API Endpoints Documentation

Overview

This document outlines all API endpoints for the messaging app, their purposes, request/response formats, and relevant instructions for developers.

Base URL

Production: `https://api.domain.com/v1`
Development: `https://api-dev.domain.com/v1`

Authentication Endpoints

1. **Register User**

- **URL**: `/auth/register`
- **Method**: `POST`
- **Description**: Registers a new user.
- **Request**:

```
{
  "name": "John Doe",
  "email": "johndoe@example.com",
  "password": "password123"
}
```

- **Response**:

```
{
  "message": "Registration successful",
  "userId": "123456"
}
```

- **Error Codes**:
  - `400`: Validation error.
  - `409`: Email already exists.

---

2. **Login User**

- **URL**: `/auth/login`
- **Method**: `POST`

- **Description**: Logs in an existing user.
- **Request**:

```
{
  "email": "johndoe@example.com",
  "password": "password123"
}
```

- **Response**:

```
{
  "token": "eyJhbGciOiJIUzI1...",
  "userId": "123456"
}
```

- **Error Codes**:
  - `401`: Invalid email or password.

---

3. **Forgot Password**

- **URL**: `/auth/forgot-password`
- **Method**: `POST`
- **Description**: Sends a password reset email.
- **Request**:

```
{
  "email": "johndoe@example.com"
}
```

- **Response**:

```
{
  "message": "Password reset email sent"
}
```

User Endpoints

1. **Get User Profile**

- **URL**: `/users/:userId`
- **Method**: `GET`
- **Description**: Retrieves the user's profile details.

- **Headers**:

```
{
  "Authorization": "Bearer eyJhbGciOiJIUzI1..."
}
```

- **Response**:

```
{
  "userId": "123456",
  "name": "John Doe",
  "email": "johndoe@example.com",
  "avatar": "https://example.com/avatar.png"
}
```

---

2. **Update User Profile**

- **URL**: `/users/:userId`
- **Method**: `PUT`
- **Description**: Updates the user's profile.
- **Headers**:

```
{
  "Authorization": "Bearer eyJhbGciOiJIUzI1..."
}
```

- **Request**:

```
{
  "name": "Johnathan Doe",
  "avatar": "https://example.com/new-avatar.png"
}
```

- **Response**:

```
{
  "message": "Profile updated successfully"
}
```

Chat Endpoints

1. **Get Chat List**

- **URL**: `/chats`
- **Method**: `GET`
- **Description**: Retrieves a list of the user's recent chats.
- **Headers**:

```
{
  "Authorization": "Bearer eyJhbGciOiJIUzI1..."
}
```

- **Response**:

```
[
  {
    "chatId": "c123",
    "participant": {
      "userId": "456",
      "name": "Jane Smith",
      "avatar": "https://example.com/avatar2.png"
    },
    "lastMessage": "Hello there!",
    "timestamp": "2024-11-15T14:30:00Z"
  },
  ...
]
```

---

2. **Send Message**

- **URL**: `/chats/:chatId/messages`
- **Method**: `POST`
- **Description**: Sends a message in a chat.
- **Headers**:

```
{
  "Authorization": "Bearer eyJhbGciOiJIUzI1..."
}
```

- **Request**:

```
{
  "content": "Hello!",
  "type": "text"
}
```

- **Response**:

```
{
  "messageId": "m123",
  "content": "Hello!",
  "timestamp": "2024-11-15T14:32:00Z"
}
```

---

3. **Retrieve Messages**

- **URL**: `/chats/:chatId/messages`
- **Method**: `GET`
- **Description**: Fetches the message history for a chat.
- **Headers**:

```
{
  "Authorization": "Bearer eyJhbGciOiJIUzI1..."
}
```

- **Response**:

```
[
  {
    "messageId": "m123",
    "content": "Hello!",
    "type": "text",
    "timestamp": "2024-11-15T14:32:00Z",
    "senderId": "123456"
  },
  ...
]
```

Notifications Endpoint

1. **Send Push Notification**

- **URL**: `/notifications/send`
- **Method**: `POST`
- **Description**: Sends a push notification to a user.
- **Request**:

```
{
  "userId": "123456",
  "title": "New Message",
  "body": "You have a new message from Jane!"
}
```

- **Response**:

```
{
  "message": "Notification sent successfully"
}
```

Error Response Format

All error responses follow this format:

```
{
  "error": "Error message",
  "code": 400
}
```

Notes for API Development

1. Ensure proper validation for all input fields.
2. Test endpoints thoroughly using tools like Postman or Swagger.
3. Follow RESTful principles and ensure proper HTTP status codes are used.
