

REST-API-for-contacts-management-system

Introduction

This is a RESTful API for a contacts management system, built using Node.js and Express.js. The API supports CRUD operations (Create, Read, Update, Delete) for managing contacts.

API Endpoints

The following API endpoints are available:

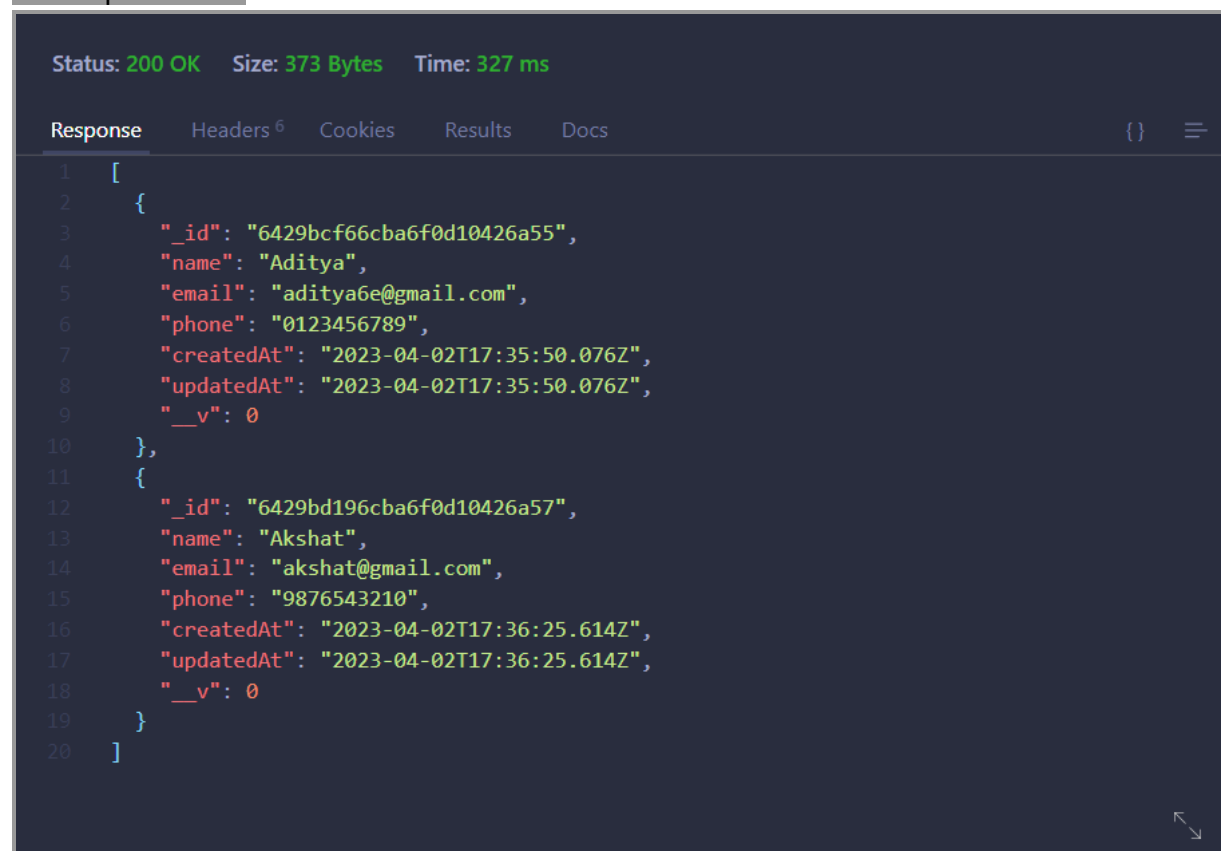
1. **Get all contacts** : This endpoint retrieves all contacts stored in the system.

Query Parameters:

page (optional): the page number to retrieve, defaults to 1.

limit (optional): the number of items to retrieve per page, defaults to 10...

GET /api/contacts



```
Status: 200 OK   Size: 373 Bytes   Time: 327 ms

Response  Headers 6  Cookies  Results  Docs  {}  ≡

1  [
2    {
3      "_id": "6429bcf66cba6f0d10426a55",
4      "name": "Aditya",
5      "email": "aditya6e@gmail.com",
6      "phone": "0123456789",
7      "createdAt": "2023-04-02T17:35:50.076Z",
8      "updatedAt": "2023-04-02T17:35:50.076Z",
9      "__v": 0
10   },
11   {
12     "_id": "6429bd196cba6f0d10426a57",
13     "name": "Akshat",
14     "email": "akshat@gmail.com",
15     "phone": "9876543210",
16     "createdAt": "2023-04-02T17:36:25.614Z",
17     "updatedAt": "2023-04-02T17:36:25.614Z",
18     "__v": 0
19   }
20 ]
```

2. **Get a specific contact** : This endpoint retrieves a specific contact by its ID.

GET /api/contacts/:id

GET `http://localhost:5001/api/contacts/6429bcf66cba6f0d10426a55` Send

Status: **200 OK** Size: **186 Bytes** Time: **1.05 s**

Query Headers ² Auth **Body ¹** Tests Pre Run

Json Xml Text Form Form-encode GraphQL Binary

Json Content Format

```
1 {
2   "name": "Akshat",
3   "email": "akshat@gmail.com",
4   "phone": "9876543210"
5 }
```

Response Headers ⁶ Cookies Results Docs

```
1 {
2   "_id": "6429bcf66cba6f0d10426a55",
3   "name": "Aditya",
4   "email": "aditya6e@gmail.com",
5   "phone": "0123456789",
6   "createdAt": "2023-04-02T17:35:50.076Z",
7   "updatedAt": "2023-04-02T17:35:50.076Z",
8   "__v": 0
9 }
```

3. Create a new contact : This endpoint creates a new contact with the specified information.

POST /api /contacts

POST `http://localhost:5001/api/contacts/` Send

Status: **201 Created** Size: **192 Bytes** Time: **1.25 s**

Query Headers ² Auth **Body ¹** Tests Pre Run

Json Xml Text Form Form-encode GraphQL Binary

Json Content Format

```
1 {
2   "name": "Rita Joshi",
3   "email": "rita.joshi@gmail.com",
4   "phone": "9963852147"
5 }
```

Response Headers ⁶ Cookies Results Docs

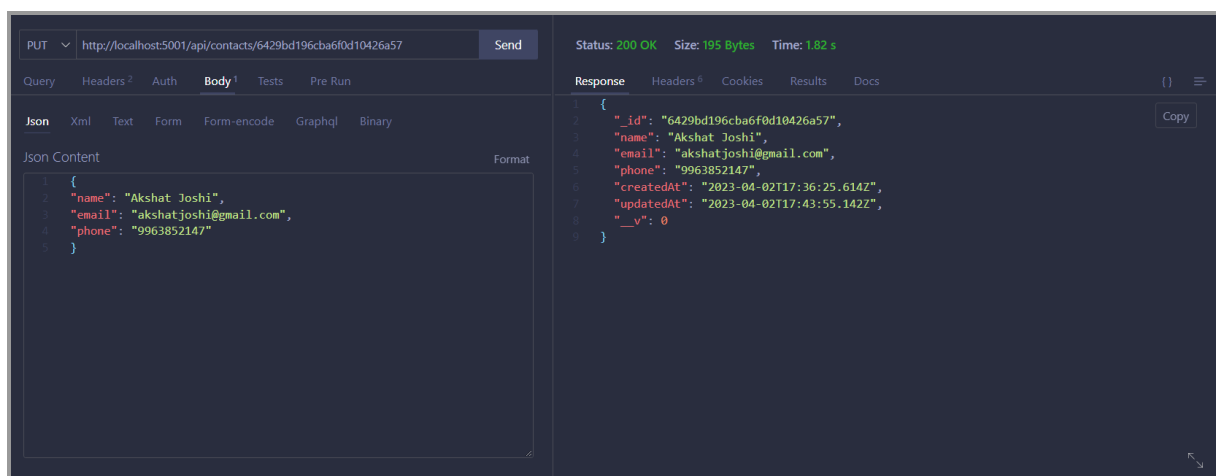
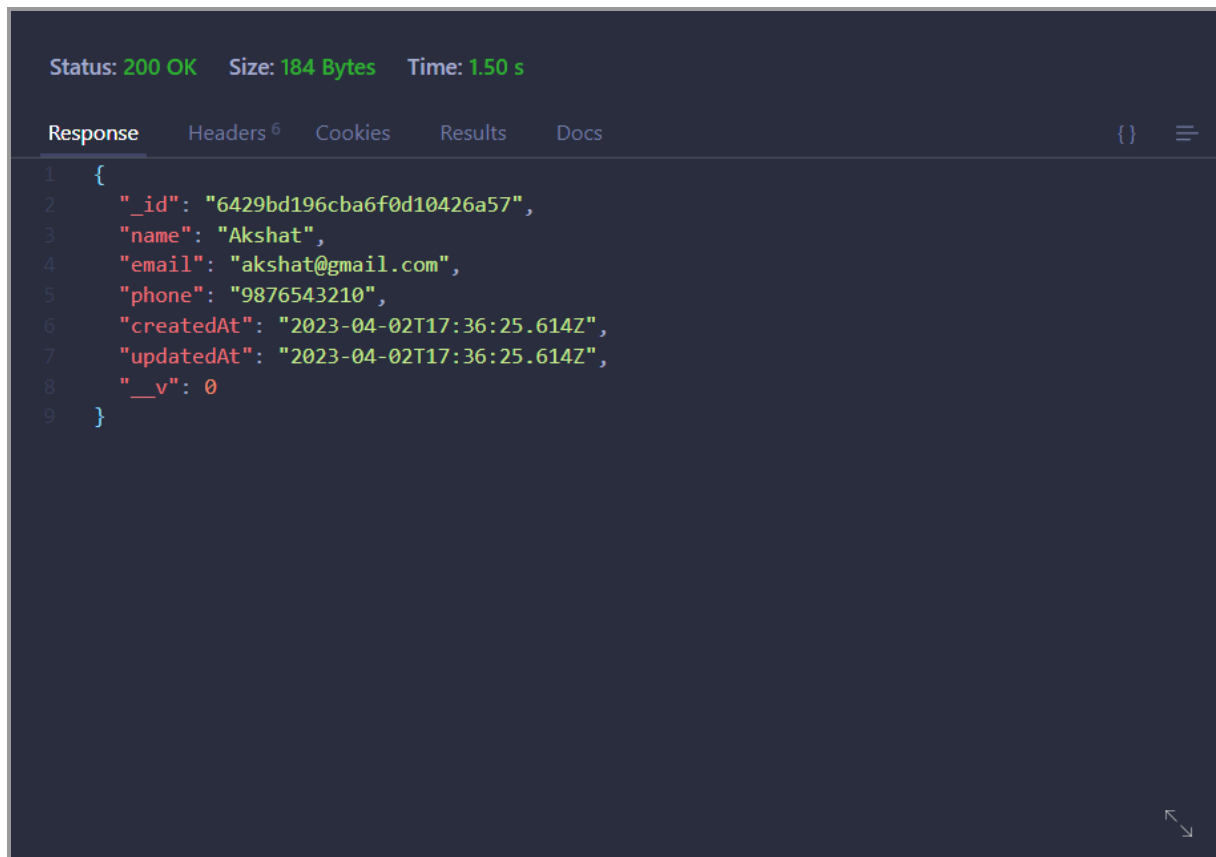
```
1 {
2   "name": "Rita Joshi",
3   "email": "rita.joshi@gmail.com",
4   "phone": "9963852147",
5   "_id": "6429be466cba6f0d10426a5b",
6   "createdAt": "2023-04-02T17:41:26.983Z",
7   "updatedAt": "2023-04-02T17:41:26.983Z",
8   "__v": 0
9 }
```

PROBLEMS OUTPUT **TERMINAL** DEBUG CONSOLE

```
[nodemon] starting `node server.js`
Server running on port 5001
Database connected: ac-4hapz1n-shard-00-00.y17jggw.mongodb.net mycontacts-backend
The request body is : { name: 'Aditya', email: 'aditya6e@gmail.com', phone: '0123456789' }
The request body is : { name: 'Akshat', email: 'akshat@gmail.com', phone: '9876543210' }
The request body is : {
  name: 'Rita Joshi',
  email: 'rita.joshi@gmail.com',
  phone: '9963852147'
}
```

4. Update an existing contact : This endpoint updates an existing contact with the specified information.

PUT /api/contacts/:id



5. **Delete a contact** : This endpoint deletes a contact with the specified ID.

DELETE /api/contacts/:id

Status: 200 OK Size: 577 Bytes Time: 1.28 s

Response

Headers ⁶

Cookies

Results

Docs

{}

```
1  [
2    {
3      "_id": "6429bcf66cba6f0d10426a55",
4      "name": "Aditya",
5      "email": "aditya6e@gmail.com",
6      "phone": "0123456789",
7      "createdAt": "2023-04-02T17:35:50.076Z",
8      "updatedAt": "2023-04-02T17:35:50.076Z",
9      "__v": 0
10   },
11   {
12     "_id": "6429bd196cba6f0d10426a57",
13     "name": "Akshat Joshi",
14     "email": "akshatjoshi@gmail.com",
15     "phone": "9963852147",
16     "createdAt": "2023-04-02T17:36:25.614Z",
17     "updatedAt": "2023-04-02T17:43:55.142Z",
18     "__v": 0
19   },
20   {
21     "_id": "6429be466cba6f0d10426a5b",
22     "name": "Rita Joshi",
23     "email": "rita.joshi@gmail.com",
24     "phone": "9963852147",
25     "createdAt": "2023-04-02T17:41:26.983Z",
26     "updatedAt": "2023-04-02T17:41:26.983Z",
27     "__v": 0
28   }
29 ]
```

Copy

DELETE ¹ http://localhost:5001/api/contacts/6429bd196cba6f0d10426a57

Send

Status: 200 OK Size: 195 Bytes Time: 653 ms

Query Headers ² Auth Body ¹ Tests Pre Run

Response Headers ⁶ Cookies Results Docs

Json Xml Text Form Form-encode GraphQL Binary

Json Content

Format

```
1 {
2   "name": "Akshat Joshi",
3   "email": "akshatjoshi@gmail.com",
4   "phone": "9963852147"
5 }
```

```
1 {
2   "_id": "6429bd196cba6f0d10426a57",
3   "name": "Akshat Joshi",
4   "email": "akshatjoshi@gmail.com",
5   "phone": "9963852147",
6   "createdAt": "2023-04-02T17:36:25.614Z",
7   "updatedAt": "2023-04-02T17:43:55.142Z",
8   "__v": 0
9 }
```

```
Status: 200 OK   Size: 381 Bytes   Time: 334 ms

Response  Headers 6  Cookies  Results  Docs  {}  ≡

1  [
2  {
3    "_id": "6429bcf66cba6f0d10426a55",
4    "name": "Aditya",
5    "email": "aditya6e@gmail.com",
6    "phone": "0123456789",
7    "createdAt": "2023-04-02T17:35:50.076Z",
8    "updatedAt": "2023-04-02T17:35:50.076Z",
9    "__v": 0
10  },
11  {
12    "_id": "6429be466cba6f0d10426a5b",
13    "name": "Rita Joshi",
14    "email": "rita.joshi@gmail.com",
15    "phone": "9963852147",
16    "createdAt": "2023-04-02T17:41:26.983Z",
17    "updatedAt": "2023-04-02T17:41:26.983Z",
18    "__v": 0
19  }
20  ]
```

Error Handling

The API uses HTTP status codes to indicate the status of a request. The following status codes may be returned:

- 200 OK - The request was successful.
- 201 Created - The request was successful and a new resource was created.
- 204 No Content - The request was successful, but there is no content to return.
- 400 Bad Requests - The request was malformed or invalid.
- 401 Unauthorised - The user is not authenticated.
- 403 Forbidden - The user does not have permission to perform the requested action.
- 404 Not Found - The requested resource could not be found.

In addition to the HTTP status code, error responses also include a JSON payload with a message property that provides a description of the error.

Status: 404 Not Found Size: 152 Bytes Time: 18 ms

Response Headers 7 Cookies Results Docs

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <title>Error</title>
6   </head>
7   <body>
8     <pre>Cannot PUT /api/contacts/</pre>
9   </body>
10 </html>
```

Preview

Status: 400 Bad Request Size: 1.21 KB Time: 36 ms

Response Headers 6 Cookies Results Docs

```
1 {
2   "title": "Validation Failed",
3   "message": "All fields are mandatory !!",
4   "stackTrace": "Error: All fields are mandatory !!\n    at C
      :\\Users\\user\\Desktop\\contact_api\\controllers\\contactController.js:19:11\n
      at asyncUtilWrap (C:\\Users\\user\\Desktop\\contact_api\\node_modules\\express-async
      -handler\\index.js:3:20)\n    at Layer.handle [as handle_request] (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\layer.js
      :95:5)\n    at next (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\route.js
      :144:13)\n    at next (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\route.js
      :140:7)\n    at Route.dispatch (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\route.js
      :114:3)\n    at Layer.handle [as handle_request] (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\layer.js
      :95:5)\n    at C:\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\ro
      uter\\index.js:284:15\n    at Function.process_params (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\index.js
      :346:12)\n    at next (C
      :\\Users\\user\\Desktop\\contact_api\\node_modules\\express\\lib\\router\\index.js
      :280:10)"
5 }
```

Copy

Running the Application

To run the application, follow these steps:

1. **Clone the repository:**

```
git clone
```

```
https://github.com/aakshatjoshii/-RESTful-API-for-a-contacts-management-system.git
```

2. **Install dependencies listed in package.json file :**

```
cd -RESTful-API-for-a-contacts-management-system
```

```
npm install
```

3. **Start the server:**

```
npm start
```

The server will start running at <http://localhost:5001>. You can now use the API endpoints to manage contacts.

Testing

To run the unit tests for the API endpoints, use the command `npm start`. Test the api endpoints using Postman or thunder client [VS code plugin].