Visa Management System API

Overview

You are required to build a RESTful API for a Visa Management System. Users can request visas for different countries, and Visa Officers can review and approve these applications.

General Constraints

- **Authentication:** All authentication and authorization must be implemented using **JWT Tokens**.
- **Authorization Header:** The token must be sent as Authorization: Bearer <SPACE> <JWT TOKEN>.
- **Case Sensitivity:** All inputs and outputs are case-sensitive.
- **Class Usage:** Use the specific class names and structures provided.

1. Database Models

A. UserInfo Model

Used to store system users (Applicants and Officers).

Field Name	Data Type	Constraints	Comments
id	Integer	Primary Key	Auto-generated
name	String		
email	String	Unique	Used for login
password	String		Encoded
roles	String		Values: "OFFICER" or "USER"

B. Visa Model

Stores the visa application details.

Field Name	Data Type	Constraints	Comments
id	Integer	Primary Key	Auto-generated
applicationId	String	Unique	E.g., "AID1234"
country	String		Target country
visaType	String		E.g., "Tourist", "Work"
duration	Float		In Months (e.g., 2.5)
nationality	String		
passportNumber	String		
status	String		Default = "approval pending"
phoneNumber	String		(Input may be int, stored as String/Int based on Entity)
applicant	UserInfo	Foreign Key	Many-to-One with UserInfo

2. Pre-Loaded Data (Data Loader)

The system must be initialized with the following users:

ID	Username	Email	Password	Role
1	Daniel	daniel123@gmail.com	pass1	OFFICER
2	Alice	alice456@gmail.com	pass2	USER
3	Jack	jack789@gmail.com	pass3	USER

3. API Endpoints & Business Logic

A. Login

- **URL:** POST /user/login
- **Access:** Public (No Token required)
- **Input:**
- JSON
- { "email": "...", "password": "..." }
- **Logic:**
  1. Check if email exists in UserInfo.
  2. Check if password matches.
- **Response (Success - 200 OK):**
- JSON
- { "username": "...", "token": "..." }
- **Response (Failure):**
  - Email not found OR Password mismatch -> **400 Bad Request**.

B. List Applications

- **URL:** GET /visa/list?applicationID=AID12345
- **Access:** Authenticated Users (Any Role)
- **Logic:** Retrieve application details by applicationID passed as a query parameter.
- **Response (Success - 200 OK):** Returns the Visa object.
- **Response (Failure):**
  - No value given in URL parameter -> **400 Bad Request**.

C. Add Visa Application

- **URL:** POST /visa/add
- **Access:** Authenticated **USER** only.
- **Input:**
- JSON
- {
- "applicationId": "...",
- "country": "...",
- "visaType": "...",
- "duration": 2.5,
- "nationality": "...",

- "passportNumber": "...",
- "phoneNumber": 1234567890
- }
- **Logic:**
  1. Check if applicationId is unique.
  2. Link the current logged-in user as the applicant.
  3. Set default status to "approval pending".
- **Response (Success - 201 Created):** Returns the created Visa object including the generated ID.
- **Response (Failure):**
  - applicationId already exists -> **400 Bad Request**.
  - Token missing -> **401 Unauthorized**.
  - User does not have role "USER" -> **403 Forbidden**.

D. Update Application Status

- **URL:** PATCH /visa/update/{id}
- **Access:** Authenticated **OFFICER** only.
- **Input:** Use UpdateDto class.
- JSON
- { "status": "Approved" }
- **Logic:** Find the visa by integer id (path variable) and update its status.
- **Response (Success - 200 OK):** Returns the updated Visa object.
- **Response (Failure):**
  - ID not found -> **404 Not Found**.
  - User does not have role "OFFICER" -> **403 Forbidden**.

E. Delete Application

- **URL:** DELETE /visa/delete/{id}
- **Access:** Authenticated **USER** (Creator) only.
- **Logic:**
  1. Find visa by integer id.
  2. **Ownership Check:** Ensure the currently logged-in user is the one who created this application.
- **Response (Success - 204 No Content):** Empty body.
- **Response (Failure):**
  - ID not found -> **404 Not Found**.
  - User is not the creator OR does not have role "USER" -> **403 Forbidden**.

Instructions

To install: mvn install

To run: mvn spring-boot:run

To test: mvn test