

Method ☐ Request ☐ Response ☐

POST: /login

This is the user login authentication API. The request and response should be over a secured communication.

Request:

```
{"username": "john.doe", "password": "thisismysecret"}
```

Response:

200 OK

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** localhost:8000/login
- Body:**

```
1 [{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}]
```
- Response:** Status: 200 OK, Time: 80 ms, Size: 156 B
- Response Body:**

```
1 logged in!
```

GET: /users

(Login required) Gets information about the logged in user. A successfully authenticated request returns information related to the user and the books ordered.

Request:

Response:

200 OK

```
{  
  "name": "john",  
  "surname": "doe",  
  "date_of_birth": "15/01/1985",
```

```
  "books": [1, 4]  
}
```

For this part, I created a .json file to store the provided information.

Login is required.

The screenshot shows a REST client interface with a POST request to `localhost:8000/login`. The request body is a JSON object: `{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}`. The response status is 200 OK, with a time of 80 ms and a size of 156 B. The response body is `logged in!`.

```
POST localhost:8000/login
```

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

```
1 {"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}
```

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 80 ms Size: 156 B

Pretty Raw Preview Visualize Text

```
1 logged in!
```

Then getting the information from my created .json file.

The screenshot shows a REST client interface with a GET request to `localhost:8000/users`. The response status is 200 OK. The response body is a JSON array: `[{"user": {"name": "john", "surname": "doe", "date_of_birth": "15/01/1985", "books": [1, 4]}}]`.

```
GET localhost:8000/users
```

Params Authorization Headers (6) Body Pre-request Script Tests Settings

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

```
1 [{"user": {"name": "john", "surname": "doe", "date_of_birth": "15/01/1985", "books": [1, 4]}}]
```

Body Cookies Headers (5) Test Results

Status: 200 OK

Pretty Raw Preview Visualize Text

POST: /users

(Login not required) Create a user account and store user's information in Users table (DB).

Request:

```
{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}
```

Response:

200 OK

I create a user account and store in my created .json file instead DB.

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** localhost:8000/users
- Body:**

```
1 [{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}]
```
- Status:** 200 OK
- Time:** 44 ms
- Size:** 145 B

The account is added.

The screenshot shows a code editor with the following content:

```
{} user.json • JS app.js {} package-lock.json
{} user.json > ...
1 [{"user": [{"name": "john", "surname": "doe", "date_of_birth": "15/01/1985", "books": [1, 4]},
2 [{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}]]]
```

POST: /users/orders

(Login required) Order books and store order information in Orders table (DB). This returns the price for a successful order.

Request:
{"orders": [1, 4]}

Response:
200 OK
{"price": 519.00}

Login is required.

The screenshot shows a REST client interface with a POST request to `localhost:8000/login`. The request body is a JSON object: `{"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}`. The response status is 200 OK, with a time of 80 ms and a size of 156 B. The response body is `logged in!`.

POST localhost:8000/login

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

1 {"username": "john.doe", "password": "thisismysecret", "date_of_birth": "15/01/1985"}

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 80 ms Size: 156 B

Pretty Raw Preview Visualize Text

1 logged in!

Then posting the orders to get their price.

The screenshot shows a REST client interface with a POST request to `localhost:8000/users/orders`. The request body is a JSON object: `{"orders": [1, 4]}`. The response status is 200 OK, with a time of 1031 ms and a size of 159 B. The response body is `{"price": 835}`.

POST localhost:8000/users/orders

Params Authorization Headers (8) Body Pre-request Script Tests Settings

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

1 {"orders": [1, 4]}

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 1031 ms Size: 159 B

Pretty Raw Preview Visualize JSON

1 {"price": 835}

GET: /books

(Login not required) Gets a list of books from an external book publisher's web services and returns the list sorted alphabetically with the **recommended books always appears first**. The should be **no duplicated books** in the list.

Request:

Response:

200 OK

```
{
  "books": [
    {"id": 5, "name": "An American Princess: The Many Lives of Allene Tew", "author": "Annejet van der Zijl, Michele Hutchison", "price": 149.00, "is_recommended": true},
    {"id": 4, "name": "The Great Alone: A Novel Kristin Hannah", "author": "Kristin Hannah", "price": 495.00, "is_recommended": true},
    {"id": 2, "name": "When Never Comes", "author": "Barbara Davis", "price": 179.00, "is_recommended": true},
    {"id": 1, "name": "Before We Were Yours: A Novel", "author": "Lisa Wingate", "price": 340.00, "is_recommended": false},
```

```
    {"id": 3, "name": "Giraffes Can't Dance", "author": "Giles Andreae, Guy Parker-Rees", "price": 200.50, "is_recommended": false},
    ...
  ]
}
```

Since the books information is updated, my result quite different from the requirement.

GET: <https://scb-test-book-publisher.herokuapp.com/books> shows a list of books from the publisher (normal + recommended). This list is updated on a weekly basis (every Sunday at midnight).

GET: <https://scb-test-book-publisher.herokuapp.com/books/recommendation> shows a list of recommended books from the publisher.

GETlocalhost:8000/books

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettings

BodyCookiesHeaders (5)Test Results

Status: 200 OKTime: 1793 msSize: 1.3 KB

PrettyRawPreviewVisualizeJSON

```
1  {
2    "books": [
3      {
4        "price": 149,
5        "id": 5,
6        "author_name": "Annejet van der Zijl, Michele Hutchison",
7        "book_name": "An American Princess: The Many Lives of Allene Tew"
8      },
9      {
10       "price": 495,
11       "id": 4,
12       "author_name": "Kristin Hannah",
13       "book_name": "The Great Alone: A Novel Kristin Hannah"
14     },
15     {
16       "price": 340,
17       "id": 1,
18       "author_name": "Lisa Wingate",
19       "book_name": "Before We Were Yours: A Novel"
20     },
21     {
22       "price": 200.5,
23       "id": 3,
24       "author_name": "Giles Andreae, Guy Parker-Rees",
25       "book_name": "Giraffes Can't Dance"
26     },
27     {
28       "price": 1379.78,
29       "id": 12,
30       "author_name": "British Library",
31       "book_name": "Harry Potter - A History of Magic"
32     },
33     {
34       "price": 290.06,
35       "id": 8,
36       "author_name": "Carol McCloud",
37       "book_name": "Have You Filled A Bucket Today?"
38     },
39     {
40       "price": 929.7,
41       "id": 6,
42       "author_name": "Anthony William",
43       "book_name": "Medical Medium Life-Changing Foods"
44     },
45     {
46       "price": 393.22,
47       "id": 11,
48       "author_name": "Kate Quinn",
49       "book_name": "The Girl on the Train"
```

```
49     "book_name": "The Alice Network"
50   },
51   {
52     "price": 319.16,
53     "id": 10,
54     "author_name": "Angie Thomas",
55     "book_name": "The Hate U Give"
56   },
57   {
58     "price": 208.51,
59     "id": 9,
60     "author_name": "Eric Carle",
61     "book_name": "The Very Hungry Caterpillar"
62   },
63   {
64     "price": 897.96,
65     "id": 7,
66     "author_name": "Gaz Oakley",
67     "book_name": "Vegan 100"
68   },
69   {
70     "price": 179,
71     "id": 2,
72     "author_name": "Barbara Davis",
73     "book_name": "The Alice Network"
```