



รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ

Web Application For API Testing

นายอิสรา นรานิรััติย์

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ

Web Application For API Testing

นายอิสรา นรานิรัตติชัย

สาขาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา	เว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ
ชื่อ-สกุล นักศึกษา	นายอิสรา นรานิรัติกัย
คณะ วิศวกรรมศาสตร์	ภาควิชา วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ	นายจิระศักดิ์ สิทธิกร และ นายบัณฑิต พัสยา
ชื่อ-สกุล ผู้นิเทศงาน	นาย Sparsh Choudhary
สถานประกอบการ	บริษัท อโกดา เซอร์วิสเชส จำกัด

บทคัดย่อ

ในการทดสอบเอพีไอ(Application programming interface) นั้นทางทีมพัฒนาได้ใช้ RESTFUL API Client อย่างเช่น Insomnia, Postman ฯลฯ ซึ่งสามารถดูผลลัพธ์ในรูปแบบต่างๆ เช่น JSON หรือ RAW Data และการที่จะสร้าง JSON Request ในการร้องขอข้อมูลจาก API ของทางทีมที่ข้าพเจ้าได้เข้าทำการสหกิจศึกษานั้น ต้องใช้รูปแบบของ JSON

เพื่อให้ง่ายต่อการทดสอบเอพีไอ ข้าพเจ้าจึงได้จัดทำ API Test Tool ขึ้น เพื่อช่วยให้การใส่ข้อมูลจากเดิมที่ต้องเป็น JSON นั้นเป็นรูปแบบฟอร์ม และในหน้าแสดงผลได้นำข้อมูลจากเดิมเป็น JSON Response นั้นมาแสดงผลในรูปแบบที่ทำมาเฉพาะเพื่อทีมพัฒนาในรูปแบบต่างๆเช่นตาราง ลิสต์ ฯลฯ รวมทั้งสามารถเปรียบเทียบความแตกต่างของ JSON Response ระหว่าง 2 API Server ได้ด้วย

คำสำคัญ : เอพีไอ, เว็บแอปพลิเคชัน, เจชัน, เซิร์ฟเวอร์

Co-operative Title: Web Application For API Testing
Student Intern Name: Mr. Isara Naranirattisai
Faculty: Engineering **Department:** Computer Engineering
Advisor Name: Mr. Jirasak Sittigorn and Mr. Bundit Pasaya
Mentor Name: Mr.Sparsh Choudhary
Company: Agoda Services Co., Ltd.

ABSTRACT

In API Testing, my team are using the tool “Restful API Client” i.e. Insomnia, Postman, ETC. to see the result in JSON format or Raw data format. For testing API we have to create some JSON request data to request some data from API.

To make it more readable I created the web application’s form instead of writing JSON request data. And Create a better display the JSON response data in reasonable format such as table, list specially for the team. And for checking the differents between 2 API Server I also create a JSON Comparator features inside the API Test Tool.

Keywords: API, Web Application, JSON, Server

กิตติกรรมประกาศ

เว็บแอปพลิเคชันสำหรับทดสอบเอพีไอและรายงานสหกิจศึกษานี้สำเร็จลุล่วงไปได้ด้วยความ
กรุณาจาก Mr. Sparsh Choudhary, พี่ๆในทีมขณะเข้าทำการสหกิจศึกษา นายภูริพันธ์ จุลศิริกุล (เพื่อนร่วมทีม
ในช่วงฝึกงานฤดูร้อน) ตลอดจนอาจารย์ที่ปรึกษาทุกๆท่านที่ให้คำแนะนำเสมอมา ตลอดจนแก้ไขข้อบกพร่อง
จนชิ้นงานและรายงานฉบับนี้เสร็จสมบูรณ์

ขอขอบพระคุณคุณพ่อ คุณแม่ และครอบครัว สำหรับคำปรึกษาในเรื่องต่างๆ และ ให้กำลังใจเสมอมา

อิสรา นรานิรัตติชัย

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมา และความสำคัญ.....	1
1.2 วัตถุประสงค์ของงาน	1
1.3 ขอบเขตของการทำงาน.....	1
1.4 วิธีดำเนินการทำงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	3
2.1 Application programming interface (API)	3
2.2 Cross-Origin Resource Sharing (CORS).....	3
2.3 HTTP Request Methods	3
2.4 JavaScript Object Notation (JSON).....	3
บทที่ 3 วิธีดำเนินงาน	5
3.1 วิธีดำเนินการทำงาน.....	5
3.2 แผนการดำเนินงาน	9
บทที่ 4 ผลการทำงาน	10

สารบัญ (ต่อ)

	หน้า
4.1 ผลการทำงานในฝั่ง API Test Tool	10
4.2 ผลการทำงานในฝั่ง URL Shortener.....	10
4.3 ผลการทำงานในฝั่ง API (เรื่อง Cross-Origin).....	10
บทที่ 5 สรุปผลการทำงาน และข้อเสนอแนะ	11
5.1 บทสรุป	11
5.2 ปัญหาอุปสรรค และแนวทางแก้ไข.....	11
5.3 แนวทางการพัฒนาต่อ	11
บรรณานุกรม.....	12

สารบัญตาราง

ตารางที่	หน้า
3.1 แผนการดำเนินงาน	9

สารบัญภาพ

ภาพที่	หน้า
3.1 ตัวอย่างการ Request ข้อมูลโดยใช้ Insomnia.....	5
3.2 โครงสร้างการทำงานของระบบ.....	6

บทที่ 1

บทนำ

1.1 ความเป็นมา และความสำคัญ

ทีมที่ข้าพเจ้าได้เข้าทำการเข้าร่วมโครงการสหกิจนั้นเป็นทีมที่พัฒนาระบบ เอพีไอ (API) สำหรับให้บริการข้อมูลของโรงแรม ซึ่งในอดีตนั้นการทดสอบข้อมูลใช้วิธีพิมพ์ JSON Request ลงในโปรแกรม REST API Client อย่างเช่น Postman หรือ Insomnia เพื่อให้ได้มาซึ่ง JSON Result ซึ่งเป็นข้อมูลที่อ่านและวิเคราะห์ยากมากเมื่อเป็นข้อมูลที่มีขนาดใหญ่ และต้องเปรียบเทียบกันโดย Manual ทาง Manager ของข้าพเจ้าจึงมอบหมายให้ทำ API Test Tool ซึ่งเป็นเว็บแอปพลิเคชันสำหรับทดสอบ API สำหรับทีมที่ข้าพเจ้าได้เข้าร่วมนั้น โดยเฉพาะ เพื่อให้ง่ายต่อการทดสอบ เปรียบเทียบข้อมูลระหว่าง 2 เซิร์ฟเวอร์ แสดงผลให้อ่านง่ายขึ้น รวมทั้งรองรับการแชร์ผลอย่างง่ายเพื่อให้สามารถแชร์ผลลัพธ์การทำงานให้ทีมต่างๆทราบในกรณีที่เกิดปัญหาได้เรียงง่ายและกระทัดรัดมากขึ้น

1.2 วัตถุประสงค์ของงาน

- 1) เพื่อใช้เป็นเว็บแอปพลิเคชันที่ใช้ในการทดสอบเอพีไอให้ง่ายขึ้น
- 2) เพื่อใช้เปรียบเทียบ(Server) ข้อมูลระหว่างเอพีไอสองเอพีไอ

1.3 ขอบเขตของการทำงาน

สำหรับระบบเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอนั้น มีส่วนที่ต้องพัฒนา 3 ส่วนหลักหลักดังนี้

- 1) เว็บแอปพลิเคชันสำหรับเอพีไอ (API Test Tool)
- 2) การยินยอมให้ API Test Tool ใช้ Cross-Origin Resource Sharing ในฝั่ง API
- 3) API สำหรับย่อ URL (URL Shortener)

1.4 วิธีดำเนินการทำงาน

- 1) ศึกษาการทำงานและโครงสร้างของเอพีไอ
- 2) รับผิดชอบต่อความต้องการของระบบเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ
- 3) ศึกษาเครื่องมือที่ใช้ในการพัฒนา
- 4) ออกแบบระบบ
- 5) เขียนโปรแกรม
- 6) ทดสอบระบบ
- 7) สาธิตระบบแก่ทีม
- 8) จัดทำรายงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1) สามารถเปรียบเทียบข้อมูลที่ได้รับจากเอพีไอได้ง่ายขึ้น
- 2) แสดงผลข้อมูล JSON Response ในรูปแบบที่ง่ายขึ้น
- 3) สามารถแชร์ผลลัพธ์ให้ผู้อื่นในทีมพัฒนาได้ตรวจสอบได้ง่ายขึ้น

บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 Application programming interface (API)

เอพีไอ หรือ ส่วนต่อประสานโปรแกรมประยุกต์ เป็นระบบ หรือบริการที่ให้โปรแกรมคอมพิวเตอร์ หรือแอปพลิเคชันต่างๆ เรียกใช้บริการได้ เช่นการ Request และ Response ข้อมูลต่างๆ

2.2 Cross-Origin Resource Sharing (CORS)

CORS คือกลไกที่ทำให้เว็บเบราว์เซอร์ (ในที่นี้คือ API) อนุญาตหรือไม่อนุญาต ในการร้องขอทรัพยากรหรือบริการต่างๆ จากโดเมน (Domain) อื่น (ในที่นี้คือ Test Tool)

2.3 HTTP Request Methods

HTTP ได้กำหนดเซตของ Request Methods (บางครั้งจะอ้างอิงถึง HTTP verbs) ไว้สำหรับระบุการทำ (Action) ที่จะกระทำกับ Server ไว้ต่างๆ โดยมี Methods ที่ใช้บ่อยๆดังนี้ GET , POST , PUT และ DELETE

2.4 JavaScript Object Notation (JSON)

JSON เป็นรูปแบบ (Format) ในการแสดงค่าของข้อมูลในรูปแบบที่มนุษย์สามารถอ่านและเขียนได้ง่าย และเป็นรูปแบบที่ใช้กันอย่างแพร่หลายในหลายภาษาโปรแกรมมิ่ง (Programming Lanaguage) โดยจะมีรูปแบบเป็นคู่ๆกันของ Key และ Value โดยที่ Value

เช่น {"name": "Peter Quill"} จะได้ Key คือ "name" และมี value เป็น "Peter Quill"

ตัวอย่างข้อมูลในรูปแบบ JSON

```
{
  "hotelName": "KMITL Resort",
  "rooms": [
    {
      "name": "Standard Rooms",
      "pricePerNight": 2000,
      "available": true
    }
  ]
}
```

```
},  
{  
  "name": "Suite Rooms",  
  "pricePerNight": 5700,  
  "available": false  
}  
],  
"city": "Bangkok",  
"country": "Thailand"  
}
```

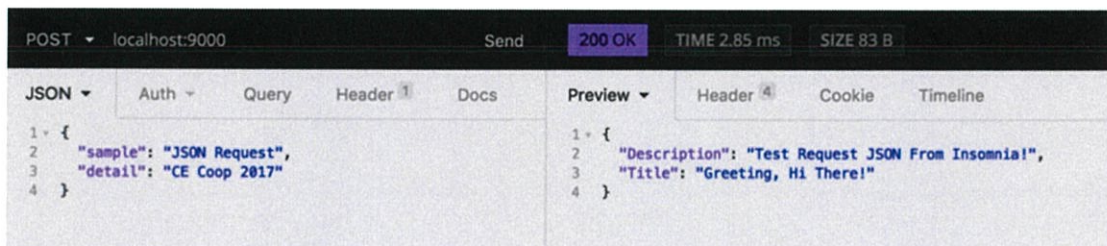
บทที่ 3

วิธีดำเนินงาน

3.1 วิธีดำเนินการทำงาน

1) ศึกษาการทำงานและโครงสร้างของเอพีไอ

ศึกษาการทำงานของ API ที่ทีมพัฒนารับผิดชอบอยู่ว่าทำงานอย่างไรมีรูปแบบโครงสร้างข้อมูล (Data Structure) ของ JSON Request และ JSON Response อย่างไร เช่นชื่อฟิลด์ต่างๆ (Field names) และลำดับชั้น (Hierarchy) ของข้อมูล และ Endpoint ต่างๆที่ต้องการใส่ไปใน Test Tool และได้ลองใช้ JSON Request ในการ Request ข้อมูลตัวอย่างจาก API Server ผ่าน POST Methods จากโปรแกรม Insomnia



ภาพที่ 3.1 ตัวอย่างการ Request ข้อมูล โดยใช้ Insomnia

2) รับทราบความต้องการของระบบเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ

ความต้องการของระบบคือเป็นเว็บแอปพลิเคชันที่พัฒนาโดยใช้ AngularJS 2 และสามารถส่ง Request (POST Methods) ไปยัง API ใน Endpoint ต่างๆเช่น

- 1) หน้าข้อมูลที่ปัก (ต่อ 1 ที่ปัก)
- 2) หน้าข้อมูลรายการที่ปัก (รายการที่ปักต่อ 1 เมือง/พื้นที่)

และมี Features ต่างๆเพิ่มเติมดังนี้

- 1) Auto-Fill Field/ Clear Fields
- 2) การใช้ Filters แบบ Hierarchy
- 3) การเช็คความแตกต่างของ JSON Request ระหว่าง 2 Target Server (JSON Comparator)
- 4) JSON-to-Field Mapper
- 5) Field-to-JSON Mapper
- 6) เก็บข้อมูล Fields ต่างๆในโครงสร้าง Model Class เพื่อง่ายต่อการพัฒนาต่อ
- 7) มีรายละเอียดข้อมูลสำหรับทุกๆ Fields และอ้างอิงกับเว็บไซต์เอกสารอธิบายเพิ่มเติม

- 8) มีตาราง Fields (Fields Table) ของ JSON Response ที่สามารถ Search-Filters ได้
- 9) มี URL Shortener สำหรับแต่ละ Request โดยเฉพาะสำหรับนำไปอ้างอิงในงานอื่นๆ
- 10) มี Function ในภาษา C# สำหรับการร้องขอ URL Shortener ของ Test Tool จาก JSON สำหรับระบบรายงานความผิดพลาดอัตโนมัติ

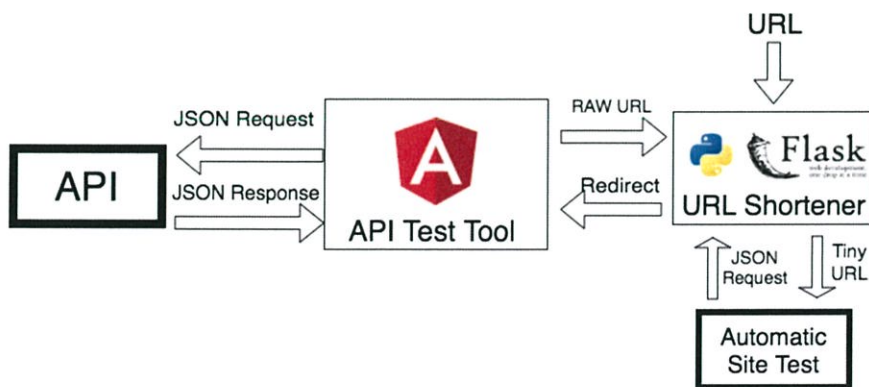
3) ศึกษาเครื่องมือที่ใช้ในการพัฒนา

ศึกษา Angular 2 ซึ่งเป็น Open Source JavaScript Framework สำหรับการสร้างเว็บแอปพลิเคชัน Test Tool ศึกษา Flask ซึ่งเป็น Python Web Framework สำหรับพัฒนา URL-Shortener API และศึกษาการใช้งาน Cross-Origin Resource Sharing (CORS) ในฝั่ง API ของทีม

4) ออกแบบระบบ

หลังจากได้ศึกษาโครงสร้างของ JSON Request ของ API แล้วก็นำมาออกแบบให้อยู่ในรูปแบบของ Model Class เพื่อความเป็นระเบียบและง่ายต่อการพัฒนาต่อจากนั้นก็ออกแบบหน้าเว็บแอปพลิเคชันด้วยภาษา HTML และทำฟังก์ชันให้เมื่อทำการตกลง(Submmit) แบบฟอร์มแล้วจะนำช่องรับข้อมูลต่างๆ(Input) ไปเทียบกับค่าตัวแปรใน Class นั้นๆ จากนั้นให้ทำการ Serializing เป็น JSON Request แล้วส่งไปยัง API

สำหรับรูปแบบการทำงานของส่วนต่างๆ จะเป็นในแผนภาพดังนี้ (โดยไม่ได้กล่าวถึงส่วนการทำงานของ API)



ภาพที่ 3.2 โครงสร้างการทำงานของระบบ

5) เขียนโปรแกรม

ในขั้นตอนการเขียนโปรแกรม ได้แบ่งขั้นการเขียนโปรแกรมออกเป็นขั้นๆ ดังนี้

- 1) ทดลองเขียน Web Application จาก AngularJS 2 ทำการ Request ข้อมูลจาก API Server (ในขั้นนี้ให้ API ขอมรับ ทุกๆ โดเมน (Allow-All Origin))
- 2) สร้างฟอร์มรับข้อมูลจากนั้นนำข้อมูลนั้นๆ ไปสร้าง JSON Request (ในขั้นตอนนี้ให้สร้าง Class สำหรับ Request แบบต่างๆ ไปด้วย)
- 3) ออกแบบหน้าแสดงผลข้อมูลผลลัพธ์ (JSON Response) ให้ตรงกับความต้องการ
- 4) เพิ่มตาราง Response Fields ที่สามารถ Search และ Filters ในหน้าผลลัพธ์
- 5) เพิ่มการกรอกข้อมูลและลบข้อมูลอัตโนมัติสำหรับหน้า Request
- 6) เขียนโปรแกรมสำหรับการเปรียบเทียบข้อมูล Response ระหว่าง 2 เซิร์ฟเวอร์ (JSON Comparator)
- 7) สร้าง JSON-to-Fields Mapper
- 8) เพิ่มการใช้ Filter แบบลำดับชั้น (Hierarchy)
- 9) พัฒนา URL Shortener และพัฒนาให้ Test Tool รองรับ URL Shortener
- 10) สร้าง Function ในภาษา C# สำหรับการร้องขอ URL Shortener ของ Test Tool สำหรับระบบรายงานความผิดพลาดอัตโนมัติ
- 11) นำ Test Tool และ URL Shortener ไปติดตั้ง (Deploy) บนเครื่อง QA Machine (ระบบปฏิบัติการ Linux)
- 12) ตั้งค่าให้ API ขอมรับการ Cross-Origin เฉพาะเครื่องที่ได้นำไปติดตั้งแล้ว

6) ทดสอบระบบ

การทดสอบระบบหลักได้ลองทดสอบโดยใช้ข้อมูลจากตัวอย่าง JSON Request ดังเดิมลองใช้กับ API Test Tool ทั้งจากการนำข้อมูลมาใส่ในฟอร์ม และการใช้ JSON-to-Fields จะได้ข้อมูลในส่วนของ JSON Request ครบถ้วนไม่มี Field ใดขาดหายไปและนำส่วนของ JSON Response มาเปรียบเทียบกับ JSON Response ที่ได้จาก Insomnia (REST API Client Application) แล้วได้ข้อมูลที่เหมือนกัน

การทดสอบ Function สำหรับการสร้าง Shortened URL ของ Test Tool ในภาษา C# สำหรับระบบรายงานความผิดพลาดอัตโนมัติ ได้ลองให้ระบบสร้าง Shortened URL และนำมาเปิดได้ JSON Request ตรงกับ JSON Request ต้นฉบับในการสร้าง Shortened URL

ในการทดสอบ JSON Comparator ได้ลองสร้างข้อมูล JSON Response เติงขึ้นมาจากข้อมูลเดิมโดยมีการแก้ไขในบาง Fields นั้น JSON Comparator สามารถตรวจพบความแตกต่างนั้นได้

7) สาริระบบแก้ที่ม

สาริวิธีการใช้งานเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ (Web Application for API Testing) รวมถึงสาริตถึงความสามารถต่างๆ ที่กล่าวมาข้างต้น ได้รับความถึงพอใจ

8) จัดทำรายงาน

จัดทำรายงานสหกิจศึกษา และ รายงานวิธีการใช้งานและการพัฒนาเพิ่มเติม

3.2 แผนการดำเนินงาน

ตารางที่ 3.1 แผนการดำเนินงาน

ลำดับ	หัวข้องาน	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ศึกษาการทำงานและโครงสร้างของเอพีไอ	■	■														
2	รับทราบความต้องการของระบบเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอ		■	■													
3	ศึกษาเครื่องมือที่ใช้ในการพัฒนา				■				■								
4	ออกแบบระบบ				■	■	■	■	■								
5	เขียนโปรแกรม					■	■	■	■	■	■	■					
6	ทดสอบระบบ										■	■					
7	สาธิตระบบแก่ทีม													■			
8	จัดทำรายงาน													■	■		

บทที่ 4

ผลการทำงาน

4.1 ผลการทำงานในฝั่ง API Test Tool

- 1) ใส่ค่าลงในช่องรับค่าและ Submit จะได้ JSON Request ที่ถูกต้อง
- 2) สามารถแสดงผลค่า Response ได้ถูกต้อง
- 3) JSON-to-Field สามารถทำงานได้ถูกต้อง
- 4) Filters แบบเป็นลำดับขั้นสามารถทำงานได้ถูกต้อง
- 5) สามารถเปรียบเทียบความต่างของ 2 Server ที่ JSON Response ข้อมูลไม่เหมือนกันได้

4.2 ผลการทำงานในฝั่ง URL Shortener

- 1) สามารถสร้าง Shortened URL (Tiny URL) และสามารถเปิดเข้าไปได้ข้อมูลถูกต้อง
- 2) สามารถสร้าง Shortened URL จาก JSON Request สำหรับ Automatic Site Test ได้ถูกต้อง

4.3 ผลการทำงานในฝั่ง API (เรื่อง Cross-Origin)

- 1) API ปฏิเสธ JSON Request จากโดเมนที่ไม่ได้ระบุไว้
- 2) API ยอมรับ JSON Request จากโดเมนที่ระบุไว้ได้ถูกต้อง (ในที่นี้คือ API Test Tool)

บทที่ 5

สรุปผลการทำงาน และข้อเสนอแนะ

5.1 บทสรุป

ระบบเว็บแอปพลิเคชันสำหรับทดสอบเอพีไอสามารถใช้งานได้ตรงตามความต้องการของทีมอย่างครบถ้วน

5.2 ปัญหาอุปสรรค และแนวทางแก้ไข

- 1) เนื่องจากการทดสอบในบางรูปแบบใช้ JSON Response ที่เป็นชนิดข้อความ (String) จะได้ผลลัพธ์ดีกว่าจึงเพิ่มฟังก์ชันการคัดลอกลงคลิปบอร์ดและทางลัดสู่เว็บไซต์ “JSON Editor Online”
- 2) เนื่องจาก JSON Request มีจำนวน Fields ที่ใช้เยอะมากจึงทำให้ในหน้ารับข้อมูลของ API Test Tool มีจำนวน Fields มารับข้อมูลมากขึ้นจึงอาจไม่เป็นมิตรกับผู้ใช้มากนัก จึงมีฟังก์ชันการใส่ค่าเริ่มต้น และลบค่าเริ่มต้นเพื่อให้ใช้ได้ง่ายยิ่งขึ้น

5.3 แนวทางการพัฒนาต่อ

- 1) แก้ไขหน้ารับข้อมูลให้ใช้ง่ายมากขึ้น
- 2) เพิ่มการทำงานในส่วนของ Text-Search
- 3) เพิ่มความเร็วให้กับ API Test Tool โดยการลดขั้นตอนที่ไม่สำคัญออกไป

บรรณานุกรม

- [1] W3Schools. *JSON – Introduction*. Retrieved December 8, 2017, from https://www.w3schools.com/js/js_json_intro.asp
- [2] Mozilla. (2017) *Cross-Origin Resource Sharing (CORS) – HTTP*. Retrieved December 8, 2017, from <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- [3] jkinkead. (2014) *Handling CORS headers with spray-routing*. Retrieved December 8, 2017, from <http://jkinkead.blogspot.com/2014/11/handling-cors-headers-with-spray-routing.html>
- [4] Muhammad hassan. (2016) *ANGULAR2: HTTP POST WITH COMPLEX OBJECT USING OBSERVABLE IN ANGULAR 2 APPLICATION*. Retrieved December 8, 2017, from <https://hassantariqblog.wordpress.com/2016/12/03/angular2-http-post-with-complex-object-using-observable-in-angular-2-application/>
- [5] Gary Simon (2016), *How to Install Angular 2 with the Angular-cli*, Retrieved December 8, 2017, from <https://coursetro.com/posts/code/16/How-to-Install-Angular-2-with-the-Angular-cli>
- [6] Nuttavut Thongjor (2016), *สอนสร้างและใช้งานเว็บแอปพลิเคชันด้วย Angular2*, Retrieved December 8, 2017, from <https://www.babelcoder.com/blog/series/angular2>
- [7] Armin Ronacher, *Welcome to Flask*, Retrieved December 8, 2017, from <http://flask.pocoo.org/docs/0.12/>
- [8] Isara Naranirattisai (2017), *JSON Comparator*, Retrieved December 8, 2017, from <https://github.com/DreamN/JSON-Comparator>