



Quem se prepara, não para.

Arquitetura de Aplicações Web

5º período

Professora: Michelle Hanne

Sumário

- JSON JWT

Passo a Passo

1) Instalar o Postman, caso não tenha
– <https://www.postman.com/downloads/>

2) Criar uma pasta com o nome **Exemplo_JSONWebToken**, acessar o VS Code e incluir a pasta no Workspace. Abrir o terminal e iniciar o projeto:

```
npm init -y
```

Passo a Passo

3) Instalar o Express e o body-parser
npm i express body-parser

4) Criando os arquivo com as variáveis de ambiente:

.env.example

.env.example, commit **to** repo

SECRET=

.env

#.env, don't commit to repo

SECRET=mysecret

5) Instalar o Express e o body-parser

```
npm i express body-parser
```

6) Instalando o JSON Web Token

jsonwebtoken: pacote que implementa o protocolo JSON Web Token;

```
npm i jsonwebtoken
```

7) Criando o arquivo index.js

```
const http = require('http');
const express = require('express');
const app = express();
const server = http.createServer(app);

require("dotenv-safe").config();
const jwt = require('jsonwebtoken');

const bodyParser = require('body-parser');
app.use(bodyParser.json());

app.get('/', (req, res, next) => {
  res.json({message: "Tudo ok por aqui!"});
})

app.get('/clientes', verifyJWT, (req, res, next) => {
  console.log(req.userId + ' autenticado');
  res.json([{"id":1,nome:'Michelle Hanne'}]);
})
```

Passo a Passo

7) Criando o arquivo index.js

```
//authentication
app.post('/login', (req, res, next) => {
  //esse teste abaixo deve ser feito no seu banco de dados
  if(req.body.user === 'arquiteturaWeb' && req.body.password === '123'){
    //auth ok
    const id = 1; //esse id viria do banco de dados
    const token = jwt.sign({ id }, process.env.SECRET, {
      expiresIn: 300 // expires in 5min
    });
    return res.json({ auth: true, token: token });
  }

  res.status(500).json({message: 'Login inválido!'});
})
```


Passo a Passo

```
app.post('/logout', function(req, res) {  
  res.json({ auth: false, token: null });  
})
```

```
function verifyJWT(req, res, next){  
  const token = req.headers['x-access-token']; // cabeçalho para conter o token  
  if (!token) return res.status(401).json({ auth: false, message: 'No token provided.' });  
  
  jwt.verify(token, process.env.SECRET, function(err, decoded) {  
    if (err) return res.status(500).json({ auth: false, message: 'Failed to authenticate token.' });  
  
    // se tudo estiver ok, salva no request para uso posterior  
    req.userId = decoded.id;  
    next();  
  });  
}
```

```
server.listen(3000);  
console.log("Servidor escutando na porta 3000...")
```

Colocar para executar o arquivo:

node index.js

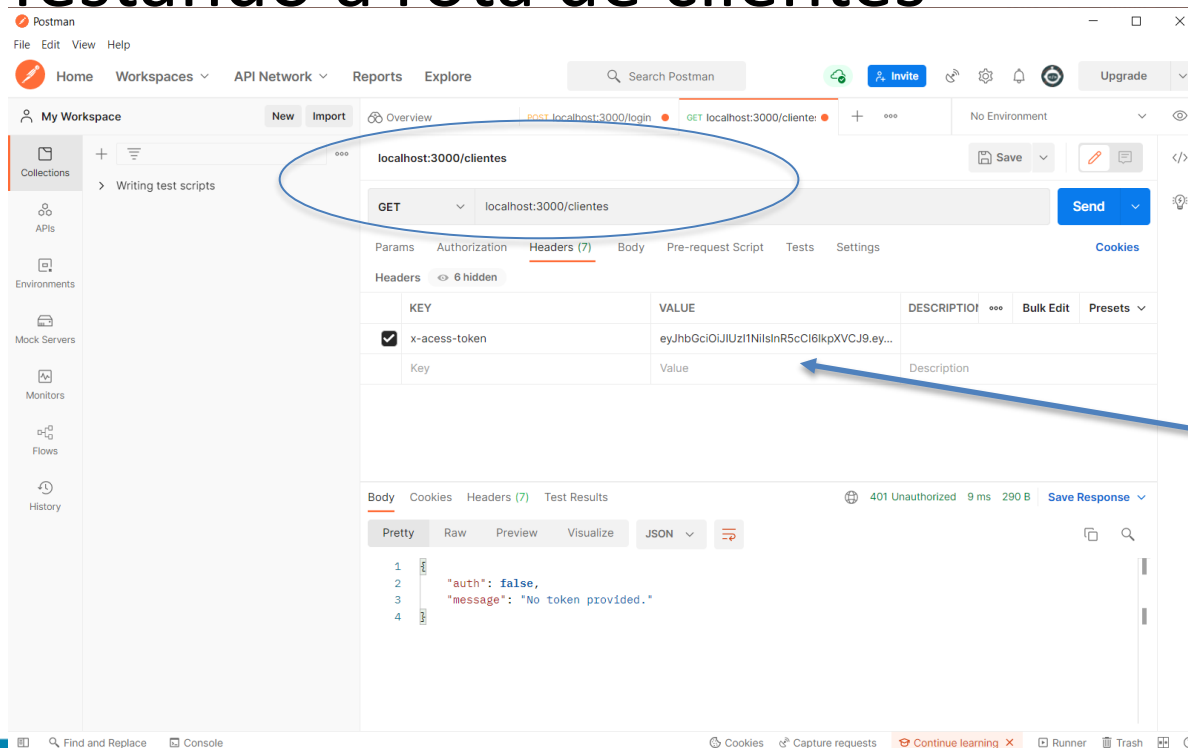
[illegible]

Token gerado

<https://www.base64decode.org/>

Passo a Passo

10) Testando a rota de clientes



The screenshot shows the Postman interface. A GET request is configured for the URL `localhost:3000/clientes`. The **Headers** tab is selected, showing a table with headers. The `x-access-token` header is checked and its value is `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...`. A blue arrow points from the text on the right to the value of the `x-access-token` header.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> x-access-token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...	
Key	Value	Description

The response body shows a 401 Unauthorized status with the following JSON:

```
1 {
2   "auth": false,
3   "message": "No token provided."
4 }
```

Colocar no Header
O x-access-token o
token gerado no
login