที่

- ไม่มี Input เพิ่มเติม
- Output จะต้องแสดงจำนวนคนในแต่ละพื้นที่ใน format ที่เข้าใจง่าย

สำหรับจำนวนสถานที่ที่ให้ผู้ใช้งานใช้นั้น สามารถ hard code ลงไปในโปรแกรมได้เลย เพียงแต่ต้อง ≥ 5 สถานที่ หรือสามารถทำเป็น dynamic (เพิ่มลดได้) ซึ่งก็มี feature สำหรับให้ผู้ใช้งานสามารถเพิ่มสถานที่เองได้

Feature อื่น ๆ นอกจากที่กล่าวมาข้างต้น เป็นสิ่งที่ไม่ได้บังคับ สามารถทำเพิ่มเติม หากอยากโชว์ของ

โจทย์ข้อที่ 3 API

จุดประสงค์

- วัดความรู้พื้นฐานในการออกแบบ API Endpoint
- วัดความรู้พื้นฐานในการเลือกใช้ HTTP Methods, Status Codes
- เพื่อประเมินความสามารถในการแก้ปัญหาของผู้สมัคร

รายละเอียดโจทย์

บริษัท SGCU ใช้วิธีการเขียนข้อมูลของพนักงานไว้ในที่สมุดบันทึกมาโดยตลอด คุณและทีม Developer ของบริษัท SGCU ได้มองเห็นถึงปัญหาของวิธีการดังกล่าว จึงต้องการที่จะเปลี่ยนระบบจัดการข้อมูลพนักงานบริษัท ไปเป็นแบบ online เพื่อที่จะเปลี่ยนระบบนี้ให้ทันสมัยยิ่งขึ้น สามารถเก็บและแก้ไขข้อมูลได้ง่ายขึ้น คุณจึงยื่นเรื่องให้แก่ท่านประธาน

จากการเสนอข้อมูลให้ประธานบริษัท เขาได้ถูกใจข้อเสนอดังกล่าวเป็นอย่างมาก จึงได้ร้องขอให้คุณและทีม Developer พัฒนาเว็บไซต์เว็บไซด์ดังกล่าวขึ้นเพื่อมาใช้งาน คุณซึ่งเป็นหนึ่งในทีมได้รับหน้าที่ในการ **วางแผนและออกแบบ API** เพื่อที่จะให้เว็บไซต์สามารถทำงานได้อย่างราบรื่น

API คืออะไร? <https://www.youtube.com/watch?v=OVvTv9Hy91Q>

API vs Endpoints <https://stevenpcurtis.medium.com/endpoint-vs-api-ee96a91e88ca>

งานของคุณ

ออกแบบ API Documentation ที่ใช้ในระบบนี้เพื่อให้สามารถทำตาม requirements ที่ระบุไว้ด้านล่างได้ โดยแต่ละ API Endpoint (URL) ขอให้อธิบายให้ละเอียดที่สุด เช่น

1. Endpoint Path (เช่น /user ฯลฯ)
2. HTTP Method (เช่น GET POST PUT DELETE ฯลฯ)
3. Headers (ถ้ามีนอกเหนือจาก header ทั่วไป เช่น Authorization)
4. Request Body (ถ้ามี) อาจเป็นในรูปแบบ plain text หรือ JSON
5. Request Params (ถ้ามี)
6. Request Query (ถ้ามี)
7. Response Body (ถ้ามี) อาจเป็นในรูปแบบ plain text หรือ JSON
8. Response Status Codes เป็นต้น เช่น (200, 400, 401, 403, 404, 500)

เป็นไปได้ที่ 1 endpoint จะมีได้หลาย response ขึ้นอยู่กับความเป็นไปได้ที่เกิดขึ้น ขอให้ระบุให้ครบถ้วนที่สุด ลองดูตัวอย่างด้านล่าง

ลักษณะการจัดเก็บข้อมูลพนักงานเป็นดังนี้

1. id: เลขประจำตัวพนักงาน (string)
2. password : รหัสผ่านซึ่งใช้ในการ login (string)
3. firstName: ชื่อจริงของพนักงาน (string)
4. lastName: นามสกุลของพนักงาน (string)
5. salary: เงินเดือนของพนักงาน (number)
6. role: ตำแหน่งของพนักงาน (string) **(optional)**

```
interface officer {  
  id: string  
  password : string  
  firstName: string  
  lastName: string  
  role: string  
  salary : number  
}
```

Feature ที่ต้องทำได้

Minimum Requirements

1. สามารถเพิ่มพนักงานใหม่เข้าไปในระบบได้ (Create)
2. สามารถดูข้อมูลของพนักงานทุกคนได้ (Read)
3. สามารถแก้ไขข้อมูลของพนักงานได้ เช่นชื่อ-สกุล ตำแหน่ง และเงินเดือนของพนักงานได้ (Update)
4. สามารถลบข้อมูลพนักงานในระบบได้ (Delete)
5. สามารถค้นหาพนักงานโดยใช้ ชื่อ นามสกุล หรือ ตำแหน่งได้

Optional

1. สามารถเข้าสู่ระบบได้ (โดยใช้ เลขพนักงาน, รหัสผ่าน)
2. สามารถแบ่งแยก user ออกเป็น 2 role คือ Employee กับ HR
 - Employee
 1. สามารถ login ด้วย username และ password ได้
 2. สามารถแก้ไข password ของตนเองได้
 3. สามารถดูข้อมูลของตนเองได้ (นั่นคือ API มีวิธีระบุตัวตนว่าใครเป็นคนยิง API)
 4. **ไม่สามารถ** ทำสิ่งที่ระบุไว้ 5 ข้อด้านบนได้ (ให้เฉพาะ HR ใช้งานเท่านั้น)
 - HR
 1. ทำสิ่งที่ employee ทำได้
 2. **สามารถ** สิ่งทีระบุไว้ 5 ข้อด้านบนได้ (ให้เฉพาะ HR ใช้งานเท่านั้น)

Tips

1. โจทย์ข้อนี้ไม่ต้องการให้ทำ API นั้นขึ้นมาจริงๆ เพียงแต่**ออกแบบ** API Endpoints ขึ้นมาเท่านั้น (แต่หากต้องการเขียน code backend ขึ้นมาจริง ๆ ก็ทำได้)
2. **การออกแบบ API Documentation ไม่มีถูกผิด** สามารถสร้าง API Documentation ขึ้นมาแบบไหนก็ได้ เช่น text file, PDF, [Swagger](#) และอื่นๆ ที่สามารถแสดงข้อมูลได้อย่างถูกต้องและครบถ้วน

3. การตรวจจะพิจารณาจากความเหมาะสมในการเลือกใช้ชื่อ Path, HTTP Method, Status Codes รวมถึงอื่น ๆ ตาม Convention ทั่วไปด้วย สามารถศึกษาเพิ่มเติมได้เองเลย
4. HTTP Methods สามารถศึกษาเพิ่มเติมได้ที่
(<https://www.restapitutorial.com/lessons/httpmethods.html>)
5. Status Codes สามารถศึกษาเพิ่มเติมได้ที่
(<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>)
ไม่จำเป็นต้องใช้หรือทราบทั้งหมด ที่ใช้ประจำมีไม่กี่อัน เช่น 200, 201, 301, 400, 401, 403, 404, 500, 503
6. การทำ Authorization (login, user identification) สามารถเลือกใช้วิธีใดก็ได้ เช่น JWT (JSON Web Token), session, หรือแม้แต่ HTTP Basic Auth
7. สำหรับ Authorization Header สามารถศึกษาเพิ่มเติมได้ที่
(<https://www.loginradius.com/blog/async/everything-you-want-to-know-about-authorization-headers/>)
8. หากสงสัยโจทย์ในส่วนใด สามารถติดต่อสอบถามได้ที่ isd.sgcu64@gmail.com เสมอ

ตัวอย่าง ของ Endpoint (ไม่จำเป็นต้องเป็นแบบในตัวอย่าง)

ทำด้วย Text ส่งเป็น PDF

1. Create task
 - Description: Creates a task with the given input. Returns the instance of the task stored in database. Task must have both name and deadline.
 - Path: /task
 - Method: POST
 - Headers: ไม่มีเพิ่มเติม
 - Request Body: JSON

```
{
  "name": "Buy chicken",
  "deadline": "21/07/2021"
}
```

- Request Params: ไม่มี
- Request Query: ไม่มี
- Responses
 1. Successful task creation

- Response Body: JSON

```
{
  "id": 1,
  "name": "Buy chicken",
  "deadline": "21/07/2021"
}
```

- Status Code: 200 OK

2. Incorrect request body

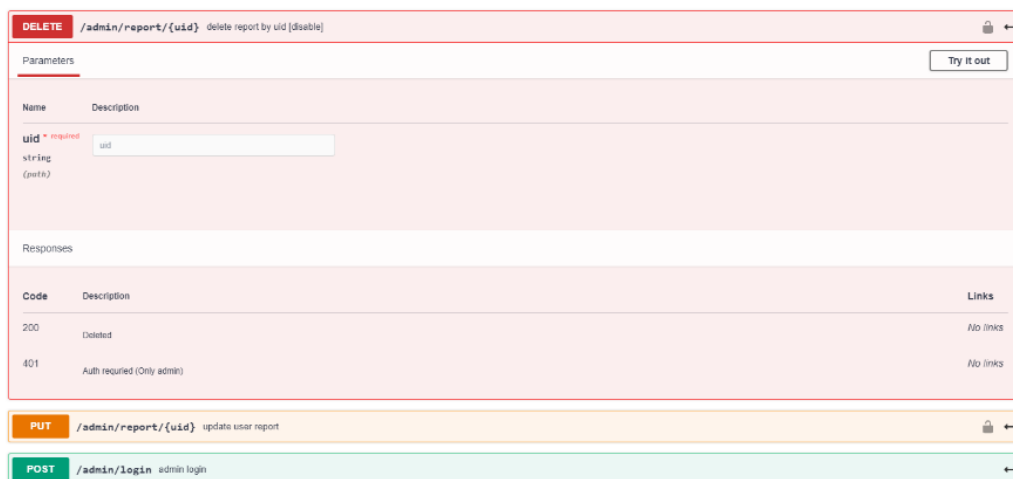
- Response Body: JSON

```
{
  "message": "name or deadline does not exist."
}
```

- Status Code: 400 Bad Request

... /* endpoint อื่น ๆ */

ทำด้วย Swagger



ตัวอย่าง End Point สำหรับ Swagger

การส่งงาน

ขอให้อัปโหลด documentation (PDF or Swagger) ขึ้นบน repository ใน folder ของข้อนี้