

## Evidências da collection restful-booker API\_json server

Documentação/API de teste: [json-server-auth](#)

Referência de endpoints: [restful-booker](#)

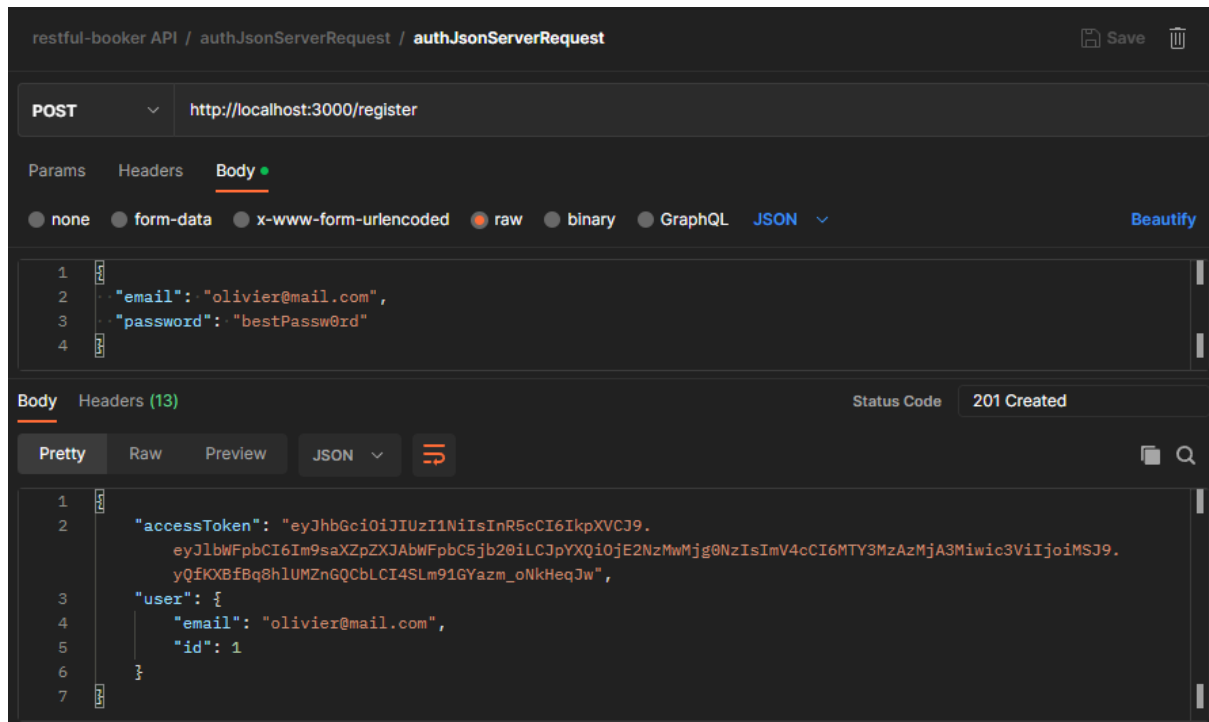
Postman: [Postman Project - DIO Workspace - Restful Booker API](#)

**Endpoints:**

- Auth json server request;
- Login json server;
- Route json server;
- Get user by id json server;
- Update json server;
- Partial update json server;
- Delete user json server.

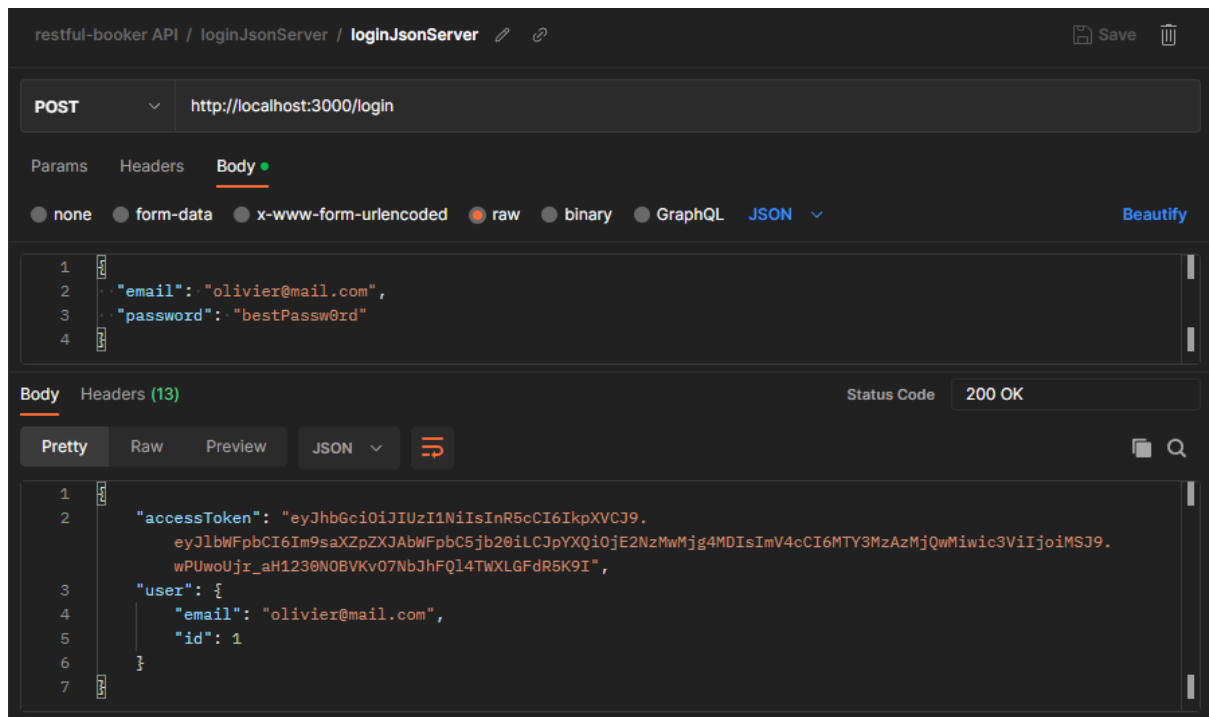
## POST authJsonServerRequest

Response: 201



## POST loginJsonServer

Response: 200



## GET routeJsonServer

Example 1

Response: 200

The screenshot shows a REST client interface for the 'routeJsonServer' API. The request is a GET to 'http://localhost:3000/users/1'. The 'Body' tab is selected, showing a JSON payload with two paths: '/users\*' pointing to '/600/users\$1' and '/messages\*' pointing to '/640/messages\$1'. The response status is '200 OK'. The response body is displayed in 'Pretty' JSON format, showing user details for ID 1.

```
1 {
2   "/users*": "/600/users$1",
3   "/messages*": "/640/messages$1"
4 }
```

Body Headers (13) Status Code 200 OK

Pretty Raw Preview JSON

```
1 {
2   "email": "olivier@mail.com",
3   "password": "$2a$10$b/v004c2flAM0xmImjsYeCg3nC0eQXSPo2PA4CMVl/kQ4n4zM8E.",
4   "id": 1
5 }
```

Example 2

Response: 401

The screenshot shows a REST client interface for the 'routeJsonServer' API. The request is a GET to 'http://localhost:3000/bookings'. The 'Body' tab is selected, showing a JSON payload with a path 'bookings' pointing to '600'. The response status is '401 Unauthorized'. The response body is displayed in 'Pretty' JSON format, showing an error message.

```
1 {
2   "...bookings": 600
3 }
```

Body Headers (13) Status Code 401 Unauthorized

Pretty Raw Preview JSON

```
1 "Incorrect authorization scheme"
```

## GET getUserByIdJsonServer

Example 1

Response: 200

The screenshot shows the REST Client interface for the endpoint `http://localhost:3000/users/2`. The request method is `GET`. The response status is `200 OK`. The response body is displayed in JSON format, showing a single user object with the following fields:

```
1 {
2   "email": "testejuliao1@mail.com",
3   "password": "$2a$10$dqC0j9FmH5pqya7E1HbNjeIneN1rPD0PiQxYhuJyTxu7rNF7owRFUm",
4   "id": 2
5 }
```

Example 2

Response: 404

The screenshot shows the REST Client interface for the endpoint `http://localhost:3000/users/50`. The request method is `GET`. The response status is `404 Not Found`. The response body is empty, indicating that no user was found for the specified ID.

## PUT updateJsonServer

Response: 200

restful-booker API / updateJsonServer / **updateJsonServer** Save Trash

**PUT** ▼ `http://localhost:3000/users/1`

Params Headers **Body** ●

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```
1 2
2  {
3    "email": "olivier@mail.com",
4    "password": "bestPassw0rd",
5    "firstname": "Olivier",
6    "lastname": "Monge",
7    "age": 32
8  }
```

Body Headers (13) Status Code **200 OK**

**Pretty** Raw Preview JSON ▼ ≡ 🔍

```
1 2
2  {
3    "email": "olivier@mail.com",
4    "password": "$2a$10$iZM7npQxjGKjkxcTEuCA5./zPjJlpne23LMhYX88JQesXcZv1f3vK",
5    "firstname": "Olivier",
6    "lastname": "Monge",
7    "age": 32,
8    "id": 1
9  }
```

## PATCH parcialUpdateJsonServer

Response: 200

restful-booker API / parcialUpdateJsonServer / **parcialUpdateJsonServer** Save Trash

**PATCH** ▼ `http://localhost:3000/users/1`

Params Headers **Body** ●

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☒ JSON Beautify

```
1 2
2  {
3    "email": "olivier@mail.com",
4    "password": "bestPassw0rd",
5    "firstname": "Olivier",
6    "lastname": "Monge",
7    "age": 42
8  }
```

Body Headers (13) Status Code **200 OK**

**Pretty** Raw Preview JSON ▼ ≡ 🔍

```
1 2
2  {
3    "email": "olivier@mail.com",
4    "password": "$2a$10$oX9Kc9DqeykC4X5SkVLipeSEnd8qzDwlxyHjuzikp3W/QxM0Wb56K",
5    "firstname": "Olivier",
6    "lastname": "Monge",
7    "age": 42,
8    "id": 1
9  }
```

*DELETE deleteUserJsonServer*

*Response: 200*

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

- URL:** `http://localhost:3000/users/3`
- Method:** `DELETE`
- Body:** A JSON object with the following structure:

```
{  "email": "testejulias2@mail.com",  "password": "PasswordTest"}
```
- Status Code:** `200 OK`
- Response Body:** An empty JSON object: `{}`