

# Project Routes Documentation

## User Management Endpoints

- **GET /api/users** — Retrieves a list of all users. (*ADMIN role required*)
- **GET /api/users/{id}** — Fetches user data by ID, excluding sensitive fields. (*USER, EVENT\_ORGANIZER, ADMIN*)
- **DELETE /api/users/{id}** — Deletes a user by ID. (*ADMIN role required*)
- **GET /api/users/all** — Public endpoint returning "Public Content." (*No auth required*)
- **GET /api/users/user** — Returns "User Content." (*USER, EVENT\_ORGANIZER, ADMIN*)
- **GET /api/users/organizer** — Returns "Event Organizer Board." (*EVENT\_ORGANIZER only*)
- **GET /api/users/admin** — Returns "Admin Board." (*ADMIN only*)

## User–Event Association Endpoints

- **POST /api/users/{userId}/events/{eventId}** — Adds event to user's list. (*Success/Error JSON*)
- **DELETE /api/users/{userId}/events/{eventId}** — Removes event from user's list. (*Success/Error JSON*)

## Wallet Service Endpoints

- **GET /api/wallet/balance-alt?userId={userId}** — Alias for balance endpoint. (*Bearer token required*)
- **GET /api/wallet/balance?userId={userId}** — Get user balance.
- **POST /api/wallet/add-funds?userId={userId}&amount={amount}** — Add funds. (*Bearer token required*)
- **POST /api/wallet/deduct-funds?userId={userId}&amount={amount}** — Deduct funds. (*Bearer token required*)
- **POST /api/wallet/create-wallet?userId={userId}&initialBalance={balance}** — Create wallet. (*Bearer token required*)
- **GET /api/wallet/transactions?userId={userId}** — Get wallet transaction history. (*Bearer token required*)
- **GET /api/wallet/check-wallet?userId={userId}** — Check wallet existence. (*Bearer token required*)
- **GET /api/wallet/test-token** — Echoes Authorization header (debug).
- **POST /api/wallet/init-db** — Initializes mock DB. (*For testing*)

## Web Page Controller Endpoints

- GET **/google-login** — Google login HTML page.
- GET **/auth-success** — OAuth success page.
- GET **/auth-error** — OAuth error page.

## AuthController

- POST **/signin** — Sign in via Supabase
- POST **/signup** — Sign up via Supabase
- POST **/signout** — Sign out
- POST **/reset-password** — Initiate password reset
- POST **/email/signup, /email/signin** — Auth using email (raw payload)
- POST **/phone/signup, /phone/signin** — Auth via phone/OTP
- POST **/phone/verify-otp** — Verify OTP

## OAuthController **/api/oauth**

- GET|POST **/google** — Get Google OAuth URL
- GET **/callback?code={code}** — Handle callback
- GET **/success?token={token}&refresh\_token={refresh\_token} [&expires\_in={expiresIn}]** — Redirect with tokens
- GET|POST **/google/redirect** — Redirect to OAuth consent screen

## SupabaseAuthController **/api/auth**

- POST **/signup** — Email signup
- POST **/signin** — Email signin  
POST **/signout** — Signout by token
- GET **/validate** — Validate session token
- POST **/reset-password** — Send reset email

## NotificationController **/api/notifications**

- POST **/send** — Send notification (*Returns Notification*)
- GET **/user/{userId}** — Get user notifications (*Returns List*)
- POST **/command/{action}/{id}** — Mark/delete notification (*Specify action, returns updated object or success flag*)

## SearchController **/api/search**

- GET **/** — Search events (*Query: keyword, location, fromDate, toDate*)

## BookingController

- POST **/bookings?roomType={roomType}&nights={nights}** — Book room
- POST **/bookings/events/{eventId}?userId={userId}** — Book event
- DELETE **/bookings/{bookingId}** — Cancel booking
- POST **/bookings/{bookingId}/refund** — Refund booking
- POST **/bookings/events/payment?userId={userId}&eventId={eventId}** — Pay for event
- POST **/bookings/events/refund?userId={userId}&eventId={eventId}** — Refund for event

## CategoryController

- GET **/api/categories** — Get all categories
- GET **/api/categories/{id}** — Get category by ID
- POST **/api/categories** — Create category
- PUT **/api/categories/{id}** — Update category
- DELETE **/api/categories/{id}** — Delete category

## EventController **/api/events**

- GET **/** — All events (*page, size, sortBy, direction*)
- GET **/id** — Get event by ID
- POST **/** — Create event (*with optional category params*)
- PUT **/id** — Update event (*with optional category params*)
- DELETE **/id** — Delete event
- GET **/organizer/{organizerId}** — Events by organizer (*page, size*)
- GET **/public** — Public events (*page, size*)
- GET **/upcoming** — Upcoming events (*page, size*)
- GET **/category/{categoryId}** — Events by category (*page, size*)
- GET **/search** — Search by keyword (*keyword, page, size*)
- PATCH **/id/publish** — Publish event
- PATCH **/id/unpublish** — Unpublish event
- PATCH **/id/type** — Change type (*eventType param*)

- **POST** `/id/signup?userId={userId}` — Signup user to event
- **GET** `/participant/{userId}` — User's participated events (*page, size*)
- **GET** `/user/{userId}` — All user-related events (*page, size*)
- **GET** `/available` — Events with tickets (*page, size*)
- **GET** `/id/available-tickets` — Ticket count
- **PUT** `/id/available-tickets?delta={delta}` — Adjust tickets
- **POST** `/id/book?userId={userId}&numberOfTickets={n}` — Book tickets
- **POST** `/eventId/participants/{userId}` — Add participant
- **DELETE** `/eventId/participants/{userId}` — Remove participant
- **GET** `/eventId/participants/{userId}` — Check registration