# DestinEase API Documentation

## Overview

**DestinEase** is a comprehensive travel API that offers features such as user registration, personalized recommendations, flight searches, and weather information. This API uses **JWT-based authentication** to secure endpoints.

### Base URL

http://localhost:5980/api

## Authentication

### JWT Authentication

* To access secured endpoints, include a valid JWT in the Authorization header:
* Authorization: Bearer <JWT>

## Endpoints

### 1. **User Management**

#### **Register**

* **Endpoint:** POST /register
* **Description:** Register a new user and retrieve a JWT.
* **Request Body:**
* {  
   "email": "example@mail.com",  
   "password": "yourpassword"  
  }
* **Response:**
* {  
   "token": "<JWT>",  
   "user": {  
   "id": 1,  
   "email": "example@mail.com"  
   }  
  }

#### **Login**

* **Endpoint:** POST /login
* **Description:** Authenticate a user and retrieve a JWT.
* **Request Body:**
* {  
   "email": "example@mail.com",  
   "password": "yourpassword"  
  }
* **Response:**
* {  
   "token": "<JWT>",  
   "user": {  
   "id": 1,  
   "email": "example@mail.com"  
   }  
  }

### 2. **Preferences**

#### **Save Preferences**

* **Endpoint:** POST /preferences
* **Authentication:** Required
* **Description:** Save user preferences such as budget, weather, and food.
* **Request Body:**
* {  
   "budget": "low",  
   "weather": "sunny",  
   "foodPreferences": ["vegan", "seafood"]  
  }
* **Response:**
* {  
   "userId": 1,  
   "budget": "low",  
   "weather": "sunny",  
   "foodPreferences": ["vegan", "seafood"],  
   "timestamp": 1678901234567  
  }

#### **Get Preferences**

* **Endpoint:** GET /preferences
* **Authentication:** Required
* **Description:** Retrieve saved user preferences.
* **Response:**
* {  
   "userId": 1,  
   "budget": "low",  
   "weather": "sunny",  
   "foodPreferences": ["vegan", "seafood"],  
   "timestamp": 1678901234567  
  }

### 3. **Flights**

#### **Search Flights**

* **Endpoint:** GET /flights/search
* **Authentication:** Required
* **Description:** Search for flights between two destinations.
* **Query Parameters:**
  + origin (required): Origin city name.
  + destination (required): Destination city name.
  + date (required): Date of the flight in YYYY-MM-DD.
  + passengers: Number of passengers (default: 1).
* **Response:**
* [  
   {  
   "id": "12345",  
   "price": 350.00,  
   "currency": "USD",  
   "airline": "Airways X",  
   "departure": {  
   "airport": "JFK",  
   "time": "2024-01-10T08:00:00Z",  
   "city": "New York"  
   },  
   "arrival": {  
   "airport": "LAX",  
   "time": "2024-01-10T11:00:00Z",  
   "city": "Los Angeles"  
   },  
   "duration": 180,  
   "stops": 0  
   }  
  ]

### 4. **Weather**

#### **Get Weather for a Destination**

* **Endpoint:** GET /weather/:destinationId
* **Authentication:** Required
* **Description:** Fetch weather details for a specified destination.
* **Path Parameter:**
  + destinationId (required): The unique ID of the destination.
* **Response:**
* {  
   "destination": "Bali",  
   "weather": {  
   "type": "tropical",  
   "temperature": "25°C",  
   "humidity": "50%"  
   }  
  }

### 5. **Recommendations**

#### **Get Recommendations**

* **Endpoint:** GET /recommendations
* **Authentication:** Required
* **Description:** Get personalized destination recommendations based on user preferences.
* **Response:**
* [  
   {  
   "id": 1,  
   "name": "Bali",  
   "country": "Indonesia",  
   "budget": "moderate",  
   "cuisines": ["Asian", "Local"],  
   "description": "Tropical paradise with rich culture and beautiful beaches",  
   "imageUrl": "https://example.com/bali.jpg",  
   "currentWeather": {  
   "type": "tropical",  
   "temperature": "25°C"  
   },  
   "matchScore": 80,  
   "matchDetails": ["weather", "budget", "cuisine"]  
   }  
  ]

### Error Handling

#### Common Error Codes

| Status Code | Description |
| --- | --- |
| 400 | Bad Request. Missing or invalid inputs. |
| 401 | Unauthorized. Invalid or missing token. |
| 404 | Not Found. Resource does not exist. |
| 500 | Internal Server Error. |

#### Error Response Format

{  
 "error": "Description of the error",  
 "details": "Additional context if applicable"  
}

### Notes

1. Ensure destinationId matches the IDs in the /weather and /recommendations endpoints.
2. Replace <JWT> in requests with your actual token obtained from /register or /login.