Contact Management API Documentation

February 27, 2025

Contents

1 Introduction

The Contact Management API is a robust RESTful service built with Express.js and TypeScript, featuring real-time capabilities through Socket.IO and secure data persistence using MongoDB.

2 System Architecture

2.1 Core Components

- Express.js Server
- Socket.IO Real-time Server
- MongoDB Database
- JWT Authentication
- TypeScript Type System

3 Authentication System

3.1 JWT Implementation

- Token Duration: 30 days
- Payload Structure:
  - User ID
  - Username
- Security Features:
  - Password Hashing (bcrypt)
  - Salt Rounds: 10
  - Environment-based Secret Key

4 Lock Management System

4.1 Lock Cleanup Mechanisms

The system implements three levels of lock cleanup to ensure contact management:

4.1.1 Periodic Cleanup

- **A**utomatic cleanup every 5 minutes, (we can improve it)

4.1.2 Lock Cleanup Components

- **Periodic Timer**: 5-minute interval
- **Lock Validation**: Checks timestamp against current time
- **Automatic Removal**: Deletes expired locks

4.2 Lock Cleanup Strategy

The system employs a multi-layered cleanup approach:

1. **Real-time Cleanup**
    - On-demand lock validation during acquisition
    - Immediate cleanup of expired locks when detected
2. **Disconnection Cleanup**
    - Automatic release of all user locks on socket disconnect
    - Prevents orphaned locks from disconnected users
3. **Periodic Cleanup**
    - Background task running every 5 minutes
    - Sweeps through all locks to remove expired ones
    - Ensures system stability and resource management
4. **Session Timeout Handling**
    - Unlock contact
    - Visual timeout
    - Automatic page refresh option
    - User-friendly error messages

5 Data Models

When the api starts, data is initially fetched from the database(only one time). All (update, delete) operations are performed directly on the database. This process can be optimized for better performance

5.1 Contact Model

```
interface IContact {
    _id: string;
    name: string;
    email: string;
    phone: string;
    notes?: string;
    address?: {
        street: string;
        city: string;
        country: string;
    };
}
```

5.2 User Model

```
interface IUser {
    _id: string;
    username: string;
    password: string;
}
```

6 API Endpoints

6.1 Authentication Endpoints

- **POST /api/users/register**
  - Creates new user account
  - Returns JWT token
- **POST /api/users/login**
  - Authenticates user
  - Returns JWT token

6.2 Contact Endpoints

- **GET /api/contacts**
  - Supports pagination
  - Search functionality
  - Requires authentication
- **POST /api/contacts**
  - Creates new contact
  - Validates input
  - Real-time updates
- **PUT /api/contacts/:contactId**
  - Updates existing contact
  - Supports partial updates
  - Lock mechanism
- **DELETE /api/contacts/:contactId**
  - Removes contact
  - Real-time notification

7 Real-time Features

7.1 Socket Events

- contact:updated
- contact:deleted
- contact:created
- contact:locked
- contact:unlocked
- contact:lockState

7.2 Contact Locking System

- Lock Duration: 5 minutes
- Auto-release on:
  - Timeout (idle)
  - User disconnect
  - Manual release

8 Validation System

8.1 Contact Validation Rules

- Name: 2-40 characters
- Email: Valid email format
- Phone: Valid Egyptian format
- Notes: 2-40 characters (optional)

9 Pagination System

9.1 Configuration

- Default Page Size: 5
- Searchable Fields:
  - name
  - email
  - phone
  - address (all fields) like: city, country, street

10 Security Measures

- JWT Authentication
- Password Hashing
- Input Validation
- CORS Protection
- Type Safety

11 Error Handling

11.1 HTTP Status Codes

- 200: Success
- 201: Created
- 400: Bad Request
- 401: Unauthorized
- 404: Not Found