

# 00 Academy

## Plataforma de Cursos

1 importar no postman a collection que se encontra na raiz do projeto.

Nela teremos os endpoints.

Registraremos cursos e alocaremos usuarios aos cursos. Todas as rotas são protegidas, menos login e healthcheck. Algumas rotas somente usuario com o role de admin poderá utilizar.

## Endpoints

### Geral

health\_check – verifica o funcionamento e versionamento do serviço,

Login – utilizado para logar

usuario mockado:

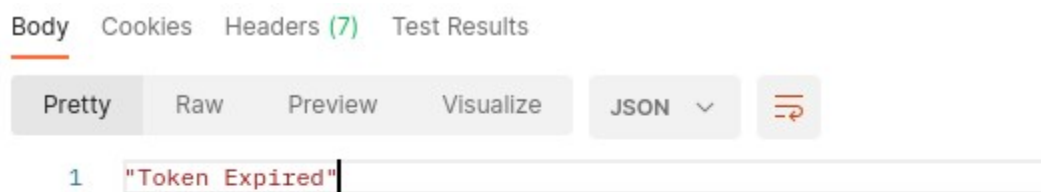
```
{  
  "email": "user@user.com",  
  "password": "password"  
}
```

VerifyToken – rota utilizada no inicio para construir o middleware de verificacao. No env há o tempo de expiracao do token em MINUTOS. Por default está 20minutos

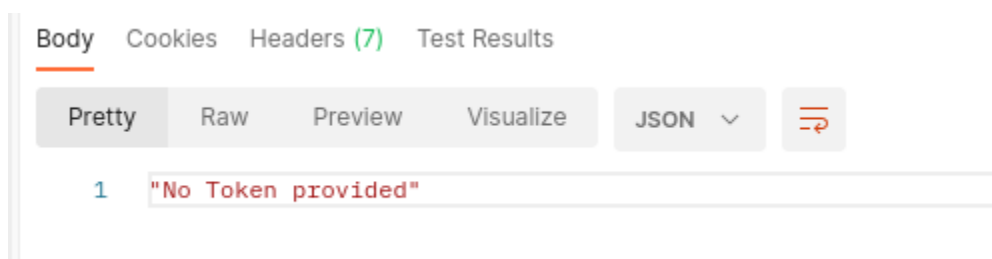
**USAR BEARER TOKEN**

TOKEN\_EXPIRATION\_TIME=20

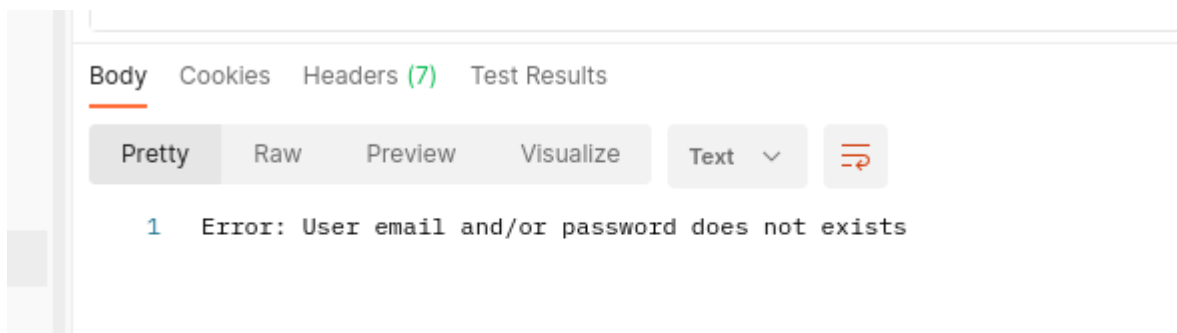
Exemplo token expirado:



Exemplo requisicao sem token:



Exemplo um token valido criado anteriormente, porém o usuario nao existe mais na base:



## Users

Create [POST] (ADMIN) – cria um usuario

```
{  
  "email": "joneco@hotmail.com",  
  "password": "flathebest",  
  "admin": false,  
  "status": true,  
}
```

List [GET] – lista todos usuarios na base

não há body

Search [GET] – procura um usuario pelo id

não há body

Update [PUT] (ADMIN) - edita um usuario pelo id, todos os campos sao opcionais e podem ser suprimidos, sendo necessario apenas informar os que deseja realmente mudar

```
{  
  "email": "joneco@hotmail.com",  
  "password": "flathebest",  
  "admin": false,  
  "status": true  
}
```

delete [DELETE] (ADMIN) – delete um usuario pelo ID

não há body

## Courses

Um curso pode ter nenhum ou um ou vários alunos inscritos.

Create [POST] (ADMIN) – cria um novo curso e aloca ou não estudantes nele.

```
{
  "descricao": "teste2",
  "dataInicio": "2022-09-02",
  "status": true,
  "students": [1,2,3]
}
RETURN:
{
  "id": 1,
  "descricao": "teste2",
  "dataInicio": "2022-09-02",
  "status": true,
  "students":
  [
    {
      "id": 1,
      "email": "user@user.com",
      "password": "password",
      "admin": true,
      "status": true
    },
    {
      "id": 2,
      "email": "joneco@hotmail.com",
      "password": "flathebest",
      "admin": false,
      "status": true
    },
    {
      "id": 3,
      "email": "jonecos@hotmail.com",
      "password": "flathebest",
      "admin": false,
      "status": true
    }
  ]
}
```

List [GET] – lista todos os cursos na base com seus alunos

não há body

RETURN: mesmo de cima

Search [GET] – procura um curso pelo id e lista as infos e seus alunos

RETURN: mesmo de cima

Update [PUT] (ADMIN) - edita um curso pelo id, todos os campos são opcionais e podem ser suprimidos, sendo necessário apenas informar os que deseja realmente mudar. Para remover alunos imputar array vazio, para editar inputar array com o id de todos os alunos presentes no curso

```
{  
  "descricao": "teste2",  
  "dataInicio": "2022-09-02",  
  "status": false,  
  "students": []  
}
```

delete [DELETE] (ADMIN) – delete um curso pelo ID

não há body

TODO – faltou rotas / usecases mais específicos tipo remover apenas um aluno do curso ou adicionar apenas um aluno do curso. Faltou colocar lessons/aulas nos cursos com a duração de cada e um outro model de progressão de aluno no curso.