

**October 07, 2022**

**SSAFY 7th TEAM 너나두리**

팀장: 박명규

팀원: 정지현, 김상현, 박종찬, 권민지, 이선호

목차

[개발 동기 2](#_Toc111773023)

[기술 스택 2](#_Toc111773024)

[빌드 상세내용 3](#_Toc111773025)

[백엔드 빌드 3](#_Toc111773026)

[빌드 4](#_Toc111773027)

[프론트 엔드 빌드 5](#_Toc111773028)

[배포 특이사항 6](#_Toc111773029)

[DB설정 7](#_Toc111773030)

[1. MySQL WorkBench 추가하기 7](#_Toc111773031)

[2. 계정 정보 입력 7](#_Toc111773032)

[프로퍼티 정의 8](#_Toc111773034)

[스프링부트 HTTPS 설정 10](#_Toc111773035)

[외부서비스 10](#_Toc111773036)

[TMap API 10](#_Toc111773037)

# 개발 동기

어떻게 하면 여행객들에게 더 직관적이고 더 빠른 정보를 전달할 수 있을까를 고민해서 나온 서비스가 바로 ‘너나두리’입니다. 여행을 가서 검색엔진(Naver, Google)을 통해 관광지와 주변 명소를 들어가기까지 4~5번의 광고를 접하고나서, 실제 사용자들이 작성한 포스팅을 볼 수 있습니다. 이러한 불편한 서비스를 이용하는 것이 아니라, 관광지와 주변 명소를 바로 확인하고 사용자들의 후기를 바로 볼 수 있는 서비스가 바로 너나두리입니다.

# 기술 스택

1. 이슈관리 : Jira

2. 형상관리 : Gitlab

3. 커뮤니케이션 : mattermost, notion, webex

4. 개발 환경

1. OS : window 10

2. IDE

1. Intellij IDEA 2022.1.3

2. Visual Studio Code

3. UI/UX : Miro

3. Database : MySQL Workbench

4. Server : AWS EC2 (MobaXterm)

1. Ubuntu 20.04.4 LTS

5. 상세 사용

1. Backend

1. Java 11

2. Spring Boot gradle

3. Spring Boot 2.7.1

4. Lombok, Jpa

2. Frontend

1. Nuxt.js 2.15.8

2. HTML5, CSS3, JavaScript(ES6)

3. vuex 4.1.0, vuetify 2.6.8, bootstrap 4.6.1

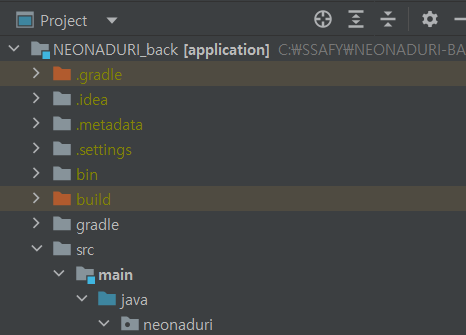
3. AWS

1. RDS(mysql)

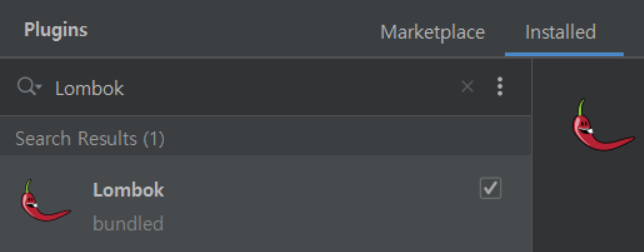
2. EC2

# 빌드 상세내용

## 백엔드 빌드

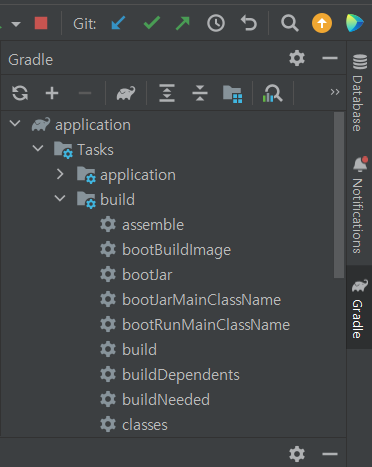


Intellij 에서 NEONADURI\_back을 gradle으로 import 합니다. import 이후, Lombok의 플러그인 설치를 해야합니다. Intellij의 File → Settings → Plugins 로 들어갑니다. Lombok을 검색한 뒤 설치 해줍시다.



installed에 Lombok이 설치되면 intellij를 재실행 해줍니다.

### 빌드



intellij 우측 상단에 Gradle 버튼을 클릭합니다.

위와 같이 뜨는 화면에 clean, validate, compile, install 순서대로 클릭을 하고 각각 실행이 완료 될 때까지 기다려 줍니다.

후에 NEONADURI\_back/target 폴더에 들어가면



jar 파일이 생성된 것을 확인할 수 있습니다.

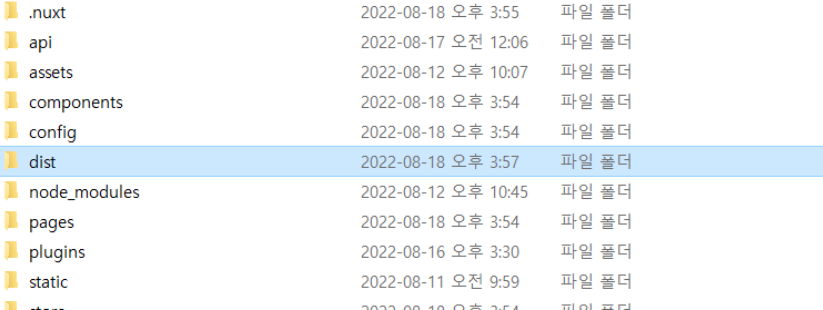
## 프론트 엔드 빌드

1. node\_module를 위한 기본 install

npm i

1. 빌드 명령어

npm run generate



dist 폴더 생성

## Jenkins 설정

프론트엔드 excute shell

sudo rm -rf /var/www/html/dist/

cd /var/lib/jenkins/workspace/NEONADURI\_front/NEONADURI\_front

sudo npm install --legacy-peer-deps

sudo npm run generate

sudo mv dist /var/www/html/

백엔드 excute shell

cd /var/lib/jenkins/workspace/NEONADURI\_back/NEONADURI\_back

sudo chmod 544 gradlew

./gradlew clean build

./gradlew build

sudo chmod 666 /var/run/docker.sock

docker image prune

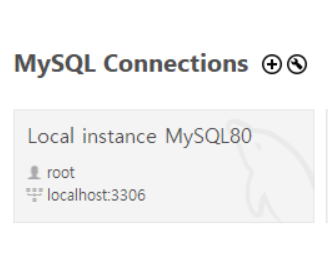
sudo docker rm -f neonaduri\_back

sudo docker build -t neonaduri\_back:0.1 .

sudo docker run -d -p 7070:7070 --name neonaduri\_back neonaduri\_back:0.1

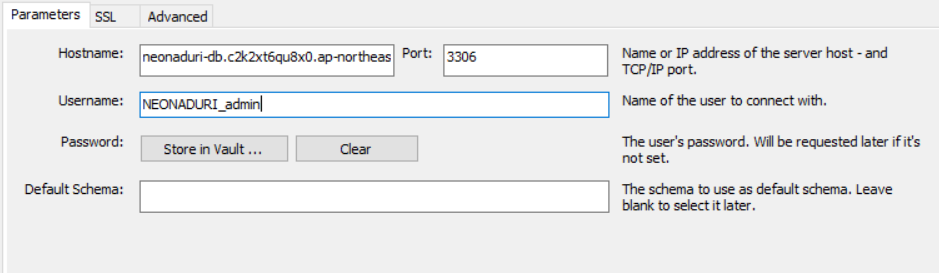
## DB설정

### 1. MySQL WorkBench 추가하기



MySQL WorkBench를 열어서 + 버튼을 눌러줍니다.

### 2. 계정 정보 입력



Hostname: jdbc:mysql://neonaduri-db.c2k2xt6qu8x0.ap-northeast-2.rds.amazonaws.com

Username: NEONADURI\_admin

Password: ssafyA702

# 프로퍼티 정의

1. nginx 세팅
   1. ec2에서 /etc/nginx/sites-available 파일로 접근

sudo apt-get update

* 1. default 파일 편집

sudo vi default

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html/dist;

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

server\_name \_;

location / {

try\_files $uri $uri/ =404;

}

}

server {

root /var/www/html/dist;

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

server\_name j7a702.p.ssafy.io; # managed by Certbot

location / {

try\_files $uri $uri/ =404;

}

location /api {

proxy\_pass http://j7a702.p.ssafy.io;

proxy\_redirect off;

charset utf-8;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-NginX-Proxy true;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

listen [::]:443 ssl ipv6only=on; # managed by Certbot

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/j7a702.p.ssafy.io/fullchain.pem;

ssl\_certificate\_key /etc/letsencrypt/live/j7a702.p.ssafy.io/privkey.pem;

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

server {

if ($host = j7a702.p.ssafy.io) {

return 301 https://$host$request\_uri;

}

listen 80 ;

listen [::]:80 ;

server\_name j7a702.p.ssafy.io;

return 404

}

## 스프링부트 HTTPS 설정

SSL 인증서 없이 웹 애플리케이션을 서버에 올리면 http로 접속이 되고 경고 메시지가 나타납니다.

application.yaml 파일에 위의 코드를 넣어주어 해결합니다.

ssl:

key-store: classpath:keystore.p12

key-store-type: PKCS12

key-password: ssafyA702

# Redis 설정

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get install redis-server

$ redis-server –version #Redis 설치 확인

$ sudo vi /etc/redis.conf

bind 127.0.0.1 >> bind 0.0.0.0 으로 변경

port 6379

$ sudo systemctl restart redis-server.service

application.yaml

spring:

redis:

host: j7a702.p.ssafy.io

port: 6379

timeout: 6

# S3

application.yaml

cloud:

aws:

s3:

bucket: neonaduri

region:

static: ap-northeast-2

credentials:

accessKey: AKIAVVSSTXJTQ36A5EH5

secretKey: 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

stack:

auto: false

# 외부서비스

# TMap 서비스

Tmap API url : <https://tmapapi.sktelecom.com/index.html>

SK Open API url : <https://openapi.sk.com/>

1. 회원가입
   * SK Open API 홈페이지에서 회원가입을 한다.
   * 마이페이지 → 앱
   * 앱 만들기를 통해서 NeoNaDuRi 앱을 만든다.
   * 앱이 만들어지면, 앱을 들어가서 앱키를 확인한다.
2. 앱키 사용
   * 경로안내 API : <https://tmapapi.sktelecom.com/main.html#webservice/sample/WebSampleRoutes>
   * 이 부분의 sample code에서 앱키에 해당하는 부분에 만들어진 앱키를 입력한다.
3. 요청 변수
   * reqCoordType : 요청 좌표 타입
   * resCoordType : 응답 좌표 타입
   * startName : 출발지 이름
   * startX : 출발지 경도
   * startY : 출발지 위도
   * endName : 도착지 이름
   * endX : 도착지 경도
   * endY : 도착지 위도
   * searchOption : 경로탐색 옵션
   * viaPoints : 경유지 목록
   * viaX : 경유지 경도
   * viaY : 경유지 위도
4. 응답 변수
   * Properties : 사용자 정의 프로퍼티 목록
   * totalDistance : 경로 총 길이
   * totalTime : 경로 총 소요 시간
   * type : geometry의 type
   * coordinates : 좌표 정보
   * index : 경로 순번
   * pointType : 안내지점의 구분