



04. JSON Web Token

📅 Date	@31/01/2023
📁 Categoria	Java API
📖 Curso	Spring Boot 3: aplique boas práticas e proteja uma API Rest

Tópicos

- Projeto da aula anterior
- Adicionando a lib auth0 jwt
- Para saber mais: JSON Web Token
- Gerando tokens JWT
- Ajustes na geração do token
- Para saber mais: Outras informações no token
- Injeção de propriedades
- Faça como eu fiz: geração de Tokens
- O que aprendemos?

Para saber mais: JSON Web Token

JSON Web Token, ou JWT, é um padrão utilizado para a geração de *tokens*, que nada mais são do que Strings, representando, de maneira segura, informações que serão compartilhadas entre dois sistemas. Você pode conhecer melhor sobre esse padrão em seu [site oficial](#).

Aqui na Alura temos o artigo [O que é JSON Web Tokens?](#) e o Alura+ [O que é Json Web Token \(JWT\)?](#), que também explicam o funcionamento do padrão JWT.

Para saber mais: Outras informações no token

Além do Issuer, Subject e data de expiração, podemos incluir outras informações no token JWT, de acordo com as necessidades da aplicação. Por exemplo, podemos incluir o *id* do usuário no token, bastando para isso utilizar o método `withClaim`:

```
return JWT.create()
    .withIssuer("API Voll.med")
    .withSubject(usuario.getLogin())

    .withClaim("id", usuario.getId())

    .withExpiresAt(dataExpiracao())
    .sign(algoritmo);
```

O método `withClaim` recebe dois parâmetros, sendo o primeiro uma String que identifica o nome do claim (propriedade armazenada no token), e o segundo a informação que se deseja armazenar.

Injeção de propriedades

Vimos ao longo dessa aula que podemos injetar uma propriedade declarada no arquivo `application.properties` em uma classe gerenciada pelo Spring, com a utilização da anotação `@Value`. Supondo que o arquivo `application.properties` tenha a seguinte propriedade declarada:

```
app.teste=true
```

Qual das opções a seguir indica a maneira **CORRETA** de injetá-la em um atributo de uma classe gerenciada pelo Spring?

- `@Value("${app.teste}")`
 - Esse é o jeito correto de utilizar a anotação `@Value`.
- `@Value("{app.teste}")`
 - Faltou incluir o símbolo de \$.
- `@Value("app.teste")`

- A String `app.teste` deveria ser delimitada por `${}`.
-

O que aprendemos?

Nessa aula, você aprendeu como:

- Adicionar a biblioteca `Auth0 java-jwt` como dependência do projeto;
 - Utilizar essa biblioteca para realizar a geração de um *token* na API;
 - Injetar uma propriedade do arquivo `application.properties` em uma classe gerenciada pelo Spring, utilizando a anotação `@Value`;
 - Devolver um *token* gerado na API quando um usuário se autenticar nela.
-