

Name: Abhishek Singh , BuBusiness test

The project is done using mongoDb for the database and nodeJs for talking to the mongoDb

To start the project

Install mongo shell <https://docs.mongodb.com/manual/mongo/>

Clone the git repo : https://github.com/abhishek8290/uBusiness_test_abhishek_singh-.git

cd uBusiness_test_abhishek_singh-

npm install

Node index.js or npm run server

Adding the book to the database

POST : http://localhost:8000/book_register

The screenshot shows a REST client interface with a POST request to `http://localhost:8000/book_register`. The request body is a JSON object with the following fields: `_id`, `id`, `title`, `isbn`, `author`, and `__v`. The response status is 200 OK, and the response body is a JSON object with the same fields, where `_id` is a long hexadecimal string and `__v` is 0.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	1	
<input checked="" type="checkbox"/> title	book1	
<input checked="" type="checkbox"/> isbn	isbn1	
<input checked="" type="checkbox"/> author	author1	
Key	Value	Description

```
1 {
2   "_id": "5eff8583d7d347091daa5e92",
3   "id": "1",
4   "title": "book1",
5   "isbn": "isbn1",
6   "author": "author1",
7   "__v": 0
8 }
```

```
1 {
2   "_id": "5eff8583d7d347091daa5e92",
3   "id": "1",
4   "title": "book1",
5   "isbn": "isbn1",
6   "author": "author1",
7   "__v": 0
8 }
```

Adding the cardowner to the DB.

POST : http://localhost:8000/cardholder_register

POST http://localhost:8000/cardholder_register

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	1	
<input checked="" type="checkbox"/> lastName	kumar1	
<input checked="" type="checkbox"/> firstName	sohan1	
<input checked="" type="checkbox"/> cardNumber	101	
Key	Value	Description

Status: 200 OK Time: 227 ms Size: 324 B Save Response

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "5eff85c7d7d347091daa5e93",
3   "id": "1",
4   "lastName": "kumar1",
5   "firstName": "sohan1",
6   "cardNumber": "101",
7   "__v": 0
8 }
```

Step 3 : finding the book by isbn

GET : http://localhost:8000/findBook

GET http://localhost:8000/findBook

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> isbn	isbn4	
Key	Value	Description

Status: 200 OK Time: 26 ms Size: 314 B Save Response

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "5eff8601d7d347091daa5e96",
3   "id": "4",
4   "title": "book4",
5   "isbn": "isbn4",
6   "author": "author4",
7   "__v": 0
8 }
```

Step 4 : finding the cardowner by cardNumber

GET : http://localhost:8000/findBook

The screenshot shows a Postman interface with a GET request to `http://localhost:8000/findBook`. The request is sent, and the response is displayed in the 'Body' tab. The response is a JSON object with the following structure:

```
{
  "_id": "5eff8601d7d347091daa5e96",
  "id": "4",
  "title": "book4",
  "isbn": "isbn4",
  "author": "author4",
  "__v": 0
}
```

The status bar indicates a 200 OK response with a time of 26 ms and a size of 314 B.

Step 5: loaning the book to some customer using his cardNumber and isbn of the book

POST : http://localhost:8000/rentalRegister

The screenshot shows a Postman interface with a POST request to `http://localhost:8000/rentalRegister?id=1&isbn=isbn1&cardNumber=103`. The request is sent, and the response is displayed in the 'Body' tab. The response is a JSON object with the following structure:

```
{
  "_id": "5eff8679d7d347091daa5e9c",
  "id": "1",
  "isbn": "isbn1",
  "cardNumber": "101",
  "cardHolder": [
    {
      "_id": "5eff8679d7d347091daa5e9d",
      "id": "1",
      "lastName": "kumar1",
      "firstName": "sohan1",
      "cardNumber": "101"
    }
  ],
  "book": [
    {
      "_id": "5eff8679d7d347091daa5e9e",
      "id": "1",
      "title": "book1",
    }
  ]
}
```

The status bar indicates a 200 OK response with a time of 54 ms and a size of 521 B.

Step 6 : returning the book by an owner by isbn and card number

POST : http://localhost:8000/findBook

POST http://localhost:8000/returnBook

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies Code

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE	DESCRIPTION
isbn	isbn1	
cardNumber	101	
Key	Value	Description

body Cookies Headers (6) Test Results Status: 200 OK Time: 21 ms Size: 255 B Save Response

Pretty Raw Preview Visualize JSON

```
1 "the entry of the returned book is deleted"
```

Step 7: getting all the loaned books

POST : http://localhost:8000/findBook

POST http://localhost:8000/returnBook POST http://localhost:8000/returnBook GET http://localhost:8000/getallbooksonloan GET http://localhost:8000/getallbooksonloan POST http://localhost:8000/returnBook POST http://localhost:8000/returnBook GET http://localhost:8000/getallbooksonloan

GET http://localhost:8000/getallbooksonloan

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (6) Test Results Status: 200 OK Time: 38 ms Size: 309 B Save Response

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "_id": "5eff86b2d7d347091daa5ea1",
4     "id": "1",
5     "title": "book1",
6     "isbn": "isbn1",
7     "author": "author1"
8   }
9 ]
10
11 ]
```

After Adding more

GET

http://localhost:8000/getallbooksonloan

Send

Save

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

This request does not have a body

body

Cookies

Headers (6)

Test Results

Status: 200 OK

Time: 14 ms

Size: 503 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  [
2    [
3      {
4        "_id": "5eff86b2d7d347091daa5ea1",
5        "id": "1",
6        "title": "book1",
7        "isbn": "isbn1",
8        "author": "author1"
9      }
10   ],
11   [
12     {
13       "_id": "5eff86cad7d347091daa5ea4",
14       "id": "2",
15       "title": "book2",
16       "isbn": "isbn2",
17       "author": "author2"
18     }
19   ],
20   [
21     {
22       "_id": "5eff86d1d7d347091daa5ea7",
23       "id": "3",
24       "title": "book3",
25       "isbn": "isbn3",
26       "author": "author3"
27     }
28   ]
29 ]
```