

สยามชัมนาณ์คิท
SIAM CHAMNAEN KIT



Shu Ha

Ri



SEAL



We Love
Bug

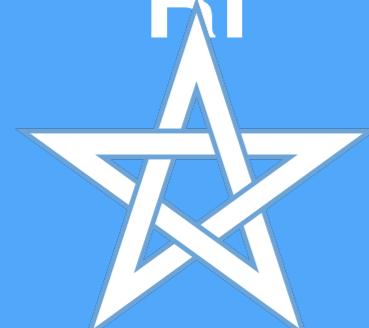
API TEST WITH
POSTMAN





Shu Ha

Ri



SEAL



We Love

Bug



Agenda

Day 1

- What is testing level
- Why we should test API
- What is API Documents
- Working with API request
- Working with Collection

Day 2

- Working with Test
- Data file (CSV, JSON)

Day 3

- Environment
- Newman
- Script Test

Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

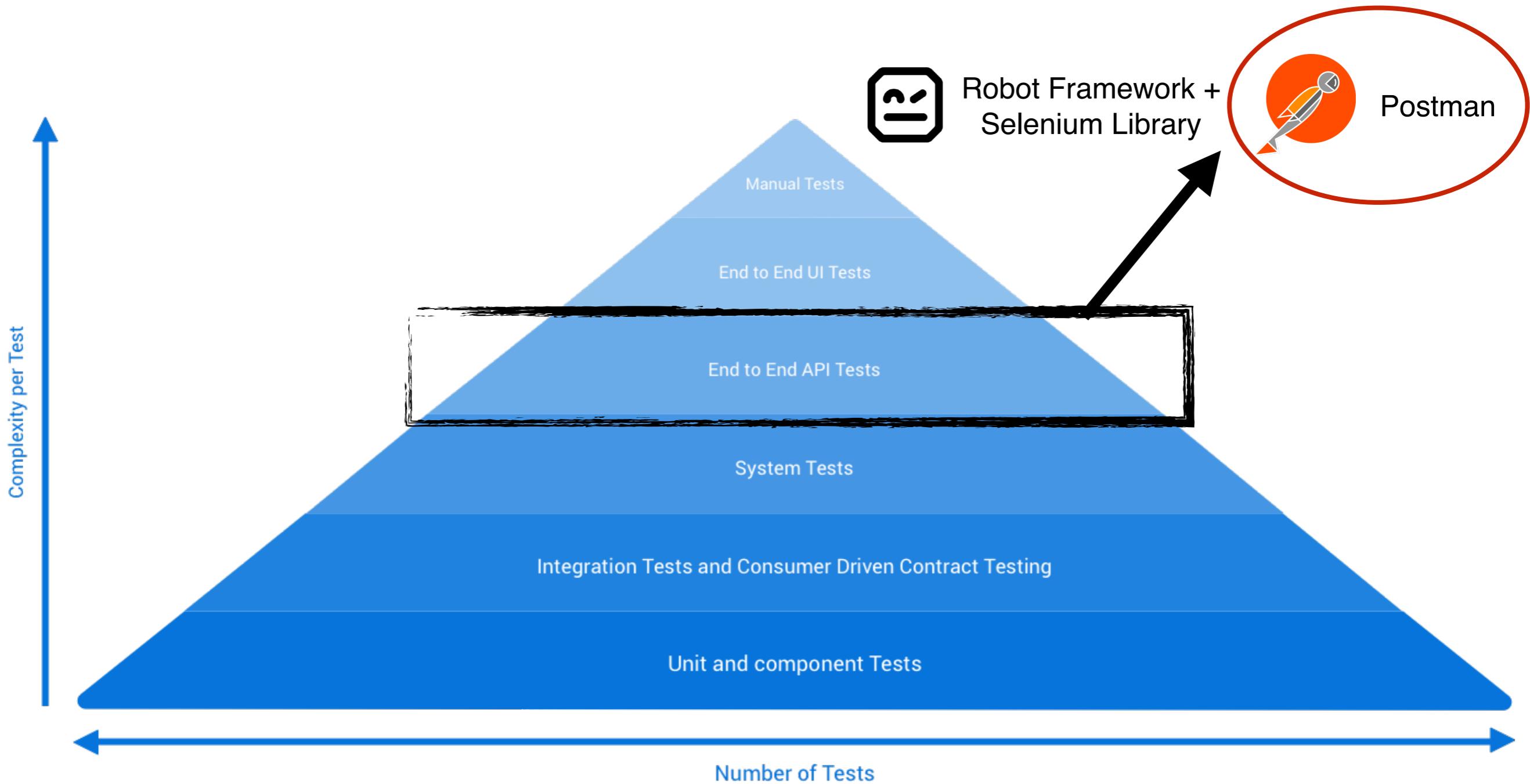
This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#).

Day 1



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#).

What is testing level



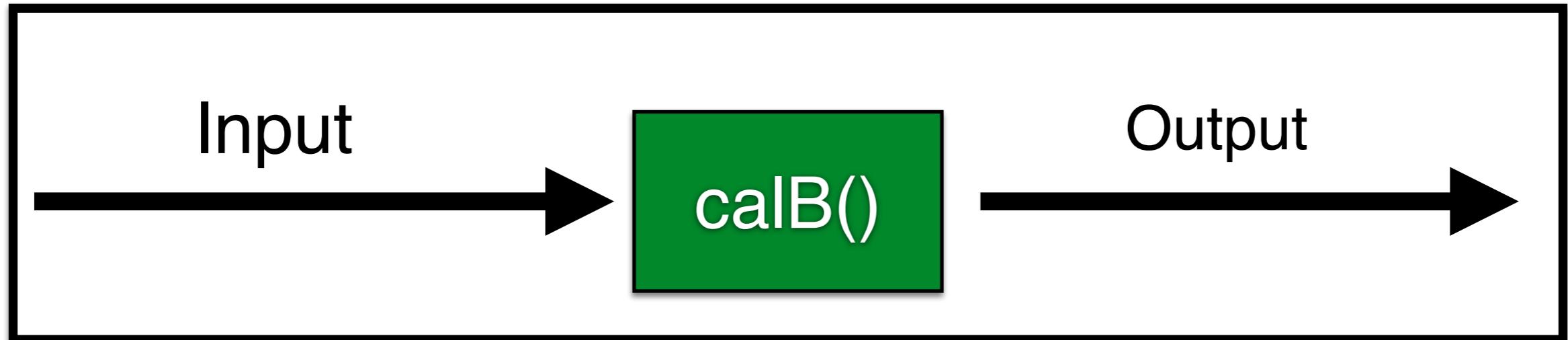
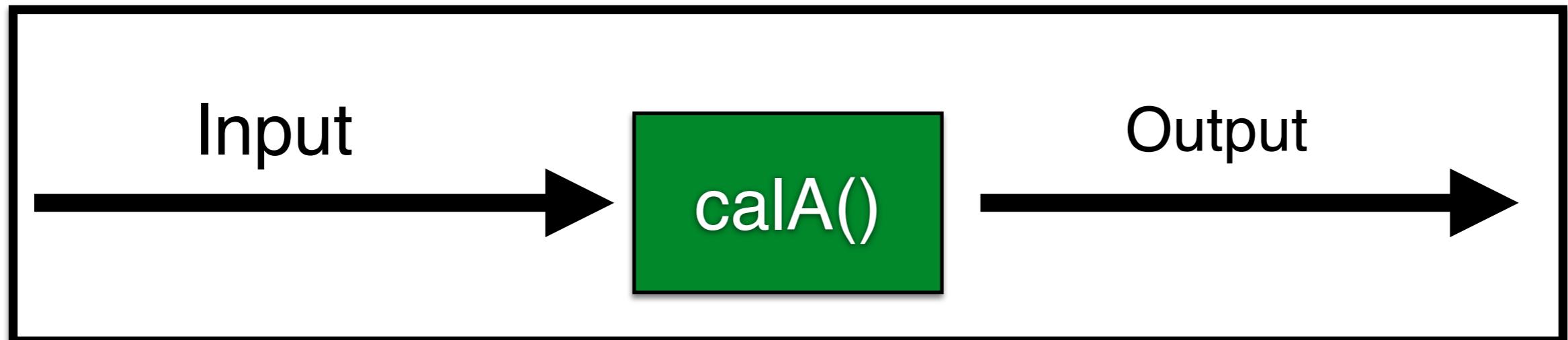
What is testing level

UI	 Robot Framework	 Cypress
API	 Robot Framework	 Postman
Function / Method	 Java	 Jest  Golang
Database	 Java	 Jest  Golang
Third Party	 Stubby4j	 Mountebank  Golang



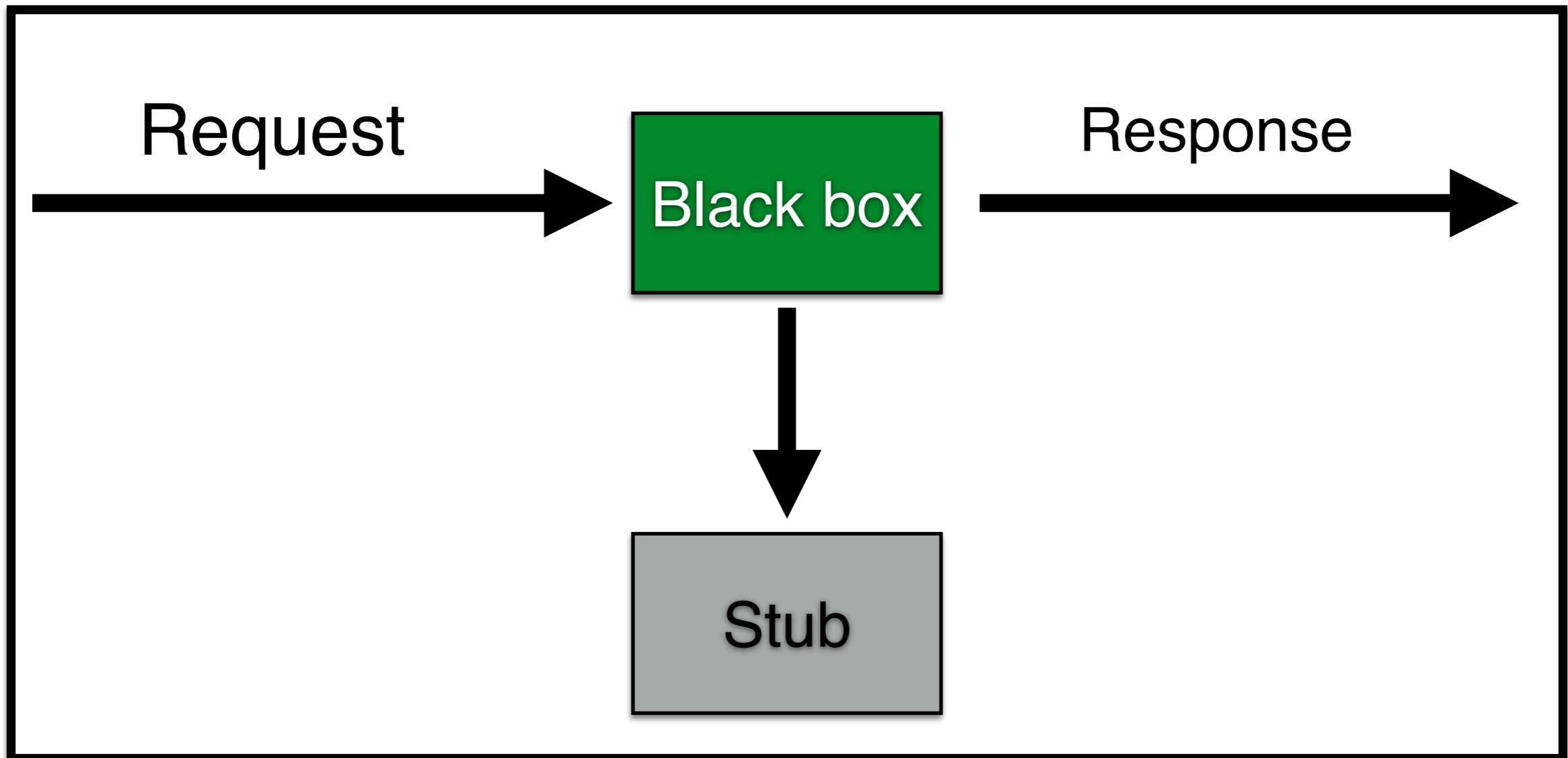
What is testing level

Unit Test = Functions, Methods



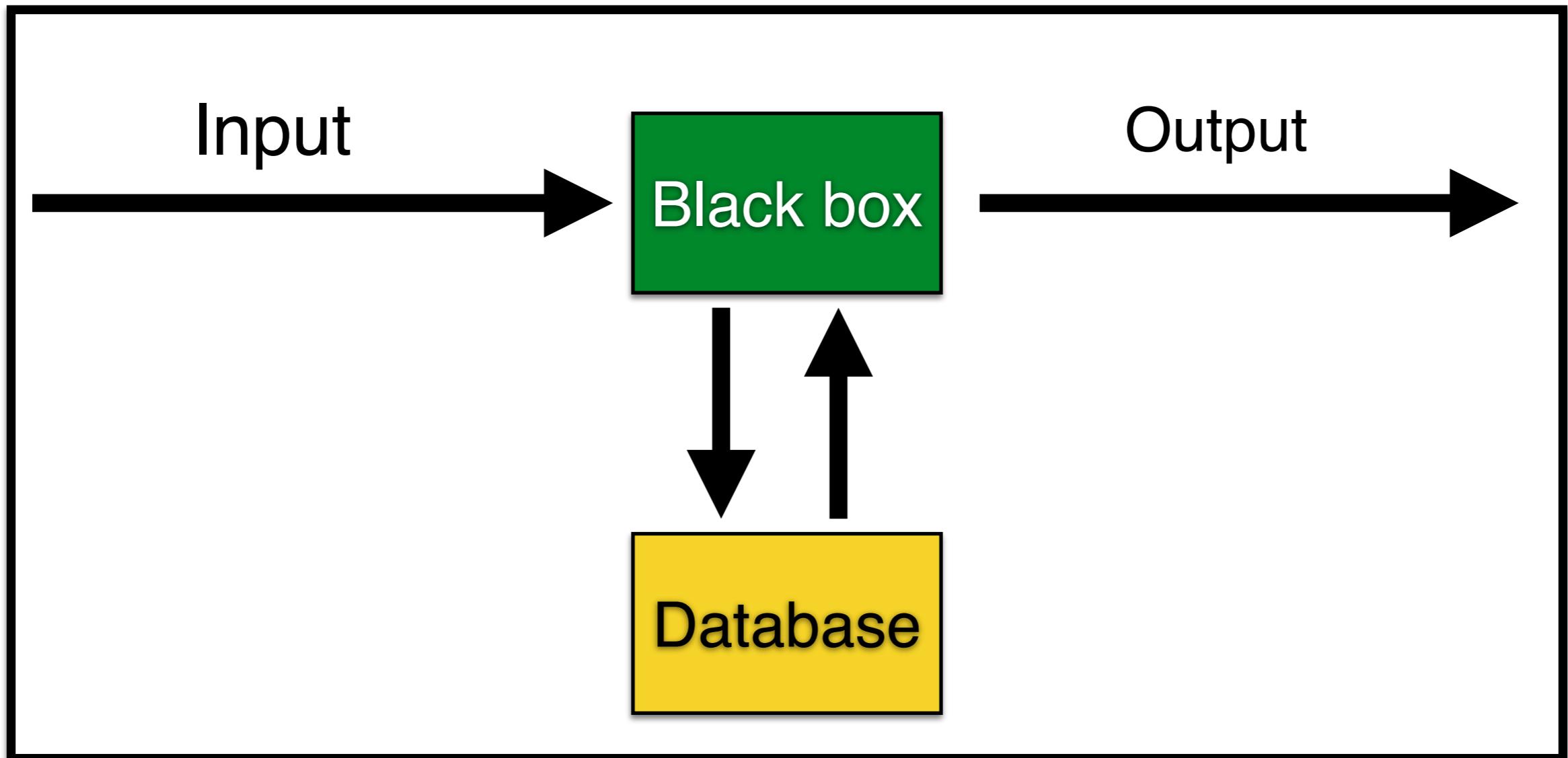
What is testing level

Component Test = Functions + System + API



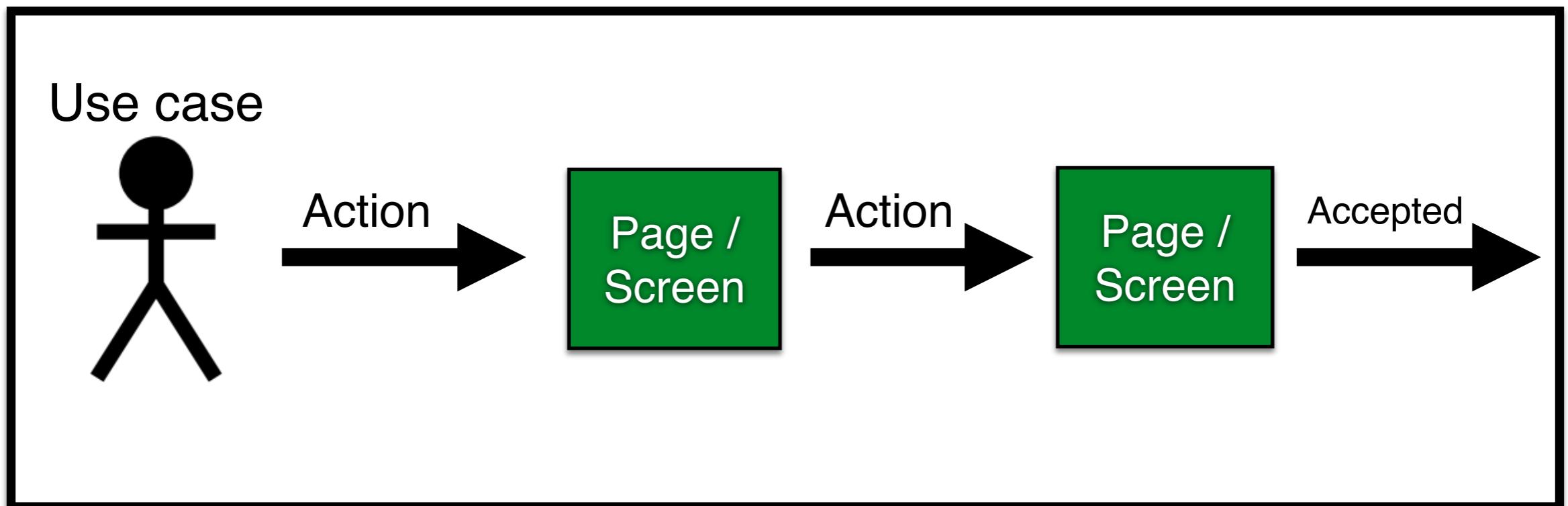
What is testing level

Integration Test = Functions, Classes, Modules



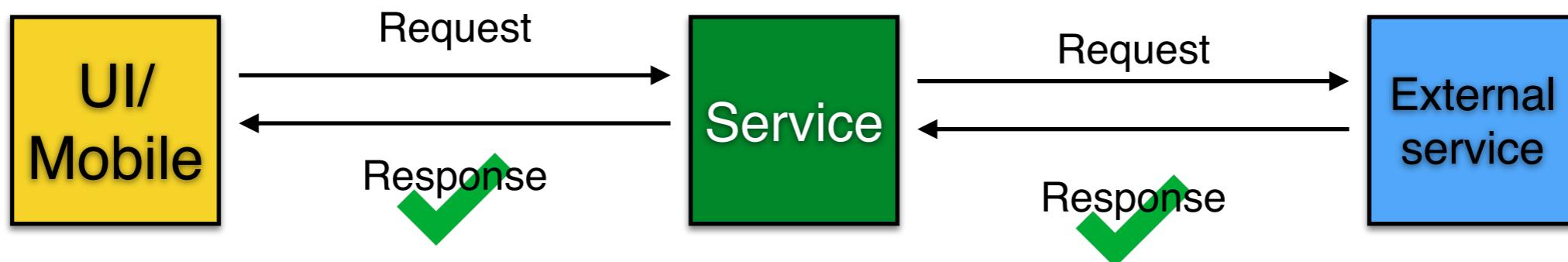
What is testing level

End-to-End Test / Functional = Simulate User Interaction

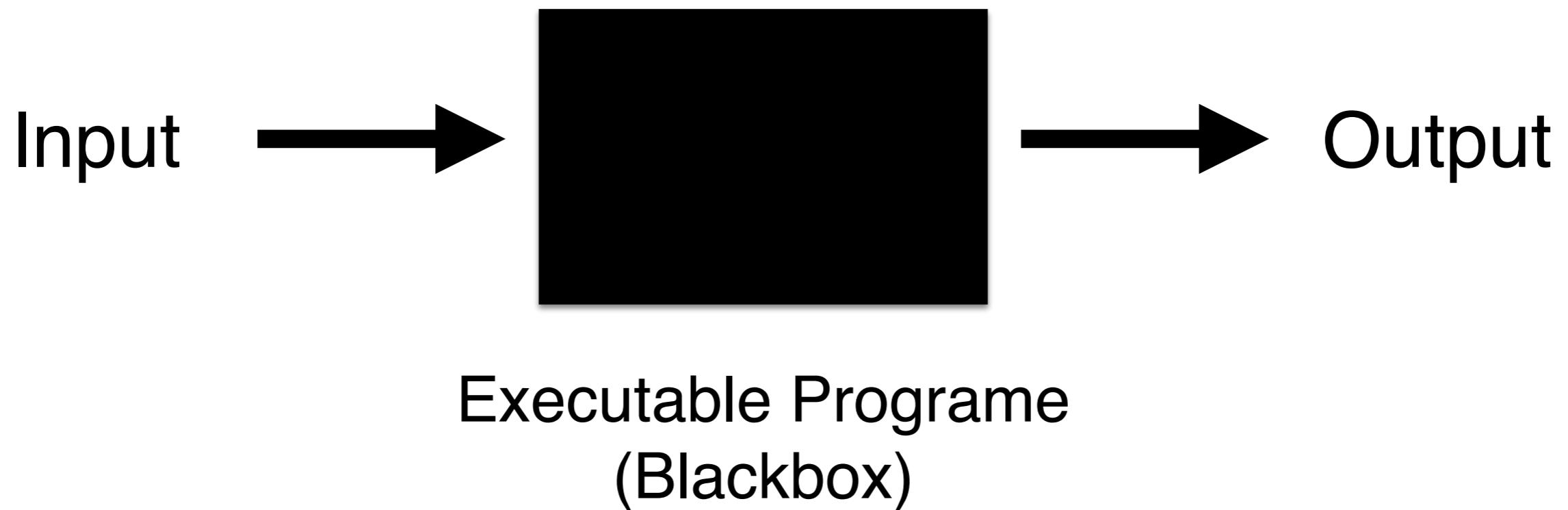


Why we should test API

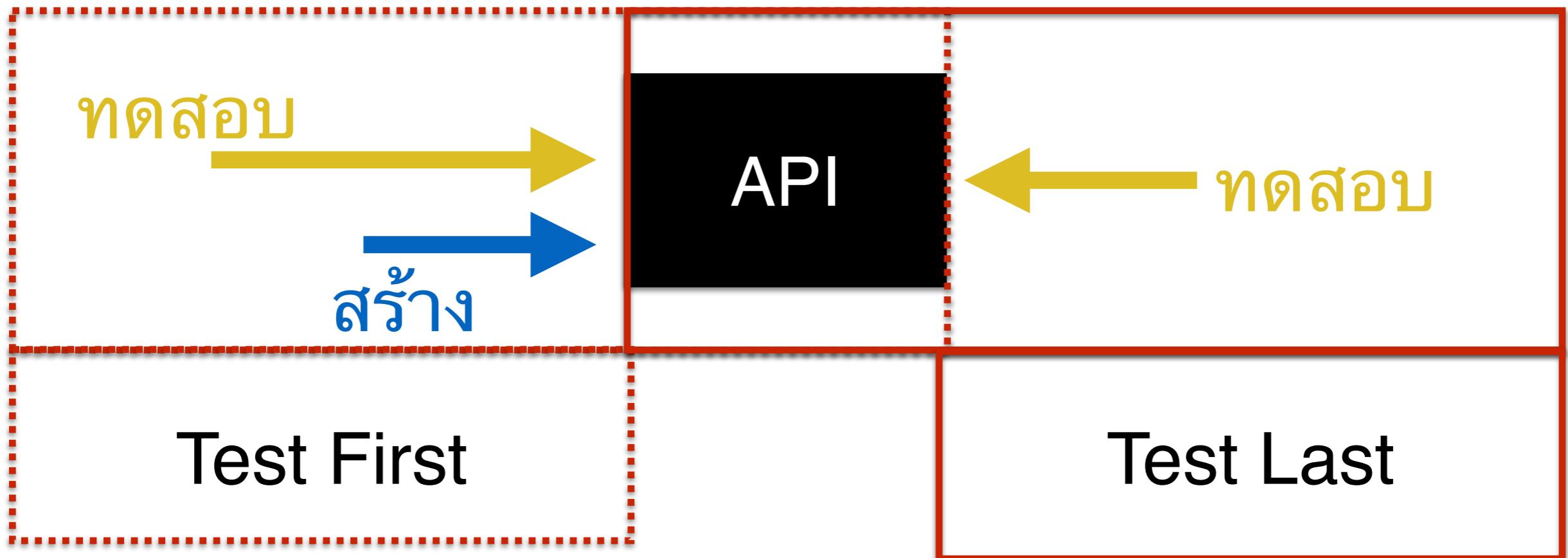
Application programming interface(API) example RESTs, SOAP



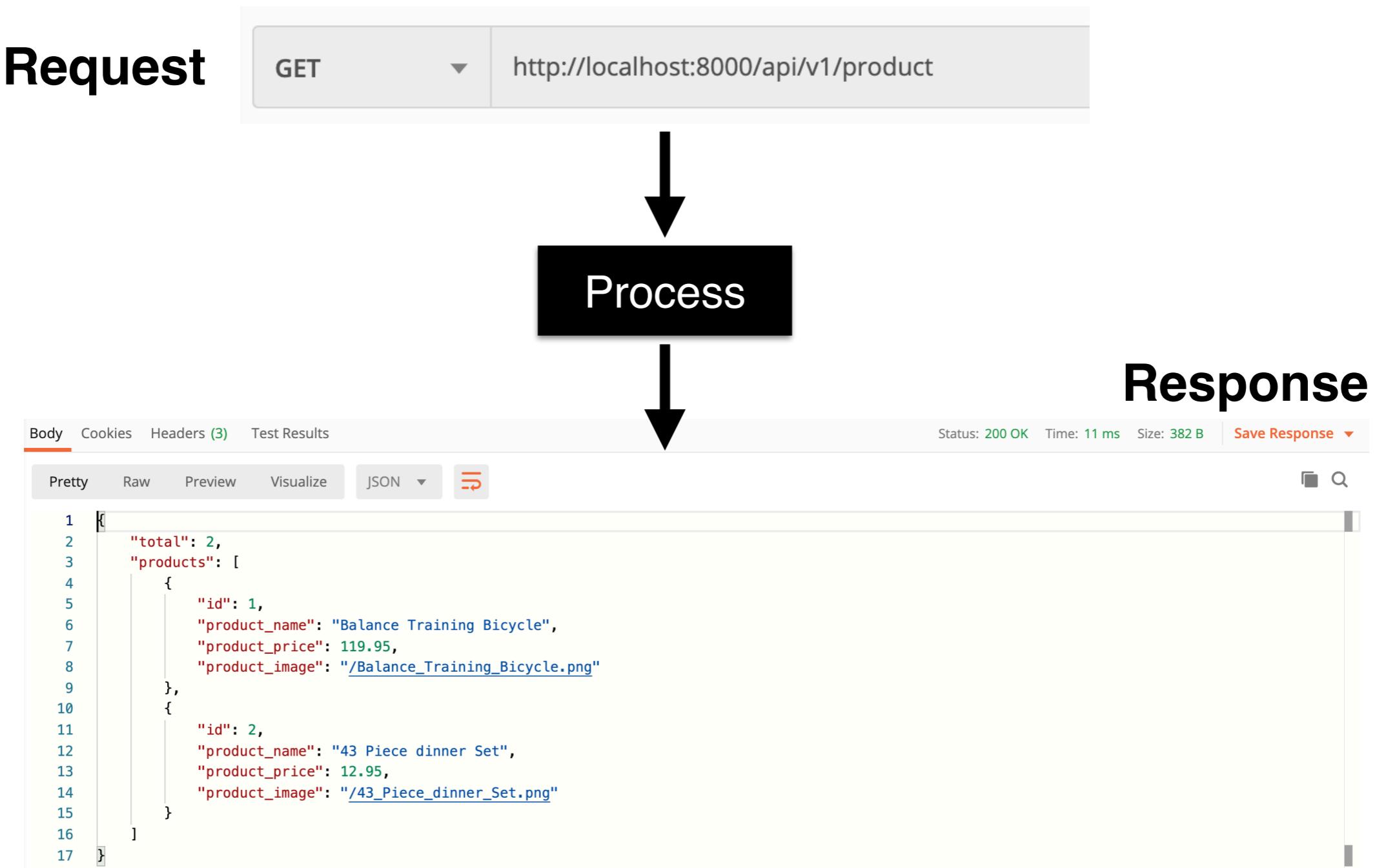
Why API testing is important



API : Test First vs Test Last



API Testing Methodologies



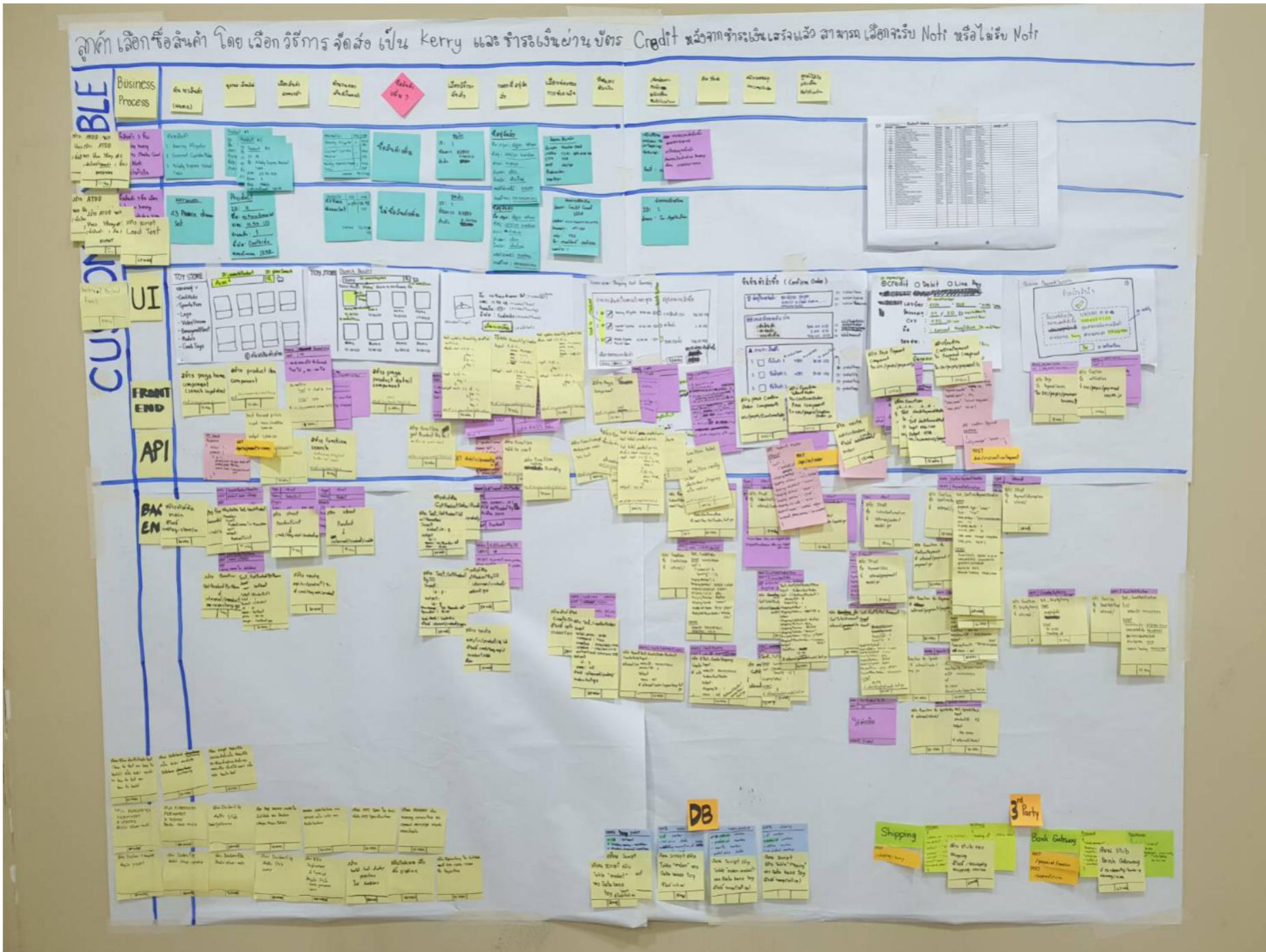
Before write test API

Acceptance test
Data test
API Specification
HTTP Methods
HTTP Status code



Acceptance Test

Story #1 : ลูกค้าเลือกซื้อสินค้าโดยเลือกวิธีการจัดส่งเป็น Kerry และชำระเงินผ่านบัตร Credit หลังจากชำระเงินเสร็จแล้ว สามารถเลือกจะรับการแจ้งเตือนหรือไม่รับแจ้งเตือน



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data test

ชื่อสินค้า	ราคา	จำนวน	ที่อยู่	รูปแบบการชำระเงิน	ประเภทการขนส่ง	ค่าส่ง	เลขบัตรเครดิต
43 Piece dinner set	12.95	1	405/37 ถ.มหิดล ต.ท่าศาลา อ.เมือง จ.เชียงใหม่ 50000	Credit Visa	Kerry	2	47197005 91590995

เลข CVV	วันหมดอายุบัตร (MM/YY)	ชื่อ-นามสกุล	ราคารวมทั้งหมด	ข้อความการแจ้งเตือน
752	02/20	Karnawat Wongudom	14.95	วันเวลาที่ชำระเงิน 1/3/2020 13:30:00 หมายเลขคำสั่งซื้อ 8004359103 คุณสามารถติดตามสินค้าผ่านช่องทาง Kerry ด้วยหมายเลข 1785261900



API Specification

Home

Nareenart edited this page yesterday · 4 revisions

API Specification

1. [API Search](#)
2. [API Product detail](#)
3. [API Submit Order](#)
4. [API Confirm Payment](#)

Database and Data Dictionary

- Database name : [toy](#)
- Table name : [products](#)
- Table name : [order_product](#)
- Table name : [shipping](#)
- Table name : [orders](#)

s



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

HTTP Methods

C -> Create -> Post
R -> Read -> Get
U -> Update -> Put
D -> Delete -> Delete



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

HTTP Methods

Using HTTP Methods for RESTful Services

[Quick-Tips](#)[Resource Naming](#)

The HTTP verbs comprise a major portion of our “uniform interface” constraint and provide us the action counterpart to the noun-based resource. The primary or most-commonly-used HTTP verbs (or methods, as they are properly called) are POST, GET, PUT, PATCH, and DELETE. These correspond to create, read, update, and delete (or CRUD) operations, respectively. There are a number of other verbs, too, but are utilized less frequently. Of those less-frequent methods, OPTIONS and HEAD are used more often than others.

Below is a table summarizing recommended return values of the primary HTTP methods in combination with the resource URIs:

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace	405 (Method Not Allowed), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update/Modify	405 (Method Not Allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	405 (Method Not Allowed), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

Below is a more-detailed discussion of the main HTTP methods. Click on a tab for more information about the desired HTTP method.

ที่มา : <https://www.restapitutorial.com/lessons/httpmethods.html>



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

HTTP Status code

200 - The request is OK

201 - The request is complete

400 - Bad request

401 - You're not getting in

403 - Access is forbidden to the requested page

404 - Can't find it

500 - The server met an unexpected condition

502 - The server received an invalid response from
the upstream server.

ที่มา : https://www.tutorialspoint.com/http/http_status_codes.htm



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

What is Postman?



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Install Postman

<https://www.postman.com/downloads/>

POSTMAN Product ▾ How Collaboration Works Use Cases ▾ Pricing Enterprise Explore Learning Center Dashboard

Get Postman for Mac

Join 10 million developers and download the **ONLY** complete API Development Environment.

[Apple Download](#)

Version 7.19.1 | [RELEASE NOTES](#) | [PRODUCT ROADMAP](#)

Not your OS? Download for Windows ([x32](#) / [x64](#)) or Linux ([x64](#))

This website uses cookies to ensure you get the best experience on our website. [Learn more](#)

[Got it!](#)

Postman

My Workspace Invite Sign In

No Environment Comments (0)

Send Save

Untitled Request

GET postman-echo.com/get

Params Auth Headers Body Pre-req. Tests Settings Cookies Code Response

none

form-data

Request does not have a body

Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



หน้าหลัก Postman

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'New', 'Import', 'Runner', and other icons. Below it is a search bar labeled 'Filter' and tabs for 'History', 'Collections', and 'APIs'. A 'Save Responses' toggle is also present. The main area is titled 'Launchpad' and features a 'Good afternoon, Karnawat ss!' greeting. It includes sections for 'Start something new' (with 'Create a request', 'Create a collection', 'Create an environment', 'Create an API', and 'View More' options), 'Recent workspaces' (listing 'ArtisanXSCK' and 'Team Workspace'), and 'Customize' (with 'Dark mode', 'Enable Launchpad', and 'More settings' options). To the right, there are three large boxes: 'Work smarter with Postman' (listing 'Designing and mocking APIs', 'Debugging and manual testing', 'Automated testing', 'API documentation', 'Monitoring', and 'Collaboration' with their respective icons and lesson counts), 'What's new with Postman' (listing 'What's New in Postman v7.24?' with a link to 'Get v7.24 Now', 'Introducing Improved Filters for Your Postman Monitor Results' with a link to 'Learn More', and tabs for 'Resources' and 'Events'), and a bottom row of buttons for 'Bootcamp', 'Build', 'Browse', and help icons. A small figure icon is located at the bottom right.



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with API request

1. กดเครื่องหมายลักษณ์ +

2. เลือก Method = GET

3. พิมพ์ URL = https://api.umd.io/v1/courses

4. กด Send

5. ดู Response

The screenshot shows the Postman interface with the following details:

- Request URL:** https://api.umd.io/v1/courses
- Method:** GET
- Response Status:** 200 OK
- Response Time:** 1266 ms
- Response Size:** 21.05 KB
- Response Content (Pretty JSON):**

```
1 [  
2 {  
3   "course_id": "AASP100",  
4   "semester": "202008",  
5   "name": "Introduction to African American Studies",  
6   "dept_id": "AASP",  
7   "department": "African American Studies",  
8   "credits": "3",  
9   "description": "Significant aspects of the history of African Americans with particular emphasis on the evolution and development of black communities from slavery to the present.",  
10  "grading_method": [  
11    "Regular",  
12    "Pass-Fail",  
13    "Audit"  
14  ],  
15  "gen_ed": [  
16    [  
17      "DSHS",  
18      "DVUP"  
19    ]  
20  ],  
21  "core": [  
22    "SH",  
23    "D"  
24  ],  
25  "relationships": {  
26    "  
27    "
```



Working with API request

Filter

History Collections APIs

Save Responses Clear all

▼ Today

GET https://api.umd.io/v1/courses

GET https://api.umd.io/v1/courses

GET https://api.umd.io/v1/courses

GET https://api.umd.io/v1/courses

GET https://api.umd.io/v1/bus/route
S

GET https://api.umd.io/v1/bus

หลังกด Send มาเลือก History จะเห็นว่ามีประวัติการเรียก Request



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with API request

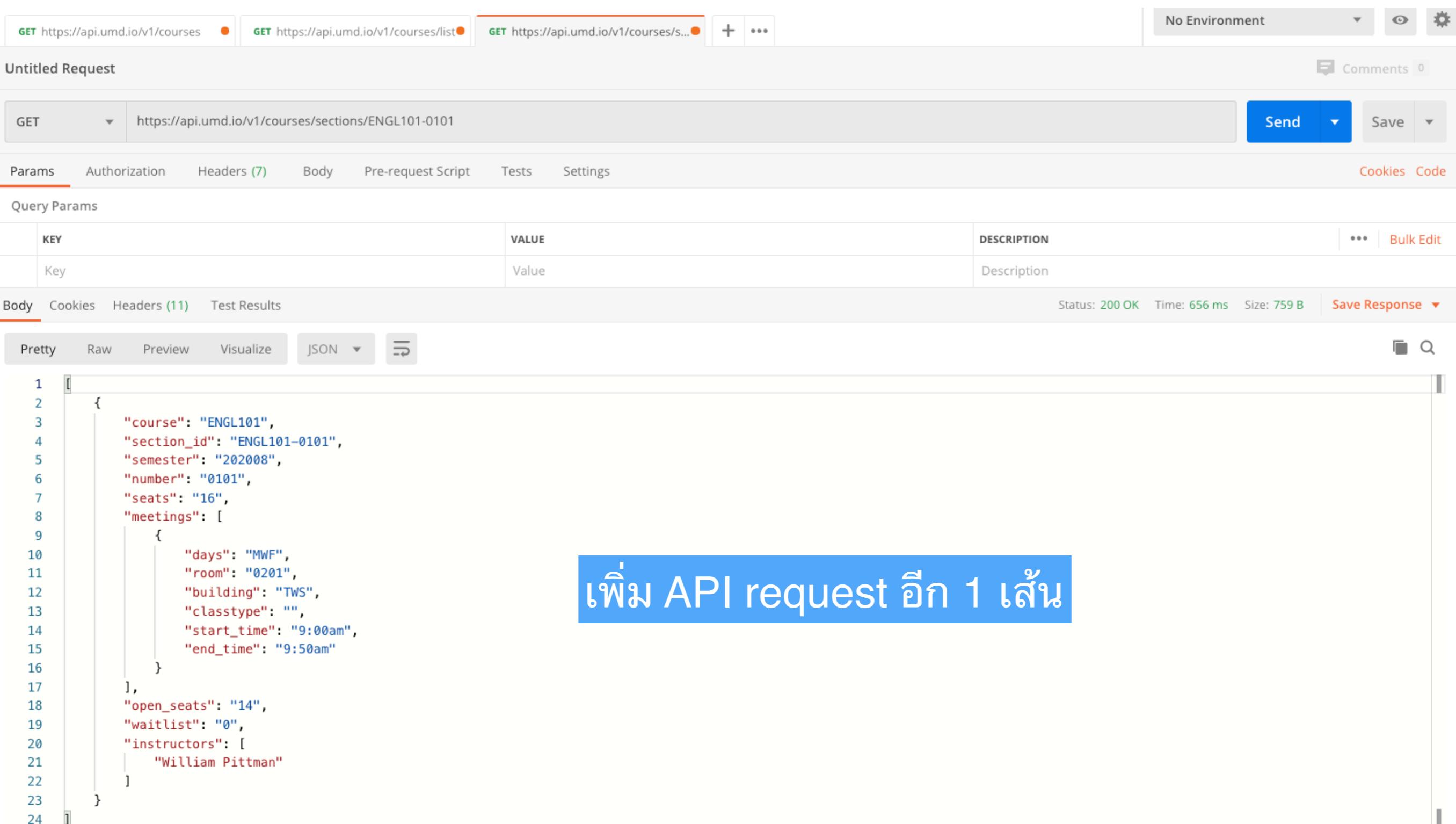
The screenshot shows the Postman interface with a successful API request. The request URL is `https://api.umd.io/v1/courses/list`. The response status is 200 OK, time is 3.99 s, and size is 324.6 KB. The response body is a JSON array of course objects:

```
1
2 {
3     "course_id": "AASP100",
4     "name": "Introduction to African American Studies"
5 },
6 {
7     "course_id": "AASP100H",
8     "name": "Introduction to African American Studies"
9 },
10 {
11     "course_id": "AASP101",
12     "name": "Public Policy and the Black Community"
13 },
14 {
15     "course_id": "AASP200",
16     "name": "African Civilization"
17 },
18 {
19     "course_id": "AASP202",
20     "name": "Black Culture in the United States"
21 },
```

A blue overlay box with white text "เพิ่ม API request อีก 1 เส้น" is overlaid on the JSON response.



Working with API request



The screenshot shows the Postman interface with an 'Untitled Request' tab. The request URL is `https://api.umd.io/v1/courses/sections/ENGL101-0101`. The response status is 200 OK, time 656 ms, size 759 B. The response body is a JSON object:

```
1 [  
2 {  
3   "course": "ENGL101",  
4   "section_id": "ENGL101-0101",  
5   "semester": "202008",  
6   "number": "0101",  
7   "seats": "16",  
8   "meetings": [  
9     {  
10       "days": "MWF",  
11       "room": "0201",  
12       "building": "TWS",  
13       "classtype": "",  
14       "start_time": "9:00am",  
15       "end_time": "9:50am"  
16     }  
17   ],  
18   "open_seats": "14",  
19   "waitlist": "0",  
20   "instructors": [  
21     "William Pittman"  
22   ]  
23 }  
24 ]
```

A blue overlay box with white text 'เพิ่ม API request อีก 1 เส้น' is overlaid on the JSON response.



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with API request

Untitled Request Comments 0

GET https://api.umd.io/v1/courses/MATH140 Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (11) Test Results Status: 200 OK Time: 1141 ms Size: 2.22 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2 {
3     "course_id": "MATH140",
4     "semester": "202008",
5     "name": "Calculus I",
6     "dept_id": "MATH",
7     "department": "Mathematics",
8     "credits": "4",
9     "description": "Introduction to calculus, including functions, limits, continuity, derivatives and applications of the derivative, sketching of graphs of functions, definite and indefinite integrals, and applications of integration. Prerequisite: MATH 100 or equivalent.", "grading_method": [
10         "Regular",
11         "Pass-Fail",
12         "Audit"
13     ],
14     "gen_ed": [
15         [
16             "FSAR",
17             "FSMA"
18         ]
19     ],
20     "core": [
21         "MS"
22     ],
23 ]
```

เพิ่ม API request อีก 1 เส้น



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with API request

The image shows a screenshot of a Postman API testing interface. At the top, there are four tabs showing different API requests. Below the tabs, the main area is titled "Untitled Request". The request method is set to "GET" and the URL is "https://api.umd.io/v1/courses/MATH140". On the right side, there are "Send" and "Save" buttons. Below the URL, there are tabs for "Params", "Authorization", "Headers (7)", "Body", "Pre-request Script", "Tests", and "Settings". The "Params" tab is selected, showing a table for "Query Params" with columns "KEY" and "VALUE". There is one row with "Key" and "Value" both empty. To the right of the table are "Bulk Edit" and other options. The "Body" tab is selected, showing a JSON response. The response is a large object with many fields, including course_id, semester, name, dept_id, department, credits, description, grading_method, gen_ed, and core. The "description" field contains a detailed description of calculus. The "grading_method" field lists Regular, Pass-Fail, and Audit. The "gen_ed" field lists FSAR and FSMA. The "core" field lists MS. The JSON is displayed in a pretty-printed format with line numbers on the left. At the bottom right of the interface, there are status metrics: Status: 200 OK, Time: 1141 ms, Size: 2.22 KB, and a "Save Response" button.



Working with API request



The screenshot shows a REST client interface with a search bar labeled 'Filter' and three tabs: 'History' (selected), 'Collections', and 'APIs'. A 'Save Responses' toggle is on, and a 'Clear all' button is visible. The 'History' section shows a list of requests from today:

- POST http://localhost:8000/api/v1/confirmPayment
- POST http://localhost:8000/api/v1/confirmPayment
- POST http://localhost:8000/api/v1/order
- POST http://localhost:8000/api/v1/submitOrder
- POST http://localhost:8000/api/v1/submitOrder
- GET http://localhost:8000/api/v1/product/1
- GET http://localhost:8000/api/v1/product/:id
- GET http://localhost:8000/api/v1/product
- GET http://localhost:8000/api/v1/product
- GET http://localhost:8000/api/v1/product

A large blue diagonal banner across the middle of the screen contains the text 'ถ้าอยากดูผล Response ก่อนหน้า
ใช้ค้นจาก History?' in white.



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Collection

Filter

History Collections APIs Trash

+ New Collection

1. เลือก “+ New Collection”

CREATE A NEW COLLECTION

Name
Collection Name

Description Authorization Pre-request Scripts Tests Variables

This description will show in your collection's documentation, along with the descriptions of its folders and requests.

Make things easier for your teammates with a complete collection description.

Descriptions support [Markdown](#)

2. ตั้งชื่อ Collection

3. ระบุ Description
ว่าทดสอบงานของอะไร (optional)

4. Create

Create



Working with Collection

หลังจากสร้างจะขึ้น Collection ที่สร้างขึ้นมา
ให้กลับมาที่ Request ที่สร้างไว้

This collection is empty. [Add requests](#) to this collection and create folders to organize them

GET https://api.umd.io/v1/courses

GET https://api.umd.io/v1/courses/list

1. กด Save Request

SAVE REQUEST

2. ตั้งชื่อ (Request name)

Request name

https://api.umd.io/v1/courses

Request description (Optional)

Make things easier for your teammates with a complete request description.

Descriptions support [Markdown](#)

3. เลือก Collection ที่สร้างไว้

Search for a collection or folder

◀ Example UMD + Create Folder

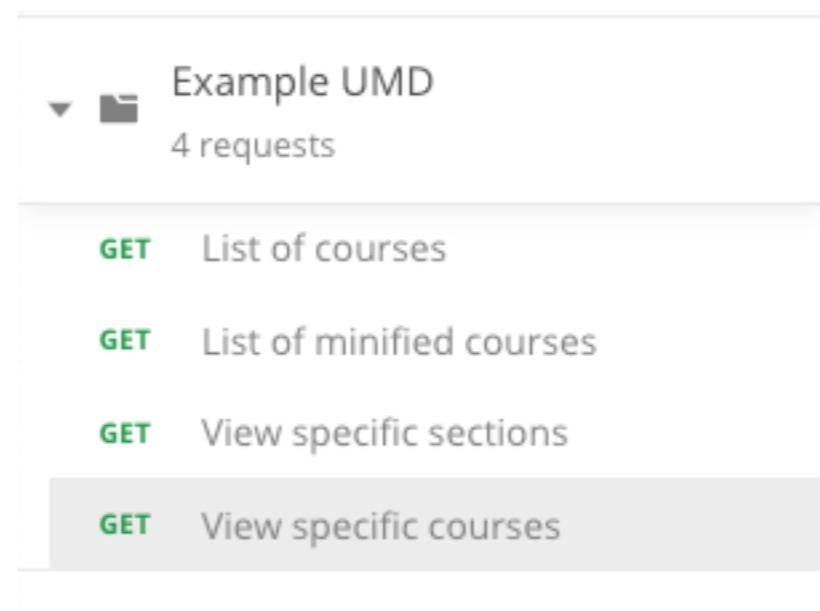
4. Save to Collection ที่สร้างไว้

Cancel Save to Example UMD



Working with Collection

หลังจาก Save to Collection จะมี API ที่สร้างไว้ตอนแรกเพิ่มมาใน Collection ดังรูป



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Collection

Example UMD

API This collection is not linked to any API

Share Run View in web ...

กด Run เพื่อเข้าสู่หน้า Collection Runner

Learn how to document your requests

Add a description

GET List of courses

GET List of minified courses

GET View specific sections

GET View specific courses



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Collection

The screenshot shows the "Collection Runner" application window. At the top, there's a navigation bar with "Collection Runner", "My Workspace", "Run In Command Line", and "Docs". Below the navigation bar, a sidebar on the left contains a search bar and a list of collections under "Example UMD": "List of courses", "List of minified courses", "View specific sections", and "View specific courses". On the right, the main area is titled "RUN ORDER" and lists four items, all of which have checkboxes checked: "GET List of courses", "GET List of minified courses", "GET View specific sections", and "GET View specific courses". At the bottom, there are settings for "Environment" (set to "No Environment"), "Iterations" (set to 1), "Delay" (set to 0 ms), and "Data" (with a "Select File" button). There are also several checkboxes: "Save responses" (checked), "Keep variable values" (checked), "Run collection without using stored cookies" (unchecked), and "Save cookies after collection run" (checked). Two large blue buttons at the bottom are "Run Example UMD" (which has a red border) and "Run".



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Collection

The screenshot shows the Collection Runner interface with the following details:

- Collection Runner** tab is selected.
- Run Results** tab is visible.
- My Workspace** dropdown shows "My Workspace".
- Run In Command Line** and **Docs** buttons are present.
- Example UMD** collection was run **just now**.
- Iteration 1** results:
 - GET List of courses** (https://api.umd.io/v1/co...): Status 200 OK, 1553 ms, 21.552 KB. This request does not have any tests.
 - GET List of minified courses** (https://api.umd.io/v1/co...): Status 200 OK, 2442 ms, 332.395 KB. This request does not have any tests.
 - GET View specific sections** (https://api.umd.io/v1/co...): Status 200 OK, 514 ms, 759 B. This request does not have any tests.
 - GET View specific courses** (https://api.umd.io/v1/co...): Status 200 OK, 533 ms, 2.277 KB. This request does not have any tests.
- A large blue button at the bottom center says **ผลลัพธ์ที่ได้** (Results obtained).



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Day 2



Install Mountebank



```
npm install mountebank -g
```

Requires nodeJS of v6

```
npm install npm -g
```

Start mountebank
mb start --configfile deposit.json



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

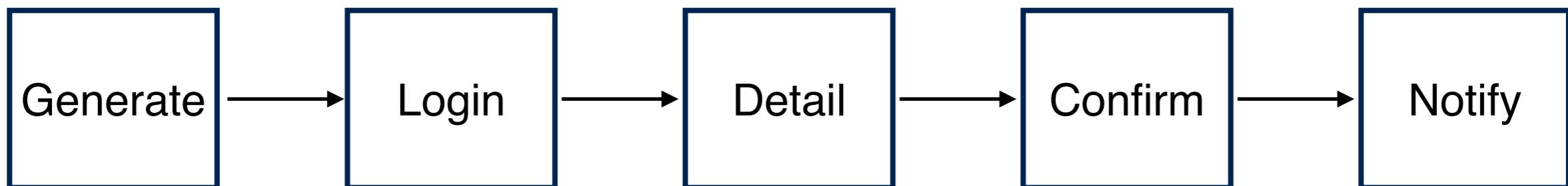
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

ลูกค้ายื่น QR Code พนักงานล็อคอิน และง่ายๆ แล้ว
ให้พนักงาน Scan เพื่อรับ Token สำหรับทำการ
สำหรับทำการ

ยืนยันการ
ทำการ

ได้รับข้อความว่า
“ได้รับเงิน 485 บ.
จากการฝากเงินผ่าน 7-ELEVEN”



API Generate

API Login

API Detail

API Confirm

API Notify

ดูเพิ่มเติม : api-specification-for-deposit.pdf



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

Story	API: Generate Reference Code	
	Method : POST, URL : /api/v1/transaction/generate	
	Request	Response
<p>เลือกไปฝากเงินผ่าน 7-11 จำนวนเงิน 500 บาท ชั่งเลือกอายุ 24 ปี จึงเสียค่าธรรมเนียม 15 บาท ในวันที่ 27 ธันวาคม 2020 เวลา 14.00 น. ด้วยจำนวนเงิน 500 บาท เสียค่าธรรมเนียมให้กับ 7-11 จำนวน 15 บาท เหลือเงินฝากเข้าจริง คือ 485 บาท</p> <p>พนักงาน 7-11 ล็อคอินเข้าสู่ระบบเพื่อรับ Token สำหรับทำรายการฝากเงินให้กับลูกค้า ทำการสแกน QR Code จากโทรศัพท์ลูกค้า สถานะของ QR Code เปลี่ยนเป็น Used และแสดงรายละเอียดข้อมูลของลูกค้า</p> <p>กดยืนยัน สถานะของ QR Code จะต้องเปลี่ยนเป็น Complete ลูกค้าได้รับข้อความแจ้งเตือนว่าทำรายการฝากเงินสำเร็จ</p>	<pre>{ "type": "deposit", "account_number": "2000000012541", "account_name": "Panumars Saento", "user_id": "573020145", "fee": "15.00", "cash_paid": "500.00", "deposit_amount": "485.00", "transaction_datetime": "20201227140030", "reference_code": "M181130101242968173", "transaction_id": "3258001" }</pre>	<pre>// Success (status code: 200) { "status": "ready to scan", "type": "deposit", "account_number": "2000000012541", "account_name": "Panumars Saento", "fee": "15.00", "cash_paid": "500.00", "deposit_amount": "485.00", "reference_code": "M181130101242968173", "expired_datetime": "20201227140219" }</pre>

ดูเพิ่มเติม : Google Sheet ใน MATERIAL/Day2/DATA-TEST-DEPOS



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

GET http://localhost:8000/api/v1/product Send Save

Params Authorization Headers (7) Body Pre-request Script Tests ● Settings Cookies Code

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 };  
4  
5 pm.test("มีสินค้าชื่อ 43 Piece dinner Set", function () {  
6     var jsonData = pm.response.json();  
7     pm.expect(jsonData.products[1].product_name).to.eql("43 Piece dinner Set");  
8 };  
9  
10 pm.test("สินค่าราคา 12.95", function () {  
11     var jsonData = pm.response.json();  
12     pm.expect(jsonData.products[1].product_price).to.eql(12.95);  
13 };  
14  
15 pm.  
16     1 {  
17         "total": 2,  
18     };  
19         "products": [  
20             {  
21                 "id": 1,  
22                 "product_name": "Balance Training Bicycle",  
23                 "product_price": 119.95,  
24                 "product_image": "/Balance_Training_Bicycle.png"  
25             },  
26             {  
27                 "id": 2,  
28                 "product_name": "43 Piece dinner Set",  
29                 "product_price": 12.95,  
30                 "product_image": "/43_Piece_dinner_Set.png"  
31             }  
32         ]  
33     };
```

2. ตั้งชื่อ Test Case

3. Assert ของที่คาดหวัง

1. เลือกว่าจะทดสอบอะไร

Learn more about tests scripts

SNIPPETS

Send a request

Status code: Code is 200

Response body: Contains string

Response body: JSON value check

Response body: Is equal to a string

Response headers: Content-Type header check

Response time is less than 200ms

Status code: Successful POST request



Working with Test

Run Collection Runner สิ่งที่เปลี่ยนไปจาก จะมี Test Case เพิ่มเข้ามา

The screenshot shows the Postman Collection Runner interface with the following details:

- Collection Runner** tab is selected.
- Run Results** tab is visible.
- My Workspace** dropdown shows "Example (Class Postman)".
- Run Summary** button is orange.
- Export Results**, **Retry**, and **New** buttons are also present.
- Iteration 1** results:
 - GET search product**: Status code is 200, 200 OK, 6 ms, 382 B. Contains:
 - Status code is 200
 - มีสินค้าชื่อ 43 Piece dinner Set
 - สินค้าราคา 12.95
 - จะต้องแสดงรูป 43_Piece_dinner_Set
 - GET selected product**: Status code is 200, 200 OK, 10 ms, 276 B. Contains:
 - Status code is 200
 - เลือกสินค้า 43 Piece dinner Set id = 2
 - สินค้าชื่อ 43 Piece dinner Set
 - สินค้ามีราคา 12.95
 - เลือกจำนวนสินค้า 10 ชิ้น
 - อยู่ในหมวดหมู่ของ CoolKidz
 - POST order**: Status code is 200, 200 OK, 26 ms, 168 B. Contains:

This request does not have any tests.

A blue callout box labeled "หลังเขียน Test" points to the first test block under the first request. Another blue callout box labeled "ก่อนเขียน Test" points to the message "This request does not have any tests." under the third request.



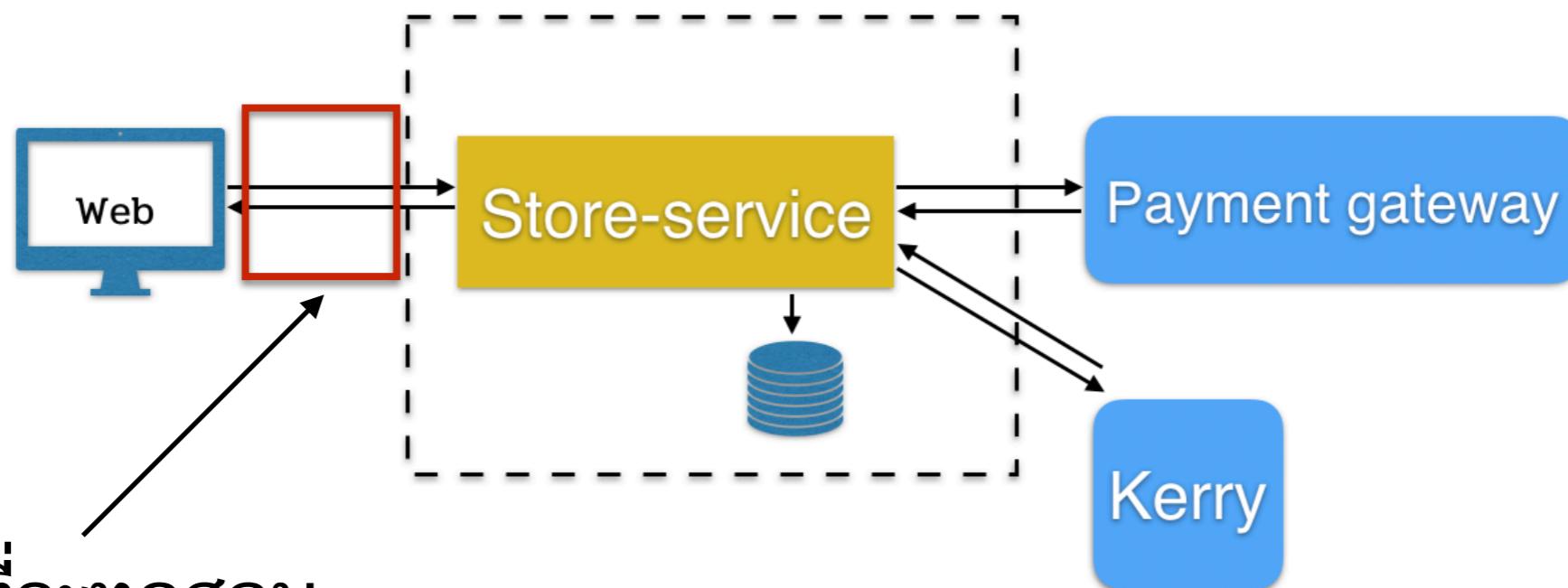
Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

Architecture service



จุดที่จะทดสอบ

Store-service ทำงานถูกต้อง



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

เขียนการทดสอบให้ครบ



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Working with Test

▶ Example (Class Postman)

4 requests

▶ Example2 (Class Postman)

4 requests

▶ Example3 (Class Postman)

4 requests

```
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("เลือกสินค้า 43 Piece dinner Set id = 2", function () {  
6     var jsonData = pm.response.json();  
7     pm.expect(jsonData.id).to.eql(2);  
8 });  
9  
10 pm.test("สินค้าชื่อ 43 Piece dinner Set", function () {  
11    var jsonData = pm.response.json();  
12    pm.expect(jsonData.product_name).to.eql("43 Piece dinner Set");  
13});  
14  
15 pm.test("สินค้ามีราคา 12.95", function () {  
16    var jsonData = pm.response.json();  
17    pm.expect(jsonData.product_price).to.eql(12.95);  
});  
  
1 pm.test("Status code is 200", function () {  
2     pm.response.to.have.status(200);  
3 });  
4  
5 pm.test("เลือกสินค้า 43 Piece dinner Set id = 2", function () {  
6     var jsonData = pm.response.json();  
7     pm.expect(jsonData.id).to.eql(2);  
8 });  
9  
10 pm.test("สินค้าชื่อ 43 Piece dinner Set", function () {  
11    var jsonData = pm.response.json();  
12    pm.expect(jsonData.product_name).to.eql("43 Piece dinner Set");  
13});  
14  
15 pm.test("สินค้ามีราคา 12.95", function () {  
16    var jsonData = pm.response.json();  
17    pm.expect(jsonData.product_price).to.eql(12.95);  
});
```

ถ้ามีการเปลี่ยนแปลง จะต้องแก้ไขทั้ง 3 ที่



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data File

product_id	product_name	product_price	product_brand	product_image
2	43 Piece dinner Set	12.95	CoolKidz	/43_Piece_dinner_Set.png

CSV : Keyword (Column), Iteration Data (Row)



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data File

```
1 {  
2   "cart": [  
3     {  
4       "product_id": "{{product_id}}",  
5       "quantity": "{{quantity}}"  
6     }  
7   ],  
8   "shipping_method": "{{shipping_method}}",  
9   "shipping_address": "405/37 ถ.มหิดล",  
10  "shipping_sub_district": "ต.ท่าศาลา",  
11  "shipping_district": "อ.เมือง",  
12  "shipping_province": "จ.เชียงใหม่",  
13  "shipping_zip_code": "50000",  
14  "recipient_name": "ณัฐญา ชุดบุตร",  
15  "recipient_phone_number": "0970809292"  
16 }
```

การกำหนดชื่อตัวแปรใน postman ให้ใส่ {{ตัวแปร}}



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data File

Params Authorization Headers (7) Body Pre-request Script **Tests** ● Settings

```
1 pm.test("Status code is 200", function () {
2     pm.response.to.have.status(200);
3 );
4
5 pm.test("มีสินค้าชื่อ 43 Piece dinner Set", function () {
6     var jsonData = pm.response.json();
7     pm.expect(jsonData.products[1].product_name).to.eql(data["product_name"]);
8 );
9
10 pm.test("สินคาราคา 12.95", function () {
11     var jsonData = pm.response.json();
12     pm.expect(jsonData.products[1].product_price).to.eql(12.95);
13 );
```

เปลี่ยนจาก Hard code ใน Test เป็นดึงจาก Data File
โดยเขียน data[“variable”]



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data File

Collection Runner My Workspace ▾ Run In Command Line Docs

Choose a collection or folder

Search for a collection or folder

◀ Example (Class Postman)

GET search product
GET selected product
POST order
POST confirm payment

RUN ORDER

Deselect All | Select All | Reset

✓ GET search product
✓ GET selected product
✓ POST order
✓ POST confirm payment

Environment No Environment

Iterations 1

Delay 0 ms

Data Select File example.csv X

Data File Type text/csv ▾ Preview

Save responses ⓘ
 Keep variable values ⓘ
 Run collection without using stored cookies
 Save cookies after collection run ⓘ

Run Example (Clas...)

Import file data ที่สร้างไว้เข้ามา



Data File

ผลของ Collection Runner ยังต้องเหมือนเดิม

The screenshot shows the Collection Runner interface with the following details:

- Collection Runner** tab is selected.
- Run Results** tab is visible.
- My Workspace** dropdown shows "No Environment".
- Run In Command Line** and **Docs** buttons are present.
- Iteration 1** results:
 - GET search product** (Status code is 200):
 - Result: 200 OK (6 ms, 382 B)
 - Details: 43 Piece dinner Set, 12.95, 43_Piece_dinner_Set
 - GET selected product** (Status code is 200):
 - Result: 200 OK (10 ms, 276 B)
 - Details: Selected product id = 2, 43 Piece dinner Set, 12.95, 10 pieces, CoolKidz
 - POST order**: This request does not have any tests.
- Run Summary**, **Export Results**, **Retry**, and **New** buttons are available.



Data File

เปลี่ยนจาก Hard code ให้ใช้ Data File แทน



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Day 3



Data File

ปัญหาที่เจอใน Excel แปลง CSV
ในการนี้ที่ Response / Output เป็น String

Excel

	A	B	C
1	account_number	fee	cash_paid
2	2000000012541	15.00	500.00
3			



CSV

```
data-file > Book1.csv
1 account_number,fee,cash_paid
2 2000000012541,15.00,500.00
```



Data File

ผลลัพธ์ที่ได้

Iteration	account_number	fee	cash_paid
1	2000000012541	15	500

Postman จะอ่านเป็น float โดยจะไม่มีทศนิยม 00 แสดง



POST generate http://localhost:3000/api... deposit_csv แบบ 3 / generate 200 OK 13 ms 473 B

- Status code is 200
- reference code is M181130101242968173
- status is ready to scan
- type is deposit

ผลลัพธ์ที่ได้จะไม่ผ่าน เนื่องจาก response เป็น 15.00 (string) เทียบกับ 15 (float)

- fee is 15.00 | AssertionError: expected '15.00' to deeply equal 15
- cash_paid is 500.00
- deposit_amount is 485.00



Data File

วิธีแก้

แบบที่ 1 แก้ใน file CSV โดยใส่ “-“ เข้าไป

```
data-file > Book2.csv
1 account_number,fee,cash_paid
2 2000000012541,"15.00",500.00
```



Iteration	account_number	fee	cash_paid
1	2000000012541	"15.00"	500

Field ที่ใส่ “-“

Field ที่ไม่ใส่ “-“



Data File

ผลลัพธ์ที่ได้

POST generate http://localhost:3000/api... deposit_csv แบบ 3 / generate

200 OK 37 ms 473 B

- Status code is 200
- reference code is M181130101242968173
- status is ready to scan
- type is deposit
- fee is 15.00
- cash_paid is 500.00
- deposit_amount is 485.00

สามารถเทียบกันได้



Data File

วิธีแก้

แบบที่ 2 แก้ใน test postman โดยใส่ data[“fee”] + “.00” เข้าไป

```
20 pm.test("fee is 15.00", function () {  
21     var jsonData = pm.response.json();  
22     pm.expect(jsonData.fee).to.eql(data["fee"] + ".00");  
23 };  
24
```



Iteration	account_number	fee	cash_paid
1	2000000012541	15	500

จะต่อค่า “.00” มาให้



Data File

ผลลัพธ์ที่ได้

POST generate http://localhost:3000/api... deposit_csv แบบ 3 / generate

200 OK 37 ms 473 B

- Status code is 200
- reference code is M181130101242968173
- status is ready to scan
- type is deposit
- fee is 15.00
- cash_paid is 500.00
- deposit_amount is 485.00

สามารถเทียบกันได้



Data File

วิธีแก้

แบบที่ 2 แก้ใน test postman โดยใส่ data[“fee”] + “.00” เข้าไป

```
20 pm.test("fee is 15.00", function () {  
21     var jsonData = pm.response.json();  
22     var newValue = data["fee"].toFixed(2).toString();  
23     pm.expect(jsonData.fee).to.eql(newValue);  
24 });
```



Iteration	account_number	fee	cash_paid
1	2000000012541	15	500

จะรับค่าใน file data และ fix ให้มีทศนิยม 2 ตำแหน่ง
จากนั้นทำการแปลงให้เป็น type string



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Data File

ผลลัพธ์ที่ได้

POST generate http://localhost:3000/api... deposit_csv แบบ 3 / generate

200 OK 37 ms 473 B

- Status code is 200
- reference code is M181130101242968173
- status is ready to scan
- type is deposit
- fee is 15.00
- cash_paid is 500.00
- deposit_amount is 485.00

สามารถเทียบกันได้



Data File

```
[  
  {  
    "product_id": 2,  
    "product_name": "43 Piece dinner Set",  
    "product_price": 12.95,  
    "product_brand": "CoolKidz",  
    "product_image": "/43_Piece_dinner_Set.png"  
  }  
]
```

JSON : key, value



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



10 Minutes Break



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](#).

Environment



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Environment

1. เลือกที่ฟันเฟือง

The screenshot shows the Postman application interface. At the top, there are buttons for 'New', 'Import', 'Runner', and a dropdown for 'My Workspace'. Below the header, there's a search bar labeled 'Filter' and tabs for 'History', 'Collections' (which is selected), and 'APIs'. In the center, there are two requests: 'GET product detail' and 'POST submit order'. To the right, it says 'No Environment' with a dropdown arrow. A red box highlights the gear icon in the top right corner of the main workspace area. A modal window titled 'MANAGE ENVIRONMENTS' is open in the center. It contains a brief description: 'An environment is a set of variables that allow you to switch the context of your requests. Environments can be shared between multiple workspaces.' followed by a link 'Learn more about environments'. Below this, three environments are listed: 'local', 'token_thaipostData', and 'UAT', each with 'Share', 'Download', and 'More' buttons. At the bottom of the modal, there are buttons for 'Globals', 'Import', and a large orange 'Add' button, which is also highlighted with a red box.

2. กด Add



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Environment

MANAGE ENVIRONMENTS X

1. Name environment

Environment Name
local

2. ชื่อตัวแปร และ endpoint ที่ใช้ยิง request

VARIABLE	INITIAL VALUE i	CURRENT VALUE i
<input checked="" type="checkbox"/> url	http://localhost:8000	http://localhost:8000

Add a new variable ... | Persist All | Reset All

3. Update

Cancel Update

Tip: Use variables to reuse values in different places. Work with the current value of a variable to prevent sharing sensitive values with your team. [Learn more about variable values](#) X



Environment

1. เลือกจาก No Environment -> local ที่สร้างไว้

GET search product

search product

GET {{url}}/api/v1/product

Send Save

Params Authorization Headers (7) Body Pre-request Script Tests ● Settings Cookies Code

2. เปลี่ยน URL จาก <http://localhost:8000> เป็น {{url}}

```
1 <v>{
2   "total": 2,
3   "products": [
4     {
5       "id": 1,
6       "product_name": "Balance Training Bicycle",
7       "product_price": 119.95,
8       "product_image": "/Balance_Training_Bicycle.png"
9     },
10    {
11      "id": 2,
12      "product_name": "43 Piece dinner Set",
13      "product_price": 12.95,
14      "product_image": "/43_Piece_dinner_Set.png"
15    }
16  ]
17 }
```

3. กด Send จะต้องได้ Response เหมือนเดิม



Environment

The screenshot shows the Collection Runner interface in a web browser. On the left, there's a sidebar with tabs for History, Collections (which is selected), and APIs. A red box highlights the three-dot menu icon next to the 'shopping cart success' collection. The main area displays a collection named 'shopping cart success' with a single request 'GET product detail'. To the right, a 'RUN ORDER' section shows a checked item 'GET product detail'. A large blue banner in the center says '1. เลือก Run จะแสดงหน้า Collection Runner'. Below it, configuration options include an 'Environment' dropdown set to 'local' (highlighted with a red dashed box), 'Iterations' set to 1, 'Delay' set to 0 ms, and several checkboxes: 'Save responses', 'Keep variable values' (which is checked), 'Run collection without using stored cookies', and 'Save cookies after collection run' (which is checked). At the bottom, a blue button labeled '2. Run' is followed by another blue button labeled 'Run shopping cart...' (also highlighted with a red box).



Environment

ผลของ Collection Runner ยังต้องเหมือนเดิม

The screenshot shows the Postman Collection Runner interface with the following details:

- Collection Runner** tab is selected.
- Run Results** tab is visible.
- My Workspace** dropdown shows "No Environment".
- Run Summary** button is orange.
- Export Results**, **Retry**, and **New** buttons are also present.
- Iteration 1** results:
 - GET search product**: Status code is 200, 200 OK, 6 ms, 382 B. Includes tests: "Status code is 200", "สินค้าชื่อ 43 Piece dinner Set", "สินค่าราคา 12.95", "จะต้องแสดงรูป 43_Piece_dinner_Set".
 - GET selected product**: Status code is 200, 200 OK, 10 ms, 276 B. Includes tests: "Status code is 200", "เลือกสินค้า 43 Piece dinner Set id = 2", "สินค้าชื่อ 43 Piece dinner Set", "สินค่ามีราคา 12.95", "เลือกจำนวนสินค้า 10 ชิ้น", "อยู่ในหมวดหมู่ของ CoolKidz".
 - POST order**: This request does not have any tests.



Environment

เปลี่ยนจาก Hard code ให้ใช้ตัวแปร
ของ Environment แทน



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Newman



Newman is a command line Collection Runner for Postman. It allows you to run and test a Postman Collection directly from the command line



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Newman

Example (Class Postman) ★
4 requests

GET search product
GET selected product
POST order
POST confirm payment

- Share Collection
- Manage Roles
- Rename
- Edit
- Create a fork
- Create Pull Request
- Merge changes
- GET Add Request
- Add Folder
- Duplicate
- Export**
- Monitor Collection
- Mock Collection
- Publish Docs
- Remove from workspace

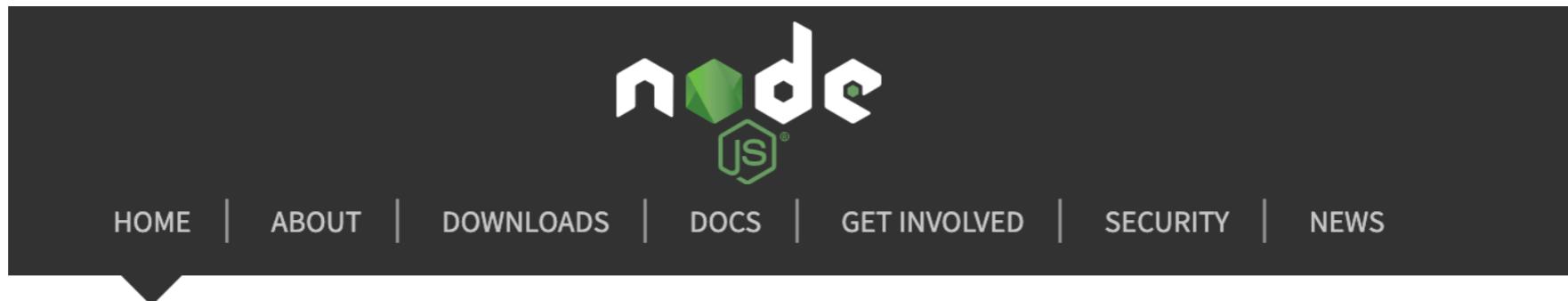
- แนะนำให้อ้าไว้ที่เดียวกับ Project ที่จะทดสอบ
- กำหนดการตั้งชื่อไฟล์

Export Script Test ออกรมา



Newman

ก่อนติดตั้ง newman เช็คก่อนว่ามี node หรือไม่
<https://nodejs.org/en/>



Node.js® is a JavaScript runtime built on [Chrome's V8 JavaScript engine](#).

Download for macOS (x64)

12.16.3 LTS

Recommended For Most Users

14.3.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#) [Other Downloads](#) | [Changelog](#) | [API Docs](#)

Or have a look at the [Long Term Support \(LTS\) schedule](#).



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Newman

พิมพ์คำสั่งใน command line เพื่อติดตั้ง newman

```
npm install -g newman
```

คำสั่ง run newman

```
newman run (ที่อยู่ของไฟล์)/shopping_cart_success.json -e (ที่อยู่ของไฟล์)/local_environment.json
```

อ่านต่อ : <https://www.npmjs.com/package/newman>



Share — copy and redistribute the material in any medium or format.

Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

Result

```
→ shopping-cart git:(master) make run_newman
newman run atdd/api/shopping_cart_success.json -e atdd/api/environment/local_environment.json -d atdd/api/data/shopping_cart_success.json
newman

shopping cart success

→ submit order
POST http://localhost:8000/api/v1/order [200 OK, 167B, 71ms]
✓ Status code is 200
✓ OrderId is existed
✓ Total price 14.95

→ confirm payment
POST http://localhost:8000/api/v1/confirmPayment [200 OK, 145B, 11ms]
✓ Status code is 200
1. notify_message วันเวลาที่ชำระเงิน 1/3/2563 13:30:00 หมายเลขคำสั่งซื้อ 8004359103 คุณสามารถติดตามลิ้นค่าพานของทาง Kerry ด้วยหมายเลข 1785261900
```

	executed	failed
iterations	1	0
requests	2	0
test-scripts	2	0
prerequest-scripts	0	0
assertions	5	1
total run duration:	165ms	
total data received:	66B (approx)	
average response time:	41ms [min: 11ms, max: 71ms, s.d.: 30ms]	

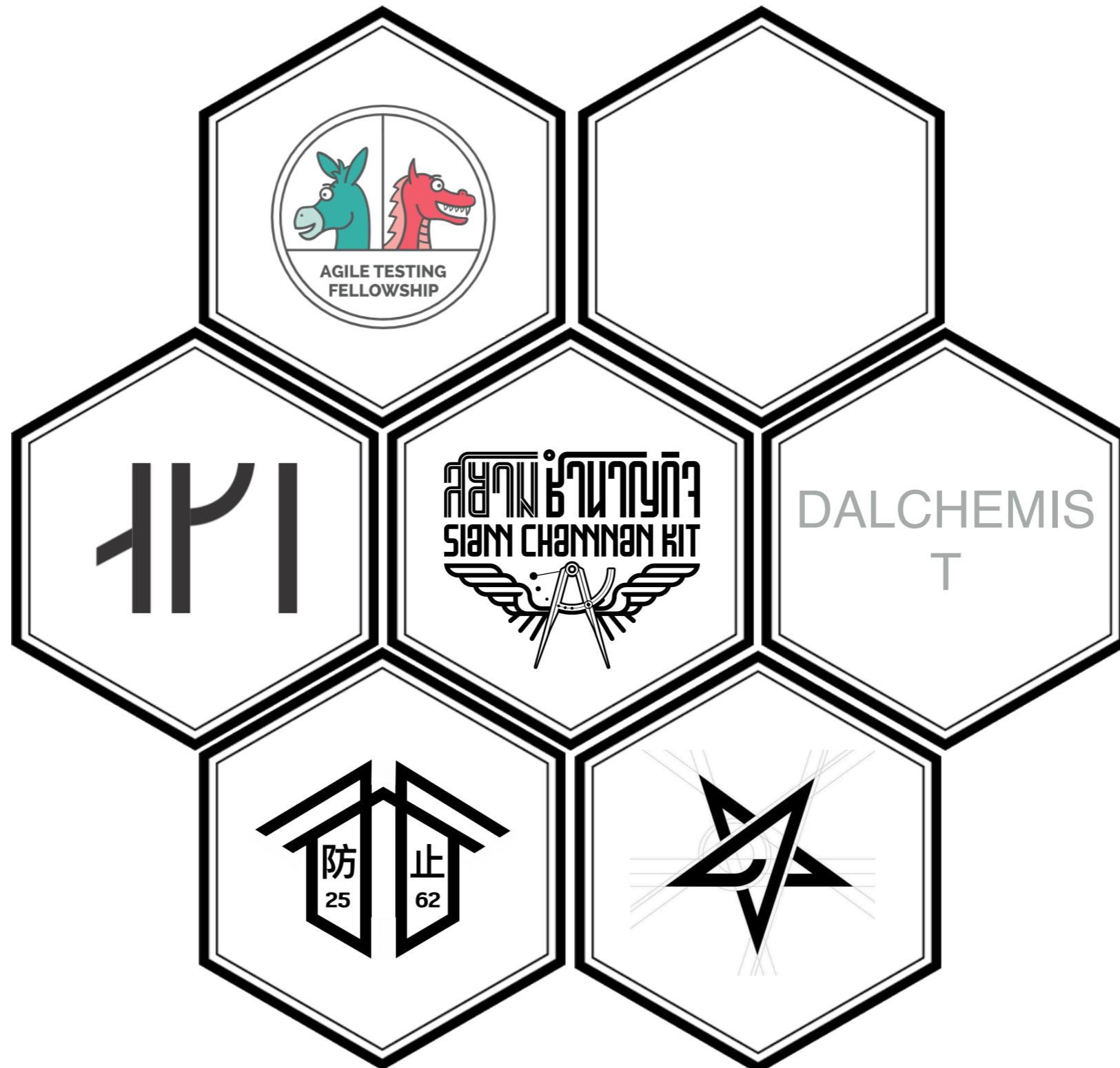
```
# failure detail
1. AssertionError
    notify_message วันเวลาที่ชำระเงิน 1/3/2563 13:30:00 หมายเลขคำสั่งซื้อ 8004359103 คุณสามารถติดตามลิ้นค่าพานของทาง
    Kerry ด้วยหมายเลข 1785261900
    expected '' to deeply equal 'วันเวลาที่ชำระเงิน 1/3/2563 13:30:00 หมายเลขคำสั่งซื้อ 8004359103
    คุณสามารถติดตามลิ้นค่าพานของทาง Kerry ด้วยหมายเลข 1785261900'
    at assertion:1 in test-script
    inside "confirm payment"
make: *** [run_newman] Error 1
```



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.

NonCommercial — You may not use the material for commercial purposes.

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



Share — copy and redistribute the material in any medium or format.
Adapt — remix, transform, and build upon the material.
NonCommercial — You may not use the material for commercial purposes.
This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.