

DESAFIO WEBEDIA

Documentação API

ROTAS API AUTOR

CRIAR AUTOR (Cadastro)

url: <http://localhost:5000/api/autores/cadastro>

método: **POST**

campos obrigatórios: nome, e-mail, senha, senha2 (confirmação)

POST <http://localhost:5000/api/autores/cadastro> Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> nome	Erickson			
<input checked="" type="checkbox"/> email	teste@teste.com			
<input checked="" type="checkbox"/> senha	123456			
<input checked="" type="checkbox"/> senha2	123456			
<input type="text"/> New key	<input type="text"/> Value	<input type="text"/> Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1870 ms

Pretty Raw Preview JSON

```
1 {
2   "_id": "5be372896b15f007ac9f4999",
3   "nome": "Erickson",
4   "email": "teste@teste.com",
5   "senha": "$2a$10$t01fZ44YfVADfdFP8/bheM61bxT/XVwm/o05iRa4141XCG93Dovq",
6   "__v": 0
7 }
```

LOGIN AUTOR

url: <http://localhost:5000/api/autores/login>

método: **POST**

campos obrigatórios: e-mail, senha

O Login efetuado com sucesso, irá gerar um token, que poderá ser usado para realizar operações como: criar artigo, criar comentários e deletar comentários

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

- URL:** `http://localhost:5000/`
- Method:** `POST`
- Path:** `http://localhost:5000/api/autores/login`
- Form:** `x-www-form-urlencoded`
- Body:**

Key	Value	Description
<input checked="" type="checkbox"/> email	teste15@teste.com	
<input checked="" type="checkbox"/> senha	123456	
New key	Value	Description
- Response:**

```
{  "success": true,  "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVlZTM4MmM1NmFjMTIhMwUzOGI2YTU0NSIsIm5vbmUiOiJ1aGFpcyIsIm1hdCI6MTU0MTYzNjgzNywiZXhwIjoxNTQxNjQwNDM3fQ.BA_2CTpPwtFbV2QXeBvFV7XdbRMw91KpmqITekTNvic"}
```
- Status:** 200 OK
- Time:** 220 ms

LISTA DE AUTORES

url: <http://localhost:5000/api/autores>

método: **GET**

Irá retornar todos os autores cadastrados, com paginação.

Passar parâmetro 'porPagina', para obter a quantidade de docs por página. Passar parâmetro 'pagina', para escolher qual pagina que deverá ser exibida.

GET http://localhost:5000/api/autores?pagina=2&porPagina=3 Params Send Save

Body Cookies Headers (8) Test Results Status: 200 OK Time: 67 ms

Pretty Raw Preview JSON

```
1 {
2   "docs": [
3     {
4       "_id": "5be382346ac19a1e38b6a539",
5       "nome": "Fabio",
6       "email": "teste3@teste.com",
7       "senha": "$2a$10$p2IrTAYZ14yJ04hj60DeJOeH3u.fE/CuXHkXQAw5sNk5/6MLX00FO",
8       "__v": 0
9     },
10    {
11      "_id": "5be3823a6ac19a1e38b6a53a",
12      "nome": "Marco",
13      "email": "teste4@teste.com",
14      "senha": "$2a$10$iokeX08crk17jV7JwN0210LRcQjt3MtJkn.ayouJdtYPHyFRfp91i",
15      "__v": 0
16    },
17    {
18      "_id": "5be382456ac19a1e38b6a53b",
19      "nome": "Helio",
20      "email": "teste5@teste.com",
21      "senha": "$2a$10$TbR27/nwbEc2T9v89..LR0zEi1x01x50.h41oZ52Xx4mwG33oJWrW",
22      "__v": 0
23    }
24  ],
25  "total": 17,
26  "limit": 3,
27  "page": 2,
28  "pages": 6
29 }
```

ATUALIZANDO AUTOR

url: <http://localhost:5000/api/autores/:id>

método: PUT

Passar id do autor por parâmetro.

Dados que poderão ser atualizados: nome e e-mail.

PUT http://localhost:5000/api/autores/5be3823a6ac19a1e38b6a53a Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> nome	Marco Modificado			
<input checked="" type="checkbox"/> email	modificado@teste.com			
<input type="text"/> New key	<input type="text"/> Value	<input type="text"/> Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 68 ms

Pretty Raw Preview JSON

```
1 {
2   "_id": "5be3823a6ac19a1e38b6a53a",
3   "nome": "Marco Modificado",
4   "email": "modificado@teste.com",
5   "senha": "$2a$10$iokeX08crk17jV7JwN0210LRcQjt3MtJkn.ayouJdtYPHyFRfp91i",
6   "__v": 0
7 }
```

DELETANDO AUTOR

url: <http://localhost:5000/api/autores/:id>

método: **DELETE**

Passar id do autor por parâmetro.

FAZER LOGIN E PASSAR HEADER 'AUTHORIZATION' COM O VALOR DO TOKEN GERADO PÓS LOGIN COM SUCESSO

DELETE <http://localhost:5000/api/autores/5be3823a6ac19a1e38b6a53a> Params Send Save

Authorization Headers Body Pre-request Script Tests Code

Key	Value	Description
New key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 146 ms

Pretty Raw Preview JSON

```
1 {  
2   "msg": "Autor excluido com sucesso"  
3 }
```

ROTAS API ARTIGOS

CRIAR NOVO ARTIGO

url: <http://localhost:5000/api/artigos>

method: **POST**

campos obrigatórios: titulo, subtítulo e conteúdo

FAZER LOGIN E PASSAR HEADER 'AUTHORIZATION' COM O VALOR DO TOKEN GERADO PÓS LOGIN COM SUCESSO

Passando token

POST <http://localhost:5000/api/artigos> Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

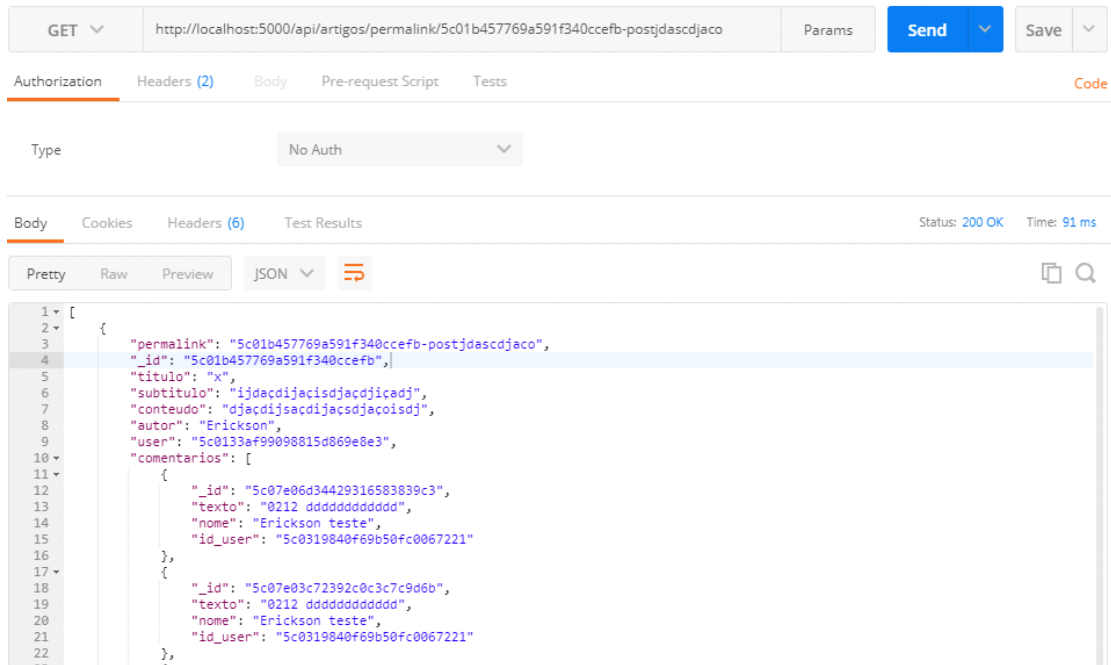
Key	Value	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjVlZTM3Mjg5NmlxNWYwMDdhYzlmNDk5OjI0MTYzNzgxOCwiZXhwIjoxNTQxNjQxNDE4fQ.MRp0GT0DgftuN4zcTBQFdQv1xrK2p0B1-c-nM5tsQdd	Description
New key		Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 146 ms

BUSCAR ARTIGO POR PERMALINK

url: <http://localhost:5000/api/artigos/permalink/:permalink>

método: **GET**



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

- Method:** GET
- URL:** `http://localhost:5000/api/artigos/permalink/5c01b457769a591f340ccefb-postjdascdjaco`
- Params:** (empty)
- Authorization:** No Auth
- Status:** 200 OK
- Time:** 91 ms
- Body:** JSON (Pretty)

```
[
  {
    "permalink": "5c01b457769a591f340ccefb-postjdascdjaco",
    "_id": "5c01b457769a591f340ccefb",
    "titulo": "x",
    "subtitulo": "ijdaçdiçaisdjaçdjiçadj",
    "conteudo": "djaçdijsaçdijsdjaçoisdj",
    "autor": "Erickson",
    "user": "5c0133af99098815d869e8e3",
    "comentarios": [
      {
        "_id": "5c07e06d34429316583839c3",
        "texto": "0212 ddddddddddd",
        "nome": "Erickson teste",
        "id_user": "5c0319840f69b50fc0067221"
      },
      {
        "_id": "5c07e03c72392c0c3c7c9d6b",
        "texto": "0212 ddddddddddd",
        "nome": "Erickson teste",
        "id_user": "5c0319840f69b50fc0067221"
      }
    ]
  }
]
```

ATUALIZAR ARTIGO

url: <http://localhost:5000/api/artigos/:id>

método: **PUT**

Passar por parâmetro o id do artigo a ser atualizado. Poderá atualizar o título, subtítulo e conteúdo.

Passar por parâmetro o id do artigo. O retorno será os comentários contidos nele.

GET <http://localhost:5000/api/comentarios/5be3877808a60019c870529e> Params Send Save

Authorization Headers Body Pre-request Script Tests Code

Type No Auth

Body Cookies Headers (8) Test Results Status: 200 OK Time: 48 ms

Pretty Raw Preview JSON

```
1 {
2   "_id": "5be3877808a60019c870529e",
3   "comentarios": [
4     {
5       "_id": "5be38a1a08a60019c87052a9",
6       "texto": "Adicionando mais um comentario",
7       "nome": "Thais"
8     },
9     {
10      "_id": "5be38a0408a60019c87052a8",
11      "texto": "Adicionando comentario",
12      "nome": "Thais"
13    }
14  ]
15 }
```

ATUALIZAR COMENTARIO

url: http://localhost:5000/api/comentarios/:comentario_id

método: PUT

Passar por parâmetro o id do comentário a ser atualizado. Inserir no body, o campo texto. FAZER LOGIN E PASSAR HEADER 'AUTHORIZATION' COM O VALOR DO TOKEN GERADO PÓS LOGIN COM SUCESSO

PUT <http://localhost:5000/api/comentarios/5be38a1a08a60019c87052a9> Params

Authorization Headers (1) Body Pre-request Script Tests

☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary

Key	Value	Description
<input checked="" type="checkbox"/> texto	Modificando comentario	
New key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {
2   "n": 1,
3   "nModified": 1,
4   "ok": 1
5 }
```

DELETAR COMENTARIO DE UM ARTIGO ESPECIFICO

url: <http://localhost:5000/api/comentarios/:artigoId/:comentarioid>

method: **DELETE**

Passar por parâmetro o id do artigo e do comentário a ser deletado.

FAZER LOGIN E PASSAR HEADER 'AUTHORIZATION' COM O VALOR DO TOKEN GERADO PÓS LOGIN COM SUCESSO

The screenshot displays a REST client interface with the following components:

- Method and URL:** A dropdown menu shows 'DELETE' and the URL is 'http://localhost:5000/api/comentarios/5be38d4c0b4bf51fbcebe422/5be391120335b1169039367f'.
- Params:** A button labeled 'Params' is next to the URL.
- Send Button:** A blue button labeled 'Send' with a dropdown arrow.
- Headers Tab:** The 'Headers (1)' tab is active, showing a table with one header: 'Authorization' with a value starting with 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Ij...'. Below the table are input fields for 'New key', 'Value', and 'Description'.
- Body Tab:** The 'Body' tab is active, showing a JSON response in 'Pretty' format:

```
{  "msg": "Comentario excluido"}
```

. The status is '200 OK'.