

00 LOJA

Início

Criar servidor mongo de acordo com as informacoes oficiais do mongo no docker hub

https://hub.docker.com/_/mongo/

SEEDS

Não fiz as seeds por código, os passwords estão hasheados para a key padrao

PRIVATE_TOKEN=A1B2C3D4E5, caso não utilizar esse terá que debugar o login e forcar o hash em algum password que escolher e editar na base.

```
db.users.insertMany(
[
  {
    email: 'admin@admin.com',
    password: '4c86b96fb4621fcdd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e',
    courses: [],
    admin: true,
    status: true,
    created_at: null,
    updated_at: null,
  },
  {
    email: 'user@user.com',
    password: 'user',
    courses: [],
    admin: false,
    status: true,
    created_at: null,
    updated_at: null,
  }
]
)

db.users.updateMany(
{
  $or:
  [
    {
      email: 'admin@admin.com'
```

```

        email: 'user@user.com'
      }
    ]
  }
},
{
  $currentDate:
  {
    created_at: {$type: 'date'},
    updated_at: {$type: 'date'},
  }
}
)

db.users.createIndex({admin:1,status:1})
db.users.createIndex({email:1}, { unique: true })

db.courses.insertMany(
[
  {
    name: 'Curso de React',
    slug: 'react-course',
    users:[],
    lessons:[],
    status:true,
    released_at: null,
    created_at: null,
    updated_at: null,
  },
  {
    name: 'Curso de Typescript',
    slug: 'typescript-course',
    users:[],
    lessons:[],
    status:true,
    released_at: null,
    created_at: null,
    updated_at: null,
  },
]
)

db.courses.updateMany(
{
  $or:
  [
    {

```

```

        slug: 'react-course'
      },
      {
        slug: 'typescript-course'
      }
    ]
  }
},
{
  $currentDate:
  {
    created_at: {$type: 'date'},
    updated_at: {$type: 'date'},
    released_at: {$type: 'date'},
  }
}
)

db.courses.createIndex({released_at:1,status:1})
db.courses.createIndex({slug:1}, { unique: true })

```

Modelo .env

```

PORT=3001
TOKEN_EXPIRATION_TIME=20
PRIVATE_TOKEN=A1B2C3D4E5
MONGO_DB=mongodb://localhost/zerozeroloja
AWS_PUBLIC_TOKEN=AKIA42Y6GSL6YIMV36N7
AWS_PRIVATE_TOKEN=fFkA2X8AhbPDVoQX4BhX9RofdjoiYknMeYTCmjlx
AWS_REGION=us-east-1
AWS_REGISTERED_EMAIL=jonas@grr.la

```

EndPoints

Base

Health Check

BodyCookiesHeaders (7)Test Results

PrettyRawPreviewVisualize

JSON ⌵⌵

1{

2 "name": "00academy",

3 "version": "1.2.1",

4 "author": "joneco"

5}

Login

The screenshot shows the Postman interface for a POST request to `localhost:3001/auth`. The 'Body' tab is selected, and the request body is a JSON object: `{ "email": "admin@admin.com", "password": "admin" }`. The 'JSON' format is chosen from the dropdown menu. The bottom bar shows 'Body', 'Cookies', 'Headers (7)', and 'Test Results' tabs, with 'Body' being the active tab. The 'Pretty' button is also visible.

Verify Token

00academy / verify_token

GET

localhost:3001/auth/validate

Params

Authorization ●

Headers (9)

Body ●

Pre-request Script

Tests

Settings

Type

Bearer Token ▼

The authorization header will be automatically generated when you send the request. [Learn more about authorization ↗](#)

ⓘ Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborator, we've redacted some values. [Learn more about variables ↗](#)

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm9udGVudCI6ImFkbWVudCJ9.eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm9udGVudCI6ImFkbWVudCJ9.

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

🔧

1

"sucess"

Cursos

Listar Todos

Apenas para fins de estudo de comportamento / comparação essa rota ficou desprotegida, não sendo necessário por o token

GET localhost:3001/courses

Params Authorization Headers (6) Body Pre-request Script

Type Bearer Token

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Token

body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "id": "633388ab6bca2a4e5deba6a1",
4     "name": "Curso de React",
5     "slug": "react-course",
6     "status": true,
7     "students": [],
8     "lessons": [],
9     "released_at": "2022-09-27T23:35:32.334Z",
10    "updated_at": "2022-09-27T23:35:32.334Z",
11    "created_at": "2022-09-27T23:35:32.334Z"
12  },
13  {
14    "id": "633388ab6bca2a4e5deba6a2",
15    "name": "Curso de Typescript",
16    "slug": "typescript-course",
17    "status": true,
18    "students": [],
19    "lessons": [],
20    "released_at": "2022-09-27T23:35:32.334Z",
21    "updated_at": "2022-09-27T23:35:32.334Z",
22    "created_at": "2022-09-27T23:35:32.334Z"
23  }
24 ]
```

Listar um

Também desprotegida

GET

localhost:3001/courses/633388ab6bca2a4e5deba6a1

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Query Params

	KEY
	Key

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "id": "633388ab6bca2a4e5deba6a1",
3    "name": "Curso de React",
4    "slug": "react-course",
5    "status": true,
6    "students": [],
7    "lessons": [],
8    "released_at": "2022-09-27T23:35:32.334Z",
9    "updated_at": "2022-09-27T23:35:32.334Z",
10   "created_at": "2022-09-27T23:35:32.334Z"
11 }
```

Criar

Erro de token (funcionando para todas as rotas protegidas por bearer token)

POST

localhost:3001/courses/

Params

Authorization ●

Headers (9)

Body ●

● none

● form-data

● x-www-form-urlencoded

```
1
2  {
3    "name": "teste",
4    "slug": "curso-teste",
5    "status": true,
6    "students": ["633240540e8c5d1aff54c
7    "released_at": "2022-09-02"
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1 "Token Expired"

POST

localhost:3001/courses/

Params

Authorization ●

Headers (8)

Body ●

Type

Bearer Token

The authorization header will be automatically generated w
you send the request. [Learn more about authorization](#)

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1 "No Token provided"

Editar

Editando informações e integrantes dentro do curso

PUT

localhost:3001/courses/633388ab6bca2a4e5deba6a1

Params

Authorization ●

Headers (9)

Body ●

Pre-request Script

Tests

Settings

● none

● form-data

● x-www-form-urlencoded

● raw

● binary

● GraphQL

JSON ▼

```
1
2  {
3    "name": "teste de update 3",
4    "slug": "teste-update3",
5    "status": false,
6    "students": ["633240540e8c5d1aff54ce16", "633240540e8c5d1aff54ce17", "633240540e8c5d1aff54ce17", "633240540e8c5d1aff54ce17"],
7    "released_at": "2022-09-02"
```

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON ▼

≡

```
1
2  {
3    "id": "633388ab6bca2a4e5deba6a1",
4    "name": "teste de update 3",
5    "slug": "teste-update3",
6    "status": false,
7    "students": [
8      "633240540e8c5d1aff54ce16",
9      "633240540e8c5d1aff54ce17"
10   ],
11   "lessons": [],
12   "released_at": "2022-09-02T00:00:00-03:00Z",
13   "updated_at": "2022-09-27T23:35:32.334Z",
14   "created_at": "2022-09-27T23:35:32.334Z"
```


Deletar

DELETE

localhost:3001/courses/633388ab6bca2a4e5deba6a1

Params

Authorization ●

Headers (7)

Body

Pre-request Script

Tests

Type

Bearer Token

! Heads up! T

Learn more

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Token

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1 "Course ID 633388ab6bca2a4e5deba6a1 successfully deleted"

Usuários

Listar Todos

00academy / users / list-users

GET

localhost:3001/users

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Query Params

KEY

Key

ody

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "id": "633240540e8c5d1aff54ce16",
4     "email": "admin@admin.com",
5     "password": "4c86b96fb4621fcdd6eff7cd863cf4052b35027",
6     "admin": true,
7     "status": true
8   },
9   {
10    "id": "633240540e8c5d1aff54ce17",
11    "email": "joneco@hotmail.com",
12    "password": "user",
13    "admin": false,
14    "status": true
15  },
16  {
17    "id": "63365beea4032fdd51011360",
18    "email": "jonasssss@hotmail.com",
19    "password": "4c86b96fb4621fcdd6eff7cd863cf4052b35027",
20    "admin": false,
21    "status": true
22  }
23 }
```

Listar um

GET

localhost:3001/users/633240540e8c5d1aff54ce16

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

```
"id": "633240540e8c5d1aff54ce16",
"email": "admin@admin.com",
"password": "4c86b96fb4621fcdd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e",
"admin": true,
"status": true
```

Pesquisa

Personalizada

Coloquei um exemplo de filtrar qualquer campo de determinado dado

GET

localhost:3001/users/search?email=admin@admin.com

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	email	admin@admin.com
	Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "id": "633240540e8c5d1aff54ce16",
4     "email": "admin@admin.com",
5     "password": "4c86b96fb4621fcdd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e",
6     "admin": true,
7     "status": true
8   }
9 }
```

GET

localhost:3001/users/search?admin=false

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	admin	false
	Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   {
3     "id": "633240540e8c5d1aff54ce17",
4     "email": "joneco@hotmail.com",
5     "password": "user",
6     "admin": false,
7     "status": true
8   },
9   {
10    "id": "63365beea4032fdd51011360",
11    "email": "jonasssss@hotmail.com",
12    "password": "4c86b96fb4621fcdd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e",
13    "admin": false,
14    "status": true
15  },
16  {
17    "id": "63365d90185cfbbdff23520e",
18    "email": "jonasssss@hotmail.com",
19    "password": "4c86b96fb4621fcdd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e",
20    "admin": false,
21    "status": true
22  }
23 }
```

Criar (ENVIA EMAIL PELO AMAZON SES, Demonstracao + a frente)

POST localhost:3001/users/

Params Authorization Headers (9) **Body** Pre-rec

none form-data x-www-form-urlencoded raw

```
1 {
2   "email": "jonasssss@hotmail.com",
3   "password": "admin",
4   "admin": false,
5   "status": true
6 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text

```
1 Error: Error: Email already Exists
```

POST localhost:3001/users/

Params Authorization Headers (9) **Body** Pre-rec

none form-data x-www-form-urlencoded raw

```
1 {
2   "email": "testeFinal@hotmail.com",
3   "password": "testee",
4   "admin": false,
5   "status": true
6 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 true
```

Editor

PUT

localhost:3001/users/63365d90185cfbbdff23520e

Params

Authorization

Headers (9)

Body

Pre-request Script

Tests

Settings

Query Params

KEY	VALUE
Key	Value

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

```
"id": "63365d90185cfbbdff23520e",
"email": "jonecao@hotmail.com",
"password": "4c86b96fb4621fcd6eff7cd863cf4052b3502731c9afa57fd337c9a85ba0a3e",
"admin": false,
"status": true
```

Deletar

DELETE

localhost:3001/users/63365d90185cfbbdff23520e

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Type

Bearer Token

! Heads up!

Learn more

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Token

Body

Cookies

Headers (7)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

"User ID 63365d90185cfbbdff23520e successfully deleted"

Envio de Email pelo SES

Coloquei para enviar um email apenas quando um usuário é criado.

O email está no .env, coloquei um email temporario que pode facilitar a verificação.



Para verificar, é só ir no <https://www.guerrillamail.com> e setar o alias do email como mencionado.

Clique no nome antes do @, sete como jonas e selecione o dominio como grr.la



The screenshot shows the Guerrilla Mail interface. At the top, there is a text input field containing the alias 'jonas'. To its right are 'Set' and 'cancel' buttons. Further right is an '@' symbol followed by a dropdown menu currently showing 'grr.la'. To the right of the dropdown are 'Forget Me' and 'WTF?' buttons. Below the main input area, a generated email address 'qi8gya+6jstrcos@grr.la' is displayed next to a checked checkbox and a 'Scramble Address' button. At the bottom of the interface, there is a row of four buttons: 'EMAIL', 'COMPOSE', 'TOOLS', and 'ABOUT'.

agora receberá email. Como o SES exige um email ou verificacao de um dominio do site, coloquei esse email do guerrila pois facilita o avaliador.