

DESENVOLVIMENTO DE APIS

MÓDULO ESPECÍFICO

Webinar 2 – Token JWT

The SENAI logo is located in the bottom right corner of the slide. It consists of the word "SENAI" in a bold, white, sans-serif font, flanked by three horizontal lines on each side. The logo is set against a dark blue circular background that overlaps with the slide's decorative elements.

SENAI

Segurança em APIS

- a) Token JWT*
- b) Restrição de acesso*

a) Token JWT

- O que é um Token JWT?
- *O JWT é um padrão (RFC-7519) de mercado que define como transmitir e armazenar objetos JSON de forma compacta e segura entre diferentes aplicações.*
- *É um tipo de senha gerada pela API para verificar se o usuário que esta tentando realizar o acesso é válido, ela é criada a partir da criptografia dos dados que à compõe (Header, Payload e Signature).*

a) Token JWT

- *Seções do token*
- *Header*
 - *O Header é um objeto JSON que define informações sobre o tipo do token (typ), nesse caso JWT, e o algoritmo de criptografia usado em sua assinatura (alg), normalmente HMACSHA256 ou RSA.*

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

a) Token JWT

- *Seções do token*
- *Payload*
 - *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.*
 - ***Public claims:*** *atributos que usamos em nossas aplicações. Normalmente armazenamos as informações do usuário autenticado na aplicação.*
 - ***Private claims:*** *atributos definidos especialmente para compartilhar informações entre aplicações.*

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

a) Token JWT

- **Seções do token**
- **Signature**
 - Chave que garante que o token foi gerado por você.
 - A assinatura é a concatenação dos hashes gerados a partir do Header e Payload usando base64UrlEncode, com uma chave secreta ou certificado RSA.
 - Essa assinatura é utilizada para garantir a integridade do token, no caso, se ele foi modificado e se realmente foi gerado por você.
 - Isso previne ataques do tipo man-in-the-middle, onde o invasor poderia interceptar a requisição e modificar seu conteúdo, desta forma personificando o usuário com informações falsas. Caso o payload seja alterado, o hash final não será válido pois não foi assinado com sua chave secreta.

Apenas quem está de posse da chave pode criar, alterar e validar o token.

a) Token JWT

- *Funcionamento*
 - <https://jwt.io>

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEyMzQ1Njc4OTAiLCJuYW11IjoiT2RpcmxlaSI6InRpcG8iOjF9.2t26SXNhYfLoDkRxVwUZCiwIyiNwJUZI9gQYd_OGNG4
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{  
  "id": "1234567890",  
  "name": "Odirlei",  
  "tipo": 1  
}
```

VERIFY SIGNATURE

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

a) Token JWT

- Primeiro, em /api/login, o cliente envia uma requisição para gerar o token, junto com um email e senha ou usuário. O backend, então, valida essa informação e retorna com o token, se bem sucedido.*



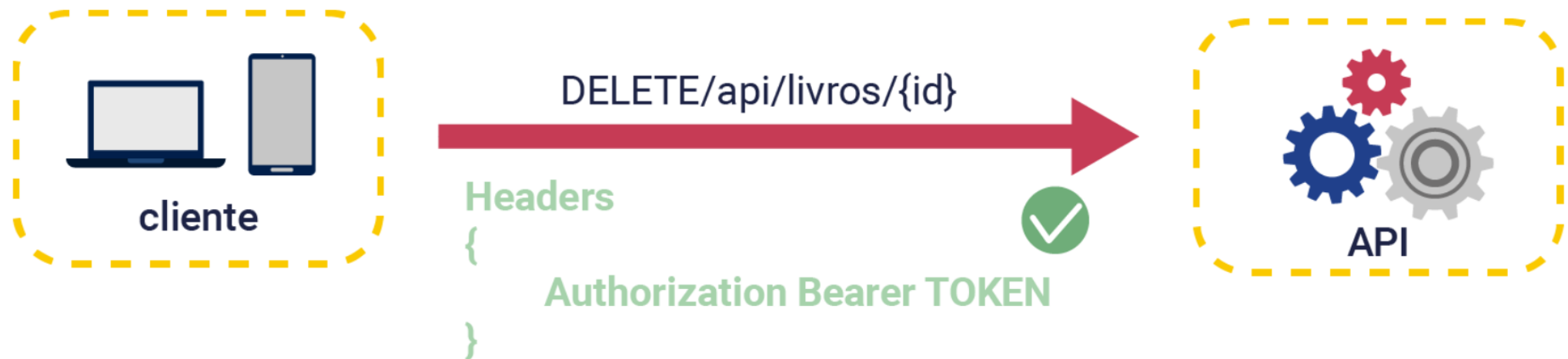
a) Token JWT

- *Após a autenticação dos dados no servidor, ele cria um token JWT com uma chave interna da API e envia token para cliente.*



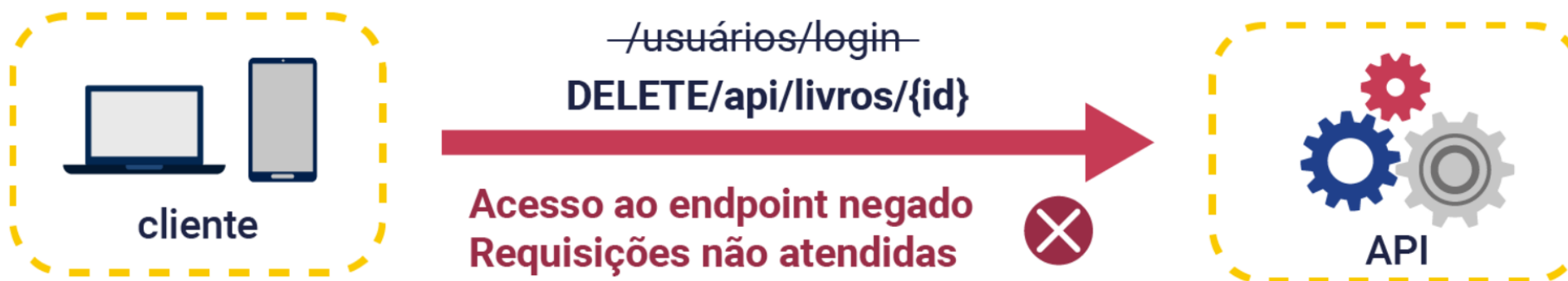
a) Token JWT

- *Com o token, a próxima requisição do usuário para DELETE /api/livros/{id} será feita com sucesso.*



a) Token JWT

- *Se o token não for o correto, por qualquer motivo, o acesso é negado e a requisição do cliente não é atendida.*



a) Token JWT

- *Funcionamento*
 - <https://token.dev>

Algorithm HS256

JWT String ⓘ Signature verification failed

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEyMzQ1Njc4OTAiLCJuYW11IjoiT2RpcmxlaSIsInRpcG8iOiJF9.2t26SXNhYfLoDkRxVwUZCiwIyiNwJUZI9gQYd_OGNG4
```

Header	Payload
<pre>{ "alg": "HS256", "typ": "JWT" }</pre>	<pre>{ "id": "1234567890", "name": "Odirlei", "tipo": 1 }</pre>

b) Restrição de acesso

- **Autenticação**

A autenticação verifica a identidade digital do usuário, ou seja, processo de verificação de uma identidade. Em termos mais simples, é quando o usuário prova de fato quem ele é.

Um exemplo bem comum de autenticação é a combinação Username e Password (Usuário e senha).

Desta forma, ao logar-se em qualquer sistema que necessite deste procedimento, o usuário está passando por um processo de autenticação. Porém, não é apenas este procedimento que autentica um usuário.

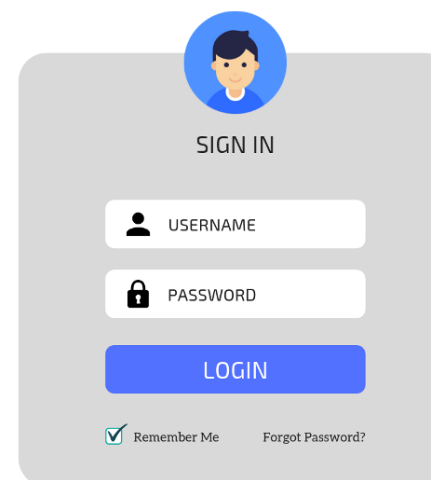
Podemos citar como exemplos:

Validações por token;

CPF e senha;

E-mail e senha;

Certificado digital, entre outros.



The illustration shows a login interface. At the top is a circular profile icon of a person. Below it is the text 'SIGN IN'. There are two input fields: the first is labeled 'USERNAME' with a person icon, and the second is labeled 'PASSWORD' with a lock icon. Below these fields is a blue button labeled 'LOGIN'. At the bottom, there is a 'Remember Me' checkbox (checked) and a 'Forgot Password?' link.

b) Restrição de acesso

- Autorização


Por sua vez, a autorização é o processo que ocorre após ser validada a autenticação. Diz respeito aos privilégios que são concedidos a determinado usuário ao utilizar uma aplicação.

Serve para verificar se determinado usuário terá a permissão para utilizar, executar recursos ou manipular determinadas ações, que é de fundamental importância dentro de uma aplicação.




Um exemplo que podemos atribuir a autorização é o uso de um ERP de uma determinada empresa. Após realizar a autenticação no sistema, o usuário do financeiro terá acesso apenas aos módulos correspondentes à realização de seu trabalho, como contas a pagar, contas a receber, etc.







b) Restrição de acesso

- Autorização



CONTROLE DE ACESSO

 123.456.789-10
 Maria Antônia Silva Santos
 Financeiro

Contas a Pagar	
Contas a Receber	
Controle de Estoque	
Cadastro de Usuários	
Cadastro de Clientes	
Vendas	
Pedidos	

```
[Authorize(Roles = "1,2")]  
[HttpGet]  
0 referências  
public IActionResult Listar()  
{
```

**Bons estudos e
até breve!**