

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			
New key	Value	Description		

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

Pretty Raw Preview JSON

```
1 {
2   "_id": "5be372896b15f007ac9f4999",
3   "name": "Erickson",
4   "email": "teste@teste.com",
5   "senha": "$2a$10$0t0lFZ44YfVADfdFP8/bheMG1bxT/XVwm/o05iRa414lXCG93Dovq",
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

http://localhost:5000/ + ... No Environment

POST http://localhost:5000/api/autores/login Params Send Save

Authorization Headers (1) Body Pre-request Script Tests

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

Key	Value	Description	...	Bulk E
<input checked="" type="checkbox"/> email	teste15@teste.com			
<input checked="" type="checkbox"/> senha	123456			
New key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 220

Pretty Raw Preview JSON

```

1 {
2   "success": true,
3   "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
4     .eyJpZCI6IjVlZTM4MmM1NmFjMTlhMTU0NSIsIm5vbmUiOiJ1aGFpcyIsIm1hdCI6MTU0MTYzNjgzNywiZXhwIjoxNTQxNjQwNDM3fQ
5     .BA_2CTpPwtFbV2QXeBvFV7XdbRMw91KpmqITeKTNvic"
6 }

```

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/CuXHKXQAw5sNkS/6MLX00F0",
8       "__v": 0
9     },
10    {
11      "_id": "5be3823a6ac19a1e38b6a53a",
12      "nome": "Marco",
13      "email": "teste4@teste.com",
14      "senha": "$2a$10$iokeX08crk17jV7JwNO21OLRcQjt3MtJkn.ayouJdtYPHyFRfp91i",
15      "__v": 0
16    },
17    {
18      "_id": "5be382456ac19a1e38b6a53b",
19      "nome": "Helio",
20      "email": "teste5@teste.com",
21      "senha": "$2a$10$TbR27/nwbEc2T9v89...LR0zEi1x01x50.h41oZS2Xx4wwG33oJWrW",
22      "__v": 0
23    }
24  ],
25   "total": 17,
26   "limit": 3,
27   "page": 2,
28   "pages": 6

```

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.

The screenshot shows a REST client interface with a PUT request to `http://localhost:5000/api/autores/5be3823a6ac19a1e38b6a53a`. The request body is in x-www-form-urlencoded format. The response status is 200 OK.

Key	Value	Description
nome	Marco Modificado	
email	modificado@teste.com	
New key	Value	Description

```
{  "_id": "5be3823a6ac19a1e38b6a53a",  "nome": "Marco Modificado",  "email": "modificado@teste.com",  "senha": "$2a$10$iokeX0Bcrk17jV7JwN021OLRcQjt3MtJkn.ayouJdtYPhyFRfp91i",  "__v": 0}
```

DELETANDO AUTOR

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

método: **DELETE**

Passar id do autor por parâmetro.

The screenshot shows a REST client interface with a DELETE request to `http://localhost:5000/api/autores/5be3823a6ac19a1e38b6a53a`. The response status is 200 OK.

```
{  "msg": "Autor excluido com sucesso"}
```

ROTAS API ARTIGOS

CRIAR NOVO ARTIGO

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

method: **POST**

campos obrigatórios: título, 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

Send

Save

AuthorizationHeaders (1)BodyPre-request ScriptTestsCode

form-data

x-www-form-urlencodedrawbinary

Key	Value	Description
titulo	Artigo Teste	
subtitulo	teste	
conteudo	testetestestetestetestetetetestetetetestetetetestetet...	
New key	Value	Description

Body

Cookies

Headers (8)

Test Results

Status: 200 OKTime: 840 ms

PrettyRawPreviewJSONCopyDownload

```
1 {  
2   "id": "5be3872f08ae0019c870529d",  
3   "titulo": "Artigo Teste",  
4   "subtitulo": "teste",  
5   "conteudo": "testetestestetestetestetetetestetetetestetetetestetetetestetetetestetetetestetetetestetetetestetestes  
               testetestestetestetestetetetestetetetestetetetestetetetestetetetestetetetestetetetestetetetestetete...",  
6   "autor": "Erickson",  
7   "comentarios": [],  
8   "data_criacao": "2018-11-08T00:45:35.888Z",  
9   "data_atualizacao": "2018-11-08T00:45:35.891Z",  
10  "__v": 0,  
11 }
```

LISTA DE ARTIGOS

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

método: **GET**

Irá retornar todos os artigos, com paginação.

Passar parâmetro 'porPagina', para obter a quantidade de docs, por página.

Passar parâmetro 'pagina', para escolher qual pagina que deverá ser exibida.

método: DELETE

Passar por parâmetro o id do artigo a ser deletado.

The screenshot shows the Postman interface with the following details:

- Method:** DELETE
- URL:** http://localhost:5000/api/artigos/5be3872f08a60019c870529d
- Params:** (Empty)
- Authorization:** No Auth
- Response Status:** 200 OK
- Response Time:** 53 ms
- Response Body (JSON):**

```
{
  "msg": "Artigo excluido com sucesso"
}
```

ROTAS API COMENTARIOS

CRIAR COMENTARIO

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

método: POST

Passar por parâmetro o id do artigo e inserir no body, o campo texto.

POST http://localhost:5000/api/comentarios/5be3877808a60019c870529e

Authorization Headers Body Pre-request Script Tests

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

Key	Value	Description
texto	Adicionando mais um comentario	
New key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 101 ms

Pretty Raw Preview JSON

```
{
  "_id": "5be3877808a60019c870529e",
  "titulo": "Artigo Teste",
  "subtitulo": "teste",
  "conteudo": "testetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestetestest",
  "autor": "Chico",
  "comentarios": [
    {
      "_id": "5be38a1a08a60019c87052a9",
      "texto": "Adicionando mais um comentario",
      "nome": "Thais"
    },
    {
      "_id": "5be38a0408a60019c87052a8",
      "texto": "Adicionando comentario",
      "nome": "Thais"
    }
  ],
  "data_criacao": "2018-11-08T00:46:48.487Z",
  "data_atualizacao": "2018-11-08T00:58:02.627Z",
  "__v": 2
}
```

BUSCAR COMENTARIO

url: <http://localhost:5000/api/comentarios/comentario/:artigold>

método: GET

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

The screenshot shows a REST client interface with a GET request to `http://localhost:5000/api/comentarios/5be3877808a60019c870529e`. The response is a JSON array of two comments, each with an ID, text, and name.

```
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

The screenshot shows a REST client interface with a PUT request to `http://localhost:5000/api/comentarios/5be38a1a08a60019c87052a9`. The body is set to `x-www-form-urlencoded` and contains a `texto` field with the value `Modificando comentario`. The response is a JSON object with status fields.

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

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

DELETAR COMENTARIO DE UM ARTIGO ESPECIFICO

url: <http://localhost:5000/api/comentarios/:artigold/:comentariold>

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 set to 'DELETE' and a text field containing the URL `http://localhost:5000/api/comentarios/5be38d4c0b4bf51fbcebe422/5be391120335b1169039367f`.
- Params:** A button labeled 'Params' and a 'Send' button.
- Headers:** A tab labeled 'Headers (1)' is active, showing a table with one header row and one data row.

Key	Value	Description
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Ij...	
- Body:** A tab labeled 'Body' is active, showing a JSON response in 'Pretty' format.

```
1 {
2   "msg": "Comentario excluido"
3 }
```
- Status:** A label indicating 'Status: 200 OK'.