

Contact Notes API Documentation

Overview

This API provides endpoints for managing contacts and their associated notes with full CRUD operations. All endpoints except authentication require a valid JWT token for authorization.

Base URL

<http://localhost:5000>

Authentication

Register a new user

Endpoint: POST /auth/register

Request Body:

```
{
  "username": "string",
  "password": "string"
}
```

Response:

```
{
  "message": "User created successfully",
  "user_id": 1
}
```

Status Codes:

- 201 Created: User created successfully
- 400 Bad Request: Missing username or password
- 409 Conflict: Username already exists

Login

Endpoint: POST /auth/login

Request Body:

```
{  
  "username": "string",  
  "password": "string"  
}
```

Response:

```
{  
  "access_token": "string",  
  "refresh_token": "string"  
}
```

Status Codes:

- 200 OK: Login successful
- 401 Unauthorized: Invalid credentials

Refresh Token

Endpoint: POST /auth/refresh

Headers:

- Authorization: Bearer <refresh_token>

Response:

```
{  
  "access_token": "string"
```

```
}
```

Status Codes:

- 200 OK: Token refreshed
- 401 Unauthorized: Invalid token

Logout

Endpoint: POST /auth/logout

Headers:

- Authorization: Bearer <token>

Response:

```
{  
  "message": "Successfully logged out"  
}
```

Status Codes:

- 200 OK: Logout successful
- 401 Unauthorized: Invalid or missing token

Contacts

Create a new contact

Endpoint: POST /contacts

Headers:

- Authorization: Bearer <token>

Request Body:

```
{
  "name": "string",
  "email": "string"
}
```

Response:

```
{
  "id": 1,
  "name": "string",
  "email": "string"
}
```

Status Codes:

- 201 Created: Contact created successfully
- 400 Bad Request: Name is required
- 401 Unauthorized: Invalid or missing token

List all contacts

Endpoint: GET /contacts

Headers:

- Authorization: Bearer <token>

Response:

```
[
  {
    "id": 1,
    "name": "string",
    "email": "string"
  }
]
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token

Get a contact by ID

Endpoint: GET /contacts/{contact_id}

Headers:

- Authorization: Bearer <token>

Response:

```
{  
  "id": 1,  
  "name": "string",  
  "email": "string"  
}
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Contact not found

Update a contact

Endpoint: PUT /contacts/{contact_id}

Headers:

- Authorization: Bearer <token>

Request Body:

```
{  
  "name": "string",  
  "email": "string"
```

```
}
```

Response:

```
{  
  "id": 1,  
  "name": "string",  
  "email": "string"  
}
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Contact not found

Delete a contact

Endpoint: DELETE /contacts/{contact_id}

Headers:

- Authorization: Bearer <token>

Response:

```
{  
  "message": "Contact deleted successfully"  
}
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Contact not found

Notes

Create a note for a contact

Endpoint: POST /contacts/{contact_id}/notes

Headers:

- Authorization: Bearer <token>

Request Body:

```
{  
  "body": "string"  
}
```

Alternatively, you can use these field names (field normalization):

```
{  
  "note_body": "string"  
}
```

or

```
{  
  "note_text": "string"  
}
```

Response:

```
{  
  "id": 1,  
  "body": "string",  
  "created_at": "ISO-datetime"  
}
```

Status Codes:

- 201 Created: Note created successfully
- 400 Bad Request: Note content is required
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Contact not found

Note: When a note is created, an asynchronous task is triggered to process the note.

List all notes for a contact

Endpoint: GET /contacts/{contact_id}/notes

Headers:

- Authorization: Bearer <token>

Response:

```
[
  {
    "id": 1,
    "body": "string",
    "created_at": "ISO-datetime"
  }
]
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Contact not found

Get a note by ID

Endpoint: GET /contacts/{contact_id}/notes/{note_id}

Headers:

- Authorization: Bearer <token>

Response:


```
{
  "id": 1,
  "body": "string",
  "created_at": "ISO-datetime"
}
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Note not found

Update a note

Endpoint: PUT /contacts/{contact_id}/notes/{note_id}

Headers:

- Authorization: Bearer <token>

Request Body:

```
{
  "body": "string"
}
```

Field normalization applies as in the create method.

Response:

```
{
  "id": 1,
  "body": "string",
  "created_at": "ISO-datetime"
}
```

Status Codes:

- 200 OK: Success
- 400 Bad Request: Note body is required
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Note not found

Delete a note

Endpoint: DELETE /contacts/{contact_id}/notes/{note_id}

Headers:

- Authorization: Bearer <token>

Response:

```
{  
  "message": "Note deleted successfully"  
}
```

Status Codes:

- 200 OK: Success
- 401 Unauthorized: Invalid or missing token
- 404 Not Found: Note not found

Rate Limiting

All endpoints are rate-limited to 100 requests per minute per IP address. If the rate limit is exceeded, the API will return a 429 Too Many Requests status code. The API implements exponential backoff for retries when communicating with upstream services.

Asynchronous Processing

When a note is created or updated, an asynchronous task is triggered to process the note. This processing includes:

1. Sending the note to an upstream service
2. Handling retries with exponential backoff if the upstream service is unavailable

3. Proper error handling and timeout management

Error Responses

All error responses follow this format:

```
{  
  "error": "Error message"  
}
```

In case of an unexpected server error:

```
{  
  "error": "An unexpected error occurred",  
  "detail": "Error details (only in development mode)"  
}
```