

Backend 개발 명세서

1. 기술 스택

- 언어(Language): Java 17
- 프레임워크(Framework): Spring Boot 3.4.7, Spring Security, Spring Data JPA
- 인증(Authentication): JWT (JSON Web Token)
- API 문서화: Springdoc, Swagger
- 데이터베이스(Database): MariaDB
- 빌드 도구(Build Tool): Maven

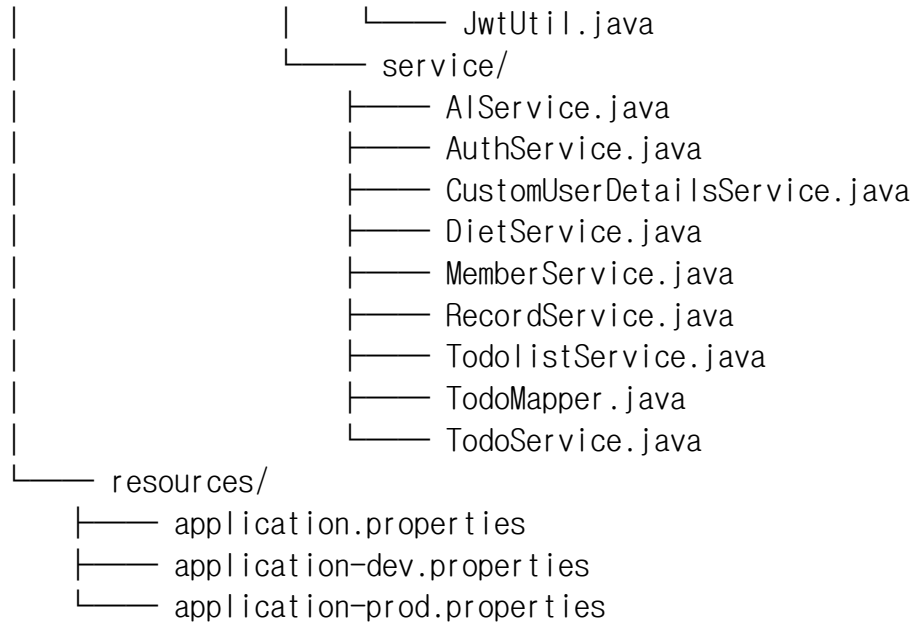
2. 프로젝트 구조

```
backend/
├── pom.xml
├── Dockerfile
├── docker-compose.yml
├── mvnw
├── mvnw.cmd
├── .gitignore
├── .gitattributes
├── .idea/
├── .mvn/
├──   ├── wrapper/
├──   └── maven-wrapper.properties
├── src/
├──   ├── main/
├──   └──   ├── java/
├──   └──   └──   ├── com/
├──   └──   └──   └── example/
├──   └──   └──   └── thirdprojectback/
├──   └──   └──   └──   ├── ThirdprojectBackApplication.java
├──   └──   └──   └──   ├── config/
├──   └──   └──   └──   └── SecurityConfig.java
├──   └──   └──   └──   ├── controller/
├──   └──   └──   └──   ├── AuthController.java
├──   └──   └──   └──   ├── DietController.java
├──   └──   └──   └──   ├── MemberController.java
├──   └──   └──   └──   └── RecordController.java
```

```

├── TodoController.java
├── TodoListController.java
├── converter/
│   └── StringListConverter.java
├── dto/
│   ├── AIRequestDto.java
│   ├── AIResponseDto.java
│   ├── DietDto.java
│   ├── DietResponseDto.java
│   ├── GraphResponseDto.java
│   ├── LoginRequest.java
│   ├── MemberRequestDto.java
│   ├── MemberResponseDto.java
│   ├── MemberUpdateDto.java
│   ├── RecordRequestDto.java
│   ├── RecordResponseDto.java
│   ├── SignupResponse.java
│   ├── TodoListRequestDto.java
│   ├── TodoListResponseDto.java
│   ├── TodoRequestDto.java
│   ├── YouTubeResponseDto.java
│   └── YouTubeVideoDto.java
├── entity/
│   ├── Diet.java
│   ├── Disease.java
│   ├── Member.java
│   ├── Record.java
│   ├── Todo.java
│   └── TodoList.java
├── repository/
│   ├── DietRepository.java
│   ├── MemberRepository.java
│   ├── RecordRepository.java
│   ├── TodoRepository.java
│   └── TodoListRepository.java
├── runner/
│   └── DataInitRunner.java
├── security/
│   ├── CustomUserDetails.java
│   └── JwtAuthenticationFilter.java

```



3. 주요 설정

```
#application-dev.properties
spring.datasource.url=jdbc:mariaadb://127.0.0.1:3306/healthdb
spring.datasource.username=gym
spring.datasource.password=gym
spring.datasource.driverClassName=org.mariadb.jdbc.Driver

spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
spring.jpa.database-platform=org.hibernate.dialect.MariaDBDialect

logging.level.org.springframework.security=DEBUG
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.orm.jdbc.bind=TRACE

ai.server.url=http://localhost:8003/healthai/invoke
```

4. 주요 클래스

4.1 Entity

- Diet
- Disease
- Member

- Record
- Todo
- TodoList

4.2 Controller

- AuthController
- DietController
- MemberController
- RecordController
- TodoController
- TodoListController

4.3 Service

- AIService
- AuthService
- CustomUserDetailsService
- DietService
- MemberService
- RecordService
- TodoListService
- TodoMapper
- TodoService

4.4 Repository

- DietRepository
- MemberRepository
- RecordRepository
- TodoRepository
- TodoListRepository

5. 보안 구현

- JWT 토큰 기반 인증
 - Spring Security 설정
 - 사용자 인증을 위한 JwtAuthenticationFilter, CustomUserDetails 구성
 - CORS 설정은 SecurityConfig 를 통해 허용 도메인 정의
-

6. AI 모듈 연동

- AIService 를 통해 AI 추천 로직 처리
 - 외부 AI 응답 데이터를 주고받기 위한 통신 구현 (RestTemplate)
-

7. 파일 처리

- 향후 MultipartFile, FileStorageService 등을 통해 PDF, DOCX, TXT 업로드 기능 확장 가능
-

8. 빌드 및 배포

- Maven 으로 프로젝트 빌드 (pom.xml 기반)
- Dockerfile 및 docker-compose.yml 을 이용한 도커 기반 배포 환경 구성
- Spring 프로파일 분리를 위한 application-dev.properties, application-prod.properties 구성