

# 한국스마트정보교육원

## 51기 2팀 프로젝트 완료 보고서

프로젝트명	
1인 가구 공동구매 매칭 플랫폼	

팀장	고윤영
팀원	오민호
	이희상
	전우석
	한동환
작성일	2024.07.22

호스팅	<a href="https://gongzone.shop">https://gongzone.shop</a>
GitHub	<a href="https://github.com/ks-gongzone.git">https://github.com/ks-gongzone.git</a>

## 목차

### 1 . 프로젝트 기본정보

#### 1.1 시스템 흐름도

### 2 . 요구사항 정의서

#### 2.1 기능적 요구사항

### 3 . 업무분담(담당업무)

### 4 . 설계사양서

#### 4.1 ERD 물리모드

#### 4.2 테이블 구조

#### 4.3 네이밍 규칙

#### 4.4 패키지 구조

### 5 . 개발환경

### 6 . 주요기능별정리

## 1. 프로젝트 기본정보

프로젝트명	1인 가구 공동구매 매칭 플랫폼	
개발기간	■ 2024.04.16 ~2024.08.01 (122일)	
	2024.04.01 ~2024.04.22 (21일)	프로젝트 주제 선정 및 사전조사
	2024.04.22 ~2024.05.21 (30일)	<ul style="list-style-type: none"> <li>- 시스템 구조도 작성</li> <li>- 설계자료 취합, 수정 및 보완</li> <li>- 테이블 설계</li> <li>- 와이어프레임 작성</li> </ul>
	2024.05.21 ~ 2024.06.06 (16일)	<ul style="list-style-type: none"> <li>- 서버 설정</li> <li>- 개발환경 설정</li> <li>- Hosting 설정</li> </ul>
	2024.06.06 ~2024.07.22 (46일)	- 기능 구현(개인 분담 프로그래밍)
	2024.07.22 ~ 2024.07.26(5일)	<ul style="list-style-type: none"> <li>- 테스트, 오류 수정 및 보완</li> <li>- 미완성 기능 구현</li> </ul>
	2024.07.26 ~ 2024.08.01(7일)	- 완료보고서 작성
개발목적	<ol style="list-style-type: none"> <li>1. 1인 가구 간에 식품(배달 포함)을 매칭 서비스를 통하여 공유할 수 있도록 함</li> <li>2. 1인 가구가 증가함에 따라 높아지는 사회적으로 낭비되는 자원을 최소화할 수 있음</li> <li>3. 팬데믹 시기를 겪으며 소외된 1인 가구의 사회 활동 유도할 수 있음</li> </ol>	
개발방식	 <pre> graph TD     A[계획] --&gt; B[요구분석]     B --&gt; C[설계]     C --&gt; D[중요설정 우선 코딩, 테스트]     D --&gt; E[우선순위 코딩, 테스트]     E --&gt; F[출시]     F --&gt; G[낮은 순위 코딩, 테스트]     G --&gt; E     E --&gt; C     C --&gt; B     B --&gt; A     </pre>	

## 2. 요구사항 정의서

### 2.1 기능적 요구사항

전체기능	관리자	파티장	일반이용자
1.회원관리	1. 회원관리	1. 회원관리	1. 회원관리
1.1 회원등록		1.1 회원등록	1.1 회원등록
1.2 회원정보수정	1.2 회원정보수정	1.2 회원정보수정	1.2 회원정보수정
1.3 회원정보조회	1.3 회원정보조회	1.3 회원정보조회	1.3 회원정보조회
1.4 회원삭제	1.4. 회원삭제	1.4. 회원탈퇴	1.4. 회원탈퇴
2.로그인 관리	2.로그인 관리	2.로그인 관리	2.로그인 관리
2.1 로그인	2.1 로그인	2.1 로그인	2.1 로그인
2.2 로그아웃	2.2 로그아웃	2.2 로그아웃	2.2 로그아웃
3.게시글 기능	3.게시글 기능	3.게시글 기능	3.게시글 기능
3.1 게시물 등록	3.1 게시물 등록	3.1 게시물 등록	3.1 게시물 등록
3.2 게시물 조회	3.2 게시물 조회	3.2 게시물 조회	3.2 게시물 조회
3.3 게시물 수정	3.3 게시물 수정	3.3 게시물 수정	
3.4 게시물 삭제	3.4 게시물 삭제	3.4 게시물 삭제	
4.찜 기능	4.찜 기능	4.찜 기능	4.찜 기능
4.1 찜목록 조회	4.1찜목록 조회	4.1 찜목록 조회	4.1 찜목록 조회
4.2 찜목록 추가		4.2 찜목록 추가	4.2 찜목록 추가
4.3 찜목록 삭제		4.3 찜목록 삭제	4.3 찜목록 삭제
5.카테고리 기능	5.카테고리 기능	5.카테고리 기능	5.카테고리 기능
5.1 카테고리 등록			
5.2 카테고리 조회	5.2 카테고리 조회	5.2 카테고리 조회	5.2 카테고리 조회
5.3 카테고리 수정			

5.4 카테고리 삭제			
6.위치기반 서비스	6.위치기반 서비스	6.위치기반 서비스	6.위치기반 서비스
6.1 위치기반 등록	6.1 위치기반 등록	6.1 위치기반 등록	6.1 위치기반 등록
6.2 위치기반 조회	6.2 위치기반 조회	6.2 위치기반 조회	6.2 위치기반 조회
6.3 위치기반 수정	6.3 위치기반 수정	6.3 위치기반 수정	6.3 위치기반 수정
6.4 위치기반 삭제	6.4 위치기반 삭제	6.4 위치기반 삭제	6.4 위치기반 삭제
7.모임(파티) 기능	7.모임(파티) 기능	7.모임(파티) 기능	7.모임(파티) 기능
7.1 파티 목록 조회	7.1파티 목록 조회	7.1파티 목록 조회	7.1파티 목록 조회
7.2 파티 생성		7.2 파티 생성	7.2 파티 생성
7.3 파티 수정	7.3 파티 수정	7.3 파티 수정	
7.4 파티 삭제	7.4 파티 삭제	7.4 파티 삭제	
8.포인트 기능	8.포인트 기능	8.포인트 기능	8.포인트 기능
8.1 포인트 충전	8.1 포인트 충전	8.1 포인트 충전	8.1 포인트 충전
8.2 포인트 조회	8.2 포인트 조회	8.2 포인트 조회	8.2 포인트 조회
8.3 포인트 출금	8.3 포인트 출금	8.3 포인트 출금	8.3 포인트 출금
9.결제 기능	9.결제 기능		9.결제 기능
9.1 대금 결제	9.1 대금 결제		9.1 대금 결제
9.2 환불(거래전)	9.2 환불(거래전)		9.2 환불(거래전)
10.거래 현황 기능	10.거래 현황 기능	10.거래 현황 기능	10.거래 현황 기능
10.1 거래 현황 등록	10.1 거래 현황 등록	10.1 거래 현황 등록	10.1 거래 현황 등록
10.2 거래 현황 조회	10.2 거래 현황 조회	10.2 거래 현황 조회	10.2 거래 현황 조회
10.3 수정	10.3 수정	10.3 수정	
10.4 삭제	10.4 삭제	10.4 삭제	
11.정산 기능	11.정산 기능	11.정산 기능	11.정산 기능

11.1 대금 지급	11.1 대금 지급	11.1 대금 지급	
11.2 정산 현황 조회	11.2 정산 현황 조회	11.2 정산 현황 조회	11.2 정산 현황 조회
11.3 환불(거래후)			11.3 환불(거래후)
12.알림 기능	12.알림 기능	12.알림 기능	12.알림 기능
12.1 알림 전송	12.1 알림 전송		
12.2 알림 조회	12.2 알림 조회	12.2 알림 조회	12.2 알림 조회
12.3 알림 삭제	12.3 알림 삭제	12.3 알림 삭제	12.3 알림 삭제
13.회원 상호작용	13.회원 상호작용	13.회원 상호작용	13.회원 상호작용
13.1 회원 팔로우 기능	13.1 회원 팔로우 기능	13.1 회원 팔로우 기능	13.1 회원 팔로우 기능
13.2 회원 차단 기능	13.2 회원 차단 기능		13.2 회원 차단 기능
13.3 쪽지 기능	13.3 쪽지 기능	13.3 쪽지 기능	13.3 쪽지 기능
14.사용자평가관리	14.사용자평가관리	14.사용자평가관리	14.사용자평가관리
14.1 평가 등록		14.1 평가 등록	14.1 평가 등록
14.2 평가 조회	14.2 평가 조회	14.2 평가 조회	14.2 평가 조회
15. 고객지원 기능	15.고객지원 기능	15.고객지원 기능	15.고객지원 기능
15.1 공지사항	15.1 공지사항	15.1 공지사항	15.1 공지사항
15.2 FAQ 기능	15.2 FAQ 기능	15.2 FAQ 기능	15.2 FAQ 기능
15.3 버그 제보	15.3 버그 제보	15.3 버그 제보	15.3 버그 제보
15.4 회원 제재	15.4 회원 제재		
16.통계 관리	16.통계 관리		
16.1 사용자 통계	16.1 사용자 통계		
16.2 파티 통계	16.2 파티 통계		

### 2.1.1 기능 보완사항

No.	기능적 요구사항	관리자	일반 사용자
	회원 등급 관리	회원 등급 관리	회원 등급 관리
	결제		토스 외 타사 api 연결
	결제 취소 및 환불		마일리지 변경 및 api

### 3. 업무분담(담당 업무)

이름	업무 내용
고윤영	<p>기획, 설계</p> <ul style="list-style-type: none"> <li>- PM 경험을 활용하여 회의 진행 주도</li> <li>- PPT 형태 기획 발표 자료 작성</li> </ul> <p>프론트엔드</p> <ul style="list-style-type: none"> <li>- react-dom-route를 활용한 React 폴더트리 및 아키텍처 설계</li> <li>- 홈페이지 전반 레이아웃 구성 및 재사용할 수 있는 basic 컴포넌트 작성</li> <li>- Axios 및 Zustand를 활용한 Rest API 통신환경 구축</li> </ul> <p>위치 기반 기능</p> <ul style="list-style-type: none"> <li>- 카카오 지도 API를 활용하여 사용자 반경 1km에 있는 파티 조회 기능 구현</li> <li>- 게시글 작성 시 주소 입력 및 클릭 이벤트로 해당 위치의 주소를 자동으로 입력하는 기능 구현</li> </ul> <p>파티 관리 기능</p> <ul style="list-style-type: none"> <li>- 게시글 작성에 따른 공동구매 희망자 파티 생성</li> <li>- 스프링부트를 활용하여 파티 신청 및 수락, 거절, 강퇴 등 파티 진행 프로세스 기능 구현</li> </ul> <p>관리자 페이지 기능</p> <ul style="list-style-type: none"> <li>- 비정상 유저를 파악을 위한 게시글 작성 횟수가 많은 회원 조회 기능 구현</li> <li>- 파티 별 현재 프로세스 진행 현황 조회 기능 구현</li> <li>- 현재 처리되지 않은 고객 컴플레인 조회 기능 구현</li> <li>- 로그인 로그 조회 기능 구현</li> </ul>
오민호	<p>회원 기능</p> <p>JWT토큰 방식을 활용한 로그인 및 회원가입 구현</p> <ul style="list-style-type: none"> <li>- 세션을 사용하지 않고 토큰을 활용하여 사용자 인증을 처리</li> <li>- 스프링 시큐리티를 활용하여, 토큰을 소지하지 않은 사용자에게 대한 접근을 제한</li> <li>- 인증이 필요한 엔드포인트에 대해 시큐리티 설정을 통해 보호</li> <li>- 주스탠드를 사용하여 경량화 및 상태 관리</li> </ul> <p>리프레시 토큰 사용</p> <ul style="list-style-type: none"> <li>- 소셜 로그인 통합</li> <li>- 카카오, 네이버, 구글 소셜 로그인 API를 연동하여, 사용자가 소셜 계정으로 로그인 및 회원가입 할 수 있도록 구현</li> <li>- 소셜 로그인 성공 시, 소셜 사용자 정보를 기반으로 토큰을 발급</li> </ul>

	<p>로그인 로그</p> <p>파티 기능</p> <ul style="list-style-type: none"> <li>- 파티 정보 조회 및 관련된 정보 조회 기능</li> <li>- 파티 수락 거절 강퇴 구현</li> </ul> <p>쪽지 및 알람 기능</p> <ul style="list-style-type: none"> <li>- webflux 방식으로 쪽지 기능 구현</li> <li>- SSE 방식을 활용한 실시간 알람 기능 구현</li> </ul> <p>관리자 기능</p> <ul style="list-style-type: none"> <li>- 회원 탈퇴, 휴면, 제재 관리 기능, 문의 내역 관리 기능</li> </ul> <p>에러 처리 시스템 구현, ERD 작성</p>
<p><b>이희상</b></p>	<p>게시글 작성 기능 구현</p> <ul style="list-style-type: none"> <li>- 프로젝트 메인 프로세스의 시작</li> <li>- 이미지 미리보기 및 업로드 기능 구현</li> <li>- 카카오 지도 API와 DAUM 주소 API를 활용하여 지도 기능 구현 <ul style="list-style-type: none"> <li>↳ 검색한 지번 주소를 도로명 주소로 자동 변환</li> <li>↳ 검색한 주소 카카오 지도와 연동하여 지도에 표시</li> <li>↳ 지도 클릭 시 해당 위치의 주소를 도로명 주소로 표시</li> </ul> </li> <li>- CKEditor 라이브러리를 활용하여 본문 내용 작성 기능 구현</li> <li>- React-Datepicker 라이브러리를 활용하여 달력 클릭 시 날짜 입력 기능 구현</li> </ul> <p>게시글 관리 기능 구현</p> <ul style="list-style-type: none"> <li>- 조건에 맞는 게시글 검색 기능 구현</li> <li>- 게시글 수정 및 삭제 기능 구현</li> <li>- 사용자 게시글 찜 기능 구현</li> <li>- 게시글 조회 수 기능 구현</li> </ul> <p>게시글 댓글 기능 구현</p> <ul style="list-style-type: none"> <li>- 사용자 댓글 실시간 반영</li> <li>- 댓글 수정 및 삭제 기능 구현</li> </ul> <p>관리자 기능</p> <ul style="list-style-type: none"> <li>- 게시글 작성 횟수 Top 5 회원 조회 기능 구현</li> <li>- 파티 별 진행 현황 조회 기능 구현</li> </ul>
<p><b>전우석</b></p>	<p>포인트 기능</p> <ul style="list-style-type: none"> <li>- PG사 연동 API를 통한 전자 결제/인출 프로세스 구현</li> <li>- 결제 프로세스 중 발생할 수 있는 위/변조 위험 검증</li> <li>- Spring WebFlux 활용하여 서버 측 비동기 통신 구현</li> <li>- 외부 서버 JSON 데이터 파싱 및 클라이언트 제공 로직 개발</li> </ul> <p>파티 결제 및 정산 기능</p> <ul style="list-style-type: none"> <li>- 포인트 기반 파티 결제 시스템 개발 및 트랜잭션 관리</li> <li>- 프로세스 자동화를 통해 사용자 편의성 및 시스템 효율성 확보</li> </ul> <p>프론트엔드 구현</p> <ul style="list-style-type: none"> <li>- Zustand와 React Context API를 활용한 상태 관리 최적화</li> <li>- Custom Hook을 이용한 상태 관리 효율성 확보</li> </ul>



	<ul style="list-style-type: none"> <li>- Router를 이용한 공통 레이아웃 관리 및 렌더링 성능 최적화</li> <li>- Tailwind CSS를 이용한 반응형 웹 구현</li> </ul> <p>아키텍처 설계</p> <ul style="list-style-type: none"> <li>- Spring MVC 아키텍처 구조 설계</li> <li>- MyBatis 영속성 계층 설계</li> <li>- RESTful 아키텍처를 적용한 API 엔드포인트 설계</li> </ul> <p>배포 환경 구축</p> <ul style="list-style-type: none"> <li>- Jenkins 활용하여 CI/CD 파이프라인 구축</li> <li>- Linux(CentOS) 기반 지속적 통합/배포 환경 구성</li> <li>- SSL 인증서 발급 및 HTTPS 프로토콜 적용</li> </ul>
<p><b>한동환</b></p>	<p>전체 회원 프로필 카드 구현</p> <ul style="list-style-type: none"> <li>↳ 회원별 상태 구분기능 구현(인기유저, 공존의난동꾼, 일반유저)</li> <li>↳ 회원리스트 페이지네이션 처리(이전 페이지 다음페이지 버튼 클릭하여 이동)</li> <li>↳ 회원 이름 검색 후 조건에 맞는 데이터 노출 기능</li> </ul> <p>마이페이지에서 특정 회원 정보 수정 및 탈퇴</p> <ul style="list-style-type: none"> <li>↳ 프로필 사진 등록시 React-easy-crop 라이브러리를 이용한 사진 편집 후 파일파싱 (blob으로 반환 된 데이터 확장자 및 파일 이름 파싱 로직 구현)</li> <li>↳ React-csv 라이브러리를 이용한 국토교통부 법정동 주소명 csv 파일 로드 후 db에 맞게 데이터 가공처리 (read, update)</li> <li>↳ 알람 수신 동의 데이터 변경 시 변경 날짜 로그 기록</li> </ul> <p>공지사항 타입별 작성 기능 (FAQ, 공지, 프로모션)</p> <ul style="list-style-type: none"> <li>↳ CKEditor라이브를 이용한 본문 작성 기능 구현</li> <li>↳ 전체 공지사항 타입별 노출 및 10개의 데이터씩 페이지네이션 처리(1~n 번 페이지 클릭 시 해당 페이지 이동)</li> <li>↳ 공지사항 클릭시 조회수 증가 counting 및 최 상단으로 이동</li> </ul>







## 2) 게시물





#### 4) 포인트

<b>회원보유포인트</b>	
	회원보유포인트고유번호
	회원고유번호 포인트잔액
<b>포인트증감내역</b>	
	포인트증감내역고유번호
	회원보유포인트고유번호
	유형코드 보유포인트(변동전) 포인트변동량 보유포인트(변동후) 변동일시
	상태코드
<b>포인트충전내역</b>	
	포인트충전고유번호
	포인트증감내역고유번호
	결제방법 트랜잭션타입 결제bid 결제id 결제코드 결제메시지 요청일시
	상태코드
<b>포인트인출내역</b>	
	포인트인출고유번호
	포인트증감내역고유번호
	지급은행 지급계좌 예금주 지급금액 요청일시
	상태코드

## 5) 파일



## 6) 코드 Enum 테이블





## 4.2 테이블 구조

### - code

테이블 정의서		물리 테이블명	논리 테이블명	유형 코드	승인자	작성일	페이지수				
단계	설계	작성자	작성일	2024-07-29	변천	1	1				
순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	유형 코드	type_code	VARCHAR	20	Y	N	NOT NULL				
2	코드 그룹명	code_group	VARCHAR	10	N	N	NOT NULL				
3	코드 중분류	code_group_sub	VARCHAR	10	N	N	NOT NULL				
4	코드 설명	code_description	VARCHAR	50	N	N	NOT NULL				

테이블 정의서		물리 테이블명	논리 테이블명	유형 코드	승인자	작성일	페이지수				
단계	설계	작성자	작성일	2024-07-29	변천	1	1				
순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	상태 코드	status_code	VARCHAR	20	Y	N	NOT NULL				
2	코드 그룹명	code_group	VARCHAR	10	N	N	NOT NULL				
3	코드 중분류	code_group_sub	VARCHAR	10	N	N	NOT NULL				
4	코드 설명	code_description	VARCHAR	50	N	N	NOT NULL				

### - member

테이블 정의서		물리 테이블명	논리 테이블명	유형 코드	승인자	작성일	페이지수				
단계	설계	작성자	작성일	2024-07-29	변천	1	1				
순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	회원 코드	m_level	INT		Y	N	NOT NULL				
2	회원명	m_level_name	VARCHAR	10	N	N	NOT NULL				

테이블 정의서		물리 테이블명	논리 테이블명	유형 코드	승인자	작성일	페이지수				
단계	설계	작성자	작성일	2024-07-29	변천	1	1				
순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	회원 그룹 번호	m_no	VARCHAR	10	Y	N	NOT NULL				
2	회원 코드	m_level	INT		N	Y	NOT NULL		member_level	m_level	
3	아이디	m_id	VARCHAR	50	N	N	NOT NULL				
4	비밀번호	m_pw	VARCHAR	50	N	N	NULL	NULL			소셜로그인 회원은 비밀번호가 없음
5	이름	m_name	VARCHAR	20	N	N	NOT NULL				
6	이메일	m_email	VARCHAR	40	N	N	NOT NULL				
7	전화번호	m_phone	VARCHAR	20	N	N	NOT NULL				
8	성별	m_gender	CHAR	1	N	N	NOT NULL				
9	주소	m_addr	VARCHAR	100	N	N	NULL	NULL			
10	생년월일	m_birthday	DATE		N	N	NULL	NULL			
11	비밀번호	m_pwd	VARCHAR	20	N	N	NULL	NULL			
12	회원 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

## - login

테이블 정의서		물리 테이블명	token	논리 테이블명	스플릿 로그인 토큰	승인자
단계	설계	작성자	조현	작성일	2024-07-23	변경
단개	설계	업무명				페이지수
						1
						1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	토큰 고유번호	token_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	회원 고유번호	u_no	VARCHAR	10	N	Y	NOT NULL		member	u_no	
3	토큰 타입	token_type	VARCHAR	20	N	N	NOT NULL				
4	토큰 값(유효성)	token_value_acc	VARCHAR	500	N	N	NOT NULL				
5	토큰 값(링크주소)	token_value_ref	VARCHAR	500	N	N	NULL	NULL			
6	만료일(유효성)	token_expire_acc	DATE		N	N	NOT NULL				
7	만료일(링크주소)	token_expire_ref	DATE		N	N	NULL	NULL			
8	최근수정일	token_last_update	DATE		N	N	NOT NULL				

테이블 정의서		물리 테이블명	login	논리 테이블명	로그인 기록	승인자
단계	설계	작성자	조현	작성일	2024-07-23	변경
단개	설계	업무명				페이지수
						1
						1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	로그인 기록 고유번호	login_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	회원 고유번호	u_no	VARCHAR	10	N	Y	NOT NULL		member	u_no	
3	로그인 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	사용 브라우저	login_browser	VARCHAR	20	N	N	NOT NULL				
5	로그인 시도 일시	login_in_time	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	세션 종료 일시	login_out_time	TIMESTAMP		N	N	NULL	NULL			
7	로그인 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

## - note

테이블 정의서		물리 테이블명	note	논리 테이블명	목지 내용	승인자
단계	설계	작성자	조현	작성일	2024-07-23	변경
단개	설계	업무명				페이지수
						1
						1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	목지 고유 번호	note_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	본문 회원 고유 번호	u_no	VARCHAR	10	N	Y	NOT NULL		member	u_no	
3	수신 회원 고유 번호	u_target_no	VARCHAR	10	N	Y	NOT NULL		member	u_no	
4	목지 내용	note_body	VARCHAR	500	N	N	NOT NULL				
5	본문 일시	note_send_time	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	읽은 일시	note_read_time	TIMESTAMP		N	N	NULL	NULL			
7	목지 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	
8	수신 회원 목지 상태	status_code_target	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	



- alert

테이블 정의서		물리 테이블명	alert	논리 테이블명	최근 유효 내역	승인자	
		작성자	그림	작성일	2024-07-23	버전	1
단계	설계	업무명				페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	발행 기록 고유 번호	alert_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	발행 수신 회원 고유 번호	rn_no	VARCHAR	10	N	Y	NOT NULL		member	rn_no	
3	발행 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	발행 상세 메시지	alert_detail	VARCHAR	500	N	N	NOT NULL				
5	발행 분반 일시	alert_utime	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	발행 읽은 일시	alert_readtime	TIMESTAMP		N	N	NULL	NULL			
7	발행 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

테이블 정의서		물리 테이블명	alert_allow	논리 테이블명	발행 수신 여부	승인자	
		작성자	그림	작성일	2024-07-23	버전	1
단계	설계	업무명				페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	발행 수신 여부 고유 번호	alert_allow_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	회원 고유 번호	rn_no	VARCHAR	10	N	Y	NOT NULL		member_no	rn_no	
3	SMS 수신 여부	allow_sms	TINYINT	1	N	N	NOT NULL				
4	Email 수신 여부	allow_email	TINYINT	1	N	N	NOT NULL				
5	광고성 영문 수신 여부	allow_marketing	TINYINT	1	N	N	NOT NULL				
6	회원 발행 수신 여부	allow_member	TINYINT	1	N	N	NOT NULL				
7	문자 발행 수신 여부	allow_note	TINYINT	1	N	N	NOT NULL				
8	게시물 발행 수신 여부	allow_bulletin	TINYINT	1	N	N	NOT NULL				
9	편지 발행 수신 여부	allow_party	TINYINT	1	N	N	NOT NULL				

테이블 정의서		물리 테이블명	alert	논리 테이블명	최근 유효 내역	승인자	
		작성자	그림	작성일	2024-07-23	버전	1
단계	설계	업무명				페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	발행 기록 고유 번호	alert_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	발행 수신 회원 고유 번호	rn_no	VARCHAR	10	N	Y	NOT NULL		member	rn_no	
3	발행 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	발행 상세 메시지	alert_detail	VARCHAR	500	N	N	NOT NULL				
5	발행 분반 일시	alert_utime	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	발행 읽은 일시	alert_readtime	TIMESTAMP		N	N	NULL	NULL			

## - 팔로우, 차단

테이블 정의서		물리 테이블명	논리테이블명	표준 팔로우 목록	승인자
단계	설명	작성자	작성일	2024-07-29	배진
		업무명			페이지수
					1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	팔로우 고유 번호	follow_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	팔로우 원 팔로우 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	팔로우 대상 팔로우 번호	n_target_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
4	팔로우 일시	follow_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

테이블 정의서		물리 테이블명	논리테이블명	표준 차단 목록	승인자
단계	설명	작성자	작성일	2024-07-29	배진
		업무명			페이지수
					1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	차단 고유 번호	block_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	차단원 팔로우 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	차단 대상 팔로우 번호	n_target_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
4	차단 일시	block_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

## - 탈퇴, 휴면

테이블 정의서		물리 테이블명	논리테이블명	표준 탈퇴 목록	승인자
단계	설명	작성자	작성일	2024-07-29	배진
		업무명			페이지수
					1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	탈퇴 기록 고유 번호	n_quit_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	탈퇴 고유 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	탈퇴 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	탈퇴 상세 사유	quit_reason_detail	VARCHAR	200	N	N	NULL	NULL			
5	탈퇴 일시	quit_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

테이블 정의서		물리 테이블명	논리테이블명	표준 휴면 목록	승인자
단계	설명	작성자	작성일	2024-07-29	배진
		업무명			페이지수
					1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	휴면 기록 고유 번호	n_sleep_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	휴면 고유 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	마지막 로그인 일시	n_last_login	TIMESTAMP		N	N	NOT NULL				
4	휴면 종료 일시	n_sleep_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

## - 제재, 신고

테이블 정의서		물리 테이블명	논리 테이블명	생성일	승인자
단개	복개	member_punish	member_punish	2024-07-25	변진
작성자	업무명	그룹	작성일	페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	제재 기록 고유 번호	n_punish_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	제재 회원 고유 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	제재 관리자 고유 번호	n_admin_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
4	제재 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
5	제재 사유 상세	punish_reason	VARCHAR	500	N	N	NOT NULL				
6	제재 일시	punish_start_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
7	제재 기간	punish_period	VARCHAR	50	N	N	NOT NULL				
8	제재 해제 예정 일시	punish_end_date	TIMESTAMP		N	N	NOT NULL				
9	제재 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

테이블 정의서		물리 테이블명	논리 테이블명	생성일	승인자
단개	복개	member_report	member_report	2024-07-25	변진
작성자	업무명	그룹	작성일	페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	신고 기록 고유 번호	n_report_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	신고된 회원 고유 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	신고 대상 회원 고유 번호	n_target_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
4	신고 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
5	신고 사유 상세	n_report_reason	VARCHAR	500	N	N	NOT NULL				
6	신고 일시	n_report_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
7	신고 접수 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

## - 질문, 공지

테이블 정의서		물리 테이블명	논리 테이블명	생성일	승인자
단개	복개	member_question	member_question	2024-07-25	변진
작성자	업무명	그룹	작성일	페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	문의 고유 번호	n_question_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	회원 고유 번호	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	문의 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	문의 내용	n_question_body	VARCHAR	500	N	N	NOT NULL				
5	문의 일시	n_question_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	문의 접수 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

테이블 정의서		물리 테이블명	논리 테이블명	생성일	승인자
단개	복개	announce	announce	2024-07-25	변진
작성자	업무명	그룹	작성일	페이지수	1

순번	컬럼명	컬럼 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 컬럼 ID	비고
1	공지자 게시글 고유 번호	announce_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	게시글 작성 관리자 고유 번호	n_admin_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	공지자 게시글 유형	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	게시글 제목	announce_title	VARCHAR	50	N	N	NOT NULL				
5	게시글 본문	announce_body	TEXT		N	N	NOT NULL				
6	등록 일시	announce_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
7	게시글 조회 수	announce_view_count	INT		N	N	NOT NULL				

- point

데이터 정의서		원리 데이터명	member_point	논리 데이터명	리플트 데이터	승인자	
구분	구분	작성자	김영	작성일	2024-07-23	버전	1
구분	구분	업무명				페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조 테이블	참조 필드 ID	비고
1	포인트 번호	n_point_no	VARCHAR	10	Y	N	NOT NULL				
2	포인트 종류	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	포인트 단위	n_point	INT		N	N	NOT NULL	0			

데이터 정의서		원리 데이터명	point_history	논리 데이터명	포인트 내역	승인자	
구분	구분	작성자	김영	작성일	2024-07-23	버전	1
구분	구분	업무명				페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조 테이블	참조 필드 ID	비고
1	포인트 내역 번호	point_history_no	VARCHAR	10	Y	N	NOT NULL				
2	포인트 번호	n_point_no	VARCHAR	10	N	Y	NOT NULL		member_point	n_point_no	
3	종류 코드	type_code	VARCHAR	20	N	Y	NOT NULL		type_code	type_code	
4	포인트(변동전)	point_history_before	INT		N	N	NOT NULL				
5	포인트(변동후)	point_history_change	INT		N	N	NOT NULL				
6	포인트(변동후)	point_history_after	INT		N	N	NOT NULL				
7	등록 일시	point_history_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
8	처리 날짜	status_code	VARCHAR	20	N	Y	NOT NULL	3030102	status_code	status_code	

데이터 정의서		원리 데이터명	point_payment	논리 데이터명	포인트 승인 내역	승인자	
구분	구분	작성자	김영	작성일	2024-07-23	버전	1
구분	구분	업무명				페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조 테이블	참조 필드 ID	비고
1	포인트 승인 번호	pay_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	포인트 내역 번호	point_history_no	VARCHAR	10	N	Y	NOT NULL		point_history	point_history_no	
3	결제 방법	pay_type	VARCHAR	20	N	N	NOT NULL				
4	결제명도 타입	pay_tv_type	VARCHAR	50	N	N	NOT NULL				
5	결제 tv id	pay_tv_id	VARCHAR	50	N	N	NOT NULL				
6	결제 id	pay_id	VARCHAR	50	N	N	NOT NULL				
7	결제 코드	pay_code	VARCHAR	20	N	N	NULL	NULL			
8	결제 메시지	pay_message	VARCHAR	50	N	N	NULL	NULL			
9	등록 일시	pay_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
10	처리 날짜	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

데이터 정의서		원리 데이터명	point_widraw	논리 데이터명	포인트 인출 내역	승인자	
구분	구분	작성자	김영	작성일	2024-07-23	버전	1
구분	구분	업무명				페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조 테이블	참조 필드 ID	비고
1	포인트 인출 번호	widraw_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	포인트 내역 번호	point_history_no	VARCHAR	10	N	Y	NOT NULL		point_history	point_history_no	
3	지급 은행	widraw_bank	VARCHAR	10	N	N	NOT NULL				
4	지급 계좌	widraw_account	VARCHAR	20	N	N	NOT NULL				
5	이름부	widraw_name	VARCHAR	20	N	N	NOT NULL				
6	지급 금액	widraw_amount	INT		N	N	NOT NULL				
7	등록 일시	widraw_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
8	처리 날짜	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

- board

데이터 설명서		관리 테이블명	board	관리 테이블명	게시물	승인자					
		작성자	김민	작성일	2024-07-29	버전	1				
단계	상위	업무명				필수지수	1				
순번	필요명	필요 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필명 ID	비고
1	게시물 고유 번호	b_no	VARCHAR	10	Y	N	NOT NULL				
2	게시물 작성자	n_no	VARCHAR	10	N	Y	NOT NULL		member	n_no	
3	게시물 제목	b_title	VARCHAR	50	N	N	NOT NULL				
4	게시물 내용	b_body	VARCHAR	500	N	N	NOT NULL				
5	등록 일시	b_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	로그 기간	b_period	TIMESTAMP		N	N	NOT NULL				
7	게시물 조회수	b_view_count	INT		N	N	NOT NULL	0			
8	게시물 열 수	b_like_count	INT		N	N	NOT NULL	0			
9	게시물 신고 횟수	b_report_count	INT		N	N	NOT NULL	0			
10	게시물 상태	status_code	VARCHAR	20	N	Y	NOT NULL	S040101	status_code	status_code	

- location

데이터 설명서		관리 테이블명	location	관리 테이블명	위치	승인자					
		작성자	김민	작성일	2024-07-29	버전	1				
단계	상위	업무명				필수지수	1				
순번	필요명	필요 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필명 ID	비고
1	위치 고유 번호	location_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	게시물 고유 번호	b_no	VARCHAR	10	N	Y	NOT NULL		board	b_no	
3	위치(도)	location_dong	VARCHAR	10	N	N	NOT NULL				
4	위치(시/군)	location_si	VARCHAR	10	N	N	NOT NULL				
5	위치(구)	location_gu	VARCHAR	10	N	N	NOT NULL				
6	위치(동)	location_dong	VARCHAR	10	N	N	NOT NULL				
7	위치 주소	location_detail	VARCHAR	50	N	N	NULL	NULL			
8	위치(X좌표)	location_x	DOUBLE		N	N	NOT NULL	0			
9	위치(Y좌표)	location_y	DOUBLE		N	N	NOT NULL	0			

- file

데이터 설명서		관리 테이블명	file	관리 테이블명	파일	승인자					
		작성자	김민	작성일	2024-07-29	버전	1				
단계	상위	업무명				필수지수	1				
순번	필요명	필요 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필명 ID	비고
1	파일 고유 번호	file_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파일 이름 이름	file_origin_name	VARCHAR	100	N	N	NOT NULL				
3	파일 신규 이름	file_new_name	VARCHAR	100	N	N	NOT NULL				
4	파일 경로	file_path	VARCHAR	500	N	N	NOT NULL				
5	파일 사이즈	file_size	INT		N	N	NOT NULL				
6	파일 등록 일자	file_date	TIMESTAMP		N	N	NOT NULL				

데이터 설명서		관리 테이블명	file_relation	관리 테이블명	파일 관계	승인자					
		작성자	김민	작성일	2024-07-29	버전	1				
단계	상위	업무명				필수지수	1				
순번	필요명	필요 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필명 ID	비고
1	파일 관계 고유 번호	b_image_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파일 고유 번호	b_no	INT		N	Y	NOT NULL		file	file_no	
3	파일 사본처	b_image_source	VARCHAR	10	N	N	NOT NULL				



- 댓글, 찜

테이블 명		테이블 명		테이블 명		테이블 명		테이블 명	
		테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명

순번	필드명	필드 ID	타입	길이	PK 여부	FK 여부	NULL 여부	기본값	참조 테이블	참조 필드 ID	비고
1	댓글 고유 번호	reply_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	게시물 고유 번호	b_no	VARCHAR	10	N	Y	NOT NULL		board	b_no	
3	게시자 고유 번호	m_no	VARCHAR	10	N	Y	NOT NULL		member	m_no	
4	댓글 내용	reply_body	VARCHAR	200	N	N	NOT NULL				
5	등록 일시	reply_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	신고 횟수	reply_report_count	INT		N	N	NOT NULL				
7	댓글 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

테이블 명		테이블 명		테이블 명		테이블 명		테이블 명	
		테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명

순번	필드명	필드 ID	타입	길이	PK 여부	FK 여부	NULL 여부	기본값	참조 테이블	참조 필드 ID	비고
1	찜 고유 번호	wish_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	게시물 고유 번호	b_no	VARCHAR	10	N	Y	NOT NULL		board	b_no	
3	회원 번호	m_no	VARCHAR	10	N	Y	NOT NULL		member	m_no	
4	찜 일시	wish_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

- category

테이블 명		테이블 명		테이블 명		테이블 명		테이블 명	
		테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명

순번	필드명	필드 ID	타입	길이	PK 여부	FK 여부	NULL 여부	기본값	참조 테이블	참조 필드 ID	비고
1	카테고리 코드	c_code	VARCHAR	20	Y	N	NOT NULL				
2	그룹1	c_group1	VARCHAR	10	N	N	NOT NULL				
3	그룹2	c_group2	VARCHAR	10	N	N	NOT NULL				
4	그룹3	c_group3	VARCHAR	10	N	N	NOT NULL				

- party

테이블 명		테이블 명		테이블 명		테이블 명		테이블 명	
		테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명

순번	필드명	필드 ID	타입	길이	PK 여부	FK 여부	NULL 여부	기본값	참조 테이블	참조 필드 ID	비고
1	파티 고유 번호	p_no	VARCHAR	10	Y	N	NOT NULL				
2	게시물 고유 번호	b_no	VARCHAR	10	N	Y	NOT NULL		board	b_no	
3	카테고리	c_code	VARCHAR	10	N	Y	NOT NULL		category	c_code	
4	상품 URL	p_url	VARCHAR	500	N	N	NOT NULL				
5	상품 수량	p_amount	INT		N	N	NOT NULL				
6	구매 수량	p_amount_remain	INT		N	N	NOT NULL				
7	상품 가격	p_price	INT		N	N	NOT NULL				
8	남은 금액	p_price_remain	INT		N	N	NOT NULL				
9	파티 시작 일시	p_start_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
10	파티 종료 일시	p_end_date	TIMESTAMP		N	N	NULL				
11	파티 상태	status_code	VARCHAR	20	N	Y	NOT NULL	000101	status_code	status_code	

테이블 명		테이블 명		테이블 명		테이블 명		테이블 명	
		테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명
테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명	테이블 명

순번	필드명	필드 ID	타입	길이	PK 여부	FK 여부	NULL 여부	기본값	참조 테이블	참조 필드 ID	비고
1	파티 신청 상품 고유 번호	p_request_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파티 고유 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	신청자 번호	m_no	VARCHAR	10	N	Y	NOT NULL		member	m_no	
4	구매 희망 수량	request_amount	INT		N	N	NOT NULL				
5	파티 신청 일시	request_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
6	신청 상태	status_code	VARCHAR	20	N	Y	NOT NULL		status_code	status_code	

데이터 정의서		플린 테이블명	party_member	논리 테이블명	파트너 정보	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	파트너 정보 그룹 번호	p_member_no	VARCHAR	10	Y	N	NOT NULL				
2	파트너 그룹 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	파트너 정보 그룹 번호	m_no	VARCHAR	10	N	Y	NOT NULL		member	m_no	
4	구매 금액 수합	on_amount	INT		N	N	NOT NULL				
5	결제 예정 금액	on_price	INT		N	N	NOT NULL				
6	파트너 가입 일시	member_join_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			
7	파트너명 / 파트너	member_ni_leader	VARCHAR	10	N	N	NOT NULL	파트너			파트너명, 파트너 여부별 표기

데이터 정의서		플린 테이블명	party_purchase	논리 테이블명	파트너 결제 기록	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	결제 번호 그룹 번호	purchase_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파트너 그룹 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	파트너 정보 그룹 번호	p_member_no	VARCHAR	10	N	Y	NOT NULL		party_member	p_member_no	
4	결제 예정 금액	purchase_price	INT		N	N	NOT NULL				
5	결제 날짜	status_code	VARCHAR	20	N	Y	NOT NULL	300001	status_code	status_code	

데이터 정의서		플린 테이블명	party_purchase_detail	논리 테이블명	파트너 결제 상세 기록	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	결제 상세 번호 그룹 번호	purchase_detail_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	결제 번호	purchase_no	INT		N	Y	NOT NULL		party_purchase	purchase_no	
3	포인트 적립 내역 그룹 번호	point_history_no	VARCHAR	10	N	Y	NOT NULL		point_history	point_history_no	
4	결제 예정 금액	purchase_price	INT		N	N	NOT NULL				
5	결제 일시	purchase_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			

데이터 정의서		플린 테이블명	party_shipping	논리 테이블명	상품 배송 기록	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	상품 배송 번호 그룹 번호	shipping_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파트너 그룹 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	회사사명	invoice_courier	VARCHAR	10	N	N	NULL				
4	상품 배송일 번호	invoice_code	VARCHAR	20	N	N	NULL				
5	등록 일시	add_date	TIMESTAMP		N	N	NULL				
6	배송 날짜	status_code	VARCHAR	20	N	Y	NOT NULL	300001	status_code	status_code	

데이터 정의서		플린 테이블명	party_reception	논리 테이블명	상품 수취 기록	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	상품 수취 번호 그룹 번호	r_no	VARCHAR	10	Y	N	NOT NULL				
2	파트너 그룹 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	파트너 정보 그룹 번호	p_member_no	VARCHAR	10	N	Y	NOT NULL		party_member	p_member_no	
4	파트너 코멘트	reception_comment	VARCHAR	200	N	N	NULL				
5	수취 확인 일시	reception_date	TIMESTAMP		N	N	NULL				
6	수취 날짜	status_code	VARCHAR	20	N	Y	NOT NULL	300001	status_code	status_code	

데이터 정의서		플린 테이블명	party_settlement	논리 테이블명	파트너 회산 기록	승인자	
단계	속성	주석	그룹	작성일	2024-07-23	버전	1
단계	속성	영역명		필수성		페이지수	1

순번	필드명	필드 ID	타입	길이	PK여부	FK여부	NULL여부	기본값	참조테이블	참조 필드 ID	비고
1	파트너 회산 그룹 번호	s_settle_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	파트너 그룹 번호	p_no	VARCHAR	10	N	Y	NOT NULL		party	p_no	
3	파트너 정보 그룹 번호	p_member_no	VARCHAR	10	N	Y	NOT NULL		party_member	p_member_no	
4	회산 예정 금액	s_settle_price	INT		N	N	NOT NULL				
5	회산 날짜	status_code	VARCHAR	20	N	Y	NOT NULL	300001	status_code	status_code	

데이터 영역서		원본 데이터명	party_settlement_detail	논리 데이터명	관리 대상 상세 명칭	승인자	
		작성자	오영	작성일	2024-07-26	버전	1
간격	주기	빈도명				원어지수	1

순번	필명명	필명 ID	유형	길이	PK여부	FK여부	NULL여부	기본값	참조 테이블	참조 필명 ID	비고
1	정산 상세 포함 고유 번호	settle_detail_no	INT		Y	N	NOT NULL	AUTO_INCREMENT			
2	관리 정산 고유 번호	p_settle_no	INT		N	Y	NOT NULL		party_settlement	p_settle_no	
3	포인트 포함 내역 고유 번호	point_history_no	VARCHAR	10	N	Y	NOT NULL		point_history	point_history_no	
4	정산 환율율리	p_settle_price	INT		N	N	NOT NULL				
5	정산 일시	p_settle_date	TIMESTAMP		N	N	NOT NULL	CURRENT_TIMESTAMP			



### 4.3 네이밍 규칙

■ 패키지 구조																			
▶ Level 0 : 배포할 도메인 명을 뒤에서 부터 명칭 ( ex: 도메인명 com.gongzone.central)																			
▶ Level 1 : 기능 구분 ex: 회원 관련 기능(member), 게시판 관련 기능(board)																			
▶ Level 2 : Web Application 구성 요소르 구분 (ex: controller, service, mapper, domain, xml)																			
▶ Level 3 : 실구형체 이름 + 접미사: 상위 패키지 네이밍 (파스칼 표기법) (ex: MemberController )																			
	<table border="1"> <thead> <tr> <th>Level 0</th> <th>Level 1</th> <th>Level 2</th> <th>Level 3</th> </tr> </thead> <tbody> <tr> <td rowspan="5">com.gongzone.central</td> <td rowspan="5">member</td> <td>controller</td> <td>MemberController.java</td> </tr> <tr> <td>service</td> <td>MemberService.java</td> </tr> <tr> <td>domain</td> <td>Member.java</td> </tr> <tr> <td>mapper</td> <td>MemberMapper.java</td> </tr> <tr> <td>xml</td> <td>MemberMapper.xml</td> </tr> </tbody> </table>	Level 0	Level 1	Level 2	Level 3	com.gongzone.central	member	controller	MemberController.java	service	MemberService.java	domain	Member.java	mapper	MemberMapper.java	xml	MemberMapper.xml		
Level 0	Level 1	Level 2	Level 3																
com.gongzone.central	member	controller	MemberController.java																
		service	MemberService.java																
		domain	Member.java																
		mapper	MemberMapper.java																
		xml	MemberMapper.xml																
■ 명명규칙																			
▶ 공통	모든 이름은 a-z, A-Z, 0-9의 글자로 조합한다.																		
	프로그램 내에서 사용하는 모든 이름은 64자를 넘지 않는 것으로 한다.																		
	정확한 의미 전달을 위해 약어의 사용은 피한다.																		
	특수기호는 사용하지 않는 것을 원칙으로 한다.																		
	약어 사용이 의미 전달에 도움이 되는 경우에는 표준용어사전에 포함되어있는 약속된 약어를 사용할 수 있으나, 그렇지 않은 경우에는 단어의 spelling을 모두 적는 것으로 한다.																		
▶ 파일명	<table border="1"> <tbody> <tr> <td>Java</td> <td>대소문자를 사용하며 단어의 시작부분은 대문자로 표기한다. (파스칼 표기법)</td> </tr> <tr> <td></td> <td>단어는 표준용어사전을 따른다.</td> </tr> <tr> <td></td> <td>명사 또는 동사 + 명사의 표현된 형태로 이름을 구성한다.</td> </tr> <tr> <td></td> <td>첫 번째 문자는 항상 대문자를 사용한다.</td> </tr> <tr> <td></td> <td>클래스 유형에 따라 다음 표를 참고하여 구성한다.</td> </tr> <tr> <td>mapper</td> <td>Mapper (java파일)의 명명 규칙을 따른다. (ex: MemberMapper)</td> </tr> <tr> <td></td> <td>파일의 확장자는 .xml로 한다.</td> </tr> <tr> <td>jsx</td> <td>대소문자를 사용하며 단어의 시작부분은 소문자로 표기하고 구분자는 카멜표기법을 사용한다.</td> </tr> <tr> <td></td> <td>파일의 확장자는 .jsx로 한다.</td> </tr> </tbody> </table>	Java	대소문자를 사용하며 단어의 시작부분은 대문자로 표기한다. (파스칼 표기법)		단어는 표준용어사전을 따른다.		명사 또는 동사 + 명사의 표현된 형태로 이름을 구성한다.		첫 번째 문자는 항상 대문자를 사용한다.		클래스 유형에 따라 다음 표를 참고하여 구성한다.	mapper	Mapper (java파일)의 명명 규칙을 따른다. (ex: MemberMapper)		파일의 확장자는 .xml로 한다.	jsx	대소문자를 사용하며 단어의 시작부분은 소문자로 표기하고 구분자는 카멜표기법을 사용한다.		파일의 확장자는 .jsx로 한다.
Java	대소문자를 사용하며 단어의 시작부분은 대문자로 표기한다. (파스칼 표기법)																		
	단어는 표준용어사전을 따른다.																		
	명사 또는 동사 + 명사의 표현된 형태로 이름을 구성한다.																		
	첫 번째 문자는 항상 대문자를 사용한다.																		
	클래스 유형에 따라 다음 표를 참고하여 구성한다.																		
mapper	Mapper (java파일)의 명명 규칙을 따른다. (ex: MemberMapper)																		
	파일의 확장자는 .xml로 한다.																		
jsx	대소문자를 사용하며 단어의 시작부분은 소문자로 표기하고 구분자는 카멜표기법을 사용한다.																		
	파일의 확장자는 .jsx로 한다.																		

css	대소문자를 사용하여 단어의 시작부분은 소문자로 표기하고 구분자는 카멜표기법 사용한다.		
	파일의 확장자는 .css로 한다.		
▶ method	동사 표는 동사 + 명사로 작성하며, 혼합된 형태로 이름을 구성할 때에는 각 단어의 첫 글자는 대문자로 작성한다. (카멜표기법)		
	상황에 따라 다음 표를 참고하여 일반적인 Prefix를 적용하며, 나머지는 필요에 따른 적절한 이름을 사용한다.		
	기능유형	대표용어	기능유형
	get	조회	search
	add	등록	call
	update	수정	send
	delete	삭제	receive
	confirm	확인	transfer
	refuse	취소	check
	request	요청	is
			유무(여부)확인
▶ 변수명	대소문자를 사용하여 변수명의 시작부분을 소문자로 표기한다.		
	명사로 작성하며 각 단어의 첫 글자에는 대문자로 표기한다.		
	변수명에는 기호나 특수문자를 사용하지 않는다.		
	변수명에 의미 있는 단어를 사용하도록 한다.(표준 용어사전 참고)		
	변수에 해당하는 컬럼이 테이블에 존재하는 경우에는 컬럼명 기준으로 camel 표기법을 준수하여 사용한다		
	변수명은 중복되는 이름이 나오지 않도록 하여 임팩트성을 높여 사용하는 경우가 없도록 한다.		
	상수명은 해당 코드를 제외한 변수명 전체를 대문자로 하고, 단어 연결에는 _를 사용한다		
	예시		
java	변수	String memberNo / Date endDate / int shippingNo	
	상수	static final int WIDTH = 4; / static final long DEFAULT_EXPIRES_TIME = TimeUnit.MINUTES.toSeconds(30);	
	사용자 정의 타입	private MemberService memberService;	
JavaScript	변수	const handleMainPage / const memberNo / const currentMemberNo	
	상수	const WIDTH = 4; / const DEFAULT_PER_PAGE = 10;	

## 4.4 패키지 구조

no	com.gongzone.central. 패키지명	클래스 또는 인터페이스 명	메소드 명	설명
<b>announce</b>				
1	com.gongzone.central.announce.controller	AnnounceController	getTypeCode	공지별 타입코드 목록
2	com.gongzone.central.announce.controller	AnnounceController	findAllAnnounce	전체 공지 별 제목, 날짜, 조회수, 공지 타입 조회
3	com.gongzone.central.announce.controller	AnnounceController	incrementViews	공지사항 조회 수 증가
4	com.gongzone.central.announce.controller	AnnounceController	findAnnounceDetail	공지사항 상세 조회
5	com.gongzone.central.announce.controller	AnnounceController	createAnnounce	공지사항 작성
6	com.gongzone.central.announce.controller	AnnounceController	updateAnnounce	공지사항 수정 (공지 타입, 제목, 본문)
7	com.gongzone.central.announce.controller	AnnounceController	deleteAnnounce	공지사항 삭제
8	com.gongzone.central.announce.domain	Announce		공지사항 DTO
9	com.gongzone.central.announce.mapper	AnnounceMapper	createAnnounce	공지사항 삽입 Mapper
10	com.gongzone.central.announce.mapper	AnnounceMapper	findAllAnnounce	전체 공지 조회 Mapper
11	com.gongzone.central.announce.mapper	AnnounceMapper	findAnnounceByType	타입별 공지 Mapper
12	com.gongzone.central.announce.mapper	AnnounceMapper	countAllAnnounce	전체 공지 수량 Mapper
13	com.gongzone.central.announce.mapper	AnnounceMapper	countAnnounceByType	타입별 개수 Mapper
14	com.gongzone.central.announce.mapper	AnnounceMapper	findAnnounceDetail	공지사항 상세 Mapper
15	com.gongzone.central.announce.mapper	AnnounceMapper	incrementViews	공지사항 조회 수 증가 Mapper
16	com.gongzone.central.announce.mapper	AnnounceMapper	updateAnnounce	공지사항 수정 Mapper
17	com.gongzone.central.announce.mapper	AnnounceMapper	deleteAnnounce	공지사항 삭제 Mapper
18	com.gongzone.central.announce.service	AnnounceServiceImpl	findAllAnnounce	전체 공지 별 제목, 날짜, 조회수, 공지 타입 조회
19	com.gongzone.central.announce.service	AnnounceServiceImpl	findAnnounceByType	타입별 공지사항 조회
20	com.gongzone.central.announce.service	AnnounceServiceImpl	findAnnounceDetail	공지사항 상세 조회
21	com.gongzone.central.announce.service	AnnounceServiceImpl	countAllAnnounce	전체 공지사항 수 개수
22	com.gongzone.central.announce.service	AnnounceServiceImpl	incrementViews	공지사항 조회 수 증가
23	com.gongzone.central.announce.service	AnnounceServiceImpl	createAnnounce	공지사항 작성
24	com.gongzone.central.announce.service	AnnounceServiceImpl	updateAnnounce	공지사항 수정
25	com.gongzone.central.announce.service	AnnounceServiceImpl	deleteAnnounce	공지사항 삭제
26	com.gongzone.central.announce.service	AnnounceServiceImpl	countAnnounceByType	타입 별 공지사항 수 개수
<b>board</b>				
<b>admin</b>				
27	com.gongzone.central.board.admin.controller	BoardAdminController	getBoardAdminInfo	관리자 페이지 게시물 로드 컨트롤러
28	com.gongzone.central.board.admin.domain	BoardAdminInfo		BoardProgress, BoardWriteData, BoardWriteMember 개체로 DTO
29	com.gongzone.central.board.admin.domain	BoardProgress		전체 상태 별 관리 권한 게시물 로드 DTO
30	com.gongzone.central.board.admin.domain	BoardWriteData		게시물 조회 수 및 작성 날짜 DTO
31	com.gongzone.central.board.admin.domain	BoardWriteMember		게시물 작성자 및 참가 인원 DTO
32	com.gongzone.central.board.admin.mapper	BoardAdminMapper	getBoardProgress	전체 상태 별 관리 권한 게시물 로드 Mapper
33	com.gongzone.central.board.admin.mapper	BoardAdminMapper	getBoardWriteMember	게시물 조회 수 및 작성 날짜 Mapper
34	com.gongzone.central.board.admin.mapper	BoardAdminMapper	getBoardWriteData	게시물 작성자 및 참가 인원 Mapper
35	com.gongzone.central.board.admin.service	BoardAdminServiceImpl	getBoardAdminInfo	BoardProgress, BoardWriteData, BoardWriteMember 로드 서비스
<b>member</b>				
36	com.gongzone.central.board.controller	BoardController	createBoard	게시물 상세 컨트롤러
37	com.gongzone.central.board.controller	BoardController	updateBoard	게시물 수정 컨트롤러
38	com.gongzone.central.board.controller	BoardController	getBoardInfo	게시물 정보 요청 컨트롤러
39	com.gongzone.central.board.controller	BoardController	setWatch	게시물 찜 요청 컨트롤러
40	com.gongzone.central.board.controller	BoardController	addView	조회수 증가 컨트롤러
41	com.gongzone.central.board.controller	BoardController	getBoardList	게시물 목록 요청 컨트롤러
42	com.gongzone.central.board.controller	BoardController	writeBoard	게시물 작성 컨트롤러
43	com.gongzone.central.board.controller	BoardController	getMyWatchList	게시물 찜 목록 요청 컨트롤러
44	com.gongzone.central.board.domain	Board		게시물 정보 DTO
45	com.gongzone.central.board.domain	BoardMyPage		자신의 찜한 게시물 정보 DTO
46	com.gongzone.central.board.domain	BoardReply		게시물 댓글 DTO
47	com.gongzone.central.board.domain	BoardResponse		게시물 상세 정보 DTO
48	com.gongzone.central.board.domain	BoardSearchRequest		게시물 검색 요청 DTO
49	com.gongzone.central.board.mapper	BoardMapper	updateBoardPeriodStatus	프로세스 진행 별 게시물 상태 로드 Mapper

49	com.gongzone.central.board.mapper	BoardMapper	updateBoardPeriodStatus	프로세스 진행 별 게시물 상태 후보 Mapper
50	com.gongzone.central.board.mapper	BoardMapper	deleteParty	파티 삭제 Mapper
51	com.gongzone.central.board.mapper	BoardMapper	deleteBoard	게시물 삭제 Mapper
52	com.gongzone.central.board.mapper	BoardMapper	countPartyMember	파티 참여 인원 카운팅 Mapper
53	com.gongzone.central.board.mapper	BoardMapper	getFileInfo	게시물 파일(사진) 정보 후보 Mapper
54	com.gongzone.central.board.mapper	BoardMapper	getPartyNo	파티 번호 별 파티 조회 Mapper
55	com.gongzone.central.board.mapper	BoardMapper	updatePartyMember	파티 참여 회원 추가 Mapper
56	com.gongzone.central.board.mapper	BoardMapper	updateParty	파티 정보 수정 Mapper
57	com.gongzone.central.board.mapper	BoardMapper	updateLocation	지역 정보 수정 Mapper
58	com.gongzone.central.board.mapper	BoardMapper	updateBoard	게시물 수정 Mapper
59	com.gongzone.central.board.mapper	BoardMapper	getBoardInfo	게시물 정보 후보 Mapper
60	com.gongzone.central.board.mapper	BoardMapper	deleteWish	원할 글 삭제 Mapper
61	com.gongzone.central.board.mapper	BoardMapper	insertWish	원할 글 등록 Mapper
62	com.gongzone.central.board.mapper	BoardMapper	updateViewCount	조회수 증가 Mapper
63	com.gongzone.central.board.mapper	BoardMapper	insertBoard	게시물 등록 Mapper
64	com.gongzone.central.board.mapper	BoardMapper	insertLocation	파티 장소 등록 Mapper
65	com.gongzone.central.board.mapper	BoardMapper	insertFileRelation	파일(사진) 사본지 등록 Mapper
66	com.gongzone.central.board.mapper	BoardMapper	insertParty	파티 등록 Mapper
67	com.gongzone.central.board.mapper	BoardMapper	insertPartyMember	파티원 등록 Mapper
68	com.gongzone.central.board.mapper	BoardMapper	getBoardList	게시물 목록 요청 Mapper
69	com.gongzone.central.board.mapper	BoardMapper	getBoardReplyList	게시물 별 댓글 목록 후보 Mapper
70	com.gongzone.central.board.mapper	BoardMapper	getBoardWish	원할 목록 요청 Mapper
71	com.gongzone.central.board.mapper	BoardMapper	wishListMyPage	내장부 원할 목록 요청 Mapper
72	com.gongzone.central.board.mapper	BoardMapper	fileParty/boardNo	게시물 별 파일(사진) 번호 요청 Mapper
73	com.gongzone.central.board.service	BoardServiceImpl	deleteBoard	게시물 삭제 서비스
74	com.gongzone.central.board.service	BoardServiceImpl	updateBoardImageUrl	게시물 이미지 수정 서비스
75	com.gongzone.central.board.service	BoardServiceImpl	updateBoard	게시물 정보 수정 서비스
76	com.gongzone.central.board.service	BoardServiceImpl	getBoardInfo	게시물 정보 요청 서비스
77	com.gongzone.central.board.service	BoardServiceImpl	deleteWish	게시물 원할 삭제 서비스
78	com.gongzone.central.board.service	BoardServiceImpl	updateViewCount	조회수 증가 서비스
79	com.gongzone.central.board.service	BoardServiceImpl	getBoardList	게시물 목록 요청 서비스
80	com.gongzone.central.board.service	BoardServiceImpl	getWishListMyPage	내장부 원할 목록 요청 서비스
81	com.gongzone.central.board.service	BoardServiceImpl	deleteParty/boardNo	게시물 별 파티 정보 요청 서비스
82	com.gongzone.central.board.service	BoardServiceImpl	setValue	게시물 값 세팅 서비스
83	com.gongzone.central.board.reply.controller	BoardReplyController	deleteReply	게시물 댓글 삭제 DTO
84	com.gongzone.central.board.reply.controller	BoardReplyController	updateReply	게시물 댓글 수정 DTO
85	com.gongzone.central.board.reply.controller	BoardReplyController	addReply	게시물 댓글 삽입 DTO
86	com.gongzone.central.board.reply.mapper	BoardReplyMapper	deleteReply	게시물 댓글 삭제 Mapper
87	com.gongzone.central.board.reply.mapper	BoardReplyMapper	updateReply	게시물 댓글 수정 Mapper
88	com.gongzone.central.board.reply.mapper	BoardReplyMapper	insertReply	게시물 댓글 삽입 Mapper
89	com.gongzone.central.board.reply.service	BoardReplyServiceImpl	deleteReply	게시물 댓글 삭제 서비스
90	com.gongzone.central.board.reply.service	BoardReplyServiceImpl	getViewReply	게시물 목록 조회 서비스
91	com.gongzone.central.board.reply.service	BoardReplyServiceImpl	updateReply	게시물 수정 서비스
92	com.gongzone.central.board.reply.service	BoardReplyServiceImpl	addReply	게시물 등록 서비스
<b>common</b>				
93	com.gongzone.central.common.constants	ConstantsString	toBing	프로세스 진행 별 일할 ENUM
94	com.gongzone.central.common.pagination	Pagination		페이지네이션 DTO
95	com.gongzone.central.common.Response	Result		응답 결과 DTO
<b>configs</b>				
96	com.gongzone.central.config	SecretConfig	secretKey	Bearer SecretKey 주입
97	com.gongzone.central.config	SwaggerConfig	customOpenAPI	Swagger API 등록용 예제용 (custom)
98	com.gongzone.central.config	SwaggerConfig	publicApi	Swagger API 등록용 예제용 (public)
99	com.gongzone.central.config	WebClientConfig	webClientBuilder	웹 클라이언트 빌더
100	com.gongzone.central.config	WebConfig	addCorsMappings	허용 (restful) API 요청 처리를 위한 설정

101	com.gongzone.central.config	WebConfig	addResourceHandlers	주소요청이 다른 리우파일 업로드
102	com.gongzone.central.config	WebConfig	getOFilePath	파일업로드 경로
<b>file</b>				
103	com.gongzone.central.file.domain	FileUpload		파일업로드 업로드 DTO
104	com.gongzone.central.file.mapper	FileMapper	updateFile	파일업로드 수정 Mapper
105	com.gongzone.central.file.mapper	FileMapper	addFile	파일업로드 저장 Mapper
106	com.gongzone.central.file.mapper	FileMapper	getBoardFileList	게시판별 파일 목록 Mapper
107	com.gongzone.central.file.util	FileUtil	parseFileInfo	파일업로드 파일 정보 처리
<b>location</b>				
108	com.gongzone.central.location.controller	LocationController	getLocation	위치정보 조회 컨트롤러
109	com.gongzone.central.location.domain	LocationSearch		게시물, 위치별 장소 및 마감일 DTO
110	com.gongzone.central.location.mapper	LocationMapper	getLocationSearch	게시물, 위치별 장소 및 마감일 요청 Mapper
111	com.gongzone.central.location.service	LocationServiceImpl	getLocationSearch	게시물, 위치별 장소 및 마감일 요청 서비스
<b>member</b>				
112	com.gongzone.central.member.controller	MemberController	register	회원 가입 요청 컨트롤러
113	com.gongzone.central.member.controller	MemberController	findMemberBy/c	아이디 별 회원 검색 요청 컨트롤러
114	com.gongzone.central.member.controller	MemberController	findMemberByEmail	이메일 별 회원 검색 요청 컨트롤러
115	com.gongzone.central.member.controller	MemberController	getInfo	회원 정보 조회 요청 컨트롤러
116	com.gongzone.central.member.domain	Member		회원 정보 DTO
117	com.gongzone.central.member.domain	MemberLevel	fromLevel	회원 권한 정보 DTO
118	com.gongzone.central.member.domain	Token		회원 정보 관련 DTO
119	com.gongzone.central.member.mapper	MemberMapper	insert	회원 가입 요청 Mapper
120	com.gongzone.central.member.mapper	MemberMapper	findAll	전체 회원 조회 Mapper
121	com.gongzone.central.member.mapper	MemberMapper	findBy/c	회원번호 별 회원 조회 Mapper
122	com.gongzone.central.member.mapper	MemberMapper	findBy/e	아이디 별 회원 조회 Mapper
123	com.gongzone.central.member.mapper	MemberMapper	findByEmail	이메일 별 회원 조회 Mapper
124	com.gongzone.central.member.mapper	MemberMapper	findByEmailFromToken	소셜 로그인 회원 조회 Mapper
125	com.gongzone.central.member.mapper	MemberMapper	findBy/status	회원 상태 별 회원 조회 Mapper
126	com.gongzone.central.member.mapper	MemberMapper	update	회원 정보 수정 Mapper
127	com.gongzone.central.member.mapper	MemberMapper	delete	회원 정보 삭제 Mapper
128	com.gongzone.central.member.mapper	MemberMapper	info	회원 정보 조회 Mapper
129	com.gongzone.central.member.mapper	TokenMapper	insert	회원 로그인 토큰 정보 생성 Mapper
130	com.gongzone.central.member.mapper	TokenMapper	findAll	로그인 전체 회원 조회 Mapper
131	com.gongzone.central.member.mapper	TokenMapper	findBy/member/c	회원 번호 별 회원 조회 Mapper
132	com.gongzone.central.member.mapper	TokenMapper	update	회원 로그인 토큰 정보 수정 Mapper
133	com.gongzone.central.member.mapper	TokenMapper	delete	회원 로그인 토큰 정보 삭제 Mapper
134	com.gongzone.central.member.mapper	TokenMapper	getlastToken/c	아이디로 조회된 토큰 정보 조회 Mapper
135	com.gongzone.central.member.mapper	MemberServiceImpl	registerMember	회원 가입 요청 서비스
136	com.gongzone.central.member.mapper	MemberServiceImpl	getallMembers	전체 회원 조회 요청 서비스
137	com.gongzone.central.member.mapper	MemberServiceImpl	getMemberBy/c	회원 번호 별 회원 조회 서비스
138	com.gongzone.central.member.mapper	MemberServiceImpl	getMemberBy/c	회원 아이디 별 회원 조회 서비스
139	com.gongzone.central.member.mapper	MemberServiceImpl	getMemberByEmail	회원 이메일 별 회원 조회 서비스
140	com.gongzone.central.member.mapper	MemberServiceImpl	getMemberBy/status	회원 상태 별 회원 조회 서비스
<b>alerSSE</b>				
141	com.gongzone.central.member.alerSSE.controller	AlerSSEController	getVoteBy/c	특정 번호 별 정보 조회 컨트롤러
142	com.gongzone.central.member.alerSSE.controller	AlerSSEController	AlerSSEList	알림 목록 조회 컨트롤러
143	com.gongzone.central.member.alerSSE.controller	AlerSSEController	insertAlerSSE	알림 생성 컨트롤러
144	com.gongzone.central.member.alerSSE.controller	AlerSSEController	updateReadTimeAlerSSE	읽은 시간 수정 컨트롤러
145	com.gongzone.central.member.alerSSE.controller	AlerSSEController	deleteAlerSSE	알림 삭제 요청 컨트롤러
146	com.gongzone.central.member.alerSSE.controller	AlerSSEController	getAlerListAndCount	알림 수가 포함된 컨트롤러
147	com.gongzone.central.member.alerSSE.domain	AlerFlow		회원별 알림 전송 DTO
148	com.gongzone.central.member.alerSSE.domain	AlerSSE		회원별 알림 상태 DTO
149	com.gongzone.central.member.alerSSE.mapper	AlerSSEMapper	getAlerSSEBy/c	알림 번호 별 내용 Mapper
150	com.gongzone.central.member.alerSSE.mapper	AlerSSEMapper	AlerSSEList	알림 목록 Mapper

151	com.gongzone.central.member.alert99E.mapper	Alert99EMapper	insertAlert99E	일일 삽입 Mapper
152	com.gongzone.central.member.alert99E.mapper	Alert99EMapper	updateReadTimeAlert99E	일일 읽은 시간 수정 Mapper
153	com.gongzone.central.member.alert99E.mapper	Alert99EMapper	updateDeleteAlert99E	일일 삭제 Mapper
154	com.gongzone.central.member.alert99E.mapper	Alert99EMapper	getAlertAllow	일일 허용 여부 요청 Mapper
155	com.gongzone.central.member.alert99E.mapper	Alert99EMapper	countNewAlerts	최종 별 일일 추가 카운팅 Mapper
156	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	getAlert99EByNo	일일 번호 별 내용 서비스
157	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	getAlert99EByMemberNo	최종 별 일일 요청 서비스
158	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	saveAlert99E	일일 저장 서비스
159	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	updateReadAlert99E	일일 시간 수정 서비스
160	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	updateDeleteAlert99E	일일 삭제 요청 서비스
161	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	sendAlert	일일 전송 서비스
162	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	shouldSendAlert	일일 수신 권한 별 조회 서비스
163	com.gongzone.central.member.alert99E.service	Alert99EServiceImpl	countNewAlerts	일일 추가 카운팅 서비스
<b>note</b>				
164	com.gongzone.central.member.note.controller	NoteController	getNoteByNo	특지 번호 별 정보 조회 컨트롤러
165	com.gongzone.central.member.note.controller	NoteController	noteList	특지 목록 조회 컨트롤러
166	com.gongzone.central.member.note.controller	NoteController	insertNote	특지 삽입 컨트롤러
167	com.gongzone.central.member.note.controller	NoteController	updateReadTimeNote	특지 읽은 시간 컨트롤러
168	com.gongzone.central.member.note.controller	NoteController	updateDeleteNote	특지 삭제 요청 컨트롤러
169	com.gongzone.central.member.note.controller	NoteController	updateDeleteNoteTarget	특지 전송 대상 삭제 컨트롤러
170	com.gongzone.central.member.note.controller	NoteController	countNewNotes	특지 추가 카운팅 컨트롤러
171	com.gongzone.central.member.note.domain	Note		특지 정보 DTO
172	com.gongzone.central.member.note.mapper	NoteMapper	getNoteByNo	특지 번호 별 정보 조회 Mapper
173	com.gongzone.central.member.note.mapper	NoteMapper	noteList	특지 목록 조회 Mapper
174	com.gongzone.central.member.note.mapper	NoteMapper	insertNote	특지 삽입 Mapper
175	com.gongzone.central.member.note.mapper	NoteMapper	updateReadTimeNote	특지 읽은 시간 Mapper
176	com.gongzone.central.member.note.mapper	NoteMapper	updateDeleteNote	특지 삭제 요청 Mapper
177	com.gongzone.central.member.note.mapper	NoteMapper	updateDeleteNoteTarget	특지 전송 대상 삭제 Mapper
178	com.gongzone.central.member.note.mapper	NoteMapper	countNewNotes	특지 추가 카운팅 Mapper
179	com.gongzone.central.member.note.service	NoteServiceImpl	getNoteByNo	특지 번호 별 정보 조회 서비스
180	com.gongzone.central.member.note.service	NoteServiceImpl	getNoteListByMemberNo	특지 목록 조회 서비스
181	com.gongzone.central.member.note.service	NoteServiceImpl	saveNote	특지 삽입 서비스
182	com.gongzone.central.member.note.service	NoteServiceImpl	updateReadTime	특지 읽은 시간 서비스
183	com.gongzone.central.member.note.service	NoteServiceImpl	updateDeleteNote	특지 삭제 요청 서비스
184	com.gongzone.central.member.note.service	NoteServiceImpl	updateDeleteNoteTarget	특지 전송 대상 삭제 서비스
185	com.gongzone.central.member.note.service	NoteServiceImpl	countNewNotes	특지 추가 카운팅 서비스
<b>login</b>				
186	com.gongzone.central.member.login.controller	LoginController	login	로그인 요청 컨트롤러
187	com.gongzone.central.member.login.controller	LoginController	refreshToken	리프레시 토큰 정보 요청 컨트롤러
188	com.gongzone.central.member.login.controller	LoginController	logout	로그아웃 요청 컨트롤러
189	com.gongzone.central.member.login.controller	LoginController	adminLoginStatisticalData	관리자 로그인 상태 요청 컨트롤러
190	com.gongzone.central.member.login.domain	LoginLog		로그인 로그 정보 DTO
191	com.gongzone.central.member.login.domain	LoginRequest		로그인 정보 요청 DTO
192	com.gongzone.central.member.login.domain	LoginResponse		로그인 정보 응답 DTO
193	com.gongzone.central.member.login.domain	LoginStatistical		로그인 통계를 의미해 정보 DTO
194	com.gongzone.central.member.login.domain	RefreshTokenRequest		리프레시 토큰 정보 요청 DTO
195	com.gongzone.central.member.login.domain	RefreshTokenResponse		리프레시 토큰 응답 요청 DTO
196	com.gongzone.central.member.login.mapper	LoginMapper	getMemberByNo	회원번호 별 아이디 정보 요청 Mapper
197	com.gongzone.central.member.login.mapper	LoginMapper	getMemberByEmail	회원번호 별 이메일 정보 요청 Mapper
198	com.gongzone.central.member.login.mapper	LoginMapper	getMemberByNo	회원번호 별 정보 요청 Mapper
199	com.gongzone.central.member.login.mapper	LoginMapper	loginLogInsert	로그인 로그 삽입 Mapper
200	com.gongzone.central.member.login.mapper	LoginMapper	loginLogUpdate	로그아웃 상태 수정 Mapper
201	com.gongzone.central.member.login.mapper	LoginMapper	loginFalseLogUpdate	로그인 실패 상태 수정 Mapper
202	com.gongzone.central.member.login.mapper	LoginMapper	loginNoByMemberNo	회원번호 별 로그인 번호 Mapper



203	com.gongzone.central.member.login.mapper	LoginMapper	loginInTimeDay	로그인 시간 Mapper
204	com.gongzone.central.member.login.mapper	LoginMapper	loginNoByUseAgent	로그인 번호 Mapper
205	com.gongzone.central.member.login.security	Http401UnauthorizedEntryForm	commence	Http 401 권한 없음 시 요청, 응답 security
206	com.gongzone.central.member.login.security	JwtRequestFilter	doFilterInternal	JWTUtil 값 보존이 가장
207	com.gongzone.central.member.login.security	JwtUtil	generateToken	토큰생성
208	com.gongzone.central.member.login.security	JwtUtil	extractClaim	클레임에서 값 추출
209	com.gongzone.central.member.login.security	JwtUtil	getAIClaimsByToken	토큰 클레임
210	com.gongzone.central.member.login.security	JwtUtil	extractMemberNo	클레임 아이디
211	com.gongzone.central.member.login.security	JwtUtil	extractEmail	클레임 이메일
212	com.gongzone.central.member.login.security	JwtUtil	extractMemberNo	클레임 멤버번호
213	com.gongzone.central.member.login.security	JwtUtil	extractPointNo	클레임 포인트 번호
214	com.gongzone.central.member.login.security	JwtUtil	validateToken	토큰 유효성 검사
215	com.gongzone.central.member.login.security	JwtUtil	validateToken	토큰 유효성 검사
216	com.gongzone.central.member.login.security	JwtUtil	generateRefreshToken	리프레시 토큰 생성
217	com.gongzone.central.member.login.security	JwtUtil	extractExpiration	토큰 만료 날짜
218	com.gongzone.central.member.login.security	JwtUtil	isTokenExpired	토큰 만료 확인
219	com.gongzone.central.member.login.security	SecurityConfig	securityFilterChain	보안 필터 체인 정의
220	com.gongzone.central.member.login.security	SecurityConfig	authenticationManager	인증 주입
221	com.gongzone.central.member.login.security	SecurityConfig	passwordEncoder	비밀번호 인코더
222	com.gongzone.central.member.login.service	CheckStatusCode	checkStatus	유휴 상태 문의 서비스 (유연, 제거, 탈퇴, 정산)
223	com.gongzone.central.member.login.service	LoginLogServiceImpl	logLoginAttempt	로그인 시도 로그 서비스
224	com.gongzone.central.member.login.service	LoginLogServiceImpl	logLogout	로그아웃 로그 생성 서비스
225	com.gongzone.central.member.login.service	LoginLogServiceImpl	logLoginFailure	로그인 실패 로그 생성 서비스
226	com.gongzone.central.member.login.service	LoginLogServiceImpl	getLoginInfoByMemberNo	유휴 번호별 로그인 상태 서비스
227	com.gongzone.central.member.login.service	LoginLogServiceImpl	getLoginBrowserByCode	로그인 별 브라우저 코드 목록 서비스
228	com.gongzone.central.member.login.service	LoginLogServiceImpl	getLoginStatisticsDate	로그인 날짜 서비스
229	com.gongzone.central.member.login.service	MemberDetailsService	loadUserByUsername	유휴별 이름 로드 서비스

**Management**

230	com.gongzone.central.member.Management.controller	ManagementController	listAll	전체 유휴 조회 컨트롤러
231	com.gongzone.central.member.Management.controller	ManagementController	quitListAll	탈퇴 유휴 조회 컨트롤러
232	com.gongzone.central.member.Management.controller	ManagementController	sleepListAll	휴면 유휴 조회 컨트롤러
233	com.gongzone.central.member.Management.controller	ManagementController	punishListAll	제거 유휴 조회 컨트롤러
234	com.gongzone.central.member.Management.controller	ManagementController	statusUpdate	유휴 상태 코드 조회 컨트롤러
235	com.gongzone.central.member.Management.controller	ManagementController	punishUpdate	유휴 제거 기간 수정 컨트롤러
236	com.gongzone.central.member.Management.controller	ManagementController	punishInsert	휴면 제거 컨트롤러
237	com.gongzone.central.member.Management.domain	ManagementRequest		유휴별 상태 DTO
238	com.gongzone.central.member.Management.domain	MemberPunish		제거 유휴 DTO
239	com.gongzone.central.member.Management.domain	MemberQuit		탈퇴 유휴 DTO
240	com.gongzone.central.member.Management.domain	MemberSleep		휴면 유휴 DTO
241	com.gongzone.central.member.Management.domain	ReasonsAdminMember		관리자 유입 상태 DTO
242	com.gongzone.central.member.Management.domain	ReasonsInsert		관리자 유입 DTO
243	com.gongzone.central.member.Management.mapper	ManagementMapper	findAll	전체 유휴 조회 Mapper
244	com.gongzone.central.member.Management.mapper	ManagementMapper	findQuitList	탈퇴 유휴 조회 Mapper
245	com.gongzone.central.member.Management.mapper	ManagementMapper	findSleepList	휴면 유휴 조회 Mapper
246	com.gongzone.central.member.Management.mapper	ManagementMapper	findPunishList	제거 유휴 조회 Mapper
247	com.gongzone.central.member.Management.mapper	ManagementMapper	updateStatus	유휴 상태 코드 조회 Mapper
248	com.gongzone.central.member.Management.mapper	ManagementMapper	updatePunish	유휴 제거 기간 수정 Mapper
249	com.gongzone.central.member.Management.mapper	ManagementMapper	insertPunish	휴면 제거 Mapper
250	com.gongzone.central.member.Management.service	ManagementServiceImpl	getAllMembers	전체 유휴 조회 서비스
251	com.gongzone.central.member.Management.service	ManagementServiceImpl	getQuitAllMembers	탈퇴 유휴 조회 서비스
252	com.gongzone.central.member.Management.service	ManagementServiceImpl	getSleepAllMembers	휴면 유휴 조회 서비스
253	com.gongzone.central.member.Management.service	ManagementServiceImpl	getPunishAllMembers	제거 유휴 조회 서비스
254	com.gongzone.central.member.Management.service	ManagementServiceImpl	getStatusUpdate	유휴 상태 코드 조회 서비스
255	com.gongzone.central.member.Management.service	ManagementServiceImpl	getPeriodUpdate	유휴 제거 기간 수정 서비스

256	com.gongzone.central.member.Management.service	ManagementServiceImpl	getPunishment	회원 제재 서비스
<b>question</b>				
257	com.gongzone.central.member.question.controller	QuestionController	listQuestionAll	전체 질문 조회 컨트롤러
258	com.gongzone.central.member.question.controller	QuestionController	statusUpdate	질문 처리 상태 조회 컨트롤러
259	com.gongzone.central.member.question.controller	QuestionController	questionMemberInsert	질문 등록 컨트롤러
260	com.gongzone.central.member.question.domain	QuestionMember		질문 DTO
261	com.gongzone.central.member.question.mapper	QuestionMapper	findQuestionAll	전체 질문 조회 Mapper
262	com.gongzone.central.member.question.mapper	QuestionMapper	updateQuestionStatus	질문 처리 상태 조회 Mapper
263	com.gongzone.central.member.question.mapper	QuestionMapper	insertQuestion	질문 등록 Mapper
264	com.gongzone.central.member.question.mapper	QuestionMapper	getMemberNoByQuestionNo	회원 번호 질문 검색 Mapper
265	com.gongzone.central.member.question.service	QuestionServiceImpl	getQuestionMembers	전체 질문 조회 서비스
266	com.gongzone.central.member.question.service	QuestionServiceImpl	getQuestionStatusUpdate	질문 처리 상태 조회 서비스
267	com.gongzone.central.member.question.service	QuestionServiceImpl	getQuestionMemberInsert	질문 등록 서비스
<b>report</b>				
268	com.gongzone.central.member.report.controller	ReportController	listReportAll	전체 report 조회 컨트롤러
269	com.gongzone.central.member.report.controller	ReportController	statusUpdate	report 수정 컨트롤러
270	com.gongzone.central.member.report.controller	ReportController	punishment	제재 요청 컨트롤러
271	com.gongzone.central.member.report.controller	ReportController	reportMemberInsert	report 회원 등록 컨트롤러
272	com.gongzone.central.member.report.controller	ReportController	adminReportStatisticalData	관리자 승인 컨트롤러
273	com.gongzone.central.member.report.domain	ReportMember		회원 회원 신고 DTO
274	com.gongzone.central.member.report.domain	ReportStatistical		신고 신고 조회 DTO
275	com.gongzone.central.member.report.domain	RequestReportMember		신고 요청 DTO
276	com.gongzone.central.member.report.mapper	ReportMapper	findReportAll	전체 report 조회 Mapper
277	com.gongzone.central.member.report.mapper	ReportMapper	updateReportStatus	report 수정 Mapper
278	com.gongzone.central.member.report.mapper	ReportMapper	updateStatus	제재 상태 수정 Mapper
279	com.gongzone.central.member.report.mapper	ReportMapper	insertReport	report 회원 등록 Mapper
280	com.gongzone.central.member.report.mapper	ReportMapper	reportTimeDay	관리자 승인 Mapper
281	com.gongzone.central.member.report.service	ReportServiceImpl	getAllReportMember	전체 report 조회 서비스
282	com.gongzone.central.member.report.service	ReportServiceImpl	getReportStatusUpdate	report 수정 서비스
283	com.gongzone.central.member.report.service	ReportServiceImpl	getStatuUpdate	제재 요청 서비스
284	com.gongzone.central.member.report.service	ReportServiceImpl	getReportMemberInsert	report 회원 등록 서비스
285	com.gongzone.central.member.report.service	ReportServiceImpl	getReportStatisticalData	관리자 승인 서비스
<b>socialLogin</b>				
286	com.gongzone.central.member.socialLogin.domain	SocialMember		소셜 회원 정보 DTO
287	com.gongzone.central.member.socialLogin.domain	SocialRequest		회원 별 소셜로그인 회원정보 DTO
<b>Google</b>				
288	com.gongzone.central.member.socialLogin.google.controller	GoogleController	googleToken	구글 회원 정보 처리 요청 컨트롤러
289	com.gongzone.central.member.socialLogin.google.service	GoogleService	googleToken	구글 회원 정보 처리 요청 서비스
290	com.gongzone.central.member.socialLogin.google.service	GoogleService	updateTokens	구글 회원 정보 갱신 요청 서비스
<b>Kakao</b>				
291	com.gongzone.central.member.socialLogin.kakao.controller	KakaoController	kakaoToken	카카오 회원 정보 처리 요청 컨트롤러
292	com.gongzone.central.member.socialLogin.kakao.service	KakaoService	kakaoToken	카카오 회원 정보 처리 요청 서비스
293	com.gongzone.central.member.socialLogin.kakao.service	KakaoService	updateTokens	카카오 회원 정보 갱신 요청 서비스
<b>naver</b>				
294	com.gongzone.central.member.socialLogin.naver.controller	NaverController	naverToken	네이버 회원 정보 처리 요청 컨트롤러
295	com.gongzone.central.member.socialLogin.naver.service	NaverService	naverToken	네이버 회원 정보 처리 요청 서비스
296	com.gongzone.central.member.socialLogin.naver.service	NaverService	saveMember	네이버 회원 정보 저장 서비스
297	com.gongzone.central.member.socialLogin.naver.service	NaverService	updateTokens	네이버 회원 정보 갱신 요청 서비스
<b>myinfo</b>				
298	com.gongzone.central.member.myinfo.controller	MyInfoController	getMemberInfo	회원 정보 요청 컨트롤러
299	com.gongzone.central.member.myinfo.controller	MyInfoController	updatePassword	회원 비밀번호 수정 컨트롤러
300	com.gongzone.central.member.myinfo.controller	MyInfoController	getNickname	회원 닉네임 요청 컨트롤러
301	com.gongzone.central.member.myinfo.controller	MyInfoController	updateNickname	회원 닉네임 수정 컨트롤러
302	com.gongzone.central.member.myinfo.controller	MyInfoController	getAddress	회원 주소 정보 요청 컨트롤러



303	com.gonggong.central.member.myinfo.controller	MyInfoController	updateMemberAddress	회원 주소 정보 수정 컨트롤러
304	com.gonggong.central.member.myinfo.controller	MyInfoController	getMemberPhone	회원 전화번호 정보 요청 컨트롤러
305	com.gonggong.central.member.myinfo.controller	MyInfoController	updateStatus	회원 상태 정보 수정 컨트롤러
306	com.gonggong.central.member.myinfo.domain	MyInfoMason		회원 정보 DTO
307	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	findByNo	회원 번호별 회원 조회 Mapper
308	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	findByNickname	회원 닉네임 조회 Mapper
309	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	findByAddress	회원 주소 조회 Mapper
310	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	findByPhone	회원 전화번호 조회 Mapper
311	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	updatePassword	회원 비밀번호 수정 Mapper
312	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	updateMemberNick	회원 닉네임 수정 Mapper
313	com.gonggong.central.member.myinfo.mapper	MyInfoMapper	updateMemberAddress	회원 주소 수정 Mapper
314	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	findByNo	회원 정보 요청 서비스
315	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	findByNickname	회원 닉네임 정보 요청 서비스
316	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	findByAddress	회원 주소 요청 서비스
317	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	findByPhone	회원 전화번호 조회 서비스
318	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	updatePassword	회원 비밀번호 수정 서비스
319	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	updateMemberNick	회원 닉네임 수정 서비스
320	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	updateMemberAddress	회원 주소 수정 서비스
321	com.gonggong.central.member.myinfo.service	MyInfoServiceImpl	updateStatusCode	회원 상태코드 변경 서비스
<b>alert</b>				
322	com.gonggong.central.member.myinfo.alert.controller	AlertController	getAlertSettings	회원 알림 설정 조회 컨트롤러
323	com.gonggong.central.member.myinfo.alert.controller	AlertController	updateAlertSettings	회원 알림 수정 컨트롤러
324	com.gonggong.central.member.myinfo.alert.controller	AlertController	insertAlertSettings	회원 알림 삽입 컨트롤러
325	com.gonggong.central.member.myinfo.alert.domain	MyAlert		알림 param DTO
326	com.gonggong.central.member.myinfo.alert.mapper	AlertMapper	findAlertByMemberNo	회원 알림 설정 조회 Mapper
327	com.gonggong.central.member.myinfo.alert.mapper	AlertMapper	updateAlertSettings	회원 알림 수정 Mapper
328	com.gonggong.central.member.myinfo.alert.mapper	AlertMapper	insertAlertSettings	회원 알림 삽입 Mapper
329	com.gonggong.central.member.myinfo.alert.service	AlertServiceImpl	getAlertByMemberNo	회원 알림 설정 조회 서비스
330	com.gonggong.central.member.myinfo.alert.service	AlertServiceImpl	updateAlertSettings	회원 알림 수정 서비스
331	com.gonggong.central.member.myinfo.alert.service	AlertServiceImpl	insertAlertSettings	회원 알림 삽입 서비스
332	com.gonggong.central.member.myinfo.alert.service	AlertServiceImpl	createDefaultAlert	회원 알림 기본 알림 설정 서비스
333	com.gonggong.central.member.myinfo.alert.service	AlertServiceImpl	setDefaultAlertValues	회원 알림 기본 알림 상태 값 설정 서비스
<b>dropdown</b>				
334	com.gonggong.central.member.myinfo.dropdown.controller	DropDownController	getDropDownData	그룹다운 정보 요청 컨트롤러
335	com.gonggong.central.member.myinfo.dropdown.domain	DropDownInfo		그룹다운 정보 DTO
336	com.gonggong.central.member.myinfo.dropdown.mapper	DropDownMapper	findByData	회원 별 정보 그룹 Mapper
337	com.gonggong.central.member.myinfo.dropdown.service	DropDownServiceImpl	findByData	회원 별 정보 그룹 서비스
<b>interaction</b>				
338	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	extractCurrentUserInfo	현재 접속중인 유저 정보 요청 서비스
339	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	getPublicMembers	공개회원 조회 컨트롤러
340	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	getFollowMembers	특정 회원 팔로우 멤버 요청 컨트롤러
341	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	getBlockMembers	특정 회원 차단 멤버 요청 컨트롤러
342	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	followMember	팔로우 추가 요청 컨트롤러
343	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	unfollowMember	팔로우 삭제 요청 컨트롤러
344	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	blockMember	차단 추가 요청 컨트롤러
345	com.gonggong.central.member.myinfo.interaction.controller	InteractionController	unblockMember	차단 해제 요청 컨트롤러
346	com.gonggong.central.member.myinfo.interaction.domain	InteractionMember		회원 상호 작용 DTO
347	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	findAllMembers	공개회원 조회 Mapper
348	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	getWarningCount	특정 회원 차단 수 가운팅 Mapper
349	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	getMemberFollowCount	특정 회원 팔로우 수 가운팅 Mapper
350	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	getTotalMembers	전체 회원 수 가운팅 Mapper
351	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	insertFollow	팔로우 추가 요청 Mapper
352	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	deleteFollow	팔로우 삭제 요청 Mapper
353	com.gonggong.central.member.myinfo.interaction.mapper	InteractionMapper	insertBlock	차단 추가 요청 Mapper

354	com.gongzone.central.member.myInfo.interaction.mapper	InteractionMapper	deleteBlock	차량 삭제 요청 Mapper
355	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	getTotalMembers	전체 회원 수 가용성 서비스
356	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	findAllMembers	전체회원 조회 서비스
357	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	followMember	팔로우 추가 요청 서비스
358	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	unFollowMember	팔로우 삭제 요청 서비스
359	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	blockMember	차량 추가 요청 서비스
360	com.gongzone.central.member.myInfo.interaction.service	InteractionServiceImpl	unBlockMember	차량 삭제 요청 서비스
<b>profile</b>				
361	com.gongzone.central.member.myInfo.profile.controller	ProfileController	addProfilePicture	프로필 사진 추가 컨트롤러
362	com.gongzone.central.member.myInfo.profile.controller	ProfileController	updateProfilePicture	프로필 사진 수정 컨트롤러
363	com.gongzone.central.member.myInfo.profile.controller	ProfileController	getProfile	프로필 카드 정보 조회 요청 컨트롤러
364	com.gongzone.central.member.myInfo.profile.controller	ProfileController	getAllProfiles	전체 회원 프로필 사진 조회 요청 컨트롤러
365	com.gongzone.central.member.myInfo.profile.controller	ProfileController	parseFile	파일 파싱 요청 컨트롤러
366	com.gongzone.central.member.myInfo.profile.domain	Profile		프로필 카드 정보 DTO
367	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	addFile	프로필 사진 정보 추가 Mapper
368	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	addFileRelation	프로필 사진 사용자 추가 Mapper
369	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	updateFileRelation	프로필 사진 사용자 수정 Mapper
370	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	eventsFileRelation	프로필 사진 사용 확인 요청 Mapper
371	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	getProfile	회원 회원 프로필 사진 조회 요청 Mapper
372	com.gongzone.central.member.myInfo.profile.mapper	ProfileMapper	getAllProfiles	전체 회원 프로필 사진 조회 요청 Mapper
373	com.gongzone.central.member.myInfo.profile.service	ProfileServiceImpl	getProfile	프로필 카드 정보 조회 서비스
374	com.gongzone.central.member.myInfo.profile.service	ProfileServiceImpl	addProfilePicture	프로필 사진 추가 서비스
375	com.gongzone.central.member.myInfo.profile.service	ProfileServiceImpl	updateProfilePicture	프로필 사진 수정 서비스
376	com.gongzone.central.member.myInfo.profile.service	ProfileServiceImpl	getAllProfiles	전체 회원 프로필 사진 조회 요청 서비스
377	com.gongzone.central.member.myInfo.profile.service	ProfileServiceImpl	parseFile	파일 파싱 요청 서비스
<b>party</b>				
<b>accept</b>				
	com.gongzone.central.party.accept.controller	AcceptController	getPartyList	파티 리스트 요청 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	getPartyDetail	파티 상세 정보 조회 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	getParticipants	특정 파티 전체 참여자 조회 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	getPurchaseInfo	구매 정보 조회 요청 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	getRequestMember	파티 참가자 조회 요청 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	updateStatus	파티 상태 처리 수정 컨트롤러
	com.gongzone.central.party.accept.controller	AcceptController	updatePartyAndBoardStatus	게시물 상태 변경 및 파티 정보 수정 컨트롤러
	com.gongzone.central.party.accept.domain	AcceptDetail		파티 구성 상세 정보 DTO
	com.gongzone.central.party.accept.domain	AcceptMember		파티 구성원 정보 DTO
	com.gongzone.central.party.accept.domain	PartyMemberPurchase		파티원 구매 정보 DTO
	com.gongzone.central.party.accept.domain	RequestMember		파티 신청 회원 정보 DTO
	com.gongzone.central.party.accept.domain	RequestParty		파티 신청 대상 파티 정보 DTO
	com.gongzone.central.party.accept.domain	RequestStatus		파티 신청 시 현황 및 회원 정보 DTO
	com.gongzone.central.party.accept.mapper	AcceptMapper	getPartyList	파티리스트 조회 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	getPartyDetail	파티 상세정보 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	getParticipants	파티 구성원 정보 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	getRequestMember	파티 참가자 조회 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	findPointByMemberNo	회원 번호별 포인트 조회 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	updatePartyStatus	파티 상태 수정 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	insertPartyMember	파티 구성원 추가 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	requestMemberByMemberNo	회원 번호별 회원 정보 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	updateAmountMember	파티 회원 구매 현황 수정 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	insertPartyMemberNo	회원 번호 신청 회원 번호 조회 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	findPartyMember	파티원별 인원 조회 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	updateAmountAttendick	수입 현황 수정 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	deletePartyRequest	파티 정보 삭제 요청 Mapper
	com.gongzone.central.party.accept.mapper	AcceptMapper	getPartyUnitPrice	파티 단위별 가격 요청 Mapper

com.gongzone.central.party.accept.mapper	AcceptMapper	requestJoin	모의 참여 요청 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	completeBoardStatus	게시물 마감 상태 변경 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	completePartyStatus	모의 마감 상태 변경 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	insertPartyPurchase	모의 구입 정보 추가 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	getPartyMemberPurchase	모의 구입 정보 조회 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	getPartyLeaderPartyNo	모의장 정보 조회 Mapper
com.gongzone.central.party.accept.mapper	AcceptMapper	updatePartyLeaderPurchase	모의장 구매 정보 수정 Mapper
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPartyList	모의 목록 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getParticipate	모의 구입 정보 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getRequestMember	모의 요청 회원 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPartyList	모의 목록 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPartyNo	모의 번호별 모의 목록 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPartyStatusPartyNo	모의 상태 정보 조회 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getRequestMemberByMemberNo	회원 번호 별 모의 가입 요청 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPartyDetailByPartyNo	모의 번호 별 모의 상세 정보 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	completeParty	모의 마감 서비스
com.gongzone.central.party.accept.service	AcceptServiceImpl	getPurchaseInfo	모의 구매 정보 서비스
<b>after</b>			
<b>admin</b>			
com.gongzone.central.party.after.controller.admin	AdminPartyAfterController	adminPostPartySettlement	특정 모의의 마감 작업을 요청 컨트롤러
com.gongzone.central.party.after.service.admin	AdminPartyAfterServiceImpl	settlement	partyNo에 해당하는 모의의 마감 작업 서비스
<b>member</b>			
com.gongzone.central.party.after.controller	PartyAfterController	postPartyPurchase	모의 참여자의 결제 정보 추가하는 컨트롤러
com.gongzone.central.party.after.controller	PartyAfterController	postPartyShipping	모의의 배송 정보를 입력하는 컨트롤러
com.gongzone.central.party.after.controller	PartyAfterController	postPartyShippingComplete	모의 배송 입력을 완료하는 컨트롤러
com.gongzone.central.party.after.controller	PartyAfterController	patchPartyReception	모의 신청 현황 컨트롤러
com.gongzone.central.party.after.controller	PartyAfterController	postTextParty	게스트문 발송 모의 발송 메시지
com.gongzone.central.party.after.domain	PartyPurchase		모의 참여자의 결제 정보 DTO
com.gongzone.central.party.after.domain	PartyPurchaseDetail		모의 참여자의 결제 세부 정보 DTO
com.gongzone.central.party.after.domain	Reception		모의의 신청 DTO
com.gongzone.central.party.after.domain	Settlement		모의의 마감 정보 DTO
com.gongzone.central.party.after.domain	SettlementDetail		모의의 마감 세부 정보 DTO
com.gongzone.central.party.after.domain	Shipping		모의의 배송 정보 DTO
com.gongzone.central.party.after.mapper	PartyAfterMapper	insertPurchaseDetail	모의 참여자의 결제 세부 정보를 데이터베이스에 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updatePurchaseComplete	특정 모의 신청을 완료 상태로 업데이트합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	checkPurchaseComplete	특정 모의의 신청 완료 여부를 확인합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	insertShipping	특정 모의에 대한 배송 정보를 데이터베이스에 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updateShipping	특정 배송 정보를 업데이트합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getPartyMembers	특정 모의에 참여한 모든 회원의 목록을 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getPartyMembersExcludeLeader	모의 회장을 제외한 특정 모의에 참여한 모든 회원의 목록을 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getLastOfReception	마지막으로 삽입된 신청의 번호를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	insertReception	특정 모의에 대한 신청 정보를 데이터베이스에 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updateReception	특정 신청 현황 정보를 업데이트합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	checkReceptionComplete	특정 모의의 신청 현황이 완료되었는지 확인합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	calculateSettlementPrice	특정 모의의 마감 금액을 계산합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	insertPartySettlement	특정 모의에 대한 마감 정보를 데이터베이스에 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updateShippingStatus	특정 모의의 배송 상태를 업데이트합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertParty	테스트용으로 모의를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertBoard	테스트용으로 게시판 항목을 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertLocation	테스트용으로 위치 정보를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertFile	테스트용으로 위치 정보를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertFileRelation	테스트용으로 파일과 게시글 간의 관계를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertPartyMember	테스트용으로 모의 멤버를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testGetLastOfBoard	테스트용으로 마지막으로 삽입된 게시글 번호를 가져옵니다.

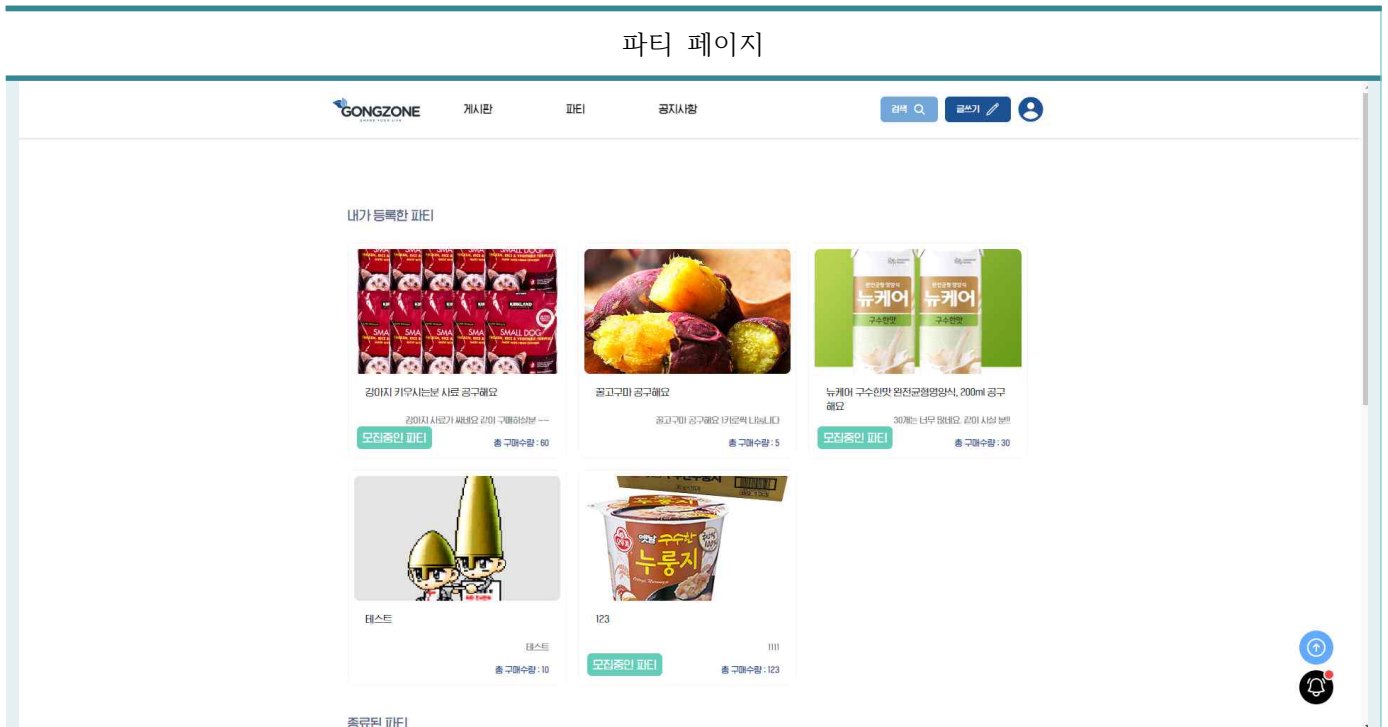
com.gongzone.central.party.after.mapper	PartyAfterMapper	testGetLastJoinParty	테스트용유료 마지막유료 가입된 파티의 인덱스를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testGetLastJoinPartyMember	테스트용유료 마지막유료 가입된 파티 멤버의 인덱스를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testChangeBoardStatus	테스트용유료 게시판 상태를 변경합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	testInsertPartyPurchase	테스트용유료 파티 결제 정보를 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	insertSettlementDetail	집안 세부 정보를 적외레이스에 삽입합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getSettlementPrice	특정 파티의 집안 금액을 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getLeasePartyMemberNo	특정 파티의 회원 번호를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getPartySettlementNo	특정 파티의 집안 번호를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updateSettlementState	특정 집안의 상태를 업데이트합니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	getMemberNoByPartyMemberNo	파티 멤버 번호를 이용하여 회원 번호를 가져옵니다.
com.gongzone.central.party.after.mapper	PartyAfterMapper	updatePartyComplete	특정 파티를 완료 상태로 업데이트합니다.
com.gongzone.central.party.after.service	PartyAfterServiceImpl	purchase	특정 파티의 특정 회원에 대한 결제 정보를 처리합니다.
com.gongzone.central.party.after.service	PartyAfterServiceImpl	updateShipping	특정 파티의 특정 배송 정보를 업데이트합니다.
com.gongzone.central.party.after.service	PartyAfterServiceImpl	updateShippingComplete	특정 파티의 특정 배송 완료 상태를 업데이트합니다.
com.gongzone.central.party.after.service	PartyAfterServiceImpl	updateReceptionComplete	특정 파티의 특정 수취 현황을 완료 상태로 업데이트합니다.
com.gongzone.central.party.after.service	PartyAfterServiceImpl	testInsertParty	테스트용유료 새로운 파티를 적외레이스에 삽입합니다.
<b>point</b>			
com.gongzone.central.point.controller	PointController	getMemberPoint	현재 포인트 보유량을 요청한다.
com.gongzone.central.point.controller	PointController	getMemberPointHistory	특정 회원의 포인트 내역 전체를 요청한다.
com.gongzone.central.point.controller	PointController	getMemberPointHistory	특정 회원의 특정 포인트 내역을 요청한다.
com.gongzone.central.point.service	PointHistoryService	insert	포인트 내역을 삽입한다.
com.gongzone.central.point.service	PointHistoryService	get	포인트 내역을 가져온다.
com.gongzone.central.point.service	PointHistoryService	getMany	특정 회원의 포인트 내역을 반환한다.
com.gongzone.central.point.service	PointHistoryService	updateSuccess	포인트 내역 업데이트 성공을 기록한다.
com.gongzone.central.point.service	PointService	getCurrentPoint	현재 포인트를 반환한다.
com.gongzone.central.point.service	PointService	getPoint	포인트 정보를 반환한다.
com.gongzone.central.point.service	PointService	updatePoint	포인트 정보를 업데이트한다.
com.gongzone.central.point.service	PointService	getMemberPointNo	회원 고유번호를 반환한다.
com.gongzone.central.point.service.impl	PointHistoryServiceImpl	insert	포인트 내역을 삽입한다.
com.gongzone.central.point.service.impl	PointHistoryServiceImpl	get	포인트 내역을 가져온다.
com.gongzone.central.point.service.impl	PointHistoryServiceImpl	getMany	특정 회원의 포인트 내역을 반환한다.
com.gongzone.central.point.service.impl	PointHistoryServiceImpl	updateSuccess	포인트 내역 업데이트 성공을 기록한다.
com.gongzone.central.point.service.impl	PointServiceImpl	getCurrentPoint	현재 포인트를 반환한다.
com.gongzone.central.point.service.impl	PointServiceImpl	getPoint	포인트 정보를 반환한다.
com.gongzone.central.point.service.impl	PointServiceImpl	getMemberPointNo	회원 고유번호를 반환한다.
com.gongzone.central.point.service.impl	PointServiceImpl	updatePoint	포인트 정보를 업데이트한다.
com.gongzone.central.point.mapper	PointHistoryMapper	insert	포인트 내역을 삽입한다.
com.gongzone.central.point.mapper	PointHistoryMapper	get	포인트 내역을 가져온다.
com.gongzone.central.point.mapper	PointHistoryMapper	getMany	특정 회원의 포인트 내역을 가져온다.
com.gongzone.central.point.mapper	PointHistoryMapper	updateHistorySuccess	포인트 내역 업데이트 성공을 기록한다.
com.gongzone.central.point.mapper	PointHistoryMapper	getLatestIndex	마지막 인덱스를 가져온다.
com.gongzone.central.point.mapper	PointHistoryMapper	getTotalCount	총 포인트 내역 개수를 가져온다.
com.gongzone.central.point.mapper	PointMapper	insert	포인트 정보를 삽입한다.
com.gongzone.central.point.mapper	PointMapper	getLatestIndex	마지막 인덱스를 가져온다.
com.gongzone.central.point.mapper	PointMapper	getPoint	포인트 정보를 가져온다.
com.gongzone.central.point.mapper	PointMapper	getCurrentPoint	현재 포인트를 가져온다.
com.gongzone.central.point.mapper	PointMapper	getMemberPointNo	회원 고유번호를 가져온다.
com.gongzone.central.point.mapper	PointMapper	update	포인트 정보를 업데이트한다.
<b>point</b>			
com.gongzone.central.point.payment.controller	PaymentController	postMemberPointCharge	포인트 충전을 요청한다.
com.gongzone.central.point.payment.controller	PaymentController	getPointPayment	특정 결제 정보를 요청한다.
com.gongzone.central.point.payment.controller	PaymentController	getPointPaymentDetail	특정 결제 현황의 상세 정보를 요청한다.
com.gongzone.central.point.payment.controller	PaymentController	getPointPayments	회원의 결제 정보를 요청한다.

com.gongzone.central.point.payment.service	PaymentService	insert	결제 정보를 삽입한다.
com.gongzone.central.point.payment.service	PaymentService	get	결제 정보를 반환한다.
com.gongzone.central.point.payment.service	PaymentService	getDetail	결제 상세 정보를 반환한다.
com.gongzone.central.point.payment.service	PaymentService	getMany	결제 정보를 여러 개 반환한다.
com.gongzone.central.point.payment.service	PaymentService	charge	회심 포인트를 충전한다.
com.gongzone.central.point.payment.api	PortOneClient	getPaymentInfo	결제 정보를 가져온다.
com.gongzone.central.point.payment.api	PortOneClient	getRequeue	결제 요청 정보를 가져온다.
com.gongzone.central.point.payment.service.impl	PaymentServiceImpl	insert	결제 정보를 삽입한다.
com.gongzone.central.point.payment.service.impl	PaymentServiceImpl	get	결제 정보를 반환한다.
com.gongzone.central.point.payment.service.impl	PaymentServiceImpl	getDetail	결제 상세 정보를 반환한다.
com.gongzone.central.point.payment.service.impl	PaymentServiceImpl	getMany	결제 정보를 여러 개 반환한다.
com.gongzone.central.point.payment.service.impl	PaymentServiceImpl	charge	회심 포인트를 충전한다.
com.gongzone.central.point.payment.mapper	PaymentMapper	insert	결제 정보를 삽입한다.
com.gongzone.central.point.payment.mapper	PaymentMapper	get	결제 정보를 반환한다.
com.gongzone.central.point.payment.mapper	PaymentMapper	getMany	결제 정보를 여러 개 가져온다.
<b>withdraw</b>			
com.gongzone.central.point.withdraw.controller	WithdrawController	postMemberPointWithdraw	포인트 인출을 요청한다.
com.gongzone.central.point.withdraw.controller	WithdrawController	getPointWithdraw	특정 포인트 인출 현황을 요청한다.
com.gongzone.central.point.withdraw.controller	WithdrawController	getPointWithdraw	포인트 인출 현황을 요청한다.
com.gongzone.central.point.withdraw.service	WithdrawService	insert	포인트 인출 정보를 삽입한다.
com.gongzone.central.point.withdraw.service	WithdrawService	get	포인트 인출 정보를 반환한다.
com.gongzone.central.point.withdraw.service	WithdrawService	getMany	포인트 인출 정보를 여러 개 반환한다.
com.gongzone.central.point.withdraw.service	WithdrawService	withdraw	회심 포인트를 인출한다.
com.gongzone.central.point.withdraw.service.impl	WithdrawServiceImpl	insert	포인트 인출 정보를 삽입한다.
com.gongzone.central.point.withdraw.service.impl	WithdrawServiceImpl	get	포인트 인출 정보를 반환한다.
com.gongzone.central.point.withdraw.service.impl	WithdrawServiceImpl	getMany	포인트 인출 정보를 여러 개 반환한다.
com.gongzone.central.point.withdraw.service.impl	WithdrawServiceImpl	withdraw	회심 포인트를 인출한다.
com.gongzone.central.point.withdraw.mapper	WithdrawMapper	insert	포인트 인출 정보를 삽입한다.
com.gongzone.central.point.withdraw.mapper	WithdrawMapper	get	포인트 인출 정보를 반환한다.
com.gongzone.central.point.withdraw.mapper	WithdrawMapper	getMany	포인트 인출 정보를 여러 개 가져온다.
<b>scheduler</b>			
com.gongzone.central.scheduler	CommonScheduler	boardStatusUpdate	게시판 날짜 변경에 따른 상태 업데이트 스케줄러
<b>utils</b>			
com.gongzone.central	CentralApplication		스프링부트 설정 애플리케이션

## 5. 개발 / 서비스 환경

*	구 분	개발 환경		서비스 환경	
S / W	OS	Windows 10		CentOS Stream release 8	
	웹 브라우저	Chrome		Chrome 최적화	
	WAS	apache-tomcat-10.1		apache-tomcat-10.1	
	개발언어	서버	java		-
		클라이언트	javascript / html5 / css / thymeleaf / jquery / ajax		
	Framework	Tailwind / Spring boot / Mybatis 3.0 / React			
	DBMS	MySQL8.0.3		MySQL8.0.3	
Tools	Spring Tool Suite / Visual studio code / IntelliJ IDEA				

## 6. 주요 기능별 정리



### jsx 파일

```
import PartyListCard from "../../components/page/party/PartyListCard";
import { Party } from "../../utils/repository";
import { useEffect, useState } from "react";
import AuthStore from "../../utils/zustand/AuthStore";
import { useNavigate, useParams } from "react-router-dom";
import Skeleton from "react-loading-skeleton";
import "react-loading-skeleton/dist/skeleton.css";

export default function PartyList() {
  const { id } = useParams(); // URL 파라미터에서 ID 가져오기
  const [data, setData] = useState([]);
  const [isLoading, setIsLoading] = useState(true);
  const memberNo = AuthStore((state) => state.userInfo.memberNo);
  const navigate = useNavigate();
  const baseUrl = "https://gongzone.duckdns.org";

  const CustomSkeleton = ({ width, height }) => (
    <Skeleton width={width} height={height} className="custom-skeleton" />
  );

  const PartyListSkeleton = () => (
    <div className="mt-20">
      <div className="border rounded-lg p-4 shadow-md">
        <CustomSkeleton width={270} height={200} />
        <div className="mt-4">
          <CustomSkeleton width="60%" height={20} />
        </div>
      </div>
    </div>
  );
}
```

```

    </div>
    <div className="mt-2">
      <CustomSkeleton width="40%" height={20} />
    </div>
    <div className="mt-4 flex justify-between items-center">
      <CustomSkeleton width="30%" height={30} />
      <CustomSkeleton width="30%" height={20} />
    </div>
  </div>
</div>
);

const fetch = async () => {
  const detailData = await Party.PartyAccept(id);
  const responseData = Array.isArray(detailData.data)
    ? detailData.data
    : [detailData.data]; // 배열로 변환
  setData(responseData);
  setIsLoading(false);
  console.log("Fetched data:", responseData);
};

useEffect(() => {
  if (id) {
    fetch();
  }
}, [id]);

const handleCardClick = (partyNo) => {
  navigate(`/party/detail/${memberNo}/${partyNo}`, {
    state: { partyNo, data },
  });
};

const processedParties = new Set();

const waitingForPurchaseParties = data.filter((e) => {
  const isWaiting = e.status === "S060104";
  if (isWaiting) processedParties.add(e.partyNo);
  return isWaiting;
});

const completedParties = data.filter((e) => {
  const isCompleted = e.status === "S060108";
  if (isCompleted) processedParties.add(e.partyNo);
  return isCompleted;
});

const closedParties = data.filter((e) => {
  const isClosed = !processedParties.has(e.partyNo) && e.status === "S060103";
  if (isClosed) processedParties.add(e.partyNo);
  return isClosed;
});

```



```

const registeredParties = data.filter((e) => {
  const isRegistered =
    !processedParties.has(e.partyNo) && e.partyLeader === memberNo;
  if (isRegistered) processedParties.add(e.partyNo);
  return isRegistered;
});

const recruitingParties = data.filter((e) => {
  const isRecruiting =
    !processedParties.has(e.partyNo) && e.status === "S060101";
  if (isRecruiting) processedParties.add(e.partyNo);
  return isRecruiting;
});

const requestedParties = data.filter((e) => {
  const isRequested =
    !processedParties.has(e.partyNo) &&
    e.requestMember.some(r => r.memberNo === memberNo);
  if (isRequested) processedParties.add(e.partyNo);
  return isRequested;
});

const renderPartyCards = (parties) => {
  return parties.map((e) => (
    <div key={e.partyNo} onClick={() => handleCardClick(e.partyNo)}>
      <PartyListCard
        img={`/${baseURL}${e.img}`}
        desc={e.boardBody}
        title={e.boardTitle}
        id={e.partyNo}
        memberNo={memberNo}
        note={e.partyCateCode}
        like={e.like}
        amount={e.partyAmount}
        status={e.status}
      >
        <div className="text-sm px-3 pb-3">
          <div className="flex justify-between mb-3 text-[#888888]" ></div>
        </div>
      </PartyListCard>
    </div>
  ));
};

const renderSkeletons = () => {
  return Array.from({ length: 3 }).map((_, index) => (
    <PartyListSkeleton key={index} />
  ));
};

return (
  <div className="max-w-5xl mx-auto mb-10 mt-14 px-4 sm:px-6 lg:px-8">
    {isLoading ? (
      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">

```

```

    {renderSkeletons()}
  </div>
) : (
  <>
    {waitingForPurchaseParties.length > 0 && (
      <>
        <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
          제품 구매 대기중인 파티
        </div>
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
          {renderPartyCards(waitingForPurchaseParties)}
        </div>
      </>
    )}

    {closedParties.length > 0 && (
      <>
        <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
          모집 완료된 파티
        </div>
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
          {renderPartyCards(closedParties)}
        </div>
      </>
    )}

    {requestedParties.length > 0 && (
      <>
        <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
          참여 신청한 파티
        </div>
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
          {renderPartyCards(requestedParties)}
        </div>
      </>
    )}

    {registeredParties.length > 0 && (
      <>
        <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
          내가 등록한 파티
        </div>
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
          {renderPartyCards(registeredParties)}
        </div>
      </>
    )}

    {recruitingParties.length > 0 && (
      <>
        <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
          참여 중인 파티
        </div>
        <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">

```

```

        {renderPartyCards(recruitingParties)}
      </div>
    </>
  )}

  {completedParties.length > 0 && (
    <
      <div className="w-full mb-6 text-lg font-bold text-[#526688] mt-10">
        종료된 파티
      </div>
      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
        {renderPartyCards(completedParties)}
      </div>
    </>
  )}
</>
)}
</div>
);
}

```

```

package com.gongzone.central.party.accept.service;

import com.gongzone.central.board.domain.BoardReply;
import com.gongzone.central.board.mapper.BoardMapper;
import com.gongzone.central.member.alertSSE.domain.AlertSSE;
import com.gongzone.central.member.alertSSE.service.AlertSEEService;
import com.gongzone.central.member.mapper.MemberMapper;
import com.gongzone.central.party.accept.domain.AcceptDetail;
import com.gongzone.central.party.accept.domain.AcceptMember;
import com.gongzone.central.party.accept.domain.PartyMemberPurchase;
import com.gongzone.central.party.accept.domain.RequestMember;
import com.gongzone.central.party.accept.domain.RequestParty;
import com.gongzone.central.party.accept.mapper.AcceptMapper;
import com.gongzone.central.utils.MySqlUtil;
import com.gongzone.central.utils.StatusCode;
import java.util.ArrayList;
import java.util.List;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import reactor.core.publisher.Flux;
import reactor.core.publisher.Mono;
import reactor.core.scheduler.Schedulers;

@Service
@RequiredArgsConstructor
public class AcceptServiceImpl implements AcceptService {

    private final AcceptMapper acceptMapper;
    private final BoardMapper boardMapper;
    private final MemberMapper memberMapper;

```

```

private final AlertSEEService alertSEEService;

@Override
public PartyMemberPurchase getPurchaseInfo(String memberNo, String partyNo) {
    return acceptMapper.getPartyMemberPurchase(memberNo, partyNo);
}

@Override
public List<AcceptDetail> getPartyList(String partyNo) {
    AcceptDetail detail = acceptMapper.getPartyList(partyNo);
    System.out.println("detail1266 : " + detail);

    String partyNos = detail.getPartyNo();
    List<AcceptMember> participants = acceptMapper.getParticipants(partyNos);
    System.out.println("participants : " + participants);
    detail.setParticipants(participants);

    List<RequestMember> requestMember = acceptMapper.getRequestMember(partyNos);
    System.out.println("requestMember : " + requestMember);
    detail.setRequestMember(requestMember);

    List<AcceptDetail> details = new ArrayList<>();
    details.add(detail);
    return details;
}

@Override
public AcceptDetail getPartyDetailByPartyNo(String partyNo) {
    AcceptDetail detail = acceptMapper.getPartyDetail(partyNo);
    System.out.println("detail1266 : " + detail);

    String partyNos = detail.getPartyNo();
    List<AcceptMember> participants = acceptMapper.getParticipants(partyNos);
    System.out.println("participants : " + participants);
    detail.setParticipants(participants);

    List<RequestMember> requestMember = acceptMapper.getRequestMember(partyNos);
    System.out.println("requestMember : " + requestMember);
    detail.setRequestMember(requestMember);

    String boardNos = detail.getBoardNo();
    List<BoardReply> boardReply = boardMapper.getBoardReplyList(boardNos);
    for (int i = 0; i < boardReply.size(); i++) {
        String memberNo = boardReply.get(i).getMemberNo();
        String memberId = (memberMapper.info(memberNo).getMemberId());
        boardReply.get(i).setMemberId(memberId);
    }
    detail.setBoardReply(boardReply);

    return detail;
}

@Override
public List<AcceptMember> getParticipants(String partyNo) {

```

```

        return acceptMapper.getParticipants(partyNo);
    }

    @Override
    public List<String> getPartyNo(String memberNo) {
        return acceptMapper.findPointNoByMemberNo(memberNo);
    }

    @Override
    public List<RequestMember> getRequestMember(String partyNo) {
        return acceptMapper.getRequestMember(partyNo);
    }

    @Override
    public List<AcceptDetail> getListParty(String memberNo) {
        List<String> partyNos = getPartyNo(memberNo);
        System.out.println("partyNos :" + partyNos);
        List<AcceptDetail> details = new ArrayList<>();

        for (String partyNo : partyNos) {
            List<AcceptDetail> detail = getPartyList(partyNo);
            details.addAll(detail);
        }
        System.out.println("details on array : " + details);

        return details;
    }

    @Override
    public Mono<Void> completeParty(String partyNo) {
        AcceptDetail detail = acceptMapper.getPartyList(partyNo);
        String partyNos = detail.getPartyNo();
        List<AcceptMember> participants = acceptMapper.getParticipants(partyNos);

        // 안전장치 추가: remainAmount가 0이 아닌 경우 메서드 종료
        if (Integer.parseInt(detail.getRemainAmount()) != 0) {
            System.out.println("remainAmount가 0이 아니므로 completeParty 메서드를 종료
합니다.");
            return Mono.empty();
        }
        System.out.println("detail in complete" + detail);
        acceptMapper.completeBoardStatus(detail.getBoardNo());
        acceptMapper.completePartyStatus(detail.getPartyNo());
        System.out.println("participants" + participants);
        String leaderNo = detail.getPartyLeader();
        for (AcceptMember member : participants) {

            // 확인후 주식 제거해주세요; 0716 우석
            // 파티장 결제완료 상태로 넣는 것에서 -> 파티장은 결제현황에 넣지 않도록 수정
            if (member.getMemberNo().equals(leaderNo)) {
                continue;
            }
        }

        acceptMapper.insertPartyPurchase(detail.getPartyNo(),

```



```

        acceptMapper.updateAmountAfterKick(requestParty);
        return kickAlert(memberNo);
    } else if (statusCode == StatusCode.REQUEST) {
        System.out.println("55555555555555555555555555555555");
        RequestParty requestParty = new RequestParty();
        requestParty.setPartyNo(partyNo);
        requestParty.setMemberNo(memberNo);
        requestParty.setRequestAmount(requestAmount);
        System.out.println("requestParty in request :" + requestParty);

        acceptMapper.requestJoin(requestParty);

        String leaderNo = acceptMapper.getPartyLeaderByPartyNo(partyNo);
        // memberNo 대신 leaderNo == 결국 memberNo
        return requestAlert(leaderNo);
    }
    return Mono.empty();
}

@Override
public RequestParty getRequestMemberBymemberNo(String memberNo, String partyNo) {
    return acceptMapper.requestMemberBymemberNo(memberNo, partyNo);
}

private Mono<Void> acceptAlert(String memberNo) {
    System.out.println("22222222222222222222222222222222");
    System.out.println("22222222222222222222222222222222");
    System.out.println("22222222222222222222222222222222");
    System.out.println("newLastPartyMemberNo : " + memberNo);
    return Mono.fromCallable(() -> {
        AlertSSE alertSSE = new AlertSSE();
        alertSSE.setMemberNo(memberNo);
        alertSSE.setTypeCode("T010206"); // 알림 유형 코드
        alertSSE.setAlertDetail("파티에 수락되었습니다.");
        return alertSSE;
    })
        .flatMap(alertSSE -> alertSEEService.sendAlert(alertSSE)) // 단일 줄
        .subscribeOn(Schedulers.boundedElastic())
        .doOnError(e -> {
            e.printStackTrace();
        })
        .then(); // sendAlert의 반환 타입을 Mono<Void>로 변경
}

private Mono<Void> requestAlert(String memberNo) {
    System.out.println("22222222222222222222222222222222");
    System.out.println("22222222222222222222222222222222");
    System.out.println("22222222222222222222222222222222");
    System.out.println("newLastPartyMemberNo : " + memberNo);
    return Mono.fromCallable(() -> {

```

로 변경



```

        AlertSSE alertSSE = new AlertSSE();
        alertSSE.setMemberNo(memberNo);
        alertSSE.setTypeCode("T010206"); // 알람 유형 코드
        alertSSE.setAlertDetail("파티 신청이 있습니다.");
        return alertSSE;
    })
    .flatMap(alertSSE -> alertSEEService.sendAlert(alertSSE)) // 단일 줄
    .subscribeOn(Schedulers.boundedElastic())
    .doOnError(e -> {
        e.printStackTrace();
    })
    .then(); // sendAlert의 반환 타입을 Mono<Void>로 변경
}

private Mono<Void> kickAlert(String memberNo) {
    System.out.println("22222222222222222222");
    System.out.println("22222222222222222222");
    System.out.println("22222222222222222222");
    System.out.println("newLastPartyMemberNo : " + memberNo);
    return Mono.fromCallable(() -> {
        AlertSSE alertSSE = new AlertSSE();
        alertSSE.setMemberNo(memberNo);
        alertSSE.setTypeCode("T010206"); // 알람 유형 코드
        alertSSE.setAlertDetail("파티에서 강퇴되었습니다.");
        return alertSSE;
    })
    .flatMap(alertSSE -> alertSEEService.sendAlert(alertSSE)) // 단일 줄
    .subscribeOn(Schedulers.boundedElastic())
    .doOnError(e -> {
        e.printStackTrace();
    })
    .then(); // sendAlert의 반환 타입을 Mono<Void>로 변경
}

private Mono<Void> completeAlert(String memberNo) {
    System.out.println("22222222222222222222");
    System.out.println("22222222222222222222");
    System.out.println("22222222222222222222");
    System.out.println("newLastPartyMemberNo : " + memberNo);
    return Mono.fromCallable(() -> {
        AlertSSE alertSSE = new AlertSSE();
        alertSSE.setMemberNo(memberNo);
        alertSSE.setTypeCode("T010206"); // 알람 유형 코드
        alertSSE.setAlertDetail("파티 모집이 완료되었습니다. 결제를
진행해주세요!");
        return alertSSE;
    })
    .flatMap(alertSSE -> alertSEEService.sendAlert(alertSSE)) // 단일 줄
    .subscribeOn(Schedulers.boundedElastic())
    .doOnError(e -> {
        e.printStackTrace();
    })
}

```

```

        })
        .then(); // showAlert의 반환 타입을 Mono<Void>로 변경
    }
    // @Override
    // public void deletePartyStatusByNo(String memberNo) {
    //     acceptMapper.deletePartyStatus(memberNo);
    // }
}

```

## Mapper

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.gongzone.central.party.accept.mapper.AcceptMapper">
    <select id="getPartyList" resultType="AcceptDetail">
        SELECT
            b.b_title AS boardTitle,
            CONCAT('[', c.c_group1, ']', c.c_group2, ' - ', c.c_group3) AS partyCateCode,
            p.p_url AS productUrl,
            b.b_body AS boardBody,
            p.p_amount AS partyAmount,
            p.p_amount_remain AS remainAmount,
            p.p_price AS partyPrice,
            p.p_price_remain AS remainPrice,
            b.b_period AS endDate,
            p.status_code AS status,
            CONCAT(l.location_do, ' ', l.location_si, ' ', l.location_gu, ' ', l.location_dong, ' ',
                l.location_detail) AS address,
            p.p_no AS partyNo,
            b.m_no AS partyLeader,
            f.file_path AS img,
            b.b_no AS boardNo,
            b.status_code AS boardStatus,
            ps.shipping_no AS shippingNo,
            pts.status_code AS settlementCode

        FROM
            party AS p
            INNER JOIN
            board AS b
            ON
                p.b_no = b.b_no
            INNER JOIN
            location AS l
            ON
                b.b_no = l.b_no
            category AS c
            ON
                c.c_code = p.c_code
    </select>
</mapper>

```

```

        INNER JOIN
        file_relation AS fr
        ON
            b.b_no = fr.file_usage
        INNER JOIN
        file AS f
        ON
            f.file_no = fr.file_no
        LEFT JOIN
        party_shipping AS ps
        ON
            p.p_no = ps.p_no
        LEFT JOIN
        party_settlement AS pts
        ON
            p.p_no = pts.p_no
    WHERE
        p.p_no = #{partyNo};
</select>

```

```

<select id="getParticipants" resultType="com.gongzone.central.party.accept.domain.AcceptMember">
    SELECT
        m.m_nick AS memberNick,
        pm.pm_amount AS memberAmount,
        m.m_email AS memberEmail,
        m.m_no AS memberNo,
        pm.p_no AS partyNo,
        pm.pm_price AS requestPrice,
        pm.p_member_no AS partyMemberNo,
        pr.r_no AS receptionNo
    FROM
        party_member AS pm
        INNER JOIN
        member AS m
        ON
            pm.m_no = m.m_no
        LEFT JOIN
        party_reception AS pr
        ON
            pm.p_member_no = pr.p_member_no
    WHERE
        pm.p_no = #{partyNo};
</select>

```

```

<select id="getRequestMember"
resultType="com.gongzone.central.party.accept.domain.RequestMember">
    SELECT
        m.m_nick AS memberNick,
        pr.request_amount AS memberAmount,
        m.m_email AS memberEmail,
        pr.status_code AS requestStatus,
        m.m_no AS memberNo,
        pr.p_no AS partyNo
    FROM
        party_request AS pr

```

```

        INNER JOIN
        member AS m ON pr.m_no = m.m_no
    WHERE
        pr.p_no = #{partyNo};
</select>

```

<!-- 파티 번호 찾기 -->

```

<select id="findPointNoByMemberNo" parameterType="String" resultType="String">
    (SELECT DISTINCT p.p_no AS partyNo
    FROM party AS p
    INNER JOIN party_member AS pm ON p.p_no = pm.p_no
    <where>
        <if test="memberNo != '_admin'">
            pm.m_no = #{memberNo}
        </if>
    </where>
    )
    UNION
    (SELECT DISTINCT p.p_no AS partyNo
    FROM party AS p
    INNER JOIN party_request AS pr ON p.p_no = pr.p_no
    <where>
        <if test="memberNo != '_admin'">
            pr.m_no = #{memberNo}
        </if>
    </where>
    )
</select>

```

```

<update id="updatePartyStatus">
    UPDATE
        party_request
    SET
        status_code = #{statusCode.code}
    WHERE
        m_no = #{memberNo};
</update>

```

```

<insert
    id="insertPartyMember"
    parameterType="com.gongzone.central.party.accept.domain.RequestParty">
    <selectKey
        keyColumn="requestPrice"
        keyProperty="requestPrice"
        resultType="int"
        order="BEFORE">
        SELECT
            (#{requestAmount} * (SELECT CEIL(ROUND(p.p_price / p.p_amount, -1)) FROM party AS p
            WHERE p.p_no = #{partyNo})) AS requestPrice
    </selectKey>
    INSERT INTO party_member
    (p_member_no, p_no, m_no, pm_amount, pm_price, member_join_date, member_is_leader)
    VALUES (
        #{partyMemberNo},
        #{partyNo},

```

```

        #{memberNo},
        #{requestAmount},
        #{requestPrice},
        NOW(),
        '파티원'
    )
</insert>

<update id="updateAmountMember"
parameterType="com.gongzone.central.party.accept.domain.RequestParty">
    UPDATE party
    SET
        p_amount_remain = p_amount_remain - #{requestAmount},
        p_price_remain = p_price_remain - #{requestPrice},
        status_code = CASE
            WHEN p_amount_remain - #{requestAmount} = 0 THEN 'S060102'
            ELSE status_code
        END
    WHERE p_no = #{partyNo};
</update>

<delete id="deletePartyStatus">
    UPDATE
        party_request
    SET
        status_code = #{statusCode.code}
    WHERE
        m_no = #{memberNo};
</delete>

<select id="requestMemberBymemberNo"
resultType="com.gongzone.central.party.accept.domain.RequestParty">
    SELECT
        p_request_no AS requestNo,
        p_no AS partyNo,
        m_no AS memberNo,
        request_amount AS requestAmount,
        request_date AS requestDate,
        status_code AS statusCode
    FROM
        party_request
    WHERE
        m_no = #{memberNo} AND p_no = #{partyNo};
</select>

<select id="lastPartyMemberNo" resultType="String">
    select
        MAX(p_member_no)
    FROM
        party_member
</select>

```

```

<delete id="kickPartyMember">
  DELETE FROM party_member
  WHERE
    p_no = #{partyNo} AND m_no = #{memberNo};
</delete>

<delete id="deletePartyRequest">
  DELETE FROM party_request
  WHERE
    p_no = #{partyNo} AND m_no = #{memberNo};
</delete>

<select id="getPartyUnitPrice" parameterType="String" resultType="int">
  SELECT CEIL(ROUND(p.p_price / p.p_amount, -1))
  FROM party AS p
  WHERE p.p_no = #{partyNo}
</select>

<update id="updateAmountAfterKick"
parameterType="com.gongzone.central.party.accept.domain.RequestParty">
  UPDATE party
  SET
    p_amount_remain = p_amount_remain + #{requestAmount},
    p_price_remain = p_price_remain + #{requestPrice}
  WHERE p_no = #{partyNo};
</update>

<insert id="requestJoin" parameterType="com.gongzone.central.party.accept.domain.RequestParty">
  INSERT INTO party_request
    (p_no, m_no, request_amount, request_date, status_code)
  VALUES
    (
      #{partyNo},
      #{memberNo},
      #{requestAmount},
      NOW(),
      "S060201"
    )
</insert>

<select id="getPartyDetail" resultType="AcceptDetail">
  SELECT
    b.b_title AS boardTitle,
    CONCAT('[', c.c_group1, ']', c.c_group2, ' - ', c.c_group3) AS partyCateCode,
    p.p_url AS productUrl,
    b.b_body AS boardBody,
    p.p_amount AS partyAmount,
    p.p_amount_remain AS remainAmount,
    p.p_price AS partyPrice,
    p.p_price_remain AS remainPrice,
    b.b_period AS endDate,
    p.status_code AS status,
    CONCAT(I.location_do, ' ', I.location_si, ' ', I.location_gu, ' ', I.location_dong, ' ',
I.location_detail) AS address,

```

```

p.p_no AS partyNo,
b.m_no AS partyLeader,
f.file_path AS img,
b.b_no AS boardNo,
ps.shipping_no AS shippingNo,
pts.status_code AS settlementCode

FROM
  party AS p
    INNER JOIN
  board AS b
    ON
      p.b_no = b.b_no
    INNER JOIN
  location AS l
    ON
      b.b_no = l.b_no
    INNER JOIN
  category AS c
    ON
      c.c_code = p.c_code
    INNER JOIN
  file_relation AS fr
    ON
      b.b_no = fr.file_usage
    INNER JOIN
  file AS f
    ON
      f.file_no = fr.file_no
    LEFT JOIN
  party_shipping AS ps
    ON
      p.p_no = ps.p_no
    LEFT JOIN
  party_settlement AS pts
    ON
      p.p_no = pts.p_no
WHERE
  p.p_no = #{partyNo};
</select>

```

```

<update id="completeBoardStatus">
  UPDATE board
  SET
    status_code='S040102'
  WHERE
    b_no = #{boardNo};
</update>

```

```

<update id="completePartyStatus">
  UPDATE party
  SET
    status_code='S060103'

```



```

WHERE
    p_no = #{partyNo};
</update>

<insert id="insertPartyPurchase">
    INSERT INTO party_purchase
        (p_no, p_member_no, purchase_price, status_code)
    VALUES
        (#{partyNo}, #{partyMemberNo}, #{requestPrice}, 'S060301')
</insert>

<insert id="updatePartyLeaderPurchase">

    UPDATE
        party_purchase
    SET
        status_code = 'S060302'
    WHERE
        m_no = #{memberNo};

</insert>

<select id="getPartyMemberPurchase" resultType="PartyMemberPurchase">
    SELECT
        pp.purchase_no AS purchaseNo,
        pp.purchase_price AS purchasePrice,
        mp.m_point AS memberPoint
    FROM
        party_purchase AS pp
        INNER JOIN
        party_member AS pm
        ON
            pp.p_member_no = pm.p_member_no
        INNER JOIN
        member_point AS mp
        ON
            pm.m_no = mp.m_no
    WHERE
        pm.m_no = #{memberNo} AND pm.p_no = #{partyNo}
</select>

<select id="getPartyLeaderByPartyNo">
    SELECT
        m_no AS memberNo
    FROM
        party_member
    WHERE
        p_no = #{partyNo}
        AND
        member_is_leader = '파티장'
</select>

</mapper>

```

# 게시글

게시판
관리
공지사항

---

**게시글 제목**

**카테고리**

**구매 URL**

**제품 총 가격 (배송비 포함)**

**전체 수량**

**구매 희망 수량**

**제품 이미지**

파일 선택

**게시글 상세 내용**

Paragraph
B
I
🔗
📷
📄
📄
📄
📄

게시글 상세 내용을 입력해주세요

**제품 수령 주소**

**게시글 마감일**

July 2024

Su	Mo	Tu	We	Th	Fr	Sa
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

완료

---

더 많은  
 공지사항  
 확인하기

**바로가기**

[게시판](#)  
[FAQ](#)

**이용안내**

[공지사항](#)  
[FAQ](#)

**관련 사이트**

[네이버 블로그](#)  
[다산뉴스네트웍스](#)  
[인스타그램](#)  
[Vlog](#)  
[Medium](#)

**서비스의뢰**

[네이버 예약](#)  
[네이버 예약문의](#)  
[신원스튜디오](#)  
[고객센터](#)

## jsx 파일

```
import BoardContent from "../../components/page/board/BoardContent";
import React, { useState, useEffect } from "react";
import Calendar from "../../components/page/board/BoardCalendar";
import BoardMap from "../../components/page/board/BoardMap";
import AuthStore from "../../utils/zustand/AuthStore";
import GZAPI from "../../utils/api";
import { useNavigate } from "react-router-dom";

export default function InsertForm() {
  const memberNo = AuthStore((state) => state.userInfo.memberNo);
  const navigate = useNavigate();

  const [formData, setFormData] = useState({
    memberNo: "",
    title: "",
    category: "",
    URL: "",
    price: "",
    total: "",
    amount: "",
    image: null,
    content: "",
    doCity: "",
    siGun: "",
    gu: "",
    dong: "",
    detailAddress: "",
    latitude: "",
    longitude: "",
    endDate: "",
  });

  const [numError, setNumError] = useState("");
  const [canSubmit, setCanSubmit] = useState(true);
  const [dateError, setDateError] = useState("");
  const [submitError, setSubmitError] = useState(""); // 제출 오류 메시지 상태
  const [submitSuccess, setSubmitSuccess] = useState(""); // 제출 성공 메시지 상태

  useEffect(() => {
    if (memberNo) {
      setFormData((prevFormData) => ({
        ...prevFormData,
        memberNo: memberNo,
      }));
    }
  }, [memberNo]);

  const handleChange = (e) => {
    const { name, value, type, files } = e.target;

    if (type === "file") {
      setFormData((prevFormData) => ({
```

```

    ...prevFormData,
    [name]: files[0],
  });
} else {
  setFormData((prevFormData) => ({
    ...prevFormData,
    [name]: value,
  }));
}
};

const handleBlur = (e) => {
  const { name, value } = e.target;
  if (name === "total" || name === "amount") {
    validateAmount(
      name === "total" ? value : formData.total,
      name === "amount" ? value : formData.amount
    );
  }
};

const validateAmount = (total, amount) => {
  if (parseInt(amount) > parseInt(total)) {
    setNumError("구매 희망 수량은 전체 수량보다 많을 수 없습니다.");
    setCanSubmit(false);
  } else {
    setNumError("");
    setCanSubmit(true);
  }
};

const handleEndDateChange = (date) => {
  const selectedDate = new Date(date);
  const currentDate = new Date();

  if (selectedDate < currentDate) {
    setDateError("마감일은 현재 날짜보다 이후여야 합니다.");
    setCanSubmit(false);
  } else {
    setDateError("");
    setCanSubmit(true);
  }

  setFormData((prevFormData) => ({
    ...prevFormData,
    endDate: date,
  }));
};

const handleContentChange = (content) => {
  setFormData((prevFormData) => ({
    ...prevFormData,
    content: content,
  }));
};

```

```

};

const handleLocationChange = (location) => {
  const parts = location.split(" ");
  const doCity = parts[0];
  const siGun = parts[1];
  const gu = parts[2];
  const dong = parts[3];
  const detailAddress = parts.slice(4).join(" ");

  setFormData((prevFormData) => ({
    ...prevFormData,
    doCity: doCity,
    siGun: siGun,
    gu: gu,
    dong: dong,
    detailAddress: detailAddress,
  }));
};

const handlePositionChange = (lat, lng) => {
  setFormData((prevFormData) => ({
    ...prevFormData,
    latitude: lat,
    longitude: lng,
  }));
};

const handleSubmit = async (e) => {
  e.preventDefault();

  if (
    formData.content === "" ||
    formData.doCity === "" ||
    formData.siGun === "" ||
    formData.gu === "" ||
    formData.dong === "" ||
    formData.detailAddress === "" ||
    formData.latitude === "" ||
    formData.longitude === "" ||
    formData.endDate === ""
  ) {
    alert("필수 입력 항목을 모두 작성해주세요.");
    return;
  }

  if (!canSubmit) {
    alert("입력된 데이터에 오류가 있습니다. 확인 후 다시 시도하세요.");
    return;
  }

  const formDataToSend = new FormData();
  for (const key in formData) {
    formDataToSend.append(key, formData[key]);
  }
};

```

```

}

// FormData 객체에 저장된 데이터를 콘솔에 출력
for (const [key, value] of formDataToSend.entries()) {
  console.log(`${key}:`, value);
}

try {
  const response = await GZAPI.post(
    `/api/boards/write/${memberNo}`,
    formDataToSend,
    {
      headers: {
        "Content-Type": "multipart/form-data",
      },
      transformRequest: [
        function () {
          return formDataToSend;
        },
      ],
    }
  );

  if (response.status === 200) {
    setSubmitSuccess("게시글이 성공적으로 등록되었습니다!");
    setSubmitError("");
    setTimeout(() => {
      navigate("/board/list"); // 성공 시 이동할 페이지 경로 설정
    }, 500); // 0.5초 후에 페이지 이동
  } else {
    setSubmitSuccess("");
    setSubmitError("게시글 등록에 실패했습니다.");
  }
} catch (error) {
  // 에러 처리
  console.error("폼 데이터 전송 중 오류가 발생했습니다.", error);
  setSubmitSuccess("");
  setSubmitError(
    "게시글 등록 중 오류가 발생했습니다. 나중에 다시 시도하세요."
  );
}
};

const cate = [
  { key: "c0", value: "", label: "제품 카테고리 선택" },
  { key: "c1", value: "CF0101", label: "채소" },
  { key: "c2", value: "CF0102", label: "과일" },
  { key: "c3", value: "CF0103", label: "수산물/건어물" },
  { key: "c4", value: "CF0104", label: "정육/계란류" },
  { key: "c5", value: "CF0105", label: "우유/유제품" },
  { key: "c6", value: "CF0201", label: "쌀/잡곡" },
  { key: "c7", value: "CF0202", label: "견과류" },
  { key: "c8", value: "CF0301", label: "김치/반찬" },
  { key: "c9", value: "CF0302", label: "밀키트" },

```

```

{ key: "c10", value: "CF0401", label: "면류/통조림" },
{ key: "c11", value: "CF0402", label: "양념/오일" },
{ key: "c12", value: "CF0403", label: "간식/과자" },
{ key: "c13", value: "CF0404", label: "베이커리/잼" },
{ key: "c14", value: "CF0501", label: "생수/음료" },
{ key: "c15", value: "CF0502", label: "커피/차" },
{ key: "c16", value: "CF9901", label: "건강식품" },
];

return (
  <form
    onSubmit={handleSubmit}
    encType="multipart/form-data"
    className="p-4 space-y-4 w-[1000px] mx-auto mb-10 mt-14"
  >
    <div>
      <label className="block">게시글 제목</label>
      <input
        type="text"
        name="title"
        value={formData.title}
        onChange={handleChange}
        className="border p-2 w-full"
        placeholder="제품명을 반드시 입력해주세요."
        required
      />
    </div>

    <div>
      <label className="block">카테고리</label>
      <select
        name="category"
        value={formData.category}
        onChange={handleChange}
        className="border p-2 w-full"
        required
      >
        {cate.map((item) => (
          <option key={item.key} value={item.value}>
            {item.label}
          </option>
        ))}
      </select>
    </div>

    <div>
      <label className="block">구매 URL</label>
      <input
        type="text"
        name="URL"
        value={formData.URL}
        onChange={handleChange}
        className="border p-2 w-full"
        placeholder="제품 URL을 입력해주세요."
      />
    </div>
  </form>
);

```



```

    required
  />
</div>

<div>
  <label className="block">제품 총 가격 (배송비 포함)</label>
  <input
    type="number"
    name="price"
    value={formData.price}
    onChange={handleChange}
    className="border p-2 w-full"
    placeholder="제품 총 가격을 (숫자만) 입력해주세요."
    required
  />
</div>

<div className="flex space-x-4">
  <div className="w-1/2">
    <label className="block">전체 수량</label>
    <input
      type="number"
      name="total"
      value={formData.total}
      onChange={handleChange}
      onBlur={handleBlur}
      className="border p-2 w-full"
      placeholder="제품 전체 수량을 (숫자만) 입력해주세요."
      required
    />
  </div>

  <div className="w-1/2">
    <label className="block">구매 희망 수량</label>
    <input
      type="number"
      name="amount"
      value={formData.amount}
      onChange={handleChange}
      onBlur={handleBlur}
      className="border p-2 w-full"
      placeholder="희망하시는 구매 수량을 (숫자만) 입력해주세요."
      required
    />
    {numError && <p className="text-red-500">{numError}</p>}
  </div>
</div>

<div>
  <label className="block">제품 이미지</label>
  <input
    id="fileInput"
    type="file"
    name="image"
  />
</div>

```

```

    accept="image/*"
    onChange={handleChange}
    style={{ display: "none" }}
    required
  />

  <button
    type="button"
    onClick={() => document.getElementById("fileInput").click()}
    className="bg-[#1d5091] text-white px-2 py-2 rounded-md hover:bg-[#6ea2d4]"
  >
    파일 선택
  </button>

  {/* 이미지 미리보기 */}
  {formData.image && (
    <img
      alt="메인사진"
      src={URL.createObjectURL(formData.image)}
      className="max-w-[100px] border border-gray-300 rounded"
    />
  )}
</div>

<div className="flex space-x-4">
  {/* 게시글 상세 내용 */}
  <div className="w-full">
    <BoardContent onChange={handleContentChange} />
  </div>
</div>
<div className="flex justify-between gap-10">
  {/* 지도 */}
  <div className="w-2/3">
    제품 수령 주소
    {/* <MapSearch /> */}
    <BoardMap
      onLocationChange={handleLocationChange}
      onPositionChange={handlePositionChange}
    />
  </div>
  <div className="w-1/3">
    <label>게시글 마감일</label>
    <Calendar
      selected={formData.endDate}
      onChange={handleEndDateChange}
    />
    {dateError && <p className="text-red-500">{dateError}</p>}
  </div>
</div>

<div className="flex justify-end">
  <button
    type="submit"
    className="bg-[#1d5091] text-white px-10 py-3 rounded-md hover:bg-[#6ea2d4]"
  >

```

```

    >
    제출
  </button>
</div>

  {submitSuccess && <p className="text-green-500">{submitSuccess}</p>}
  {submitError && <p className="text-red-500">{submitError}</p>}
</form>
);
}

```

## Service

```

package com.gongzone.central.board.service;

import com.gongzone.central.board.domain.*;
import com.gongzone.central.board.mapper.BoardMapper;
import com.gongzone.central.file.domain.FileUpload;
import com.gongzone.central.file.mapper.FileMapper;
import com.gongzone.central.file.util.FileUtil;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import org.springframework.web.multipart.MultipartFile;

import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

@Service
@Transactional
@RequiredArgsConstructor
public class BoardServiceImpl implements BoardService {
    private final BoardMapper boardMapper;
    private final FileMapper fileMapper;
    private final FileUtil fileUtil;

    @Override
    @Transactional
    public void setValue(BoardResponse br, MultipartFile file) {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");
        LocalDateTime endDateTime = LocalDateTime.parse(br.getEndDate(), formatter);
        Board board = Board.builder()
            .memberNo(br.getMemberNo())
            .boardTitle(br.getTitle())
            .category(br.getCategory())
            .productUrl(br.getURL())
            .totalPrice(br.getPrice())
            .total(br.getTotal())
            .amount(br.getAmount())
            .boardBody(br.getContent())

```

```
        .locationDo(br.getDoCity())
        .locationSi(br.getSiGun())
        .locationGu(br.getGu())
        .locationDong(br.getDong())
        .locationDetail(br.getDetailAddress())
        .locationX(br.getLatitude())
        .locationY(br.getLongitude())
        .endDate(endDateTime)
        .build();

    boardMapper.insertBoard(board);

    FileUpload fileUpload = fileUtil.parseFileInfo(file);
    fileMapper.addFile(fileUpload);

    board.setFileNo(fileUpload.getFileIdx());
    board.setFileUsage(board.getBoardNo());
    boardMapper.insertFileRelation(board);

    boardMapper.insertLocation(board);

    board.setRemain(board.getTotal() - board.getAmount());
    int unitPrice = (int) Math.ceil((double) board.getTotalPrice() / board.getTotal());
    board.setRemainPrice(board.getTotalPrice() - (unitPrice * board.getAmount()));
    boardMapper.insertParty(board);

    board.setAmountPrice(unitPrice * board.getAmount());
    boardMapper.insertPartyMember(board);
}
}
```

### Mapper

```
=====BoardMapper.java=====
package com.gongzone.central.board.mapper;

import com.gongzone.central.board.domain.Board;
import org.apache.ibatis.annotations.Mapper;

@Mapper
public interface BoardMapper {
    void insertBoard(Board board);

    void insertLocation(Board board);

    void insertFileRelation(Board board);

    void insertParty(Board board);

    void insertPartyMember(Board board);
}
```

```

=====FileMapper.java=====
package com.gongzone.central.file.mapper;

import com.gongzone.central.file.domain.FileUpload;
import org.apache.ibatis.annotations.Mapper;

@Mapper
public interface FileMapper {
    void addFile(FileUpload fileUpload);
}

=====BoardMapper.xml=====
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.gongzone.central.board.mapper.BoardMapper">
<insert id="insertBoard" parameterType="com.gongzone.central.board.domain.Board">
    <selectKey keyProperty="boardNo" resultType="String" order="BEFORE">
        SELECT CONCAT('B', LPAD(COALESCE(MAX(CAST(SUBSTRING(b_no, 2) AS UNSIGNED)), 0) +
1, 6, '0')) AS boardNo
        FROM board
    </selectKey>
    INSERT INTO board
    (b_no, m_no, b_title, b_body, b_date, b_period, b_view_count, b_wish_count, b_report_count,
status_code)
    VALUES (#{boardNo}, #{memberNo}, #{boardTitle}, #{boardBody}, NOW(), #{endDate}, 0, 0, 0,
'S040101')
    </insert>

    <insert id="insertFileRelation" parameterType="com.gongzone.central.board.domain.Board"
useGeneratedKeys="true"
        keyProperty="fileRelationNo" keyColumn="file_relation_no">
        INSERT INTO file_relation (file_no, file_usage)
        VALUES (#{fileNo}, #{fileUsage})
    </insert>

    <insert id="insertLocation" parameterType="com.gongzone.central.board.domain.Board"
useGeneratedKeys="true"
        keyProperty="locationNo" keyColumn="location_no">
        INSERT INTO location
        (b_no, location_do, location_si, location_gu, location_dong, location_detail, location_x,
location_y)
        VALUES (#{boardNo}, #{locationDo}, #{locationSi}, #{locationGu}, #{locationDong},
#{locationDetail},
            #{locationX}, #{locationY})
    </insert>

    <insert id="insertParty" parameterType="com.gongzone.central.board.domain.Board">
    <selectKey keyProperty="partyNo" resultType="String" order="BEFORE">
        SELECT CONCAT('P', LPAD(COALESCE(MAX(CAST(SUBSTRING(p_no, 2) AS UNSIGNED)), 0) +
1, 6, '0')) AS partyNo
        FROM party

```

```

</selectKey>
INSERT INTO party
(p_no, b_no, c_code, p_url, p_amount, p_amount_remain, p_price, p_price_remain,
p_start_date, p_end_date,
status_code)
VALUES (#{partyNo}, #{boardNo}, #{category}, #{productUrl}, #{total}, #{remain}, #{totalPrice},
#{remainPrice},
NOW(), NULL, 'S060101')
</insert>

<insert id="insertPartyMember" parameterType="com.gongzone.central.board.domain.Board">
<selectKey keyProperty="partyMemberNo" resultType="String" order="BEFORE">
SELECT CONCAT('PM', LPAD(COALESCE(MAX(CAST(SUBSTRING(p_member_no, 3) AS
UNSIGNED)), 0) + 1, 6, '0')) AS
partyMemberNo
FROM party_member
</selectKey>
INSERT INTO party_member
(p_member_no, p_no, m_no, pm_amount, pm_price, member_join_date, member_is_leader)
VALUES (#{partyMemberNo}, #{partyNo}, #{memberNo}, #{amount}, #{amountPrice}, NOW(), '
파티장')
</insert>
</mapper>

=====FileMapper.xml=====
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.gongzone.central.file.mapper.FileMapper">
<insert id="addFile" parameterType="com.gongzone.central.file.domain.FileUpload"
useGeneratedKeys="true" keyProperty="fileIdx" keyColumn="file_no">
INSERT INTO file (file_original_name, file_new_name, file_path, file_size, file_date)
VALUES (#{fileOriginalName}, #{fileNewName}, #{filePath}, #{fileSize}, NOW())
</insert>
</mapper>

```

## 포인트

포인트 충전하기

20,000 ₩

+1만 +3만 +5만 +10만 초기화

간편결제

토스페이

일반결제

신용카드 휴대폰 계좌이체 기상계좌

### jsx 파일

```
=====api.jsx=====
import axios from "axios";
const GZAPI = axios.create({
  baseURL: process.env.REACT_APP_API_URL,
  headers: {
    "Content-Type": "application/json",
  },
});
GZAPI.interceptors.request.use(
  (config) => {
    const token = window.localStorage.getItem("accessToken");
    if (token) {
      //console.log(token);
      config.headers["Authorization"] = `Bearer ${token}`;
    }
    return config;
  },
  (error) => {
    return Promise.reject(error);
  }
);
export default GZAPI;
=====PointCharge.jsx=====
```



```

requestPointCharge: async () => {
  const payData = Payment(pointCharge.value);
  const payResponse = await PortOne.requestPayment(payData); // i'mport 결제 API
  // 오류 발생
  if (payResponse.code != null) {
    console.log(payResponse.message);
    alert("오류가 발생했습니다. 잠시 후 다시 시도해주세요.");
    return; // 종료
  }
  // 정상 처리
  const data = {
    pointChange: pointCharge.value,
    changeType: "T030101",
    detail: {
      ...payResponse,
      type: "TOSSPAY",
      status: "S030201",
    },
  };
  const response = await GZAPI.post(`/api/members/${ memberNo }/point/payment`, data);
  if (response.data.result === "SUCCESS") {
    alert('포인트 결제가 완료되었습니다.');
```

## Controller

```

=====PaymentController.java=====
package com.gongzone.central.point.payment.controller;
import com.gongzone.central.common.Response.Result;
import com.gongzone.central.point.domain.request.PointDTO;
import com.gongzone.central.point.payment.domain.Payment;
import com.gongzone.central.point.payment.service.PaymentService;
import io.swagger.v3.oas.annotations.Operation;
import io.swagger.v3.oas.annotations.Parameter;
import io.swagger.v3.oas.annotations.media.Content;
import io.swagger.v3.oas.annotations.responses.ApiResponse;
import io.swagger.v3.oas.annotations.responses.ApiResponses;
import lombok.RequiredArgsConstructor;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
@RestController
@RequiredArgsConstructor

```

```

@RequestMapping("/api/members")
public class PaymentController {
    private final PaymentService paymentService;
    @Operation(summary = "포인트 충전을 요청한다.")
    @ApiResponses(value = {
        @ApiResponse(responseCode = "200",
            description = "SUCCESS",
            content = @Content(mediaType = "application/json")),
        @ApiResponse(responseCode = "500",
            description = "FAILED_INTERNAL_ERROR"),
    })
    @PostMapping("/{memberNo}/point/payment")
    public ResponseEntity<Result> postMemberPointCharge(@Parameter(description = "회원 번호
(Mxxxxxx)")

        @PathVariable String memberNo,

        @Parameter(description = "포인트 충전 요청 객체(json)")

        @RequestBody PointDTO<Payment> request) {
        ResponseEntity<Result> response;
        try {
            paymentService.charge(memberNo, request);
            response = ResponseEntity.ok(new Result("SUCCESS"));
        } catch (RuntimeException e) {
            System.err.printf("Exception:\n\t%s\n", e);
            System.err.printf("\tCaused by: %s\n", e.getCause() != null ?
e.getCause().toString() : "null");
            response = ResponseEntity.internalServerError().body(new
Result("FAILED_INTERNAL_ERROR"));
        }
        return response;
    }
    @Operation(summary = "특정 결제 정보를 요청한다.")
    @ApiResponses(value = {
        @ApiResponse(responseCode = "200",
            description = "SUCCESS",
            content = @Content(mediaType = "application/json")),
        @ApiResponse(responseCode = "500",
            description = "FAILED_INTERNAL_ERROR"),
    })
    @GetMapping("/{memberNo}/point/payment/{paymentNo}")
    public ResponseEntity<Result> getPointPayment(@PathVariable String memberNo,

    @PathVariable String paymentNo) {
        ResponseEntity<Result> response;
        try {
            response = ResponseEntity.ok().body(new
Result(paymentService.get(paymentNo)));
        } catch (Exception e) {
            System.err.printf("Exception during getPointPayment: \n\t%s\n", e);
            System.err.printf("\tCaused by: %s\n", e.getCause() != null ?
e.getCause().toString() : "null");
            response = ResponseEntity.internalServerError().body(new

```

```

Result("FAILED_INTERNAL_ERROR");
    }
    return response;
}
@Operation(summary = "특정 결제 현황의 상세 정보를 요청한다.")
@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "SUCCESS",
        content = @Content(mediaType = "application/json")),
    @ApiResponse(responseCode = "500",
        description = "FAILED_INTERNAL_ERROR"),
})
@GetMapping("/{memberNo}/point/payment/{paymentNo}/detail")
public ResponseEntity<Result> getPointPaymentDetail(@PathVariable String memberNo,

    @PathVariable String paymentNo) {
    ResponseEntity<Result> response;
    try {
        response = ResponseEntity.ok().body(new
Result(paymentService.getDetail(paymentNo)));
    } catch (Exception e) {
        System.err.printf("Exception during getPointPaymentDetail: \n\t%s\n", e);
        System.err.printf("\tCaused by: %s\n", e.getCause() != null ?
e.getCause().toString() : "null");
        response = ResponseEntity.internalServerError().body(new
Result("FAILED_INTERNAL_ERROR"));
    }
    return response;
}
@Operation(summary = "회원의 결제 정보를 요청한다. ")
@ApiResponses(value = {
    @ApiResponse(responseCode = "200",
        description = "SUCCESS",
        content = @Content(mediaType = "application/json")),
    @ApiResponse(responseCode = "500",
        description = "FAILED_INTERNAL_ERROR"),
})
@GetMapping("/{memberNo}/point/payment")
public ResponseEntity<Result> getPointPayments(@PathVariable String memberNo,

@RequestParam(defaultValue = "10") int pageSize,

@RequestParam(defaultValue = "1") int page) {
    ResponseEntity<Result> response;
    try {
        response = ResponseEntity.ok().body(new
Result(paymentService.getMany(memberNo, pageSize, page)));
    } catch (Exception e) {
        System.err.printf("Exception during getPointPayments: \n\t%s\n", e);
        System.err.printf("\tCaused by: %s\n", e.getCause() != null ?
e.getCause().toString() : "null");
        response = ResponseEntity.internalServerError().body(new
Result("FAILED_INTERNAL_ERROR"));
    }
}

```

```
        return response;
    }
}
```

### Service

```
=====PaymentService.java=====
package com.gongzone.central.point.payment.service;
import com.gongzone.central.point.domain.request.PointDTO;
import com.gongzone.central.point.payment.domain.Payment;
import com.gongzone.central.point.payment.domain.detail.PaymentInfo;
import java.util.List;
public interface PaymentService {
    void insert(Payment payment);
    Payment get(String paymentNo);
    PaymentInfo getDetail(String paymentNo);
    List<Payment> getMany(String memberNo, int size, int page);
    void charge(String memberNo, PointDTO request);
}=====PaymentServiceImpl.java=====
package com.gongzone.central.point.payment.service;
import com.gongzone.central.point.domain.request.PointDTO;
import com.gongzone.central.point.mapper.PointMapper;
import com.gongzone.central.point.payment.api.PortOneClient;
import com.gongzone.central.point.payment.domain.Payment;
import com.gongzone.central.point.payment.domain.detail.PaymentInfo;
import com.gongzone.central.point.payment.mapper.PaymentMapper;
import com.gongzone.central.point.service.PointHistoryService;
import com.gongzone.central.point.service.PointService;
import java.util.List;
import lombok.RequiredArgsConstructor;
import org.springframework.stereotype.Service;
@Service
@RequiredArgsConstructor
public class PaymentServiceImpl implements PaymentService {
    private final PortOneClient portOneClient;
    private final PointService pointService;
    private final PointHistoryService pointHistoryService;
    private final PointMapper pointMapper;
    private final PaymentMapper paymentMapper;
    @Override
    public void insert(Payment payment) {
        paymentMapper.insert(payment);
    }
    @Override
    public Payment get(String paymentNo) {
        return paymentMapper.get(paymentNo);
    }
    @Override
    public PaymentInfo getDetail(String paymentNo) {
        String paymentId = get(paymentNo).getPaymentId();
        return portOneClient.getPaymentInfo(paymentId).block();
    }
    @Override
    public List<Payment> getMany(String memberNo, int size, int page) {
        String memberPointNo = pointMapper.getMemberPointNo(memberNo);
```

```

        return paymentMapper.getMany(memberPointNo, size, page - 1);
    }
    /**
     * 요청을 기반으로 회원 포인트 충전을 처리한다.
     *
     * @param memberNo 회원 번호
     * @param request 포인트 충전 객체
     */
    @Override
    public void charge(String memberNo, PointDTO request) {
        // 1. 포인트 내역 삽입
        String historyNo = pointHistoryService.insert(memberNo, request);
        // 1-1. 충전 내역 생성
        Payment payment = (Payment) request.getDetail();
        payment.setPointHistoryNo(historyNo);
        // 충전 정보 유효할 시
        // 2. 충전 내역 삽입
        insert(payment);
        // 3. 포인트 증가
        pointService.updatePoint(memberNo, request);
        // 4. 포인트 내역 업데이트(성공)
        pointHistoryService.updateSuccess(historyNo, request);
    }
}

```

Mapper

```

=====PointMapper.xml=====
package com.gongzone.central.point.payment.mapper;
import com.gongzone.central.point.payment.domain.Payment;
import java.util.List;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
@Mapper
public interface PaymentMapper {
    void insert(Payment payment);
    Payment get(String paymentNo);
    List<Payment> getMany(@Param("memberPointNo") String memberPointNo,
                          @Param("size") int size,
                          @Param("page") int page);
}=====PointMapper.xml=====
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.gongzone.central.point.payment.mapper.PaymentMapper">
    <!-- Result Map -->
    <resultMap id="PaymentResultMap" type="Payment">
        <id property="paymentNo" column="pay_no" />
        <result property="pointHistoryNo" column="point_history_no" />
        <result property="type" column="pay_type" />
        <result property="transactionType" column="pay_tx_type" />
        <result property="txId" column="pay_tx_id" />
        <result property="paymentId" column="pay_id" />
    </resultMap>

```

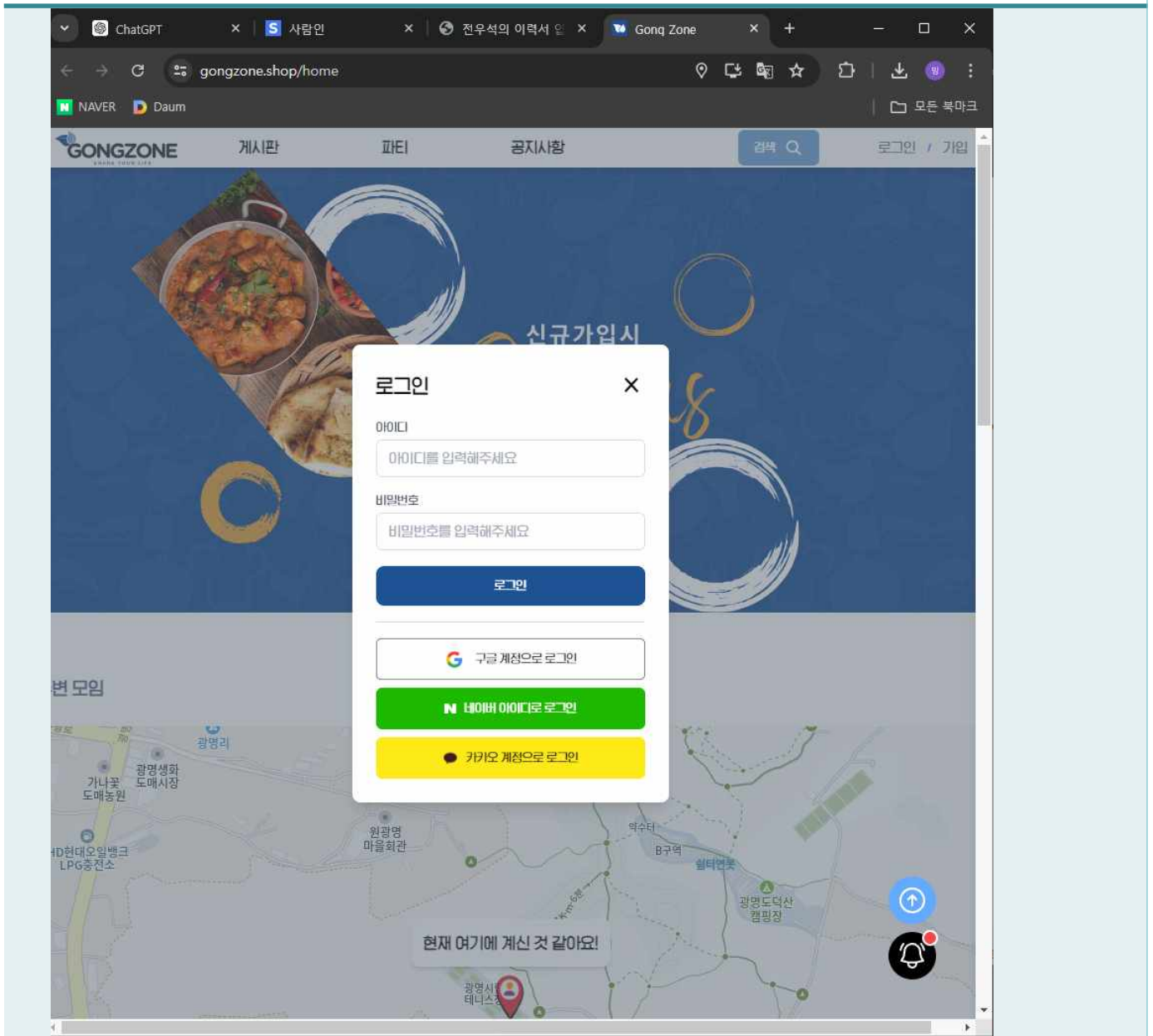
```

<result property="code" column="pay_code" />
<result property="message" column="pay_message" />
<result property="status" column="pay_date" />
</resultMap>
<!-- Query -->
<insert id="insert">
  -- insert: 포인트 충전 정보를 삽입한다.
  INSERT INTO point_payment
    (point_history_no, pay_type, pay_tx_type, pay_tx_id, pay_id, pay_code, pay_message, status_code)
  VALUES
    ({pointHistoryNo}, #{type}, #{transactionType}, {txId}, #{paymentId}, #{code}, {message},
#{status});
</insert>
<select id="get" parameterType="String" resultMap="PaymentResultMap">
  -- getMany: 특정 포인트 충전 내역을 반환한다.
  SELECT
    pay_no,
    point_history_no,
    pay_type,
    pay_tx_type,
    pay_tx_id,
    pay_id,
    pay_code,
    pay_message,
    pay_date,
    status_code
  FROM
    point_payment
  WHERE
    pay_no = #{paymentNo};
</select>
<select id="getMany" resultMap="PaymentResultMap">
  -- getMany: 회원의 포인트 충전 내역을 반환한다.
  SELECT
    p.pay_no,
    p.point_history_no,
    p.pay_type,
    p.pay_tx_type,
    p.pay_tx_id,
    p.pay_id,
    p.pay_code,
    p.pay_message,
    p.pay_date,
    p.status_code
  FROM
    point_payment AS p
  LEFT JOIN
    point_history AS h
    ON p.point_history_no = h.point_history_no
  WHERE
    m_point_no = #{memberPointNo}
  ORDER BY
    pay_no DESC
  LIMIT #{size} OFFSET #{page};

```

```
</select>
<select id="getTransactionId" parameterType="String" resultMap="PaymentResultMap">
  -- getTransactionId: 특정 결제건의 결제 서버 id를 반환한다.
  SELECT
    pay_tx_id
  FROM
    point_payment
  WHERE
    pay_no = #{payNo};
</select>
</mapper>
```

## 로그인



### jsx 파일

```
import { useState } from "react";
import AuthStore from "../../utils/zustand/AuthStore";

import Naver from "../../assets/logo/ic-sns-naver.png";
import Kk from "../../assets/logo/ic-sns-kk.png";
import Google from "../../assets/logo/ic-sns-google.png";

export default function ModalLogin({ isOpen, onClose }) {
  const [loginId, setloginId] = useState("");
  const [loginPw, setloginPw] = useState("");
```



```

const [errorMessage, setErrorMessage] = useState("");
const { statusLogin } = AuthStore();

if (!isOpen) return null;

const handleContinue = async (e) => {
  e.preventDefault();
  const loginRequest = { loginId, loginPw };
  const response = await statusLogin(loginRequest);
  console.log(loginRequest);

  if (response && response.error) {
    setErrorMessage(response.error);
  } else {
    setErrorMessage("");
    onClose();
  }
};

const handleGoogleLogin = () => {
  const clientId =
    "901670599809-s8vu30qhb5hba1r856uvj4bulra76d0s.apps.googleusercontent.com";
  const redirectUri = encodeURI("https://www.gongzone.shop/google/callback");
  const state = Math.random().toString(36).substr(2, 11);
  window.location.href =
`https://accounts.google.com/o/oauth2/v2/auth?response_type=code&client_id=${clientId}&redirect_uri=${
redirectUri}&scope=profile email&state=${state}`;
};

const handleNaverLogin = () => {
  const clientId = "ViZy5l34ZQtLQtRQEPm0";
  const redirectUri = encodeURI("https://www.gongzone.shop/naver/callback");
  const state = Math.random().toString(36).substr(2, 11);
  window.location.href =
`https://nid.naver.com/oauth2.0/authorize?response_type=code&client_id=${clientId}&redirect_uri=${redire
ctUri}&state=${state}`;
};

const handleKakaoLogin = () => {
  //여기서 상태 검증
  const clientId = "526e8f945f81281052ccfce1af49781c";
  const redirectUri = encodeURI("https://www.gongzone.shop/kakao/callback");
  const state = Math.random().toString(36).substr(2, 11);
  window.location.href =
`https://kauth.kakao.com/oauth/authorize?response_type=code&client_id=${clientId}&redirect_uri=${redir
ectUri}&scope=profile_nickname,account_email&state=${state}`;
};

return (
  <div className="fixed inset-0 bg-gray-600 bg-opacity-50 flex justify-center items-center z-50">
    <div className="bg-white p-6 rounded-lg shadow-lg w-80">
      <div className="flex justify-between items-center">
        <h2 className="text-xl font-bold">로그인</h2>
        <button onClick={onClose} className="text-black text-3xl">

```

```

        &times;
      </button>
    </div>
    <form onSubmit={handleContinue} className="mt-4 space-y-4">
      <div>
        <label
          htmlFor="loginId"
          className="block text-xs font-medium text-gray-700"
        >
          아이디
        </label>
        <input
          type="text"
          id="loginId"
          placeholder="아이디를 입력해주세요"
          value={loginId}
          onChange={(e) => setloginId(e.target.value)}
          className="mt-1 block text-sm w-full px-3 py-2 border border-gray-300 rounded-md
shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500"
          required
        />
      </div>
      <div>
        <label
          htmlFor="loginPw"
          className="block text-xs font-medium text-gray-700"
        >
          비밀번호
        </label>
        <input
          type="password"
          id="loginPw"
          placeholder="비밀번호를 입력해주세요"
          value={loginPw}
          onChange={(e) => setloginPw(e.target.value)}
          className="mt-1 text-sm block w-full px-3 py-2 border border-gray-300 rounded-md
shadow-sm focus:outline-none focus:ring-blue-500 focus:border-blue-500"
          required
        />
      </div>
      {errorMessage && (
        <div className="text-red-500 text-sm mt-2">{errorMessage}</div>
      )}
      <div className="flex justify-between mt-4">
        <button type="submit" className="bg-[#1D5091] w-full rounded-lg text-white
hover:bg-opacity-90 text-xs py-3">
          로그인
        </button>
      </div>
    </form>

    <div className="space-y-2 mt-4">

```

```

    <div className="bg-gray-300 w-full h-[1px] mb-4"></div>
    <button
      className="h-[42px] w-full text-black py-2 gap-2 rounded-md flex items-center
justify-center text-xs shadow-[inset_0_0_0.5px_black] hover:bg-opacity-90"
      onClick={handleGoogleLogin}
    >
      <img src={Google} alt="Google Logo" className="w-6"/>
      구글 계정으로 로그인
    </button>
    <button
      className="h-[42px] w-full text-white bg-[#1DB400] py-2 gap-2 rounded-md flex
items-center justify-center text-xs hover:bg-opacity-90"
      onClick={handleNaverLogin}
    >
      <img src={Naver} alt="Naver Logo" className="w-3"/>
      네이버 아이디로 로그인
    </button>
    <button
      className="h-[42px] w-full text-black bg-[#F7E318] py-2 gap-2 rounded-md flex
items-center justify-center text-xs hover:bg-opacity-90"
      onClick={handleKakaoLogin}
    >
      <img src={Kk} alt="Kakao Logo" className="w-4"/>
      카카오키계정으로 로그인
    </button>
  </div>
</div>
</div>
);
}

```

## Service

```

=====MemberDetailsService.java=====
package com.gongzone.central.member.login.service;

import com.gongzone.central.member.domain.Member;
import com.gongzone.central.member.domain.MemberLevel;
import com.gongzone.central.member.login.mapper.LoginMapper;
import com.gongzone.central.point.domain.Point;
import com.gongzone.central.point.service.PointService;
import lombok.RequiredArgsConstructor;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

@Service

```

```

@Transactional
@RequiredArgsConstructor
public class MemberDetailsService implements UserDetailsService {

    private final PointService pointService;

    private final LoginMapper loginMapper;

    @Override
    public UserDetails loadUserByUsername(String identifier) throws UsernameNotFoundException
    {
        Member member;
        if (identifier.contains("@")) {
            member = loginMapper.getMemberByEmail(identifier);
        } else if (identifier.startsWith("M")) {
            member = loginMapper.getMemberByNo(identifier);
        } else {
            member = loginMapper.getMemberById(identifier);
        }

        if (member == null) {
            throw new UsernameNotFoundException("User not found");
        }
        int memberLevelValue = member.getMemberLevel();
        if (memberLevelValue == 0) {
            throw new IllegalArgumentException("Invalid member level: " +
memberLevelValue);
        }

        MemberLevel level = MemberLevel.fromLevel(memberLevelValue);

        Point pointNo = pointService.getPoint(member.getMemberNo());
        if (pointNo == null) {
            throw new IllegalArgumentException("Point information not found for
memberNo: " + member.getMemberNo());
        }

        return new MemberDetails(member, pointNo);
    }
}

```

## Mapper

```

=====LoginMapper.java=====
package com.gongzone.central.member.login.mapper;

import com.gongzone.central.member.domain.Member;
import com.gongzone.central.member.login.domain.LoginLog;
import com.gongzone.central.member.login.domain.LoginStatistical;
import org.apache.ibatis.annotations.Mapper;

import java.util.List;

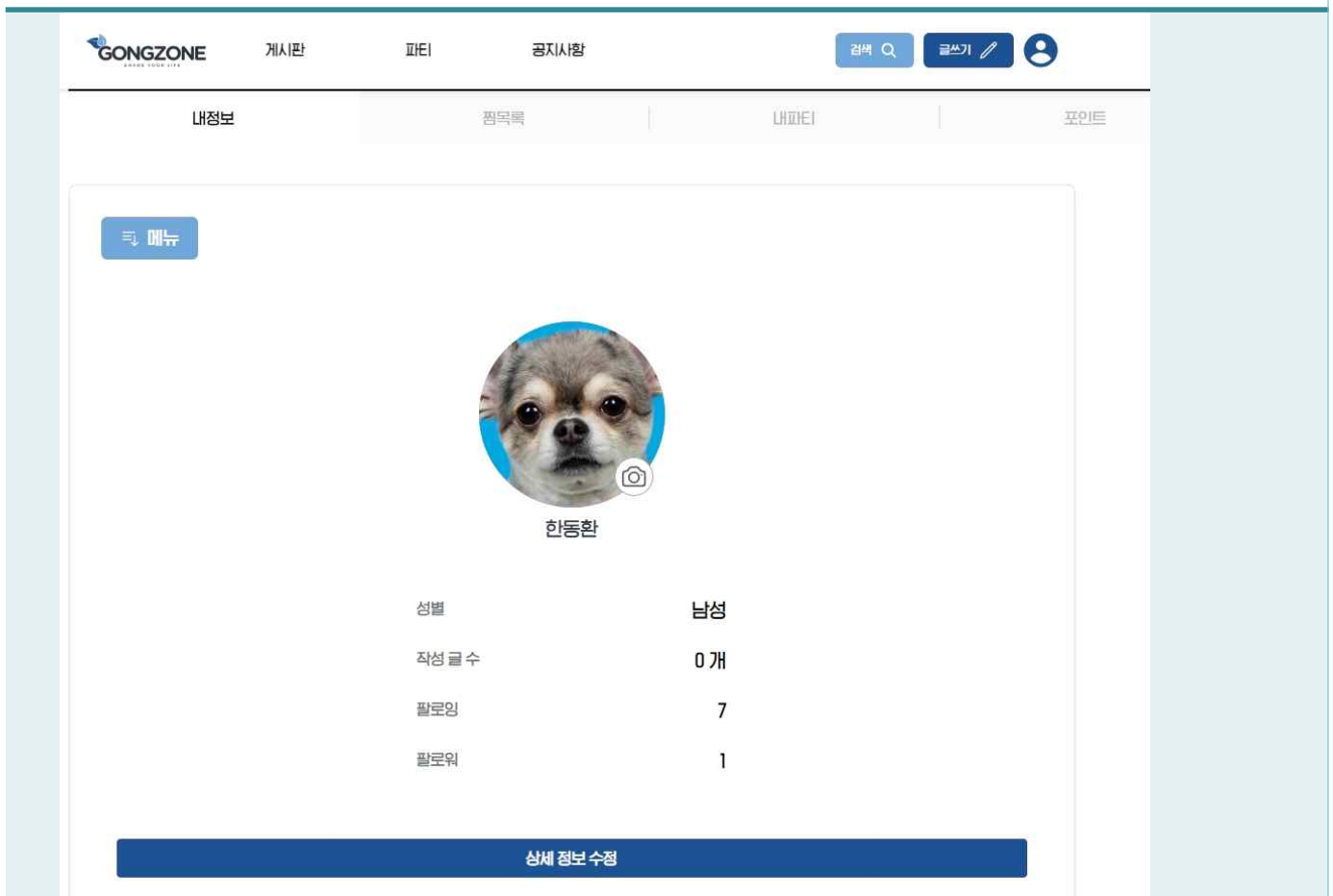
```

```
@Mapper
public interface LoginMapper {
    Member getMemberById(String loginId);

    Member getMemberByEmail(String email);

    Member getMemberByNo(String memberNo);
}
```

## 마이페이지



### jsx 파일

```
import React, { useState, useEffect, useRef } from "react";
import { Link, useNavigate } from "react-router-dom";
import FollowList from "./FollowList";
import BlockList from "./BlockList";
import { BarsArrowDownIcon } from "@heroicons/react/24/solid";
import Profile from "../../components/page/myInfo/Profile";

export default function MyInfo({ memberNo, onNextPage }) {
  const navigate = useNavigate();
  const tabItems = [
    { id: "profile", label: "프로필" },
    { id: "follow", label: "팔로우 목록" },
    { id: "block", label: "차단 목록" },
    { id: "all", label: "전체 회원 보기" },
  ];

  const [activeTab, setActiveTab] = useState("profile");
  const [isDropdownOpen, setIsDropdownOpen] = useState(false);
  const dropdownRef = useRef(null);
```

```

const handleTabClick = (id) => {
  setActiveTab(id);
  setIsDropdownOpen(false); // 드롭다운 닫기
  if (id === "all") {
    navigate("/member/list"); // 전체 회원 보기 페이지로 이동
  }
};

const toggleDropdown = () => {
  setIsDropdownOpen(!isDropdownOpen);
};

const handleClickOutside = (event) => {
  if (dropdownRef.current && !dropdownRef.current.contains(event.target)) {
    setIsDropdownOpen(false);
  }
};

useEffect(() => {
  document.addEventListener("mousedown", handleClickOutside);
  return () => {
    document.removeEventListener("mousedown", handleClickOutside);
  };
}, []);

return (
  <div className="lg:w-[65em] w-[90%] mb-10 border-gray-200 border-2 py-8 px-4 sm:px-8 rounded-lg shadow-lg mx-auto">
    <div className="relative flex justify-start">
      <button
        onClick={toggleDropdown}
        className={`gap-2 text-lg font-bold py-2 px-5 rounded-md flex items-center bg-[#6ea2d4] hover:bg-[#1d5091] text-white`}
      >
        <BarsArrowDownIcon className="w-5 h-5" /> 메뉴
      </button>
      {isDropdownOpen && (
        <div
          ref={dropdownRef}
          className="absolute top-12 mt-2 w-40 rounded-md shadow-lg bg-white ring-1 ring-black ring-opacity-5 z-40"
        >
          {tabItems.map((item) => (
            <button
              key={item.id}
              onClick={() => handleTabClick(item.id)}
              className="block w-full text-sm text-left px-4 py-2 hover:bg-gray-200"
            >
              {item.label}
            </button>
          ))}
        </div>
      )}
    </div>
  </div>
);

```

```

<div className="p-4">
  {activeTab === "profile" && (
    <div className="flex flex-col items-center mt-12">
      <div className="w-[90%] md:w-[60%] mb-4">
        <Profile memberNo={memberNo} />
      </div>
      <div className="border-t-2 border-gray-400 mb-12" />
      <button
        onClick={onNextPage}
        className="bg-[#1d5091] hover:bg-[#0d2544] text-white font-bold py-2 w-full px-4
rounded"
      >
        상세 정보 수정
      </button>
    </div>
  )}
  {activeTab === "follow" && <FollowList memberNo={memberNo} />}
  {activeTab === "block" && <BlockList memberNo={memberNo} />}
</div>
</div>
);
}

```

## Controller

```

package com.gongzone.central.member.myInfo.controller;

import com.gongzone.central.member.domain.Member;
import com.gongzone.central.member.login.security.JwtUtil;
import com.gongzone.central.member.login.service.MemberDetails;
import com.gongzone.central.member.myInfo.domain.MyInformation;
import com.gongzone.central.member.myInfo.service.MyInfoService;
import com.gongzone.central.member.service.MemberService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.security.core.Authentication;

import java.util.HashMap;
import java.util.Map;

@RestController
@RequestMapping("/api/members")
public class MyInfoController {
    private final MyInfoService myInfoService;
    private final MemberService memberService;
    private final JwtUtil jwtUtil;

    @Autowired
    public MyInfoController(MyInfoService myInfoService, MemberService memberService, JwtUtil
jwtUtil) {
        this.myInfoService = myInfoService;
        this.memberService = memberService;
    }
}

```



```

        this.jwtUtil = jwtUtil;
    }

    /**
     * @제목 : 사용자 세션 값 관리
     * @작성일: 2024-06-19
     * @수정일: 2024-06-19
     * @내용: 세션의 존재에 따라 1 또는 0반환 후 로직 처리 기대
     */
    /*
    @GetMapping("/{memberNo}/session")
    public String getSessionMember(@PathVariable String memberNo) {
        Member member = myInfoService.findByNo(memberNo);
        return (member != null) ? "1" : "0";
    }
    */

    /**
     * @제목: 특정회원 정보 토큰
     * @작성일: 2024-06-19
     * @수정일: 2024-06-19
     * @내용: 로그인 된 특정회원 정보 return
     */
    @GetMapping("/{memberNo}/memberInfo")
    public ResponseEntity<Member> getMemberInfo(@PathVariable String memberNo, Authentication
authentication) {
        String token = ((MemberDetails) authentication.getPrincipal()).getToken();
        String extractedMemberNo = jwtUtil.extractMemberNo(token);

        if (!extractedMemberNo.equals(memberNo)) {
            return ResponseEntity.status(403).build();
        }

        Member member = myInfoService.findByNo(memberNo);
        if (member != null) {
            return ResponseEntity.ok(member);
        } else {
            throw new RuntimeException("회원 정보를 가져올 수 없습니다.");
        }
    }

    // 2024-06-26 JWT토큰 유효성 검증 로직 추가
    @PostMapping("/{memberNo}/password")
    public ResponseEntity<String> updatePassword(@PathVariable String memberNo, @RequestBody
MyInformation myInformation, Authentication authentication) {
        String token = ((MemberDetails) authentication.getPrincipal()).getToken();
        String extractedMemberNo = jwtUtil.extractMemberNo(token);

        if(!extractedMemberNo.equals(memberNo)) {
            return ResponseEntity.status(403).body("유효 토큰이 아닙니다.");
        }

        Member member = myInfoService.findByNo(memberNo);
        if (member == null) {

```

```

        return ResponseEntity.status(404).body("로그인된 사용자가 없습니다.");
    }
    try {
        myInfoService.updatePassword(member, myInformation);
        return ResponseEntity.ok("비밀번호 변경 성공.");
    } catch (RuntimeException e) {
        return ResponseEntity.status(400).body(e.getMessage());
    }
}

@GetMapping("/{memberNo}/nickname")
public ResponseEntity<Map<String, String>> getNickname(@PathVariable String memberNo,
Authentication authentication) {
    System.out.println("getNickname 호출");
    System.out.println("memberNo: " + memberNo);

    MemberDetails memberDetails = (MemberDetails) authentication.getPrincipal();
    String token = memberDetails.getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    System.out.println("토큰에 담긴 멤버 정보: " + extractedMemberNo);

    if (!extractedMemberNo.equals(memberNo)) {
        System.out.println("인증 실패");
        return ResponseEntity.status(403).build();
    }
    Member member = myInfoService.findByNo(memberNo);
    if(member != null) {
        Map<String, String> response = new HashMap<>();
        response.put("nickname", member.getMemberNick());
        System.out.println("닉네임: " + member.getMemberNick()); // 테스트 후 삭제
        return ResponseEntity.ok(response);
    } else {
        System.out.println("회원 정보 없음");
        return ResponseEntity.status(404).build(); // 정보 없음 404에러
    }
}

@PostMapping("/{memberNo}/nickname")
public ResponseEntity<Void> updateNickname(@PathVariable String memberNo, @RequestBody
MyInformation myInformation, Authentication authentication) {
    System.out.println("updateNickname 메서드 호출");
    System.out.println("memberNo: " + memberNo);
    System.out.println("newMemberNick: " + myInformation.getNewMemberNick());

    MemberDetails memberDetails = (MemberDetails) authentication.getPrincipal();
    String token = memberDetails.getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    System.out.println("토큰정보: " + extractedMemberNo);

    if (!extractedMemberNo.equals(memberNo)) {
        System.out.println("인증 실패");
        return ResponseEntity.status(403).build();
    }
}

```

```

    }

    Member member = myInfoService.findByNo(memberNo);
    if (member != null) {
        System.out.println("회원 정보 존재");
    } else {
        System.out.println("회원 정보 없음");
    }

    myInfoService.updateMemberNick(member, myInformation);
    System.out.println("닉네임 수정 완료");
    return ResponseEntity.ok().build();
}

@GetMapping("/{memberNo}/locations")
public ResponseEntity<Map<String, String>> getAddress(@PathVariable String memberNo,
Authentication authentication) {
    System.out.println("장소 메서드 호출 (조회)");
    MemberDetails memberDetails = (MemberDetails) authentication.getPrincipal();
    String token = memberDetails.getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    if (!extractedMemberNo.equals(memberNo)) {
        System.out.println("장소 조회 중 인증 실패");
        return ResponseEntity.status(403).build();
    }

    Member member = myInfoService.findByNo(memberNo);
    if (member != null) {
        Map<String, String> response = new HashMap<>();
        String address = member.getMemberAddress();
        if(address == null || address.trim().isEmpty()) {
            System.out.println("저장된 주소가 없습니다.");
            response.put("memberAddress", "");
        } else {
            response.put("memberAddress", address);
        }

        System.out.println("주소: " + address);
        return ResponseEntity.ok(response);
    } else {
        return ResponseEntity.status(404).build();
    }
}

@PostMapping("/{memberNo}/locations")
public ResponseEntity<Void> updateMemberAddress(@PathVariable String memberNo,
@RequestBody MyInformation myInformation, Authentication authentication) {
    System.out.println("장소 메서드 호출(수정)");
    MemberDetails memberDetails = (MemberDetails) authentication.getPrincipal();
    String token = memberDetails.getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    if (!extractedMemberNo.equals(memberNo)) {

```

```

        System.out.println("장소수정중 403에러 처리");
        return ResponseEntity.status(403).build();
    }
    Member member = myInfoService.findByNo(memberNo);
    if (member == null) {
        System.out.println("장소수정중 404에러 처리");
        return ResponseEntity.status(404).build();
    }
    myInfoService.updateMemberAddress(member,myInformation);
    return ResponseEntity.ok().build();
}

/**
 * @제목: 핸드폰 번호 조회
 * @작성일: 2024-06-28
 * @수정일: 2024-06-28
 */
@GetMapping("/{memberNo}/phone")
public ResponseEntity<Map<String, String>> getMemberPhone(@PathVariable String memberNo,
Authentication authentication) {
    System.out.println("핸드폰번호 조회");
    MemberDetails memberDetails = (MemberDetails) authentication.getPrincipal();
    String token = memberDetails.getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    if (!extractedMemberNo.equals(memberNo)) {
        System.out.println("핸드폰 번호 조회 실패");
        return ResponseEntity.status(403).build();
    }
    Member member = myInfoService.findByPhone(memberNo);
    if (member != null) {
        Map<String,String> response = new HashMap<>();
        response.put("phone", member.getMemberPhone());
        return ResponseEntity.ok(response);
    } else {
        System.out.println("해당 회원이 없습니다.");
        return ResponseEntity.status(404).build();
    }
}

@PostMapping("/{memberNo}/updateStatus")
public ResponseEntity<String> updateStatus(@PathVariable String memberNo, @RequestBody
MyInformation myInformation, Authentication authentication) {
    String token = ((MemberDetails) authentication.getPrincipal()).getToken();
    String extractedMemberNo = jwtUtil.extractMemberNo(token);

    if (!extractedMemberNo.equals(memberNo)) {
        return ResponseEntity.status(403).body("유효 토큰이 아닙니다.");
    }

    myInfoService.updateStatusCode(memberNo, myInformation.getNewStatusCode());
    return ResponseEntity.ok("상태 코드 변경 성공.");
}
}

```

## Domain

```
package com.gongzone.central.member.myInfo.domain;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

/**
 * @제목: 내정보 설정
 * @생성일: 2024-06-18
 * @수정일: 2024-06-18
 * @내용: 변경사항들에 대한 값이므로 final선언 안했지만 필요에 따라 선언 가능
 */
@Data
@NoArgsConstructor
@AllArgsConstructor
public class MyInformation {
    private String newPassword; // null 가능
    private String newMemberNick; // null 가능
    private String newMemberAddress; // null 가능

    private String statusCode;
    private String newStatusCode;
}
```

## Mapper

```
package com.gongzone.central.member.myInfo.mapper;

import com.gongzone.central.member.domain.Member;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;

import java.util.Map;

/**
 * @제목: 비밀번호 맵핑
 * @생성일: 2024-06-18
 * @수정일: 2024-07-02
 * @내용: 닉네임 업데이트 시 받아오는 데이터 형태에 맞춰 타입 변경
 */
@Mapper
public interface MyInfoMapper {

    Member findByNo(@Param("memberNo") String memberNo);
    Member findByNickname(@Param("memberNick") String memberNick);
    Member findByAddress(@Param("memberAddress") String memberAddress);
    Member findByPhone(@Param("memberNo") String memberNo);

    void updatePassword(@Param("memberNo") String memberNo, @Param("newPassword") String newPassword);
}
```

```

void updateMemberNick(Map<String, Object> params);
void          updateMemberAddress(@Param("memberNo")          String          memberNo,
@Param("newMemberAddress") String newMemberAddress);
void updateStatusCode(Map<String, Object> params);
}

```

```

-----
-----
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
    PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!-- /**
 * @제목: 유저 정보 매퍼 xml
 * @생성일: 2024-06-18
 * @수정일: 2024-06-28
 * @내용: member의 기본키 memberNo를 유저 정보 조회 및 업데이트
 */-->
<mapper namespace="com.gongzone.central.member.myInfo.mapper.MyInfoMapper">

    <!-- Result Mapping -->
    <resultMap id="MyInfoResultMap" type="com.gongzone.central.member.domain.Member">
        <id property="memberNo" column="m_no" />
        <result property="memberId" column="m_id" />
        <result property="memberPw" column="m_pw" />
        <result property="memberPhone" column="m_phone" />
        <result property="memberAddress" column="m_addr" />
        <result property="memberNick" column="m_nick" />
        <result property="memberStatus" column="status_code" />
    </resultMap>

    <!-- Read -->
    <select id="findByNo" parameterType="String" resultMap="MyInfoResultMap">
        SELECT m_no,m_id, m_pw, m_phone, m_addr, m_nick
        FROM member
        WHERE m_no = #{memberNo}
    </select>

    <select id="findByNickname" parameterType="String" resultMap="MyInfoResultMap">
        SELECT m_nick, m_no FROM member WHERE m_nick = #{memberNick}
    </select>

    <select id="findByAddress" parameterType="String" resultMap="MyInfoResultMap">
        SELECT m_no, m_addr FROM member WHERE m_addr = #{memberAddress}
    </select>

    <select id="findByPhone" parameterType="String" resultMap="MyInfoResultMap">
        SELECT m_no, m_phone FROM member WHERE m_no = #{memberNo}
    </select>

```

```
<!-- Update -->
<update id="updatePassword" parameterType="map">
    UPDATE member SET m_pw = #{newPassword}
    WHERE m_no = #{memberNo}
</update>

<update id="updateMemberNick" parameterType="map">
    UPDATE member SET m_nick = #{newMemberNick}
    WHERE m_no = #{memberNo}
</update>

<update id="updateMemberAddress" parameterType="map">
    UPDATE member SET m_addr = #{newMemberAddress}
    WHERE m_no = #{memberNo}
</update>

<update id="updateStatusCode" parameterType="map">
    UPDATE member SET status_code = #{newStatusCode}
    WHERE m_no = #{memberNo}
</update>
</mapper>
```