

CETEJ35 - Java Web - JAVA_XXX (2024_01)

[Meus cursos](#) / [CETEJ35 - Web \(2024_01\)](#) / [Semana 09: 28/10 a 03/11](#) / [Segurança](#)

Segurança

✓ **Feito:** Ver

✓ **Feito:** Gastar pelo menos 20 minutos na atividade

A fazer: Passar pela atividade até o fim

Fecha: segunda-feira, 2 dez. 2024, 00:00

A maioria das aplicações Web têm algum tipo de segurança. Segurança é um termo amplo que abrange vários aspectos, como conexão segura, cifragem de dados entre outros. Nesta seção, nós vamos focar em dois dos mecanismos mais comuns de segurança: autenticação e autorização.

LOGOUT E TELA DE LOGIN

No momento, o aplicativo permite o login, mas não permite o logout. Se você quiser encerrar uma sessão, precisa limpar o cache do navegador. Para permitir que um usuário faça logout, vamos criar uma barra de navegação superior e um botão de logout.

Para isso, abra a página `crud.ftl` e insira a barra exatamente antes do início do *container* definido pelo Bootstrap. Veja o código a seguir.



```
13 <body>
14     <nav class="navbar navbar-expand-sm bg-dark">
15         <ul class="navbar-nav ml-auto">
16             <li class="nav-item">
17                 <a
18                     href="/logout"
19                     class="nav-link btn btn-secondary"
20                 >Sair da aplicação</a>
21             </li>
22         </ul>
23     </nav>
24
25     <div class="container-fluid">
26         <div class="jumbotron mt-5">
27             <h1>GERENCIAMENTO DE CIDADES</h1>
28             <p>UM CRUD PARA CRIAR, ALTERAR, EXCLUIR E LISTAR CIDADES</p>
29         </div>
```

As linhas 14 a 23 definem a barra de navegação, enquanto as linhas 17 a 20 definem o botão de logout. Observem que tudo que precisa ser feito para que o logout funcione, é criar um link para a URL `/logout` (linha 18). Quando o Spring Boot recebe a solicitação dessa URL, ele encerra a sessão do usuário atual e redireciona para a tela de login.

[Sair da aplicação](#)

GERENCIAMENTO DE CIDADES

UM CRUD PARA CRIAR, ALTERAR, EXCLUIR E LISTAR CIDADES

Cidade:

Estado:

CRIAR

Nome	Estado	Ações
------	--------	-------



Outra coisa que provavelmente você vai querer fazer na sua aplicação Web é criar uma tela de login personalizada. Fazer isso é simples. Vamos começar criando uma nova página Web. A tela de login é uma página Web simples, por isso, vamos criar o arquivo `login.html` em `/resources/static/`.

Vamos usar o HTML e a formatação do Bootstrap para criar um componente central com um formulário. No formulário, vamos colocar uma entrada de texto e uma entrada do tipo senha. Também precisamos de um botão para submeter o formulário. Veja o código na Figura abaixo.

Se não quiser perder tempo digitando toda essa formatação, você pode consultar o código direto no repositório no Github. A branch usada está descrita no final desta seção..

```

4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>CRUD Cidades - Login</title>
9
10     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
11 </head>
12
13 <body class="bg-light">
14     <main class="container-fluid">
15         <section class="col-md-12 d-flex justify-content-center mt-5">
16             <div class="card">
17                 <div class="card-header bg-primary text-white">Dados para Login</div>
18                 <div class="card-body">
19                     <form method="POST">
20                         <div class="form-group">
21                             <label for="usuario">Usuário:</label>
22                             <input name="username" type="text" class="form-control" placeholder="Informe usuário"
23                                 id="usuario">
24                         </div>
25                         <div class="form-group">
26                             <label for="senha">Senha:</label>
27                             <input name="password" type="password" class="form-control" placeholder="Informe senha"
28                                 id="senha">
29                         </div>
30                         <button type="submit" class="btn btn-primary">Login</button>
31                     </form>
32                 </div>
33             </div>
34         </section>
35     </main>
36 </body>

```

Não vamos entrar em detalhes da formatação usada pelo Bootstrap, já que esse não é nosso foco. O que é importante notar aqui está nas linhas 19, 22 e 27. Na linha 19, definimos o método de envio de dados do formulário como **POST**. Isso é fundamental, já que o Spring Security espera que os dados de login sejam sempre enviados por **POST**.



Na linha 22 e 27 definimos os componentes de entrada do nome do usuário (**input type="text"**) e senha (**input type="password"**), respectivamente. Para que o Spring Security entenda que estamos enviando o nome e a senha do usuário, o componente do nome do usuário deve ter o atributo **name** definido como **username**. Da mesma forma, o componente de senha deve ter o atributo **name** definido como **password**.

Agora, precisamos voltar para a classe **crudcidades.SecurityConfig** para configurar nossa nova página como página de login. Logo após o método **formLogin()**, vamos adicionar o método **loginPage(String)**. Como parâmetro do método, vamos definir a página de login **login.html**. O método **permitAll()** deve então ser anexado ao método **loginPage(String)**, significando que qualquer um pode acessar a página de login.

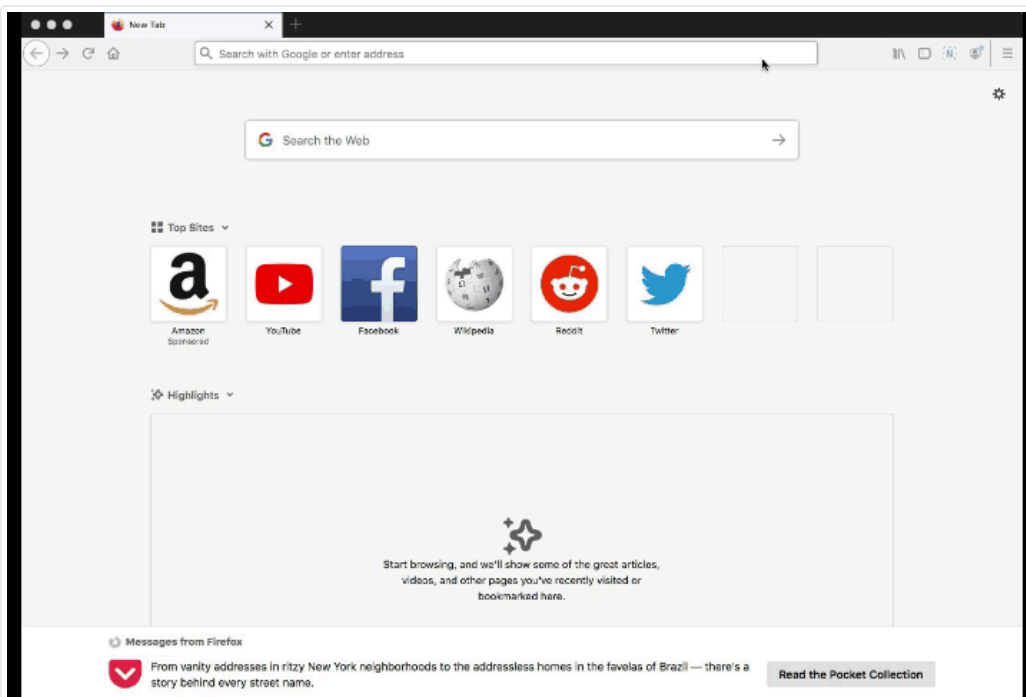
Também vamos adicionar (**and()**) uma permissão para acesso à página de logout (**logout().permitAll()**). Veja como ficou o código com as mudanças.

```
SecurityConfig.java

33 @Bean
34 protected SecurityFilterChain filter(HttpSecurity http) throws Exception {
35     return http
36         .csrf().disable()
37         .authorizeHttpRequests()
38         .requestMatchers("/").hasAnyRole("listar", "admin")
39         .requestMatchers("/criar", "/excluir", "/preparaAlterar", "/alterar").hasRole("admin")
40         .anyRequest().denyAll()
41         .and()
42         .formLogin()
43         .loginPage("/login.html").permitAll()
44         .and()
45         .logout().permitAll()
46         .and()
47         .build();
48 }
49
```

Snipped

Ao executar a aplicação, você verá a nova página de login.



O código desenvolvido nesta Seção está disponível no [Github](#) na branch `semana07-30-logout-login-form`.

[Retroceder](#)[Avançar](#)[← Atividade II WebConf](#)[Listeners & Cookies ►](#)[✉ Contate o suporte do site](#)

Você acessou como RAFAEL ROCHA DA SILVA PROENCA (Sair)

CETEJ35 - Web (2024_01)

Tema

Adaptable

Boost

Clássico

Campus

Apucarana

Campo Mourão

Cornélio Procópio

Curitiba

Dois Vizinhos

Francisco Beltrão

Guarapuava

Londrina

Medianeira

Pato Branco

Ponta Grossa

Reitoria

Santa Helena

Toledo

UTFPR

Ajuda

Chat UTFPR

Calendário Acadêmico

Biblioteca

e-Mail

Nuvem (OwnCloud)

Produção Acadêmica

Secretaria Acadêmica

Sistemas Corporativos

Sistema Eletrônico de Informação - SEI

Suporte ao usuário

Criação de curso

Comunidade

Português - Brasil (pt_br)

Deutsch (de)

English (en)

Português - Brasil (pt_br)

Resumo de retenção de dados

Baixar o aplicativo móvel.



Dê um feedback sobre este software



Universidade Tecnológica Federal do Paraná - UTFPR
Suporte ao usuário