AI 포팅 매뉴얼

python server

▼ setting_local.json

```
{
  "OPENAI_API_KEY": "YOUR_API_KEY",
  "YOUTUBE_API_KEY": "YOUR_API_KEY",
  "GPT_PROMPT_BILL": "법률안의 제목 또는 식별자는 무엇인지 알려줘.
  "GPT_PROMPT_ISSUE": "당신은 기사를 빠르게 검토하고 요점을 추출하
  "GPT_PROMPT_KEYWORD": "카테고리 목록: ['기타', '정치', '선거',
  "video_file_path": "../whisper/src/video/",
  "audio_file_path": "../whisper/src/audio/",
  "script_file_path": "../whisper/src/script/",
  "clip_file_path": "../whisper/src/clip_video/",
  "image_file_path": "../whisper/src/image/",
  "AWS_ACCESS_KEY_ID": "YOUR_API_KEY",
  "AWS_SECRET_ACCESS_KEY": "YOUR_API_KEY",
  "AWS_DEFAULT_REGION": "ap-northeast-2",
  "AWS_CLOUD_FRONT" : "https://youradress.cloudfront.net/"
  "DB_HOST": "HOST",
  "DB_PORT": "PORT",
  "DB_USER": "USER",
  "DB_PASSWORD":"PASSWORD",
  "DB_DATABASE": "DATABASE",
  "CATEGORY_LIST": "['기타','정치', '선거', '정당', '정책', '논
}
```

▼ dockerfile

```
FROM ubuntu:latest
LABEL authors="SSAFY"

ENTRYPOINT ["top", "-b"]

#
FROM python:3.10.14
```

```
#
WORKDIR /code

#
COPY ./requirements.txt /code/requirements.txt

#
RUN pip install --upgrade -r /code/requirements.txt

#
COPY ./app /code/app

#
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--po
```

▼ deploy.sh

```
sudo docker ps -a -q --filter "name=python-cd" | grep -q .

sudo docker rmi python-cd

sudo docker build -t python-cd:latest -f /home/user/ai-app

sudo docker run -d -p 8000:8000 --name python-cd --network
```

▼ ai-pipeline

```
pipeline {
    agent any

    stages {
        stage('Git Clone') {
            steps {
                git branch: 'ai-master', credentialsId: 'g
            }
        }
        stage('AI-Deploy') {
            steps {
                sshagent(credentials: ['aws_key']) {

                        sh '''
                        ssh -o StrictHostKeyChecking=no us
                        ssh -o StrictHostKeyChecking=no us
                        scp -r /var/jenkins_home/workspace
                        ssh -o StrictHostKeyChecking=no us

                        scp user@ec2ip:/home/user/ai-appli

                        ssh -o StrictHostKeyChecking=no us
                        ssh -o StrictHostKeyChecking=no us


                    '''
                }
            }
        }

    }
     post {
        success {
            script {
                def Author_ID = sh(script: "git show -s --
                def Author_Name = sh(script: "git show -s
                mattermostSend (color: 'good',
                message: "빌드 성공: ${env.JOB_NAME} #${env.
                )
            }
        }
        failure {
            script {
                def Author_ID = sh(script: "git show -s --
                def Author_Name = sh(script: "git show -s
                mattermostSend (color: 'danger',
                message: "빌드 실패: ${env.JOB_NAME} #${env.

                )
            }
        }
        }
}
```

version

▼ requirements.txt

```
aiohttp==3.9.3
aiosignal==1.3.1
alembic==1.13.1
annotated-types==0.6.0
antlr4-python3-runtime==4.9.3
anyio==4.2.0
APScheduler==3.9.0
asteroid-filterbanks==0.4.0
async-timeout==4.0.3
attrs==23.2.0
audioread==3.0.1
av==11.0.0
boto3==1.34.69
botocore==1.34.69
Bottleneck==1.3.7
cachetools==5.3.3
certifi==2024.2.2
cffi==1.16.0
charset-normalizer==3.3.2
click==8.1.7
colorama==0.4.6
coloredlogs==15.0.1
colorlog==6.8.2
contourpy==1.2.0
cryptography==42.0.5
```

```
ctranslate2==4.1.0
cycler==0.12.1
dataclasses-json==0.6.4
decorator==4.4.2
distro==1.8.0
docopt==0.6.2
einops==0.7.0
exceptiongroup==1.2.0
fastapi==0.110.0
fastapi-utils==0.2.1
faster-whisper==1.0.0
ffmpeg-python==0.2.0
filelock==3.13.1
flatbuffers==24.3.7
fonttools==4.50.0
frozenlist==1.4.1
fsspec==2024.3.1
future==1.0.0
gmpy2==2.1.2
google-api-core==2.18.0
google-api-python-client==2.123.0
google-auth==2.29.0
google-auth-httplib2==0.2.0
googleapis-common-protos==1.63.0
greenlet==3.0.3
h11==0.14.0
httpcore==1.0.2
httplib2==0.22.0
httpx==0.26.0
huggingface-hub==0.21.4
humanfriendly==10.0
HyperPyYAML==1.2.2
idna==3.4
imageio==2.34.0
imageio-ffmpeg==0.4.9
Jinja2==3.1.3
jmespath==1.0.1
joblib==1.3.2
```

```
jsonpatch==1.33
jsonpointer==2.4
julius==0.2.7
kiwisolver==1.4.5
langchain==0.1.13
langchain-community==0.0.29
langchain-core==0.1.33
langchain-text-splitters==0.0.1
langsmith==0.1.31
lazy_loader==0.3
librosa==0.10.1
lightning==2.2.1
lightning-utilities==0.11.0
llvmlite==0.42.0
Mako==1.3.2
markdown-it-py==3.0.0
MarkupSafe==2.1.3
marshmallow==3.21.1
matplotlib==3.8.3
mdurl==0.1.2
mkl-fft==1.3.8
mkl-random==1.2.4
mkl-service==2.4.0
moviepy==1.0.3
mpmath==1.3.0
msgpack==1.0.8
multidict==6.0.5
mypy-extensions==1.0.0
mysql==0.0.3
mysql-connector-python==8.3.0
mysqlclient==2.2.4
networkx==3.1
nltk==3.8.1
numba==0.59.1
numexpr==2.8.7
numpy==1.26.4
omegaconf==2.3.0
onnxruntime==1.17.1
```

```
openai==1.14.2
opencv-python==4.9.0.80
optuna==3.6.0
orjson==3.9.15
packaging==23.2
pandas==2.2.1
pandas-stubs==2.1.4.231227
pillow==10.2.0
pip==23.3.1
platformdirs==4.2.0
pooch==1.8.1
primePy==1.3
proglog==0.1.10
proto-plus==1.23.0
protobuf==4.25.3
pyannote.audio==3.1.1
pyannote.core==5.0.0
pyannote.database==5.0.1
pyannote.metrics==3.2.1
pyannote.pipeline==3.0.1
pyasn1==0.5.1
pyasn1-modules==0.3.0
pycparser==2.21
pydantic==1.10.14
pydantic_core==2.14.6
Pygments==2.17.2
PyMySQL==1.0.2
pyparsing==3.1.2
pyreadline3==3.4.1
python-dateutil==2.8.2
pytorch-lightning==2.2.1
pytorch-metric-learning==2.4.1
pytube==15.0.0
pytz==2023.3.post1
PyYAML==6.0.1
regex==2023.12.25
requests==2.31.0
rich==13.7.1
```

```
rsa==4.9
ruamel.yaml==0.18.6
ruamel.yaml.clib==0.2.8
s3transfer==0.10.1
safetensors==0.4.2
schedule==1.2.1
scikit-learn==1.4.1.post1
scipy==1.12.0
semver==3.0.2
sentencepiece==0.2.0
setuptools==68.2.2
shellingham==1.5.4
six==1.16.0
sniffio==1.3.0
sortedcontainers==2.4.0
soundfile==0.12.1
soxr==0.3.7
speechbrain==1.0.0
SQLAlchemy==1.4.52
starlette==0.36.3
sympy==1.12
tabulate==0.9.0
tenacity==8.2.3
tensorboardX==2.6.2.2
threadpoolctl==3.4.0
tiktoken==0.6.0
tokenizers==0.15.2
torch==2.0.0
torch-audiomentations==0.11.1
torch-pitch-shift==1.2.4
torchaudio==2.0.0
torchmetrics==1.3.2
tqdm==4.65.0
transformers==4.39.1
typer==0.10.0
types-pytz==2022.4.0.0
typing_extensions==4.9.0
typing-inspect==0.9.0
```

```
tzdata==2023.3
tzlocal==2.1
uritemplate==4.1.1
urllib3==2.2.1
uvicorn==0.29.0
wheel==0.41.2
whisperx==3.1.1
yarl==1.9.4
```

```
git version 2.25.1
Docker version 25.0.
PyCharm: 2023.3.5

MySQL: 8.3

FastAPI: 0.110.0
Python: 3.10.14

whisperx: 3.1.1
uvicorn: 0.29.0
```

python 가상환경 세팅

GPU 확인 후 자신에게 맞는 CUDA 버전 설치

https://www.wikiwand.com/en/CUDA#/GPUs_supported

```
# GPU 버전확인
nvidia-smi
```

가상환경 세팅전 Conda 설치 필수

https://www.anaconda.com/download#downloads

```
# 가상환경 생성
conda create --name whisperx python=3.10

# 가상환경 활성화 (git bash)
source activate whisperx

# pyTorch, CUDA 설치
conda install pytorch==2.0.0 torchaudio==2.0.0 pytorch-cuda=1

# whisperX 설치
pip install git+https://github.com/m-bain/whisperx.git


# if 가상환경 비활성화 하고싶으면
conda deactivate
```

CUDA 사용을 위한 cuDNN 설치 필수

- https://developer.nvidia.com/cudnn
- 설치 후 CUDA 경로에 복사붙여넣기

  ```
  # 예시 경로
  C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v12.0\
  ```

정상적으로 설치되었는지 확인

```
# CUDA 설치 버전확인
nvcc --version

# cuDNN 설치 버전확인 (터미널에서 입력)
python
import torch
print("cudnn version:{}".format(torch.backends.cudnn.version(
print("cuda version: {}".format(torch.version.cuda))
```

라이브러리 설치

```
# 필요한 라이브러리 설치
pip install > requirements.txt
```

YouTube.py 사용법

- 해당 유튜브 채널에서 페이지 소스보기 버튼 클릭 후 “exteralId”값 추출

view_youtube_list.py

```
channel_id = '추출한 exteralId'

# 추출한 id값 입력후 이후
python youtube_main
```