

API 명세서

기본 정보

- Base URL: `http://localhost:8080/api`
- 모든 응답은 JSON 입니다.
- JWT 인증(Authorization 헤더)에 따라 `@AuthenticationPrincipal` 또는 `userId` 쿼리/바디로 사용자 식별

1. 인증 (Auth)

Method	Path	설명	Request	Response
POST	<code>/auth/login</code>	로그인 → 액세스/리프레시 토큰	<code>{ "email": "...", "password": "..." }</code>	<code>{ "accessToken": "..." }</code>
POST	<code>/auth/refresh</code>	리프레시 토큰으로 재발급	(쿠키 <code>refreshToken</code>)	<code>{ "accessToken": "..." }</code>
POST	<code>/auth/logout</code>	로그아웃 → 쿠키 삭제	—	No Content (204)

2. 사용자 (Users)

Method	Path	설명	Request	Response
POST	<code>/users/signup</code>	회원가입	<code>{ "email","password","name",... }</code>	"회원가입 완료"
GET	<code>/users/check-email?email=</code>	이메일 중복 체크	—	<code>{ "available": true }</code>
POST	<code>/users/email/send</code>	인증코드 발송	<code>{ "email": "... " }</code>	No Content
POST	<code>/users/email/verify</code>	인증코드 검증	<code>{ "email":"...", "code":"1234" }</code>	<code>{ "verified": true }</code>
POST	<code>/users/find-email/send</code>	(아이디찾기) 인증코드 발송	<code>{ "name":"...", "phoneNumber":"...", "email":"..." }</code>	No Content
POST	<code>/users/find-email/verify</code>	(아이디찾기) 검증 후 이메일 반환	<code>{ "name":"...", "phoneNumber":"...", "email":"...", "code":"..." }</code>	<code>{ "email":"...", "success": true }</code>
POST	<code>/users/find-password</code>	임시 비밀번호 발송	<code>{ "email": "... " }</code>	"임시 비밀번호 발송 되었습니다."
POST	<code>/users/verify-password</code>	비밀번호 확인	<code>{ "password":"..." }</code> (토큰 기반 이메일)	"비밀번호 인증 완료"
GET	<code>/users/me</code>	내 프로필 조회	(Bearer 토큰)	<code>{ userId, email, name, ... }</code>
PUT	<code>/users/me</code>	내 프로필 수정	<code>{ "name":"...", "phoneNumber":"...",... }</code>	No Content
POST	<code>/users/delete-user</code>	회원 탈퇴	<code>{ "password":"..." }</code>	"회원 삭제 완료"

3. 마이페이지 (MyPage)

Method	Path	설명	Request	Response
GET	<code>/mypage/drawings</code>	내 그림 목록 조회	(Bearer 토큰)	<code>[{ drawingId, originalUrl,... }]</code>
DELETE	<code>/mypage/drawings/{drawingId}</code>	내 그림 삭제	(Bearer 토큰)	No Content

GET	/mypage/trip-plans?userId={userId}	내 여행 계획 조회	—	[{ planBoardId, route,... }]
DELETE	/mypage/trip-plans/{planCode}?userId={userId}	내 여행 계획 삭제	—	No Content

4. 여행 게시판 (Trip Boards)

4.1. 게시물 CRUD

Method	Path	설명	Request Body	Response
POST	/trip/boards	게시글 생성	{ userId, title, content, planCode,... }	201 Created + 생성된 boardId
GET	/trip/boards/{id}?userId={uid}	게시글 단건 조회	—	{ planBoardId,..., favorite: true }
GET	/trip/boards?userId={uid}	게시글 리스트	—	[{ planBoardId,..., favorite }]
PUT	/trip/boards/{id}	게시글 수정	{ title, content,... }	No Content
DELETE	/trip/boards/{id}	게시글 삭제	—	No Content

4.2. 댓글 (Comments)

Method	Path	설명	Request Body	Response
POST	/trip/boards/{boardId}/comments	댓글 작성	{ userId, content }	201 Created + commentId
GET	/trip/boards/{boardId}/comments	댓글 조회	—	[{ commentId,... }]
PUT	/trip/boards/{boardId}/comments/{cId}	댓글 수정	{ content }	No Content
DELETE	/trip/boards/{boardId}/comments/{cId}	댓글 삭제	—	No Content

4.3. 즐겨찾기 (Favorites)

Method	Path	설명	Request	Response
POST	/trip/boards/{id}/favorite?userId={userId}	즐거찾기 추가	—	200 OK
DELETE	/trip/boards/{id}/favorite?userId={userId}	즐거찾기 해제	—	204 No Content

4.4. 상세 + 일차별 장소

Method	Path	설명	Request	Response
GET	/trip/boards/{id}/detail-with-locations?userId={userId}	메타 + 모든 장소	—	{ boardMeta,..., locationsByDay: {1:[...],2:[...]} }

5. 여행 계획 (Trip Plans & Locations)

5.1. AI 일정 생성

Method	Path	설명	Request Body	Response
POST	/itinerary	GPT 일정 생성	{ "places": ["Paris"], "days": 3 }	[[{ name,... }], [...], ...]
POST	/itinerary/update-times	시간 재계산	{ "itinerary": [[{ name,... }], ...] }	같은 형태의 일정 배열

5.2. 주변 장소 추천

Method	Path	설명	Query	Response
GET	/nearby?lat={lat}&lng={lng}	주변 장소 10개 보장	—	{ tours:[...], restaurants:[...], cafes:[...], ... }

5.3. 계획 저장 & 조회

Method	Path	설명	Request Body	Response
POST	/plans-with-locations	계획 + 장소 저장	{ userId, region, locations:[{dayNo, addressName, addressCategory,...}] }	"PLAN_CODE_ABC123"
GET	/plans-with-locations/{planCode}	계획 + 장소 불러오기	—	{ dailyPlans:[{dayNo, activities:[...]}] }

6. 자유 게시판 (Free Board)

6.1. 글 & 이미지

Method	Path	설명	Request	Response
POST	/free/createFree	글 + 이미지 업로드	multipart/form-data userId, title, content, images[]=File...	freeld
GET	/free/{freeld}	글 조회	—	{ freeld, title,..., images:[...] }
GET	/free	글 리스트	—	[{ freeld,... }]
PUT	/free/{freeld}	글 수정	{ title, content,... }	No Content
DELETE	/free/{freeld}	글 삭제	—	No Content

6.2. 댓글 (FreeComment)

Method	Path	설명	Request	Response
GET	/free/{freeld}/comment	댓글 조회	—	[{ commentId,... }]
POST	/free/{freeld}/comment?userId=	댓글 작성	{ content:"..." }	{ commentId,... }
PUT	/free/{freeld}/comment/{cld}	댓글 수정	{ content:"..." }	No Content
DELETE	/free/{freeld}/comment/{cld}	댓글 삭제	—	No Content

6.3. 리액션 (Reaction)

Method	Path	설명	Query	Response
POST	/free/{freeld}/reaction	좋아요/싫어요	{type, userId}	—

7. 그림 게시판 (Drawing Board)

Method	Path	설명	Request	Response
GET	/drawingboard/popular	이번 주 인기	(Bearer 토큰)	[DrawingBoardDto,...]
GET	/drawingboard?page={n}	페이지별 리스트	(Bearer 토큰)	[DrawingBoardDto,...]
GET	/drawingboard/{id}	상세 조회	(Bearer 토큰)	DrawingBoardDto

POST	/drawingboard	작성	{ userId, title, content, originalUrl,... }	DrawingBoardDto
PUT	/drawingboard/{id}	수정	{ title, content,... }	No Content
DELETE	/drawingboard/{id}	삭제	—	No Content
POST	/drawingboard/{id}/like	좋아요	(Bearer 토큰)	No Content
DELETE	/drawingboard/{id}/like	좋아요 취소	(Bearer 토큰)	No Content

8. 공지사항 (Notice)

Method	Path	설명	Request	Response
POST	/notice	생성	{ title, content, isPinned }	No Content
GET	/notice/{noticeId}	단건 조회	—	NoticeDto
GET	/notice	리스트	—	[NoticeDto,...]
PUT	/notice/{noticeId}	수정	{ title, content, isPinned }	No Content
DELETE	/notice/{noticeId}	삭제	—	No Content

9. 문의 (Contact)

Method	Path	설명	Request	Response
POST	/contact	글 작성	{ name, email, content }	contactId
GET	/contact/{contactId}	단건 조회	—	ContactDto
GET	/contact	리스트	—	[ContactDto,...]
PUT	/contact/{contactId}	수정	{ name, email, content }	No Content
DELETE	/contact/{contactId}	삭제	—	No Content
POST	/contact/{contactId}/comment	댓글 작성	{ content }	No Content
GET	/contact/{contactId}/comment	댓글 조회	—	[CommentDto,...]
PUT	/contact/{contactId}/comment/{commentId}	댓글 수정	{ content }	No Content
DELETE	/contact/{contactId}/comment/{commentId}	댓글 삭제	—	No Content
POST	/contact/{contactId}/image	이미지 추가	{ imageUrl, order }	No Content
GET	/contact/{contactId}/image	이미지 조회	—	[ImageDto,...]
DELETE	/contact/{contactId}/image/{imageId}	이미지 삭제	—	No Content

10. AI 이미지 & 추천 (AI)

Method	Path	설명	Request	Response
POST	/ai/transform	DALL·E 편집 → URL 반환	multipart/form-datafile , userId	{ originalImageUrl, gptImageUrl }
POST	/ai/recommend	이미지 기반 장소 추천 (GPT)	{ "imageUrl": "... " }	[{ name, lat, lng, placeId,... }, ...]

Tip:

- Postman에선 **Environment Variable** `{{baseUrl}}` 를 `http://localhost:8080/api` 로 설정해 두면 편합니다.
 - 컬렉션에 Route, Query, Body, Auth(Bearer Token) 까지 모두 등록해 두면 API 테스트/공유가 훨씬 수월합니다.
-