

แบบระบบย่อ URL

Database : MongoDB

- เป็น NoSQL สามารถปรับปรุงและแก้ไขได้ง่ายไม่ต้องกังวลถึงโครงสร้างเหมือน SQL ที่ต้องคำนึงถึง Table ทุกครั้งเวลาปรับแก้ Database
- NoSQL มีโมเดลข้อมูลที่หลากหลาย เช่น คำหลัก เอกสาร และกราฟ ซึ่งปรับให้เหมาะกับประสิทธิภาพและขนาด
- มีรูปแบบการใช้งานคล้ายกับ SQL ทำให้เวลาการเรียนรู้น้อยลง
- มี Free Tier Database Hosting ใน <https://www.mongodb.com/free-cloud-database>
- มี Mongoose เป็น Object Data Modeling(ODM) ทำหน้าที่คล้ายกับ Object Relational Mapping(ORM) ของ SQL Database ทำให้เรียนรู้ได้เร็ว

Collection:

Column	Type	Rule
original	string	Required, Unique
slug	string	Required

1. original คือ URL ต้นฉบับก่อนทำการย่อ โดยต้องระบุค่าและต้องเป็นรูปแบบที่ถูกต้องของ URL (regex: /^(ftp|http|https):\/\/[^\s"]+\$/)
 2. slug คือ ตัวแทนของ URL ในรูป 6 ตัวอักษรและต้องระบุ
-

Back-end: NestJS

- มีโครงสร้าง Framework คล้ายกับ Angular สำหรับผู้ใช้ Angular เป็น Front-end อยู่ จะมีความคุ้นเคยง่าย
- มี Command Line Interface(CLI) ในการ Generate Code ช่วยลดเวลาในเริ่มเขียน Module ใหม่
- เป็น Typescript ตั้งแต่เริ่มสร้าง Project ไม่ต้องตั้งค่าให้รองรับ Typescript เพิ่มเติมเหมือน Express JS
- มีรูปแบบเป็น MVC
- มี @nestjs/mongoose ซึ่งทำงานกับฐานข้อมูล MongoDB ได้งานขึ้น

Design:

Http Verb	Route	Params	Body	Example
GET	/	{slug}	-	/abcdef
POST	/	-	{url : string}	body: { "url" : "http://xxx.xxx" }

- GET คือ การค้นหา URL ดัชนี โดยที่ front-end ส่ง slug ผ่าน parameter โดยที่ front-end ต้องระบุ slug ทุกครั้งที่ส่ง request ถ้าพบข้อมูลให้ response เป็น json โดยมี slug และ original ถ้าค้นหาไม่เจอให้ response เป็น http status code 404

Ex.

```
success: { "original": "http://xxx.xxx/xxx" , "slug" : "xxxx" }  
failed: 404
```

- POST คือ การย่อ URL ดัชนี โดยที่ front-end ส่ง request ที่มี body ประกอบด้วย url

Ex.

```
request body: { "url" : "http://xxxx.xxx" }  
success: { "original": "http://xxx.xxx/xxx" , "slug" : "xxxx" }  
error: { "statusCode": 400, "message": [ xxx],  
"error": "Bad Request" }
```

โดยที่ url ต้องมีรูปแบบเป็น URL ที่ถูกต้อง(regex: /^(ftp|http|https):\/\/\[^\"]+\$/)

Front-end: Angular

- เป็น Framework ที่มีเครื่องมือครบเหมาะสำหรับทำงานใหญ่ๆ ไม่ต้องคำนึงถึง Plugin อื่นๆ มากเท่าไร เพื่อตัว Framework มีให้อยู่แล้ว เช่น Router, Guard, และอื่นๆ
- มี Dependency Injector(DI) สามารถใช้งานได้ทันทีไม่ต้องกังวลเรื่องออกแบบเอง เหมือนกับ Framework อื่น
- เป็น Typescript ตั้งแต่ Angular 2 ขึ้นมาไม่ได้ต้องกังวลมากเรื่องการเขียนตัวแปรรีด
- มี Command Line Interface(CLI) ช่วยเหลือในการเขียนโค้ด

Design:



Paste the URL to be shortened

Shorten

Shortened: <http://localhost:4200/5l9wsn>

Copy

- มี input สำหรับกรอก URL โดยมีเงื่อนไขต้องเป็นรูปแบบที่ถูกเท่านั้น ถ้าถูกต้องให้ปุ่ม Shorten สามารถกดได้ ถ้าไม่ถูกต้องให้แสดง error
 - เมื่อกดที่ปุ่ม Shorten จะทำการส่ง request ไปยัง back-end เพื่อสร้าง slug หลังจากนั้นจะสร้าง link ขึ้นมาด้านล่างพร้อมกับปุ่ม Copy
 - เมื่อคลิกที่ link ที่ถูกสร้างขึ้นจะทำการ redirect ไปที่ URL ต้นฉบับ
-

Deployment: Vercel และ Atlas

- Atlas เลือกใช้เพราะเป็น Database Hosting ที่เป็น Official ของ MongoDB ใช้งานง่าย และค่อนข้างเร็วแม้ใช้ในรูปแบบ Free Tier
- Vercel ใช้เพื่อ Hosting สำหรับ Back-end และ Front-end เพราะใช้งานง่าย สามารถเชื่อมต่อกับ Github ได้ง่ายและมี Automatic Deployment กันที่เมื่อมี Code ถูก Push ขึ้นไปที่ Github

Free Question

คำถาม: Authentication และ Authorization แตกต่างกันอย่างไร

คำตอบ:

1. Authentication ใช้ในการยืนยันตัวตนของผู้ที่จะเข้าใช้งานระบบ แต่ Authorization เป็นการบริหารผู้ใช้มีสิทธิ์ของผู้ใช้งาน เช่น การเข้าถึงส่วนใดในระบบบ้าง
2. Authentication มีรูปแบบในการตรวจสอบเช่น Username และ Password, คำถาม, Fingerprint และ Facial Recognition เป็นต้น ส่วน Authorization จะถูกกำหนดตาม Policies ของแต่ละระบบ
3. เป็นกระบวนการที่ต้องเริ่มทำการก่อนการ Authorization เสมอ การจะทำ Authorization ต้องมีการระบุตัวตนก่อนถึงจะเริ่มทำงานได้