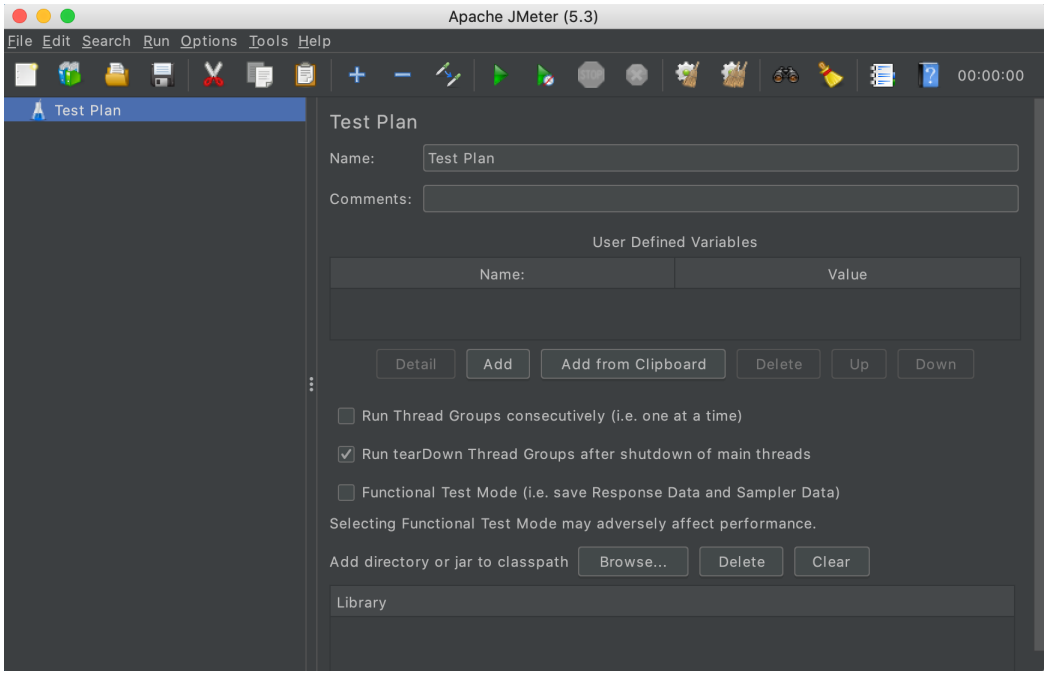
Lab 8 : Performance Testing with JMeter

01418471 Introduction to Software Engineering

I. การ download การ install และการเปิดโปรแกรม JMeter

1. Download โปรแกรม Apache JMeter ได้จาก http://jmeter.apache.org/
2. unzip ไฟล์ไว้ที่ใดที่หนึ่ง จะได้ folder apache-jmeter-x.0/
3. ไปที่ folder apache-jmeter-x.0/bin
4. คลิกเปิด ApacheJMeter.jar จะได้หน้าจอ ตามรูปด้านล่าง
    - ถ้าคลิก แล้วไม่มีหน้าจอ ให้รันจาก command line ดังนี้ (ซึ่งควรได้หน้าจอ ตามรูปด้านล่างเช่นกัน)
    - น่าจะใช้กับ Java 16 ไม่ได้ค่ะ ถ้านิสิตมี java 16 เป็น default ให้รันจาก java version อื่น

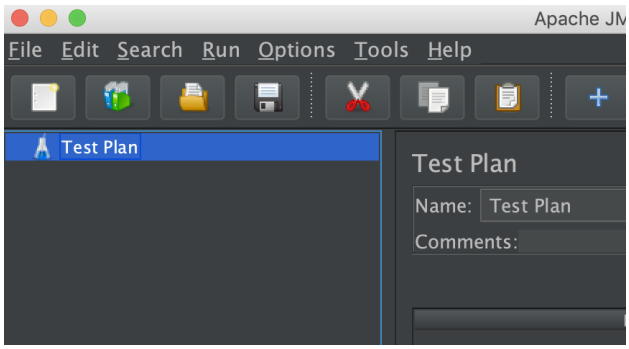
```
$ java -jar ApacheJMeter.jar
```



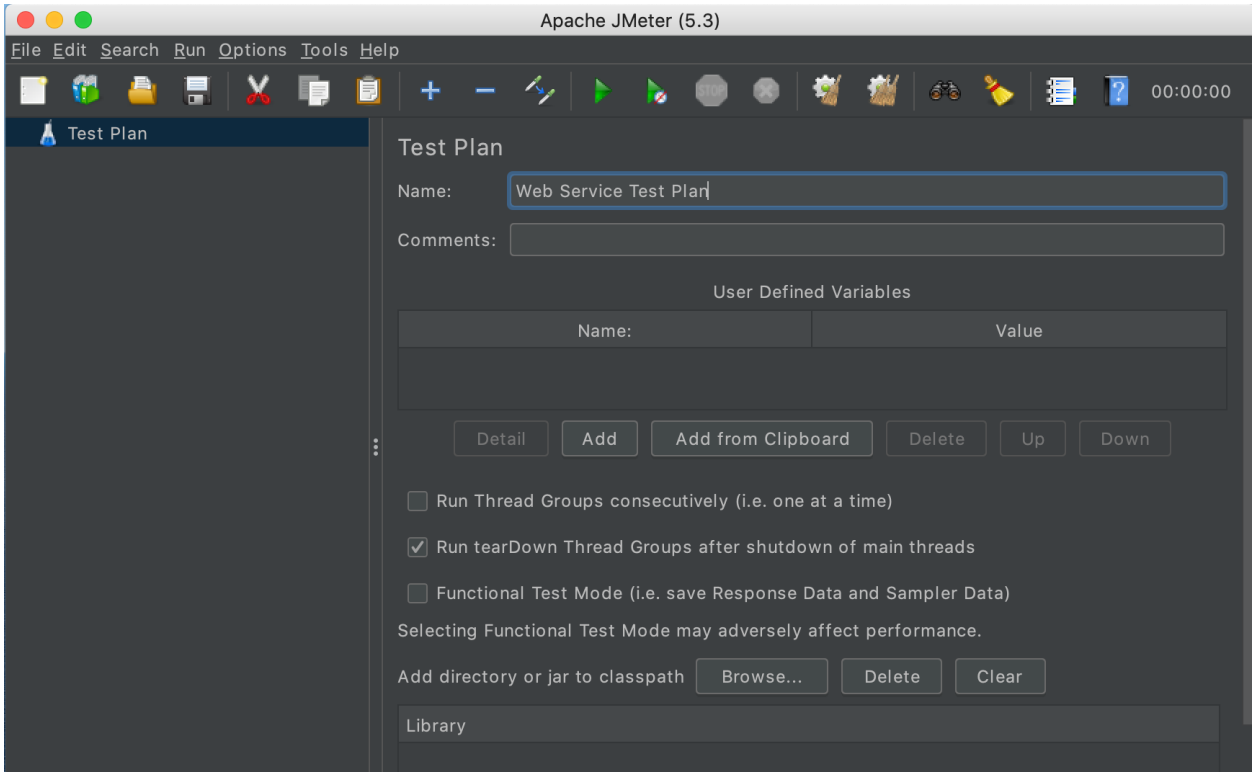
II. การ configure เบื้องต้นในการทดสอบประสิทธิภาพ web application

II.(I) การ configure Test Plan

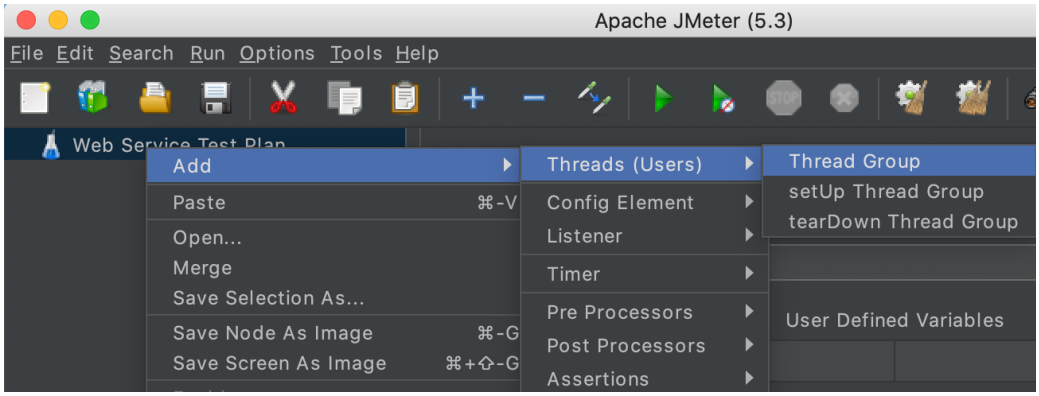
1. เมื่อเปิดโปรแกรม จะพบมีเมนูซ้ายมือและหน้าจอแสดงผลด้านขวามือ เมนูด้านซ้ายมือประกอบไปด้วย
   - Test Plan ซึ่งเป็นเมนูสำหรับการ config หลัก



2. คลิกไปที่เมนู Test Plan จะได้ผลทางขวามือตามภาพด้านล่าง แล้วจึงเปลี่ยนชื่อ Test Plan ให้สื่อ เช่น Web Service Test Plan หรือ Web Application Test Plan



3. เพิ่ม thread group โดยคลิกขวาไปที่ Test Plan แล้วเลือกเมนู Add --> Threads (Users) --> Thread Group ดังภาพต่อไปนี้ แล้วตั้งชื่อ Thread Group ตามความเหมาะสมในหน้าแสดงผลด้านขวามือ



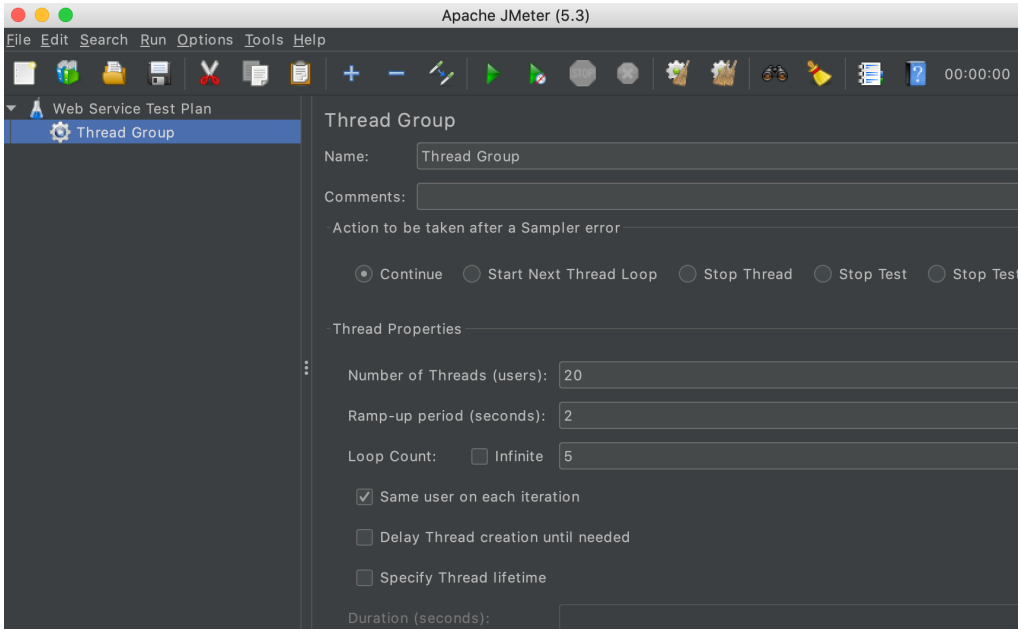
4. Configure Thread group โดยกำหนดจำนวนผู้ใช้ อัตราการเพิ่มผู้ใช้ การรันซ้ำ ตามภาพต่อไปนี้ โดยที่

- ***Number of Threads (users)***: เป็นจำนวน virtual users ทั้งหมดที่เราต้องการจะ simulate แต่ไม่ได้บอกว่าจะยิงเข้าไปเท่าไหร่ อย่างไร
- ***Ramp-Up Period***: ใน user ทั้งหมด จะให้ยิงไปทั้งหมดกี่วินาที
- ***Loop Count***: ยิง user ทั้งหมดกี่รอบ (การทำซ้ำตามรอบที่กำหนด)

ตัวอย่าง

- ถ้าให้ Number of Threads เป็น 20 และ
- Ramp-Up Period เป็น 2 และ
- Loop Count เป็น 5
- JMeter จะยิง 20 users ต่อ 2 วินาที (ซึ่งอาจจะเป็นว่า ในทุก ๆ วินาทีจะมี 10 users ยิงเข้าไป) และจะยิงแบบนี้ไป 5 รอบ
- ดังนั้น JMeter จะยิง users รวมเป็นทั้งหมด 100 users (20 * 5)

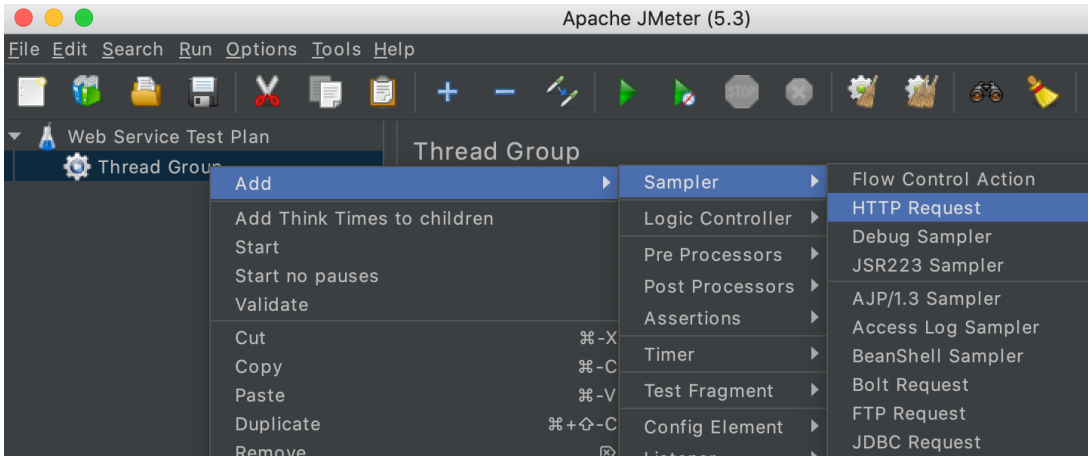
หากติ๊กตรง Loop Count ที่ Infinite จะทำให้ JMeter ยิง users ไปเรื่อย ๆ จนกว่าผู้ทดสอบจะสั่งให้หยุด



II.(II) การเพิ่ม Sampler

Sampler เป็นตัวสร้าง request จำลองต่าง ๆ และส่งไปที่ server ของเราตามจำนวนที่กำหนดไว้ใน thread group เพื่อทดสอบว่า server สามารถรองรับจำนวน request ตามที่กำหนดหรือไม่ มี sampler หลากหลายอย่างให้เราเลือกทำ performance testing เช่น HTTP Request, JDBC request เป็นต้น

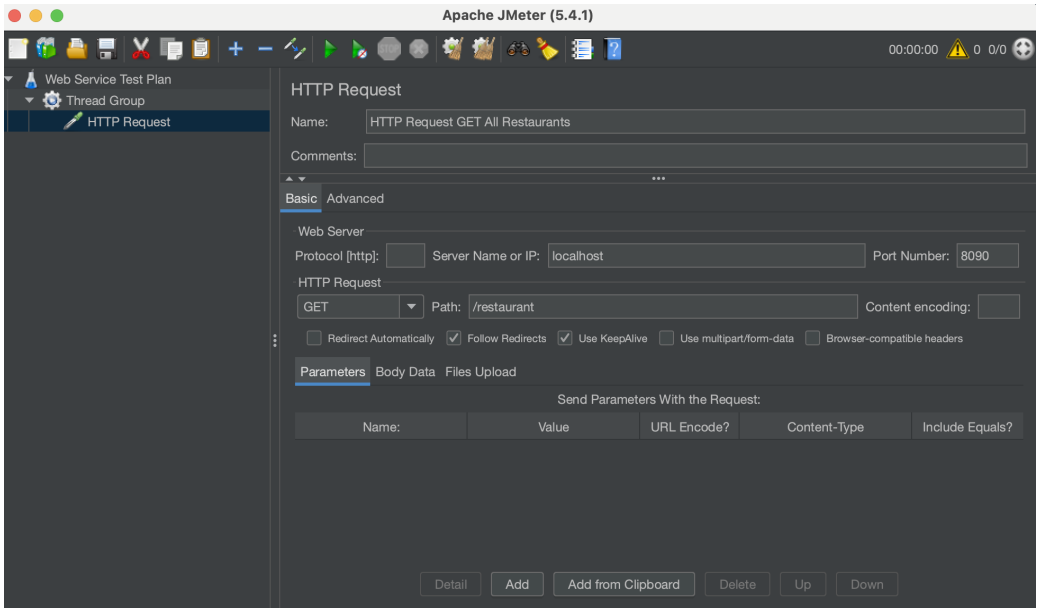
7. ขั้นตอนต่อไปจะเป็นการเพิ่ม url ของ web service ที่เราต้องการจะทดสอบ ให้คลิกขวาที่ Thread Group แล้วไปที่เมนู Add --> Sampler --> HTTP Request ตามภาพต่อไปนี้



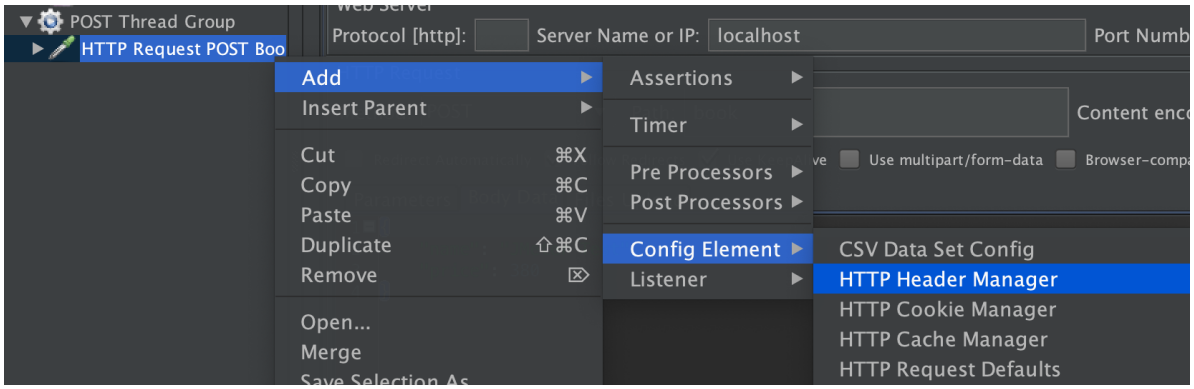
8. เราจะส่ง GET request ไปที่ web API service ของนิสิต โดยนิสิตสามารถใช้

- Web API Menu ที่นิสิต implement มาเอง
- หรือ https://github.com/ladyusa/book-spring-boot-jpa ตามในวิดีโอคลิป
- หรือ https://github.com/ladyusa/restaurant-api-exp (port: 8090) (path: /restaurant) ตามใน lab sheet นี้

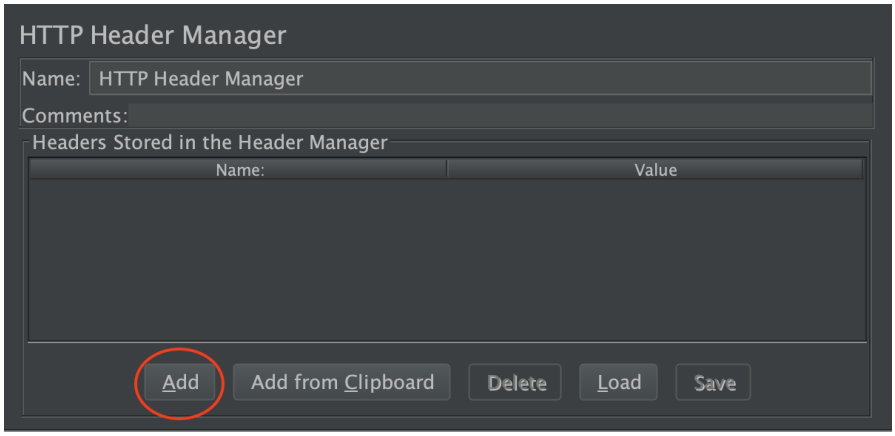
โดยให้ระบุ IP, port, Method HTTP Request เป็น GET และ path ให้ตรงกับ service ของนิสิต) และตั้งชื่อ เช่น HTTP Request GET All Menu หรือ HTTP Request GET All Restaurant ดังนี้



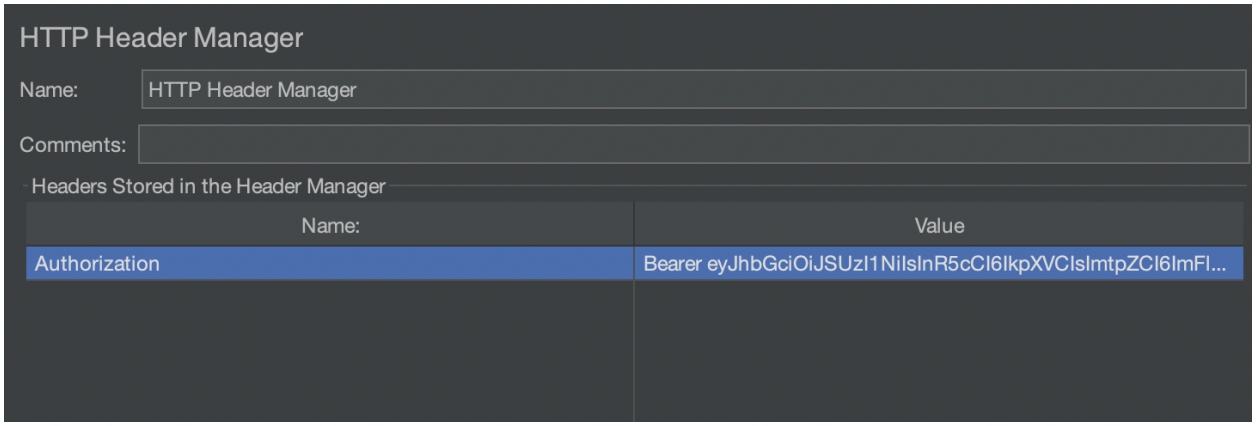
9. ถ้าต้องส่ง JWT token ไปกับ request ด้วย ต้องเพิ่ม header ของ request โดยคลิกขวาที่ Sampler สำหรับ request แล้วเลือก Add → Config Element → HTTP Header Manager



จากนั้น เมื่อได้ HTTP Header Manager มาแล้ว ให้กดปุ่ม Add



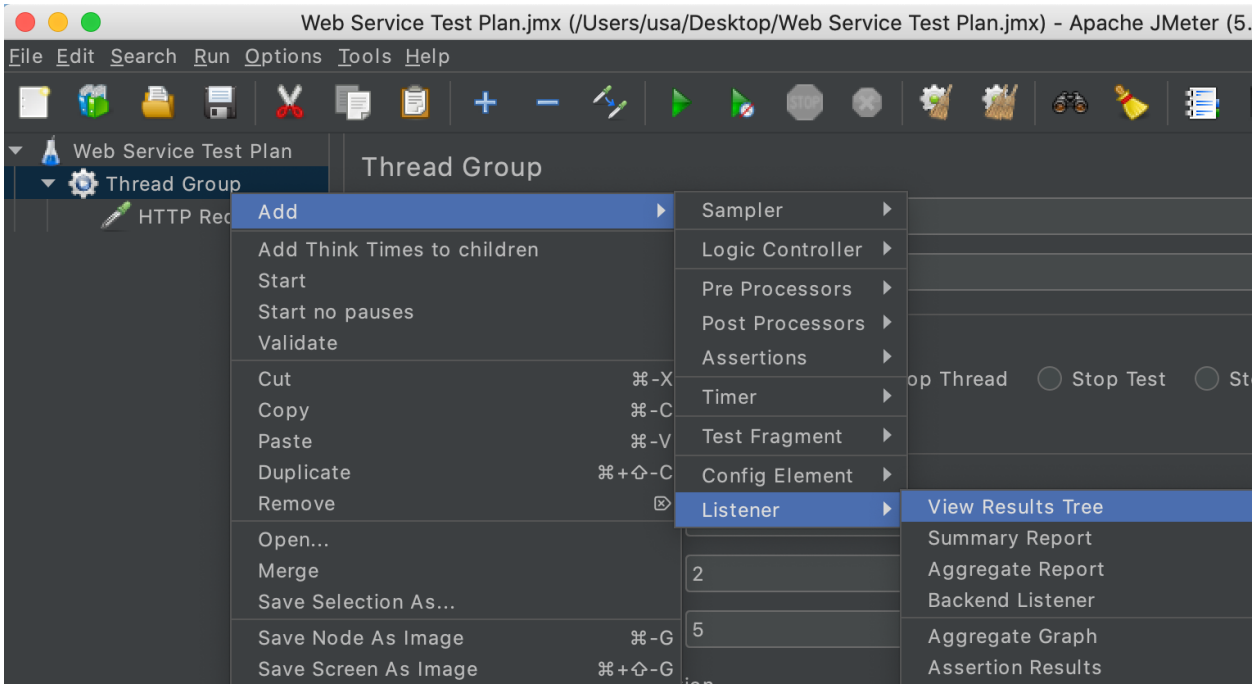
และเพิ่ม header ดังนี้
Name: Authorization
Value: Bearer YOUR_TOKEN



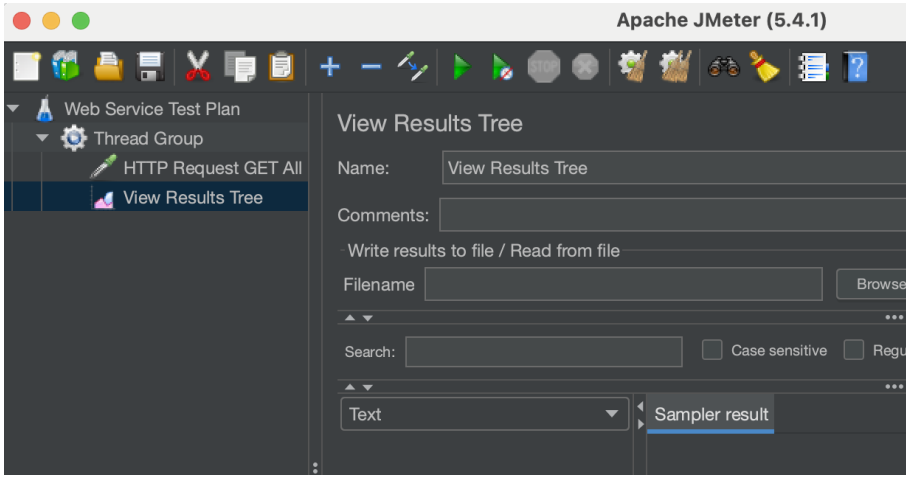
II.(III). การเพิ่ม Listener

Listener เป็นตัวดักฟังการทำงานของ web service ที่เราทดสอบเพื่อเอาข้อมูลการทำงานมาแสดงผลหรือวิเคราะห์ต่อ Listener ของ JMeter มีหลายตัว แต่ตัวพื้นฐานคือ View Results Tree, Summary Report, Graph Results, Response Time Graph

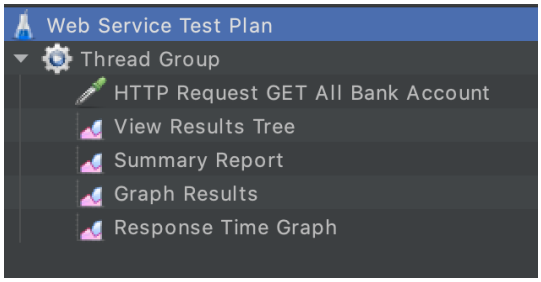
1. การเพิ่ม Listener จะทำได้โดยการคลิกขวาที่ Test Plan แล้วเลือกเมนู Add --> Listener --> View Results Tree ดังนี้



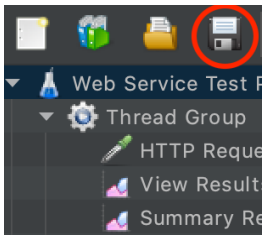
2. เมื่อเพิ่มแล้ว Listener จะปรากฏภายใต้ Test Plan ดังนี้



3. ลองเพิ่ม listener อื่น เช่น Summary Report, Graph Results, Response Time Graph ฯลฯ



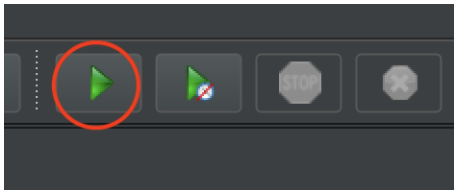
4. อย่าลืม Save กัน (ก่อนกดปุ่ม Save มา highlight ที่ Test Plan ก่อน)



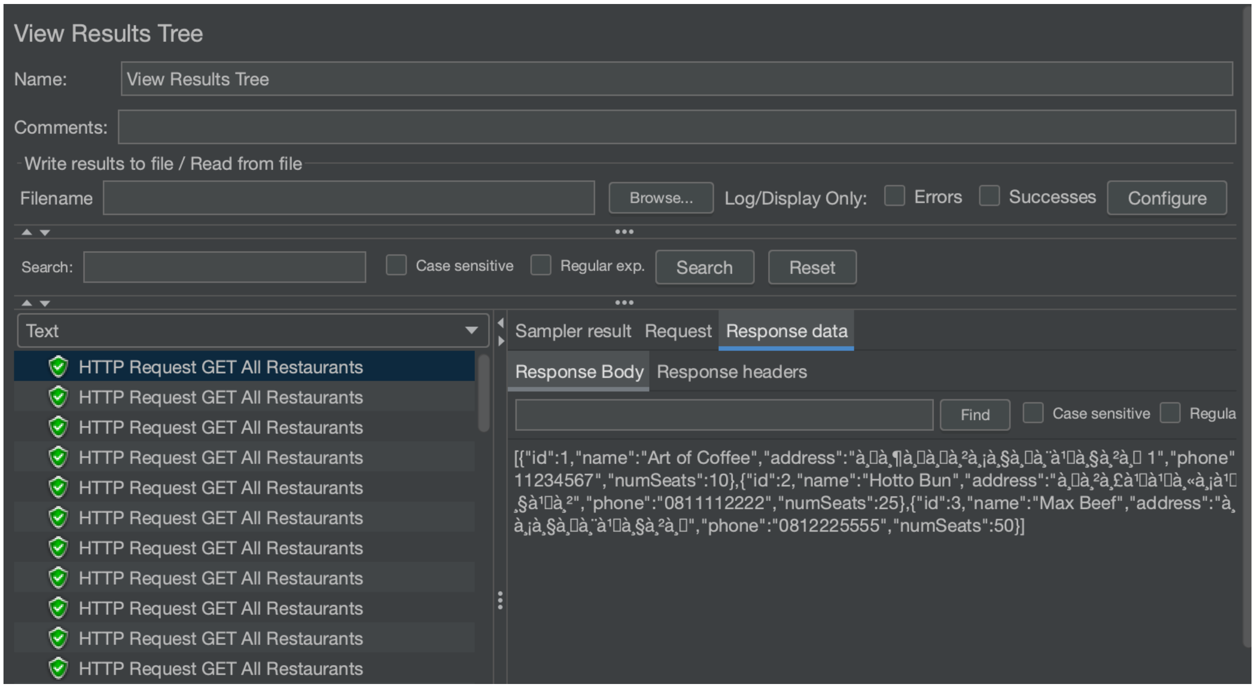
III. การรัน Performance Test และการวิเคราะห์ผลการรัน

0. เมื่อพร้อมทดสอบ ให้รัน database และโปรแกรม web API service ของนิสิต

1. เมื่อต้องการรัน ให้กดปุ่ม Play

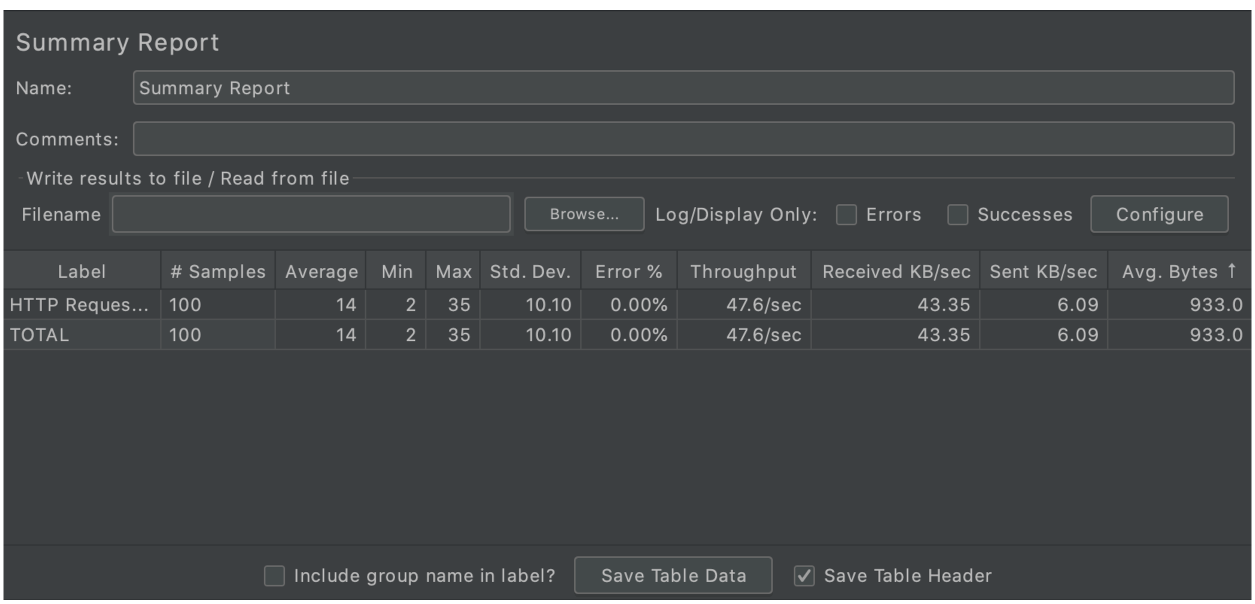


2. ตัวอย่างผลจาก View Results Tree ในหน้าถัดไป ซึ่งจะแสดงข้อมูลเกี่ยวกับการส่ง HTTP Request เช่น ส่ง request อะไรไป ได้ response data อะไรมาบ้าง header มีข้อมูลใดบ้าง เป็นต้น หากต้องการเห็นผลการส่ง request ตรงหน้าต่างด้านขวามือ คลิกที่แท็บ Response data และ Response Body จะเห็นข้อมูลที่ web service ส่งกลับมาในรูปแบบ JSON

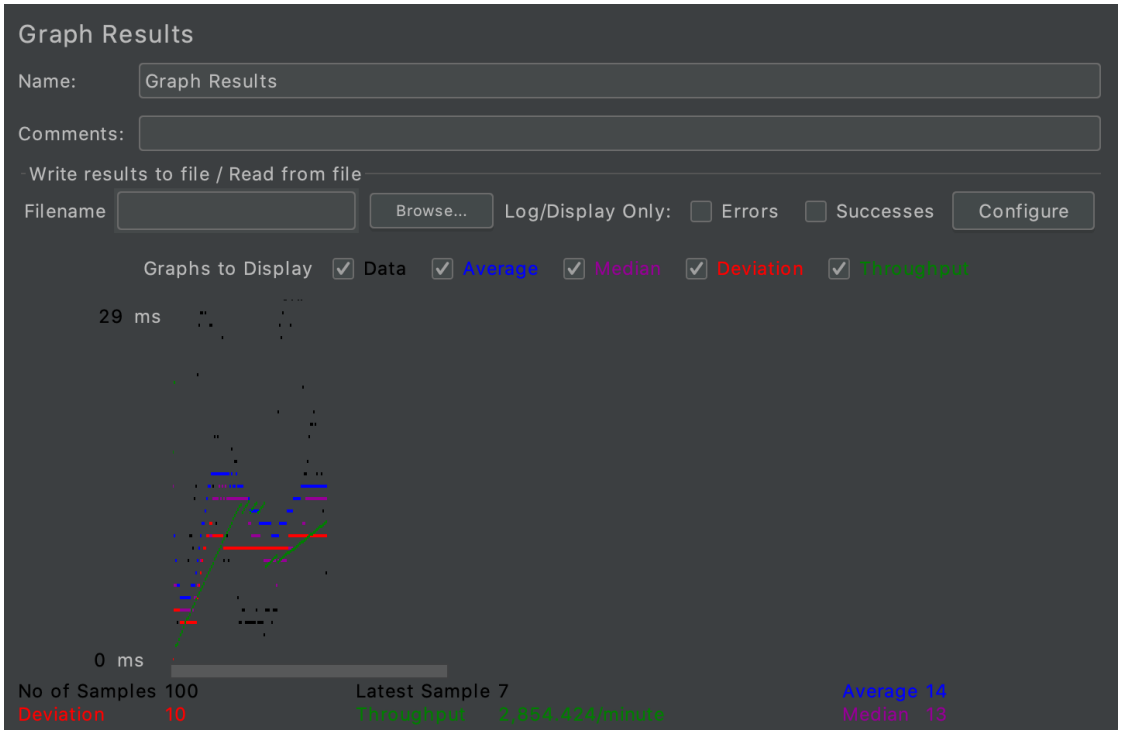


3. ผลที่ได้จาก Summary Report จะมีผลดังนี้

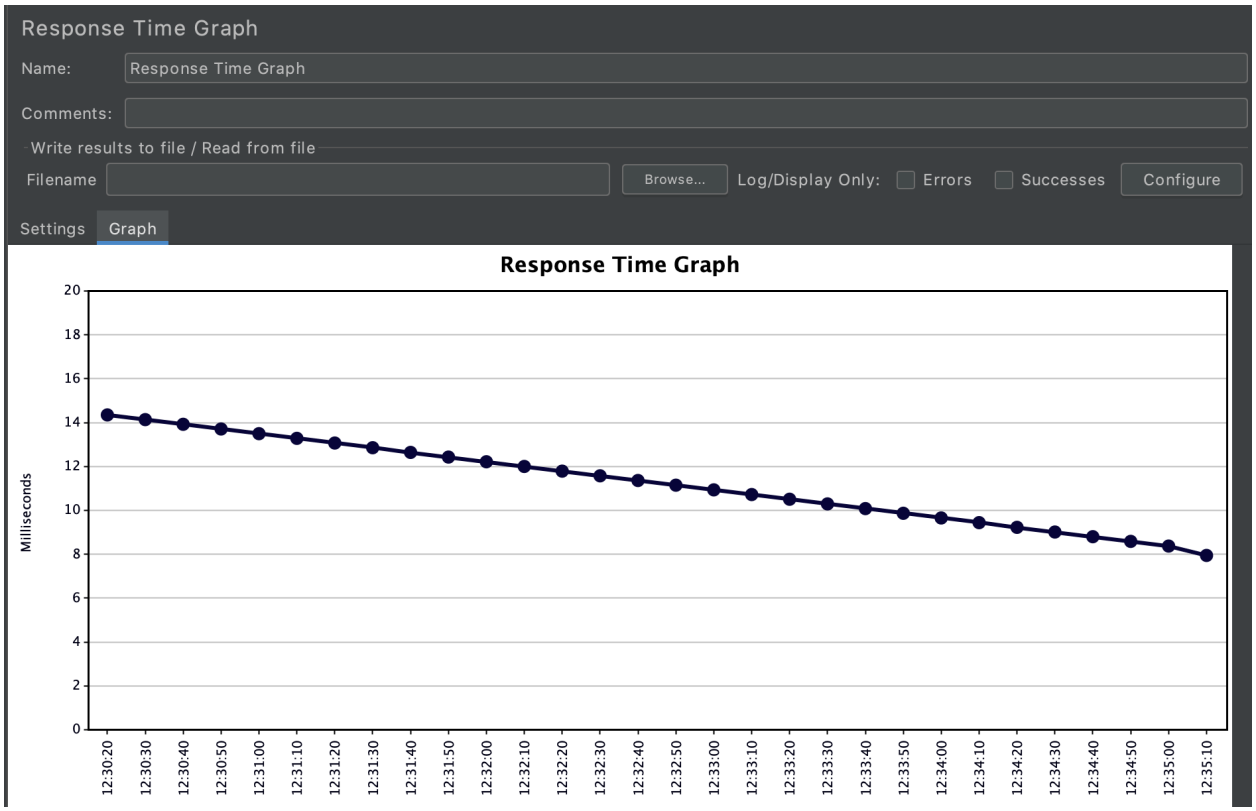
- **Label** : ชื่อ Sampler
- **#Samples** : จำนวน users
- **Average, Min, Max** : ผลทางสถิติของ response time
- **Error %** : เปอร์เซ็นของ request ที่ไม่ได้รับ response
- **Throughput** : อัตราการส่งต่อหน่วยเวลา



4. ผลที่ได้จาก Graph Results

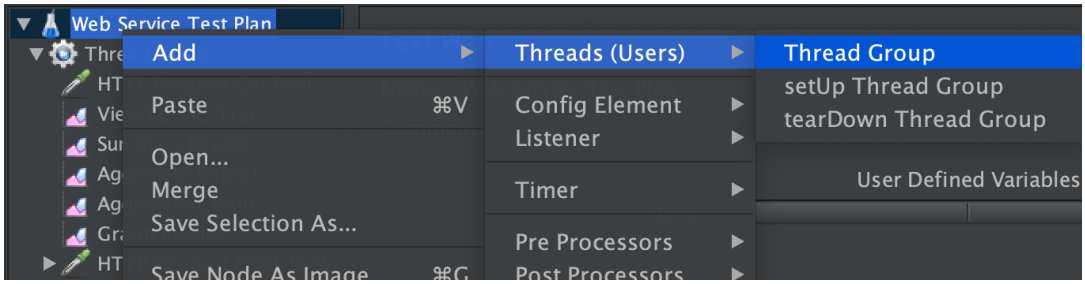


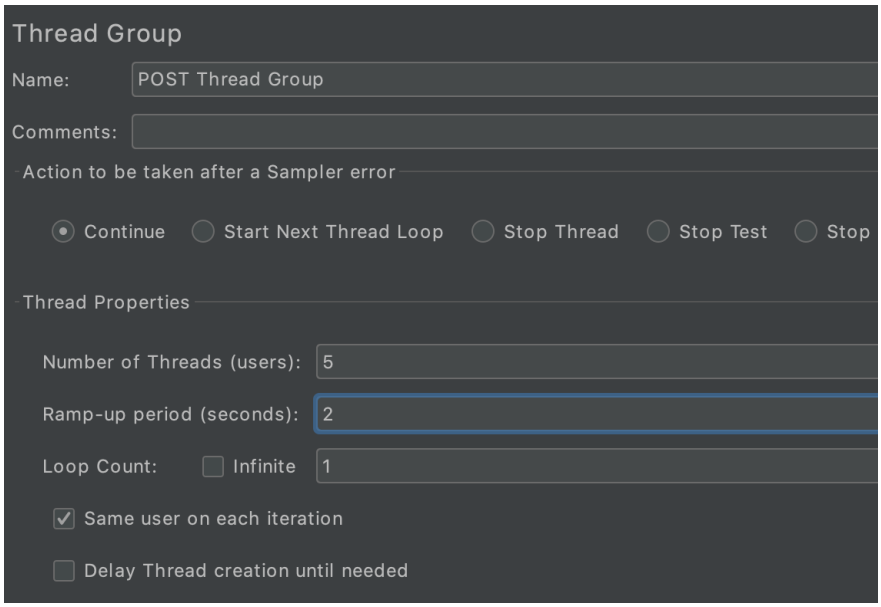
5. ผลที่ได้จาก Response Time Graph



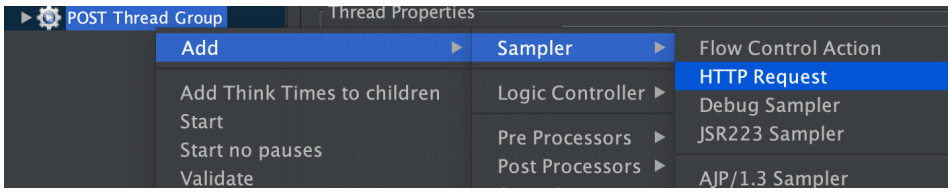
IV. ทดสอบ performance การส่ง POST Request

1. เราจะสร้าง Thread Group ให้สำหรับ POST (จะได้แยกจาก GET thread group) โดยจะลองจาก users น้อยๆ ก่อน





2. จากนั้นจะเพิ่ม Sampler สำหรับ POST request โดยเลือก Add → Sampler → HTTP Request



3. ตั้งชื่อ Sampler สำหรับ POST request ใส่ server, port และเลือก POST method ด้วย path ของเรา
   - นอกจากนั้น เลือก tab “Body Data” ด้านล่าง แล้วใส่ข้อมูลในรูปแบบ json (หรือใช้ข้อมูลอื่นก็ได้)

- สำหรับ menu service

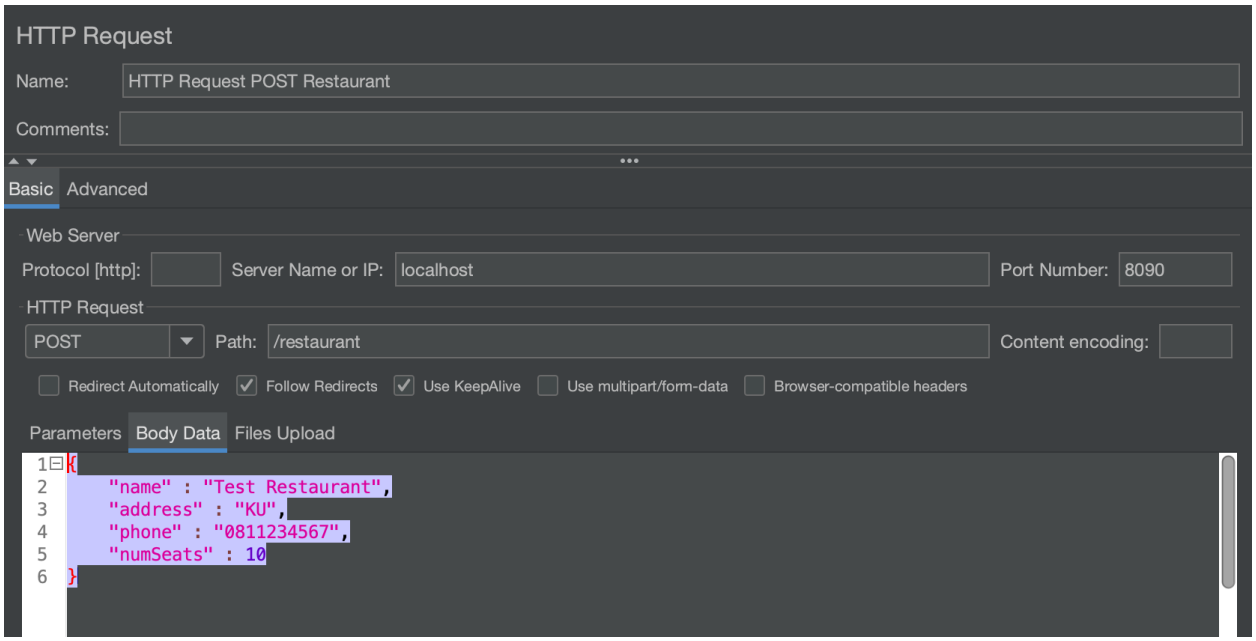
```
{
   "name" : "Cheesecake",
   "price" : 50.0,
   "category" : "Dessert"
}
```

- สำหรับ book service

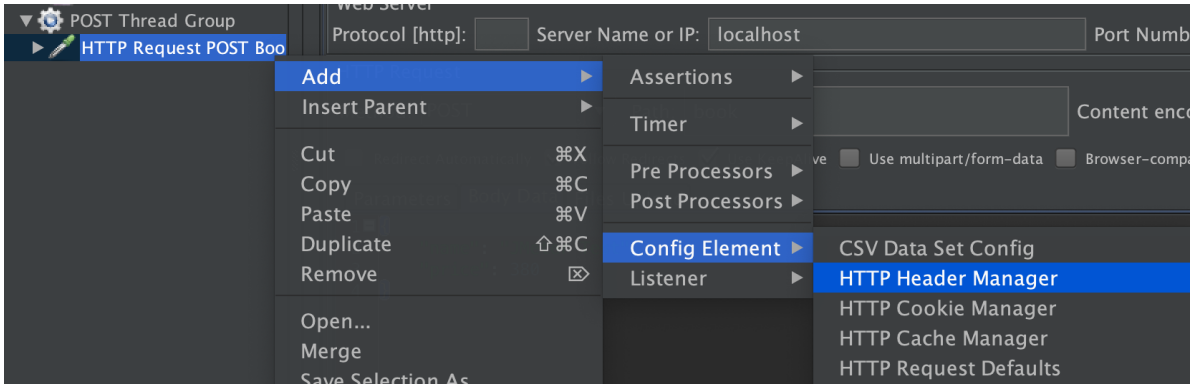
```
{
    "name": "Data Science",
    "price": 400
}
```

- สำหรับ restaurant service

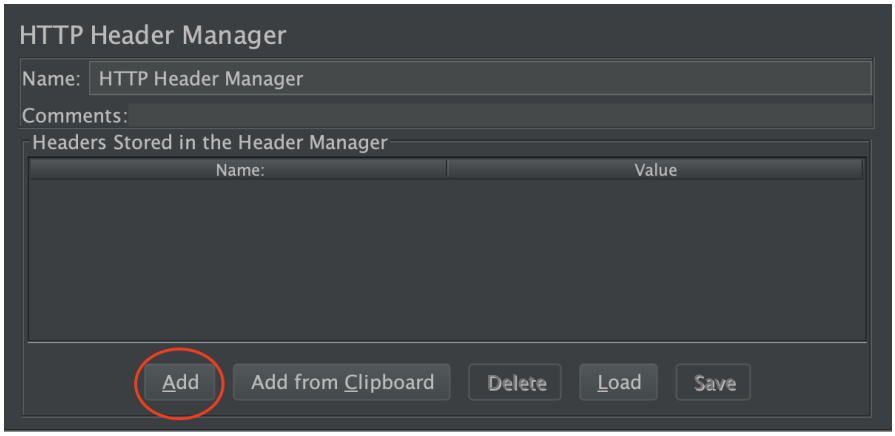
```
{
       "name" : "Test Restaurant",
       "address": "KU",
       "phone" : "0811234567",
       "numSeats" : 10
}
```



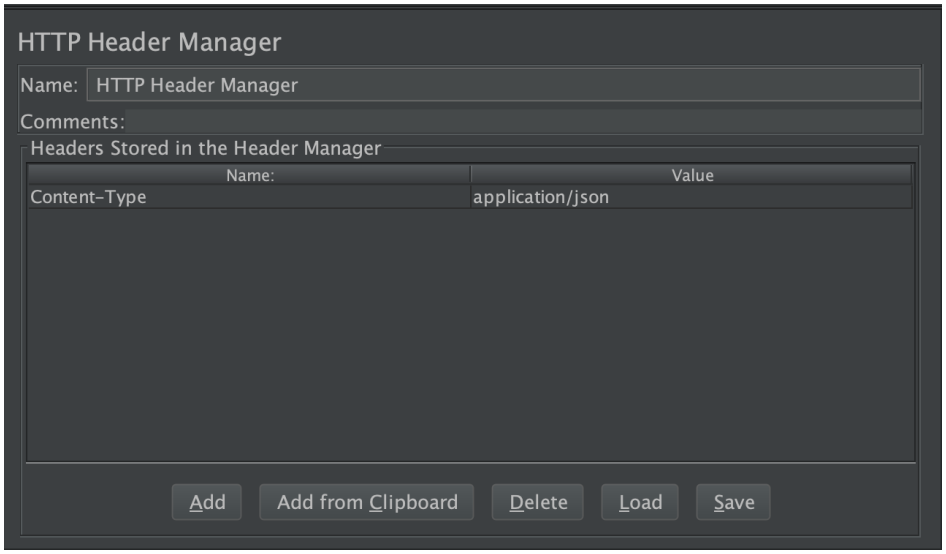
4. ต้องเพิ่ม header ของ POST request โดยคลิกขวาที่ Sampler สำหรับ POST request แล้วเลือก Add → Config Element → HTTP Header Manager



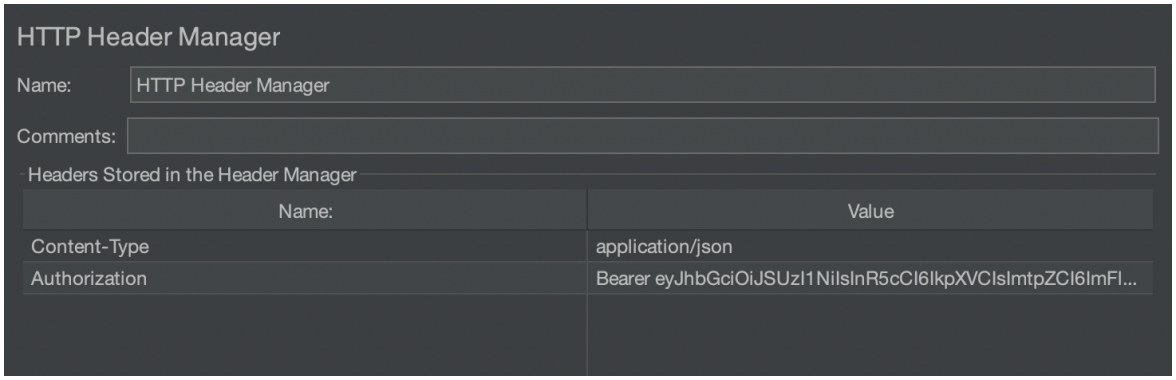
5. ต้องเพิ่ม header ของ POST request โดยกดปุ่ม Add



6. เพิ่มข้อมูล header โดยระบุประเภทข้อมูลที่จะส่งให้เป็น json
Name: Content-Type
Value: application/json



ถ้าต้องการส่ง JWT token ไปกับ request ด้วย ให้เพิ่ม header ดังนี้
Name: Authorization
Value: Bearer YOUR_TOKEN



7. Click ขวาที่ POST thread group แล้ว กดเลือกการเพิ่ม Listener ต่าง ๆ ที่ต้องการ เช่น View Results Tree, Summary Report ฯลฯ

8. Save และลองรัน
   - นิสิตสามารถเลือกรันแค่ POST Thread Group ได้ โดย highlight ที่ POST Thread Group แล้วกดรัน
   - หากเรา highlight ที่ Test Plan จะทำให้ JMeter รันทั้ง GET Thread Group และ POST Thread Group พร้อมกัน

9. ลองปรับจำนวน request ใน Thread Group และสังเกตผลการทำงาน

V. ทดสอบ performance/load testing ผ่าน command line

- ปกติ เราใช้ JMeter GUI เพื่อสร้าง test plan และลอง test สั้น ๆ เท่านั้น
- เมื่อต้องทำ performance/load testing จริง ๆ เราต้องสั่งผ่าน command line
- ผลการรันจะถูก save ใส่ไฟล์ในรูปแบบ csv

Mac OS:

```
$ jmeter -n -t [jmx file] -l [results file]
```

เช่น

```
$ jmeter -n -t perf-test.jmx -l result.csv
$ ./jmeter -n -t ~/Desktop/Web\ Service\ Test\ Plan.jmx -l
~/Desktop/result.csv
```

Windows: ใช้ jmeter.bat ด้วยคำสั่งเดียวกับ Mac OS คือ

```
$ jmeter.bat -n -t perf-test.jmx -l result.csv
```

result

timeStamp	elapsed	label	responseCode	responseMessage	threadName	dataType	success	failureMessage	bytes	sentBytes	grpThreads	allThreads
1603865111851	52	HTTP Request POST Bank Account	200		POST Thread Group 2-2	text	TRUE		231	255	2	4
1603865111851	52	HTTP Request GET All Bank Account	200		Thread Group 1-2	text	TRUE		19450	130	2	4
1603865111851	52	HTTP Request POST Bank Account	200		POST Thread Group 2-1	text	TRUE		231	255	2	4
1603865111851	52	HTTP Request GET All Bank Account	200		Thread Group 1-1	text	TRUE		19568	130	2	4
1603865111905	4	HTTP Request POST Bank Account	200		POST Thread Group 2-1	text	TRUE		231	255	2	4
1603865111905	4	HTTP Request POST Bank Account	200		POST Thread Group 2-2	text	TRUE		231	255	2	4
1603865111905	4	HTTP Request GET All Bank Account	200		Thread Group 1-2	text	TRUE		19568	130	2	4
1603865111909	3	HTTP Request POST Bank Account	200		POST Thread Group 2-2	text	TRUE		231	255	2	4
1603865111910	4	HTTP Request GET All Bank Account	200		Thread Group 1-2	text	TRUE		19686	130	2	4
1603865111909	6	HTTP Request POST Bank Account	200		POST Thread Group 2-1	text	TRUE		231	255	2	4
1603865111913	3	HTTP Request POST Bank Account	200		POST Thread Group 2-2	text	TRUE		231	255	2	4
1603865111914	5	HTTP Request GET All Bank Account	200		Thread Group 1-2	text	TRUE		19863	130	3	5
1603865111916	4	HTTP Request POST Bank Account	200		POST Thread Group 2-2	text	TRUE		231	255	2	5