

Gym Pass API Endpoints Doc - English

Gym Management API Documentation

Overview

This API allows users to register, authenticate, and manage gym check-ins. It provides endpoints for creating and managing gyms, recording check-ins, and viewing metrics about user activities.

Authentication

The API uses JWT tokens for authentication. Most endpoints require a valid Bearer token in the Authorization header.

- **Login:** After authenticating via `/sessions`, you'll receive a JWT token and a refresh token (as an HTTP cookie).
- **Token Refresh:** Use the `/token/refresh` endpoint to obtain a new token using the refresh token cookie.
- **Authorization:** Include token in requests with header `Authorization: Bearer {token}`.

Role-Based Access Control

The API implements role-based access control:

- **MEMBER:** Regular users who can check in to gyms
- **ADMIN:** Users who can create gyms and validate check-ins

Base URL

All URLs referenced in the documentation have the following base:

```
https://api.example.com
```

Endpoints

Users

Register User

POST /users

Creates a new user account.

Request Body:

```
{
  "name": "string",
  "email": "string",
  "password": "string"
}
```

Validation:

- email must be a valid email address
- password must be at least 6 characters long

Responses:

- 201 Created : User successfully registered
- 409 Conflict : Email already in use

Authenticate

POST /sessions

Authenticates a user and returns a JWT token.

Request Body:

```
{
  "email": "string",
  "password": "string"
}
```

Responses:

- 200 OK : Authentication successful

```
{
  "token": "string"
}
```

```
}
```

Also sets an HTTP-only cookie containing a refresh token.

- 400 Bad Request : Invalid credentials

Refresh Token

```
PATCH /token/refresh
```

Refreshes the JWT token using the refresh token cookie.

Responses:

- 200 OK : Token refreshed successfully

```
{
  "token": "string"
}
```

Also updates the refresh token cookie.

- 401 Unauthorized : Invalid or expired refresh token

Get User Profile

```
GET /me
```

Returns the current user's profile information.

Authorization Required: Yes (JWT)

Responses:

- 200 OK : Profile retrieved successfully

```
{
  "user": {
    "id": "string",
    "name": "string",
    "email": "string",
    "role": "ADMIN" | "MEMBER"
  }
}
```

```
}  
}
```

- 401 Unauthorized : Invalid or missing token

Gyms

Create Gym

POST /gyms

Creates a new gym. Admin access required.

Authorization Required: Yes (JWT, ADMIN role)

Request Body:

```
{  
  "title": "string",  
  "description": "string | null",  
  "phone": "string | null",  
  "latitude": "number",  
  "longitude": "number"  
}
```

Validation:

- latitude must be between -90 and 90
- longitude must be between -180 and 180

Responses:

- 201 Created : Gym successfully created
- 401 Unauthorized : User not authorized or not an admin

Search Gyms

GET /gyms/search

Searches for gyms by title.

Authorization Required: Yes (JWT)

Query Parameters:

- `query` : Search term (string, required)
- `page` : Page number for pagination (number, default: 1)

Responses:

- `200 OK` : Search results

```
{
  "gyms": [
    {
      "id": "string",
      "title": "string",
      "description": "string | null",
      "phone": "string | null",
      "latitude": "number",
      "longitude": "number"
    }
  ]
}
```

- `401 Unauthorized` : Invalid or missing token

Nearby Gyms

GET /gyms/nearby

Lists gyms near the specified location.

Authorization Required: Yes (JWT)

Query Parameters:

- `latitude` : User's latitude (number, required)
- `longitude` : User's longitude (number, required)

Validation:

- `latitude` must be between -90 and 90
- `longitude` must be between -180 and 180

Responses:

- 200 OK : Nearby gyms

```
{
  "gyms": [
    {
      "id": "string",
      "title": "string",
      "description": "string | null",
      "phone": "string | null",
      "latitude": "number",
      "longitude": "number"
    }
  ]
}
```

- 401 Unauthorized : Invalid or missing token

Check-ins

Create Check-in

POST /gyms/{gymId}/check-ins

Records a user check-in at a specific gym.

Authorization Required: Yes (JWT)

Path Parameters:

- gymId : UUID of the gym (string, required)

Request Body:

```
{
  "latitude": "number",
  "longitude": "number"
}
```

Validation:

- latitude must be between -90 and 90
- longitude must be between -180 and 180
- User must be near the gym location

Responses:

- 201 Created : Check-in recorded successfully
- 401 Unauthorized : Invalid or missing token
- 400 Bad Request : Invalid location or other validation errors

Validate Check-in

```
PATCH /check-ins/{checkInId}/validate
```

Validates a user's check-in. Admin access required.

Authorization Required: Yes (JWT, ADMIN role)

Path Parameters:

- checkInId : UUID of the check-in (string, required)

Responses:

- 204 No Content : Check-in validated successfully
- 401 Unauthorized : User not authorized or not an admin
- 404 Not Found : Check-in not found

Check-in History

```
GET /check-ins/history
```

Returns the authenticated user's check-in history.

Authorization Required: Yes (JWT)

Query Parameters:

- page : Page number for pagination (number, default: 1)

Responses:

- 200 OK : Check-in history retrieved successfully

```
{  
  "checkIns": [  
    {
```

```

      "id": "string",
      "created_at": "date",
      "validate_at": "date | null",
      "gym_id": "string",
      "user_Id": "string"
    }
  ]
}

```

- 401 Unauthorized : Invalid or missing token

Check-in Metrics

GET /check-ins/metrics

Returns metrics about the authenticated user's check-ins.

Authorization Required: Yes (JWT)

Responses:

- 200 OK : Metrics retrieved successfully

```

{
  "checkInsCount": "number"
}

```

- 401 Unauthorized : Invalid or missing token

Error Responses

All endpoints may return the following error responses:

- 400 Bad Request : Invalid request parameters or body
- 401 Unauthorized : Authentication required or insufficient permissions
- 404 Not Found : Resource not found
- 409 Conflict : Resource conflict (e.g., email already in use)
- 500 Internal Server Error : Server error

Error response body:

```
{  
  "message": "string"  
}
```

Data Models

User

```
{  
  "id": "string (uuid)",  
  "name": "string",  
  "email": "string",  
  "password_hash": "string",  
  "role": "ADMIN | MEMBER"  
}
```

Gym

```
{  
  "id": "string (uuid)",  
  "title": "string",  
  "description": "string | null",  
  "phone": "string | null",  
  "latitude": "number",  
  "longitude": "number"  
}
```

Check-in

```
{  
  "id": "string (uuid)",  
  "created_at": "date",  
  "validate_at": "date | null",  
  "gym_id": "string (uuid)",  
  "user_id": "string (uuid)"  
}
```