Building Web APIs with FastAPI

Lecturer: Bessy

Date: Sep. 1st, 2022

Whoami

- Name: Bessy, Chia, 丰嘉
- Experience:
  - 國衛院 生物資訊工程師 兼計畫助理 (現任)
  - 學生資訊社群 SIRLA、Project Abyss 共同創辦人 (現任)
  - 國衛院 Digi+ Talent 研習生
    - 專題「基因體大數據分析與精準醫療應用」

Outline

- Introduction to FastAPI
  - What is FastAPI ? What is an API ?
  - HTTP Methods for RESTful API
- Build a basic Web API with FastAPI & Google Sheet
  - Get all data, Query data
  - Insert data, Update data, Delete data (補充)

Introduction to FastAPI

What is FastAPI ?

FastAPI

web framework

FastAPI framework, high performance, easy to learn, fast to code, ready for production

What is FastAPI ?

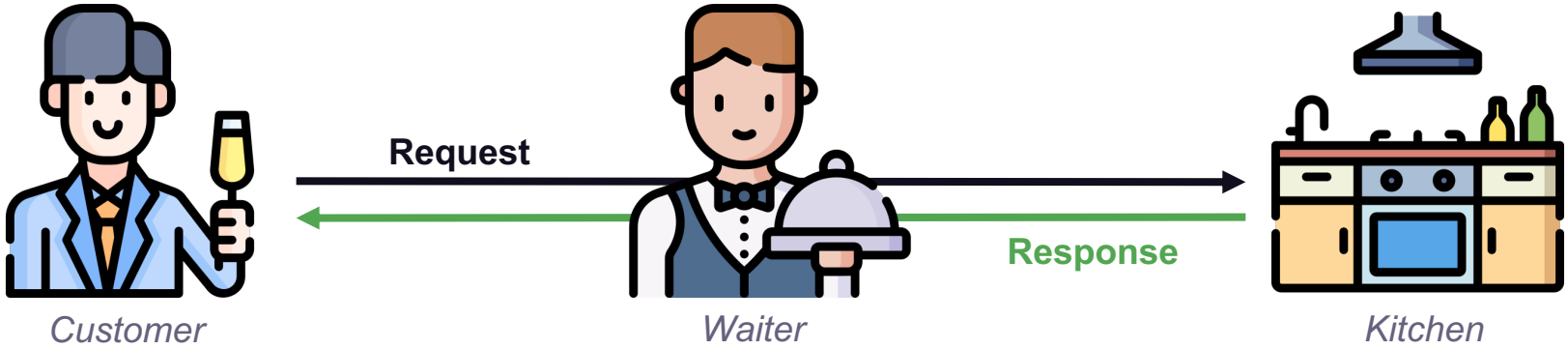
- **Key features**
  1. **Fast:** Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
  2. **Easy:** Designed to be easy to use and learn. Less time reading docs.
  3. **Fast to code:** Increase the speed to develop features by about 200% to 300%.
  4. **Robust:** Get production-ready code. With automatic interactive documentation.
  5. **Standards-based:** Based on (and fully compatible with) the open standards for APIs: OpenAPI (previously known as Swagger) and JSON Schema.

What is an API ?

“ Software products exchange data and functionalities via machine-readable interfaces – APIs (application programming interfaces). ”

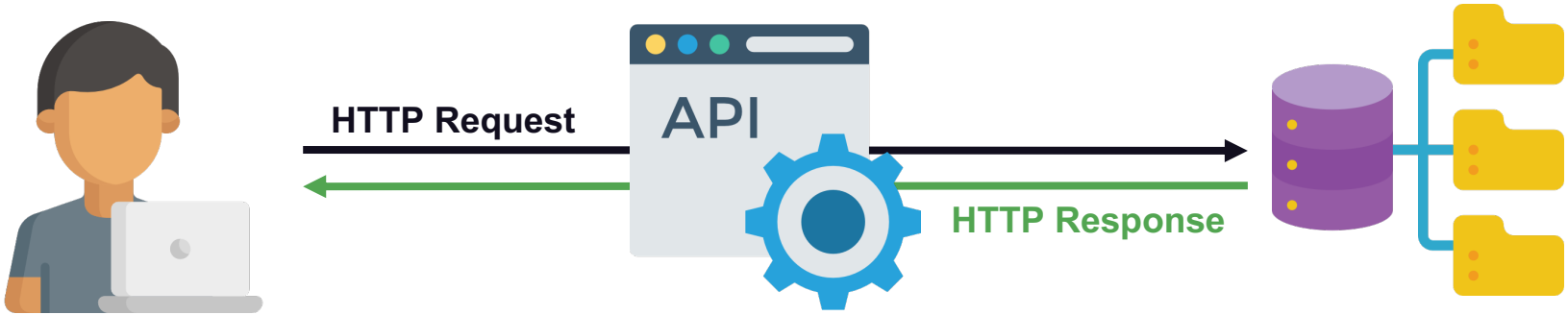
What is an API ?

Real World



What is an API ?

Web Application Development World



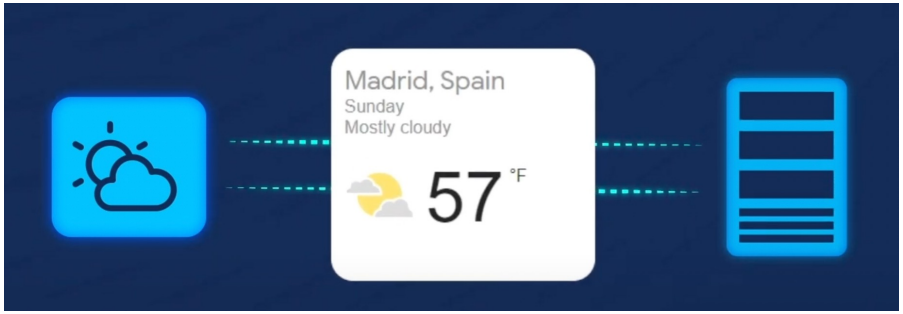
Web Client

Web APIs (e.g. FastAPI)

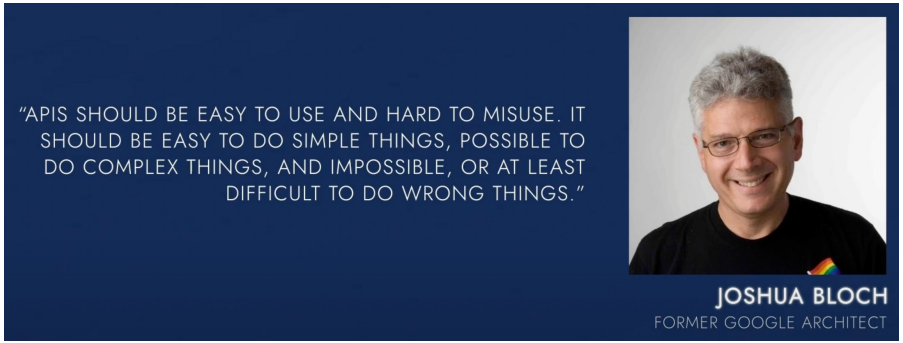
Database/System Server

API Principles

1. Limited access



2. System-Independent



3. Simple in use

HTTP Methods for RESTful API

So a client makes a request to the server,
how would the server know which operation to perform ?

Commonly-used HTTP Methods

CRUD: Read

To read/retrieve Data

CRUD: Create

To insert data

To update data

CRUD: Update/Replace

To delete data

CRUD: Delete

HTTP Status Code

HTTP Status Code	Description
1xx: Informational	The request was received and the process is continuing.
2xx: Success	The action was successfully received, understood, and accepted.
3xx: Redirection	Further action must be taken in order to complete the request.
4xx: Client Error	The request contains incorrect syntax or cannot be fulfilled.
5xx: Server Error	The server failed to fulfill an apparently valid request.

Build a basic Web API with FastAPI & Google Sheet

Overview

1. Installation
2. Run a basic Web API with FastAPI (Official sample code)
3. Setup and connect with Google Sheets API
4. Run a Web API with FastAPI & Google Sheet (Database)
    - [GET] Get all data
    - [GET] Query data
    - [POST] Insert data
    - [PUT] Update data
    - [DELETE] Delete data

1. Installation (Windows)

- 請先確定已安裝 Python 3.x 版本。
- 建立專案資料夾、Python 虛擬環境，並在虛擬環境內進行專案開發。

```
# Create isolated Python environment
$ python -m venv [MyVenv-name]
```

```
# Activate virtual environment
$ [MyVenv-name]\Scripts\activate.bat
```

```
# Deactivate virtual environment (Select 1 to use)
(MyVenv-name)$ deactivate
(MyVenv-name)$ [MyVenv-name]\Scripts\deactivate
```

1. Installation (MacOS)

- 請先確定已安裝 Python 3.x 版本。
- 建立專案資料夾、Python 虛擬環境，並在虛擬環境內進行專案開發。

```
# Create project folder & isolated Python environment
$ mkdir [Project-name]
$ cd [Project-name]
$ python3 -m venv [MyVenv-name]
```

```
# Activate virtual environment
$ source [MyVenv-name]/bin/activate
```

```
# Deactivate virtual environment
(MyVenv-name)$ deactivate
```

1. Installation (Windows/MacOS)

```
# Upgrade pip
(MyVenv-name)$ python -m pip install --upgrade pip

# Install packages
(MyVenv-name)$ pip install fastapi
(MyVenv-name)$ pip install "uvicorn[standard]"
```

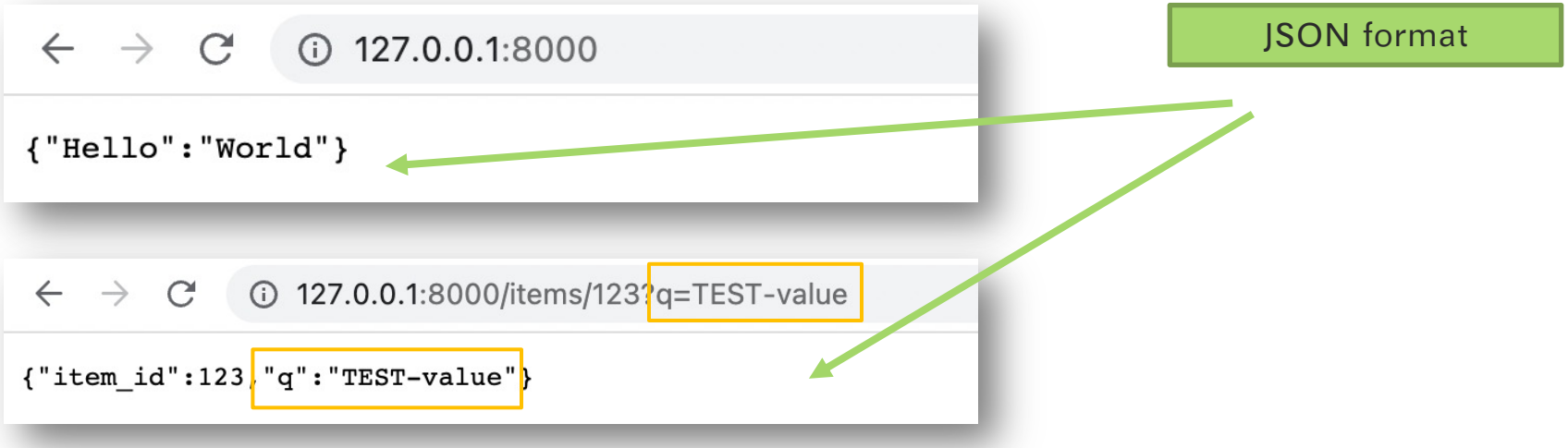
2. Run a basic Web API with FastAPI

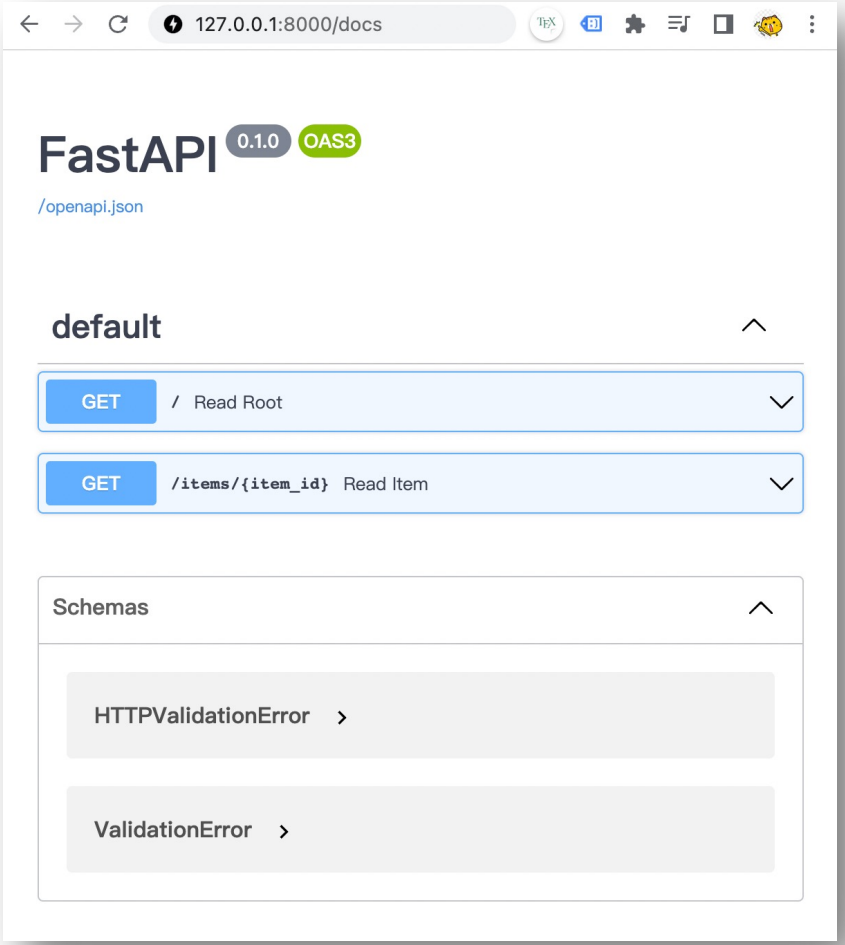
main.py

```
from typing import Union
from fastapi import FastAPI
import uvicorn

app = FastAPI()

@app.get("/")
def read_root():
    return {"Hello": "World"}

@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}

if __name__ == "__main__":
    uvicorn.run(app='main:app', host="127.0.0.1", port=8000, reload=True, debug=True)
```

2. Run a basic Web API with FastAPI

```
(MyVenv-name)$ python main.py
```

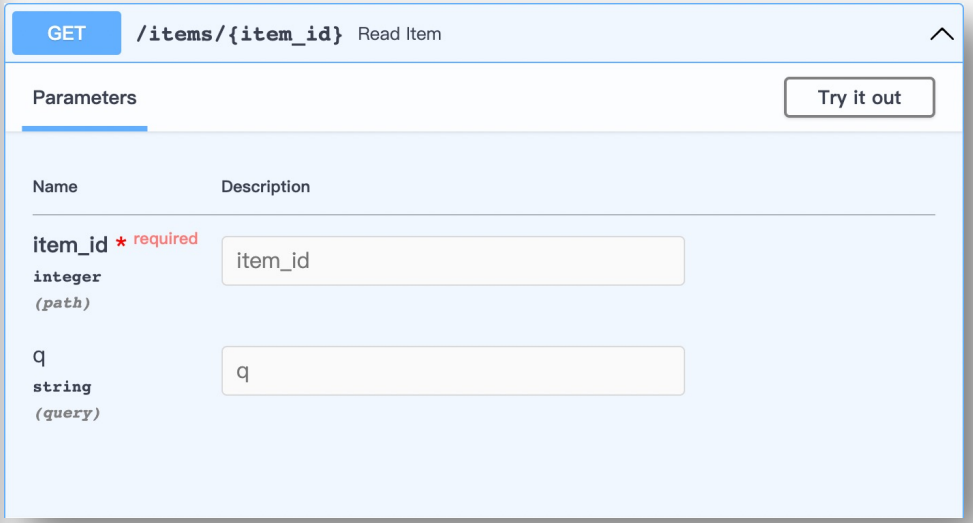


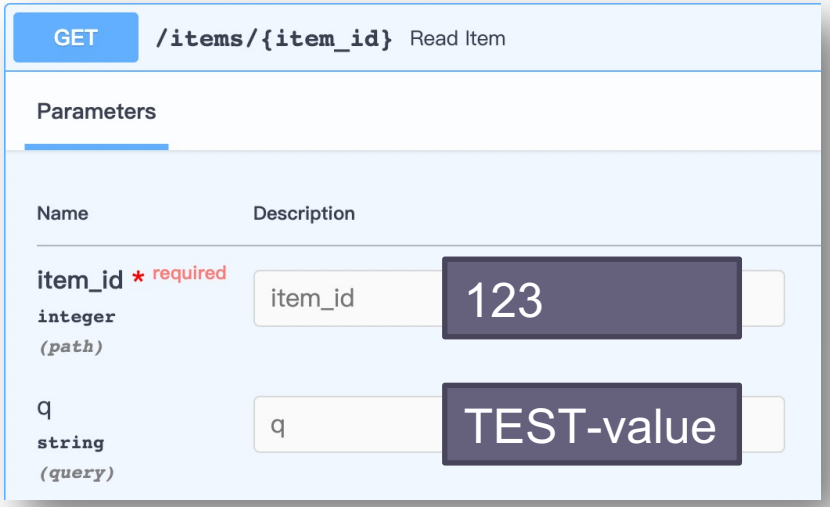


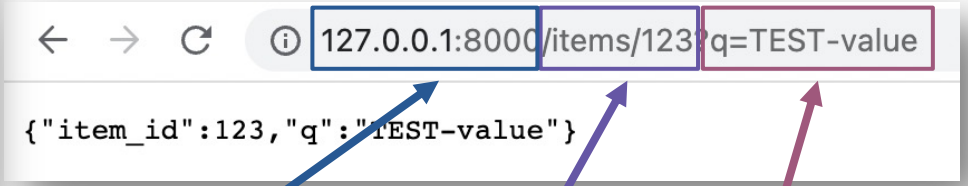
Automatic interactive API documentation

(provided by Swagger UI)

http://127.0.0.1:8000/docs



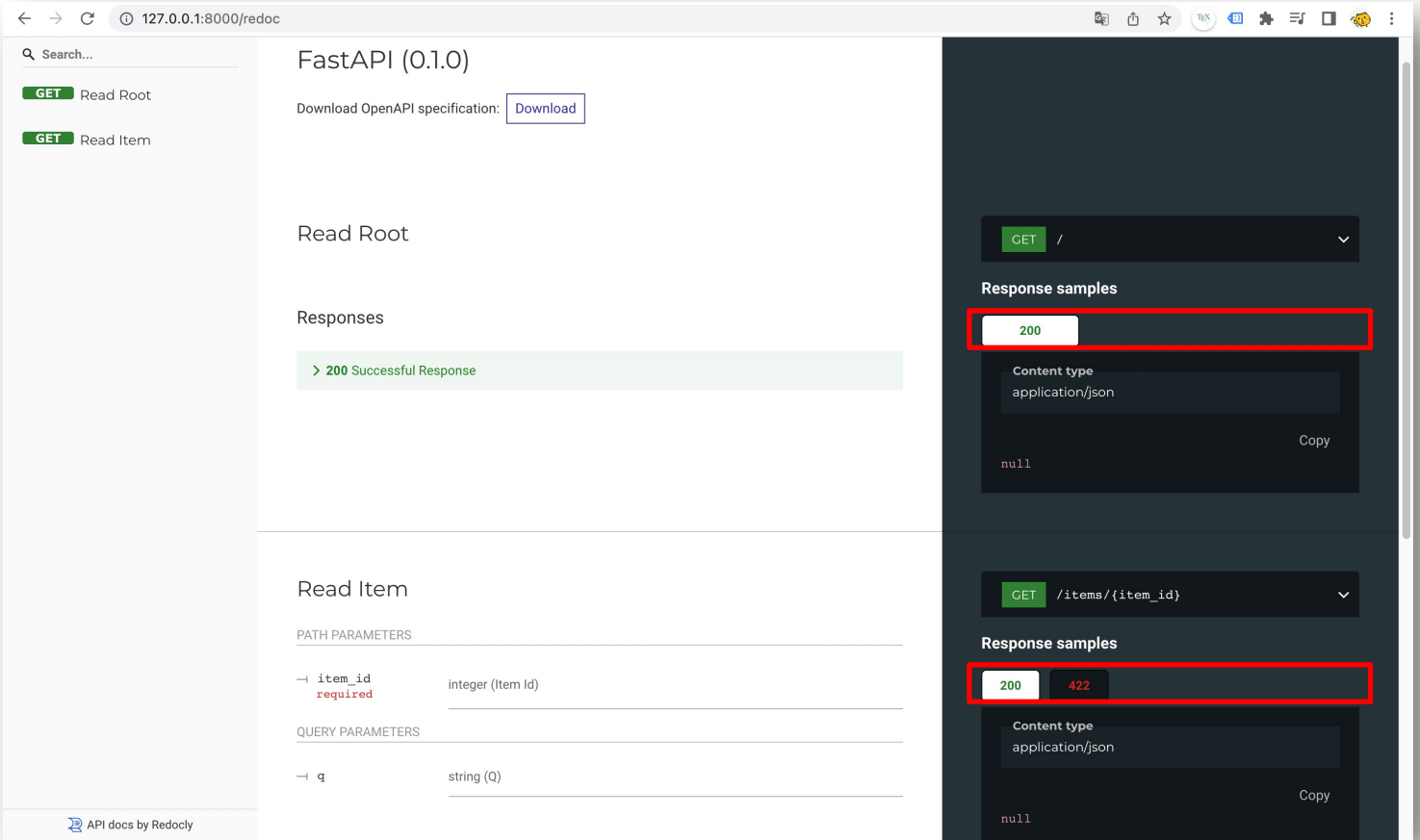




Resources:
https://www.semrush.com/blog/url-parameters/

Alternative automatic documentation (provided by ReDoc)

http://127.0.0.1:8000/redoc



200
OK

422
Unprocessable Entity

3. Setup & connect with Google Sheets API

- 目的：將 Google Sheet 當作資料庫來使用。
- 步驟：
  1. 申請 Google Sheet API > 新增專案

3. Setup & connect with Google Sheets API

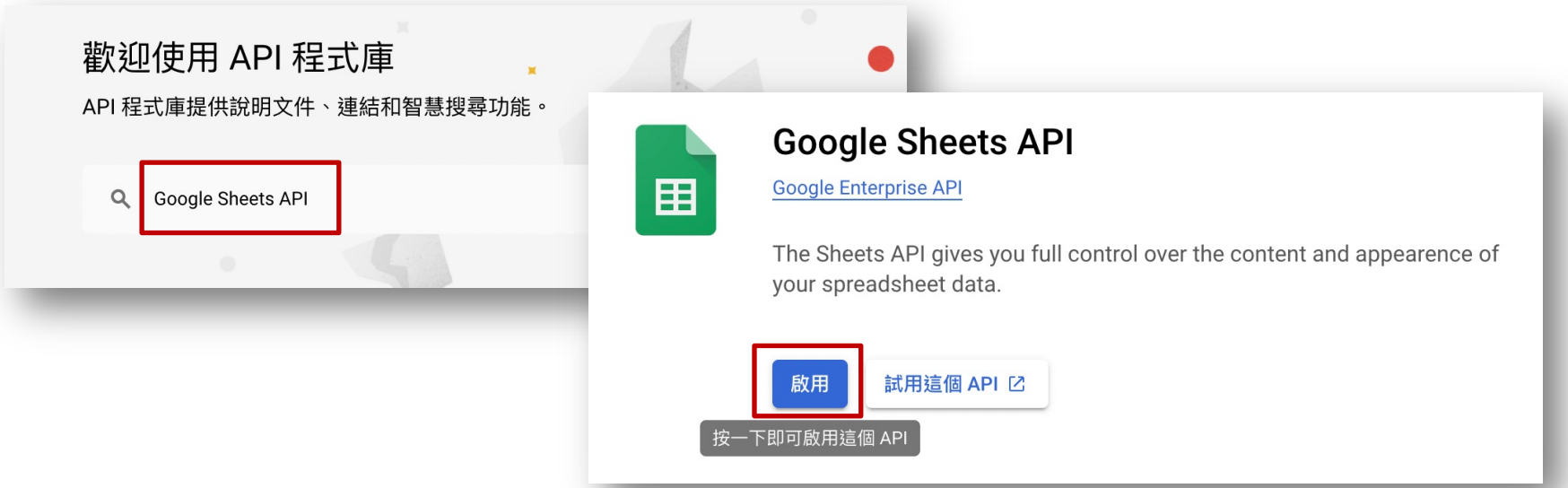
- 步驟：
  2. 建立新專案

3. Setup & connect with Google Sheets API

- 步驟：
  3. 選取「Python-FastAPI-GoogleSheet」專案，並點選上方的「啟用 API 和服務」

3. Setup & connect with Google Sheets API

- 步驟：
  4. 於搜尋框輸入「Google Sheets API」> 點選「啟用」

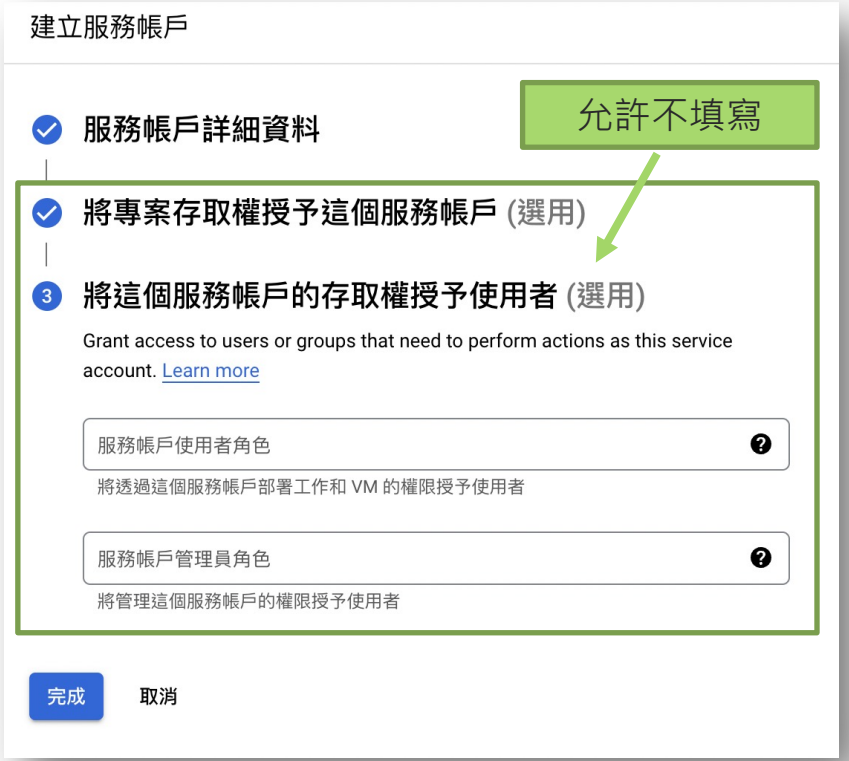


3. Setup & connect with Google Sheets API

- 步驟：

5. 啟用後，點選畫面左方的「憑證」，再點選上方「建立憑證(+CREATE CREDENTIALS)」> 「服務帳戶」> 填寫「服務帳務詳細資料」

3. Setup & connect with Google Sheets API



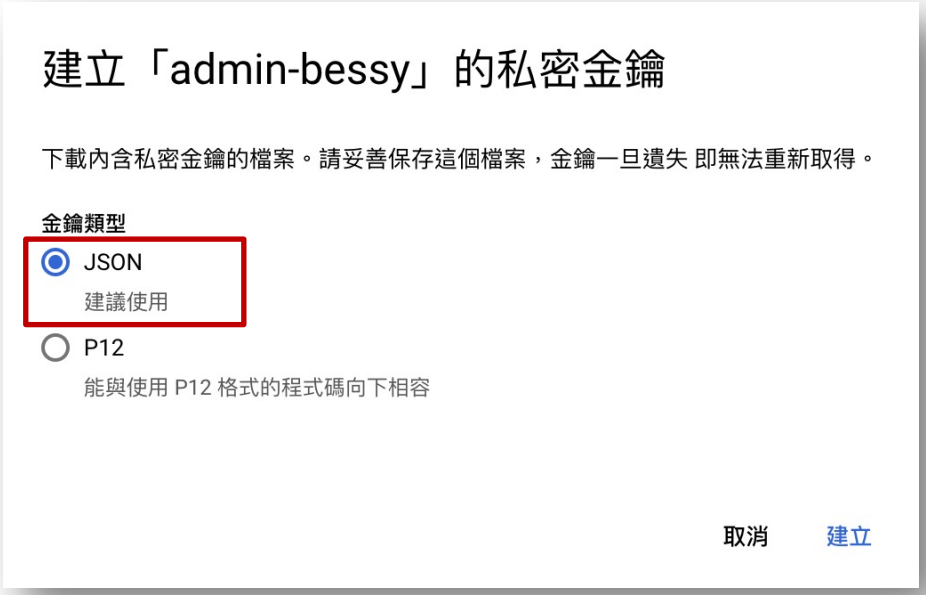
3. Setup & connect with Google Sheets API

- 步驟：

6. 點選服務帳戶的電子郵件 > 金鑰 > 建立新的金鑰

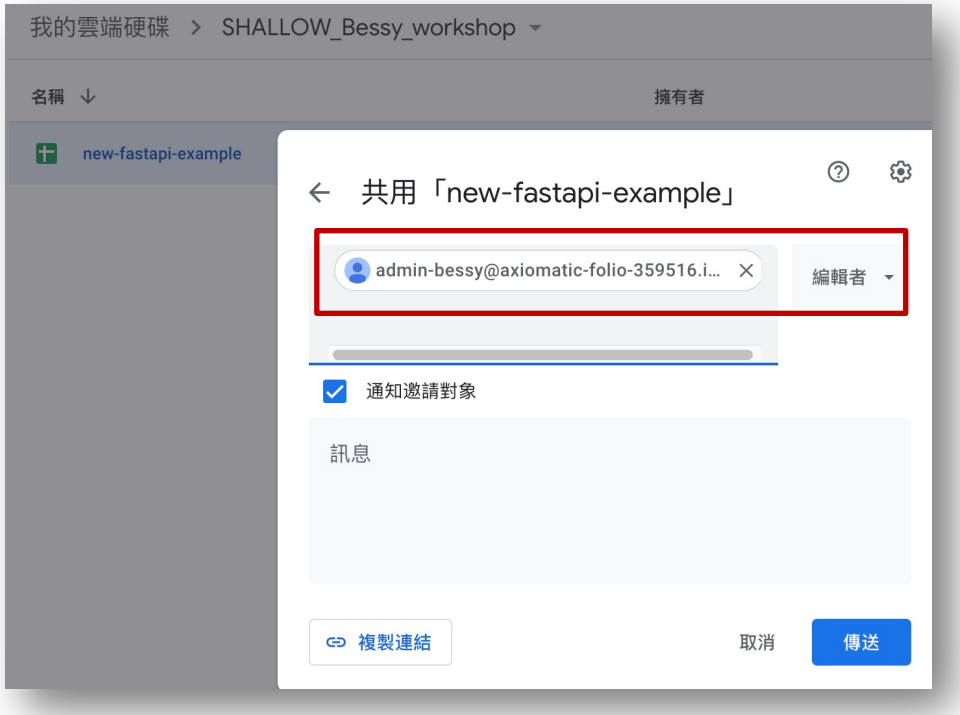
3. Setup & connect with Google Sheets API

- 步驟：
  7. 選擇 JSON 後按「建立」，將下載的 .json 存到專案的資料夾中



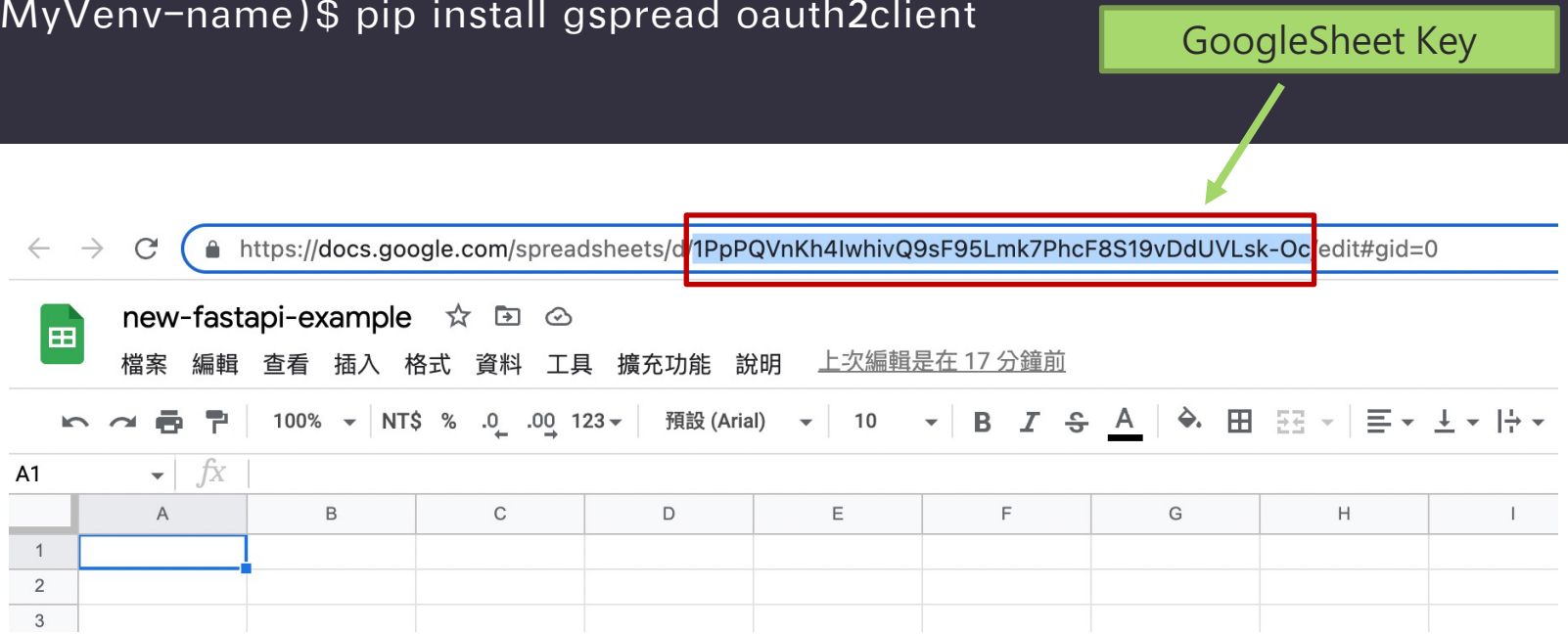
3. Setup & connect with Google Sheets API

- 步驟：
  8. 請至 Google Drive 新增一張 google sheet，「共用人員」貼上剛剛複製的電子郵件，選擇「編輯者」 > 傳送！



3. Setup & connect with Google Sheets API

```
# Install packages
(MyVenv-name)$ pip install gspread oauth2client
```



3. Setup & connect with Google Sheets API

main.py

```
from oauth2client.service_account import ServiceAccountCredentials as SAC
import gspread

# ...

Json = "/Users/bessyhuang/Downloads/axiomatic-folio-359516-0f98f8adb09d.json"
Url = ["https://spreadsheets.google.com/feeds"]
Connect = SAC.from_json_keyfile_name(Json, Url)
GoogleSheets = gspread.authorize(Connect)
Sheet = GoogleSheets.open_by_key("1PpPQVnKh4IwhivQ9sF95Lmk7PhcF8S19vDdUVLsk-Oc")
Sheets = Sheet.sheet1

app = FastAPI()

# ...
```

GoogleSheet Key

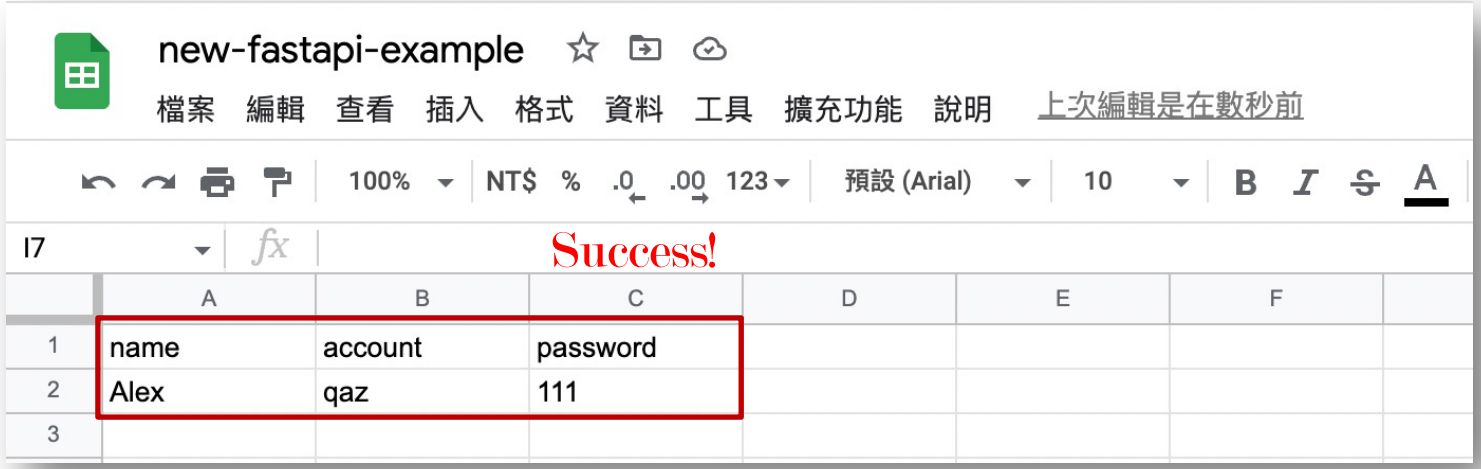
3. Setup & connect with Google Sheets API

main.py

```
# ...

if __name__ == "__main__":
    dataTitle = ["name", "account", "password"]
    datas = ["Alex", "qaz", "111"]
    Sheets.append_row(dataTitle)
    Sheets.append_row(datas)
    print("寫入成功")
    print(Sheets.get_all_values())

    uvicorn.run(app='main:app', host="127.0.0.1", port=8000, reload=True, debug=True)
```

3. Setup & connect with Google Sheets API

```
(MyVenv-name)$ python main.py
```



4. Run a Web API with FastAPI & Google Sheet (Database)

[GET] Get all data

[GET] Get all data

main.py

```
from oauth2client.service_account import ServiceAccountCredentials as SAC
from fastapi import FastAPI
import uvicorn
import gspread

Json = "json-file-path"
Url = ["https://spreadsheets.google.com/feeds"]
Connect = SAC.from_json_keyfile_name(Json, Url)
GoogleSheets = gspread.authorize(Connect)
Sheet = GoogleSheets.open_by_key("google-sheet-key")
Sheets = Sheet.sheet1

# ...
```

[GET] Get all data

main.py

```
# ...

app = FastAPI()

@app.get("/users/")
async def getAllData():
    return Sheets.get_all_records()

if __name__ == "__main__":
    uvicorn.run(app='main:app', host="127.0.0.1", port=8000, reload=True, debug=True)
```

4. Run a Web API with FastAPI & Google Sheet (Database)

[GET] Query data

[GET] Query data

main.py

```
import re

# ...
app = FastAPI()

@app.get("/users/{userName}")
async def FindUsernameInfo(userName: str):
    userName_re = re.compile(r'^{}'.format(userName))
    cell = Sheets.find(userName_re)
    if cell == None:
        return "Username {} doesn't exist in Google Sheet.".format(userName)
    else:
        r = cell.row
        x = [item for item in Sheets.row_values(r) if item]
        return f"在第{cell.col}直行,第{cell.row}橫列,{x}"
# ...
```

❖備註：僅查詢重複紀錄中的第一筆。

補充

4. Run a Web API with FastAPI & Google Sheet (Database)

[POST] Insert data

[POST] Insert data

main.py

```
from pydantic import BaseModel
from typing import List

# ...
class Info(BaseModel):
    id: int
    data: List[str] = ["Bessy", "test", "321"]

app = FastAPI()

@app.post("/users/")
async def AddNewUsernameInfo(info: Info):
        Sheets.append_row(info.data)
        return {"status": "SUCCESS", "data": info}
# ...
```

補充

4. Run a Web API with FastAPI & Google Sheet (Database)

[PUT] Update data

[PUT] Update data

main.py

```
from typing import Union

# ...
app = FastAPI()

@app.put("/users/{userName}")
async def UpdateExistUsernameInfo(userName: str, modified_account: Union[str, None] =
None, modified_password: Union[str, None] = None):
    cell = Sheets.find(userName)
    if cell == None:
        return "Username {} doesn't exist in Google Sheet.".format(userName)
    else:
        r = cell.row
        Sheets.update(f"B{r}:C{r}", [[modified_account, modified_password]])
        x = [item for item in Sheets.row_values(r) if item]
        return {"msg": x}
# ...
```

❖備註：僅修改重複紀錄中的第一筆。

補充

4. Run a Web API with FastAPI & Google Sheet (Database)

[DELETE] Delete data

[DELETE] Delete data

main.py

```
from typing import Union

# ...
app = FastAPI()

@app.delete("/users/{userName}")
async def DeleteExistUsernameInfo(userName: str):
    userName_re = re.compile(r'^{}'.format(userName))
    cell = Sheets.find(userName_re)
    if cell == None:
        return "Username {} doesn't exist in Google Sheet.".format(userName)
    else:
        r = cell.row
        Sheets.delete_row(r)
        return f"UserName {userName} 原本位於第 {cell.row} 橫列，現已刪除！"
# ...
```

❖備註：僅修改重複紀錄中的第一筆。

Reference

- What is API: Definition, Types, Specifications, Documentation
- HTTP Tutorial
- Learn REST: A RESTful Tutorial
- 什麼是真正的 RESTful API? #RESTful API應該長什麼樣子?
- API是什麼？有什麼應用？學會串接 Web API
- 網路基礎概論 - HTTP 協定、TCP/IP、API
- FastAPI + Google Sheet (Database)

Thanks!

Any questions?

You can find me at:

- @bessyhuang (GitHub)