COLÉGIO COTEMIG

DESENVOLVIMENTO WEB 2 (BACKEND)

PROF. FERNANDO – OUT 2023

COLÉGIO COTEMIG

JWT

Conceitos

Conceitos

- **“JSON Web Token (JWT) é um padrão aberto (RFC 7519) que define uma maneira compacta e independente de transmitir informações com segurança entre as partes como um objeto JSON. Essas informações podem ser verificadas e confiáveis porque são assinadas digitalmente. Os JWTs podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando RSA ou ECDSA .” (JWT.IO, 2023)**

Conceitos

- **O token é dividido em três partes, sendo:**
  - **HEADER (Cabeçalho): É um JSON que especifica o tipo do token que é JWT e qual algoritmo de assinatura foi utilizado. Ele é codificado em Base64URL.**
  - **PAYLOAD (Carga Útil): É um JSON que contém as declarações públicas e outras declarações registradas que são opcionais, como, por exemplo, tempo de expiração. Ele é codificado em Base64URL.**
    *** Atenção: Não passe dados sensíveis no payload, pois, apesar de estar codificado em base64, é fácil sua decodificação e leitura.**
  - **SIGNATURE (Assinatura): Gerada pelo algoritmo de assinatura utilizando o cabeçalho mais o payload, a partir de uma chave secreta.**
  - **Esses três itens são divididos por “.” (ponto).**

Conceitos

- Ao chegar no destino, utilizando a assinatura e a chave secreta é possível verificar se o payload continua íntegro e não foi modificado. Também é possível recuperar os dados do payload.
- Para utilizar o token para autenticação, podemos passar ele através da rota protegida via o cabeçalho da requisição no seguinte formato:

```
Authorization: Bearer <token>
```

- Assim, antes de processar a rota, podemos verificar se o token é válido. Caso seja, o processamento continua normalmente e caso não, retornamos um erro para o cliente.

COLÉGIO COTEMIG

JWT

Prática

Prática

- **Em um projeto Express, instalar a biblioteca jsonwebtoken:**

```
npm install jsonwebtoken
npm install @types/jsonwebtoken -D
```

- **Importar a biblioteca jsonwebtoken no módulo da aplicação Express:**

```
import jwt from "jsonwebtoken";
```

- **Configurar no módulo da aplicação Express uma chave secreta, exemplo:**

```
const SECRET = "COTEMIG2023";
```

Prática

- **Criar uma rota de autenticação e geração do token:**

```
const users = [
      { id: 1, username: "admin", password: "admin", name: "Administrador" },
];

app.post("/login", (req: Request, res: Response) => {
      const { username, password } = req.body;
      if (typeof username !== "string" || typeof password !== "string") {
            res.status(400).json({ message: "Usuário ou Senha inválido." });
            return;
      }
      // Aqui pode ser uma consulta na tabela de usuários do banco de dados
      // Utilizei um array somente para didática
      const user = users.find(
            (value) => value.username === username && value.password === password
      );
```

Prática

- **Criar uma rota de autenticação e geração do token (continuação):**

```
        if (!user) {
                res.status(400).json({ message: "Usuário ou Senha inválido." });
                return;
        }
        const token = jwt.sign(
                { id: user.id, username: user.username, name: user.name },
                SECRET
        );
        res.status(200).json({ token });
});

/* Exemplo de resposta com um token
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pbiIsIm5hbWUiO
iJBZG1pbmlzdHJhZG9yIiwiaWF0IjoxNjk4MTU2MTM0fQ.mEeOBDb1JwBKaeey9U6eSC7EjPdL6PxvMnXltc8vJ7k"}
*/
```

Prática

- **Criar uma rota protegida:**

```
app.get("/", (req: Request, res: Response) => {
	const authorization = req.headers.authorization?.split(" ");
	// Authorization: Bearer <token>
	if (!authorization || authorization.length !== 2) {
		res.status(401).json({ message: "Não autorizado." });
		return;
	}

	let payload: jwt.JwtPayload;
	try {
		payload = jwt.verify(authorization[1], SECRET) as jwt.JwtPayload;
	} catch (error) {
		res.status(403).json({ message: "Não autorizado." });
		return;
	}
```

Prática

- **Criar uma rota protegida (continuação):**

```
    const name = payload.name;
    res.status(200).json({ message: `Meu nome é ${name}` });
});

/* Exemplo de resposta com um token válido
{"message":"Meu nome é Administrador"}
*/
```

Referências

- Introduction to JSON Web Tokens. **JWT.IO**. Disponível em: <https://jwt.io/introduction>. Acesso em: 24 de out. de 2023.

Let´s Go