



API & Real-World Project

API & Real-World Project

- Introduction to API
- Build your first REST-API
- ChatGPT API
- Facebook Messenger Chatbot
- Facebook + ChatGPT API



Introduction to API

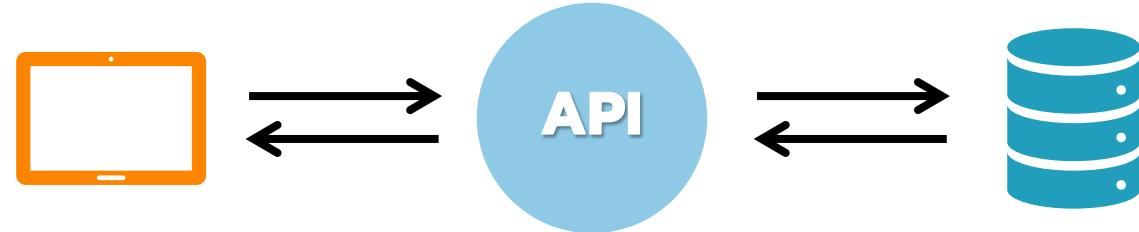
Introduction to API

- What is API?
- Type of API
- REST API



What is API?

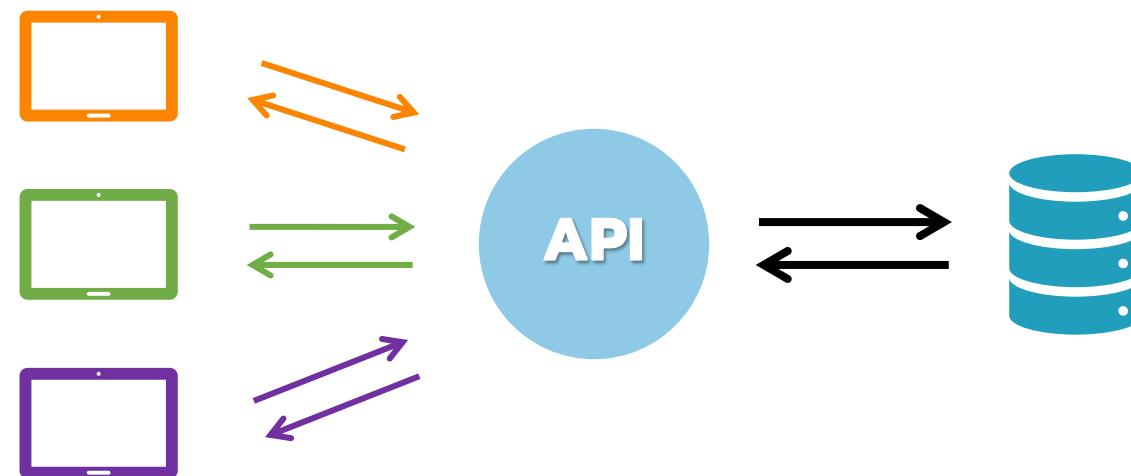
API (Application Program Interface) คือ ตัวกลางในการติดต่อสื่อสารระหว่าง Application และ Server



What is API?

ประโยชน์ของการมี API

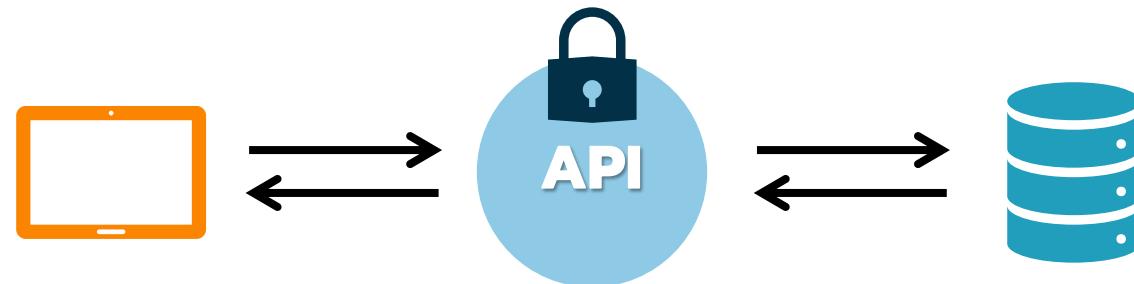
- ✓ สามารถเชื่อมได้กับหลาย Application



What is API?

ประโยชน์ของการมี API

- ✓ สามารถใช้ระบบ Authentication



Introduction to API

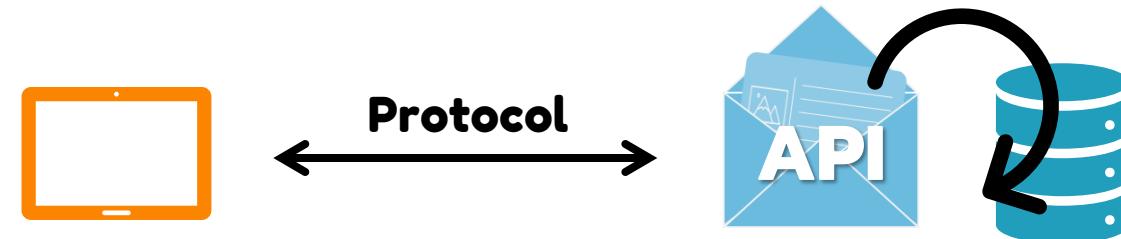
What is API?

- Type of API
- REST API



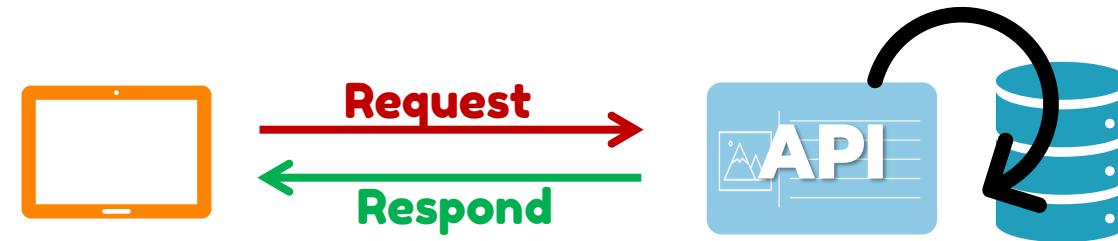
Type of API

SOAP



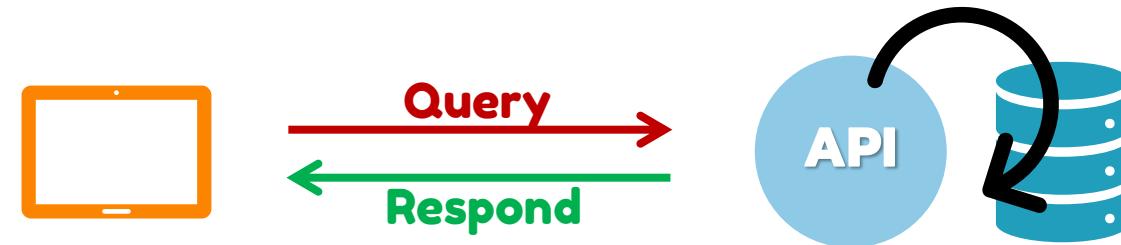
Type of API

REST 



Type of API

GraphQL



Type of API

	1999 SOAP	2000 REST	2015 GraphQL
DETAIL	<ul style="list-style-type: none">- Enveloped message structure- Widely used and established	<ul style="list-style-type: none">- Stateless- Uniform Interface- Flexible data formatting	<ul style="list-style-type: none">- รูปแบบของการดึงข้อมูลที่ Flexible
FORMAT	XML	XML, JSON, HTML,...	JSON
USE CASE	<ul style="list-style-type: none">- Payment- Identity Management- CRM solutions- Financial and telecommunication service- Legacy system support	<ul style="list-style-type: none">- Public APIs- Simple resource-driven apps	<ul style="list-style-type: none">- Mobile APIs- Complex systems- Micro-services

Introduction to API

- What is API?**
- Type of API**
- REST API



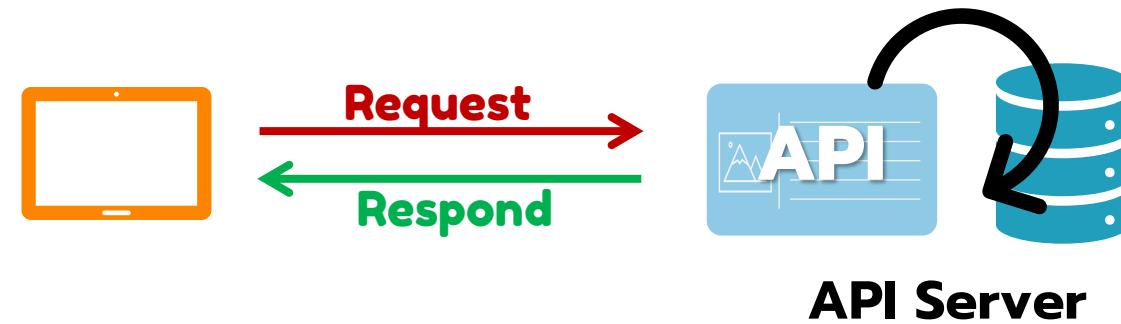
REST API

- How REST API work?
- Why REST API?
- HTTP Request component
- JSON
- HTTP Method
- HTTP Response



How REST API work?

หลักการทำงานของ REST API คือ Application ส่งคำสั่ง (Request) ไปยัง API server จากนั้น API server ทำการ process คำสั่งที่ส่งเข้ามาแล้วตอบกลับ (Respond) ไปยัง Application



REST API

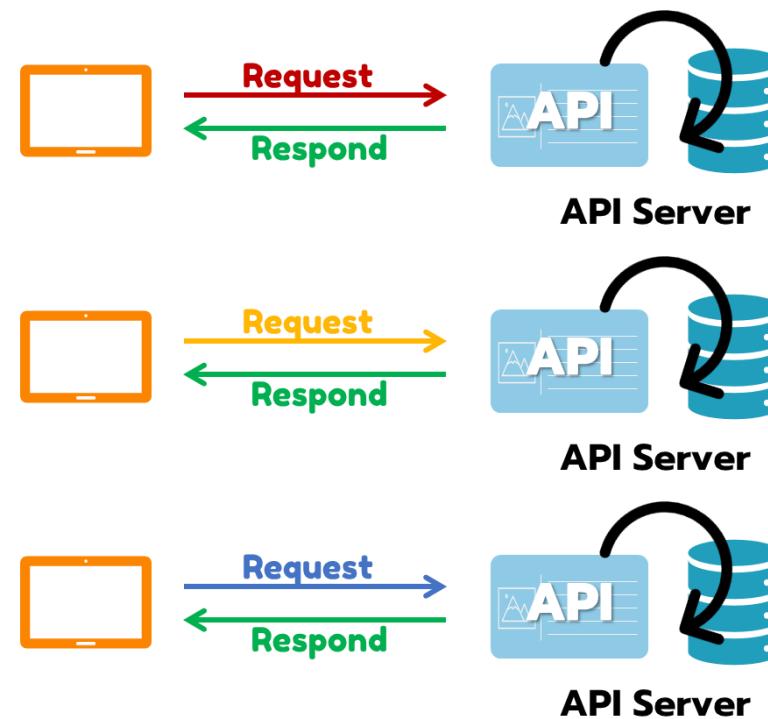
✓ How REST API work?

- Why REST API?
- HTTP Request component
- JSON
- HTTP Method
- HTTP Response



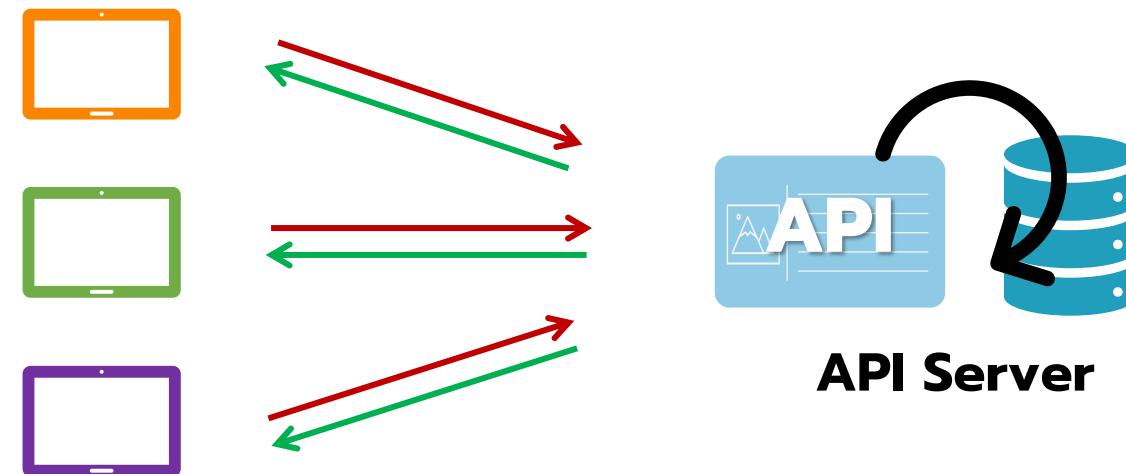
Why REST API?

1. **Stateless** : Request กี่ส่งไปให้ API Server จะไม่ขึ้นกับ Request ก่อนหน้า



Why REST API?

2. **Uniform Interface** : สามารถเชื่อมกับ Application ได้หลายเครื่อง ด้วย code ชุดเดียว กัน (ภาษาเดียว กัน)



Why REST API?

3. **Support for different data formats** : สามารถใช้งานได้กับข้อมูลหลายรูปแบบ เช่น

XML, JSON, HTML, ...

Why REST API?

4. Easy to understand and use

```
import requests
import json

url = "https://17fb-184-22-118-108.ngrok-free.app/post3"

payload = json.dumps({
    "name": "John Doe",
    "age": 30,
    "isEmployee": true,
    "address": {
        "street": "123 Main St",
        "city": "New York",
        "state": "NY",
        "zip": "10001"
    },
    "phoneNumbers": ["212-555-1234", "646-555-4567"],
    "spouse": null
})

headers = {
    'password': '12345678',
    'Content-Type': 'application/json'
}

response = requests.request("POST", url, headers=headers, data=payload)

print(response.text)
```

URL

Header

Body

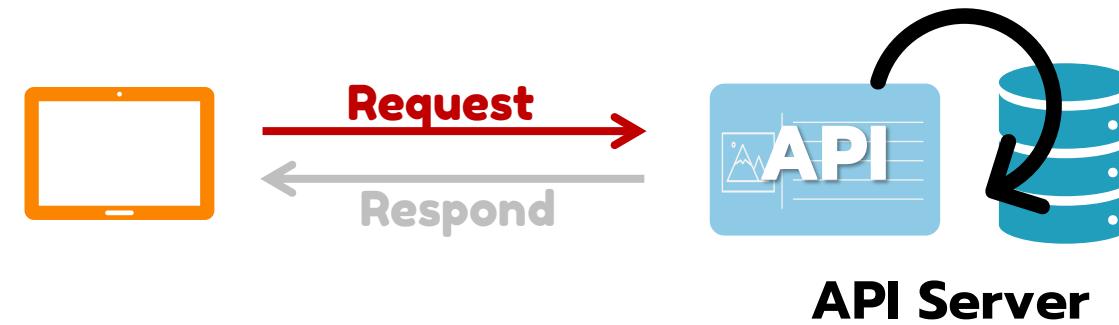
REST API

- ✓ **How REST API work?**
- ✓ **Why REST API?**
 - HTTP Request component
 - JSON
 - HTTP Method
 - HTTP Response



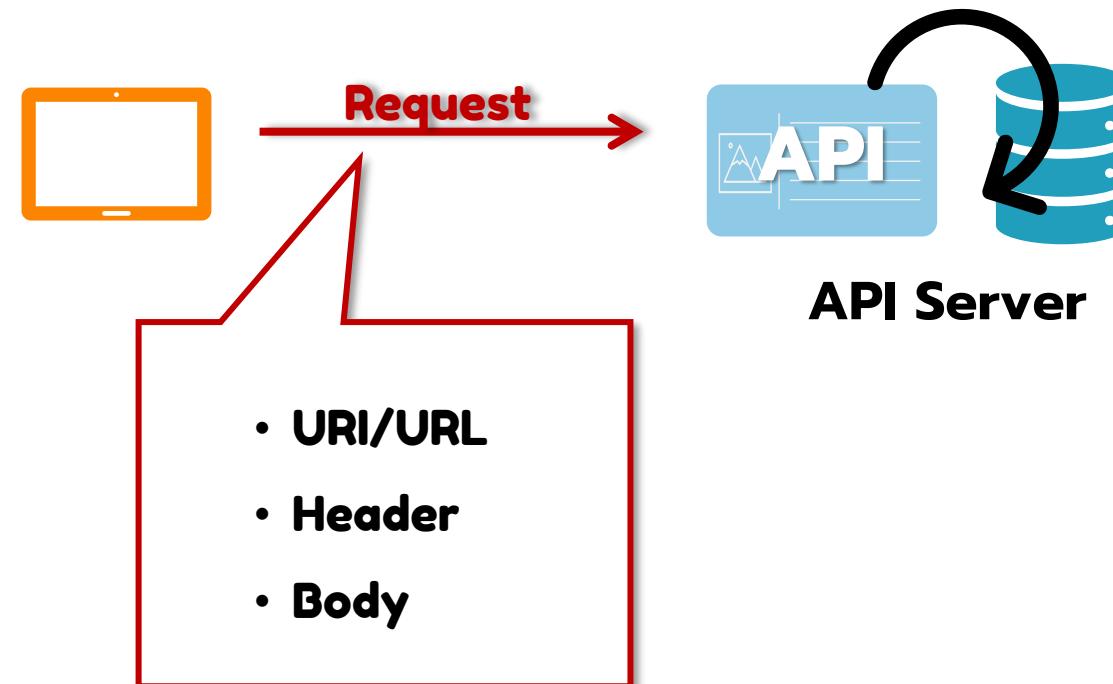
HTTP Request component

HTTP Request component คือ ส่วนของ Request ที่ส่งจาก Application ไปยัง API Server



HTTP Request component

HTTP Request component จะประกอบไปด้วย 3 ส่วน คือ **(1) URI/URL (2) Header (3) Body**



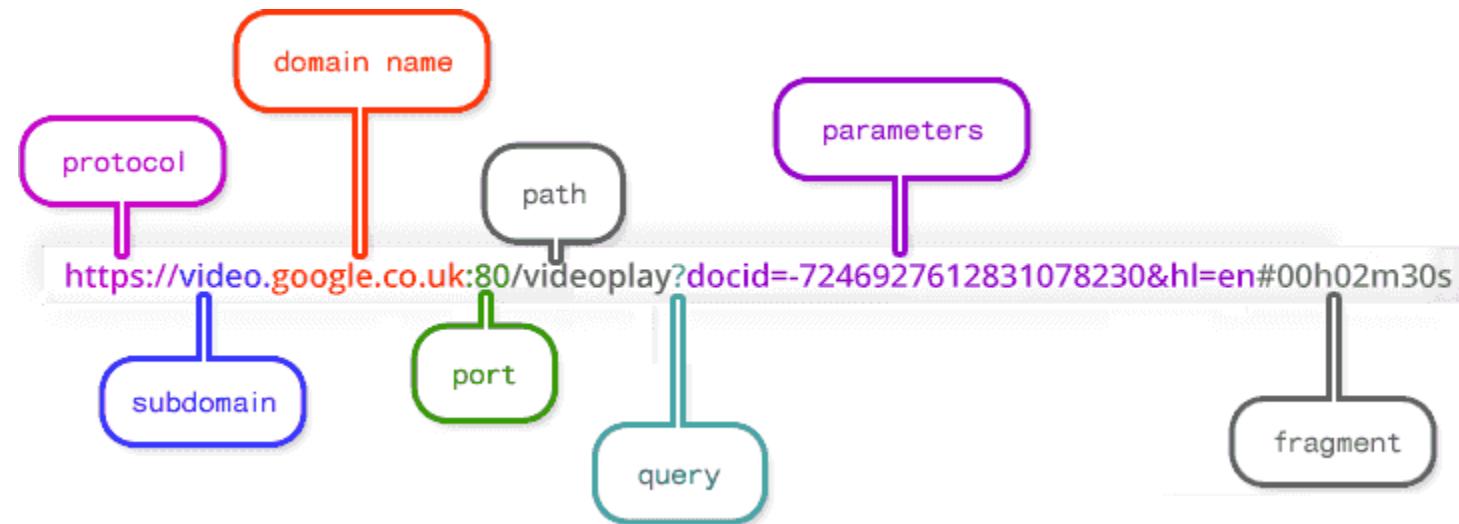
HTTP Request component

- ✍ URI / URL
- ✍ Header
- ✍ Body



URI / URL

ในส่วนนี้ของ HTTP Request จะระบุ URI/URL กี่ต้องการจะเชื่อมต่อกับ API Server



Ref : <https://medium.com/@joseph.pyram/9-parts-of-a-url-that-you-should-know-89fea8e11713>

HTTP Request component

- ✍ **URI / URL**
- ✍ Header
- ✍ Body



Header

ໃນສ່ວນນີ້ຂອງ HTTP Request ຈະປະກອບໄປຫລາກຫລາຍສ່ວນ ເຊັ່ນ

HTTP Headers

```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cache-Control: max-age=0
Host: localhost:8080
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:17.0) Gecko/20100101 Firefox/17.0
```

Header

- Content type **XML**

```
<!-- XML Example -->
<book>
    <title>Harry Potter and the Sorcerer's Stone</title>
    <author>J.K. Rowling</author>
    <year>1997</year>
</book>
```

Header

- Content type **HTML**

```
<!-- HTML Example -->
<!DOCTYPE html>
<html>
<head>
    <title>My Web Page</title>
</head>
<body>
    <h1>Welcome to my website!</h1>
    <p>This is some sample content.</p>
</body>
</html>
```

Header

- Content type

JSON 

```
// JSON Example
{
    "name": "John Doe",
    "age": 30,
    "city": "New York"
}
```

HTTP Request component

- ✍ **URI / URL**
- ✍ **Header**
- ✍ **Body**



Body

ในส่วนนี้ของ HTTP Request จะเป็นข้อมูลและคำสั่งที่ Application ต้องการส่งไปยัง API Server โดยส่วนใหญ่จะอยู่ในรูปแบบของ JSON

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

HTTP Request component

- ✍ **URI / URL**
- ✍ **Header**
- ✍ **Body**



REST API

- ✓ **How REST API work?**
- ✓ **Why REST API?**
- ✓ **HTTP Request component**
 - JSON
 - HTTP Method
 - HTTP Response



JSON

- ✍ Why JSON?
- ✍ JSON Syntax
- ✍ JSON Data Types



Why JSON?

1. **Readable** : อ่านและทำความเข้าใจได้ง่าย

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

Why JSON?

2. **Lightweight** : ใช้ตัวอักษรน้อยเมื่อเทียบกับข้อมูลรูปแบบอื่น

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <name>John Doe</name>
  <age>30</age>
  <isEmployee>true</isEmployee>
  <address>
    <street>123 Main St</street>
    <city>New York</city>
    <state>NY</state>
    <zip>10001</zip>
  </address>
  <phoneNumbers>212-555-1234</phoneNumbers>
  <phoneNumbers>646-555-4567</phoneNumbers>
  <spouse/>
</root>
```

XML

```
{
  "name": "John Doe",
  "age": 30,
  "isEmployee": true,
  "address": {
    "street": "123 Main St",
    "city": "New York",
    "state": "NY",
    "zip": "10001"
  },
  "phoneNumbers": ["212-555-1234", "646-555-4567"],
  "spouse": null
}
```

JSON

Why JSON?

3. Programming language independent : ไม่ขึ้นกับภาษาของโปรแกรม



Why JSON?

4. Supports Complex Data Types : สามารถสร้างข้อมูลที่มีความซับซ้อน

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

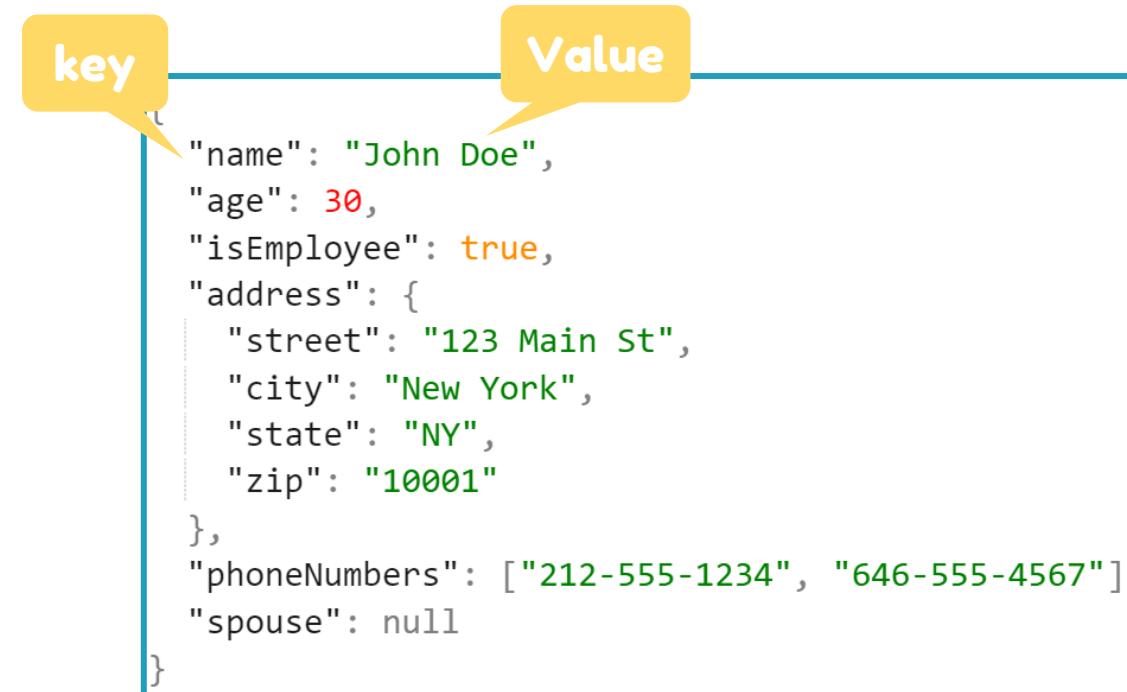
JSON

- ✍ Why JSON?
- ✍ JSON Syntax
- ✍ JSON Data Types



JSON Syntax

JSON มีรูปแบบคล้ายกับ **dictionary** ใน **python** ซึ่งจะประกอบไปด้วยค่า **key** และ **value**



JSON

-  **Why JSON?**
-  **JSON Syntax**
-  **JSON Data Types**



JSON Data Types

1. String : ข้อมูลที่อยู่ใน “ ”

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```



JSON Data Types

2. **Number** : ข้อมูลที่อยู่ในรูปแบบของตัวเลข และจะไม่อยู่ใน ""

```
{  
    "name": "Jo  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

Number

JSON Data Types

3. Boolean : ค่าความจริง true/ false

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true, Boolean  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

JSON Data Types

4. Null : ไม่มีข้อมูล

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": [123-456-1234, "646-555-4567"],  
    "spouse": null  
}
```

Null

JSON Data Types

5. **Object** : อยู่ใน { } ซึ่งจะประกอบไปด้วย key และ value โดยจะถูกคั่นด้วย ,

Object

```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```

JSON Data Types

6. **List** : อยู่ใน [] ซึ่งจะประกอบไปด้วยชนิดข้อมูลใดก็ได้ที่กล่าวมาข้างต้น (String, Number, Boolean, Null, Object} โดยจะถูกคั่นด้วย ,

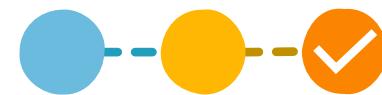
```
{  
    "name": "John Doe",  
    "age": 30,  
    "isEmployee": true,  
    "address": {  
        "street": "123 Main St",  
        "city": "New York",  
        "state": "NY",  
        "zip": "10001"  
    },  
    "phoneNumbers": ["212-555-1234", "646-555-4567"],  
    "spouse": null  
}
```



List of String

JSON

- ✍ **Why JSON?**
- ✍ **JSON Syntax**
- ✍ **JSON Data Types**



REST API

- ✓ **How REST API work?**
- ✓ **Why REST API?**
- ✓ **HTTP Request component**
- ✓ **JSON**
 - HTTP Method
 - HTTP Response



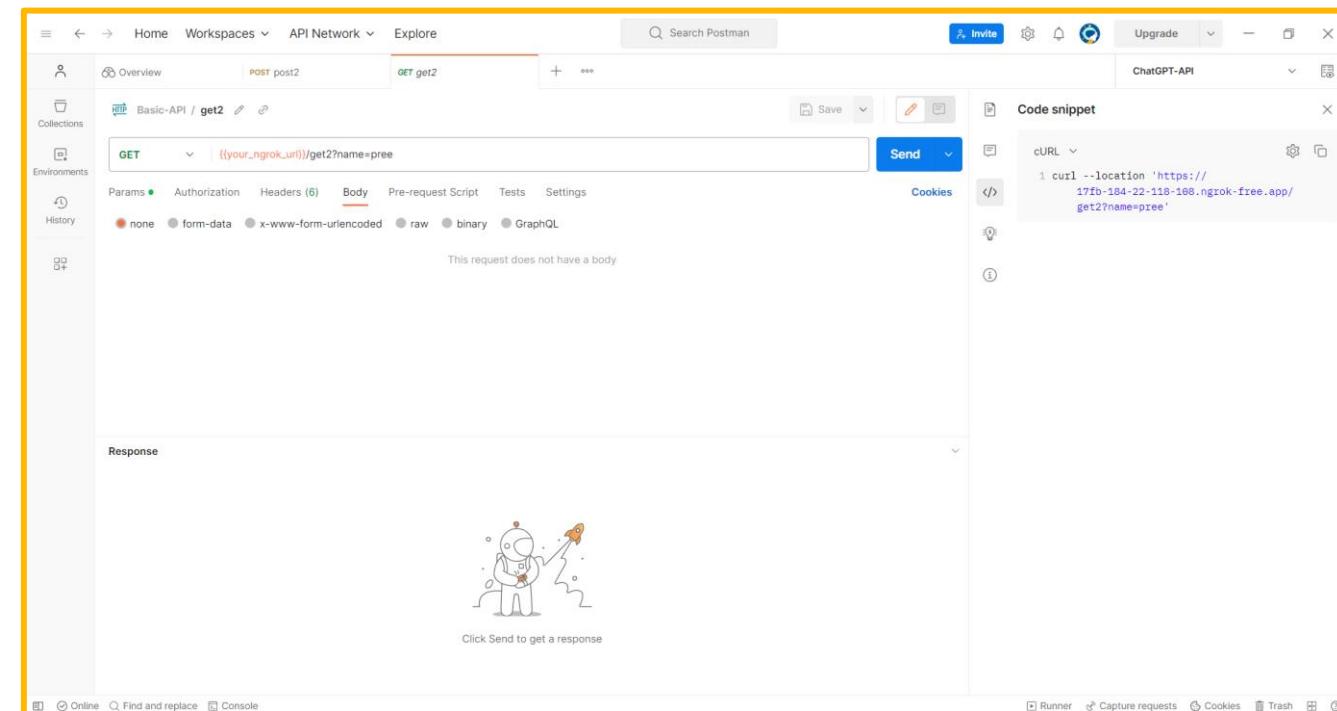
HTTP Method

HTTP method คือ วิธีการ (Method) ที่ระบุถึงการดำเนินการต่าง ๆ ของ Request ที่ส่งเข้ามา โดยวิธีการหลักที่มักใช้งานได้แก่

1. GET
2. POST
3. PUT
4. DELETE

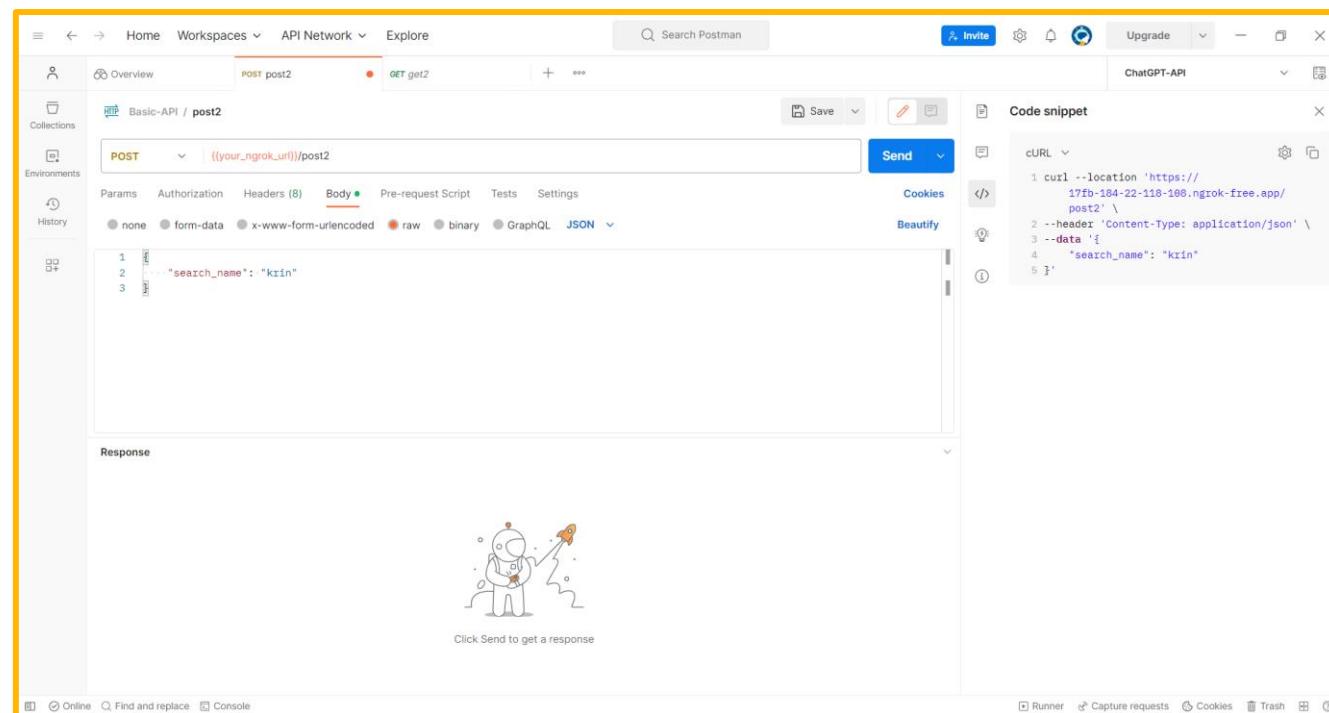
HTTP Method

1. **GET** : Method ที่ใช้เรียกข้อมูลใน API Server ไม่สามารถส่งข้อมูลที่อยู่ในส่วนของ body ถ้าต้องการส่งข้อมูลต้องส่งในรูปแบบของ parameter ที่อยู่ในส่วน URI/URL



HTTP Method

2. **POST** : Method ที่ใช้ส่งข้อมูลที่มาพร้อม Request ไปยัง API Server จากนั้นดำเนินการตาม Request



HTTP Method

3. **PUT** : Method ที่ใช้ในการอัพเดตข้อมูลใน API Server
4. **DELETE** : Method ที่ใช้ในการลบข้อมูลใน API Server

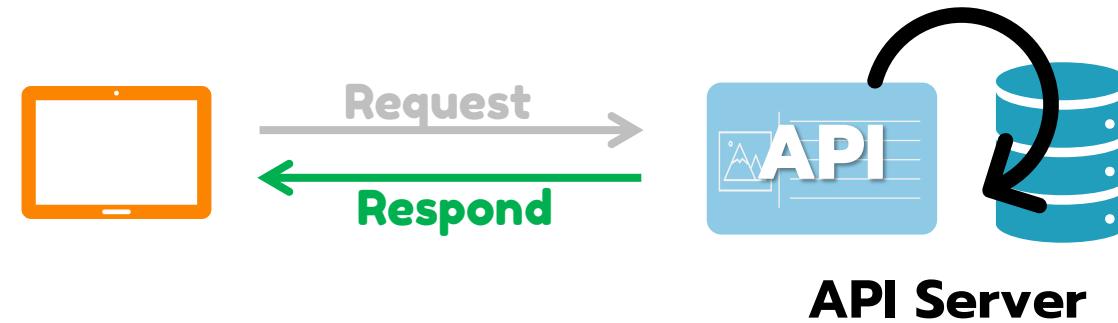
REST API

- ✓ **How REST API work?**
- ✓ **Why REST API?**
- ✓ **HTTP Request component**
- ✓ **JSON**
- ✓ **HTTP Method**
 - HTTP Response



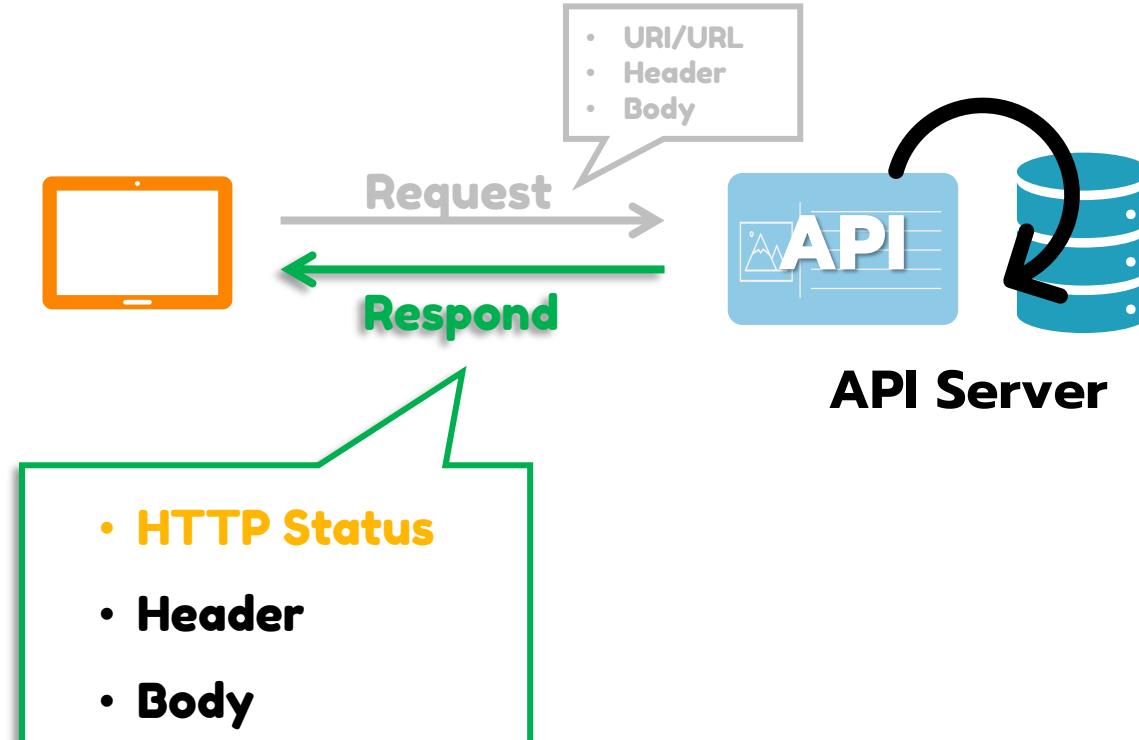
HTTP Respond

HTTP Respond คือ ส่วนของ Respond ที่ได้ดำเนินการตาม Request ที่ส่งมาแล้ว จะถูกส่งกลับจาก API Server ไปยัง Application



HTTP Respond

HTTP Respond component จะประกอบไปด้วย 3 ส่วน คือ **(1) HTTP Status** **(2) Header** **(3) Body**



HTTP Respond

HTTP Status คือ การบอกสถานะของการส่งข้อมูลกลับ โดยจะแสดงผลเป็นรหัสดังนี้

- 1xx Informational Response
- 2xx Success
- 3xx Redirection
- 4xx Client Errors
- 5xx Server Errors

HTTP Respond

HTTP Status

- **1xx Success** : กำลังประมวลผล
- **2xx Informational Response** : การประมวลผลสำเร็จ เช่น
 - **200 OK**: The request is OK and successful.
 - **204 No Content**: The server successfully processed the request, but is not returning any content.
- **3xx Redirection** : ดำเนินการบางอย่างเพื่อให้ประมวลผลสำเร็จ เช่น
 - **301 Moved Permanently** : This status code means that the URL of the requested resource has been changed permanently. The new URL is given in the response

HTTP Respond

HTTP Status

- **4xx Client Errors : เกิดข้อผิดพลาดในการประมวลผล เช่น**

- **401** Bad Request : The server could not understand the request due to invalid syntax.
- **404** Unauthorized : The request lacks valid authentication credentials for the target resource.
- **404** Not Found : The server can't find the requested resource.

- **5xx Server Errors : เกิดข้อผิดพลาดใน API Server เช่น**

- **500** Internal Server Error : The server encountered an unexpected condition which prevented it from fulfilling the request.

REST API

- ✓ **How REST API work?**
- ✓ **Why REST API?**
- ✓ **HTTP Request component**
- ✓ **JSON**
- ✓ **HTTP Method**
- ✓ **HTTP Response**



Introduction to API

- What is API?**
- Type of API**
- REST API**



API & Real-World Project

- Introduction to API
- Build your first REST-API
- ChatGPT API
- Facebook Messenger Chatbot
- Facebook + ChatGPT API



Build your first REST-API

Build your first REST-API

- API Structure
- Prerequisite
- First Flask application
- HTTP Method
- Basic Authentication



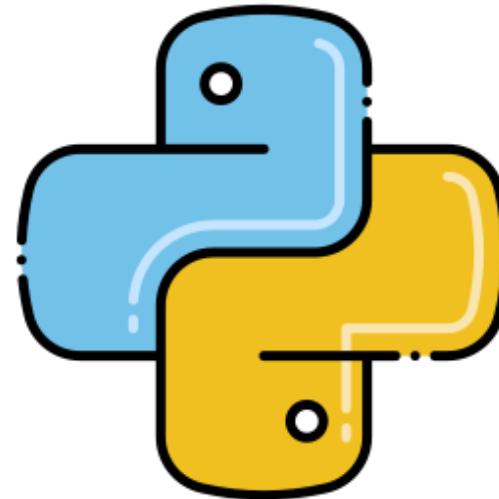
API Structure

การสร้าง API ส่วนใหญ่แล้วจะใช้เครื่องมือดังนี้

1. Python
2. Flask
3. Ngrok

API Structure

1. **Python** : เครื่องมือพื้นฐานสำหรับสร้าง API



API Structure

1. REST API : รูปแบบของ API ที่มีการจัดโครงสร้างเป็นชั้นๆ เช่น ชั้น Model, ชั้น Controller, ชั้น View และชั้น Database
2. **Flask** : Web Framework ของ Python โดยก้าวไปแล้วจะใช้โปรแกรมนี้ในการเขียน API เนื่องจากเป็นโปรแกรมที่ใช้งานง่ายในการจัดการ Request และ Respond



API Structure

3. **Ngrok** : โปรแกรมสำหรับแปลง URL จากการสร้าง API ให้เป็นแบบ Public ซึ่งจะทำให้สามารถใช้งานได้บน internet



Build your first REST-API

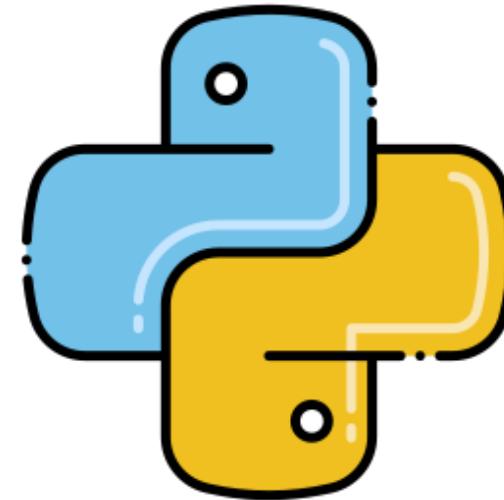
API Structure

- Prerequisite
- How to install Flask
- First Flask application
- HTTP Method
- Basic Authentication



Prerequisite

“ Install Python ”

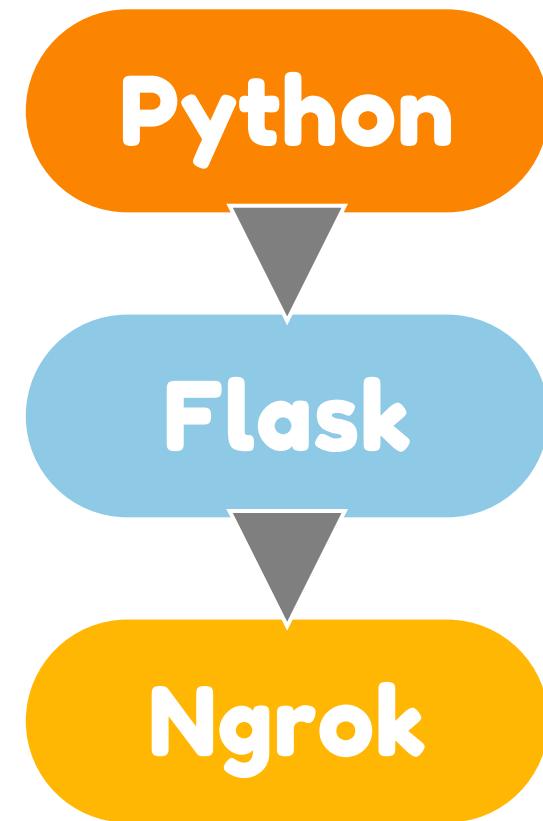


Build your first REST-API

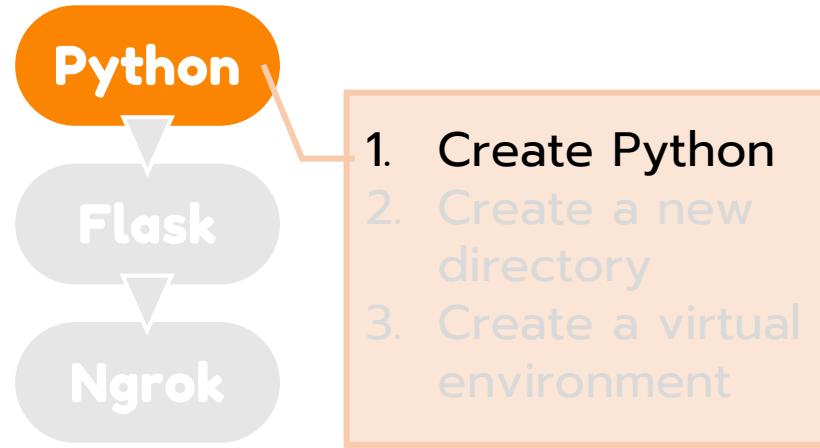
- API Structure**
- Prerequisite**
- First Flask application
- HTTP Method
- Basic Authentication



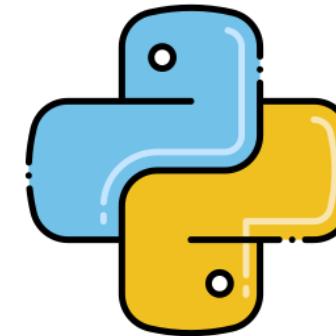
First Flask application



First Flask application

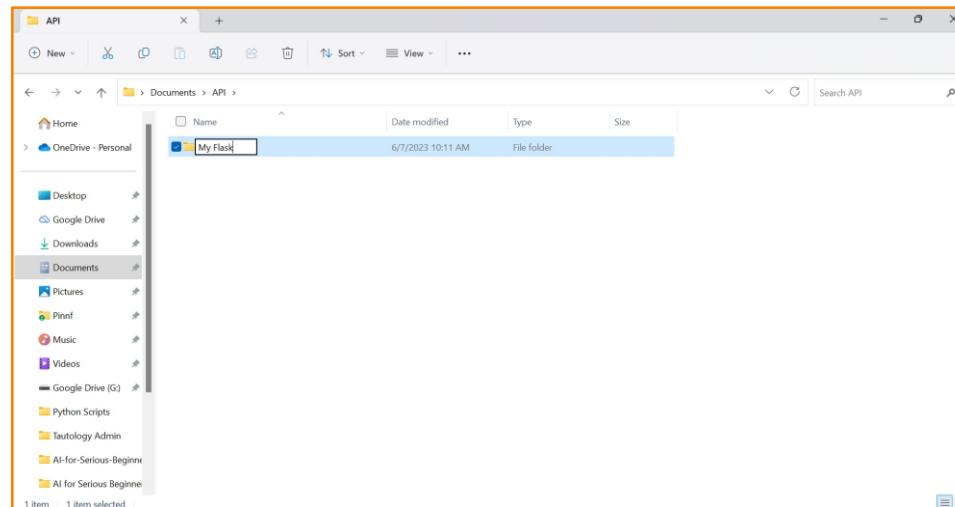
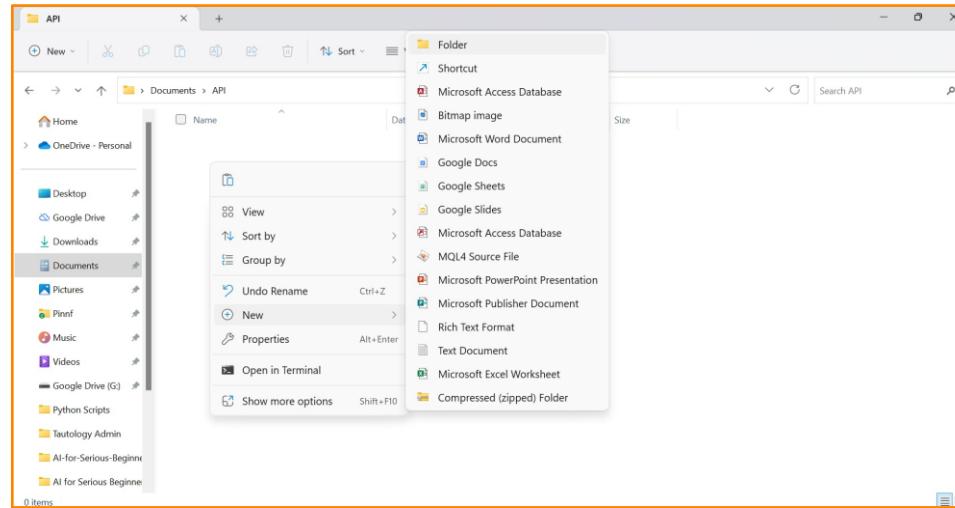
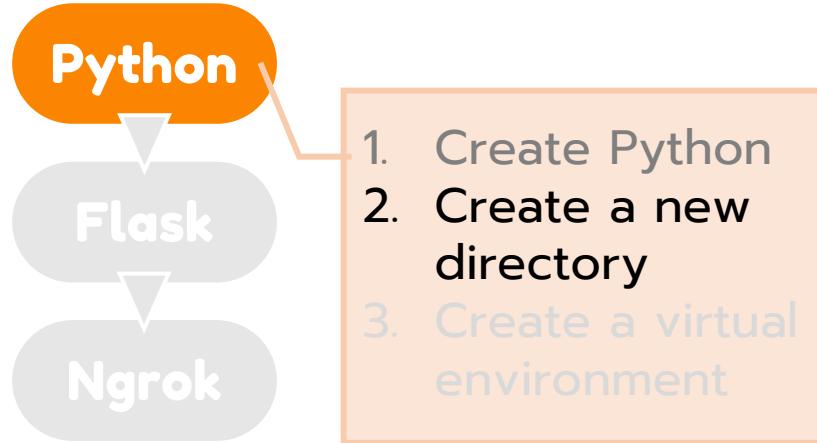


Download Python

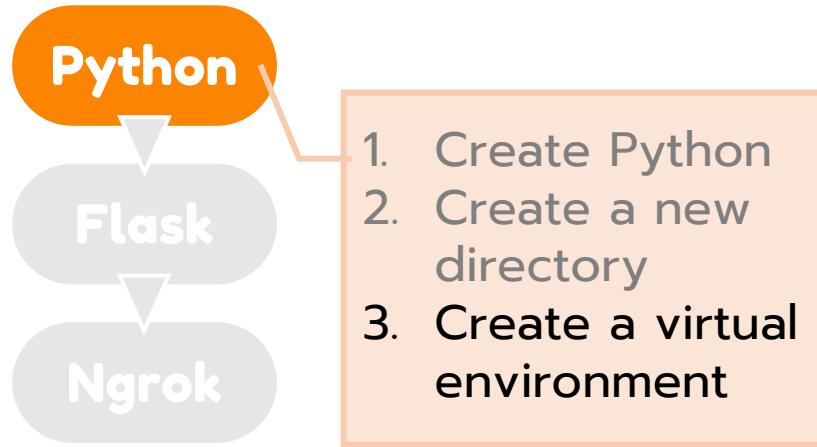


<https://www.python.org/downloads/>

First Flask application

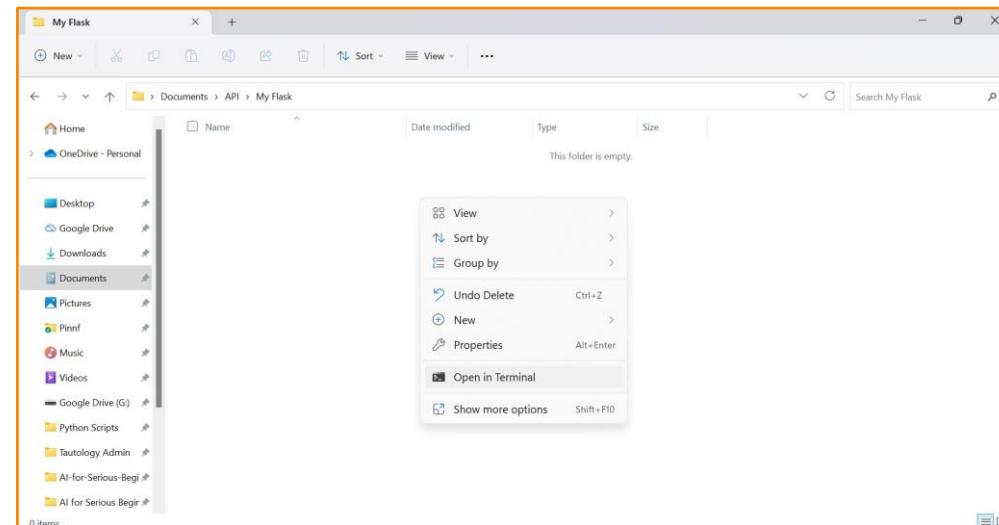


First Flask application

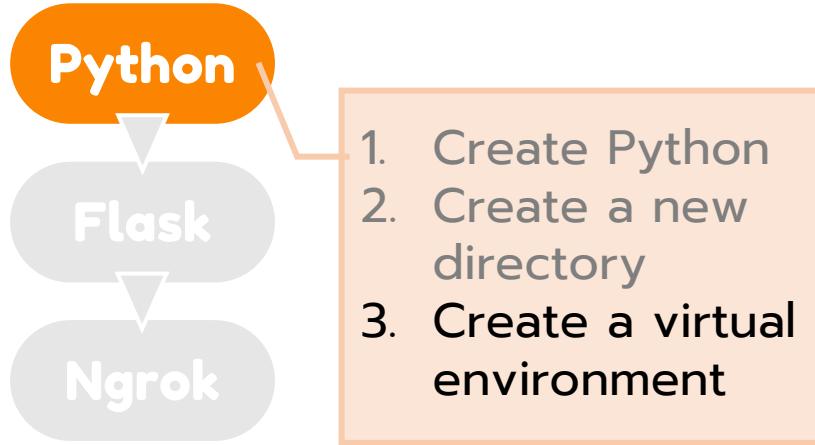


1

Open in Terminal



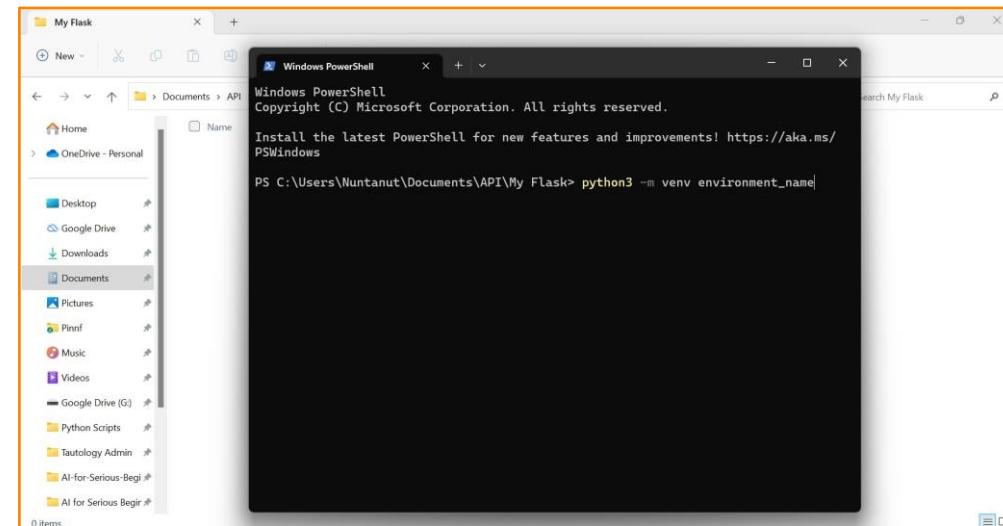
First Flask application



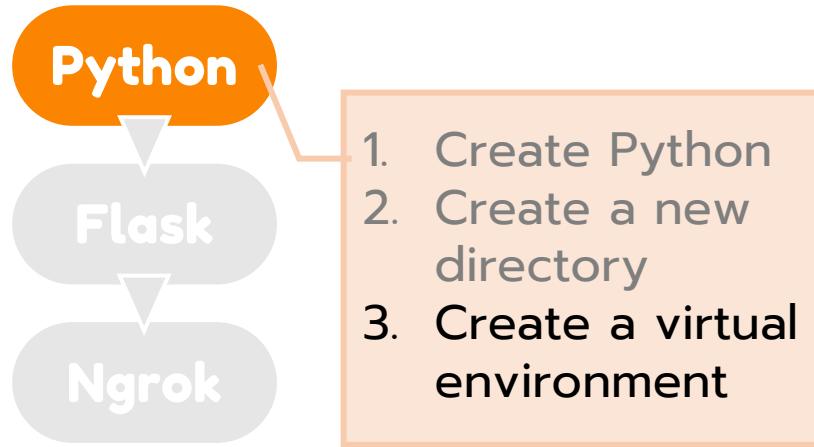
2

Create virtual environment by running

```
python3 -m venv environment_name
```

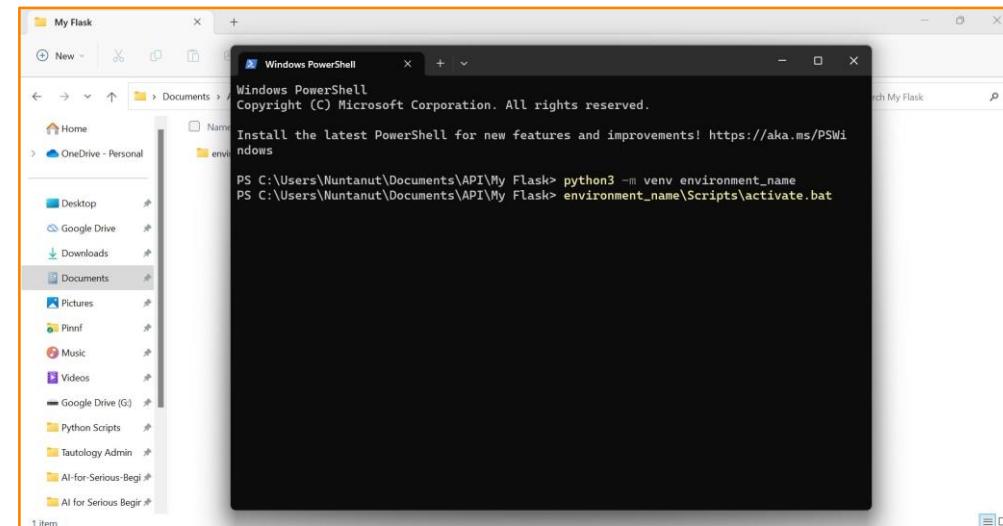


First Flask application

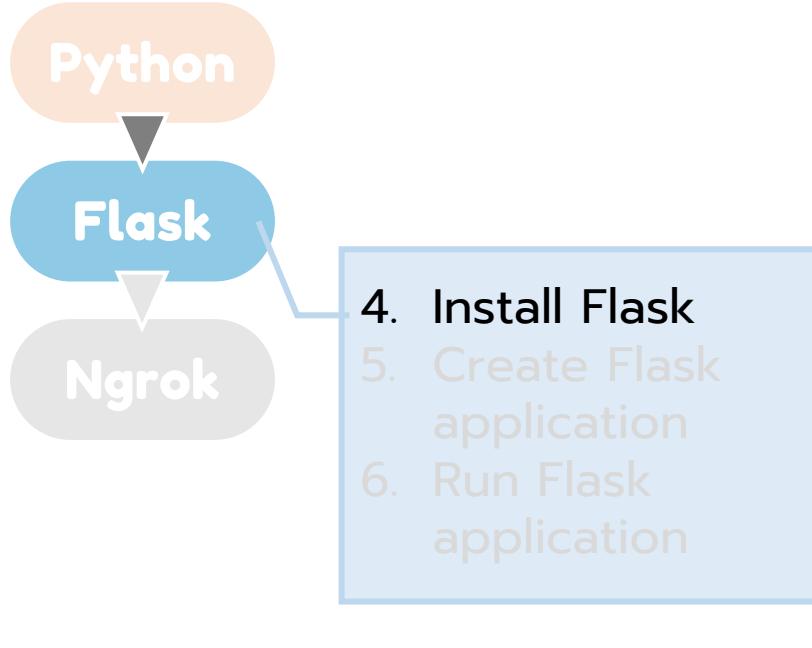


③

Activate the virtual environment by running
environment_name\Scripts\activate.bat



First Flask application

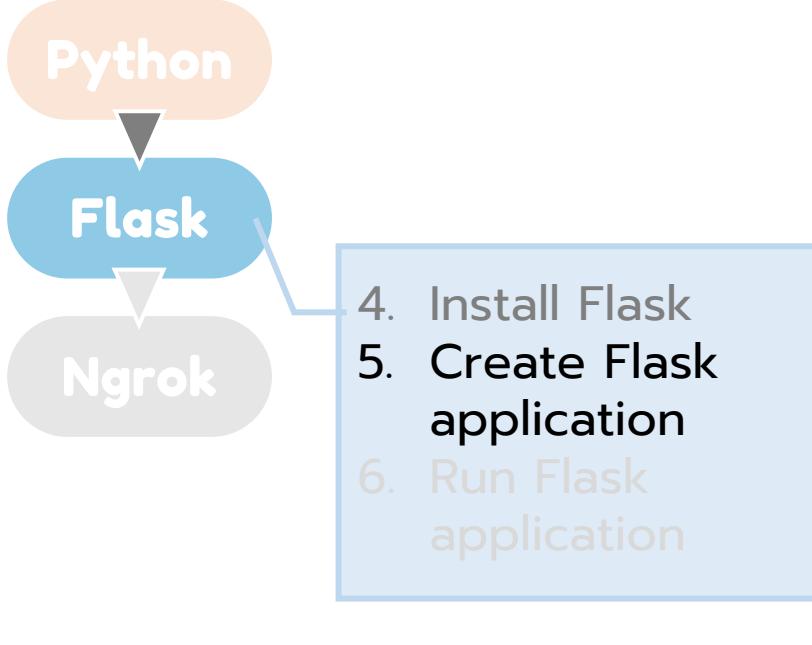


Run `pip install flask`

```
PS C:\Users\Nuntanut\Documents\API\My Flask> python3 -m venv environment_name
PS C:\Users\Nuntanut\Documents\API\My Flask> environment_name\Scripts\activate.bat
PS C:\Users\Nuntanut\Documents\API\My Flask> pip install flask
Collecting flask
  Downloading Flask-2.3.2-py3-none-any.whl (96 kB)
    96.9/96.9 kB 2.8 MB/s eta 0:00:00
Collecting Werkzeug>=2.3.3
  Downloading Werkzeug-2.3.4-py3-none-any.whl (242 kB)
    242.5/242.5 kB 3.7 MB/s eta 0:00:00
Collecting Jinja2>=3.1.2
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    133.1/133.1 kB 4.0 MB/s eta 0:00:00
Collecting itsdangerous>=2.1.2
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting click>=8.1.3
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
    96.6/96.6 kB 5.8 MB/s eta 0:00:00
Collecting blinker>=1.6.2
  Downloading blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
```

A screenshot of a Windows PowerShell window titled "Windows PowerShell". The command `pip install flask` is being run, and the output shows the progress of downloading various dependencies. The PowerShell window is overlaid on a file explorer window showing a folder structure named "My Flask".

First Flask application



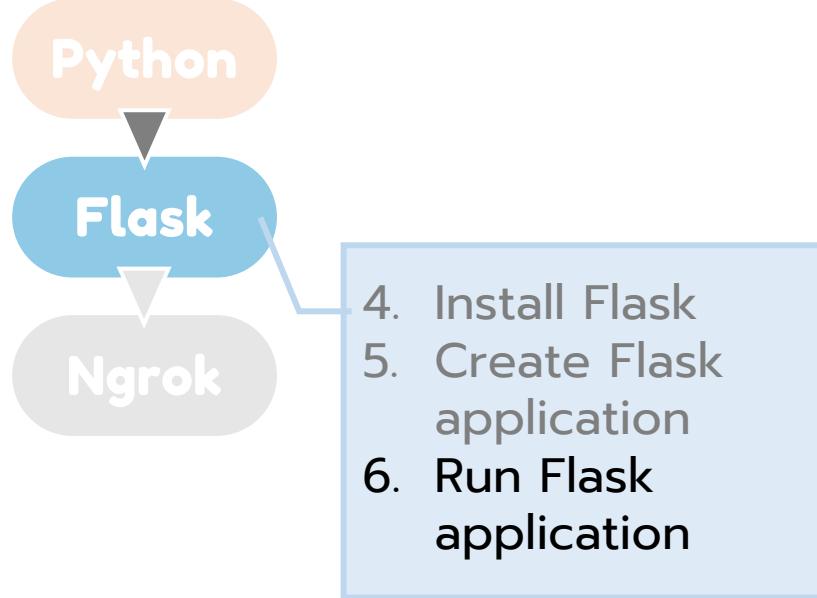
Create a new Python file

A screenshot of Visual Studio Code showing a new Python file named `app.py` in the editor. The code is as follows:

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

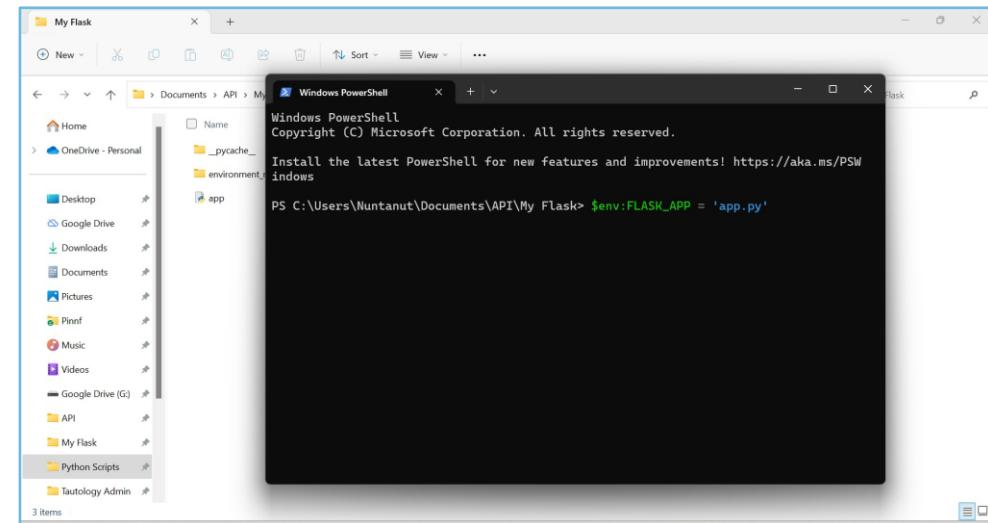
First Flask application



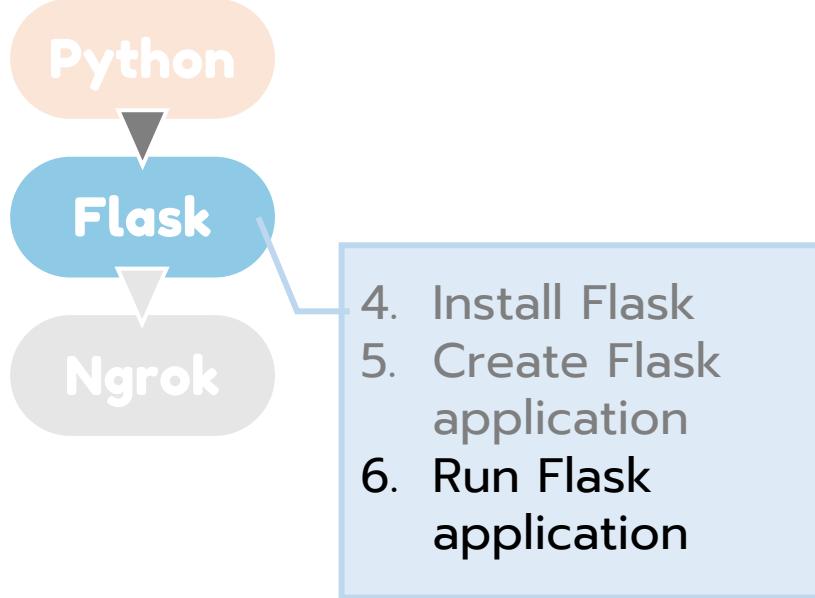
1

Run the commands

✓ `$env:FLASK_APP = 'app.py'`
`flask run --port 5000`



First Flask application



1

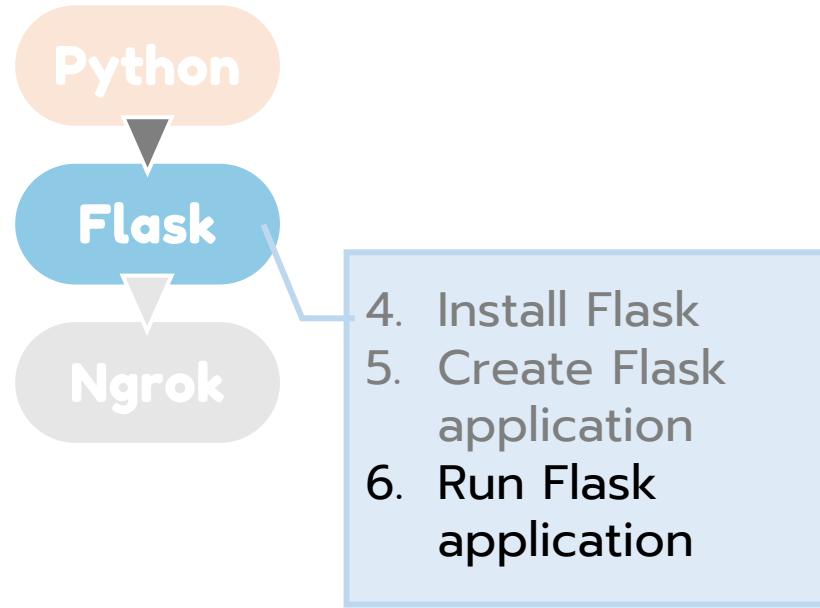
Run the commands

- ✓ \$env:FLASK_APP = 'app.py'
- ✓ flask run --port 5000

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

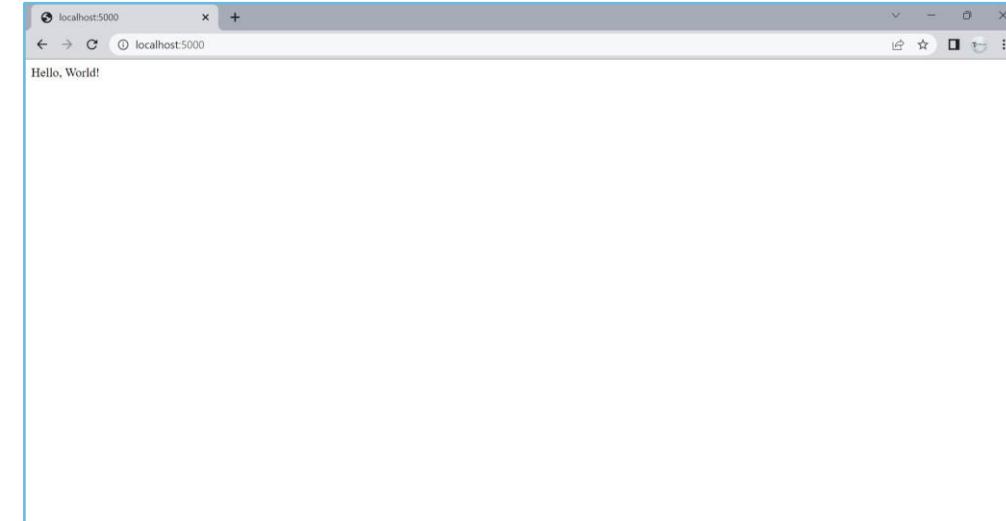
PS C:\Users\Nuntanut\Documents\API\My Flask> $env:FLASK_APP = 'app.py'
PS C:\Users\Nuntanut\Documents\API\My Flask> python3 -m flask run --port 5000
 * Serving Flask app 'app.py'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

First Flask application

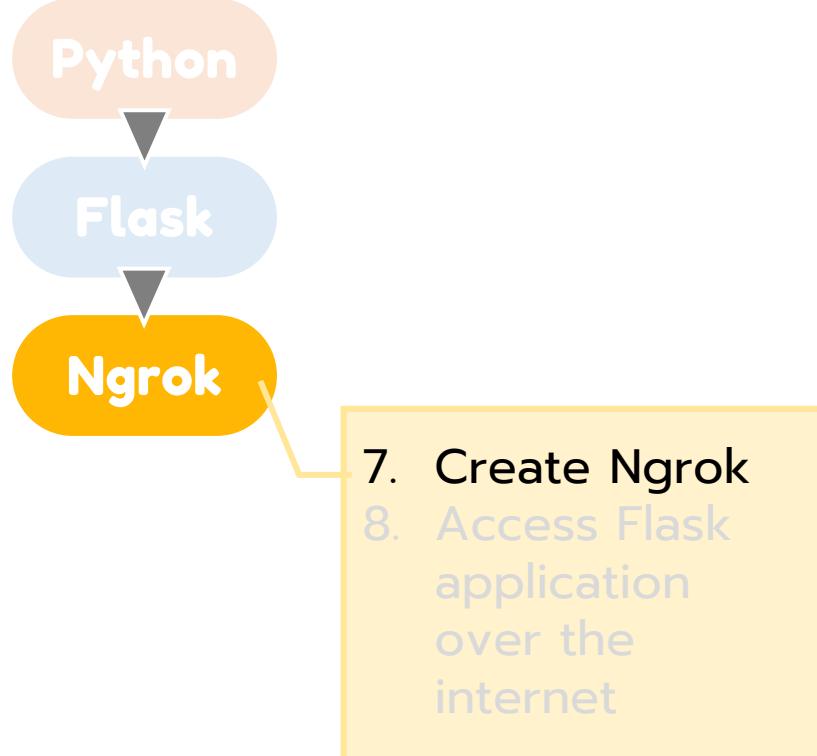


2

Open <http://localhost:5000>



First Flask application



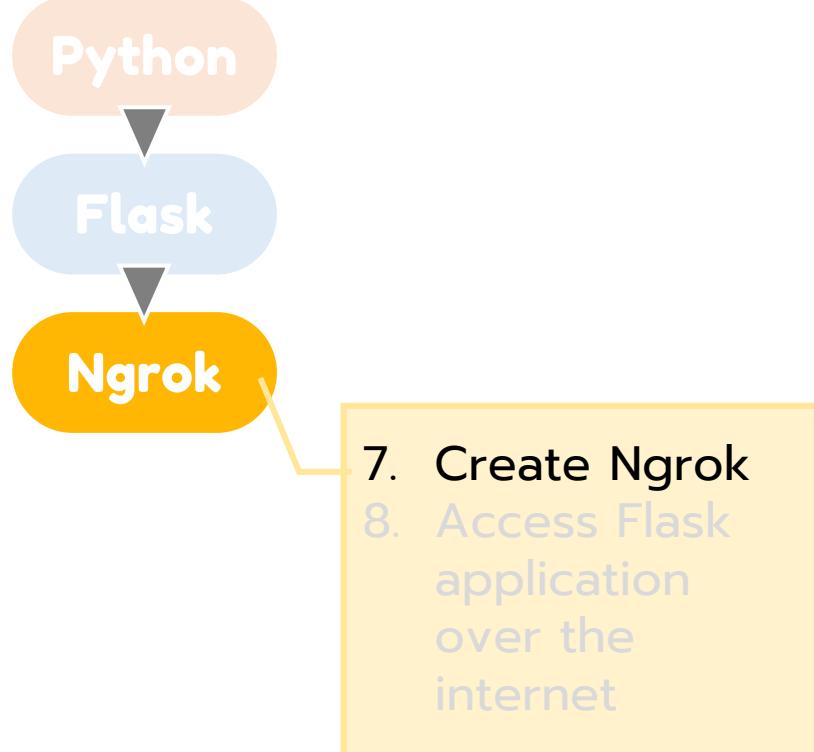
1

Download Ngrok

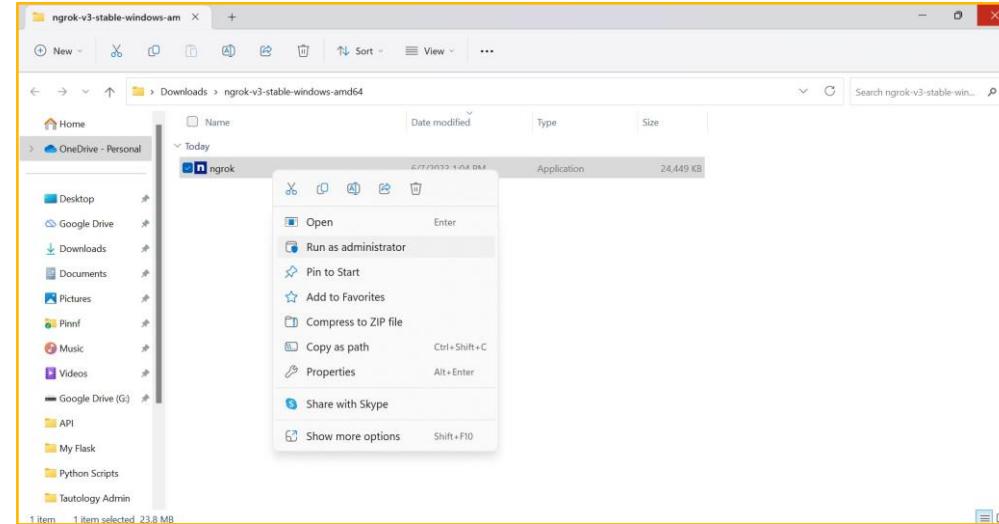


<https://ngrok.com/download>

First Flask application



2

Open Ngrok

First Flask application



4

Run

```
ngrok.exe http 5000
```

A screenshot of a Windows command prompt window titled 'ngrok-v3-stable-windows-amd64'. The window shows the output of the command 'ngrok.exe http 5000'. The output includes help text for the ngrok command, followed by a list of commands and their descriptions. The command 'ngrok http 5000' is highlighted at the bottom.

```
prints author and licensing information
diagnose connection issues
Help about any command
start an HTTP tunnel
run and control an ngrok service on a target operating system

start tunnels by name from the configuration file
start a TCP tunnel
start a TLS tunnel
start a tunnel for use with a tunnel-group backend
update ngrok to the latest version
print the version string
ngrok http foo.dev:80          # tunnel to host:port in
                               # expose a local https server
ngrok http https://localhost    # tunnel arbitrary TCP traffic to port 22
ngrok tcp 22                   # TLS traffic for foo.com to port 443
ngrok tls --domain=foo.com 443  # start tunnels from the configuration file

COMMANDS:
api                                use ngrok agent as an api client
completion                         generates shell completion code for bash or zsh
C:\Users\Nuntanut\Downloads\ngrok-v3-stable-windows-amd64>ngrok config add-authtoken 2QrhK0cR
Bqxp05FUCndIIVWRH_2tPrIgXsu14nvCA27MK5a
Authtoken saved to configuration file: C:\Users\Nuntanut\AppData\Local\ngrok/ngrok.yml

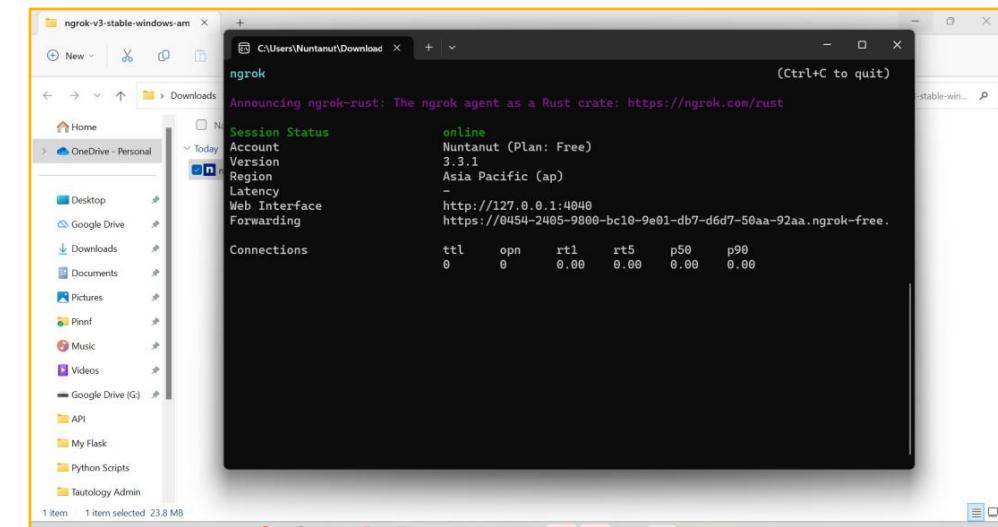
C:\Users\Nuntanut\Downloads\ngrok-v3-stable-windows-amd64>ngrok.exe http 5000
```

First Flask application

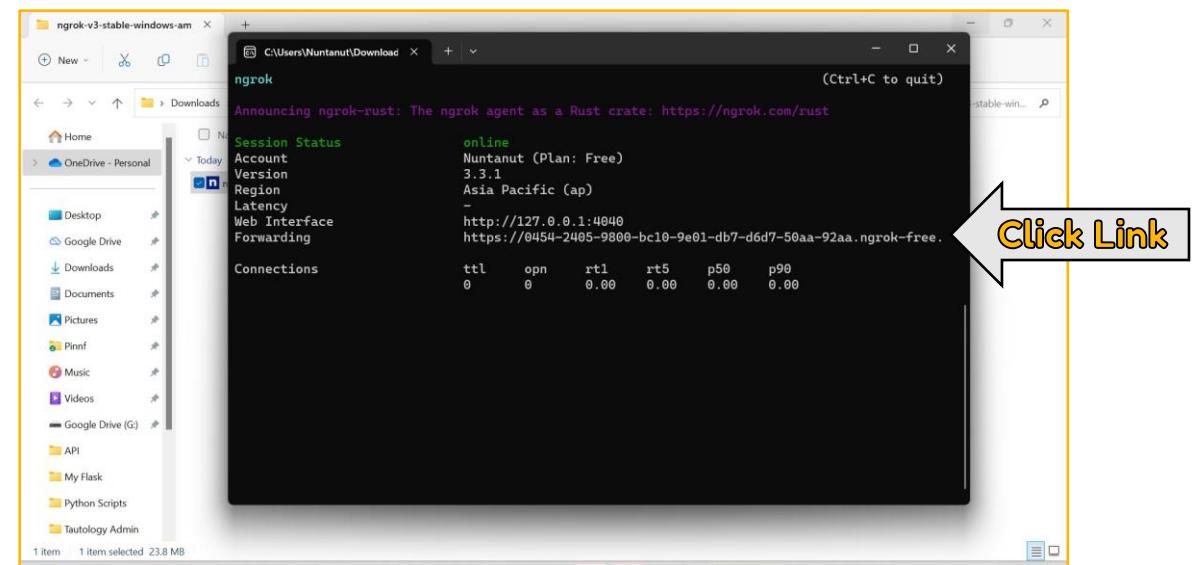
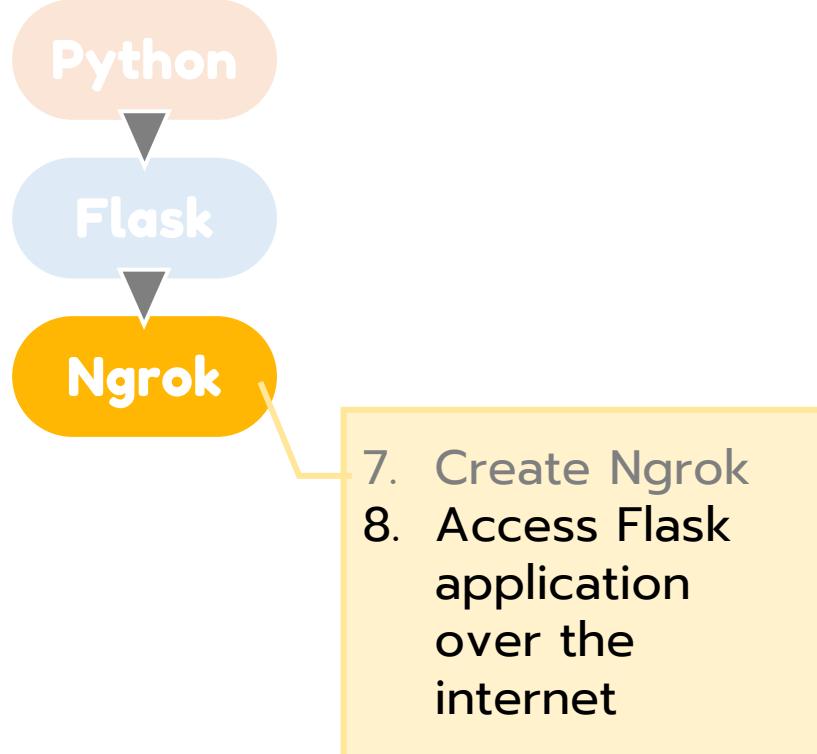


4

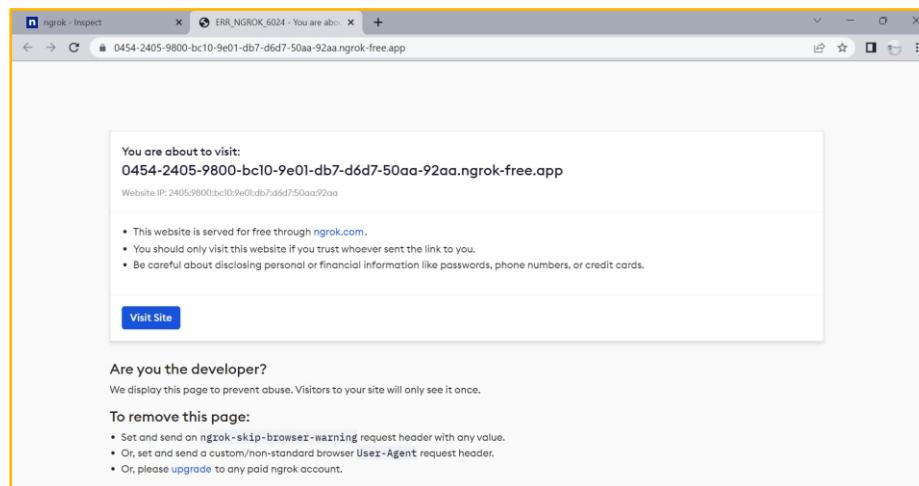
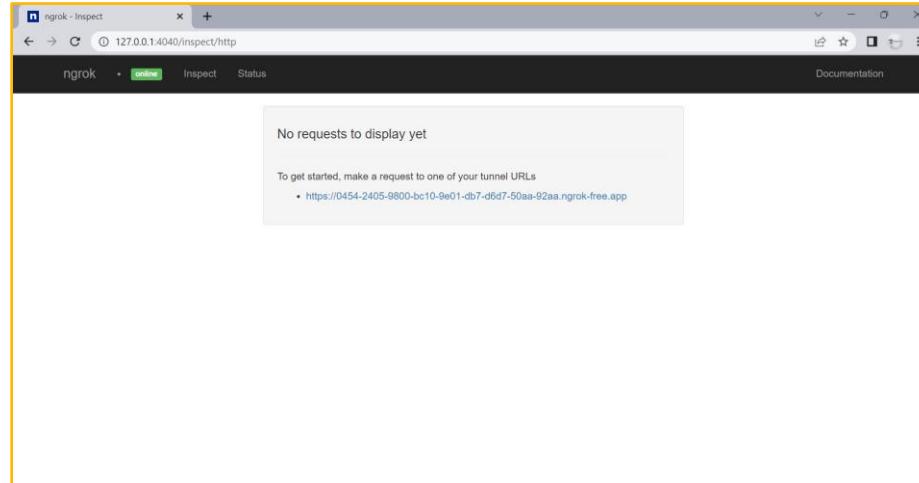
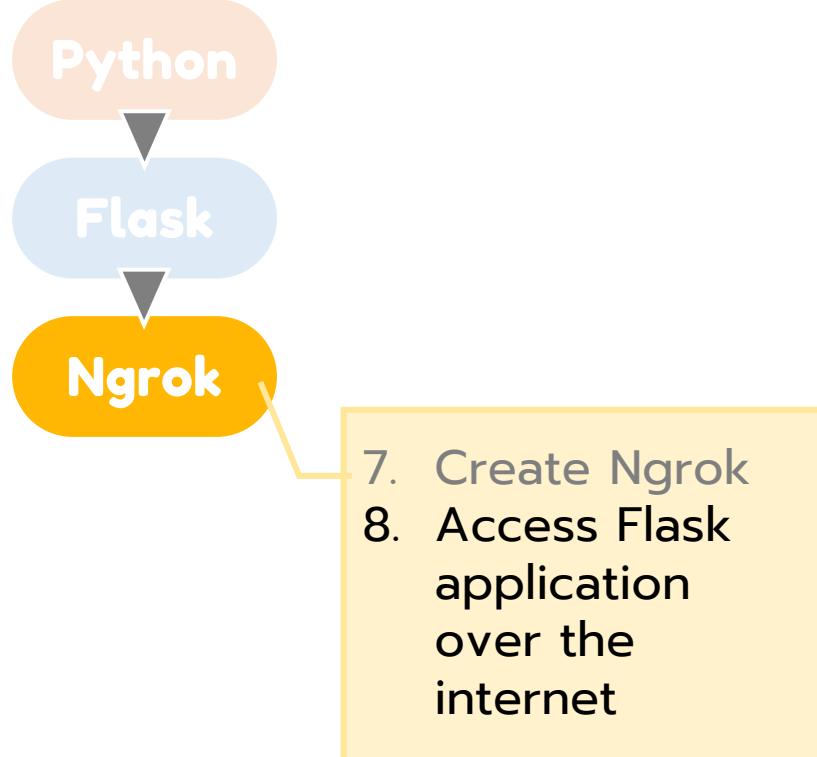
Run

`ngrok.exe http 5000`

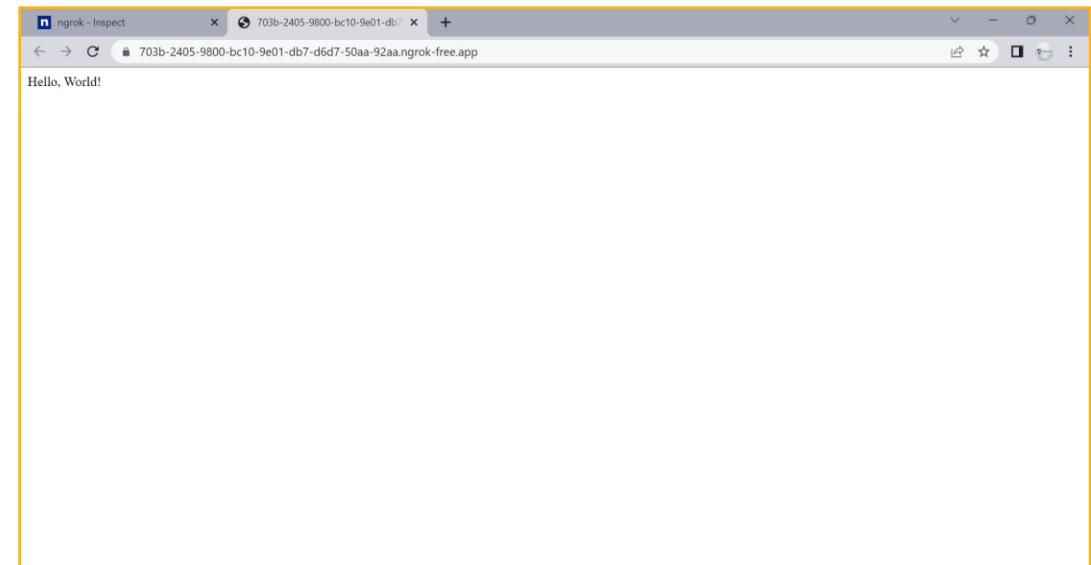
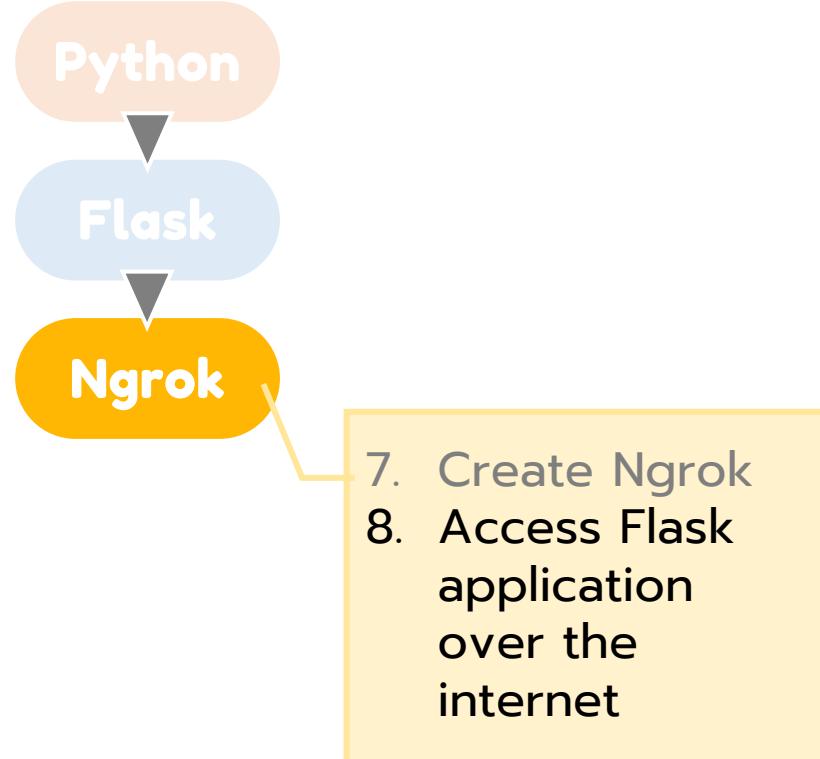
First Flask application



First Flask application



First Flask application



Build your first REST-API

- API Structure**
- Prerequisite**
- How to install Flask**
- HTTP Method
- Basic Authentication



HTTP Method

HTTP method ที่มักจะถูกเรียกใช้ง่ายบ่อยที่สุด คือ GET และ POST ซึ่งกังสอง Method มีหลักการทำงานเหมือนกัน แต่ว่าแตกต่างกันในส่วนของการใช้งาน

GET

POST

HTTP Method

- GET
- POST
- Comparison
- Test API



GET

GET : Method ที่ใช้เรียกข้อมูลใน API Server ไม่สามารถส่งข้อมูลที่อยู่ในส่วนของ body ถ้าต้องการส่งข้อมูลต้องส่งในรูปแบบของ parameter ที่อยู่ในส่วน URI/URL โดยวิธีการใช้งาน **GET** มีดังนี้

1. Import necessary modules
2. Add a GET Method to Receive URL parameters
3. Add a GET Method with Search Logic

GET

1. Import necessary modules

```
from flask import Flask, request
```

GET

2. Add a GET Method to Receive URL parameters

```
@app.route("/get1", methods=["GET"])
def read_get_data():
    name = request.args.get("name")
    return "your input name is " + name
```

GET

3. Add a GET Method with Search Logic

```
@app.route("/get2", methods=["GET"])
def find_teacher_age_get():
    data = [
        {
            "teacher": [
                {
                    "name": "pree",
                    "age": 27
                },
                {
                    "name": "krin",
                    "age": 28
                }
            ]
        }
    ]
    name = request.args.get("name")

    age = "not found"

    for teacher in data["teacher"]:
        if teacher["name"] == name:
            age = str(teacher["age"])

    return "your teacher age is " + age
```

HTTP Method

✓ **GET**

- POST
- Comparison
- Test API



Post

POST : Method ที่ใช้ส่งข้อมูลที่มาพร้อม Request ไปยัง API Server จากนั้นดำเนินการตาม Request โดยวิธีการใช้งาน โดยวิธีการใช้งาน **POST** มีดังนี้

1. Import necessary modules
2. Add a POST Method to Receive JSON request body
3. Add a POST Method with Search Logic

POST

1. Import necessary modules

```
from flask import Flask, request  
import json
```

POST

2. Add a POST Method to Receive JSON request body

```
@app.route("/post1", methods=["POST"])
def read_post_data():
    payload = request.json
    return "your input data is " + json.dumps(payload)
```

POST

3. Add a POST Method with Search Logic

```
@app.route("/post2", methods=["POST"])
def find_teacher_age_post():
    data = [
        "teacher": [
            {
                "name": "pree",
                "age": 27
            },
            {
                "name": "krin",
                "age": 28
            }
        ]
    }

    payload = request.json
    name = payload["search_name"]
    age = "not found"
    for teacher in data["teacher"]:
        if teacher["name"] == name:
            age = str(teacher["age"])

    return "your teacher age is " + age
```

HTTP Method

- ✓ **GET**
- ✓ **POST**
- Comparison
- Test API



Comparison

	GET	POST
URL	/	/
Header	/	/
Body	✗	/
วิธีการส่ง Parameter	ส่งผ่าน URL	ส่งผ่าน JSON
Key Point	ใช้รับ parameter ในมีความซับซ้อน	ใช้รับ parameter ที่มีความซับซ้อน

HTTP Method

- ✓ **GET**
- ✓ **POST**
- ✓ **Comparison**
- Test API



Test API

โดยส่วนใหญ่ เราสามารถใช้งานผ่าน Jupyter Notebook และ Postman



Jupyter Notebook

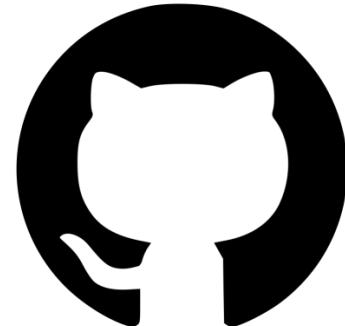


Postman

Test API



Install Jupyter Notebook

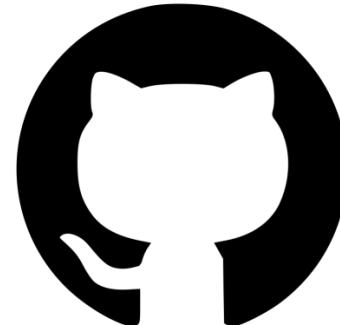


[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(4\)test_api_with_jupyter.md#1-installing-jupyter-notebook](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(4)test_api_with_jupyter.md#1-installing-jupyter-notebook)

Test API



Run API in Jupyter Notebook

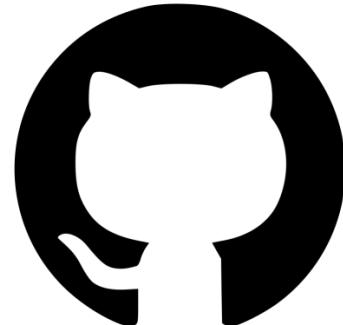


[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(4\)test_api_with_jupyter.md#4-testing-with-jupyter](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(4)test_api_with_jupyter.md#4-testing-with-jupyter)

Test API



Install Postman

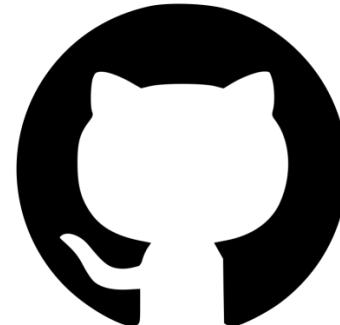


[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(3\)test_api_with_postman.md#1-install-postman](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(3)test_api_with_postman.md#1-install-postman)

Test API



Run API in Postman



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(3\)test_api_with_postman.md#2-using-postman-to-test-flask-endpoints](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(3)test_api_with_postman.md#2-using-postman-to-test-flask-endpoints)

HTTP Method

- ✓ **GET**
- ✓ **POST**
- ✓ **Comparison**
- ✓ **Test API**



Build your first REST-API

- API Structure**
- Prerequisite**
- How to install Flask**
- HTTP Method**
- Basic Authentication



Basic Authentication

เพื่อความปลอดภัยในการเข้าถึง API เราสามารถทำได้หลายวิธีการ เช่น Basic Authentication, Token-Based Authentication, OAuth 2.0, API Key ซึ่งสำหรับ Basic Authentication เราจะใช้ Password ในการเข้าถึงข้อมูลใน API



Basic Authentication

สำหรับการใช้รหัสผ่าน เราจะสามารถดำเนินการได้ผ่าน method GET และ POST

GET

```
password = "12345678" # Global password variable

@app.route("/get3", methods=["GET"])
def get_with_auth():
    req_password = request.headers.get("password")

    if req_password == password:
        name = request.args.get("name")
        return "your input name is " + name
    else:
        return "invalid password", 401 # Return HTTP 401 Unauthorized status code
```

Basic Authentication

สำหรับการใช้รหัสผ่าน เราจะสามารถดำเนินการได้ผ่าน method GET และ POST

POST

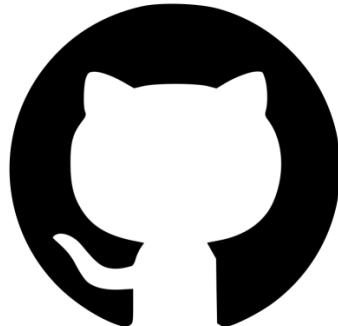
```
@app.route("/post3", methods=["POST"])
def post_with_auth():
    req_password = request.headers.get("password")

    if req_password == password:
        payload = request.json
        return "your input data is " + json.dumps(payload)
    else:
        return "invalid password", 401 # Return HTTP 401 Unauthorized status code
```

Basic Authentication



Run API with Authentication in Jupyter Notebook

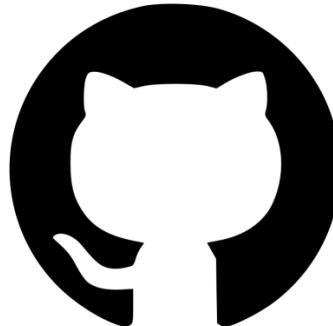


[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(6\)test_api_with_auth.md#2-jupyter-notebook-testing](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(6)test_api_with_auth.md#2-jupyter-notebook-testing)

Basic Authentication



Run API with Authentication in Postman



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/\(6\)test_api_with_auth.md#1-postman-testing](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/basic_api/basic_api_doc/(6)test_api_with_auth.md#1-postman-testing)

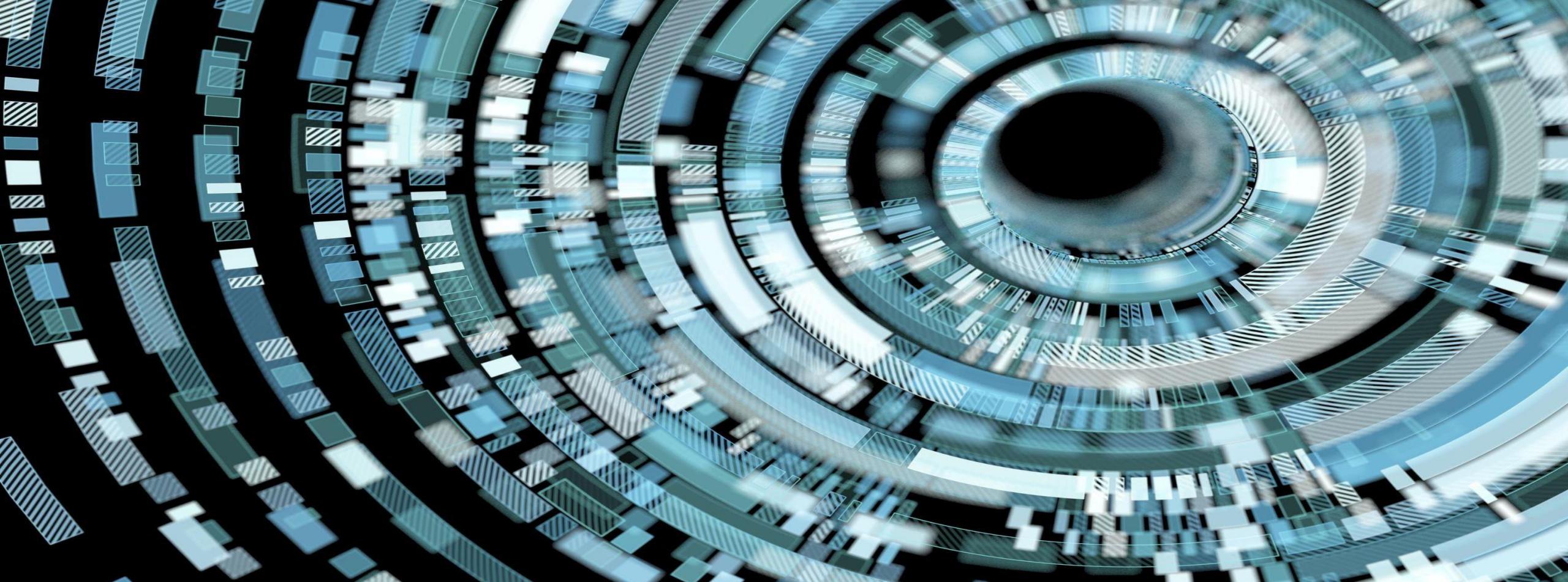
Build your first REST-API

- API Structure**
- Prerequisite**
- How to install Flask**
- HTTP Method**
- Basic Authentication**



API & Real-World Project

- ✓ Introduction to API
- ✓ Build your first REST-API
 - ChatGPT API
 - Facebook Messenger Chatbot
 - Facebook + ChatGPT API



ChatGPT API

ChatGPT API

ChatGPT API คือ API กี่สามารถเรียกได้จาก ChatGPT ของ OpenAI

```
1 import openai
2 openai.api_key = "<your_open_api_key>"
3
4 openai.ChatCompletion.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {"role": "user", "content": "Who won the football world cup in 2014?"},
8         {"role": "assistant", "content": "Germany won the football world cup in 2014."},
9         {"role": "user", "content": "Where was it played?"},
10    ],
11    temperature=1,
12    max_tokens=256,
13    top_p=1,
14    frequency_penalty=0,
15    presence_penalty=0,
16 )
```

ChatGPT API

- OpenAI API Guide
- ChatGPT API Method
- Error Handling



OpenAI API Guide

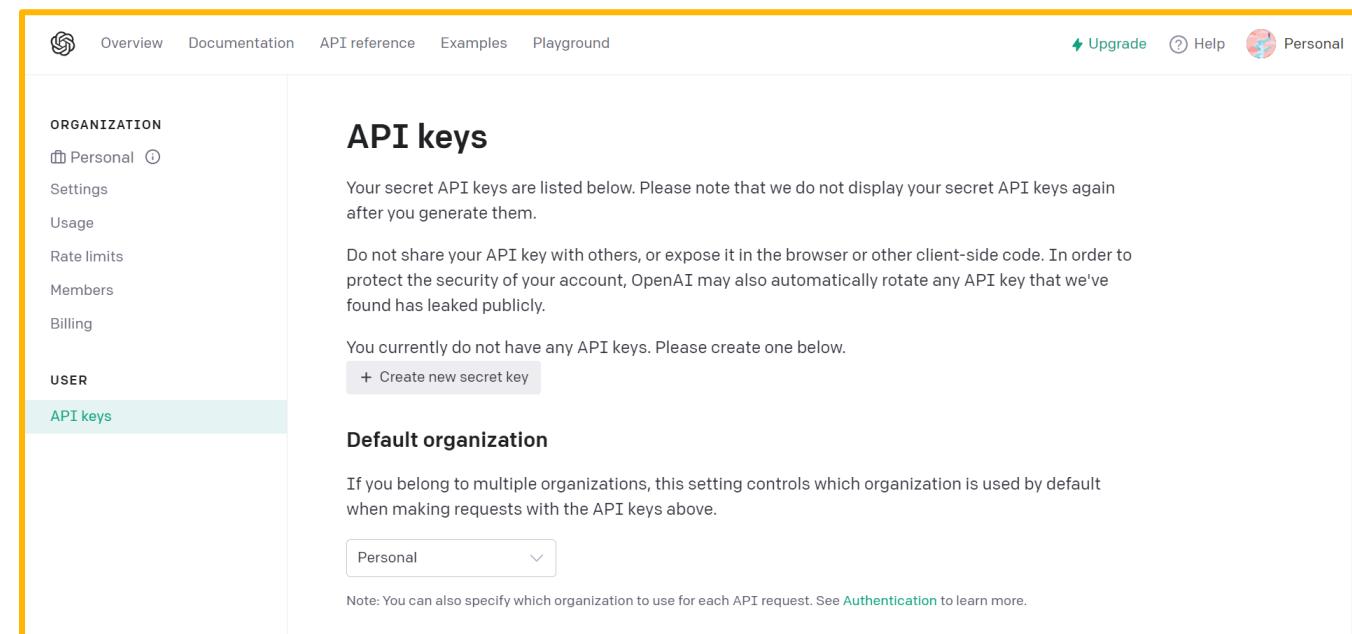
- Generate API Key
- Monitoring API Usage
- Rate Limits
- Pricing



Generate API Key

ขั้นตอนสำหรับ generate API key สำหรับ OpenAI

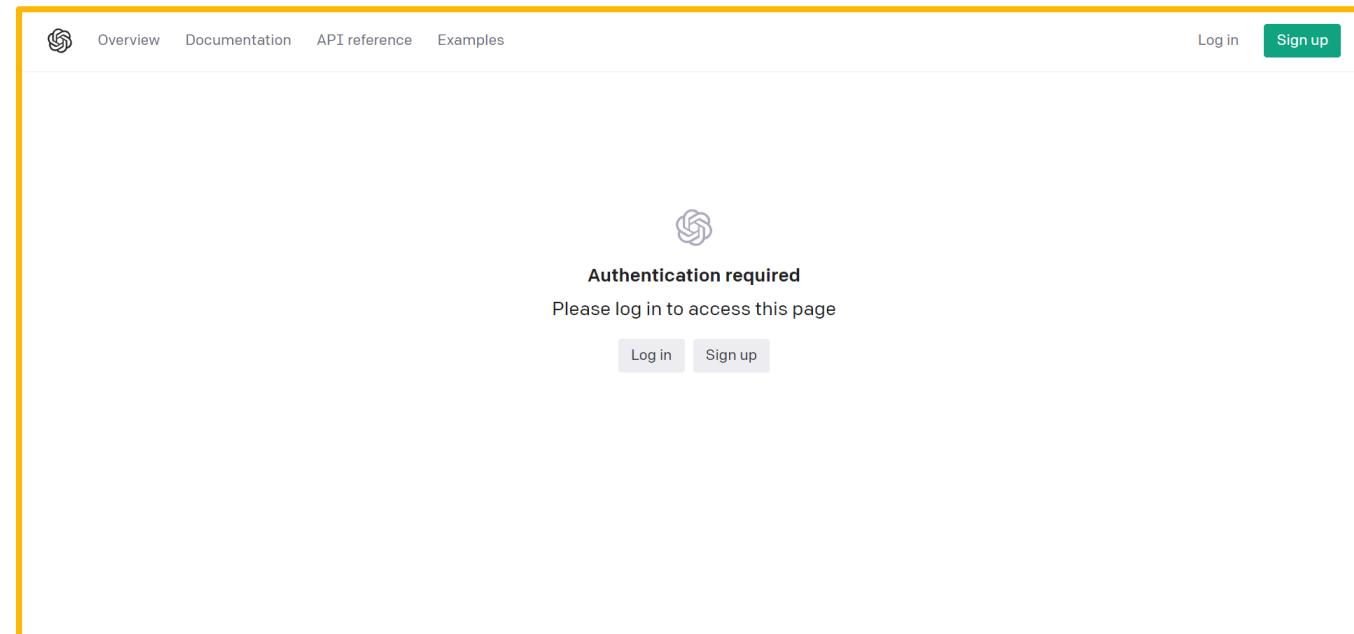
1. ไปที่ <https://platform.openai.com/account/api-keys>



Generate API Key

ขั้นตอนสำหรับ generate API key สำหรับ OpenAI

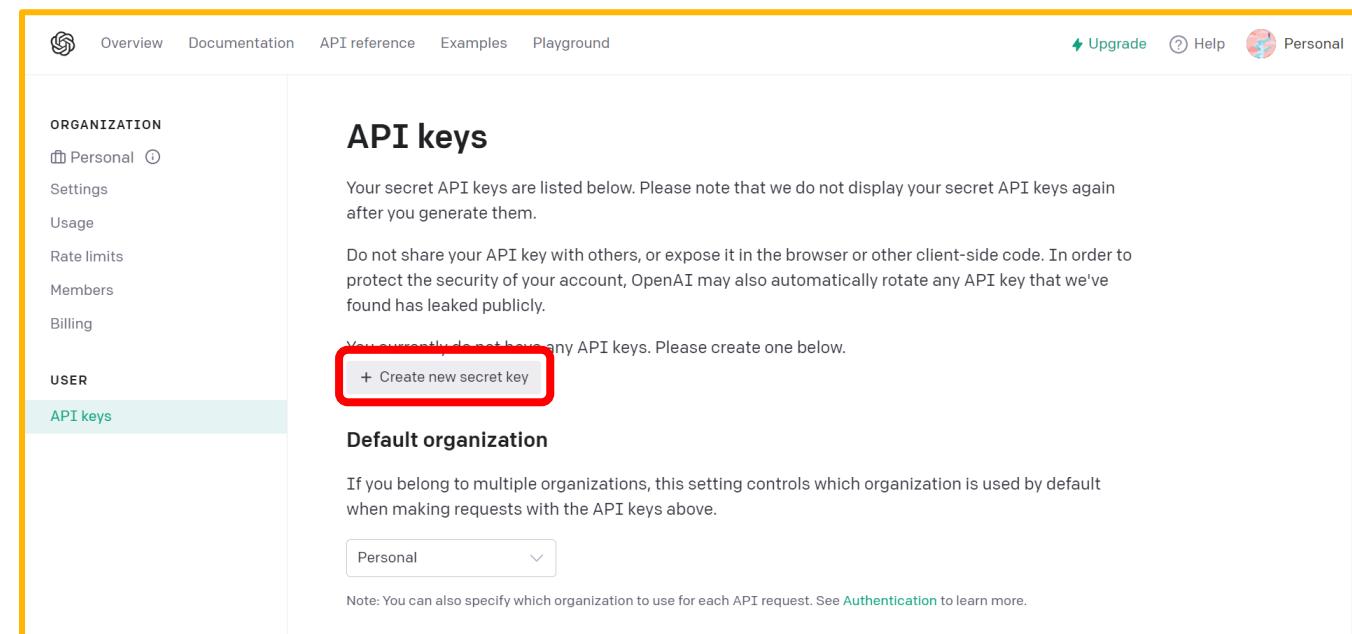
2. เข้าสู่ระบบ หรือสมัครสมาชิก กรณีที่กำบังไม่ได้เป็นสมาชิก



Generate API Key

ขั้นตอนสำหรับ generate API key สำหรับ OpenAI

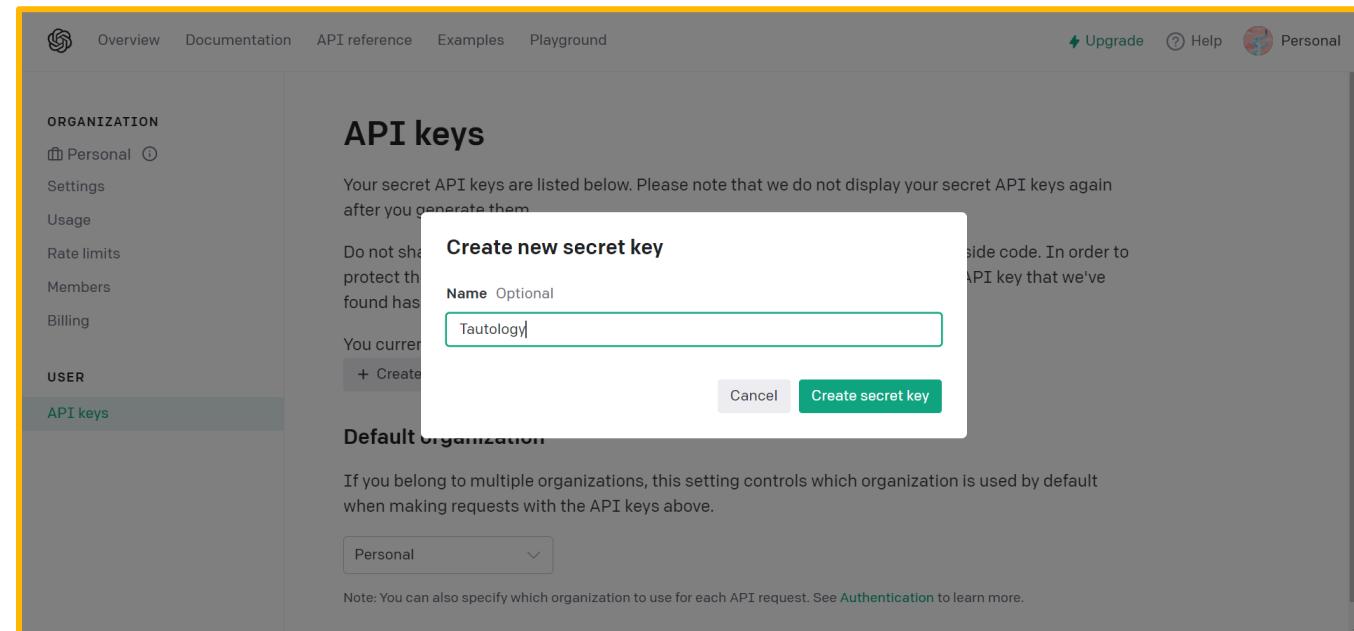
- คลิกที่ "Create a new API key".



Generate API Key

ขั้นตอนสำหรับ generate API key สำหรับ OpenAI

4. ระบุชื่อของ key จากนั้นคลิกที่ "Create".



OpenAI API Guide

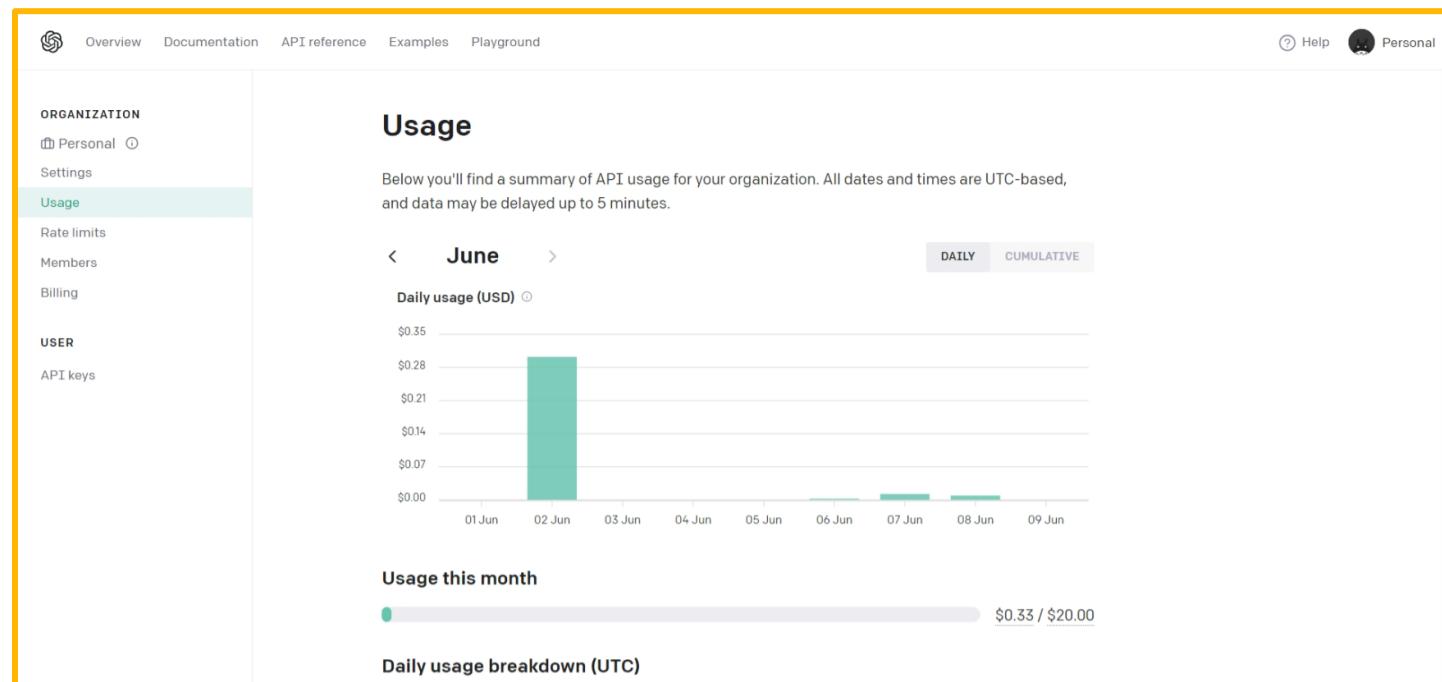
✓ Generate API Key

- Monitoring API Usage
- Rate Limits
- Pricing



Monitoring API Usage

ตรวจสอบการใช้งาน API เพื่อหลีกเลี่ยงการใช้อัตราเกินขีดจำกัด โดยไปที่ <https://platform.openai.com/account/usage>



OpenAI API Guide

- ✓ **Generate API Key**
- ✓ **Monitoring API Usage**
 - Rate Limits
 - Pricing



Rate Limits

Rate limit คือ ข้อจำกัดที่ ChatGPT API ถูกกำหนดจำนวนครั้งที่ผู้ใช้งานสามารถเข้าถึง server ในระยะเวลาที่กำหนด

Plan	Text & Embedding	Chat	Codex	Edit	Image	Audio
Free trial	3 RPM 150,000 TPM	3 RPM 40,000 TPM	3 RPM 40,000 TPM	3 RPM 150,000 TPM	5 images / min	3 RPM
Pay-as-you-go (first 48 hrs.)	60 RPM 250,000 TPM	60 RPM 60,000 TPM	20 RPM 40,000 TPM	20 RPM 150,000 TPM	50 images / min	50 RPM
Pay-as-you-go (after 48 hrs.)	3,500 RPM 350,000 TPM	3,500 RPM 90,000 TPM	20 RPM 40,000 TPM	20 RPM 150,000 TPM	50 images / min	50 RPM

Rate Limits

Rate limit จะถูกวัดจาก

1. **RPM (Requests Per Minute)** : จำนวน Request สูงสุดที่ส่งได้ต่อนาที
2. **TPM (Tokens Per Minute)** : จำนวน Token สูงสุดที่สร้างได้ต่อนาที โดยจำนวน Token กึ่งหนด คำนวณได้จาก Input Token (text ที่ input เข้าไป) และ Output Token (text ที่ตอบกลับมา)

OpenAI API Guide

- ✓ **Generate API Key**
- ✓ **Monitoring API Usage**
- ✓ **Rate Limits**
- Pricing



Pricing

ในการส่ง Request และการตอบกลับ Respond จะมีการคิดราคา Token โดยจะขึ้นกับ Model ของ ChatGPT

Model	Context Size	Prompt Price	Completion Price
GPT-4	8K	\$0.03 / 1K tokens	\$0.06 / 1K tokens
GPT-4	32K	\$0.06 / 1K tokens	\$0.12 / 1K tokens
gpt-3.5-turbo		\$0.002 / 1K tokens	
Ada		\$0.0004 / 1K tokens	
Babbage		\$0.0005 / 1K tokens	
Curie		\$0.0020 / 1K tokens	
Davinci		\$0.0200 / 1K tokens	

Pricing

การประมาณจำนวน Token ของข้อความ สามารถเข้าไปนับได้ที่
<https://platform.openai.com/tokenizer>



OpenAI API Guide

- ✓ **Generate API Key**
- ✓ **Monitoring API Usage**
- ✓ **Rate Limits**
- ✓ **Pricing**



ChatGPT API

- OpenAI API Guide**
- ChatGPT API Method
- Error Handling



ChatGPT API Method

สำหรับ ChatGPT API จะมี Method/ Mode ที่ถูกใช้งานบ่อย ดังนี้

1. **Completion** : การสร้างข้อความที่มีความสอดคล้องตามบริบทของข้อความที่ถูกส่งเข้ามา
2. **Chat** : การถาม-ตอบ/ การสนทนา
3. **Edit** : การปรับแต่ง หรือแก้ไขข้อความที่ถูกส่งเข้ามา

ChatGPT API Method

- Completion
- Chat
- Edit



Completion

Completion : การสร้างข้อความที่มีความสอดคล้องตามบริบทของข้อความที่ถูกส่งเข้ามา โดย ChatGPT API สำหรับ Completion จะสามารถแบ่งเป็น 2 ส่วนดังนี้

- Request

- Respond

- Ending Handling

Completion

- Request

Request สำหรับ Completion จะประกอบด้วย Parameter หลากหลายตัว เช่น

model, prompt, suffix, temperature, n, max_token

```
1 import openai
2 openai.api_key = "<your_open_api_key>"
3
4 openai.Completion.create(
5     model="text-davinci-003",
6     prompt="Say this is a test",
7     temperature=1,
8     max_tokens=256,
9     top_p=1,
10    frequency_penalty=0,
11    presence_penalty=0,
12 )
```

Completion

◦ Request

`model` (string, **Required**) : ID ของ model ที่เราต้องการจะใช้ ซึ่ง model ที่ถูก support ใน Completion ของ ChatGPT API ได้แก่

- `text-davinci-003`
- `text-davinci-002`
- `text-curie-001`
- `text-babbage-001`
- `text-ada-001`

Completion

- Request

`prompt` (string/array, **Optional**, Defaults to null) : ข้อความที่ส่งไป

`suffix` (string, **Optional**, Defaults to null) : ข้อความต่อท้าย inserted text

`max_token` (integer, **Optional**, Defaults to 16) : จำนวน token สูงสุดของ Respond

`n` (integer, **Optional**, Defaults to 1) : จำนวน respond ที่ตอบกลับมา

Completion

- Respond

Respond สำหรับ Completion จะประกอบด้วย Parameter หลากหลายตัว เช่น

`id, object, create, model, choice, usage`

```
1   {
2     "id": "cmpl-uqkv1QyYK7bGYrRHQ0eXlWi7",
3     "object": "text_completion",
4     "created": 1589478378,
5     "model": "text-davinci-003",
6     "choices": [
7       {
8         "text": "\n\nThis is indeed a test",
9         "index": 0,
10        "logprobs": null,
11        "finish_reason": "length"
12      }
13    ],
14    "usage": {
15      "prompt_tokens": 5,
16      "completion_tokens": 7,
17      "total_tokens": 12
18    }
19 }
```

Completion

◦ Respond

`choices` (array) : ส่วนของคำตอบสำหรับ Completion ซึ่งประกอบไปด้วย

- `text` (string) : ข้อความที่เป็นคำตอบของ Completion
- `index` (integer) : index ของคำตอบ เพื่อให้เข้าใจได้ง่ายในกรณีที่มีหลายคำตอบ (`n > 1`)
- `logprobs` (object or null) : ค่า Log Probability ของ Token ที่ถูก Generate ซึ่งจะแสดงค่าเมื่อเรา Set ค่าใน Request แต่ถ้าไม่ได้ Set ไว้ จะเป็น “null”
- `finish_reasons` (string) : ระบุการจบของคำตอบ ซึ่งมี 2 รูปแบบ
 - “stop” : คำตอบครบสมบูรณ์
 - “length” : คำตอบยังไม่ครบสมบูรณ์ เนื่องจากจำนวน `max_token` ไม่พอสำหรับคำตอบ

Completion

◦ Ending Handling

ในการนี้ที่ “`finish_reasons`” = “`length`” หรือข้อความที่ต้องกลับยังไม่ครบ
สมบูรณ์ ที่เกิดจาก `max_token` ไม่พอต่อข้อความทั้งหมดที่ต้องกลับ
เราจะมีวิธีจัดการ คือ นำข้อความเก่ามาต่อ กับ ข้อความใหม่ที่ต้องกลับมา จากนั้นส่ง
Request ข้อความที่นำมารวมกันแล้วอีกครั้ง เพื่อให้ต้องกลับจากการต่อข้อความที่
ส่งไป

ChatGPT API Method

✓ Completion

- Chat
- Edit



Chat

Chat : การถาม-ตอบ/ การสนทนา โดย ChatGPT API สำหรับ Chat จะสามารถแบ่งเป็น 2 ส่วน ดังนี้

- Request

- Respond

- Ending Handling

Chat

- Request

Request สำหรับ Chat จะประกอบด้วย Parameter หลากหลายตัว เช่น `model`, `messages`, `temperature`, `n`, `max_token`

```
1 import openai
2 openai.api_key = "<your_open_api_key>"
3
4 openai.ChatCompletion.create(
5     model="gpt-3.5-turbo",
6     messages=[
7         {"role": "user", "content": "Who won the football world cup in 2014?"},
8         {"role": "assistant", "content": "Germany won the football world cup in 2014."},
9         {"role": "user", "content": "Where was it played?"},
10    ],
11    temperature=1,
12    max_tokens=256,
13    top_p=1,
14    frequency_penalty=0,
15    presence_penalty=0,
16 )
```

Chat

◦ Request

`model` (string, **Required**) : ID ของ model ที่เราต้องการจะใช้ ซึ่ง model ที่ถูก support ใน Completion ของ ChatGPT API ได้แก่

- `text-davinci-003`
- `text-davinci-002`
- `text-curie-001`
- `text-babbage-001`
- `text-ada-001`

Chat

◦ Request

messages (string, Required) : บทบาท และข้อความในบทสนทนา ซึ่งจะประกอบไปด้วย

- “role” (string, Required) : บทบาทของผู้สนทนา เช่น “system”, “user”, “assistant”
- “content” (string, Required) : ข้อความที่ต้องการส่ง
- “name” (string, Optional) : ชื่อของผู้เขียนข้อความ

Chat

- Request

max_token (integer, Optional, Defaults to 16) : จำนวน token สูงสุดของ Respond

n (integer, Optional, Defaults to 1) : จำนวน respond ที่ตอบกลับมา

Chat

- Respond

Respond สำหรับ Chat จะประกอบด้วย Parameter หลากหลายตัว เช่น `id`, `object`, `create`, `model`, `choice`, `usage`

```
1  {
2      "id": "chatcmpl-123",
3      "object": "chat.completion",
4      "created": 1677652288,
5      "choices": [
6          {
7              "index": 0,
8              "message": {
9                  "role": "assistant",
10                 "content": "The 2014 FIFA World Cup was held in Brazil."
11             },
12             "finish_reason": "stop"
13         }
14     ],
15     "usage": {
16         "prompt_tokens": 9,
17         "completion_tokens": 12,
18         "total_tokens": 21
19     }
20 }
```

Chat

◦ Respond

`choices` (array) : ส่วนของคำตอบสำหรับ Chat ซึ่งประกอบไปด้วย

- `index` (integer) : index ของคำตอบ เพื่อให้เข้าใจได้ง่ายในกรณีที่มีหลายคำตอบ (`n` > 1)
- `message` (object) : บทบาท (role) และข้อความ (content) ในบทสนทนา
- `finish_reasons` (string) : ระบุการจบของคำตอบ ซึ่งมี 2 รูปแบบ
 - “stop” : คำตอบครบสมบูรณ์
 - “length” : คำตอบยังไม่ครบสมบูรณ์ เนื่องจากจำนวน `max_token` ไม่พอสำหรับคำตอบ

Completion

◦ Ending Handling

ในการนี้ที่ “`finish_reasons`” = “`length`” หรือข้อความที่ต้องกลับยังไม่ครบ
สมบูรณ์ ที่เกิดจาก `max_token` ไม่พอต่อข้อความทั้งหมดที่ต้องกลับ
เราจะมีวิธีจัดการ คือ นำข้อความเก่ามาต่อ กับ ข้อความใหม่ที่ต้องกลับมา จากนั้นส่ง
Request ข้อความที่นำมารวมกันแล้วอีกครั้ง เพื่อให้ต้องกลับจากการต่อข้อความที่
ส่งไป

ChatGPT API Method

- ✓ **Completion**
- ✓ **Chat**
- **Edit**



Edit

Edit : การปรับแต่ง หรือแก้ไขข้อความที่ถูกส่งเข้ามา โดย ChatGPT API สำหรับ Edit จะสามารถแบ่งเป็น 2 ส่วน ดังนี้

- Request

- Respond

Edit

- Request

Request สำหรับ Edit จะประกอบด้วย Parameter หลากหลายตัว เช่น `model`, `input`, `instruction`, `max_token`, `n`

```
1 import openai
2 openai.api_key = "<your_open_api_key>"
3
4 openai.Edit.create(
5     model="text-davinci-edit-001",
6     input="What day of the wek is it?",
7     instruction="Fix the spelling mistakes"
8 )
```

Edit

- Request

`model` (string, **Required**) : ID ของ model ที่เราต้องการจะใช้ ซึ่ง model ที่ถูก support ใน Completion ของ ChatGPT API ได้แก่

- `text-davinci-003`
- `text-davinci-002`
- `text-curie-001`
- `text-babbage-001`
- `text-ada-001`

Edit

◦ Request

`input` (string, Optional, Defaults to '') : ข้อความที่ส่งไปเพื่อให้ model แก้ไข

`instruction` (string, Required) : คำสั่งให้ model ปรับปรุง หรือแก้ไขข้อความที่ส่งไป

`max_token` (integer, Optional, Defaults to 16) : จำนวน token สูงสุดของ Respond

`n` (integer, Optional, Defaults to 1) : จำนวน respond ที่ต้องกลับมา

Edit

- Respond

Respond สำหรับ Edit จะประกอบด้วย Parameter หลากหลายตัว เช่น `object`, `create`, `choice`, `usage`

```
1  {
2      "object": "edit",
3      "created": 1589478378,
4      "choices": [
5          {
6              "text": "What day of the week is it?",
7              "index": 0,
8          }
9      ],
10     "usage": {
11         "prompt_tokens": 25,
12         "completion_tokens": 32,
13         "total_tokens": 57
14     }
15 }
```

Edit

- Respond

`choices (array)` : ส่วนของคำตอบสำหรับ Edit ซึ่งประกอบไปด้วย

- `index (integer)` : index ของคำตอบ เพื่อให้เข้าใจได้ง่ายในกรณีที่มีหลายคำตอบ (`n > 1`)
- `text (object)` : ข้อความที่ถูกปรับแต่ง หรือแก้ไขแล้ว

ChatGPT API Method

- ✓ **Completion**
- ✓ **Chat**
- ✓ **Edit**



ChatGPT API

- OpenAI API Guide**
- ChatGPT API Method**
- Error Handling



Error Handling

- o Common Error
- o Best Practices
- o OpenAI Error Codes Reference



Common Error

- **Invalid API Key Error**

```
1 import openai
2
3 openai.api_key = "invalid open api key"
4 try:
5     response = openai.Completion.create(prompt="Hello world", model="text-davinci-003")
6 except Exception as e:
7     print(f"error type is {type(e)}")
```

Common Error

- **Invalid Arguments**

```
1 import openai
2
3 openai.api_key = "<your_open_api_key>"
4 try:
5     response = openai.Completion.create(prompt="Hello world", model="text-davinci-004")
6 except Exception as e:
7     print(f"error type is {type(e)}")
```

Common Error

- **Rate Limit Exceeded**

```
1 import openai
2
3 openai.api_key = "<your_open_api_key>"
4 try:
5     for i in range(10000):
6         response = openai.Completion.create(
7             prompt="Hello world", model="text-davinci-003"
8         )
9     except Exception as e:
10        print(f"error type is {type(e)}")
```

Error Handling

- ✓ **Common Error**
 - Best Practices
 - OpenAI Error Codes Reference



Best Practices

Use Try/Except Blocks



Log Errors

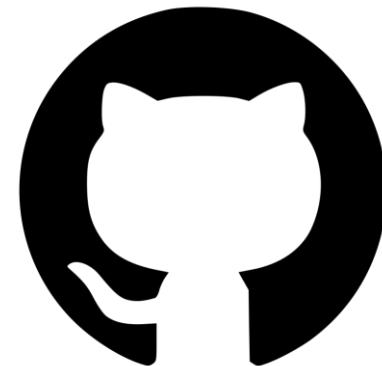


Handle Specific Errors

Best Practices



For more detailed information



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/chatgpt_api/chatgpt_api_doc/\(3\)openai_api_error_handling_doc.md#best-practices](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/chatgpt_api/chatgpt_api_doc/(3)openai_api_error_handling_doc.md#best-practices)

Error Handling

- ✓ Common Error
- ✓ Best Practices
 - OpenAI Error Codes Reference



OpenAI Error Codes Reference



**For more detailed information
about the error codes**



<https://platform.openai.com/docs/guides/error-codes/api-errors>

Error Handling

- ✓ Common Error
- ✓ Best Practices
- ✓ OpenAI Error Codes Reference



ChatGPT API

- OpenAI API Guide**
- ChatGPT API Method**
- Error Handling**



API & Real-World Project

- ✓ Introduction to API
- ✓ Build your first REST-API
- ✓ ChatGPT API
- Facebook Messenger Chatbot



Facebook Chatbot

Facebook Chatbot

- Facebook Chatbot Creation
- Validation and Authentication
- ChatGPT Integration
- Advance Usage
- Live Mode



Facebook Chatbot Creation

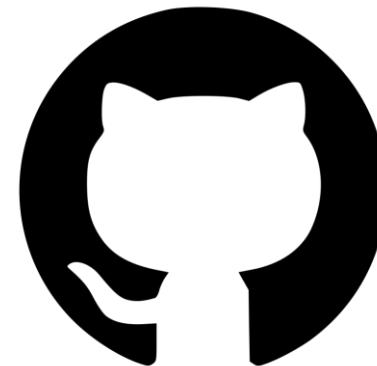
สำหรับการเปิดใช้งาน Facebook Chatbot มีขั้นตอนดังต่อไปนี้

1. Create Your First Facebook Chatbot
2. Create a Facebook Page
3. Create a Developer Account
4. Create an App
5. Generate a Page Messenger Token
6. Create a Flask App and Verify Webhook
7. Edit Page Subscriptions
8. Sending a Response Message

Facebook Chatbot Creation



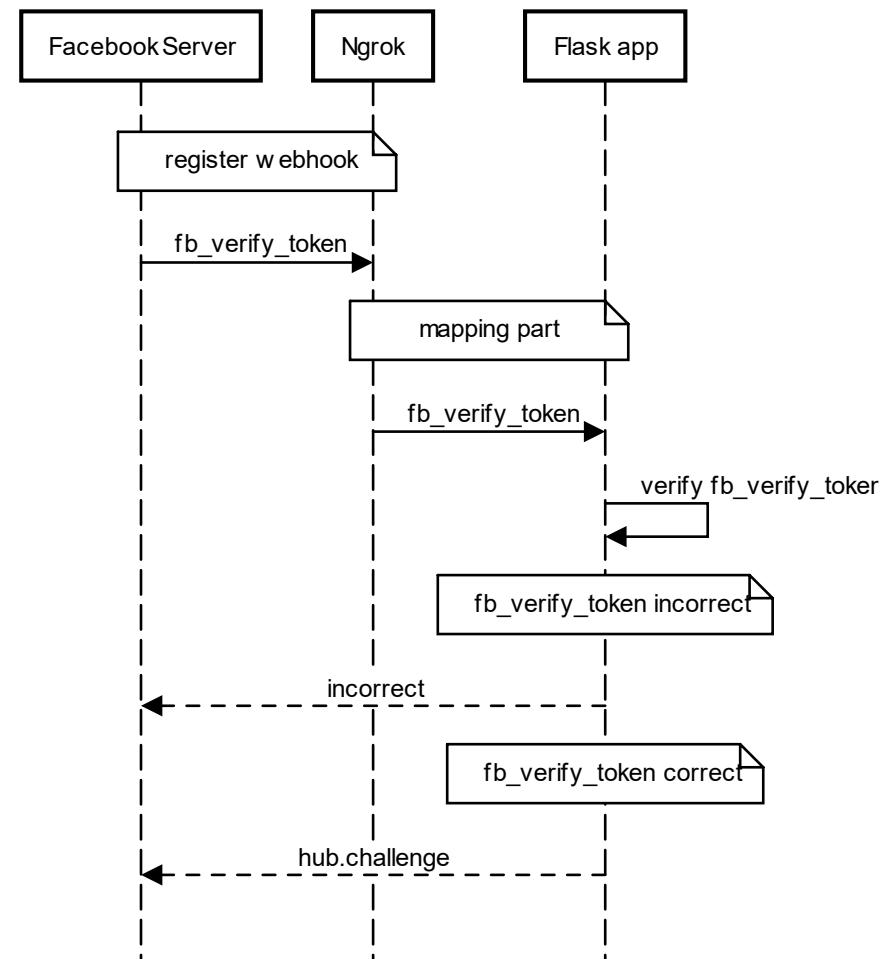
For more detailed information



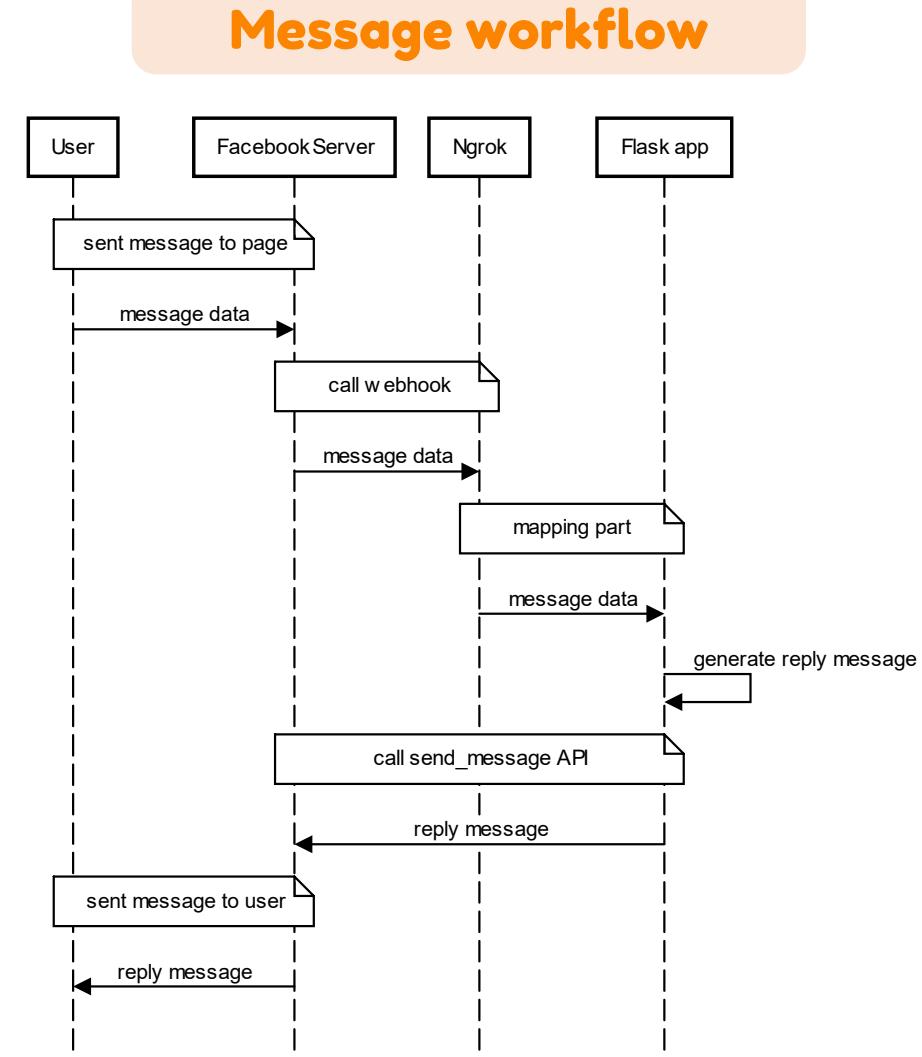
[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/\(1\)create_your_first_facebook_chatbot.md](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/(1)create_your_first_facebook_chatbot.md)

Facebook Chatbot Creation

Register webhook workflow



Facebook Chatbot Creation



Facebook Chatbot

Facebook Chatbot Creation

- Validation and Authentication
- ChatGPT Integration
- Advance Usage
- Live Mode



Validation and Authentication

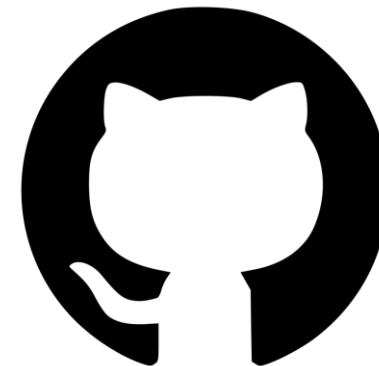
เพื่อความปลอดภัยของการเข้าถึง Facebook Chatbot สามารถใช้ 2 method ดังนี้

1. validate_request(request)
2. is_user_message(x)

Validation and Authentication



For more detailed information



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/\(2\)facebook_validation_and_authentication.md](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/(2)facebook_validation_and_authentication.md)

Facebook Chatbot

- Facebook Chatbot Creation**
- Validation and Authentication**
- ChatGPT Integration
- Advance Usage
- Live Mode



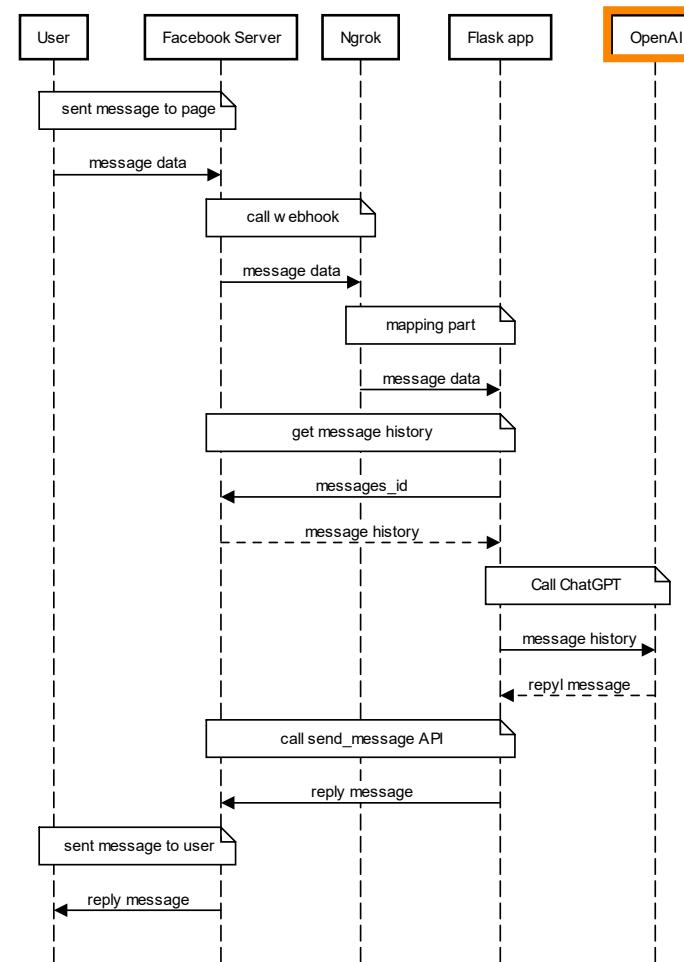
ChatGPT Integration

เพื่อปรับปรุงคุณภาพของ Facebook Chatbot เราจะนำไปรวมกับ ChatGPT ของ OpenAI



ChatGPT Integration

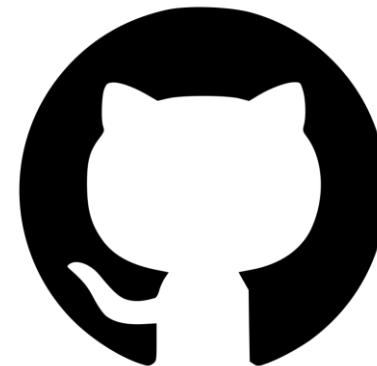
ChatGPT message workflow



ChatGPT Integration



For more detailed information



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/\(3\)chatgpt_integration.md](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/(3)chatgpt_integration.md)

Facebook Chatbot

- Facebook Chatbot Creation**
- Validation and Authentication**
- ChatGPT Integration**
- Advance Usage
- Live Mode



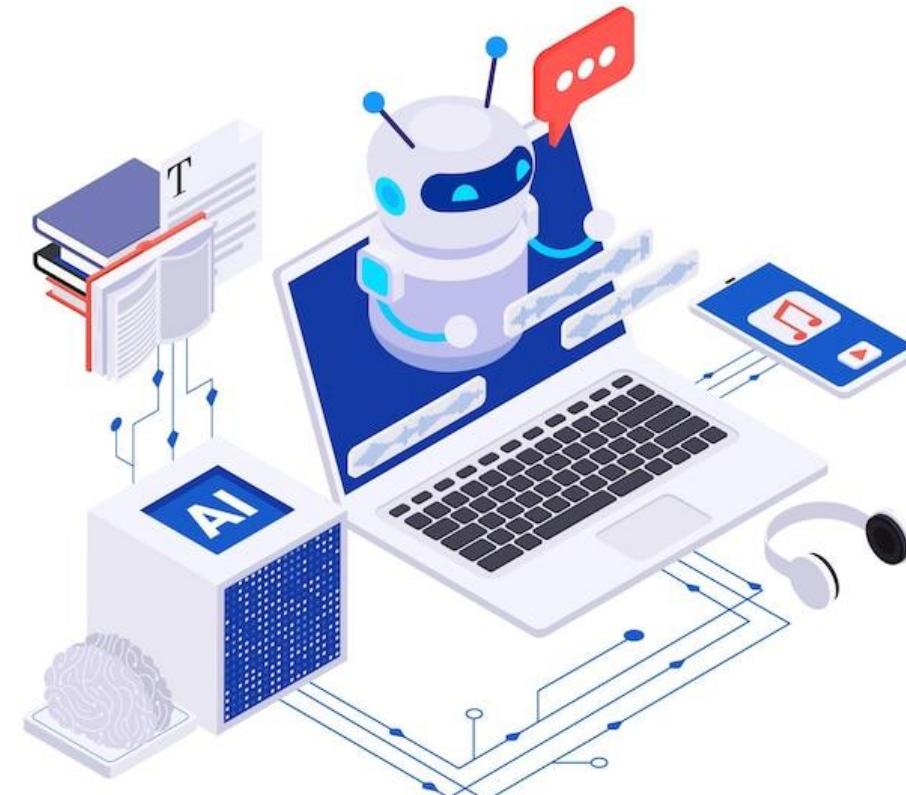
Advanced Usage

- System Role
- Preventing Duplicate Reply Messages



System Role

ในส่วนนี้เราจะทำการเพิ่มบทบาทให้กับ Facebook Chatbot เช่น ผู้ตอบเป็นครัว กัดคำตอบสำหรับบางคำ答 เป็นต้น



Advanced Usage

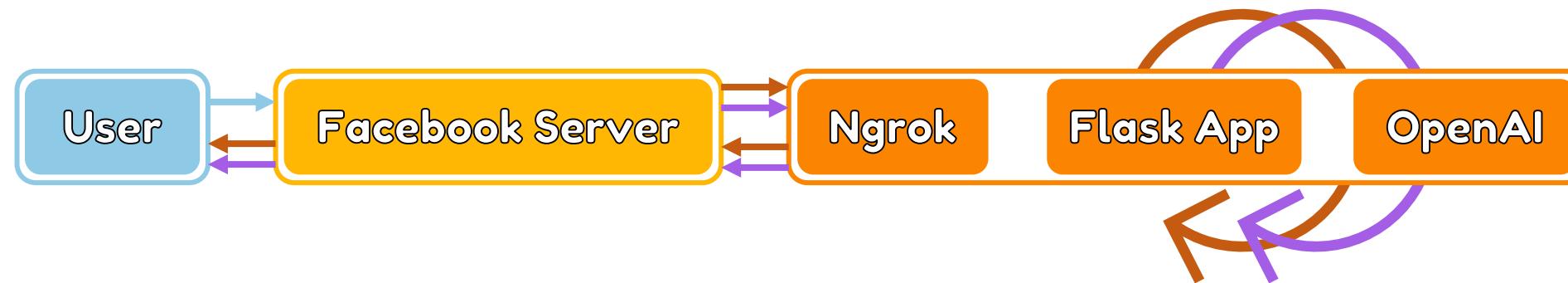
✓ System Role

- Preventing Duplicate Reply Messages



Preventing Duplicate Reply Messages

ในการส่งข้อความจาก Facebook ไปยัง Flask App ถ้า Flask App ทำการ process คำตอบนานเกินไป Facebook จะส่งข้อความใหม่ให้ Flask App อีกครั้ง ซึ่งจะทำให้คำตอบที่ส่งกลับมาซ้ำกัน



Advanced Usage

- ✓ System Role
- ✓ Preventing Duplicate Reply Messages



Facebook Chatbot

- Facebook Chatbot Creation**
- Validation and Authentication**
- ChatGPT Integration**
- Advance Usage**
- Live Mode**



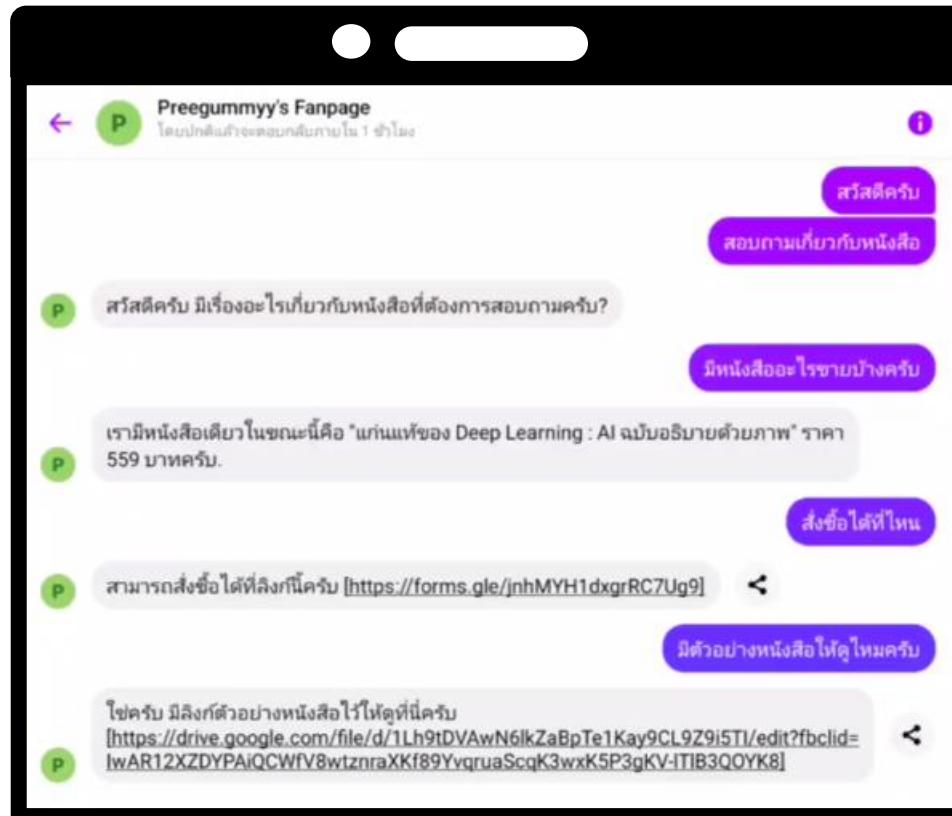
Live Mode

- Overview
- Pre-requisites
- Steps
- Things to Note



Overview

การเปิดใช้งาน Facebook Chatbot ให้สามารถโต้ตอบกับสาธารณะ และผู้ใช้ Facebook ทุกคน สามารถทำได้โดยการเปิด Live Mode



Live Mode

✓ Overview

- Pre-requisites
- Steps
- Things to Note



Pre-requisites

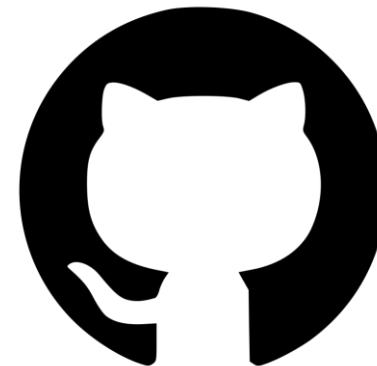
สำหรับการเปิดใช้งาน Facebook Chatbot ให้อยู่ใน Live Mode จำเป็นต้องตรวจสอบ ดังต่อไปนี้

- Testing
- Privacy Policy URL
- App Details
- App Review
- Terms and Conditions
- OAuth settings

Pre-requisites



For more detailed information



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/\(5\)release_app_to_live_mode.md#pre-requisites](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/(5)release_app_to_live_mode.md#pre-requisites)

Live Mode

- ✓ **Overview**
- ✓ **Pre-requisites**
 - Steps
 - Things to Note



Steps

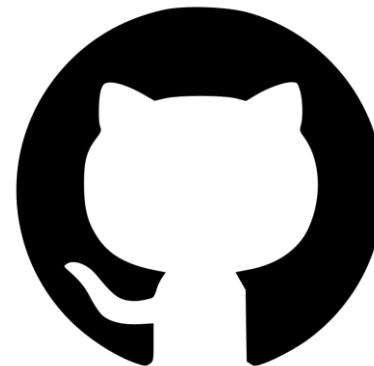
สำหรับขั้นตอนการเปิดใช้งาน Facebook Chatbot ให้อยู่ใน Live Mode มีดังนี้

1. Login to your Facebook account
2. Navigate to Facebook Developers
3. Select your app
4. Go to App Dashboard
5. Switch to Live mode
6. Confirm your decision
7. Review settings

Steps



For more detailed information



[https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/\(5\)release_app_to_live_mode.md#steps](https://github.com/TAUTOLOGY-EDUCATION/ChatGPT-API/blob/main/facebook_chatbot/(5)release_app_to_live_mode.md#steps)

Live Mode

- ✓ **Overview**
- ✓ **Pre-requisites**
- ✓ **Steps**
 - Things to Note



Things to Note

- ตรวจสอบให้แน่ใจว่าได้ทำการทดสอบระบบ Chatbot ใน Development Mode อย่างละเอียดแล้ว
- แม้จะเปลี่ยนเป็น Live Mode แต่ก็ยังสามารถเปลี่ยนแปลงการตั้งค่าต่าง ๆ ได้ และในการเปลี่ยนแปลงการตั้งค่าบางอย่างต้องผ่านการตรวจสอบจาก Facebook ก่อน

Live Mode

- ✓ **Overview**
- ✓ **Pre-requisites**
- ✓ **Steps**
- ✓ **Things to Note**



Facebook Chatbot

- Facebook Chatbot Creation**
- Validation and Authentication**
- ChatGPT Integration**
- Advance Usage**
- Live Mode**



Thank You

