포팅 매뉴얼

콰당

카페인과 당을 줄여줄
인생 콰당 방지 서비스

SSAFY

대전 10기 공통 프로젝트

콰당

❖ 개발 환경

- **BACKEND**

  - ☐ spring boot : 3.1.8
    - ☐ Project Metadata
      - ☐ Group : com.ssafy
      - ☐ Artifact : cadang
      - ☐ Name : cadang
      - ☐ Package Name : com.ssafy.cadang
  - ☐ jdk : 17
  - ☐ mysql : 8.0.36
  - ☐ intellij : 2024.1
  - ☐ git ignore : application.yml (src\main\resource 경로 위치)
  - ☐ python 3.8.10

- **FRONTEND**

  - ☐ Node : 20.9.0
  - ☐ npm: 10.1.0
  - ☐ vue : 10.1.0
  - ☐ VS Code : 1.86.1
  - ☐ git ignore : .env.local (\ 경로 위치)

- **WAS**

  - ☐ AWS

- 외부 **API (**소셜 로그인**)**

  - ☐ 카카오 로그인 : https://developers.kakao.com/product/kakaoLogin

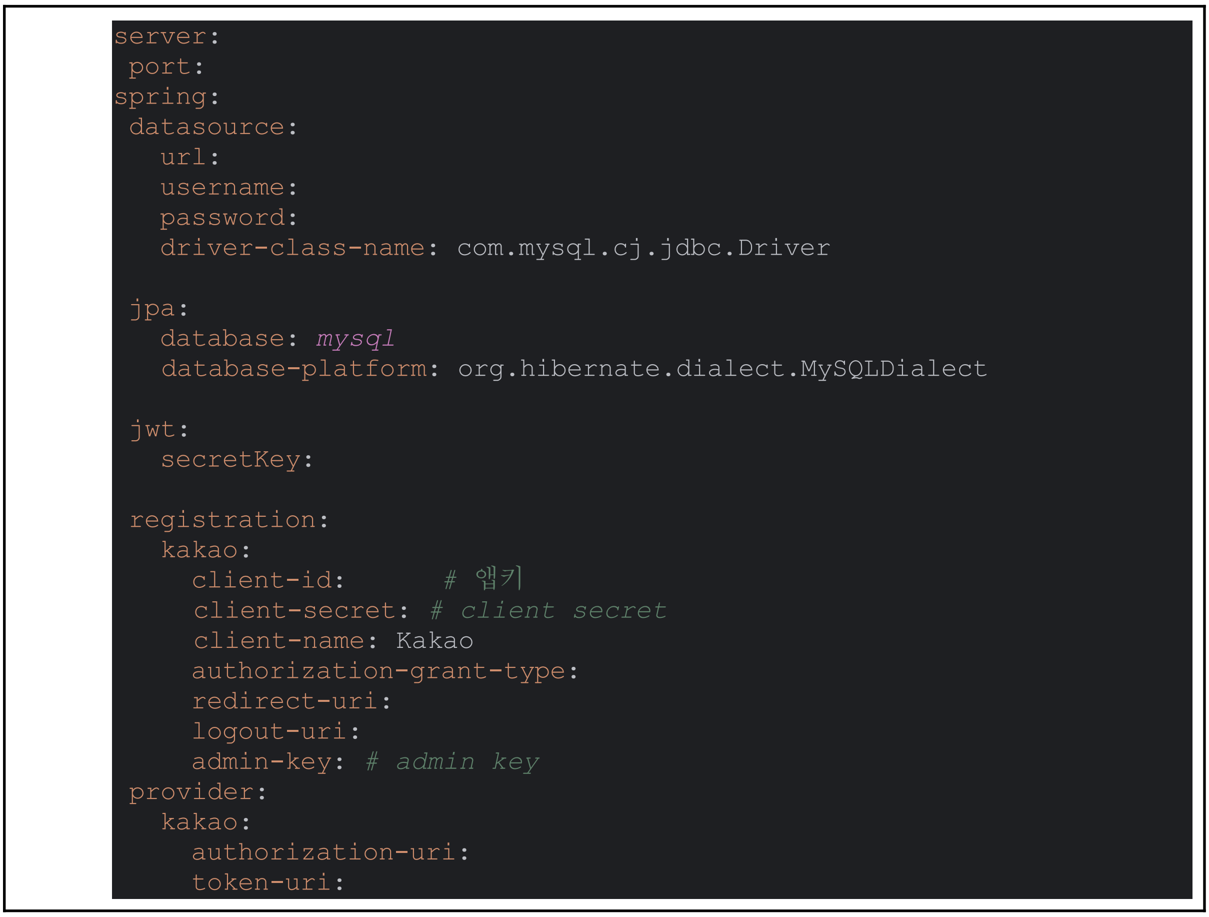
❖ BUILD

● BACK-END

=== notice appilcation.yml 파일추가 해야 빌드 ===
$ chmod +x gradlew
$ ./gradlew clean build
$ cd build/libs
$ java -jar cadang-0.0.1-SNAPSHOT.jar

Python install
apt-get install python3
pip install pandas
pip install matplotlib
pip install scikit-learn
python3 py파일명.py
백그라운드로 플라스크 서버 실행시키고 싶다면
nohup python3 app.py &

● application.yml



```
server:
 port:
spring:
 datasource:
   url:
   username:
   password:
   driver-class-name: com.mysql.cj.jdbc.Driver

 jpa:
   database: mysql
   database-platform: org.hibernate.dialect.MySQLDialect

 jwt:
   secretKey:

 registration:
   kakao:
     client-id:      # 앱키
     client-secret: # client secret
     client-name: Kakao
     authorization-grant-type:
     redirect-uri:
     logout-uri:
     admin-key: # admin key
 provider:
   kakao:
     authorization-uri:
     token-uri:
```

```
     user-info-uri:
     unlink-uri: https://kapi.kakao.com/v1/user/unlink
user-name-attribute:

notification:
 mattermost:
   enabled: true
   webhook-url:
```

실행 환경에 맞는 DB주소, 사용자 ID, PW와 카카오 로그인을 위한 앱키 입력 후 빌드

- **FRONT-END**

- 프론트엔드 프로젝트에 대하여 빌드폴더를 생성한다.
    1. 로컬에서 vscode에서 npm run build 명령어 입력
    2. 생성된 dist 빌드 폴더(혹은 빌드 폴더 내의 파일)를 서버 내로 옮긴다.

- 생성된 빌드 폴더를 nginx 환경설정에서 지정한 root 경로로 이동시킨다.
    1. root 경로는 index.html 이 존재하는 폴더 경로로 지정 dist 폴더 전체 옮길 경우 : /var/www/html/dist dist 폴더 내 파일만 옮길 경우 : /var/www/html
    2. 기존에 생성된 빌드 폴더가 존재할 경우, sudo rm -r /var/www/html/dist 명령어를 이용하여 삭제 후 진행한다. dist 폴더가 아닌 dist 폴더 내의 파일만 이동시켰을 경우, sudo rm /var/www/html/파일명 을 이용해 파일들을 삭제시킨다. ubuntu@ip-172-26-7-51:~$ sudo rm -r /var/www/html/dist
    3. 서버 내로 옮긴 빌드 폴더를 지정한 root 경로에 맞게 이동시킨다 . 복사 : sudo cp [복사할 폴더] [복사할 위치] 이동 : sudo mv [이동할 폴더] [이동할 위치] ubuntu@ip-172-26-7-51:~$ sudo mv /home/ubuntu/dist /var/www/html/dist

- nginx 재시작 ubuntu@ip-172-26-7-51:~$ sudo service nginx reload ubuntu@ip-172-26-7-51:~$ sudo service nginx restart

- **/etc/nginx/nginx.conf**

```
user nginx;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        tcp_nodelay on;
        keepalive_timeout 65;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

        ##
        # SSL Settings
        ##

        ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
        ssl_prefer_server_ciphers on;

        ##
        # Logging Settings
        ##
        access_log /var/log/nginx/access.log;
                       error_log /var/log/nginx/error.log;

        ##
        # Gzip Settings
        ##

gzip on;

        # gzip_vary on;
        # gzip_proxied any;
        # gzip_comp_level 6;
        # gzip_buffers 16 8k;
        # gzip_http_version 1.1;
        # gzip_types text/plain text/css application/json
application/javascript text/xml application/xml
application/xml+rss text/javascript;

        ##
        # Virtual Host Configs
        ##

        include /etc/nginx/conf.d/*.conf;
        include /etc/nginx/sites-enabled/*;

server {
        listen 80;
        server_name i10b106.p.ssafy.io;

        location / {
            proxy_pass http://localhost:5173; # 웹 애플리케이션 포트
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
        }
  # Backend Proxy 설정
        # 실제로 바라볼 port
        location /api {
                proxy_pass http://localhost:8000/api; # Reverse
proxy 설정
```

```
                proxy_redirect off; # 응답헤더와 바디에서 프록시 서버의 주소를 변경하지 않음
                charset utf-8;

                proxy_set_header X-Real-IP $remote_addr; # 실제IP주소를 요청헤더에 추가, 엑세스 제어, 로깅 및 분석 수행
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for; # 클라이언트로부터 받은 요청을 다른 서버로 전달
                proxy_set_header X-Forwarded-Proto $scheme; # 통신 프로토콜 식별
                proxy_set_header X-NginX-Proxy true; #원본 요청이 NGINX를 통해 전달되었는지 식별하기 위해서 사용
        }
}
}
```

- **/etc/nginx/sites-available/default**

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as
/drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        root /var/www/html/dist;

        index index.html;

        server_name i10b106.p.ssafy.io;

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a
404.
                try_files $uri $uri/ /index.html;
        }

        location /api {
                proxy_pass http://localhost:8000/api;
                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;
}


        location /api/ws/chat {
                proxy_pass http://localhost:8000/api/ws/chat;
                #proxy_http_version 1.1;
                #proxy_set_header Upgrade $http_upgrade;
            #   proxy_set_header Connection $connection_upgrade;
            #   proxy_set_header Host $host;
            #proxy_pass http://127.0.0.1:9000/; #실제 채팅서버의
아이피와 포트
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header Host $http_host;
                proxy_set_header X-Forwarded-For
$proxy_add_x_forwarded_for;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "Upgrade";
                proxy_set_header Accept-Encoding "";
        }
}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/
and symlink that
# to sites-enabled/ to enable it.
#
#server {
#       listen 80;
#       listen [::]:80;
#
#       server_name example.com;
#
#       root /var/www/example.com;
#       index index.html;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}

server {

        root /var/www/html/dist;

        # Add index.php to the list if you are using PHP
        index index.html;
    server_name i10b106.p.ssafy.io; # managed by Certbot



        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a
404.
```

```
                try_files $uri $uri/ /index.html;
        }


    listen [::]:443 ssl ipv6only=on; # managed by Certbot
    listen 443 ssl; # managed by Certbot
    ssl_certificate
/etc/letsencrypt/live/i10b106.p.ssafy.io/fullchain.pem; # managed
by Certbot
    ssl_certificate_key
/etc/letsencrypt/live/i10b106.p.ssafy.io/privkey.pem; # managed
by Certbot
    include /etc/letsencrypt/options-ssl-nginx.conf; # managed by
Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by
Certbot

}
server {
    if ($host = i10b106.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


        listen 80 ;
        listen [::]:80 ;
    server_name i10b106.p.ssafy.io;
    return 404; # managed by Certbot


}
```

- **.env.local**

```
VITE_REST_API={{서버 주소 입력}}/api
VITE_HOST_URL={{호스트 주소 입력}}

VITE_REST_USER_API=${VITE_REST_API}/user
VITE_REST_ACCUMULATE_API=${VITE_REST_API}/accumulate
VITE_REST_RECORDS_API=${VITE_REST_API}/records
VITE_REST_DRINKS_API=${VITE_REST_API}/drinks
VITE_REST_RANK_API=${VITE_REST_API}/rank
VITE_REST_SEARCH_API=${VITE_REST_API}/search
VITE_REST_RANKING_API=${VITE_REST_API}/ranking
VITE_REST_COMPARE_API=${VITE_REST_API}/compare
VITE_REST_RECOMMEND_API=${VITE_REST_API}/recommend
VITE_REST_CHAT_API=${VITE_REST_API}/chat
VITE_REST_KAKAO_API=${VITE_REST_API}/kakao-login

VITE_REST_KAKAO_LOGIN_API=http://${VITE_HOST_URL}/api/kakao-login
VITE_SOCKET_API=ws://${VITE_HOST_URL}:8000/ws/chat
VITE_KAKAO_API_KEY= {{카카오 앱키}}
```

첫번째 줄에 백엔드 주소
두번째 줄에는 *host* 주소
*VITE KAKAO API KEY*에 앱키 입력

❖ 필수 작성 사항

- application.yml 작성 (BE\src\main\resource 경로 위치)
- .env.local 작성 (FE\ 경로 위치)