



Testando o login

Transcrição

Bem vindo de volta! Agora nosso objetivo é verificar se o usuário está logado ou não, pois até o momento é possível acessar nossa aplicação mesmo sem fazer o login, o que queremos evitar.

É muito fácil saber se o usuário já fez login, basta verificarmos o `HttpSession` pois lá dentro existe um atributo chamado `usuarioLogado`. Se esse atributo existir, o usuário poderá continuar o acesso.

Vamos começar a implementar isso, de maneira inocente, na classe `ListaEmpresas`. Antes de tudo, precisamos do Servlet `HttpSession`:

```
public class ListaEmpresas implements Acao {  
  
    public String executa(HttpServletRequest request, HttpServ.  
  
        HttpSession sessao = request.getSession();
```

[COPIAR CÓDIGO](#)

Repare que esse objeto `request`, além de nos dar dados sobre a requisição, também é uma fachada que contém vários métodos que nos ajudam a acessar outros recursos do mundo Servlet, como o próprio `HttpSession`.

Com essa expressão, se não existe uma `sessao` no acesso do usuário, o Tomcat vai criar um novo objeto `HttpSession` e uma ID associada na resposta. Agora podemos fazer a verificação por meio do `if()` :

```
HttpSession sessao = request.getSession();  
if(sessao.getAttribute("usuarioLogado") == null) {  
  
}
```

[COPIAR CÓDIGO](#)

Dessa forma, temos uma condição para o usuário que não estiver logado. Nesses casos, devemos mandá-lo de volta ao formulário de login. Vamos escrever isso:

```
HttpSession sessao = request.getSession();  
if(sessao.getAttribute("usuarioLogado") == null) {  
    return "redirect:entrada?acao=LoginForm";  
}
```

[COPIAR CÓDIGO](#)

Vamos testar? Se tentarmos acessar a URL

<http://localhost:8080/gerenciador/entrada?acao=ListaEmpresas>
(<http://localhost:8080/gerenciador/entrada?acao=ListaEmpresas>) sem fazer o login, o Tomcat percebe que não temos o atributo `usuarioLogado` e nos redireciona para <http://localhost:8080/gerenciador/entrada?acao=LoginForm> (<http://localhost:8080/gerenciador/entrada?acao=LoginForm>).

Agora precisamos fazer isso para cada uma de nossas ações. Poderíamos fazer isso simplesmente copiando e colando esse código em cada uma das classes, mas isso não parece uma boa prática, certo? Seria melhor se nós pudséssemos implementar essa verificação em um único local, mas onde?

Ainda não sabe a resposta? Não se preocupe, pois no próximo vídeo iremos implementar essa verificação. Até lá!