

Brief of Assignment 3

Academic Year	2/2567
Assignment title	Simple API with Scala (Individual Work)
Assessor(s)	Dr. Issarapong Khuankrue and/or Teaching Assistant(s)
Issue Date	March 4, 2025
Submission Date	March 11, 2025 ก่อน 23.59
Score	10 Points
Course Learning Outcomes	CLO5 : สามารถเลือกใช้การเขียนโปรแกรมเชิงฟังก์ชันตามโจทย์ที่กำหนดได้
Student ID/ Name	66070286 ปนัสยา บุญประกอบ

Assignment Brief and Guidance:

Submission Format:

- ส่งไฟล์นี้ โดยตั้งชื่อนี้เป็น student_id.docx
- ส่งภาพที่ export จาก miro เป็น task1_student_id.pdf
- ส่ง source code เป็น task2_student_id.zip
- ส่งไฟล์ data.txt เพื่อประกอบการทดสอบ code

Task 1: ออกรอบแบบ API โดยใช้ข้อมูลใกล้ตัวนักศึกษา (ทำเป็นกลุ่มได้) โดยแสดงผลลัพธ์ที่นี่ (5คะแนน)

โดยต้องแสดง

- บริบท (Context) ของข้อมูล
- Methods
- Resources
- Requests / Responses

เกณฑ์ประเมินสำหรับ Task 1

	>80%	60 – 40 %	20 - 40%	< 20%
Design API	สามารถออกแบบตามโจทย์ที่กำหนดได้ครบถ้วน เหมาะสม	สามารถออกแบบตามโจทย์ที่กำหนดได้แต่ยังขาดความสมบูรณ์ตามที่โจทย์	สามารถออกแบบตามโจทย์ที่กำหนดได้บางส่วน	ไม่สามารถออกแบบ API ได้
ได้รับคะแนน	5	3-4	1-2	0

วางแผนแพล็พซ์ที่นี่			
API Bounded context			
Context: ข้อมูลเสื้อของคุณ			
Resources ข้อมูลเสื้อของคุณ https://www.wongnai.com/articles/lucky-color-shirt { id INT, day TEXT, colors TEXT, description TEXT }	Parameters: GET: /days/:id และ /days POST: /days PUT: /days/:id DELETE: /days/:id	Requests GET: /days PUT: /days/:id body:{ id: day: color: description: } POST: /days body:{ id: 1, "day": "monday", "color": "gold", "description": "lucky" } DELETE: /days/:id body:{ id: 1, "day": "monday", "color": "yellow", "description": "money" }	Responses: POST: /days สถานะ data added successfully PUT: /days/:id สถานะ "data updated successfully" DELETE: /days/:id สถานะ "data deleted successfully"
Methods GET POST PUT DELETE			Notes:
อินเทิร่า ชั้นประถักร 66070228 นีโน่ ชั้นหัวค่าจ้าง 66070248 บันสยา บุญประกอบ 66070286			

Task 2: สร้าง API โดยพิจารณาตัวอย่างดังต่อไปนี้ (5 คะแนน)

Example : AkkaHttp

1. สร้างโปรเจคใหม่ เลือก template

```
> sbt new akka/akka-http-quickstart-scala.g8
```

2. ตั้งชื่อโปรเจคใหม่ แต่ไม่จำเป็นต้องกำหนด version ต่างๆ ของ library

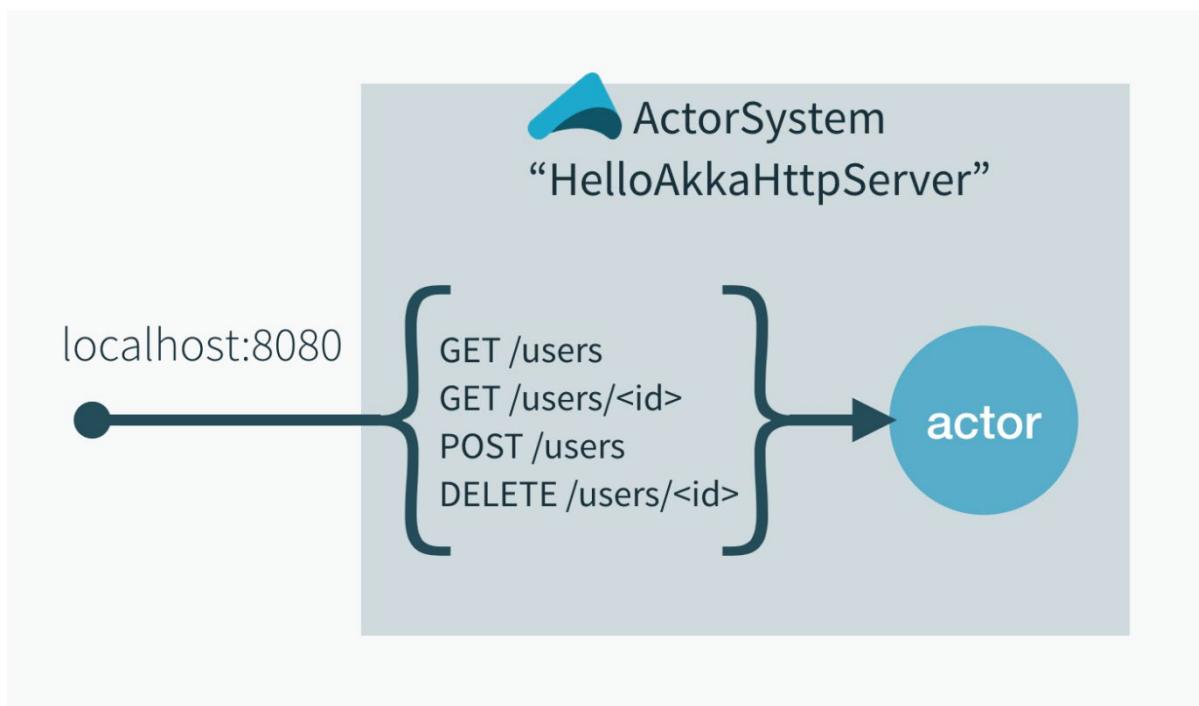
3. เรียกคำสั่ง

```
> sbt.bat
```

4. ภายใต้ sbt prompt เรียกคำสั่ง

```
> reStart
// note เรียก reStop เมื่อต้องการหยุด
```

ตัวอย่างนี้จะแสดง



ใช้ postman เรียกดู **localhost:3000/users**

และ ลอง **POST** ทีละบรรทัดข้อมูลดังนี้

```
{"name": "MrX", "age": 31, "countryOfResidence": "Canada"}  
  

{"name": "Anonymous", "age": 55, "countryOfResidence": "Iceland"}  
  

{"name": "Bill", "age": 67, "countryOfResidence": "USA"}
```

เรียก **GET** โดยใช้ **name** เป็น **id**

The screenshot shows the Postman interface. At the top, there's a header bar with "HTTP" and "localhost:3000/users/Bill". Below it is a search bar with "GET" selected and the URL "localhost:3000/users/Bill". To the right is a "Send" button. Underneath, there are tabs for "Params", "Authorization", "Headers (8)", "Body" (which is currently active), "Pre-request Script", "Tests", and "Settings". The "Body" tab has a dropdown menu showing "none", "form-data", "x-www-form-urlencoded", "raw" (which is selected), and "binary". Below that is a "JSON" dropdown with "Beautify" checked. The main body area contains the following JSON:

```
1 {"name": "Bill", "age": 67, "countryOfResidence": "USA"}
```

At the bottom, there are tabs for "Body", "Cookies", "Headers (4)", and "Test Results". The "Body" tab is active. On the right, there are status indicators: "200 OK", "7 ms", "185 B", and a "Save Response" button. Below the main body area, there are buttons for "Pretty", "Raw", "Preview", "Visualize", and "JSON".

พิจารณาตัวอย่าง code ในไฟล์ akka-example.zip

นักศึกษา สร้างโปรเจคเพื่อรับรัน API ตามที่ได้ออกแบบไว้ใน Task 1 ในครบทั้ว

เกณฑ์ประเมินสำหรับ Task 2

	>80%	60 – 40 %	20 - 40%	< 20%
Select the appropriate code to solve the problems	สามารถเลือกใช้การเขียนโปรแกรมเชิงฟังก์ชันตามโจทย์ที่กำหนดได้ถูกต้อง และสามารถอธิบายความหมายสมดุลขัดเจน	สามารถเลือกใช้การเขียนโปรแกรมเชิงฟังก์ชันตามโจทย์ที่กำหนดได้	สามารถเลือกใช้การเขียนโปรแกรมเชิงฟังก์ชันตามโจทย์ที่กำหนดได้บางส่วน ต้องการคำแนะนำเพื่อการปรับปรุงแก้ไข	ไม่สามารถเลือกใช้เครื่องมือได้
ได้รับคะแนน	5	3-4	1-2	0

วางแผนตัวอย่างผลลัพธ์ ด้วย postman ที่นี่

The screenshot shows the Postman application interface. The URL bar at the top says "POST localhost:3000/days". The main window shows a POST request to "localhost:3000/days". The "Body" tab is selected, showing a JSON payload:

```

1
2   "id": 1,
3   "day": "Monday",
4   "colors": "Red",
5   "description": "Unlucky"
6

```

Below the body, the "Headers" tab shows several headers. In the bottom right corner, the response status is shown as "201 Created 490 ms 180 B". The "Body" tab at the bottom shows the response JSON:

```

1
2   "description": "data added successfully"
3

```

The screenshot shows two separate API requests made using the Postman application.

Request 1: GET /days

- Method:** GET
- URL:** localhost:3000/days
- Body:** JSON (Pretty)
- ```
1 {
2 "id": 1,
3 "day": "Monday",
4 "colors": "Red",
5 "description": "Unlucky😱😱😱😱😱😱"
```
- Response:** 200 OK | 19 ms | 227 B

**Request 2: GET /days/1**

- Method:** GET
- URL:** localhost:3000/days/1
- Body:** JSON (Pretty)
- ```
1 {  
2   "id": 1,  
3   "day": "Monday",  
4   "colors": "Red",  
5   "description": "Unlucky😱😱😱😱😱😱"
```
- Response:** 200 OK | 28 ms | 216 B

The screenshot shows two requests made using the Postman application.

DELETE Request:

- Method: **DELETE**
- URL: **localhost:3000/days/1**
- Body (JSON):

```
1 "id": 1,
2 "day": "Monday",
3 "colors": "Red",
4 "description": "Unlucky\ud83d\udcbb\ud83d\udcbb\ud83d\udcbb\ud83d\udcbb"
```

PUT Request:

- Method: **PUT**
- URL: **localhost:3000/days/1**
- Body (JSON):

```
1 "id": 1,
2 "day": "Monday",
3 "colors": "000000Range",
4 "description": "Unlucky\ud83d\udcbb\ud83d\udcbb\ud83d\udcbb\ud83d\udcbb"
```

Both requests resulted in a **200 OK** response with a response time of **10 ms** and a response size of **175 B**.