Hotel Booking System API API Documentation

Version: 0.1.0

Generated on: 2025-11-09

Author: Aswinnath TE

API Documentation

`/roles/`

GET: List Roles

Description:

Tags: ROLES

Responses:

- `200` — Successful Response

POST: Create New Role

Description:

Tags: ROLES

Request Body Example:

Responses:

- `200` — Successful Response

- `422` — Validation Error

`/permissions/assign`

POST: Assign Permissions To Role

Description:

Tags: PERMISSIONS

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/permissions/`

GET: Get Permissions

Description:

Tags: PERMISSIONS

Parameters:

- `permission_id` (query) — Permission id to fetch roles for
- `role_id` (query) — Role id to fetch permissions for
- `resources` (query) — List of resources to filter by

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/auth/signup`

POST: Signup

Description:

Tags: AUTH

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

`/auth/otp/request`

POST: Request Otp

Description: Request an OTP for password reset or signin verification.
If SMTP is not configured the generated OTP will be returned in the response (dev convenience).

Tags: AUTH

Request Body Example:

Responses:

- `202` — Successful Response
- `422` — Validation Error

`/auth/otp/verify`

POST: Verify Otp Endpoint

Description: Verify an OTP. If verification_type is PASSWORD_RESET and new_password is provided, update the user's password.

Tags: AUTH

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/auth/login`

POST: Login

Description: OAuth2 Login endpoint that returns JWT tokens on successful authentication.

Tags: AUTH

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/auth/refresh`

POST: Refresh Tokens

Description: Refresh (rotate) access and refresh tokens using a valid refresh token.

Tags: AUTH

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/auth/logout`

POST: Logout

Description: Logout: blacklist tokens and revoke the session associated with the provided access token.

Tags: AUTH

Responses:

- `200` — Successful Response

`/auth/admin/register`

POST: Register Admin

Description: Endpoint to create Admin users.
Requires `Admin_Creation` resource with `WRITE` permission.

Tags: AUTH

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

`/api/room-types/`

POST: Create Room Type

Description:

Tags: ROOM_TYPES

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: Get Room Types

Description: Single GET for room-types.

- If `room_type_id` is provided, return the single RoomTypeResponse.
- Otherwise return a list of RoomTypeResponse (filtered by include_deleted). Authentication required; basic users are allowed.

Tags: ROOM_TYPES

Parameters:

- `room_type_id` (query) — If provided, returns the single room type with this ID
- `include_deleted` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/room-types/{room_type_id}`

PUT: Update Room Type

Description:

Tags: ROOM_TYPES

Parameters:

- `room_type_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

DELETE: Soft Delete Room Type

Description:

Tags: ROOM_TYPES

Parameters:

- `room_type_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/rooms/`

POST: Create Room

Description:

Tags: ROOMS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: Get Rooms

Description: Single GET endpoint for rooms.

Behavior:

- If `room_id` is provided: return the single room as `RoomResponse` .
- Otherwise: return a list of `Room` matching optional filters.

Access control: only non-basic users (admins/managers) may access this endpoint.

Tags: ROOMS

Parameters:

- `room_id` (query) — If provided, returns the single room with this ID
- `room_type_id` (query) —
- `status_filter` (query) —
- `is_freezed` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/rooms/{room_id}`

PUT: Update Room

Description:

Tags: ROOMS

Parameters:

- `room_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

DELETE: Delete Room

Description:

Tags: ROOMS

Parameters:

- `room_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/amenities/`

POST: Create Amenity

Description:

Tags: AMENITIES

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: Get Amenities

Description:

Tags: AMENITIES

Parameters:

- `amenity_id` (query) — If provided, return this amenity and its linked rooms

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/amenities/{amenity_id}`

DELETE: Delete Amenity

Description:

Tags: AMENITIES

Parameters:

- `amenity_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/room-amenities/`

POST: Map Amenity

Description:

Tags: ROOM_AMENITIES

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

DELETE: Unmap Amenity

Description:

Tags: ROOM_AMENITIES

Parameters:

- `room_id` (query) —
- `amenity_id` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/room-amenities/room/{room_id}`

GET: Get Amenities For Room

Description:

Tags: ROOM_AMENITIES

Parameters:

- `room_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/rooms/{room_id}/images/`

POST: Upload Image For Room

Description:

Tags: ROOM_IMAGES

Parameters:

- `room_id` (path) —

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Images For Room

Description: Retrieve all images associated with a specific room.

Tags: ROOM_IMAGES

Parameters:

- `room_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

DELETE: Delete Images For Room

Description: Hard-delete one or more images. The requester must be the uploader of each image or have room_service.WRITE.

Tags: ROOM_IMAGES

Parameters:

- `image_ids` (query) — List of image IDs to delete

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/rooms/{room_id}/images/{image_id}/primary`

PUT: Mark Image Primary

Description: Mark a specific image as primary for the room's images. Only the uploader or users with ROOM_MANAGEMENT.WRITE may do this.

Tags: ROOM_IMAGES

Parameters:

- `image_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/offers/`

POST: Create Offer

Description: Create an offer and its room mappings via the service layer.

Only users with Resources.OFFER_MANAGEMENT and PermissionTypes.WRITE may perform this action.

Tags: OFFERS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Offers

Description:

Tags: OFFERS

Parameters:

- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/offers/{offer_id}`

GET: Get Offer

Description:

Tags: OFFERS

Parameters:

- `offer_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

PUT: Edit Offer

Description:

Tags: OFFERS

Parameters:

- `offer_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

DELETE: Delete Offer

Description:

Tags: OFFERS

Parameters:

- `offer_id` (path) —

Responses:

- `204` — Successful Response
- `422` — Validation Error

`/api/bookings/`

POST: Create Booking

Description:

Tags: BOOKINGS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: Get Bookings

Description: Unified GET for bookings.

- If requester is NOT a basic user and no query params are present -> return all bookings (paginated).
- If query params (booking_id/status) are provided -> perform query. Basic users are restricted to their own bookings (user_id from token).

- Basic users (role_id == 1) always receive only their own bookings.

Tags: BOOKINGS

Parameters:

- `booking_id` (query) —
- `status` (query) —
- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/bookings/{booking_id}/cancel`

POST: Cancel Booking

Description:

Tags: BOOKINGS

Parameters:

- `booking_id` (path) —

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

`/api/bookings/{booking_id}`

GET: Get Booking

Description:

Tags: BOOKINGS

Parameters:

- `booking_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/wishlist/`

GET: List Wishlist

Description:

Tags: WISHLIST

Responses:

- `200` — Successful Response

POST: Add Wishlist

Description:

Tags: WISHLIST

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

`/api/wishlist/item`

GET: Get Wishlist Item

Description: Retrieve a single wishlist entry for the current user by room_type_id or offer_id.

Tags: WISHLIST

Parameters:

- `room_type_id` (query) —
- `offer_id` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/wishlist/{wishlist_id}`

DELETE: Delete Wishlist

Description:

Tags: WISHLIST

Parameters:

- `wishlist_id` (path) —

Responses:

- `204` — Successful Response
- `422` — Validation Error

`/api/notifications/`

POST: Create Notification

Description:

Tags: NOTIFICATIONS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Notifications

Description:

Tags: NOTIFICATIONS

Parameters:

- `include_read` (query) —
- `include_deleted` (query) —
- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/notifications/{notification_id}/read`

PUT: Mark Read

Description: Mark the authenticated user's notification as read.

Tags: NOTIFICATIONS

Parameters:

- `notification_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/`

POST: Create Issue

Description:

Tags: ISSUES

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List My Issues

Description:

Tags: ISSUES

Parameters:

- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/{issue_id}`

GET: Get My Issue

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

PUT: Update My Issue

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/{issue_id}/chat`

POST: Post Chat

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Chats

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/admin/`

GET: Admin List Issues

Description:

Tags: ISSUES

Parameters:

- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/admin/{issue_id}`

GET: Admin Get Issue

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

PUT: Admin Update Issue

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/issues/admin/{issue_id}/chat`

POST: Admin Post Chat

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: Admin List Chats

Description:

Tags: ISSUES

Parameters:

- `issue_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/reviews/`

POST: Create Review

Description:

Tags: REVIEWS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Or Get Reviews

Description: If review_id is provided return that single review (with images), otherwise return list (optionally filtered by booking_id).

Each returned review will include attached images in the `images` field.

Tags: REVIEWS

Parameters:

- `review_id` (query) —
- `booking_id` (query) —
- `room_id` (query) —
- `user_id` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/reviews/{review_id}/respond`

PUT: Respond Review

Description:

Tags: REVIEWS

Parameters:

- `review_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/reviews/{review_id}`

PUT: Update Review

Description: Allow the authenticated reviewer to update their review's rating/comment.

Tags: REVIEWS

Parameters:

- `review_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/reviews/{review_id}/images`

POST: Add Review Image

Description: Upload one or more images for a review.

captions may be provided as repeated form fields (`-F "captions=one" -F "captions=two"`) and will be applied to files by index. If there are fewer captions than files, remaining captions are None.

Tags: REVIEWS

Parameters:

- `review_id` (path) —

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

GET: List Review Images

Description:

Tags: REVIEWS

Parameters:

- `review_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

DELETE: Delete Review Images

Description: Delete specified image ids attached to a review. Only the uploader/review-owner or ROOM_MANAGEMENT.WRITE may delete.

Caller must provide a list of image_ids (JSON body).

Tags: REVIEWS

Parameters:

- `review_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/refunds/{refund_id}/transaction`

PUT: Complete Refund

Description:

Tags: REFUNDS

Parameters:

- `refund_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/refunds`

GET: Get Refunds

Description: Unified GET for refunds.

Rules:

- If `refund_id` provided -> return that refund (basic users can only access their own).
- Basic users (role_id == 1) can only query their refunds (user_id derived from token).
- Non-basic users can query across all refunds. If no filters provided, return paginated list.

Tags: REFUNDS

Parameters:

- `refund_id` (query) —
- `booking_id` (query) —
- `user_id` (query) —
- `status` (query) —
- `type` (query) —
- `from_date` (query) —
- `to_date` (query) —
- `limit` (query) —
- `offset` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/booking-edits/`

POST: Create Booking Edit

Description: Create a booking edit request (customer initiated).

Tags: BOOKING-EDITS

Request Body Example:

Responses:

- `201` — Successful Response
- `422` — Validation Error

`/booking-edits/active`

GET: Get Active Booking Edit

Description: Fetch active edit for a booking.
Returns None if no edit is active (PENDING or AWAITING_CUSTOMER_RESPONSE).

Tags: BOOKING-EDITS

Parameters:

- `booking_id` (query) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/booking-edits/{booking_id}`

GET: Get All Booking Edits

Description: Retrieve all edits linked to a specific booking.

Tags: BOOKING-EDITS

Parameters:

- `booking_id` (path) —

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/booking-edits/{edit_id}/review`

POST: Review Booking Edit

Description: Admin suggests room changes and locks them for 30 minutes.

Tags: BOOKING-EDITS

Parameters:

- `edit_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/booking-edits/{edit_id}/decision`

POST: Decision Booking Edit

Description: Customer accepts or rejects the admin-proposed edit.

Tags: BOOKING-EDITS

Parameters:

- `edit_id` (path) —

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/profile/`

GET: Get My Profile

Description: Return the authenticated user's profile.

Tags: PROFILE

Responses:

- `200` — Successful Response

PUT: Update My Profile

Description: Update allowed profile fields. Accepts either JSON (application/json) or multipart/form-data (form fields).

This endpoint handles only scalar/profile fields. Image uploads must use the dedicated POST /api/profile/image endpoint.

Tags: PROFILE

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/profile/image`

POST: Upload Profile Image

Description: Upload a profile image and set the user's profile_image_url.

Tags: PROFILE

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error

`/api/profile/password`

PUT: Change My Password

Description: Change current user's password by verifying current password.

Tags: PROFILE

Request Body Example:

Responses:

- `200` — Successful Response
- `422` — Validation Error