FEVEREIRO, 2019

# Protegendo a Web API

Sprint 2

Escola SENAI de Informática

# INTRODUÇÃO

- Controlar os acessos;
- Somente usuários autorizados/autenticados

# JWT - JSON WEB TOKEN

## PADRÃO QUE DEFINE UMA FORMA SEGURA DE TRANSMITIR INFORMAÇÕES ENTRE DUAS PARTES

Token criptografados para permitir os acessos aos recursos de uma Web API (Bearer Authentication).

# O JWT é composto por três partes:

HEADER.PAYLOAD.SIGNATURE

O Header é um objeto JSON que define informações sobre o tipo do token (typ), nesse caso JWT, e o algorítmo de criptografia usado em sua assinatura (alg), normalmente HMAC SHA256 ou RSA.

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Header

O Payload é um objeto JSON com as Claims (informações) da entidade tratada, normalmente o usuário autenticado.

Essas claims podem ser de 3 tipos:

- Reserved claims: atributos não obrigatórios (mas recomendados) que são usados na validação do token pelos protocolos de segurança das APIs.

```
sub (subject) = Entidade à quem o token pertence, normalmente o ID
do usuário;
iss (issuer) = Emissor do token;
exp (expiration) = Timestamp de quando o token irá expirar;
iat (issued at) = Timestamp de quando o token foi criado;
aud (audience) = Destinatário do token, representa a aplicação que
irá usá-lo.
```

Geralmente os atributos mais utilizados são: **sub**, **iss** e **exp**.

- Public claims: atributos que usamos em nossas aplicações. Normalmente armazenamos as informações do usuário autenticado na aplicação.

```
name
roles
permissions
```

- Private claims: atributos definidos especialmente para compartilhar informações entre aplicações.

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

Payload

*Por segurança recomenda-se não armazenar informações confidenciais ou sensíveis no token.*

## Signature

A assinatura é a concatenação dos hashes gerados a partir do Header e Payload usando base64UrlEncode, com uma chave secreta ou certificado RSA.

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) secret base64 encoded
```

Signature

Essa assinatura é utilizada para garantir a integridade do token, no caso, se ele foi modificado e se realmente foi gerado por você.

## Resultado final

O resultado final é um token com três seções (header, payload, signature) separadas por "."—*ponto*.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJz
dWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gR
G9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab3
0RMHrHDcEfxjoYZgeFONFh7HgQ
```

Token JWT

Vantagens

Não tem sessão;

É utilizado em mobile, web, desktop;