

Echo Trails API Documentation

Backend API service for Echo Trails application. Built with FastAPI and MongoDB.

Base URL

```
https://echo-trails-backend.vercel.app
```

API Endpoints

Root Endpoint

```
GET /
```

Response:

```
{
  "message": "Hello from Echo Trails API",
  "status": "ok"
}
```

Authentication Endpoints

Register New User

```
POST /users/register
Content-Type: application/json
```

Request:

```
{
  "username": "johndoe",
  "email": "john@example.com",
  "password": "securepassword"
}
```

Response (201 Created):

```
{
  "_id": "65f1a2b3c4d5e6f7g8h9i0j1",
  "username": "johndoe",
  "email": "john@example.com",
  "created_at": "2024-03-26T10:00:00.000Z"
}
```

User Login

```
POST /users/login
Content-Type: application/json
```

Request:

```
{
  "email": "john@example.com",
  "password": "securepassword"
}
```

Response (200 OK):

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIs... ",
  "token_type": "bearer"
}
```

Get User Identity

```
GET /users/identify
Authorization: Bearer <access_token>
```

Response (200 OK):

```
{
  "status": "success",
  "token_info": {
    "user_id": "65f1a2b3c4d5e6f7g8h9i0j1",
    "issued_at": "2024-03-26T10:00:00.000Z",
    "expires_at": "2024-03-26T10:30:00.000Z",
    "token_valid": true,
  }
}
```

```
    "scopes": []
  },
  "user_data": {
    "id": "65f1a2b3c4d5e6f7g8h9i0j1",
    "username": "johndoe",
    "email": "john@example.com",
    "created_at": "2024-03-26T10:00:00.000Z"
  }
}
```

Audio Endpoints

Upload Audio

```
POST /audio/upload
Authorization: Bearer <access_token>
Content-Type: multipart/form-data
```

Request Form Fields:

- **file**: Audio file (required)
- **latitude**: float (required)
- **longitude**: float (required)
- **hidden_until**: datetime ISO string (required)

Response (200 OK):

```
{
  "id": "65f1a2b3c4d5e6f7g8h9i0j1"
}
```

List User's Audio Files

```
GET /audio/user/files
Authorization: Bearer <access_token>
```

Response (200 OK):

```
{
  "audio_files": [
    {
      "_id": "65f1a2b3c4d5e6f7g8h9i0j1",
      "user_id": "65f1a2b3c4d5e6f7g8h9i0j1",

```

```
    "file_name": "recording.mp3",
    "location": {
      "latitude": 12.9716,
      "longitude": 77.5946
    },
    "hidden_until": "2024-03-27T10:00:00.000Z",
    "created_at": "2024-03-26T10:00:00.000Z",
    "audio_data": 1024
  }
]
```

Get Audio File Metadata

```
GET /audio/files/{audio_id}
Authorization: Bearer <access_token>
```

Response (200 OK):

```
{
  "_id": "65f1a2b3c4d5e6f7g8h9i0j1",
  "user_id": "65f1a2b3c4d5e6f7g8h9i0j1",
  "file_name": "recording.mp3",
  "location": {
    "latitude": 12.9716,
    "longitude": 77.5946
  },
  "hidden_until": "2024-03-27T10:00:00Z",
  "created_at": "2024-03-26T10:00:00Z"
}
```

Download Audio File

```
GET /audio/files/{audio_id}/download
Authorization: Bearer <access_token>
```

Response:

- Content-Type: audio/mpeg
- Content-Disposition: attachment; filename="original_filename.mp3"
- Binary audio data stream

Error Responses

All endpoints may return the following error responses:

```
{
  "detail": "Error message description"
}
```

Common HTTP Status Codes:

- 400: Bad Request
- 401: Unauthorized
- 403: Forbidden
- 404: Not Found
- 500: Internal Server Error

Authentication

Protected endpoints require Bearer token authentication:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIs...
```

Environment Setup

Required environment variables in `.env`:

```
JWT_SECRET_KEY=your_secret_key
ALGORITHM=HS256
MONGO_DETAILS=mongodb+srv://username:password@cluster.mongodb.net/
```

Development Setup

1. Clone the repository
2. Create virtual environment:

```
python -m venv venv
```

3. Activate virtual environment:
 - Windows: `venv\Scripts\activate`
 - Unix: `source venv/bin/activate`
4. Install dependencies:

```
pip install -r requirements.txt
```

-
5. Create `.env` file with required variables
 6. Start development server:

```
uvicorn main:app --reload
```