



산출물 정리

[기술 스택](#)

[빌드 상세](#)

[백엔드](#)

[GUI](#)

[Command](#)

[프론트엔드](#)

[EC2 설정](#)

[도커 설치](#)

[Mysql 설치 + 설정](#)

[Nginx + SSL 설정](#)

[Openvidu 설치](#)

[외부서비스](#)

[소셜 로그인](#)

[카카오 로그인](#)

[네이버 로그인](#)

[구글 로그인](#)

[SMTP 이메일 인증 설정](#)

[DB](#)

[Mysql 워크벤치 사용방법](#)

[배포](#)

[기타](#)

[ignore된 파일](#)

[*.properties](#)

[index.js](#)

[default.conf](#)

기술 스택

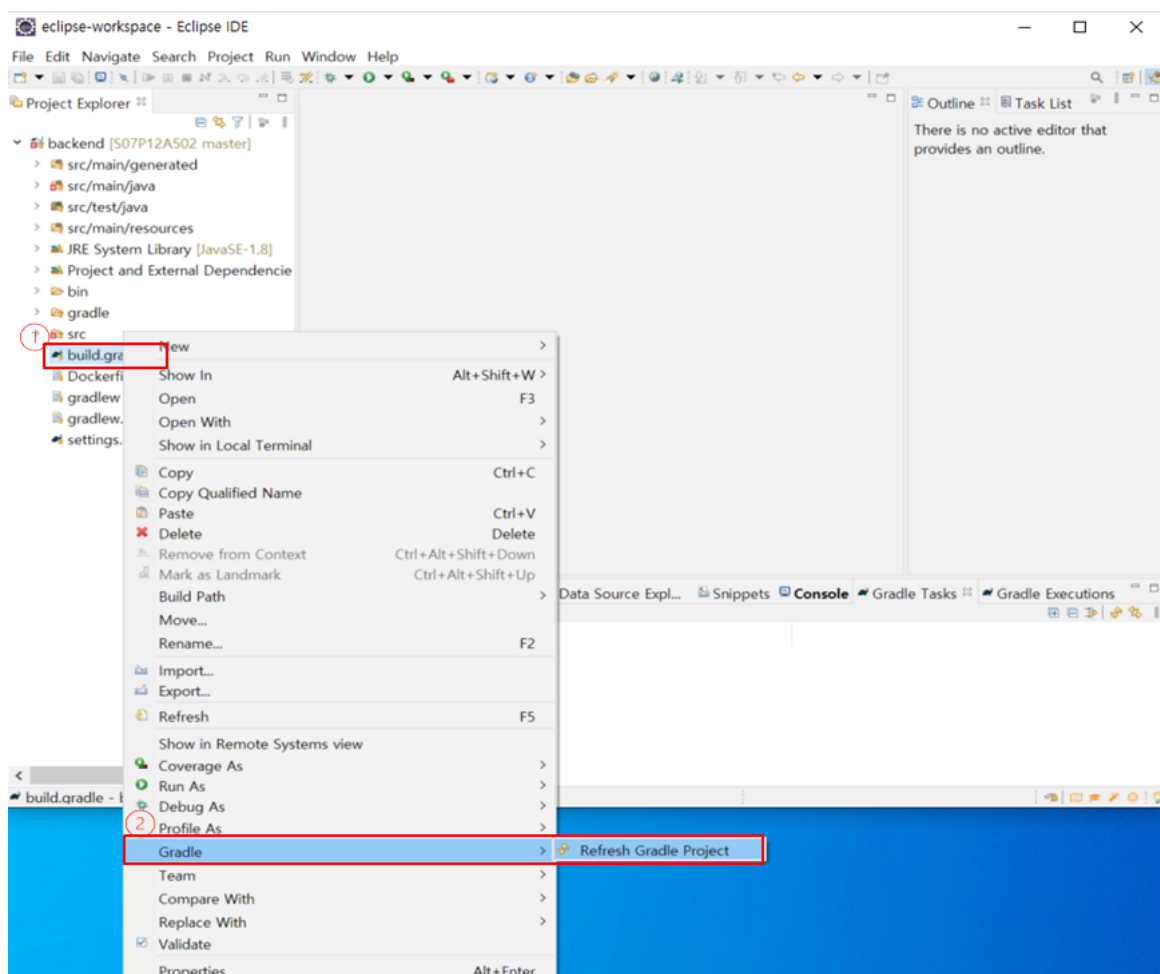
1. 작업관리 : Jira
2. 형상관리 : Gitlab
3. 메신저 : Mattermost
4. 개발환경
 - a. OS : Window 10
 - b. IDE
 - i. Visual Studio Code 1.69.2
 - ii. Eclipse 2020-06
 - iii. MySQL Workbench 8.0 CE
 - c. 데이터베이스 : Mysql 5.7
 - d. 프론트엔드 :
 - i. Vue : 3.2.13
 - ii. Node.js : 14.17.6
 - iii. Vuex : 4.0.0
 - iv. Quasar : 2.0.0
 - e. 백엔드 :
 - i. Java : OpenJDK 1.8.0_192
 - ii. Gradle : 7.5

- iii. Spring : 5.3.6
- iv. SpringBoot : 2.4.5
- v. querydsl : 4.4.0
- vi. lombok : 1.18.24
- f. 서버 : AWS EC2
 - i. Tool : MobaXterm
 - ii. OS : ubuntu 20.04
 - iii. Nginx : 1.18.0
 - iv. Docker : 20.10.17
 - v. Openvidu : 2.22.0
 - vi. Kurento-media-server : 6.16.0

빌드 상세

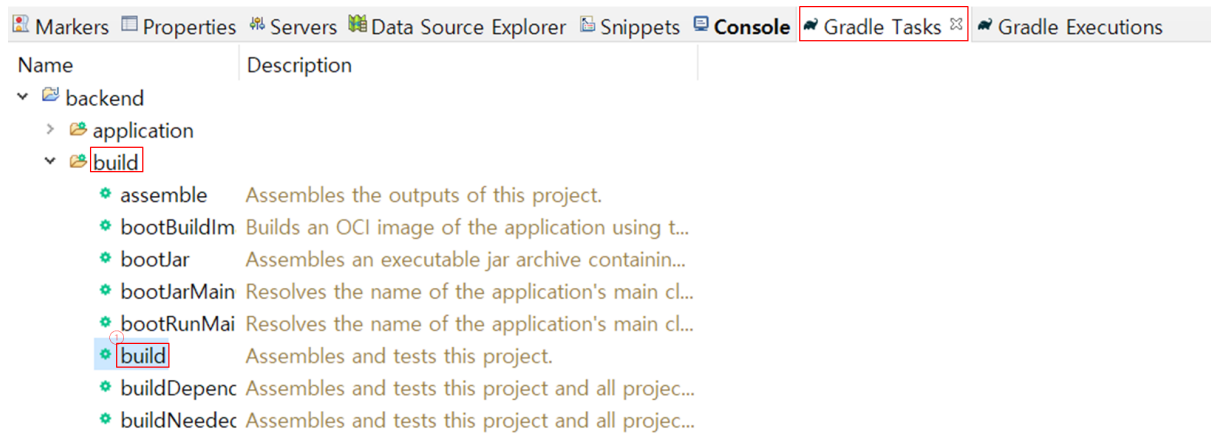
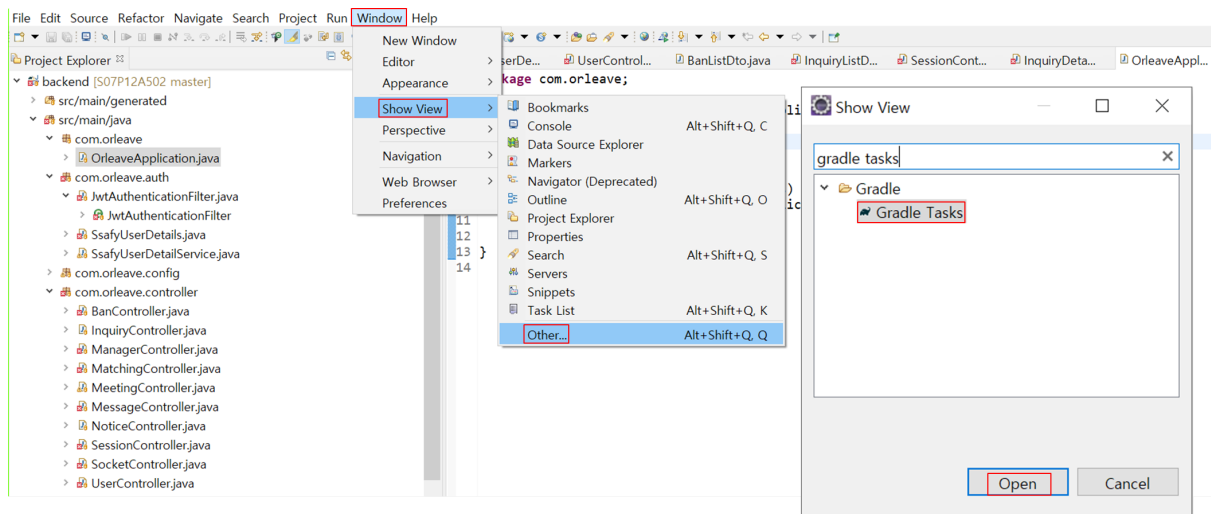
백엔드

GUI

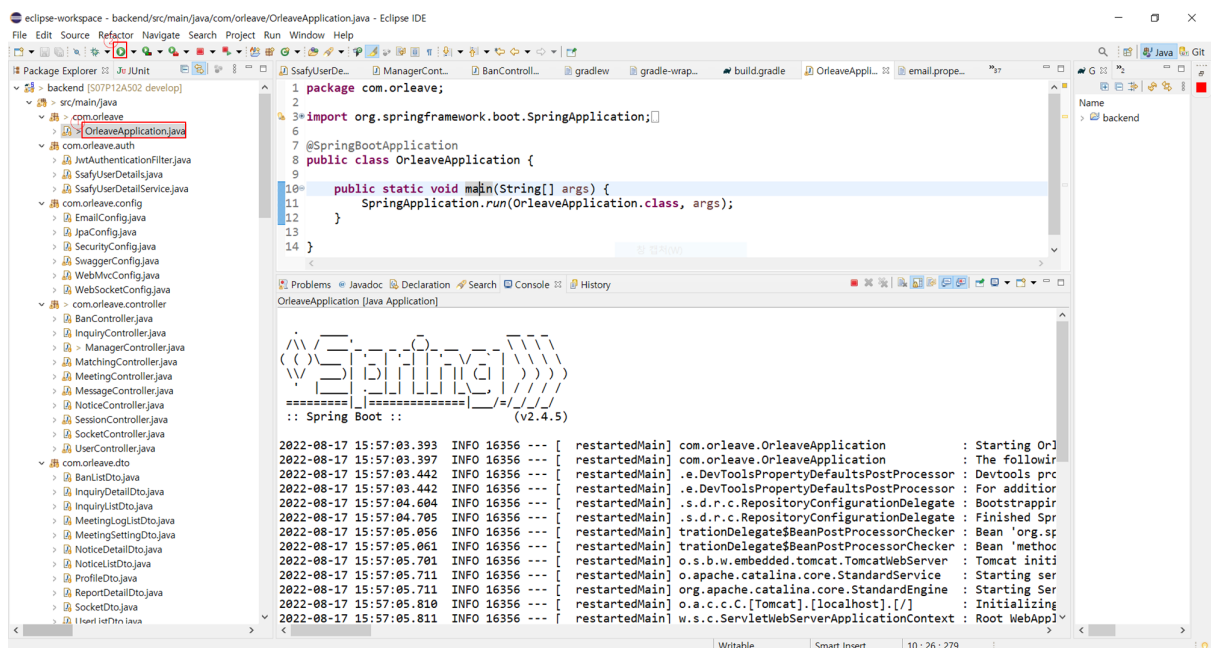


1. build.gradle 우클릭

2. Gradle 에서 Refresh Gradle Project 클릭



1. build 더블클릭



1. OrleaveApplication.java 클릭

2. 실행버튼 클릭

Command

```
#현재위치에서 gradlew 검색
find . -name gradlew
...
./jenkins/workspace/orleave-test/backend/gradlew
```

```
#해당 디렉토리로 이동
cd ./jenkins/workspace/orleave-test/backend
```

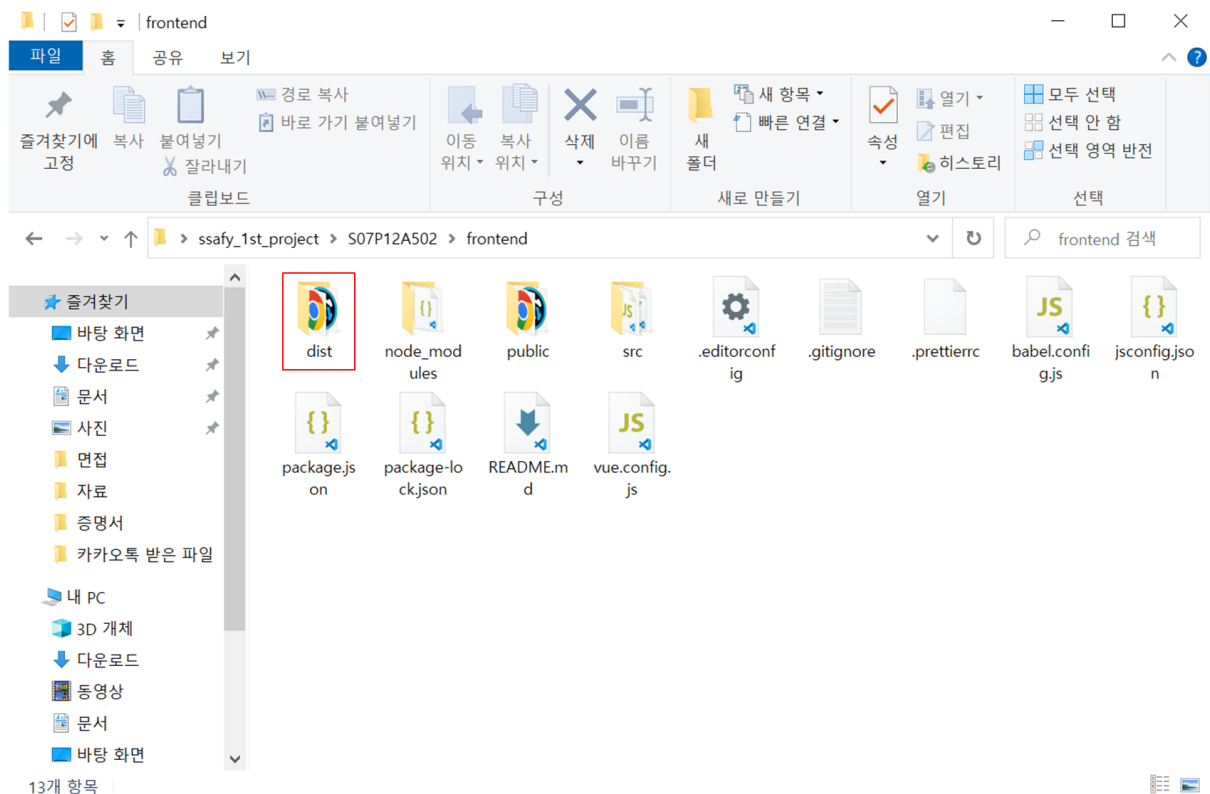
```
#빌드
./gradlew clean build
```

```
#생성된 jar 파일 확인
cd ./build/libs
ls
backend-1.0-SNAPSHOT.jar
```

docker ps	CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
	088ca0c73fb3	orleave_backend:0.1	"java -jar /app.jar"	4 hours ago	Up 4 hours	0.0.0.0:8181->8181/

프론트엔드

```
#dist 디렉토리 생성시 성공
npm i
npm run build
```



EC2 설정

도커 설치

```
#설치 고려사항 확인 64bit, 리눅스 커널정보(3.10 이상)
uname -a

#설치 가능한 리스트 업데이트
sudo apt-get update

sudo apt-get install -y \
apt-transport-https \
curl \
ca-certificates \
software-properties-common

#docker의 공식 GPG(GNU Privacy Guard) key를 추가
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

#추가된 키 + 도커에서 배포한 키확인
sudo apt-key fingerprint
sudo apt-key fingerprint 0EBFCD88

#debian 계열의 docker repository 추가 후 apt update 수행
sudo add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

sudo apt-get update
#docker CE 버전 설치
sudo apt-get install docker-ce

#sudo 없이 도커 사용
sudo usermod -aG docker zeff
sudo systemctl enable docker
sudo systemctl restart docker
sudo reboot

#도커 컴포즈 설치
sudo curl -L \
"https://github.com/docker/compose/releases/download/1.27.4/dockercompose -$(uname -s)-$(uname -m)"\
-o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

Mysql 설치 + 설정

```
#mysql 실행
docker run --name mysql -p 3306:3306 -e MYSQL_ROOT_PASSWORD=cityview1202502 \
-d mysql:5.7 --character-set-server=utf8 --collation-server=utf8_unicode_ci

#mysql 컨테이너 접속
docker exec -it mysql /bin/sh
sh-4.2# mysql -u root -p
Enter password: cityview1202502

#루트 계정 이름 변경
mysql> use mysql
mysql> update user set user='ssafy502' where user='root';
mysql> flush privileges;
mysql> exit
```

Nginx + SSL 설정

```
#nginx 설치
sudo apt-get update
sudo apt-get install nginx
sudo systemctl start nginx
sudo systemctl status nginx

#active면 동작중
• nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-08-05 23:07:26 KST; 1 weeks 4 days ago
     ...
     ...

#SSL 설정
#https://certbot.eff.org/instructions?ws=apache&os=ubuntu focal 참고

#snap 패키지 관리 툴 설치
apt-get install snapd

#snap 최신화
```

```

sudo snap install core
sudo snap refresh core

#certbot 설치
sudo snap install --classic certbot

#ln 명령어로 cli에서 certbot 커맨드 사용가능하게 설정
sudo ln -s /snap/bin/certbot /usr/bin/certbot

#ssl 인증서 발급
sudo certbot certonly --nginx -d i7a502.p.ssafy.io

#프록시 설정
cd /etc/nginx/sites-available
sudo vi orleave

server {
    location /{ # /로 시작하는 url은 http://localhost:3000으로 중계(프론트엔드)
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
    }

    location /api { # /api로 시작하는 url은 http://localhost:8181/api으로 중계(백엔드)
        proxy_pass http://localhost:8181/api;
    }

    listen 443 ssl;
    ssl_certificate /etc/letsencrypt/live/i7a502.p.ssafy.io/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/i7a502.p.ssafy.io/privkey.pem;
    # include /etc/letsencrypt/options-ssl-nginx.conf;
    # ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}

server {
    if ($host = i7a502.p.ssafy.io) {
        return 301 https://$host$request_uri;
    }

    listen 80;
    server_name i7a502.p.ssafy.io;
    return 404;
}

```

Openvidu 설치

```

#권장 설치 경로
cd /opt
#오픈비두 설치
sudo curl https://s3-eu-west-1.amazonaws.com/aws.openvidu.io/install_openvidu_latest.sh | sudo bash
cd openvidu
sudo vi .env

...
# For example: 198.51.100.1, or openvidu.example.com
DOMAIN_OR_PUBLIC_IP=(도메인 또는 아이피 주소값)

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

# Certificate type:
# - selfsigned: Self signed certificate. Not recommended for production use.
#               Users will see an ERROR when connected to web page.
# - owncert:    Valid certificate purchased in a Internet services company.
#               Please put the certificates files inside folder ./owncert
#               with names certificate.key and certificate.cert
# - letsencrypt: Generate a new certificate using letsencrypt. Please set the
#                 required contact email for Let's Encrypt in LETSENCRYPT_EMAIL
#                 variable.
CERTIFICATE_TYPE=letsencrypt

# If CERTIFICATE_TYPE=letsencrypt, you need to configure a valid email for notifications
LETSENCRYPT_EMAIL=(원하는 이메일)

# 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=8442

# Changes the port of all services exposed by OpenVidu.
# SDKs, REST clients and browsers will have to connect to this port
HTTPS_PORT=8443
...
...

# openvidu 시작
./openvidu start

...
-----

OpenVidu Platform is ready!
-----

* OpenVidu Server: https://DOMAIN_OR_PUBLIC_IP/

* OpenVidu Dashboard: https://DOMAIN_OR_PUBLIC_IP/dashboard/

-----

#실행 확인
docker ps
CONTAINER ID        IMAGE                                     COMMAND                  CREATED            STATUS              PORTS
...
536b9d2aa576       openvidu/openvidu-call:2.22.0          "docker-entrypoint.s..." 12 days ago       Up 12 days
9c545ec31f46       openvidu/openvidu-proxy:2.22.0         "/docker-entrypoint..." 12 days ago       Up 12 days
e8b8f6b32376       kurento/kurento-media-server:6.16.0   "/entrypoint.sh"        12 days ago       Up 12 days (healthy)
2d628a6220df       openvidu/openvidu-coturn:2.22.0        "docker-entrypoint.s..." 12 days ago       Up 12 days         0.0.0.0:3478->3478/
9c39965ecd39       openvidu/openvidu-server:2.22.0        "/usr/local/bin/entr..." 12 days ago       Up 12 days
```

외부서비스

소셜 로그인

- 서비스의 로그인을 카카오, 네이버, 구글 로그인을 통해서도 가능합니다.
- 소셜 로그인에서 사용자의 이메일을 확인하고, 해당 이메일로 가입한 사용자가 있는지 확인합니다. 만약 사용자가 있으면 해당 계정으로 로그인하고, 그렇지 않으면 프로필 정보 입력 페이지로 넘어가 회원가입을 새로 진행합니다.

카카오 로그인

1. 애플리케이션 추가도메인 등록

애플리케이션 추가하기

앱 아이콘



파일 선택

JPG, GIF, PNG

권장 사이즈 128px, 최대 250KB

앱 이름

OrLeave

사업자명

OrLeave

- 입력된 정보는 사용자가 카카오 로그인을 할 때 표시됩니다.
- 정보가 정확하지 않은 경우 서비스 이용이 제한될 수 있습니다.

취소

저장

2. 도메인 등록

Web

삭제

수정

사이트 도메인

http://localhost:8080
https://i7a502.p.ssafy.io

- 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. [등록하러 가기](#)

3. redirect URI 설정

Redirect URI

삭제

수정

Redirect URI

http://localhost:8083/user/login-kakao
https://i7a502.p.ssafy.io/user/login-kakao

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

4. 로그인 활성화

/v2/user/me
Authorization: Bearer \${ACCESS_TOKEN}
Content-type: application/x-www-form-urlencoded;charset=utf-8

네이버 로그인

1. 애플리케이션 등록

애플리케이션 등록 (API 이용신청)

애플리케이션의 기본 정보를 등록하면, 좌측 **내 애플리케이션** 메뉴의 서브 메뉴에 등록하신 애플리케이션 이름으로 서브 메뉴가 만들어집니다.

애플리케이션 이름	<div>OrLeave</div> <div>• 네이버 로그인할 때 사용자에게 표시되는 이름이므로 서비스 브랜드를 대표할 수 있는 이름으로 가급적 10자 이내로 간결하게 설정해주세요. • 40자 이내의 영문, 한글, 숫자, 공백문자, 원표(,), "/, "-", "_", 만 입력 가능합니다.</div>																														
사용 API	<div>선택하세요.</div> <div>네이버 로그인</div> <div>제공 정보 선택(이용자 식별자는 기본 정보로 제공) ^① 필수 항목은 개인정보보호법 제3조 제1항, 제16조 제1항 등에 따라 서비스 제공을 위해 필요한 최소한의 개인정보를 선택해야 합니다.</div> <table><thead><tr><th>권한</th><th>필수</th><th>추가</th></tr></thead><tbody><tr><td>회원이름</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>이메일 주소</td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>별명</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>프로필 사진</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>성별</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>생일</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>연령대</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>출생연도</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>휴대전화번호</td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr></tbody></table> <div>[알림] 추가권한에 대한 네이버 로그인 공자사항을 확인하세요.</div>	권한	필수	추가	회원이름	<input type="checkbox"/>	<input type="checkbox"/>	이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>	별명	<input type="checkbox"/>	<input type="checkbox"/>	프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>	성별	<input type="checkbox"/>	<input type="checkbox"/>	생일	<input type="checkbox"/>	<input type="checkbox"/>	연령대	<input type="checkbox"/>	<input type="checkbox"/>	출생연도	<input type="checkbox"/>	<input type="checkbox"/>	휴대전화번호	<input type="checkbox"/>	<input type="checkbox"/>
권한	필수	추가																													
회원이름	<input type="checkbox"/>	<input type="checkbox"/>																													
이메일 주소	<input checked="" type="checkbox"/>	<input type="checkbox"/>																													
별명	<input type="checkbox"/>	<input type="checkbox"/>																													
프로필 사진	<input type="checkbox"/>	<input type="checkbox"/>																													
성별	<input type="checkbox"/>	<input type="checkbox"/>																													
생일	<input type="checkbox"/>	<input type="checkbox"/>																													
연령대	<input type="checkbox"/>	<input type="checkbox"/>																													
출생연도	<input type="checkbox"/>	<input type="checkbox"/>																													
휴대전화번호	<input type="checkbox"/>	<input type="checkbox"/>																													
로그인 오픈 API 서비스 환경	<div>환경 추가</div> <div><div>PC 웹</div><div>서비스 URL</div><div>http://localhost:8080</div><div>서비스 URL 예시: (O) http://naver.com (X) http://www.naver.com 서비스 URL 값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다. 불법/음란성 사이트 등 이용약관에 위배되는 사이트의 경우, 이용이 제한될 수 있습니다. 서비스하려는 사이트 URL과 동일한 사이트 URL로 해주셔야 네이버 로그인이 노출됩니다.</div><div>네이버 로그인 Callback URL (최대 5개)</div><div>http://localhost:8083/user/login-naver</div><div>테스트 폼 우측 끝의 '+' 버튼을 누르면 행이 추가되며, '-' 버튼을 누르면 행이 삭제됩니다. Callback URL 은 네이버 로그인 후 이동할 페이지 URL 입니다. Callback URL 값이 잘못 입력되어 있으면 정확한 값으로 수정하실 때 까지 네이버 로그인 사용이 일시적으로 제한됩니다. 입력한 주소와 다른 Callback URL로 리다이렉트 될 경우, 이용이 제한될 수 있습니다.</div></div>																														

2. 클라이언트 ID, Secret 발급

애플리케이션 정보

Client ID	<div>XXXXXXXXXX</div>
Client Secret	<div>.....</div> <div>보기</div>

3. 네이버 로그인 연동 URL 생성

요청 URL 정보

메서드	요청 URL	출력 포맷	설명
GET / POST	https://nid.naver.com/oauth2.0/authorize	URL 리다이렉트	네이버 로그인 인증 요청

요청 변수 정보

요청 변수명	타입	필수 여부	기본값	설명
response_type	string	Y	code	인증 과정에 대한 내부 구분값으로 'code'로 전송해야 함
client_id	string	Y	-	애플리케이션 등록 시 발급받은 Client ID 값
redirect_uri	string	Y	-	애플리케이션을 등록 시 입력한 Callback URL 값으로 URL 인코딩을 적용한 값
state	string	Y	-	사이트 간 요청 위조(cross-site request forgery) 공격을 방지하기 위해 애플리케이션에서 생성한 상태 토큰값으로 URL 인코딩을 적용한 값을 사용

4. 접근 토큰 발급 요청

요청 URL 정보

메서드	요청 URL	출력 포맷	설명
GET / POST	https://nid.naver.com/oauth2.0/token	json	접근토큰 발급 요청

요청 변수 정보

요청 변수명	타입	필수 여부	기본값	설명
grant_type	string	Y	-	인증 과정에 대한 구분값 1) 발급: 'authorization_code' 2) 갱신: 'refresh_token' 3) 삭제: 'delete'
client_id	string	Y	-	애플리케이션 등록 시 발급받은 Client ID 값
client_secret	string	Y	-	애플리케이션 등록 시 발급받은 Client secret 값
code	string	발급 때 필수	-	로그인 인증 요청 API 호출에 성공하고 리턴받은 인증코드값 (authorization code)
state	string	발급 때 필수	-	사이트 간 요청 위조 (cross-site request forgery) 공격을 방지하기 위해 애플리케이션에서 생성한 상태 토큰값으로 URL 인코딩을 적용한 값을 사용
refresh_token	string	갱신 때 필수	-	네이버 사용자 인증에 성공하고 발급받은 갱신 토큰 (refresh token)
access_token	string	삭제 때 필수	-	기 발급받은 접근 토큰으로 URL 인코딩을 적용한 값을 사용
service_provider	string	삭제 때 필수	'NAVER'	인증 제공자 이름으로 'NAVER'로 세팅해 전송

5. 접근 토큰을 이용해 프로필 API 호출

3.4.5 접근 토큰을 이용하여 프로필 API 호출하기

접근 토큰을 이용하면 프로필 정보 조회 API를 호출하거나 오픈 API를 호출하는 것이 가능합니다.

사용자 로그인 정보를 획득하기 위해서는 프로필 정보 조회 API를 먼저 호출하여야 합니다.

요청 URL 정보

메서드	인증	요청 URL	출력 포맷	설명
GET / POST	OAuth2.0	https://openapi.naver.com/v1/nid/me	JSON	프로필 정보 조회

요청 변수 정보

요청 변수는 별도로 없으며, 요청 URL로 호출할 때 아래와 같이 요청 헤더에 접근 토큰 값을 전달하면 됩니다.


요청 헤더

요청 헤더명	설명
Authorization	접근 토큰 (access token)을 전달하는 헤더 다음과 같은 형식으로 헤더 값에 접근 토큰 (access token)을 포함합니다. 토큰 타입은 "Bearer"로 값이 고정되어 있습니다. Authorization: {토큰 타입} {접근 토큰}


구글 로그인

1. cloud.google.com/cloud-console/ 접속 후 콘솔로 이동
2. 새 프로젝트 만들기


새 프로젝트

 projects 할당량이 22개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)

프로젝트 이름 *
OrLeave 

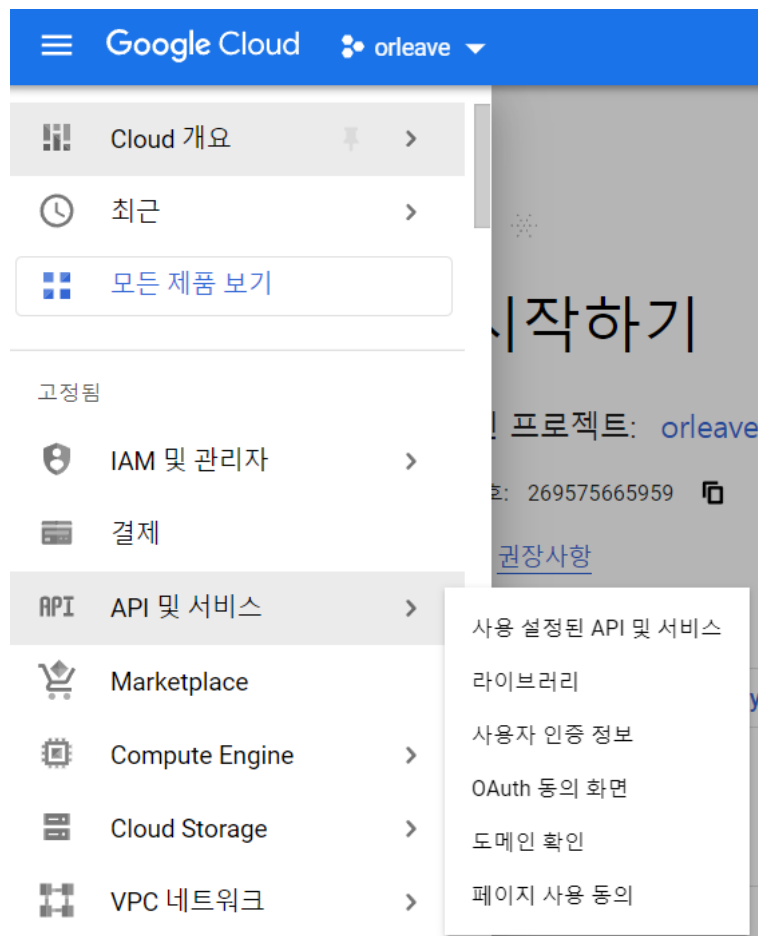
프로젝트 ID: orleave-359706입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 *
 조직 없음 [찾아보기](#)

상위 조직 또는 폴더

[만들기](#) [취소](#)

3. 왼쪽 탭에서 API 및 서비스 - OAuth 동의 화면 - 동의 화면 구성으로 이동



4. UserType 외부로 설정

OAuth 동의 화면

대상 사용자를 비롯해 앱을 구성하고 등록하려는 방식을 선택하세요. 프로젝트에는 하나의 앱만 연결할 수 있습니다.

User Type

☐ 내부 ?

조직 내 사용자만 사용할 수 있습니다. 인증을 위해 앱을 제출할 필요는 없습니다. [사용자 유형 자세히 알아보기](#)

☒ 외부 ?

Google 계정이 있는 모든 테스트 사용자가 사용할 수 있습니다. 앱이 테스트 모드로 시작되고 테스트 사용자 목록에 추가된 사용자에게만 제공됩니다. 앱을 프로덕션에 푸시할 준비가 되면 앱을 인증해야 할 수도 있습니다. [사용자 유형 자세히 알아보기](#)

만들기

5. OAuth 동의 화면 정보 입력

앱 정보

동의 화면에 표시되어 최종 사용자가 개발자를 확인하고 문의할 수 있습니다.

앱 이름 *
OrLeave

동의를 요청하는 앱의 이름

사용자 프로필 이미지 *
[이미지]

사용자가 동의 관련 질문을 위해 문의할 때 이용합니다.

앱 로고 *
[이미지] X 찾아보기

사용자가 앱을 알아보는 데 도움이 되도록 동의 화면에 대한 이미지(1MB 이하 크기)를 업로드합니다. 허용되는 이미지 형식은 JPG, PNG, BMP입니다. 최적의 결과를 위해서는 로고가 120x120픽셀 크기의 정사각형이어야 합니다.



앱 도메인

나와 내 사용자를 보호하기 위해 Google에서는 OAuth를 사용하는 앱만 승인된 도메인을 이용할 수 있도록 허용합니다. 다음 정보가 동의 화면에서 사용자에게 표시됩니다.

애플리케이션 홈페이지
사용자에게 홈페이지 링크를 제공합니다.

애플리케이션 개인정보처리방침 링크
사용자에게 공개 개인정보처리방침 링크를 제공합니다.

애플리케이션 서비스 약관 링크
사용자에게 공개 서비스 약관 링크를 제공합니다.

승인된 도메인 ?

동의 화면 또는 OAuth 클라이언트 구성에서 도메인이 사용되면 여기에서 사전 등록해야 합니다. 앱이 인증을 가져야 하는 경우 [Google Search Console](#)로 이동하여 도메인이 승인되었는지 확인하세요. 승인된 도메인 한도에 대해 [자세히 알아보기](#).

+ 도메인 추가

개발자 연락처 정보

이메일 주소 *
[이메일 주소]

이 이메일 주소는 Google에서 프로젝트 변경사항에 대해 알람을 보내기 위한 용도입니다.

저장 후 계속 취소

6. 테스트 사용자 입력

테스트 사용자

게시 상태가 '테스트 중'으로 설정된 동안에는 테스트 사용자만 앱에 액세스할 수 있습니다. 앱 인증 전에 허용되는 사용자 한도는 100명이며 앱의 전체 수명 주기에서 계산됩니다. [자세히 알아보기](#)

+ ADD USERS

필터 속성 이름 또는 값 입력
?

사용자 정보
표시할 행이 없습니다.

저장 후 계속
취소

7. 사용자 인증 정보 - CREATE CREDENTIALS - API 키 클릭

API API 및 서비스	사용자 인증 정보	+ CREATE CREDENTIALS DELETE
<div style="display: flex; align-items: center; margin-bottom: 5px;"> ⚙️ 사용 설정된 API 및 서비스 </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> 📖 라이브러리 </div> <div style="display: flex; align-items: center; margin-bottom: 5px; background-color: #e0f0ff;"> 🔑 사용자 인증 정보 </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> 🔒 OAuth 동의 화면 </div> <div style="display: flex; align-items: center; margin-bottom: 5px;"> 📄 도메인 확인 </div> <div style="display: flex; align-items: center;"> 👤 페이지 사용 동의 </div>	<div style="margin-bottom: 10px;"> <p>사용 설정한 API에 액세스하려면</p> <p>API 키</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <div style="display: flex; align-items: center;"> <input type="checkbox"/> 이름 </div> <p style="font-size: 0.8em;">표시할 API 키가 없습니다.</p> </div> <p>OAuth 2.0 클라이언트 ID</p> </div>	<div style="margin-bottom: 10px;"> <p>API 키</p> <p>할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.</p> <p>OAuth 클라이언트 ID</p> <p>앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.</p> <p>서비스 계정</p> <p>로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.</p> </div> <div> <p>사용자 인증 정보 선택 도움말</p> <p>사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.</p> </div>

8. API 키 획득

API 키 생성됨

애플리케이션에서 이 키를 사용하려면 키를 `key=API_KEY` 매개변수로 전달하세요.

API 키

⚠ 이 키는 제한되지 않습니다. 무단 사용을 방지하려면 API를 사용할 수 있는 위치와 대상을 제한하는 것이 좋습니다. 제한사항을 추가하려면 [API 키를 수정](#)하세요. [자세히 알아보기](#)

닫기

9. 사용자 인증 정보 - CREATE CREDENTIALS - OAuth 클라이언트 ID 클릭

API	API 및 서비스	사용자 인증 정보	+ CREATE CREDENTIALS	DELETE
	사용 설정된 API 및 서비스	사용 설정한 API에 액세스하려면	<div>API 키 할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.</div> <div>OAuth 클라이언트 ID 앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.</div> <div>서비스 계정 로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.</div> <div>사용자 인증 정보 선택 도움말 사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.</div>	
	라이브러리	API 키		
	사용자 인증 정보	<input type="checkbox"/> 이름 표시할 API 키가 없습니다.		
	OAuth 동의 화면	OAuth 2.0 클라이언트 ID		
	도메인 확인			
	페이지 사용 동의			

10. OAuth 클라이언트 ID 만들기

← OAuth 클라이언트 ID 만들기

클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요. OAuth 클라이언트 유형을 [자세히 알아보세요](#).

애플리케이션 유형 *
웹 애플리케이션

이름 *
OrLeave

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

❗ 아래에 추가한 URI의 도메인이 [승인된 도메인](#)으로 OAuth 동의 화면에 자동으로 추가됩니다.

승인된 자바스크립트 원본 ?

브라우저 요청에 사용

URI 1 *
http://localhost:8080

+ URI 추가

승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI 1 *
http://localhost:8083/user/login-google

+ URI 추가

참고: 설정이 적용되는 데 5분에서 몇 시간이 걸릴 수 있습니다.

만들기 취소

11. 클라이언트 ID, 보안 비밀 획득

클라이언트 ID

클라이언트 보안 비밀

생성일

12. Google OAuth 2.0 서버로 리디렉션

Google의 OAuth 2.0 엔드포인트는 <https://accounts.google.com/o/oauth2/v2/auth>에 있습니다. 이 엔드포인트는 HTTPS를 통해서만 액세스할 수 있습니다. 일반 HTTP 연결은 거부됩니다.

Google 승인 서버는 웹 서버 애플리케이션에 다음과 같은 쿼리 문자열 매개변수를 지원합니다.

매개변수	
client_id	필수 애플리케이션의 클라이언트 ID입니다. 이 값은 API Console Credentials page 에서 확인할 수 있습니다.
redirect_uri	필수 사용자가 승인 흐름을 완료한 후 API 서버가 사용자를 리디렉션하는 위치를 결정합니다. 이 값은 클라이언트의 API Console Credentials page 에서 구성한 OAuth 2.0 클라이언트에 대해 승인된 리디렉션 URI 중 하나와 정확히 일치해야 합니다. 이 값이 제공된 client_id에 대해 승인된 리디렉션 URI와 일치하지 않으면 redirect_uri_mismatch 오류가 발생합니다. http 또는 https 스킴, 대소문자, 후행 슬래시('/')는 모두 일치해야 합니다.
response_type	필수 Google OAuth 2.0 엔드포인트에서 승인 코드를 반환할지 여부를 결정합니다. 웹 서버 애플리케이션의 매개변수 값을 code로 설정합니다.
scope	필수 애플리케이션이 사용자를 대신하여 액세스할 수 있는 리소스를 식별하는 공백으로 구분된 범위 목록입니다. 이러한 값은 Google이 사용자에게 표시하는 동의 화면에 알립니다. 범위를 사용하면 애플리케이션이 필요한 리소스에 대한 액세스만 요청하는 동시에 사용자가 애플리케이션에 부여하는 액세스 양을 제어할 수 있습니다. 따라서 요청된 범위 수와 사용자 동의를 얻을 가능성 사이에는 반비례적인 관계가 있습니다. 가능한 경우 애플리케이션에서 상황에 따라 승인 범위에 대한 액세스를 요청하는 것이 좋습니다. 컨텍스트에서 사용자 데이터에 대한 액세스를 요청하면 중분 승인 을 통해 사용자가 애플리케이션에 필요한 액세스 권한이 필요한 이유를 더 쉽게 이해할 수 있습니다.

13. 갱신 및 액세스 토큰의 승인 코드 교환

엑세스 토큰에 대한 승인 코드를 교환하려면 <https://oauth2.googleapis.com/token> 엔드포인트를 호출하고 다음 매개변수를 설정합니다.

필드	
client_id	API Console Credentials page 에서 가져온 클라이언트 ID입니다.
client_secret	Credentials page 에서 가져온 API Console 클라이언트 보안 비밀번호
code	초기 요청에서 반환된 승인 코드입니다.
grant_type	OAuth 2.0 사양에 정의된 대로 이 필드의 값은 authorization_code로 설정해야 합니다.
redirect_uri	지정된 client_id의 API Console Credentials page 에 있는 프로젝트에 나열된 리디렉션 URI 중 하나입니다.

응답에는 다음 필드가 포함됩니다.

필드	
access_token	Google API 요청을 승인하기 위해 애플리케이션이 전송하는 토큰입니다.
expires_in	엑세스 토큰의 잔여 기간(초)입니다.
refresh_token	새 엑세스 토큰을 얻는 데 사용할 수 있는 토큰입니다. 갱신 토큰은 사용자가 액세스를 취소할 때까지 유효합니다. 다시 말하지만, 이 필드는 초기 요청에서 Google의 승인 서버로 access_type 매개변수를 offline로 설정한 경우에만 이 응답에 표시됩니다.
scope	access_token에서 부여하는 액세스 범위이며 공백으로 구분되고 대소문자를 구분하는 문자열 목록으로 표시됩니다.
token_type	반환된 토큰 유형입니다. 현재 이 필드의 값은 항상 Bearer로 설정됩니다.

SMTP 이메일 인증 설정

NAVER 메일

999+ ★ 100% 100% 100%

전체메일 정리하기

받은메일함 5091 수신확인

보낸메일함

임시보관함

내게쓴메일함 1 빠른정리

내 메일함

구글(지메일&구글앱스)

스마트메일함

프로모션 50

청구결제 35

SNS 9

카레 17

스팸메일함 3

휴지통 1900

외부메일 가져오기

메일 스팸투분 +25GB

메일 도움말

환경 설정 | 메일로 돌아가기

기본 환경 설정 | 메일함 관리 | ② 메일 자동 분류 | 서명/빠른답장 | 부재 중 설정 | 새 메일 알림 설정

스팸 설정 | 외부 메일 가져오기 | ✓ POP3/IMAP 설정 | 단축키

POP3/SMTP 설정 | IMAP/SMTP 설정

휴대폰, 아웃룩 등에서 네이버 메일을 확인할 수 있도록 POP3/SMTP를 설정합니다.

님께서는 현재 POP3/SMTP를 사용하지 않습니다.
POP3/SMTP를 사용하려면 [POP3/SMTP 사용]을 [사용함]으로 설정하세요.

POP3/SMTP 사용 ☒ 사용할 ☐ 사용 안 함

적용 범위 ☐ 지금부터 새로 받는 메일만 받음 ☒ 기존에 받은 메일을 포함하여 받음

읽음 표시 ☒ POP3로 읽어간 메일을 읽음 표시 ☐ POP3로 읽어간 메일을 읽지 않음으로 표시

원본 저장 ☒ 네이버 메일에 원본 저장? ☐ 메일 프로그램 설정에 따라 저장 또는 삭제?

외부메일 처리 ☒ POP3로 읽어갈 때 외부메일을 포함하지 않음 ☐ POP3로 읽어갈 때 외부메일을 포함

기본 설정으로 ③ 확인 취소

· 메일 프로그램 환경 설정 안내
휴대폰, 아웃룩 등 외부 메일 프로그램 환경설정에서 아래와 같이 등록해 주세요.

POP 서버명 : pop.naver.com SMTP 서버명 : smtp.naver.com POP 포트 : 995, 보안연결(SSL) 필요
SMTP 포트 : 465, 보안 연결(SSL) 필요 아이디 : 비밀번호 : 네이버 로그인 비밀번호

TIP POP3/SMTP 연결이 되지 않아요 메일 프로그램 POP3/SMTP 설정하기

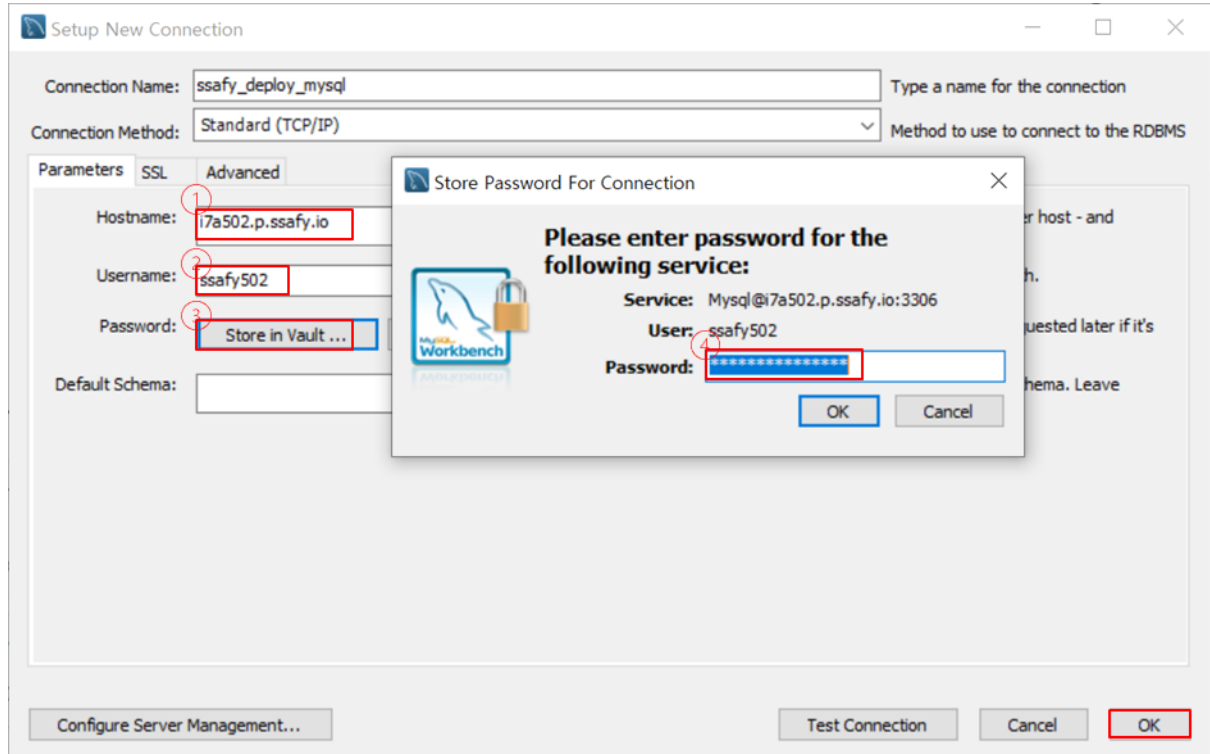
용량 759MB / 5GB | 환경설정 | 스팸설정 | POP3 로그인 기록보기 | 모바일 메일 t © NAVER Corp. All Rights Reserved. 스팸정책 | 공지사항 | 메일 고객센터

1. 환경설정 클릭
2. POP3/IMAP 설정 클릭

3. 위와 같이 설정하고 확인

DB

Mysql 워크벤치 사용방법



1. 배포서버 도메인 입력
2. Mysql 설정에서 변경한 username 입력
3. Store in Vault 클릭으로 비밀번호 입력
4. 비밀번호는 mysql 도커 실행에서 설정한 MYSQL_ROOT_PASSWORD 값

배포

```
sudo apt-get install git
git clone (해당 레포지토리 url)
(기타의 ignore파일을 지정된 장소에 복사)
(빌드 상세에서 설명한대로 백엔드, 프론트엔드 각각 빌드)
(각각의 Dockerfile이 위치한 디렉토리에서 아래의 커맨드 실행)

#백엔드
docker build -t orleave_backend:0.1 .
docker stop orleave
docker run --rm --name orleave -d -p 8181:8181 orleave_backend:0.1

#프론트엔드
docker build -t nginx_vue:0.1 .
docker stop nginx_vue
docker run --rm --name nginx_vue -d -p 3000:80 nginx_vue:0.1

docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
31f97188c93f   orleave_backend:0.1                 "java -jar /app.jar"    13 minutes ago Up 13 minutes  0.0.0.0:8181->81
e32975e63cda   nginx_vue:0.1                       "/docker-entrypoint...." 14 minutes ago Up 14 minutes  0.0.0.0:3000->80

#openvidu 배포
cd /opt/openvidu
```

```
./openvidu start

...
-----

OpenVidu Platform is ready!
-----

* OpenVidu Server: https://DOMAIN_OR_PUBLIC_IP/

* OpenVidu Dashboard: https://DOMAIN_OR_PUBLIC_IP/dashboard/

-----

#실행 확인
docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
31f97188c93f   orleave_backend:0.1                "java -jar /app.jar"    13 minutes ago Up 13 minutes   0.0.0.0:8181->81
e32975e63cda   nginx-vue:0.1                      "/docker-entrypoint..." 14 minutes ago Up 14 minutes   0.0.0.0:3000->80
536b9d2aa576   openvidu/openvidu-call:2.22.0      "docker-entrypoint.s..." 12 days ago   Up 12 days
9c545ec31f46   openvidu/openvidu-proxy:2.22.0     "/docker-entrypoint..." 12 days ago   Up 12 days
e8b8f6b32376   kurento/kurento-media-server:6.16.0 "/entrypoint.sh"        12 days ago   Up 12 days (healthy)
2d628a6220df   openvidu/openvidu-coturn:2.22.0    "docker-entrypoint.s..." 12 days ago   Up 12 days      0.0.0.0:3478->3478/
9c39965ecd39   openvidu/openvidu-server:2.22.0    "/usr/local/bin/entr..." 12 days ago   Up 12 days
```

기타

ignore된 파일

*.properties

properties 파일들은 backend/src/main/resources 디렉토리에 저장

application

```
#it will be set build date by gradle. if this value is @build.date@, front-end is development mode
build.date=@build.date@
server.port=8181
# Charset of HTTP requests and responses. Added to the "Content-Type" header if not set explicitly.
server.servlet.encoding.charset=UTF-8
# Enable http encoding support.
server.servlet.encoding.enabled=true
# Force the encoding to the configured charset on HTTP requests and responses.
server.servlet.encoding.force=true

# for SPA
spring.resources.static-locations=classpath:/dist/
spa.default-file=/dist/index.html
spring.mvc.throw-exception-if-no-handler-found=true
spring.resources.add-mappings=false

# Swagger
springfox.documentation.enabled=false
springfox.documentation.swagger.use-model-v3=false

#database
spring.jpa.hibernate.naming.implicit-strategy=org.springframework.boot.orm.jpa.hibernate.SpringImplicitNamingStrategy
spring.jpa.hibernate.naming.physical-strategy=org.springframework.boot.orm.jpa.hibernate.SpringPhysicalNamingStrategy
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL57Dialect
spring.data.web.pageable.one-indexed-parameters=true

spring.devtools.livereload.enabled=true

#gzip compression
server.compression.enabled=true
server.compression.mime-types=application/json,application/xml,text/html,text/xml,text/plain,application/javascript,text/css

#for health check
management.servlet.context-path=/manage
management.health.db.enabled=true
management.health.default.enabled=true
management.health.diskspace.enabled=true

spring.profiles.include=db,jwt,openvidu
```

application-db

```
spring.datasource.url=jdbc:mysql://(해당 주소)/(스키마 명)?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTiMeBe
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.hikari.username=(설정한 아이디)
spring.datasource.hikari.password=(설정한 비밀번호)
```

application-jwt

```
jwt.secret=(발급한 jwt secret key)
# unit is ms. 15 * 24 * 60 * 60 * 1000 = 15days
jwt.expiration=1296000000
```

application-oauth

```
kakao.client.id=(카카오에서 발급한 client id)
kakao.redirect.uri=https://i7a502.p.ssafy.io/users/login-kakao

naver.client.id=(네이버에서 발급한 client id)
naver.client.secret=(네이버에서 발급한 secret)

google.client.id=(구글에서 발급한 client id)
google.client.secret=(구글에서 발급한 secret)
google.redirect.uri=https://i7a502.p.ssafy.io/user/login-google
```

application-openvidu

```
openvidu.url: https://(도메인 명):8443
openvidu.secret: (OPENVIDU_SECRET에 설정한 비밀번호)
```

email

```
mail.smtp.auth=true
mail.smtp.starttls.required=true
mail.smtp.starttls.enable=true
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false
mail.smtp.port=465
mail.smtp.socketFactory.port=465

AdminMail.id=(SMTP 이메일 인증 설정에서 사용한 email)
AdminMail.password=(해당 email 비밀번호)
```

index.js

```
#frontend/src/api 디렉토리에 저장
import axios from 'axios'
// import { API_BASE_URL } from '@config'

function apiInstance() {
  const instance = axios.create({
    baseURL: 'https://i7a502.p.ssafy.io/api/v1',
    headers: {
      'Content-type': 'application/json'
    }
  })
  return instance
}

export { apiInstance }
```

```
#frontend/src/config 디렉토리에 저장
const API_BASE_URL = 'https://i7a502.p.ssafy.io/api/v1'
const WEBSOCKET_URL = 'https://i7a502.p.ssafy.io/api/v1/ws'

const KAKAO_CLIENT_ID = (카카오에서 발급한 client id)
const KAKAO_ADMIN_ID = (카카오에서 발급한 admin id)
const KAKAO_REDIRECT_URI = 'https://i7a502.p.ssafy.io/user/login-kakao'

const NAVER_CLIENT_ID = (네이버에서 발급한 client id)
const NAVER_REDIRECT_URI = 'https://i7a502.p.ssafy.io/user/login-naver'
```

```
const GOOGLE_CLIENT_ID = (구글에서 발급한 client id)
const GOOGLE_REDIRECT_URI = 'https://17a502.p.ssafy.io/user/login-google'

export { API_BASE_URL, WEBSOCKET_URL, KAKAO_CLIENT_ID, KAKAO_ADMIN_ID, KAKAO_REDIRECT_URI, NAVER_CLIENT_ID, NAVER_REDIRECT_URI, GOOGLE
```

default.conf

```
#frontend/nginx 디렉토리에 저장
server {
    listen 80;

    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }
}
```