

//Adding First user

POST ⌵ http://localhost:8088/api/users Send

Query Headers ² Auth Body ¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

```
1 {
2   "Firstname": "Moris",
3   "Lastname": "Ben",
4   "Email": "ben@gmail.com",
5   "age": "26"
```

Status: **200 OK** Size: **144 Bytes** Time: **183 ms** Response ⌵

```
1 {
2   "user": {
3     "Email": "ben@gmail.com",
4     "age": "26",
5     "Firstname": "Moris",
6     "Lastname": "Ben",
7     "_id": "65f62380b9b07bb0b91ceffd",
```

Copy

Response Chart ↕

Filter ⌵ Type a query: { field: 'value' } Reset Apply Options ▶

QUERY RESULTS: 1-1 OF 1

▶	<pre>_id: ObjectId('65f62380b9b07bb0b91ceffd') Email: "ben@gmail.com" age: "26" Firstname: "Moris" Lastname: "Ben" __v: 0</pre>	✎ 📄 📄 🗑️
----------------	---	--

// All users added

contact.contactlist

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 457B TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

INSERT DOCUMENT

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

```
_id: ObjectId('65f6188c23d754d5ea56ca9f')
Email: "emilie.b@gmail.com"
age: "40"
Firstname: "brouge"
Lastname: "Emilie"
```



```
_id: ObjectId('65f6192323d754d5ea56caa0')
age: "4"
Firstname: "brown"
Lastname: "Alex"
```



//Display all the information about only one person using his ID.

GET <http://localhost:8088/api/users/65f6188c23d754d5ea56ca9f>

Send

Query

Headers 2

Auth

Body 1

Tests

Pre Run

Status: 200 OK Size: 124 Bytes Time: 87 ms

Response

```
1  {
2    "user": {
3      "_id": "65f6188c23d754d5ea56ca9f",
4      "Email": "emilie.b@gmail.com",
5      "age": "40",
6      "Firstname": "brouge",
7      "Lastname": "Emilie"
8    }
9  }
```

//Display all the contacts with an age >18.

```
const getUsers = async (request, response) => {
  try {
    const users = await User.find({ age: { $gt: 18 } });
    if (users && users.length > 0) {
      response.status(200).json({ users: users });
    } else {
      response.status(404).json({ msg: "No users found" });
    }
  } catch (error) {
    console.error(error);
    response.status(500).json({ msg: "Error on getting users" });
  }
};
```

GET ⌵ http://localhost:8088/api/users Send

Query Headers² Auth Body¹ Tests Pre Run

Status: 200 OK Size: 230 Bytes Time: 122 ms Response ⌵

```
4      "_id": "65f6179223d754d5ea56ca9d",
5      "Email": "ben@gmail.com",
6      "age": 26,
7      "Firstname": "Moris",
8      "Lastname": "Ben"
9    },
10   {
11     "_id": "65f6188c23d754d5ea56ca9f",
12     "Email": "emilie.b@gmail.com",
13     "age": 40,
14     "Firstname": "brouge",
15     "Lastname": "Emilie"
16   }
17 ]
```

Copy

//Display all the contacts with an age>18 and name containing "ah".

```
const getUsers = async (request, response) => {
  try {
    const users = await User.find({
      age: { $gt: 18 },
      $or: [
        { Firstname: { $regex: /ah/i } },
        { Lastname: { $regex: /ah/i } }
      ]
    });
    if (users && users.length > 0) {
      response.status(200).json({ users: users });
    } else {
      response.status(404).json({ msg: "No users found" });
    }
  } catch (error) {
    console.error(error);
    response.status(500).json({ msg: "Error on getting users" });
  }
};
```

GET ⌵ http://localhost:8088/api/users Send

Query Headers ² Auth Body ¹ Tests Pre Run

Status: 404 Not Found Size: 24 Bytes Time: 109 ms Response ⌵

1 {
2 "msg": "No users found"
3 }

Copy

```
//Change the contact's first name from "Kefi Seif" to "Kefi Anis".
```

```
const putUser = async (request, response) => {
  try {
    const result = await User.updateOne(
      { Lastname: "Kefi", Firstname: "Seif" },
      { $set: { Firstname: "Anis" } }
    );

    if (result.nModified > 0) {
      response.status(200).json({ msg: "Contact's first name updated successfully" });
    } else {
      response.status(404).json({ msg: "Contact not found or first name already updated" });
    }
  } catch (error) {
    console.error(error);
    response.status(500).json({ msg: "Error updating contact's first name" });
  }
}
```

A screenshot of a REST client interface. The top bar shows a PUT request to 'http://localhost:8088/api/users' with a 'Send' button. Below this, tabs for 'Query', 'Headers', 'Auth', 'Body', 'Tests', and 'Pre Run' are visible, with 'Body' selected. The status bar indicates 'Status: 200 OK', 'Size: 100 Bytes', and 'Time: 121 ms'. The response body is displayed as a JSON object: { "msg": "Contact's first name updated successfully" }. A 'Copy' button is located to the right of the response.

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

age: 26

Firstname: "Moris"

Lastname: "Ben"

_id: ObjectId('65f6185d23d754d5ea56ca9e')

Email: "kefi@gmail.com"

age: 15

Firstname: "Anis"

Lastname: "Kefi"

//Delete the contacts that are aged under <5.

```
const deleteUser = async (request, response) => {
  try {
    const result = await User.deleteMany({ age: { $lt: 5 } });

    if (result.deletedCount > 0) {
      response.status(200).json({ msg: `${result.deletedCount} contacts aged under 5 deleted successfully` });
    } else {
      response.status(404).json({ msg: "No contacts aged under 5 found" });
    }
  } catch (error) {
    console.error(error);
    response.status(500).json({ msg: "Error deleting contacts aged under 5" });
  }
}
```

DELETE ⌵ http://localhost:8088/api/users Send

Query Headers ² Auth Body ¹ Tests Pre Run

Status: 200 OK Size: 54 Bytes Time: 110 ms Response ⌵

1 {
2 "msg": "2 contacts aged under 5 deleted successfully"
3 }

Copy

contact.contactlist

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 457B TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

INSERT DOCUMENT

Filter

Type a query: { field: 'value' }

Reset

Apply

Options

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('65f6179223d754d5ea56ca9d')
Email: "ben@gmail.com"
age: 26
Firstname: "Moris"
Lastname: "Ben"
```



```
_id: ObjectId('65f6185d23d754d5ea56ca9e')
Email: "kefi@gmail.com"
age: 15
Firstname: "Anis"
```



//Display all of the contacts list.

Only 3 contacts left after delete operation

```
const getUsers = async (request, response) => {
  try {
    const users = await User.find();
    if (users && users.length > 0) {
      response.status(200).json({ users: users });
    } else {
      response.status(404).json({ msg: "No users found" });
    }
  } catch (error) {
    console.error(error);
    response.status(500).json({ msg: "Error on getting users" });
  }
};
```

GET ⌵ http://localhost:8088/api/users Send

Query Headers ² Auth Body ¹ Tests Pre Run

Status: 200 OK Size: 336 Bytes Time: 1.21 s Response ⌵

```
1 {
2   "users": [
3     {
4       "_id": "65f6179223d754d5ea56ca9d",
5       "Email": "ben@gmail.com",
6       "age": 26,
7       "Firstname": "Moris",
8       "Lastname": "Ben"
9     },
10    {
11      "_id": "65f6185d23d754d5ea56ca9e",
12      "Email": "kefi@gmail.com",
13      "age": 15,
14      "Firstname": "Anis".
```

Response Chart

contact.contactlist

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 457B TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 36KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

INSERT DOCUMENT

Filter ⌵

Type a query: { field: 'value' }

Reset

Apply

Options ▸

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('65f6179223d754d5ea56ca9d')
Email: "ben@gmail.com"
age: 26
Firstname: "Moris"
Lastname: "Ben"
```

▶

```
_id: ObjectId('65f6185d23d754d5ea56ca9e')
Email: "kefi@gmail.com"
age: 15
Firstname: "Anis"
```

