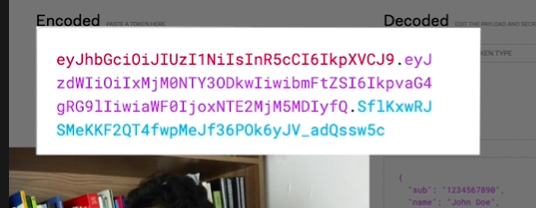


Com o JWT, a validação de identidade é baseada no corpo do JSON da requisição do JWT, não precisando mais ir até um banco de dados para poder validar a identidade de um usuário.

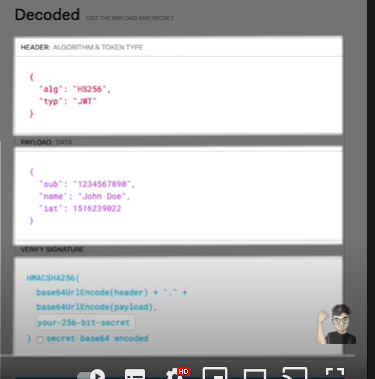
O JWT é um Token, não um Login. Ou seja, o JWT não é um login e sim um token. Um login pé uma autenticação, enquanto o token (JWT) é autorização.

A autorização utiliza o JWT para realizar a autorização da transação. Token são informações criptografadas. No JWT informações são encodadas para depois serem convertidas em JSON.

O JWT é gerado por três partes separadas por “.”:



Que correspondem, respectivamente, aos três blocos decodados abaixo:



Lembrando que o Header tem o tipo de criptografia utilizada, o Payload contém os dados, e o último bloco contém as informações de assinatura para encriptar os dados.

Esses dados do Payload podem ser chamados de Claims.

Nos claims temos o “iss” que é quem gerou a requisição.

Nos claims tem “sub” que tem o assunto da requisição.

Nos claims temos “aud” que é quem está usando a API.

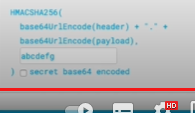
Nos claims temos o “exp” que serve para passarmos a data de expiração do JWT.

Temos o claim “ist” que é o time stamp de quando o Token foi gerado.

Esses claims acima que eu comentei são claims reservados, ou seja, que precisam ter nos nossos dados para que possamos realizar a troca de informação entre APIs usando JWT.

Dentro da JWT temos os claims privados, que são claims que somente quem está trocando as informação saberão e temos os claims públicos, que são claims públicos a qualquer um que queira visualizar.

No campo de assinatura do JWT nós encodamos com o algoritmo de codificação que escolhemos (nesse caso SHA 256) mais a chave secreta que escolhermos, conforme a imagem abaixo:



O usuário somente será autorizado caso a chave secreta que escolhermos, for a mesma.

Ou seja, quando a aplicação bater na nossa API temos que validar essa chave secreta para ver se a chave a mesma no momento da requisição.