

# FastApi (1일차)

## 공부해볼것

**class**와 생성자 공부 필요

**mypy** -> 타입 힌트에 대한 공부필요

포지셔널 아규먼트, 키워드 아규먼트

**DTO**를 왜 쓰느냐? **DTO**?

**pydantic** 에 대해?

**tortoise-orm** 에 대해?

poetry

가상환경을 쓰는 이유 : 동일한 버전

책 추천 : 구글 엔지니어는 이렇게 일한다

강의 노션 주소 :

<https://www.notion.so/FastAPI-1-2-200211914ffb4797b23976c43c738be0>

async 설명 영상 :

[https://www.youtube.com/playlist?](https://www.youtube.com/playlist?list=PLhNSoGM2ik6SIkVGXWBwerucXjgP1rHmB)

[list=PLhNSoGM2ik6SIkVGXWBwerucXjgP1rHmB](https://www.youtube.com/playlist?list=PLhNSoGM2ik6SIkVGXWBwerucXjgP1rHmB)

FastApi

- 타입 친화적인 설계
- Async await 지원 ( 요즘 장고는 지원하긴 합니다 )
- 단순함과 명료함
- 친절하고 자세한 문서
- 성능 ( 장고 orm은 성능이 안좋기로 유명합니다 )

## Swagger

- Serialize, data 리턴
- 제너레이션?

## 프로젝트 셋업

- poetry add fastapi gunicorn

## Pydantic 설치

- poetry add pydantic pydantic-settings

## lint, 정적분석

- poetry add -D black isort mypy coverage pytest pytest-asyncio
- 이번에는 coverage도 같이 설치

api를 호출하기 위해선 뭘 해야하죠? (쓰는법을 알아야한다)

포스트맨(postman) : 설치 완료

httpie : 설치 완료 (설치 확인 : which http)

curl

<자신이 만든 api를 써볼 줄 알아야 한다.>

gunicorn 말고 uvicorn이 대신한다.

http -> tcp 연결 위해서 텍스트를 정해진 규칙대로 주고 받는 것입니다.

HTTP/1.1 200 OK : 버전 1.1 / status code 200 / status code에 상응하는 문자

(첫 줄과 헤더 줄 사이에는 개행을 "한 번" 한다)

(헤더 하나 당 개행을 한 줄씩 한다.)

(request header 란 response header 가 있습니다.)

(request, response 에 둘 다 쓸 수 있는 header 도 있다.)

(response header 는 response 에 대한 메타 데이터다.)

content-length : 25

content-type : application/json

date : Wed, 30 Oct 2024 02:04:53 GMT

server : uvicorn

(body : header라인의 )

```
{  
"message" : "hello world"  
}
```

HTTP response status codes

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>

1. http 는 문자열을 주고 받습니다.
2. 그 문자열은 이미지일 수도 있고 (binary), yaml 일 수도 있고, json 일 수도 있고, html 일 수도 있습니다.
3. body 가 무엇인지는 "content-type" 헤더가 설명합니다.

헤더 = response 에 대한 설명이 들어있다.

poetry add -D black isort mypy coverage pytest pytest-asyncio

isort : 정렬

black : 포메팅

mypy : 정적 타입 검사 / 코드를 실행하지 않는다. / 그냥 읽는다. / 그만큼 빨리 에러를 잡음

(실행하는 도중에 바뀌면 동적)

coverage : 테스트 코드가 몇 %실행됐는지를 알려준다.

## test.sh

```
# Bash 스크립트에서 사용되는 옵션으로, 스크립트의 실행 방식을 제어하는데 도움을 준다.
```

```
# 위에 것이 실패하면 아래로 안내려가게 한다.
```

```
# pipfail 이 없으면 위에서 실패해도 아래로 내려간다.
```

```
set -eo pipefail
```

```
COLOR_GREEN=`tput setaf 2`;
```

```
COLOR_NC=`tput sgr0`;` # No Color
```

```
# 포메팅
echo "Starting black"
poetry run black .

# 코드 정렬
echo "Starting isort"
poetry run isort .

# 정적 타입 검사
echo "Starting mypy"
poetry run mypy .

echo "Starting pytest with coverage"
# coverage를 측정할건데 "pytest"를 실행할 것이다
poetry run coverage run -m pytest
# 그래서, 테스트는 다 끝났는데, coverage 가 어떻게 되는지 조회할 것이다.
poetry run coverage report -m
poetry run coverage html

# 모든 것이 검증됐다.
echo "${COLOR_GREEN}All tests passed
successfully!${COLOR_NC}"
```

poetry add "tortoise-orm[asyncmy]" aerich cryptography

파이썬에서 함수에 전달하는 인수(인자)는  
positional argument와 keyword argument로 나눌 수 있습니다.

positional argument : 위치 매개 변수  
keyword argument : 키워드 매개 변수

키워드로 인자를 전달하면 순서는 무시된다.