

## UniHaven API Documentation

### Base URL:

All endpoints are built upon the configured API\_BASE\_URL. For example:

<http://localhost:8000/api/>

### Authentication:

- **Token Authentication:** All protected endpoints require an HTTP header Authorization: Token <token>, where <token> is obtained from the "Me" endpoint upon login.  
TOKENS = { "HKU": "7922b1c227d795e2704d478616da6194d7506ec6",  
"CUHK": "b1d712ad4e03d24497d9d68d4b24c9c1247035b0", "HKUST":  
"a89e150b456d2a387afabe02913dcbbeba91a603" }
- **Session Authentication:** Endpoints also accept Django session cookies (if the user has logged in via the web interface).
- **Public Access:** Endpoints marked as "Public" do not require authentication.

## 1. Authentication Endpoint

### 1.1 Get Current User & Token

- **Endpoint:**  
GET /api/me/
- **Permissions:**  
Requires authentication (Token or session).
- **Description:**  
Returns the current user's profile and an auth token. If no token exists, a new one is created.

### Response (200 OK):

```
{  
  "id": 7,  
  "username": "alice123",  
  "email": "alice@example.com",  
  "university": "HKU",  
  "role": "student",      // or "staff"  
  "token": "abcdefghijklmnopqrstuvwxyz123456"  
}
```

## 2. Accommodations Endpoints

### 2.1 List All Accommodations

- **Endpoint:**  
GET /api/accommodations/
- **Permissions:**  
Public
- **Description:**  
Retrieves a list of all accommodation records. Supports filtering by query parameters.
- **Query Parameters:**

Parameter	Type	Description
type	string	Filter by accommodation type, e.g., "APARTMENT", "HOUSE", "STUDIO"
min_price	number	Filter by minimum price
max_price	number	Filter by maximum price

- **Example Request:**

GET

[http://localhost:8000/api/accommodations/?type=APARTMENT&min\\_price=5000&max\\_price=10000](http://localhost:8000/api/accommodations/?type=APARTMENT&min_price=5000&max_price=10000)

- **Example Response (200 OK):**

```
[
  {
    "id": 1,
    "title": "Studio near HKU",
    "description": "A cozy studio in the heart of Hong Kong",
    "accommodation_type": "STUDIO",
    "period_of_availability": "2025-05-01 to 2025-08-31",
    "number_of_beds": 1,
    "number_of_bedrooms": 1,
    "price": "5800.00",
    "latitude": 22.2821,
    "longitude": 114.1309,
    "total_rating": 10,
    "rating_count": 2,
    "average_rating": "5.0"
  },
  ...
]
```

## 2.2 Retrieve Accommodation Details

- **Endpoint:**

GET /api/accommodations/{pk}/

- **Permissions:**

Public

- **Description:**

Fetches detailed information for a specific accommodation by ID.

- **Example Request:**

GET <http://localhost:8000/api/accommodations/1/>

- **Example Response (200 OK):**

```
{
  "id": 1,
  "title": "Studio near HKU",
  "description": "A cozy studio in the heart of Hong Kong",
  "accommodation_type": "STUDIO",
```

```

    "period_of_availability": "2025-05-01 to 2025-08-31",
    "number_of_beds": 1,
    "number_of_bedrooms": 1,
    "price": "5800.00",
    "latitude": 22.2821,
    "longitude": 114.1309,
    "total_rating": 10,
    "rating_count": 2,
    "average_rating": "5.0"
  }

```

### 2.3 Create a New Accommodation

- **Endpoint:**  
POST /api/accommodations/
- **Permissions:**  
Authenticated CEDARS staff only
- **Description:**  
Called by a CEDARS staff member, this endpoint creates a new accommodation record. The request body should include all required fields.
- **Sample Request JSON:**

```

{
  "title": "New Apartment near HKU",
  "description": "Spacious apartment with modern amenities",
  "accommodation_type": "APARTMENT",
  "period_of_availability": "2025-06-01 to 2025-12-31",
  "number_of_beds": 2,
  "number_of_bedrooms": 2,
  "price": "8500.00",
  "latitude": 22.2840,
  "longitude": 114.1370,
  "created_by": 3,
  "owner": 3
}

```

- **Example Response (201 Created):**

```

{
  "id": 5,
  "title": "New Apartment near HKU",
  "description": "Spacious apartment with modern amenities",
  "accommodation_type": "APARTMENT",
  "period_of_availability": "2025-06-01 to 2025-12-31",
  "number_of_beds": 2,
  "number_of_bedrooms": 2,
  "price": "8500.00",
  "latitude": 22.2840,
  "longitude": 114.1370,
}

```

```

        "total_rating": 0,
        "rating_count": 0,
        "created_by": 3,
        "owner": 3
    }

```

## 2.4 Filter Accommodations

- **Endpoint:**  
GET /api/accommodations/filter/
- **Permissions:**  
Authenticated
- **Description:**  
Applies detailed filters on accommodation records based on property type, available dates, minimum number of beds, minimum number of bedrooms, price range and so on. Finally, sorts by distance.

- **Query Parameters:**

Parameter	Type	Description
property_type	string	E.g., "HOUSE", "APARTMENT"
available_from	date	Start date of availability
available_to	date	End date of availability
min_beds	integer	Minimum number of beds
min_bedrooms	integer	Minimum number of bedrooms
min_price	float	Minimum price
max_price	float	Maximum price
room_number	string	Specific room number
flat_number	string	Specific flat number
floor_number	string	Floor number
latitude	float	Base latitude for distance sorting
longitude	float	Base longitude for distance sorting

- **Behavior:**
  - Results are first filtered to is\_available=True and to those offered to the user's university.
  - If latitude and longitude are provided, results are sorted by line-of-sight distance using the equirectangular formula.

- **Example Request:**

```

GET
/api/accommodations/filter/?property_type=HOUSE&available_from=2025-
07-01&available_to=2025-12-
31&min_beds=2&room_number=101&latitude=22.28&longitude=114.13

```

- **Response:**

Returns a list of accommodations matching the filter criteria. The response format is the same as in Section 1.1.

---

### 3. Reservations Endpoints

#### 3.1 List or Filter Reservations

- **Endpoint:**  
GET /api/reservations/filter/
- **Permissions:**  
Authenticated
- **Description:**
  - If the user is a student, returns only that student's reservations.
  - If the user is CEDARS staff, returns all reservations for accommodations offered to the staff's university.
- **Example Response (200 OK):**

```
[
  { "id": 8, "student": 4, "accommodation": 1, "status": "RESERVED",
    "reservation_date": "2025-04-15T09:13:42Z" },
  ...
]
```

#### 3.2 Create a Reservation

- **Endpoint:**  
POST /api/reservations/
- **Permissions:**  
Authenticated students
- **Description:**  
Creates a reservation record for a specific accommodation. The initial status is set to "RESERVED."
- **Sample Request JSON:**

```
{
  "student": 4,
  "accommodation": 1,
  "status": "RESERVED"
}
```
- **Example Response (201 Created):**

```
{
  "id": 8,
  "student": 4,
  "accommodation": 1,
  "reservation_date": "2025-04-15T09:13:42Z",
  "status": "RESERVED"
}
```
- **Error Example:**  
If the accommodation is already reserved, the response returns HTTP 400:

```
{
```

```
    "error": "Accommodation is already reserved."
  }
}
```

### 3.3 Cancel a Reservation

- **Endpoint:**  
POST /api/reservations/{pk}/cancel/
- **Permissions:**  
Authenticated (student who owns it or CEDARS staff for that university)
- **Description:**  
Cancels an existing reservation based on its ID. Cancellation is only allowed for reservations currently in the "RESERVED" state.
- **Example Request:**  
POST http://localhost:8000/api/reservations/8/cancel/
- **Example Response (200 OK):**

```
{
  "id": 8,
  "student": 4,
  "accommodation": 1,
  "reservation_date": "2025-04-15T09:13:42Z",
  "status": "CANCELLED"
}
```
- **Error (403 Forbidden):**  
If the user is not authorized to cancel.

### 3.4 View All Reservations

- **Endpoint:**  
GET /api/reservations/
- **Permissions:**  
Authenticated (typically for admin dashboard)
- **Description:**  
Returns all reservation records. This can be used by the management dashboard or to filter reservations by student using query parameters.
- **Example Request:**  
GET http://localhost:8000/api/reservations/?student=4
- **Response:**  
Returns a list of reservations matching the filter criteria with the same format as above.

---

## 4. Ratings Endpoints

### 4.1 Submit a Rating

- **Endpoint:**  
POST /api/ratings/
- **Permissions:**  
Authenticated students
- **Description:**  
Submit a score (0–5) and optional comment for an accommodation. One rating

per student per accommodation.

- **Sample Request JSON:**

```
{
  "student": 4,
  "accommodation": 1,
  "score": 5,
  "comment": "Excellent location and facilities."
}
```

- **Example Response (201 Created):**

```
{
  "id": 3,
  "student": 4,
  "accommodation": 1,
  "score": 5,
  "comment": "Excellent location and facilities.",
  "rated_on": "2025-04-15T12:00:00Z"
}
```

## 4.2 View All Ratings

- **Endpoint:**

GET /api/ratings/

- **Permissions:**

Public

- **Description:**

Retrieves all ratings across all accommodations.

- **Example Request:**

GET http://localhost:8000/api/ratings/

- **Response:**

Returns a list of ratings in a format similar to the above example.

---

## 5. Public API Endpoints

### 5.1 Address Lookup

- **Endpoint:**

POST /api/address-lookup/

- **Permissions:**

Public

- **Description:**

Accepts an address string and uses an external service (geodata.gov.hk API) to determine the corresponding latitude, longitude, and standardized address.

- **Sample Request JSON:**

```
{
  "address": "HKU Main Building"
}
```

- **Example Response (200 OK):**

```
{
```

```
"latitude": 22.2840,  
"longitude": 114.1370,  
"geo_address": "The University of Hong Kong, Pok Fu Lam"  
}
```

- **Error Example:**

If the address parameter is missing or the external lookup fails, an appropriate error status and message will be returned.

## 5.2 Accommodation Filter

- **Endpoint:**

GET /api/accommodations/filter/

- **Permissions:**

Public

- **Description:**

Applies detailed filtering on accommodations using various query parameters. This endpoint behaves similarly to Section 2.4.

- **Example Request:**

GET

http://localhost:8000/api/accommodations/filter/?property\_type=HOUSE&min\_price=5000&max\_price=12000

- **Response:**

Same format as described in Section 2.4.

---

## 6. Users Endpoints

### 6.1 User Login

- **Endpoint:**

POST /users/login/

- **Description:**

Processes a user login using form submission. The user is authenticated using Django's built-in authentication.

- **Parameters:**

- username: The username
- password: The password

- **Response:**

On successful login, the user is redirected to their respective homepage. On failure, an error message is displayed.