

一、安裝

1-1、安裝

```
$ pip install fastapi
$ pip install uvicorn[standard]
```

1-2、基本架構

```
"""
程式名稱： main.py
"""
# 從 fastapi 載入 FastAPI 物件
from fastapi import FastAPI

# 創建 FastAPI 實例
app = FastAPI()

# 設置操作路徑（裝飾器），常用的有 @app.get、@app.post、@app.put、@app.delete
@app.get('/')
def root():
    # 函數返回值，可以是 list、dict、str、int、或是 Pydantic 物件...
    return {"message": "Hello World"}
```

1-3、啟動

A. 於命令列呼叫

```
$ uvicorn main.app --reload
```

B. 於檔案中呼叫

```
from fastapi import FastAPI
import uvicorn

app = FastAPI()
@app.get('/')
def root():
    return {'message': 'Hello World'}

if __name__ == '__main__':
    uvicorn.run(app, host='0.0.0.0', port=8000)
```

```
# 如果要設置更新後自動重啟，則 1st 參數必須為字串
# uvicorn.run('main:app', host='0.0.0.0', port=8000, reload=True)
```

二、路徑（route）操作裝飾器中的路徑參數

```
@startuml
    skinparam backgroundColor #EEEEBD
    skinparam handwritten true
    actor Customer
    Customer -> "login()" : username & password
    "login()" -> Customer : session token
    activate "login()"
    Customer -> "placeOrder()" : session token, order info
    "placeOrder()" -> Customer : ok
    Customer -> "logout()"
    "logout()" -> Customer : ok
    deactivate "login()"
@enduml
```

三、查詢參數

四、請求體（Request Body）

五、設置查詢參數的驗證條件（字串驗證）

六、設置路徑參數的驗證條件（數值驗證）

七、請求體多種參數

八、Body-Schema 模型

九、Body-Nested Models 嵌套模型

十、Extra Data Types 額外數據類型s

十一、Cookie Parameters

十二、Header Parameters

十三、響應模型 (Response Model)
