포팅 매뉴얼

1. 개발환경

1.1 Frontend

- Node JS 20.10.0
- React 18.2.0
    - Redux 9.1.0
    - Persist 6.0.0
- Axios 1.6.8
- TypeScript 4.9.5

1.2 Backend

- Java
    - Gradle 8.6
    - Netty 4.1.107
    - Spring Boot 3.2.4
        - Spring Data JPA 3.2.4
        - Spring Data MongoDB 4.2.4
        - Spring Web
        - Lombok 1.18.30
        - Spring Security 6.2.3
    - Spring Cloud 4.1.0
        - Eureka Server 2.0.1
        - Eureka Client 2.0.1
    - mySQL Connector 8.3.0

    - RabbitMQ 5.19.0
    - Prometheus 0.16.0
    - Zipkin
        - aws 0.23.5
        - brave 5.16.0
        - reporter2 2.16.3

1.3 Server

- Nginx
- Docker
- Jenkins

1.4 DataBase

- MySQL 8.3.0
- MongoDB 7.0.6

2. EC2 세팅

2.1 EC2 Port

Service	Port
React	
Client	3000
Spring Cloud	
Discovery-Service	8761
Config-Service	8763
Gateway-Service	8000
Word-Service	8081
Script-Service	8082
Test-Service	8084
Category-Service	8085
User-Service	8086
Video-Service	8087
FastAPI	
Language-Service	8778
trans-Service	8779
Monitoring & CICD & Log & MQ & Analysis	
Zipkin	9411
Grafana	13000
Prometheus	19090
Rabbitmq	5672
Jenkins	9000
Sonarqube	7199
DB	
Mysql	3306
MongoDB	27017

Service	Port
Redis	6379

2.2 Jenkins Pipeline

```
pipeline {
    agent any

    # 빌드 결과를 5회까지 저장
    options {
        buildDiscarder(logRotator(numToKeepStr: '5'))
    }

    environment {
        def Author_ID = sh(script: "git show -s --pretty=%an", returnStdout: true).trim()
        def Git_Message = sh(script: "git show -s --pretty=%s", returnStdout: true).trim()
        def Git_Branch = sh(script: "git branch --show", returnStdout: true).trim()

        DOCKERHUB_CREDENTIALS = 'credential'
        String docker_hub = "docker hub repository"
        dockerImage = ''
        String docker_name = "docker container name"

        # image version
        int build_id = Integer.parseInt("${env.BUILD_ID}")
        int version = build_id.intdiv(10).plus(build_id.mod(10).div(10))
    }

    tools {
        gradle 'gradle_'
    }

    stages {
        stage('clone') {
            steps {
                echo 'service clone'
                # git branch clone - credential 작성
            }
        }

        stage('SonarQube Analysis') {
            steps {
                dir('user_service'){
                    withSonarQubeEnv('sonar') {
                                # gradlew 권한 부여 및 정적 분석 툴 사용
                        sh 'chmod +x gradlew && ./gradlew sonar'
                    }
                }
            }
        }

        stage('Build'){
            steps{
                dir('user_service'){
                    # gradle build (-x check - test 안함)
                    sh 'gradle clean build -x check'
                }
```

```
            }
        }

        stage('Docker Build') {
            steps {
                echo 'service build'
                dir('foler name'){
                    script {
                          # docker image build
                        dockerImage = docker.build docker_hub
                    }
                }
            }
        }

        stage('Login') {
            steps {
                script {
                    # docker image push
                    'docker login with credential'{
                        dockerImage.push("latest")
                        dockerImage.push(version)
                    }
                }
            }
        }
        stage('stop prev container') {
            steps {
                script {
                    try {
                        # active container stop / remove & image clean
                        sh '''
                            docker stop ${docker_name}
                            docker rm ${docker_name}
                            docker rmi $(docker images | egrep "${docker_name}" | awk '{print $3
                            yes | sudo docker image prune
                            sudo docker system prune -f
                           '''
                    } catch (Exception e) {
                        echo 'no prev container'
                    }
                }
            }
        }

        stage('Deploy') {
            steps {
                sh '''
                    docker run -d --name ${docker_name} --network cloud -p port:port ${docker_hu
                   '''
            }
        }
    }


    post {
        success {
            script {
```

```
                mattermostSend (
                    color: '#B3A7DC',
                    message: "### :agree_pepe:  Success  ${env.JOB_NAME} #${env.BUILD_NUMBER} \n
                )
            }
        }

        failure {
            script {
                mattermostSend (
                    color: '#F17845',
                    message: "### :pepe_9: Fail  ${env.JOB_NAME} #${env.BUILD_NUMBER} \n *** \n
                )
            }
        }
    }
}
```

2.3 Nginx

/etc/nginx/site-available/default.conf

```
server {
                # 80 port listen -> 80은 nginx default port
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/html;

        index index.html index.htm index.nginx-debian.html;


        server_name j10b107.p.ssafy.io;

                # / 아래로 오는 요청은 3000번 port로 전송 -> 3000 port는 Front
        location / {
                proxy_pass http://localhost:3000;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host $http_host;
        }


        # /api 아래로 오는 요청은 8000번 port 로 전송 -> 8000은 gateway port
        location /api {

                #  "/api" 접두사를 제거함
                # rewrite ^/api(.*) $1 break;
            proxy_pass http://localhost:8000;

                        # 실제 접속자의 IP를 X-Real-IP 헤더 넣어서 전송.
                        # remote_addr : 요청한 클라이언트 주소
                        # X-Forwarded-For와 동일하게 Client IP를 확인하기 위해 사용하는 헤더
            proxy_set_header X-Real-IP $remote_addr;

            # 프록시나 로드 밸랜서를 통해 들어온 요청에서 클라이언트 원 IP 주소를 확인하기 위해 사용하는 헤더
                        # 프록시 헤더값을 변조할 수 있음
```

```
                        # X-Forwarded-For 만 사용할 경우 변조의 위험이 있으므로, X-Real-IP를 같
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        #HTTP Request 의 Host 헤더값
                        #http 요청이 들어 왔을 시 호스트 명
        proxy_set_header Host $http_host;

                        # HTTPS 서버 블록 내에서 사용할 경우 프록시 서버의 각 HTTP 응답이 HTTPS로
        proxy_set_header X-Forwared-Proto $scheme;

                        # 백엔드 서버에 의해 촉발된 리다이렉션에 대해 로케이션 HTTP 헤더에 나타나는 U
                        # off : 리다이렉션은 설정 된 그대로 전달
        proxy_redirect off;
    }

        # ssl 인증서 적용 (아래 경로는 ssl 인증 키 존재)
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/p.ssafy.io/fullchain.pem;
        ssl_certificate_key /etc/letsencrypt/live/p.ssafy.io/privkey.pem;
}

server {
                # host가 if 조건에 맞으면 https로 redirect
        if ($host = j10b107.p.ssafy.io) {
                return 301 https://$host$request_uri;
        } # managed by Certbot

        listen 80;

        server_name j10b107.p.ssafy.io;

        return 404; # managed by Certbot
}
```

2.4 Docker Compose

Docker Compose (DB)

```
version: '3'
services:
    mysql:
        image: mysql
        container_name: mysql
        # 컨테이너 실행 시 재시작
        restart: always
        # 포트 설정(externel:internel)
        ports:
            - 3306:3306
        # 볼륨 설정
        volumes:
            - ./mysql/data:/var/lib/mysql
        # 환경 변수 설정
        environment:
            MYSQL_ROOT_PASSWORD: ssafy
            TZ: Asia/Seoul
        # 명령어 설정
        command:
```

```
            - --character-set-server=utf8mb4
            - --collation-server=utf8mb4_unicode_ci
        networks:
            - cloud

    mongodb:
        image: mongo
        container_name: mongodb
        restart: always
        ports:
            - 27017:27017
        volumes:
            - ./mongodb/data:/data/db
        environment:
            - MONGO_INITDB_ROOT_USERNAME=ssafy
            - MONGO_INITDB_ROOT_PASSWORD=b107
        networks:
            - cloud

    redis:
        image: redis
        container_name: redis
        hostname: redis
        restart: always
        ports:
            - 6379:6379
        volumes:
            - ./redis/data:/data
            - ./redis/conf/redis.conf:/etc/redis/redis.conf
        command:
            - redis-server /etc/redis/redis.conf
            - redis-server --requirepass ssafyb107 --port 6379
        networks:
            - cloud

networks:
    cloud:
        external: true
```

Docker Compost Util

```
version: '3'
services:
    zipkin:
        image: openzipkin/zipkin
        container_name: zipkin
        ports:
            - 9411:9411
        restart: always
        # ****************** zipkin 로그를 sql에 저장하려면 openzipkin에서 빌드해놓은 sql 이미지도 정의해야
        # environment:
        # - STORAGE_TYPE=mysql # log 저장 storage
        # - MYSQL_DB=zipkin # db name
        # - MYSQL_HOST=mysql # db host
        # - MYSQL_USER=root # db id
        # - MYSQL_PASS=ssafy # db pwd
        # - MYSQL_TCP_PORT=3306 # db port
        # - MYSQL_MAX_CONNECTIONS=10 (default)
```

```
        # - MYSQL_USER_SSL=false (default)
        networks:
            - cloud

    rabbitmq:
        image: rabbitmq:3.12.13-management-alpine
        container_name: rabbitmq
        volumes:
            - rabbitmq_volume_config:/etc/rabbitmq/ # 설정파일 mount
            - rabbitmq_volume_data:/var/lib/rabbitmq/ # data 파일 mount
            - rabbitmq_volume_log:/var/log/rabbitmq/ # log 파일 mount
        restart: always
        ports:
            - '5672:5672' # rabbitmq port
            - '15672:15672' # rabbitmq gui port
        environment:
            RABBITMQ_ERLANG_COOKIE: 'RabbitMQ-Cookies' # Erlang cookie를 통해
            RABBITMQ_DEFAULT_USER: 'ssafy' # id
            RABBITMQ_DEFAULT_PASS: 'b107' # pwd
        networks:
            - cloud

    prometheus:
        image: prom/prometheus
        container_name: prometheus
        volumes:
            - ./prometheus/config:/etc/prometheus # config folder 위치 지정
            - ./prometheus/config/prometheus.yml:/etc/prometheus/prometheus.yml # 설청 yml 파일 지
            - ./prometheus/volume:/prometheus # prometheus의 data volume 지정
        ports:
            - 19090:9090 # 접근 포트 설정 (컨테이너 외부:컨테이너 내부)
        command:
            - '--web.enable-lifecycle' # web.enalbe-lifecycle은 api 재시작없이 설정파일들을 reload
            - '--storage.tsdb.retention=90d' # prometheus의 데이터 저장 기간이 15일인데 90일로 지정
            - '--config.file=/etc/prometheus/prometheus.yml' # 설정 파일 위치 지정
        restart: always
        networks:
            - cloud

    grafana:
        image: grafana/grafana
        container_name: grafana
        ports:
            - '13000:3000'
        environment:
            - GF_SECURITY_ADMIN_USER=ssafy # id
            - GF_SECURITY_ADMIN_PASSWORD=b107 # pw
        restart: always
        user: '$UID:$GID'
        volumes:
            - ./grafana:/var/lib/grafana
        depends_on:
            - prometheus
            - loki
        networks:
            - cloud

    loki:
```

```
        image: grafana/loki:latest
        ports:
            - '3100:3100'
        command: -config.file=/etc/loki/local-config.yaml
        networks:
            - cloud

    promtail:
        image: grafana/promtail:latest
        volumes:
            - ./logs/backend:/logs
            - ./promtail-config.yml:/etc/promtail/config.yml
        command: -config.file=/etc/promtail/config.yml
        depends_on:
            - loki
        networks:
            - cloud

# 생성할 볼륨의 위치를 docker에게 위임 - volume 위치를 rabbitmq가 못잡아서 오류 발생했었음
volumes:
    rabbitmq_volume_config:
    rabbitmq_volume_data:
    rabbitmq_volume_log:

networks:
    cloud:
        external: true
```

2.5 Prometheus

Service Registry

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  evaluation_interval: 15s
alerting:
  alertmanagers:
  - follow_redirects: true
    enable_http2: true
    scheme: http
    timeout: 10s
    api_version: v2
    static_configs:
    - targets: []
scrape_configs:
- job_name: prometheus
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
```

```
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /metrics
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - localhost:19090
- job_name: grafana
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /metrics
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - grafana:13000
- job_name: apigateway-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - gateway:8000
- job_name: word-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /word-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
```

```
  static_configs:
  - targets:
    - word-service:8081
- job_name: script-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /script-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - script-service:8081
- job_name: test-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /test-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - test-service:8084
- job_name: category-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /category-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - category-service:8085
- job_name: user-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
```

```
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /user-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - user-service:8086
- job_name: video-service
  honor_timestamps: true
  track_timestamps_staleness: false
  scrape_interval: 15s
  scrape_timeout: 10s
  scrape_protocols:
  - OpenMetricsText1.0.0
  - OpenMetricsText0.0.1
  - PrometheusText0.0.4
  metrics_path: /video-service/actuator/prometheus
  scheme: http
  enable_compression: true
  follow_redirects: true
  enable_http2: true
  static_configs:
  - targets:
    - video-service:8087
```

3. 외부 서비스

3.1 구글 소셜 로그인

> **구글 로그인 api**

3.1.1 Google Developers Setting

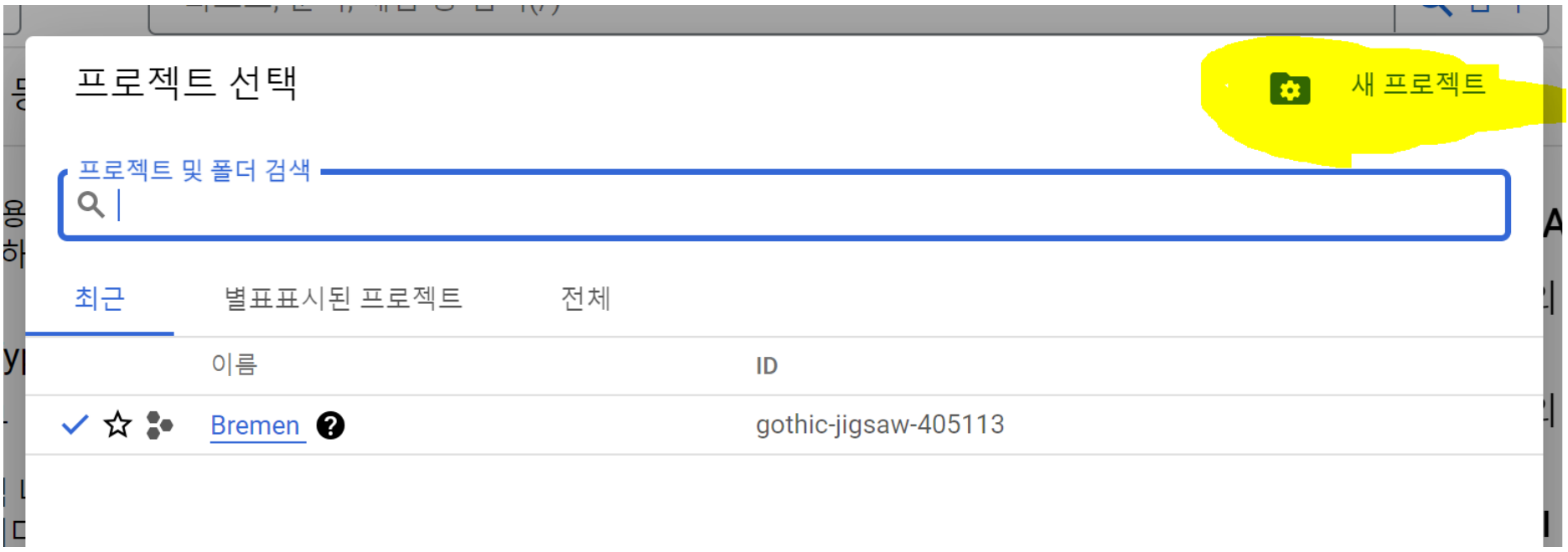
1. OAuth 동의 화면 검색 후 이동
2. Google Cloud에서 새 프로젝트 생성
3. 필수 정보만 입력 후 범위 추가

API **OAuth 동의 화면**

API 및 서비스

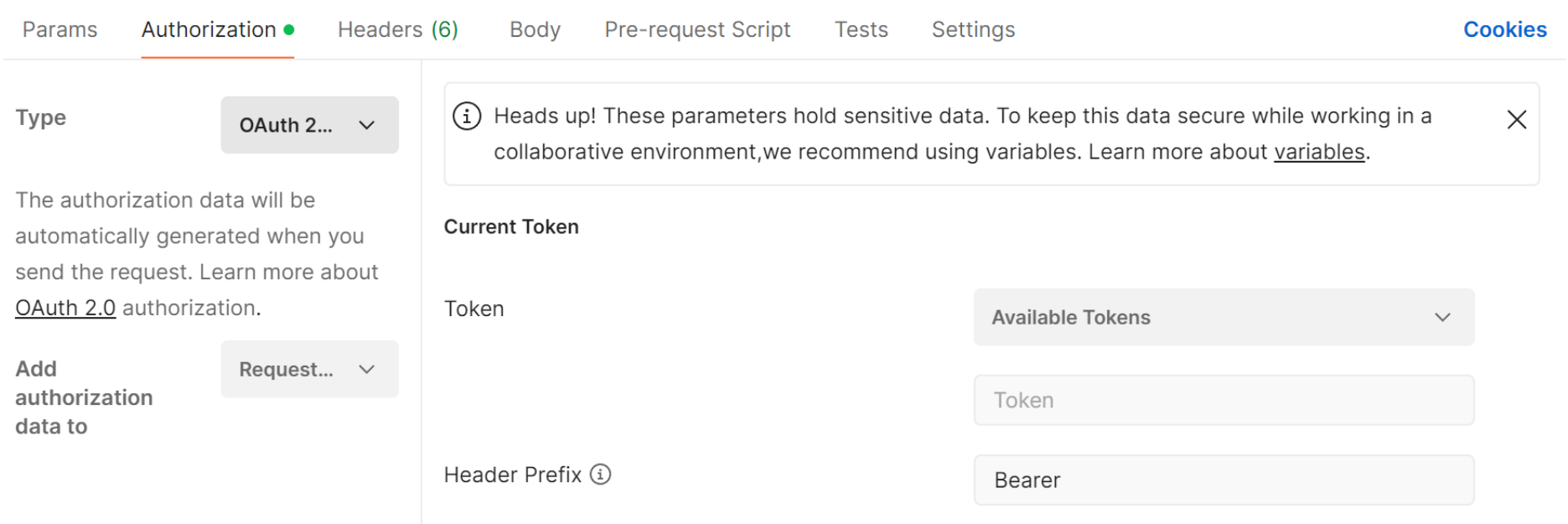
유형: 제품 또는 페이지

제품: API 및 서비스



3.1.2 Json

1. Json 다운로드 후 파일 열기
2. Postman 실행 후 OAuth 2.0 선택
3. 항목 채우고 토큰 생성하기



- token name : 아무값이나
- Callback URL : 구글 클라우드 플랫폼에서 프로젝트 생성시 넣은 값
- Auth URL : json 파일의 auth_uri
- Access Token URL : json 파일의 token_uri
- Scope : 나는 https://www.googleapis.com/auth/cloud-platform 값을 넣음. 필수값이다

3.2 유튜브 API

3.2.1 유튜브 API 사용내역

- 채널의 재생목록 조회

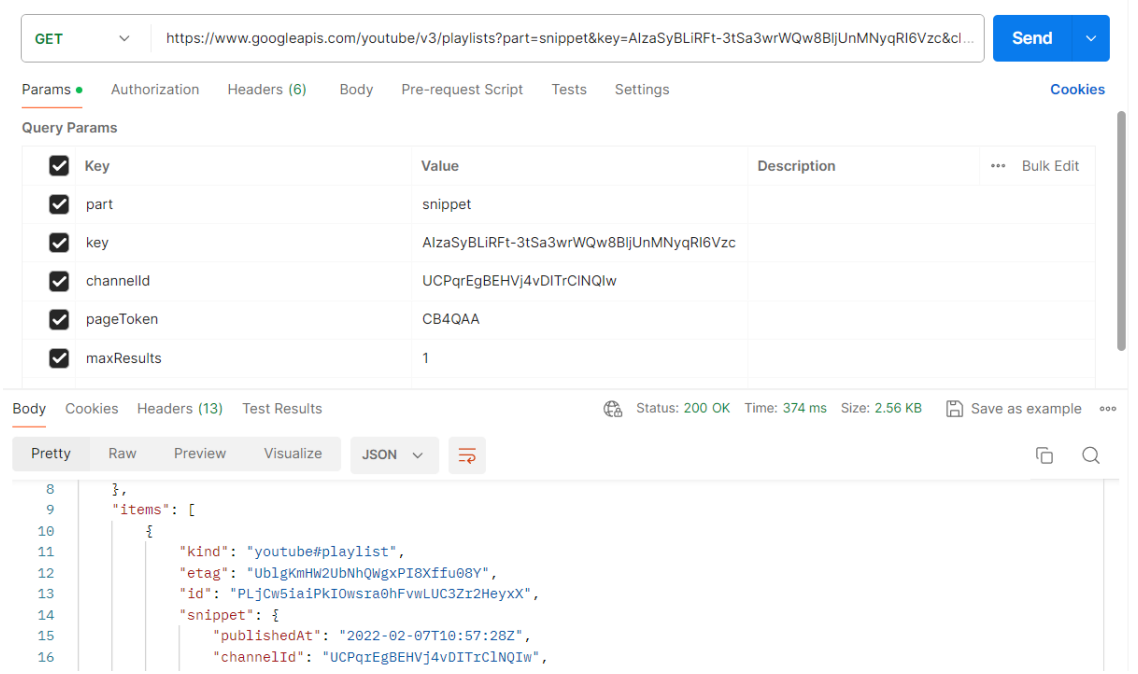
  https://developers.google.com/youtube/v3/docs/playlists/list?hl=ko
- 재생목록 내의 영상 조회

  https://developers.google.com/youtube/v3/docs/playlistItems/list?hl=ko
- 검색 결과 조회

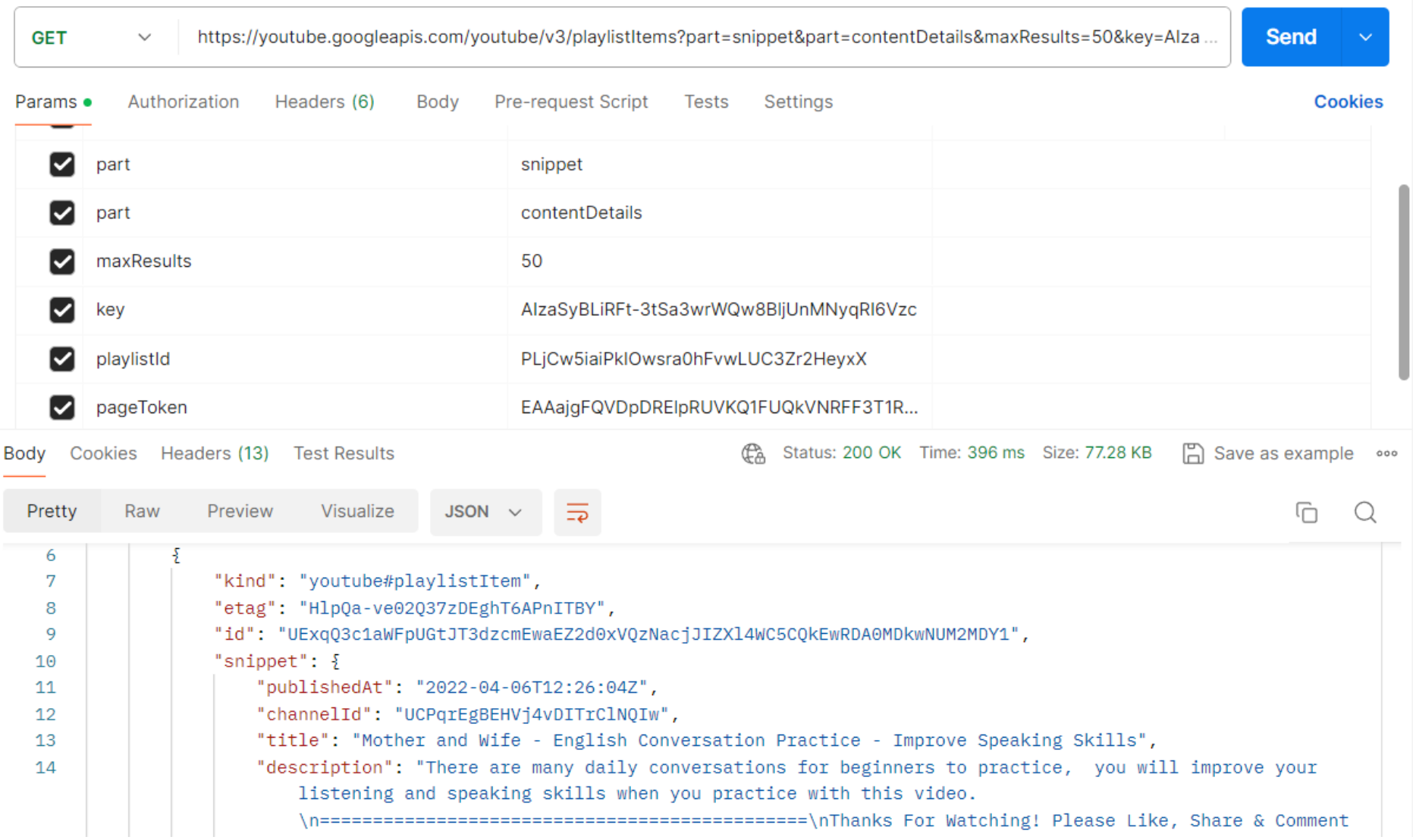
  https://developers.google.com/youtube/v3/docs/search/list?hl=ko

3.2.2 유튜브 영상 조회

1. 유튜브 채널의 재생목록 아이디 추출



2. 재생목록 아이디로 재생목록 내의 영상 추출



4. DB 덤프 파일

```
-- MySQL dump 10.13  Distrib 8.0.36, for Win64 (x86_64)
--
-- Host: j10b107.p.ssafy.io    Database: teddybear
-- ------------------------------------------------------
-- Server version   8.3.0

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
```

```
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `bookmark_video`
--

DROP TABLE IF EXISTS `bookmark_video`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `bookmark_video` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `user_id` bigint NOT NULL,
  `video` bigint NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FKmly48yl4y6ej83r1xds4ddpcs` (`video`),
  CONSTRAINT `FKmly48yl4y6ej83r1xds4ddpcs` FOREIGN KEY (`video`) REFERENCES `video` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `bookmark_word`
--

DROP TABLE IF EXISTS `bookmark_word`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `bookmark_word` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `user_id` bigint DEFAULT NULL,
  `word` bigint NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FKkje87nbm1sbnpc4smom3q82g9` (`word`),
  CONSTRAINT `FKkje87nbm1sbnpc4smom3q82g9` FOREIGN KEY (`word`) REFERENCES `word` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `category`
--

DROP TABLE IF EXISTS `category`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `category` (
  `category_seq` int NOT NULL AUTO_INCREMENT,
  `category_name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`category_seq`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;



--
-- Table structure for table `daily_word`
```

```
--

DROP TABLE IF EXISTS `daily_word`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `daily_word` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `eng` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `kor` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `part` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `tier` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `word_id` bigint DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=82981 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `laundry_entity`
--

DROP TABLE IF EXISTS `laundry_entity`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `laundry_entity` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `video_description` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_grade` varchar(5) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_id` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_playtime` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_thumbnail` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_title` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_transcript` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_url` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_l4uks0bs055rh3yb91h3107al` (`video_id`)
) ENGINE=InnoDB AUTO_INCREMENT=776 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `note`
--

DROP TABLE IF EXISTS `note`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `note` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `note` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `note_date` datetime(6) NOT NULL,
  `user_id` bigint NOT NULL,
  `video` bigint NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FKfg3j4yalrdb27ua16tov9ypfh` (`video`),
  CONSTRAINT `FKfg3j4yalrdb27ua16tov9ypfh` FOREIGN KEY (`video`) REFERENCES `video` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=24 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

```
-- Table structure for table `script`
--

DROP TABLE IF EXISTS `script`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `script` (
  `script_seq` bigint NOT NULL AUTO_INCREMENT,
  `content` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_id` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`script_seq`)
) ENGINE=InnoDB AUTO_INCREMENT=536281 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `tier`
--

DROP TABLE IF EXISTS `tier`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `tier` (
  `level` int NOT NULL DEFAULT '1',
  `level_exp` bigint DEFAULT '0',
  `tier_exp` bigint DEFAULT '0',
  `tier_seq` bigint NOT NULL AUTO_INCREMENT,
  `user_seq` bigint DEFAULT NULL,
  `tier_name` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`tier_seq`),
  UNIQUE KEY `UK_6ce9p1pnq837qljn5rpkxe5rg` (`user_seq`),
  CONSTRAINT `FKnhmmjmauohk651nyeo3qspf5m` FOREIGN KEY (`user_seq`) REFERENCES `users` (`user_se
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `user_category`
--

DROP TABLE IF EXISTS `user_category`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user_category` (
  `user_category_seq` bigint NOT NULL AUTO_INCREMENT,
  `economy` bigint DEFAULT '0',
  `it` bigint DEFAULT '0',
  `life` bigint DEFAULT '0',
  `politics` bigint DEFAULT '0',
  `society` bigint DEFAULT '0',
  `sports` bigint DEFAULT '0',
  `world` bigint DEFAULT '0',
  `user_seq` bigint DEFAULT NULL,
  PRIMARY KEY (`user_category_seq`),
  UNIQUE KEY `UK_ssgf22pmvpx3mmxqtfb3enlu3` (`user_seq`),
  CONSTRAINT `FK454v6vknodr6807kw80i0yxqe` FOREIGN KEY (`user_seq`) REFERENCES `users` (`user_se
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

```
-- Table structure for table `users`
--

DROP TABLE IF EXISTS `users`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `users` (
  `attendance` int NOT NULL DEFAULT '0',
  `user_seq` bigint NOT NULL AUTO_INCREMENT,
  `video_view_time` datetime(6) NOT NULL,
  `gender` varchar(10) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `birthday` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `concern` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  `nickname` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `refresh_token` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `role` enum('USER','GUEST') COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`user_seq`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `video`
--

DROP TABLE IF EXISTS `video`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `video` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `video_description` text COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_grade` varchar(5) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_id` varchar(100) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_thumbnail` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_time` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_title` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  `video_url` varchar(500) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_k24kj8ek7wcb81i0elhhup34o` (`video_id`)
) ENGINE=InnoDB AUTO_INCREMENT=10058 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `video_category`
--

DROP TABLE IF EXISTS `video_category`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `video_category` (
  `video_category_seq` bigint NOT NULL AUTO_INCREMENT,
  `category` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `video_id` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  PRIMARY KEY (`video_category_seq`)
) ENGINE=InnoDB AUTO_INCREMENT=8212 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
```

```
-- Table structure for table `watch_video`
--

DROP TABLE IF EXISTS `watch_video`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `watch_video` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `user_id` bigint NOT NULL,
  `video_watched` bit(1) NOT NULL,
  `video` bigint NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FKg3wsnoyj63yukjusduo5twgtc` (`video`),
  CONSTRAINT `FKg3wsnoyj63yukjusduo5twgtc` FOREIGN KEY (`video`) REFERENCES `video` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=75 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Table structure for table `word`
--

DROP TABLE IF EXISTS `word`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `word` (
  `tier` varchar(2) COLLATE utf8mb4_unicode_ci NOT NULL,
  `id` bigint NOT NULL AUTO_INCREMENT,
  `part` varchar(20) COLLATE utf8mb4_unicode_ci NOT NULL,
  `eng` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  `kor` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=8582 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
/*!40101 SET character_set_client = @saved_cs_client */;
```