

YouTube Data API 서비스 사용 설계문서

1. 개요

본 API 클라이언트는 위크맨 채널에 올라오는 다양한 유튜브 콘텐츠를 효율적으로 검색 및 필터링하고, 본편을 포함한 바이럴 가능성이 높은 동영상을 실시간으로 모니터링하여, 콘텐츠 기획과 마케팅 전략 수립에 활용하기 위한 목적으로 개발되었습니다. 이를 통해 위크맨 IP의 영향력 및 확산 동향을 체계적으로 분석합니다.

2. API 서비스 개요

- 서비스명: YouTube Data API v3
- 주요 기능:
 - 키워드 기반 영상 검색
 - 영상 상세 정보 조회 (제목, 채널명, 조회수, 썸네일 등)
 - 채널 구독자 수 조회
 - 업로드 날짜, 조회수, 구독자 수 등의 조건 필터링 및 정렬

3. 주요 기능 설명 및 사용 방법

3.1 영상 검색

- API 메서드: `youtube.search().list`
- 파라미터:
 - `q`: 검색어 (예: "위크맨")
 - `type`: "video" 고정
 - `part`: "id,snippet"
 - `maxResults`: 최대 결과 개수 (예: 10, 25, 50)
 - `order`: "relevance" (관련성 기준 정렬)
- 설명: 입력된 키워드로 유튜브 영상 검색 결과를 받아옵니다.

3.2 영상 상세 정보 조회

- API 메서드: `youtube.videos().list`
- 파라미터:
 - `id`: 검색된 영상 ID
 - `part`: "snippet,statistics"

- 설명: 영상의 제목, 채널명, 업로드일, 조회수, 썸네일 URL 등을 상세히 조회합니다.

3.3 채널 구독자 수 조회

- API 메서드: `youtube.channels().list`
- 파라미터:
 - `id`: 채널 ID
 - `part`: "statistics"
- 설명: 해당 채널의 구독자 수를 조회합니다.

3.4 데이터 필터링 및 정렬

- 업로드 날짜, 조회수, 구독자 수 기준 필터링
- 조회수 또는 구독자 수 기준 내림차순/오름차순 정렬

4. 인증 및 접근 방식

- API 키 인증 방식 사용
- API 호출 시 `developerKey` 파라미터에 API 키를 전달하여 권한 검증

5. 구현 환경 및 라이브러리

- Python 3.x
- 주요 라이브러리
 - `googleapiclient.discovery` (YouTube Data API 클라이언트 생성)
 - `streamlit` (웹 UI 구현)
 - `pandas` (데이터 테이블 처리)
 - `datetime` (날짜 처리)

6. 파이썬 코드

```
import streamlit as st
import pandas as pd
from googleapiclient.discovery import build
import datetime

API_KEY = "YOUR_API_KEY"

def search_youtube(query, max_results=10):
    youtube = build("youtube", "v3", developerKey=API_KEY)
    search_response = youtube.search().list(
```

```

        q=query, type="video", part="id,snippet", maxResults=max_results, order="relevance"
    ).execute()
    return search_response["items"]

def get_video_details(video_id):
    youtube = build("youtube", "v3", developerKey=API_KEY)
    video_response = youtube.videos().list(
        id=video_id, part="snippet,statistics"
    ).execute()
    item = video_response["items"][0]
    stats = item["statistics"]
    snippet = item["snippet"]
    return {
        "title": snippet["title"],
        "channel": snippet["channelTitle"],
        "publishedAt": snippet["publishedAt"][:10],
        "thumbnail": snippet["thumbnails"]["default"]["url"], # 기본 크기 썸네일
        "viewCount": int(stats.get("viewCount", 0)),
        "videoid": video_id
    }

st.title("🔍 안전한 YouTube 영상 검색기")

query = st.text_input("검색어", "워크맨")
max_results = st.slider("검색 결과 개수", 1, 50, 10)

if st.button("검색 시작"):
    items = search_youtube(query, max_results)
    results = []
    for item in items:
        video_id = item["id"]["videoid"]
        video_detail = get_video_details(video_id)
        results.append(video_detail)

    if not results:
        st.warning("검색 결과가 없습니다.")
    else:
        df = pd.DataFrame(results)
        df["영상링크"] = df["videoid"].apply(lambda x: f"https://www.youtube.com/watch?v={x}")

        st.write("검색 결과")
        for _, row in df.iterrows():
            st.image(row["thumbnail"], width=120)
            st.markdown(f"[{row['title']}]({row['영상링크']}) \n조회수: {row['viewCount']:,} \n채널: {
row['channel']} \n업로드일: {row['publishedAt']}")
            st.markdown("---")

```

7. 활용 목적 및 기대 효과

- 워크맨 채널 관련 콘텐츠 트렌드 실시간 모니터링

- 신규 콘텐츠 소재 발굴 및 인기 동영상 분석 지원
- 마케팅 전략 수립을 위한 데이터 기반 인사이트 확보
- IP 영향력 확대 및 바이럴 확산 효과 극대화