

Aula 19: Recuperando o Token JWT no Filtro de Segurança

September 18, 2025

1 Introdução

Na Aula 18, criamos o filtro `SecurityFilter` para interceptar requisições antes de chegarem aos controllers. Nesta aula, implementaremos a lógica para recuperar o token JWT do cabeçalho `Authorization` da requisição, removendo o prefixo “Bearer”. Testaremos a recuperação do token no Insomnia, verificando o resultado no console da IDE. As alterações serão feitas no Visual Studio Code ou sua IDE preferida.

2 Configurando o Token no Insomnia

No Insomnia, configuramos a requisição `GET http://localhost:8080/medicos` para incluir o token JWT:

- Na aba “Auth”, selecionamos a opção “Bearer”.
- Marcamos a checkbox “Enabled”.
- No campo “Token”, colamos o token JWT gerado anteriormente (ex.: obtido via `POST /login`).

Disparamos a requisição, mas inicialmente nada acontece, pois o filtro ainda não processa o token.

3 Implementando a Recuperação do Token

No `SecurityFilter` (pacote `med.voll.api.infra.security`), adicionamos a lógica para recuperar o token JWT do cabeçalho `Authorization`:

```
1 package med.voll.api.infra.security;
2
3 import jakarta.servlet.FilterChain;
4 import jakarta.servlet.ServletException;
5 import jakarta.servlet.http.HttpServletRequest;
6 import jakarta.servlet.http.HttpServletResponse;
7 import org.springframework.stereotype.Component;
8 import org.springframework.web.filter.OncePerRequestFilter;
9 import java.io.IOException;
10
11 @Component
12 public class SecurityFilter extends OncePerRequestFilter {
13
14     @Override
15     protected void doFilterInternal(HttpServletRequest request,
16                                     HttpServletResponse response, FilterChain filterChain) throws
17                                     ServletException, IOException {
18         var tokenJWT = recuperarToken(request);
19         System.out.println(tokenJWT);
20     }
21 }
```

```

18     filterChain.doFilter(request, response);
19 }
20
21 private String recuperarToken(HttpServletRequest request) {
22     var authorizationHeader = request.getHeader("Authorization");
23     if (authorizationHeader == null) {
24         throw new RuntimeException("Token JWT não enviado no cabeçalho
25         Authorization!");
26     }
27     return authorizationHeader.replace("Bearer ", "");
28 }

```

3.1 Explicação do Código

- `recuperarToken(HttpServletRequest request)`: Método que extrai o token do cabeçalho `Authorization`.
- `request.getHeader("Authorization")`: Obtém o valor do cabeçalho `Authorization` (ex.: `Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...`).
- `if (authorizationHeader == null)`: Lança uma exceção se o cabeçalho estiver ausente.
- `replace("Bearer ", "")`: Remove o prefixo "Bearer " para obter apenas o token JWT.
- `System.out.println(tokenJWT)`: Log temporário para verificar o token recuperado.
- `filterChain.doFilter(request, response)`: Permite a continuação do fluxo da requisição.

4 Testando a Recuperação do Token

No Insomnia, disparamos novamente a requisição GET `http://localhost:8080/medicos` com o token no cabeçalho `Authorization`. No console da IDE, verificamos que o token JWT é impresso sem o prefixo "Bearer", confirmando que a recuperação foi bem-sucedida. A requisição retorna código 200 (OK) com o JSON esperado:

```

1 [
2   {
3     "id": 1,
4     "nome": "Nome do Médico",
5     "email": "medico1@voll.med",
6     "crm": "123456",
7     "especialidade": "ORTOPEDIA"
8   },
9   ...
10 ]

```

5 Próximos Passos

Implementamos a recuperação do token JWT no `SecurityFilter`. Na próxima aula, adicionaremos a lógica de validação do token usando a biblioteca `java-jwt`, verificando sua integridade e autenticidade com a chave secreta. Continue praticando no Visual Studio Code ou sua IDE preferida!

6 Dica do Professor

- **Aprofunde-se em Filtros:** Consulte a documentação do Spring Security sobre manipulação de cabeçalhos (<https://docs.spring.io/spring-security/reference/servlet/architecture.html#servlet-filters>) para entender como processar requisições.
- **Comunidade no X:** Siga perfis como @SpringSecExpert e @TokenAuth no X para dicas sobre validação de tokens e filtros.
- **Pratique:** No Insomnia, envie a requisição GET /medicos sem o cabeçalho Authorization e verifique se a exceção RuntimeException é lançada no console da IDE.