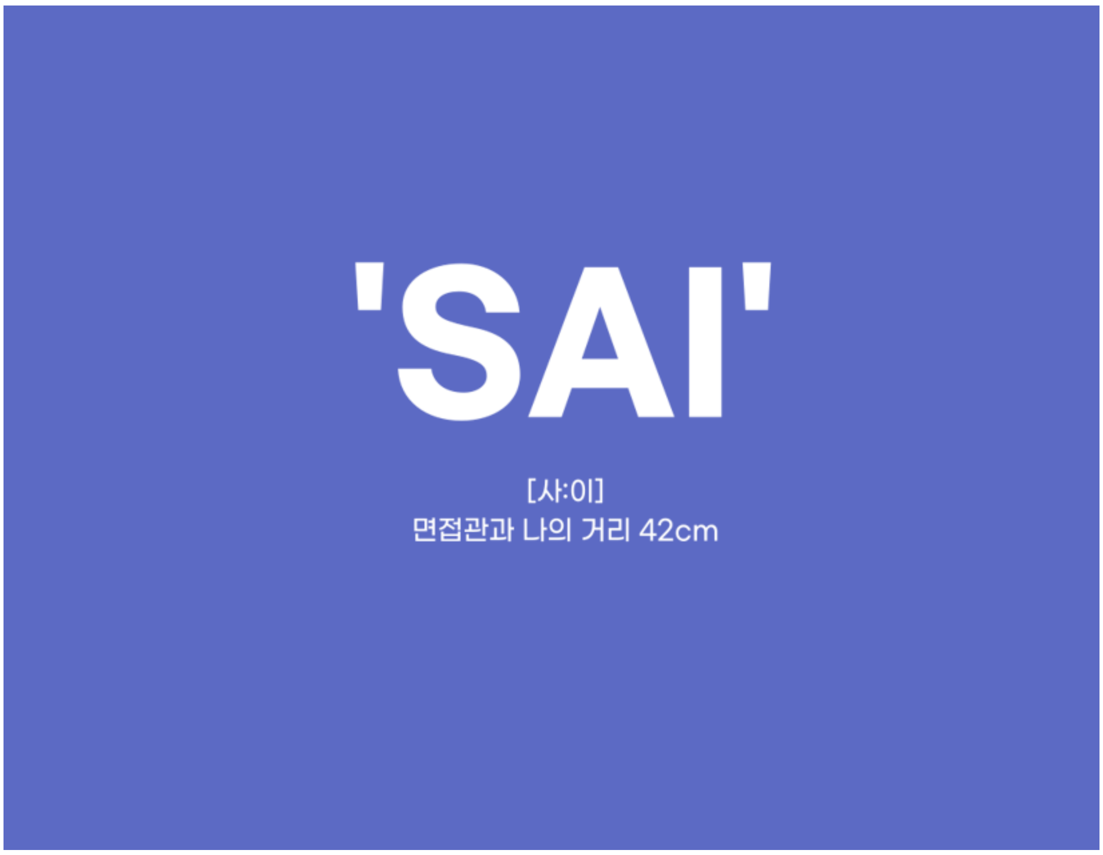
포팅매뉴얼



삼성 청년 SW 아카데미 7기

담당 컨설턴트 : 김성재

최인호 김지수 윤형준 지근 조덕희 심재서

개발 환경

IDE

Intellij `2022.1.3`

Visual Studio Code `1.70`

MySQL Workbench

Database

MySQL `8.0.29`

FrontEnd

HTML5, CSS3, JavaScript (ES6)

Vue `3.2.13`

Vuex `4.0.`

Node.js `16.16`

…

BackEnd

Java 11 `OpenJDK-11.0.15` (https://jdk.java.net/java-se-ri/11)

Gradle `7.5`

spring-boot `2.7.2`

spring-boot-data-jpa

Spring Security

jjwt `0.9.1`

lombok `1.18.24`

jave `1.0.2` - 외부 라이브러리 (https://www.sauronsoftware.it/projects/jave/)

- 위치 : /libs/jave-1.0.2.jar

WebRTC

Openvidu

Infra

AWS EC2 - deploy server (Ubuntu 20.04 LTS x86_64 GNU/Linux)

AWS S3 - file server

Nginx `1.18.0`

Gitlab, Mattermost, Notion

Jira, Figma

Tool

MobaXterm

Postman

외부서비스

Teachable Machine

STT

TTS

Face API (https://github.com/justadudewhohacks/face-api.js/#face-api.js-for-the-browser)

Nginx

Nginx 설치

```
# 서버의 패키지 목록을 업데이트 합니다.
sudo apt-get update
sudo apt upgrade
sudo apt autoremove

# Nginx를 설치합니다.
sudo apt install nginx

# Nginx를 실행합니다.
sudo service start nginx

# Nginx 상태를 확인합니다.
sudo service status nginx
```

SSL 인증서 적용

```

# apt-get update
# apt-get install software-properties-common
# add-apt-repository universe
# add-apt-repository ppa:certbot/certbot
# apt-get update

# certbot 을 설치합니다.
sudo apt-get install certbot python3-certbot-nginx
certbot --nginx

# ACME 서버 등록에 동의합니다.
(A)gree/(C)ancel: A

# 이메일 주소에 대한 공유를 설정합니다.
(Y)es/(N)o: Y

# SSL을 적용할 도메인 주소를 입력합니다.
(Enter 'c' to cancel): i7c206.p.ssafy.io www.i7c206.p.ssafy.io

# 80 포트로의 모든 트래픽을 SSL로 Redirecting 합니다
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2

# certificate and chain 저장 경로
/etc/letsencrypt/live/i7c206.p.ssafy.io/fullchain.pem

# key file 저장 경로
/etc/letsencrypt/live/i7c206.p.ssafy.io/privkey.pem

# 설정파일을 확인합니다.
vi /etc/nginx/sites-available/default

# 설정파일에 프록시패스를 추가합니다.
location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                #try_files $uri $uri/ =404;
                proxy_pass http://localhost:8080;
                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;

        }
```

```
# 다음은 default 파일 전문입니다.

server {
        listen 443 ssl;
        listen [::]:443 ssl ipv6only=on;

        ssl_certificate /etc/letsencrypt/live/i7c206.p.ssafy.io/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/i7c206.p.ssafy.io/privkey.pem; # managed by Certbo
        ssl_prefer_server_ciphers on;

        root /home/ubuntu/vue/dist;
        index index.html
        server_name i7c206.p.ssafy.io;
        location / {
                try_files $uri $uri/ /index.html;
                proxy_pass http://localhost:8081;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header Host $host;
        }
}

server {
        listen 80;
        server_name i7c206.p.ssafy.io;
        return 301 https://$host$request_uri;
}
```

스프링부트 SSL 적용

certbot을 이용해 발급받은 키(.pem)를 openssl을 이용해 .p12파일로 변경

```
# live 폴더에 접근하기 위해 관리자 권한으로 변경
sudo su

# key가 존재하는 디렉토리로 이동
xcd /etc/letsencrypt/live/<인증서 발급 시 설정한 도메인 폴더>/

# pem파일을 이용해 p12파일 생성 명령어 입력 시 비밀번호를 설정한다.
openssl pkcs12 -export -in fullchain.pem -inkey privkey.pem -out keystore.p12 -name tomcat -CAfile chain.pem -caname root

# 만약 pem 파일을 백업해두고 싶다면 아래 command를 입력한다.
cp -r * <resources 폴더 경로>

# resources 폴더로 돌아가 잘 생성되었는지 확인해본다.

# 관리자권한 종료
exit
```

keystore.p12 파일을 /src/main/resources에 이동

```
# /etc/letsencrypt/live/<인증서 발급 시 설정한 도메인 폴더>/안에 파일을 이동시키기 위해 권한 설정이 필요함
chmod -R 777 <인증서 발급 시 설정한 도메인 폴더>
```

application.properties 설정

```
#SSL
server.ssl.key-store=classpath:keystore.p12
server.ssl.key-store-type=PKCS12
server.ssl.key-store-password=내가정한password
```

프론트엔드 빌드 방법

```
#front 폴더로 이동
# cd /S07P12C206/front
npm install
# 빌드 경로 : front 폴더/dist
#(SAI 프로젝트 : /S07P12C206/front/dist)
npm run build
```

백엔드 빌드 방법

MySQL

```
# apt 패키지를 업데이트하고 MySQL을 설치합니다.
sudo apt-get update
sudo apt-get install mysql-server

# MySQL을 실행합니다.
sudo systemctl start mysql.service

# MySQL에 접속합니다.
sudo mysql -u root -p

# 사용할 계정을 생성하고 권한을 부여합니다.
CREATE USER 계정명@'%' IDENTIFIED BY 비밀번호;
GRANT ALL PRIVILEGES ON *.* 계정이름@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES

# 덤프 파일을 적용합니다.
mysql -u 계정명 -p sai_app < ./sai.sql
```

Google cloud storage 환경변수 설정

```
export GOOGLE_APPLICATION_CREDENTIALS="/home/ubuntu/psychic-habitat-358714-81bd61376e0d.json"
```

Google cloud storage 사용을 위한 JSON 파일

- resources 폴더 안에 업로드해야 함

```
{
  "type": "service_account",
  "project_id": "psychic-habitat-358714",
  "private_key_id": "81bd61376e0dc616e1f28fe5dcc5beeab98af05e",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEuwIBADANBgkqhkiG9w0BAQEFAASCBKUwggShAgEAAoIBAQC6V7ryTPQC6T1E\n7EuZcVddoEDvSBLd8zaGo
  "client_email": "sai2-444@psychic-habitat-358714.iam.gserviceaccount.com",
  "client_id": "115993588762660473939",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/sai2-444%40psychic-habitat-358714.iam.gserviceaccount.com
}
```

Openvidu Server 배포

Docker Engine

```
# apt 패키지를 업데이트하고 하위 패키지를 설치합니다.
sudo apt-get update
sudo apt-get install \ ca-certificates \ curl \ gnupg \ lsb-release

# Docker’s official GPG key를 추가
sudo mkdir -p /etc/apt/keyrings $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/do

# 레파지토리 생성
echo \ "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \ $(ls

# 도커 엔진을 설치
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

Docker

1. Install Docker Desktop on Linux (DEB package) 설치 (ubuntu용)
2. 설치 후 해당 DEB 파일을 mobaXterm을 통해 AWS ec2 서버로 업로드 (/home/ubuntu)

```
sudo apt-get update
sudo apt-get install ./docker-desktop-4.11.1-amd64.deb
```

Docker-compose 설치

```
sudo apt-get install docker-compose-plugin
```

OpenVidu Server 배포

```
# 루트 권한 필요
sudo su

# OpenVidu 설치 권장 폴더는 /opt
cd /opt

# 설치 스크립트를 다운로드 한 후, 실행
curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | bash

cd openvidu

# .env 파일 설정
```

```
nano .env

# .env 파일 설정 후, openvidu server 실행
./openvidu start

# 종종 에러가 발생해서 재부팅해주기
./openvidu stop
./openvidu start
```

.env 설정

```
# OpenVidu configuration
# ----------------------
# Documentation: https://docs.openvidu.io/en/stable/reference-docs/openvidu-config/
# Domain name. If you do not have one, the public IP of the machine.
# For example: 198.51.100.1, or openvidu.example.com
# 도메인
DOMAIN_OR_PUBLIC_IP=i7c206.p.ssafy.io

# OpenVidu SECRET used for apps to connect to OpenVidu server and users to access to OpenVidu Dashboard
# 원하는 비밀번호
OPENVIDU_SECRET=MY_SECRET

# Certificate type:
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
# 원하는 이메일
LETSENCRYPT_EMAIL=delpho@naver.com

# Proxy configuration
# If you want to change the ports on which openvidu listens, uncomment the following lines

# Allows any request to http://DOMAIN_OR_PUBLIC_IP:HTTP_PORT/ to be automatically
# redirected to https://DOMAIN_OR_PUBLIC_IP:HTTPS_PORT/.
# WARNING: the default port 80 cannot be changed during the first boot
# if you have chosen to deploy with the option CERTIFICATE_TYPE=letsencrypt
 HTTP_PORT=8082

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
 HTTPS_PORT=8083

# Whether to enable recording module or not
OPENVIDU_RECORDING=true

# Use recording module with debug mode.
OPENVIDU_RECORDING_DEBUG=false

# Openvidu Folder Record used for save the openvidu recording videos. Change it
# with the folder you want to use from your host.
OPENVIDU_RECORDING_PATH=/opt/openvidu/recordings

# System path where OpenVidu Server should look for custom recording layouts
OPENVIDU_RECORDING_CUSTOM_LAYOUT=/opt/openvidu/custom-layout

# if true any client can connect to
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT/recordings/any_session_file.mp4
# and access any recorded video file. If false this path will be secured with
# OPENVIDU_SECRET param just as OpenVidu Server dashboard at
# https://OPENVIDU_SERVER_IP:OPENVIDU_PORT
# Values: true | false
OPENVIDU_RECORDING_PUBLIC_ACCESS=false

# Which users should receive the recording events in the client side
# (recordingStarted, recordingStopped). Can be all (every user connected to
# the session), publisher_moderator (users with role 'PUBLISHER' or
# 'MODERATOR'), moderator (only users with role 'MODERATOR') or none
# (no user will receive these events)
OPENVIDU_RECORDING_NOTIFICATION=publisher_moderator

# Timeout in seconds for recordings to automatically stop (and the session involved to be closed)
# when conditions are met: a session recording is started but no user is publishing to it or a session
# is being recorded and last user disconnects. If a user publishes within the timeout in either case,
# the automatic stop of the recording is cancelled
# 0 means no timeout
OPENVIDU_RECORDING_AUTOSTOP_TIMEOUT=120

# Maximum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_RECV_BANDWIDTH=1000
```

```
# Minimum video bandwidth sent from clients to OpenVidu Server, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_RECV_BANDWIDTH=300

# Maximum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MAX_SEND_BANDWIDTH=1000

# Minimum video bandwidth sent from OpenVidu Server to clients, in kbps.
# 0 means unconstrained
OPENVIDU_STREAMS_VIDEO_MIN_SEND_BANDWIDTH=300

# true to enable OpenVidu Webhook service. false' otherwise
# Values: true | false
OPENVIDU_WEBHOOK=false

# HTTP endpoint where OpenVidu Server will send Webhook HTTP POST messages
# Must be a valid URL: http(s)://ENDPOINT
#OPENVIDU_WEBHOOK_ENDPOINT=

# List of headers that OpenVidu Webhook service will attach to HTTP POST messages
#OPENVIDU_WEBHOOK_HEADERS=

# List of events that will be sent by OpenVidu Webhook service
# Default value is all available events
OPENVIDU_WEBHOOK_EVENTS=[sessionCreated,sessionDestroyed,participantJoined,participantLeft,webrtcConnectionCreated,webrtcConnectionDes

# How often the garbage collector of non active sessions runs.
# This helps cleaning up sessions that have been initialized through
# REST API (and maybe tokens have been created for them) but have had no users connected.
# Default to 900s (15 mins). 0 to disable non active sessions garbage collector
OPENVIDU_SESSIONS_GARBAGE_INTERVAL=900

# Minimum time in seconds that a non active session must have been in existence
# for the garbage collector of non active sessions to remove it. Default to 3600s (1 hour).
# If non active sessions garbage collector is disabled
# (property 'OPENVIDU_SESSIONS_GARBAGE_INTERVAL' to 0) this property is ignored
OPENVIDU_SESSIONS_GARBAGE_THRESHOLD=3600

# Call Detail Record enabled
# Whether to enable Call Detail Record or not
# Values: true | false
OPENVIDU_CDR=false

# Path where the cdr log files are hosted
OPENVIDU_CDR_PATH=/opt/openvidu/cdr
```

참고

https://docs.openvidu.io/en/stable/deployment/ce/on-premises/

https://velog.io/@djlesque/aws-EC2에-openvidu-배포하기-oyshy1pf

프론트엔드, 백엔드 배포 방법

프론트엔드

```
# npm run build 실행 전, 설치 필수
npm i

# dist (배포 파일)을 만들기 위한 명령어
npm run build
```

백엔드

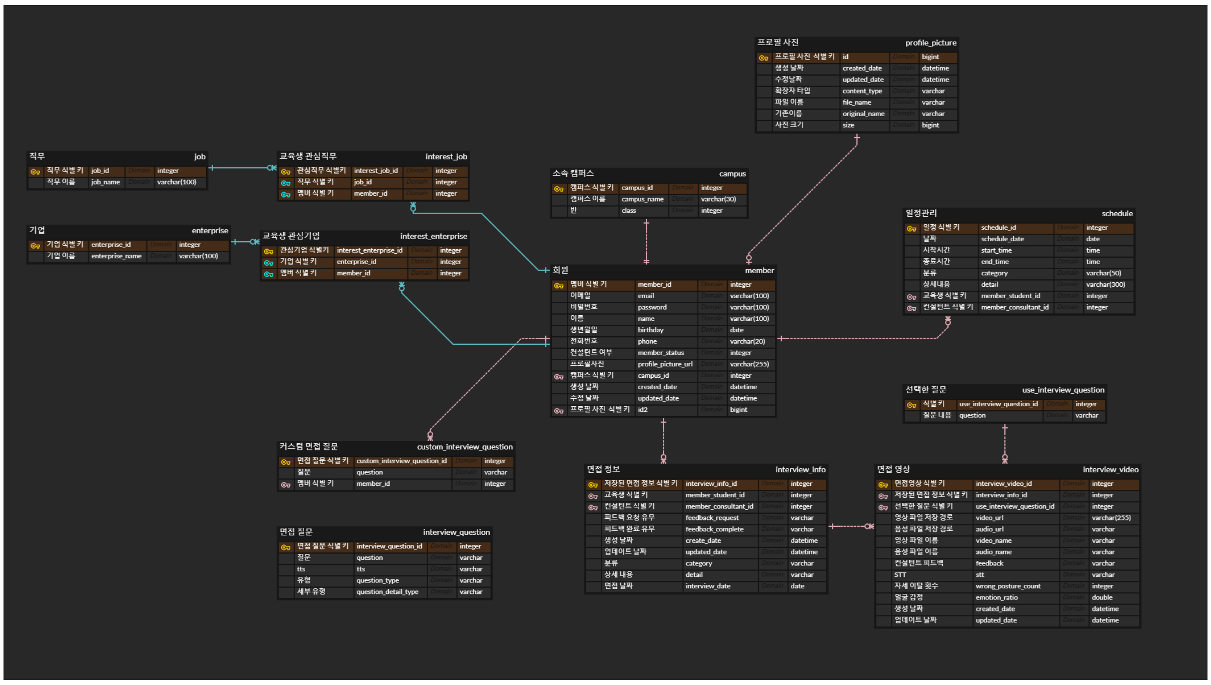
```
1. intellij 우측 gradle 클릭

# 빌드 파일 삭제
2. task>build>clean 클릭

3. task>build>bootJar 클릭

4. 프로젝트 폴더에 build파일이 생성되고, build>libs>sai-0.0.1-SNAPSHOT.jar를 ec2서버에 업로드

# 백그라운드에서 실행하고 싶으면, nohup 명령어 이용
5. java -jar sai-0.0.1-SNAPSHOT &
```

---

application.properties

```
server.port=8081

# MySQL
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/sai_app?serverTimezone=UTC&characterEncoding=UTF-8
spring.datasource.username=sai
spring.datasource.password=ssafysaipjt

# JPA
spring.jpa.show-sql=true
spring.jpa.generate-ddl=true
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.default_batch_fetch_size = 100
spring.jpa.defer-datasource-initialization=true

# File
spring.servlet.multipart.enabled=true
spring.servlet.multipart.location=/data/upload_tmp
spring.servlet.multipart.max-file-size=500MB
spring.servlet.multipart.max-request-size=500MB

# SQL
spring.sql.init.mode=always

# AWS
logging.level.com.amazonaws.util.EC2MetadataUtils= error
cloud.aws.stack.auto=false
cloud.aws.region.static=ap-northeast-2
cloud.aws.credentials.access-key=AKIA347XUSDP6B5I2VEM
cloud.aws.credentials.secret-key=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
cloud.aws.credentials.instance-profile=true
cloud.aws.s3.bucket=sai-project
cloud.aws.s3.bucket.url=https://s3.ap-northeast-2.amazonaws.com/sai-s3

#delete
spring.mvc.hiddenmethod.filter.enabled=true

#ssl
server.ssl.key-store=classpath:keystore.p12
server.ssl.key-store-type=PKCS12
server.ssl.key-store-password=ssafysaipjt

spring.cloud.gcp.storage.credentials.location=classpath:psychic-habitat-358714-81bd61376e0d.json
```

ERD



직무 job

직무 식별 키	job_id	Domain	integer
직무 이름	job_name	Domain	varchar(100)

교육생 관심직무 interest_job

관심직무 식별키	interest_job_id	Domain	integer
직무 식별 키	job_id	Domain	integer
멤버 식별 키	member_id	Domain	integer

기업 enterprise

기업 식별 키	enterprise_id	Domain	integer
기업 이름	enterprise_name	Domain	varchar(100)

교육생 관심기업 interest_enterprise

관심기업 식별키	interest_enterprise_id	Domain	integer
기업 식별 키	enterprise_id	Domain	integer
멤버 식별 키	member_id	Domain	integer

프로필 사진 profile_picture

프로필 사진 식별 키	id	Domain	bigint
생성 날짜	created_date	Domain	datetime
수정날짜	updated_date	Domain	datetime
확장자 타입	content_type	Domain	varchar
파일 이름	file_name	Domain	varchar
기존이름	original_name	Domain	varchar
사진 크기	size	Domain	bigint

소속 캠퍼스 campus

캠퍼스 식별 키	campus_id	Domain	integer
캠퍼스 이름	campus_name	Domain	varchar(30)
반	class	Domain	integer

일정관리 schedule

일정 식별 키	schedule_id	Domain	integer
날짜	schedule_date	Domain	date
시작시간	start_time	Domain	time
종료시간	end_time	Domain	time
분류	category	Domain	varchar(50)
상세내용	detail	Domain	varchar(300)
교육생 식별 키	member_student_id	Domain	integer
컨설턴트 식별 키	member_consultant_id	Domain	integer

회원 member

멤버 식별 키	member_id	Domain	integer
이메일	email	Domain	varchar(100)
비밀번호	password	Domain	varchar(100)
이름	name	Domain	varchar(100)
생년월일	birthday	Domain	date
전화번호	phone	Domain	varchar(20)
컨설턴트 여부	member_status	Domain	integer
프로필사진	profile_picture_url	Domain	varchar(255)
캠퍼스 식별 키	campus_id	Domain	integer
생성 날짜	created_date	Domain	datetime
수정 날짜	updated_date	Domain	datetime
프로필 사진 식별 키	id2	Domain	bigint

선택한 질문 use_interview_question

식별 키	use_interview_question_id	Domain	integer
질문 내용	question	Domain	varchar

커스텀 면접 질문 custom_interview_question

면접 질문 식별 키	custom_interview_question_id	Domain	integer
질문	question	Domain	varchar
멤버 식별 키	member_id	Domain	integer

면접 정보 interview_info

저장된 면접 정보 식별 키	interview_info_id	Domain	integer
교육생 식별 키	member_student_id	Domain	integer
컨설턴트 식별 키	member_consultant_id	Domain	integer
피드백 요청 유무	feedback_request	Domain	varchar
피드백 완료 유무	feedback_complete	Domain	varchar
생성 날짜	create_date	Domain	datetime
업데이트 날짜	updated_date	Domain	datetime
분류	category	Domain	varchar
상세 내용	detail	Domain	varchar
면접 날짜	interview_date	Domain	date

면접 영상 interview_video

면접영상 식별 키	interview_video_id	Domain	integer
저장된 면접 정보 식별 키	interview_info_id	Domain	integer
선택한 질문 식별 키	use_interview_question_id	Domain	integer
영상 파일 저장 경로	video_url	Domain	varchar(255)
음성 파일 저장 경로	audio_url	Domain	varchar
영상 파일 이름	video_name	Domain	varchar
음성 파일 이름	audio_name	Domain	varchar
컨설턴트 피드백	feedback	Domain	varchar
STT	stt	Domain	varchar
자세 이탈 횟수	wrong_posture_count	Domain	integer
얼굴 감정	emotion_ratio	Domain	double
생성 날짜	created_date	Domain	datetime
업데이트 날짜	updated_date	Domain	datetime

면접 질문 interview_question

면접 질문 식별 키	interview_question_id	Domain	integer
질문	question	Domain	varchar
tts	tts	Domain	varchar
유형	question_type	Domain	varchar
세부 유형	question_detail_type	Domain	varchar