

포팅 매뉴얼

프로젝트 소개

Gilin은 사용자에게 지각 방지를 위한 맞춤형 알림과 경로 안내 기능을 제공하여 시간 관리를 돕는 스마트 앱입니다. 실시간 교통 상황과 사용자의 선호도를 기반으로 최적의 이동 솔루션을 제공합니다.

목적

- 사용자 맞춤형 알림과 경로 안내를 통해 지각 방지를 돕고, 효율적인 시간 관리를 지원
- 실시간 교통 정보와 예측 알고리즘을 활용하여 최적의 경로를 제공
- 막차 시간 알림, 대체 경로 제안 등 사용자 편의성 극대화

타겟

- 출퇴근 시간을 효율적으로 관리하고 싶은 직장인
- 통학 중 지각 방지를 원하는 학생
- 이동 계획을 실시간으로 조정하고 싶은 모든 사용자

서비스 개요

- **사용자 프로필 관리:** 출발지, 목적지, 선호 교통수단, 출퇴근/통학 시간 설정
- **실시간 경로 탐색 및 알림:** 현재 위치와 교통 상황을 기반으로 경로 제시 및 지각 여부 알림
- **막차 알림 기능:** 막차 시간까지의 경고 알림과 대체 경로 안내
- **개인화 서비스:** 사용자의 이동 기록에 따른 경로 시간 측정

타 서비스와의 차별성

- **실시간 최적 경로 제안:** 단순 경로 안내를 넘어 실시간 교통 데이터를 반영한 최적 경로 제공
- **막차 알림 시스템:** 막차 시간 알림 및 긴급 경로 조정을 통한 지각 방지
- **개인화된 이동 경험:** 사용자 선호도와 이동 기록을 기반으로 맞춤형 경로 추천
- **다중 교통수단 통합:** 버스, 지하철, 도보, 택시 등 다양한 교통수단을 통합적으로 지원

협업 툴

- 이슈 관리 : JIRA
- 형상 관리: Gitlab
- 커뮤니케이션 : Notion , Mattermost
- 디자인 : Figma

개발 환경

• Frontend

- flutter 3.24.4
- dart 3.5.4
- flutter_riverpod 2.6.1
- flutter_naver_map 1.3.0
- flutter_secure_storage: 9.2.2
- geolocator: 10.0.1
- geocoding: 2.1.0

• Backend

- IntelliJ : 17.0.11+1-b1207.24 amd64
- JDK : Java17 OpenJDK
- Spring boot : 3.3.1

외부 서비스

- OAuth
 - Kakao <https://developers.kakao.com/docs/latest/ko/kakaologin/rest-api>
- ODSay <https://lab.odsay.com/>
- Tmap API <https://tmapapi.tmapmobility.com/>
- 공공데이터포털 <https://www.data.go.kr/>
- 서울열린데이터광장 <https://data.seoul.go.kr/>

환경 변수 및 설정 파일 목록

- **Backend**

- `Application.yml` (공통)

```
server:
  port: 8080
  servlet:
    context-path: /api
spring:
  config:
    import: optional:file:.env[.properties]
  application:
    name: ${APPLICATION_NAME}
  datasource:
    url: jdbc:postgresql://${HOST}:${PORT}/${DATABASE}
    username: ${USER}
    password: ${PASSWORD}
    driver-class-name: org.postgresql.Driver
  jpa:
    properties:
      hibernate:
        dialect: org.hibernate.dialect.PostgreSQLDialect
    hibernate:
      ddl-auto: ${DDL_OPTION}
      defer-datasource-initialization: true
  data:
    redis:
      port: ${REDIS_PORT}
      host: ${HOST}
      password: ${REDIS_PW}
  sql:
    init:
      mode: always
  logging:
    level:
      org.springframework.web: "info"
      org.hibernate: "info"
      com.gilin: debug
  file:
```

```

    name: app/logs/application.log
apikey:
  odsays:
    ${ODSAY_API_CSK}
  tmap:
    ${TMAP_API_DINO}
  metro:
    ${SEOUL_METRO_OPEN_API_DINO}
jwt:
  accessKey: ${JWT_ACCESS_KEY}
  accessPeriod: ${JWT_ACCESS_PERIOD}
  refreshKey: ${JWT_REFRESH_KEY}
  refreshPeriod: ${JWT_REFRESH_PERIOD}

```

- ENV

```

APPLICATION_NAME=route
HOST=k11a306.p.ssafy.io
PORT=5432
DB_NAME=gilin
USER_NAME=gilin
USER_PASSWORD=ssafy306
DDL_OPTION=update
ODSAY_API_CSK=ODSAYKEY
REDIS_PORT=6379
REDIS_PW=ssafy!a306
TMAP_API_DINO=TMAPKEY
JWT_ACCESS_KEY=cnldjqgkrhtlvekcndjqgkrhtlvekcndjqgkrhtlv
JWT_ACCESS_PERIOD=3600000
JWT_REFRESH_KEY=cnldjqgkrhtlvekcndjqgkrhtlvekcndjqgkrhtlv
JWT_REFRESH_PERIOD=86400000

```

빌드 및 배포

도커 설정

-

```

apt-get update
apt-get install docker

docker network create bookkoo-net

# 사용할 PostgreSQL DB 컨테이너들 생성
docker run -d \
  --name postgres \
  --network npm-net \
  -e POSTGRES_USER={DB 패스워드} \
  -e POSTGRES_PASSWORD={DB 패스워드} \
  -e POSTGRES_DB={DB 명} \
  -v {사용할 볼륨}

# redis DB 컨테이너 생성
docker run -d \
  --name {서비스명}-redis \
  --network npm-net \
  redis:latest

# Nginx Proxy Manager
docker run -d \
  --name nginx-proxy-manager \
  --network npm-net \
  --restart always \
  -p 80:80 \
  -p 443:443 \
  -p {외부에서 접속할 포트}:81 \
  -e DB_MYSQL_HOST={npm DB 주소} \
  -e DB_MYSQL_PORT={npm DB 주소} \
  -e DB_MYSQL_USER={npm DB 주소} \
  -e DB_MYSQL_PASSWORD={npm DB 주소} \
  -e DB_MYSQL_NAME={npm DB 주소} \
  -v {사용할볼륨} \
  jc21/nginx-proxy-manager:latest

```

빌드

```
docker compose up -d build
```