**Books Café**

**This is like a library system, that contains records of books with author and their title, please find the Routes Below:**

## ADMIN-REGISTER:-

This feature is used by the admin to register into the system. They are required to enter admin name, emailId, password, address and phone number before they are allowed to enter the system.

post API- http://localhost:3000/admin/register

body: {

"name": "Rahul Abhishek",

"email":"therahulabhishek@gmail.com",

"password":"AttainU@123",

"address":"Bihar",

"phoneno":"8210383860"

}

success response:

{

"jwt": null,

"\_id": "5f0806e7f4c9690017793d07",

"adminName": "Rahul Abhishek",

"adminEmailId": "the-rahulabhishek@gmail.com",

"adminPassword": "$2a$10$NzoL/wV6zOgLejce/678seFvlKxBeoEav6j93Gj9iBhpeN4bMyO0C",

"adminAddress": "Bihar",

"adminPhoneNo": 8210383860,

"date": "2020-07-10T06:12:55.478Z",

"\_\_v": 0

}

## ADMIN-LOGIN: -

This feature is used by the admin to login into the system. They are required to enter admin emailId and password number before they are allowed to enter the system. It will return an authentication token to add a user/book and to issue/return a book.

post API- [http://localhost:3000/admin/logi](http://localhost:3000/admin/login)​ [n](http://localhost:3000/admin/login)

Body : {

"email":"the-rahulabhishek@gmail.com",

"password":"AttainU@123"

}

success response : {​

"jwttoken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVmMDgwNmU3ZjRjOTY5MDAxNzc5M2QwNyIsImlhdCI6MTU5NDM2MTY5MywiZXhwIjoxNTk3OTYxNjkzfQ.UYAqiddEzL-0ucF2GhpfLPtG6e43FkcdWSdNE1ledQY",

"status": "successful login",

"code": "202"

}

### USER-CREATE:-

This feature is used by the admin to create a user into the system.They are required to enter name,emailId,password,address and phone number.

post API- [http://localhost:3000/user](http://localhost:3000/users)​ [s](http://localhost:3000/users)

Headers:-

Authorization: jwt token from login

Body:-

{

"name": "Rahul Abhishek",

"email":"rahulabhishek@gmail.com",

"password":"AttainU@123",

"address":"Bihar",

"phoneno":"8210383860"

}

success response : {​

"message": "create user",

"user": {

"\_id": "5f0807e5f4c9690017793d09",

"name": "rahul abhishek",

"email": "rahulabhishek@gmail.com",

"password": "$2b$10$2GGmP02tu4ukZ1hAYbfDmOjZONpw51rvXCIXaPUU5h.jxHT02xP1y",

"address": "bihar",

"phoneno": 8210383860,

"createdAt": "2020-07-10T06:17:09.762Z",

"updatedAt": "2020-07-10T06:17:09.762Z",

"\_\_v": 0

}

}

### USER-GET:-

In this feature we get the all users list and a particular user.

get API- [http://localhost:3000/user](http://localhost:3000/users)​ [s](http://localhost:3000/users) ​ this is for get all users list.

get API-[http://localhost:3000/users/i](http://localhost:3000/users/id)​ [d](http://localhost:3000/users/id) this is for a particular user.​

success response​ : -​

[

{

"\_id": "5ee18da95e65a33d14f9a04e",

"username": "THERAHULABHISHEK",

"email": "therahulabhishek@gmail.com",

"password": "1234",

"\_\_v": 0

},

{

"\_id": "5f0807e5f4c9690017793d09",

"name": "rahul abhishek",

"email": "rahulabhishek@gmail.com",

"password": "$2b$10$2GGmP02tu4ukZ1hAYbfDmOjZONpw51rvXCIXaPUU5h.jxHT02xP1y",

"address": "bihar",

"phoneno": 8210383860,

"createdAt": "2020-07-10T06:17:09.762Z",

"updatedAt": "2020-07-10T06:17:09.762Z",

"\_\_v": 0

}

]

## USER-UPDATE: -

In this feature we update all details for a user. They are required to enter \_id of the user in body. ​

put API-[http://localhost:3000/user](http://localhost:3000/users/id)​ [s](http://localhost:3000/users/id)

### DELETE-USER:-

In this feature we DELETE users.

delete API-http://localhost:3000/users/i[d](http://localhost:3000/users/id)

*BOOK-CREATE:-*

This feature is used by the admin to create a book into the system.They are required to enter name,stock,author and title .

post API- -[http://localhost:3000/book](http://localhost:3000/users/id)​ [s](http://localhost:3000/users/id)

body:- {

"name": "story-book-3",

"author":"Rahul Abhishek",

"stock":"10"

}

success response:-​ {

"message": "create book",

"book": {

"time": 1594361502751,

"\_id": "5f0809daf4c9690017793d0a",

"name": "story-book-3",

"stock": 10,

"createdAt": "2020-07-10T06:25:30.636Z",

"updatedAt": "2020-07-10T06:25:30.636Z",

"\_\_v": 0

}

}

### BOOK-GET:-

In this feature we get the all books list and a particular book.

get API--[http://localhost:3000/book](http://localhost:3000/users/id)​ [s](http://localhost:3000/users/id) this is for get all books list.​

get API-[http://localhost:3000/users/i](http://localhost:3000/users/id)​[d](http://localhost:3000/users/id) this is for a particular book.​

## BOOK-UPDATE:-

In this feature we update stock for a book. They are required to enter \_id of the book in body.​

put API--[http://localhost:3000/book](http://localhost:3000/users/id)​[s](http://localhost:3000/users/id) body:-

{

"\_id": "5f0809daf4c9690017793d0a",

"name": "story-book-3",

"stock": 20

}

success response:-​

{

"message": "update book",

"result": {

"n": 1,

"nModified": 1,

"opTime": {

"ts": "6847734512000958465",

"t": 30

},

"electionId": "7fffffff000000000000001e",

"ok": 1,

"$clusterTime": {

"clusterTime": "6847734512000958465",

"signature": {

"hash": "trZJaUb3YSk2mzNP2VcHWFVsKsg=",

"keyId": "6798448589753286659"

}

},

"operationTime": "6847734512000958465"

}

}

### DELETE-USER:-

In this feature we DELETE books.

delete API-[http://localhost:3000/books/id](http://localhost:3000/users/id)

### Issue a book :-

In this feature we issue a book. They are required to enter userId,bookId and "returnStatus":​ false . for this user all update when we get this user we also see book details.

post API-[http://localhost:3000/logs/addToLo](http://localhost:3000/users/id)​[g](http://localhost:3000/users/id)

### UPDATE-LOG: -

In this feature we update the book issue or return .

patch API-[http://localhost:3000/logs/logsi](http://localhost:3000/users/id)​ [d](http://localhost:3000/users/id)

### DELETE-LOG:-

In this feature we delete the book logs.​

delete API-[http://localhost:3000/logs/logsi](http://localhost:3000/users/id)​ [d](http://localhost:3000/users/id)

### GET-LOG:-

In this feature we get all the book logs.​

get API-[http://localhost:3000/logs/logsi](http://localhost:3000/users/id)​ [d](http://localhost:3000/users/id)