Aula 14: Configurando a Geração de Tokens JWT

September 18, 2025

1 Introdução

Na Aula 13, implementamos o controlador `AutenticacaoController` para processar requisições de login, configuramos o `AuthenticationManager` e o algoritmo BCrypt, e ajustamos a entidade `Usuario` para implementar `UserDetails`. Nesta aula, começaremos a implementar a geração de tokens JWT (JSON Web Token) para retornar no endpoint de login, utilizando a biblioteca Auth0 (`java-jwt`). Adicionaremos a dependência no `pom.xml` e prepararemos o `AutenticacaoController` para retornar o token. As alterações serão feitas no `Visual Studio Code` ou sua IDE preferida.

2 Revisão do Processo de Autenticação

O processo de autenticação envolve:

- O cliente envia uma requisição `POST /login` com um JSON contendo `login` e `senha`.
- O `AutenticacaoController` usa o `AuthenticationManager` para validar as credenciais via `AutenticacaoService`.
- Após validação, a API deve gerar e retornar um token JWT.

3 Adicionando a Biblioteca Auth0

Para gerar tokens JWT, usaremos a biblioteca `java-jwt` da Auth0. No site `https://jwt.io/`, selecionamos a seção "Libraries", filtramos por Java, e escolhemos o projeto `java-jwt` (versão 4.2.1, recomendada para consistência com esta aula). Copiamos a dependência Maven do repositório GitHub (`https://github.com/auth0/java-jwt`).

No `Visual Studio Code` ou sua IDE preferida, paramos o servidor (atalho: `Ctrl + F12`) e abrimos o arquivo `pom.xml`. Adicionamos a dependência dentro da tag `<dependencies>`:

```
<dependency>
    <groupId>com.auth0</groupId>
    <artifactId>java-jwt</artifactId>
    <version>4.2.1</version>
</dependency>
```

Após salvar, recarregamos o Maven clicando no ícone de reload no painel do Maven para baixar a biblioteca.

4 Preparando o `AutenticacaoController` para Retornar o Token

No `AutenticacaoController` (pacote `med.voll.api.controller`), o método `efetuarLogin` atualmente retorna `ResponseEntity.ok().build()`. Na próxima etapa, modificaremos este método para retornar um token JWT. O código atual é:

```java
package med.voll.api.controller;

import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.security.authentication.AuthenticationManager;
import
    org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import med.voll.api.domain.usuario.DadosAutenticacao;

@RestController
@RequestMapping("/login")
public class AutenticacaoController {

    @Autowired
    private AuthenticationManager manager;

    @PostMapping
    public ResponseEntity efetuarLogin(@RequestBody @Valid DadosAutenticacao
        dados) {
        var token = new UsernamePasswordAuthenticationToken(dados.login(),
            dados.senha());
        var authentication = manager.authenticate(token);
        return ResponseEntity.ok().build();
    }
}
```

Na próxima aula, criaremos uma classe para gerar o token JWT e atualizaremos o método `efetuarLogin` para retornar o token no corpo da resposta.

5 Próximos Passos

Com a biblioteca `java-jwt` adicionada, estamos prontos para implementar a lógica de geração de tokens JWT e retornar o token no endpoint `/login`. Na próxima aula, criaremos uma classe de serviço para gerar tokens e configuraremos o `AutenticacaoController` para incluí-los na resposta. Continue praticando no `Visual Studio Code` ou sua IDE preferida!

6 Dica do Professor

- **Aprofunde-se em JWT**: Consulte a documentação da biblioteca Auth0 (`https://github.com/auth0/java-jwt`) e o site `https://jwt.io/introduction` para entender o padrão JWT.
- **Comunidade no X**: Siga perfis como @JWTSecurity e @Auth0Dev no X para dicas sobre tokens JWT e segurança em APIs.
- **Pratique**: Verifique no `pom.xml` se a dependência `java-jwt` foi baixada corretamente e explore o repositório da Auth0 para exemplos de uso da biblioteca.