

Lab 5 - Formulários e Cookies

Neste laboratório iremos aprender a armazenar informações de formulários com um servlet e trabalharemos com manipulação de sessão e cookies.

Exercícios

Exercício 1: Criando o **Servlet e o Formulário** que recebe somente dados inseridos.

Exercício 2: Manipulando **Cookies** em um Servlet.

Exercício 1 - Criando Servlet e o formulário que recebe os dados

1. Crie uma nova Classe java com o nome **ManterCliente** dentro do pacote **threeway.projeto.servlet**, realize as alterações para que fique conforme a especificação abaixo.

```
public class ManterCliente extends HttpServlet{

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    }

}
```

2. Abra o arquivo **web.xml** e adicione as configurações do servlet.

```
<servlet>
  <servlet-name>ManterCliente</servlet-name>
  <servlet-class>threeway.projeto.servlets.ManterCliente</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ManterCliente</servlet-name>
  <url-pattern>/ManterCliente</url-pattern>
</servlet-mapping>
```

3. Altere o método **doGet()** para que fique igual a especificação abaixo.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saida de dados recebidos do Formulario -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);
    response.sendRedirect("manterCliente.jsp");
}
```

getParameter(): recupera os inputs mapeados no formulário através do atributo *name*, método do objeto *request*.

sendRedirect(): redireciona a requisição para uma página específica, método presente no objeto *response*.

4. Altere o método doPost() para que fique conforme abaixo.

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}
```

Por que o método doPost() chama o doGet() ?

Como vamos enviar o formulário por post, o método chamado será o doPost(), assim os dados serão mandados no corpo da página e não por URL. Porém, se for digitado na URL o nome e o valor, terá o mesmo efeito pois o doGet() foi implementado. Se não houver necessidade de diferenciar o método de submit do http você pode sobrepor o método **service()**.

5. Crie o arquivo **manterCliente.jsp** e acrescente o código <HTML> seguindo as marcações em amarelo para que fique igual ao formato abaixo.

```
<head>
<link href="css/bootstrap.css" rel="stylesheet">
<style type="text/css">
form {
    margin-top: 50px;
}
</style>
<title>Login</title>
</head>
<body>
    <div class="container">
        <form class="form-horizontal" action="ManterCliente" method="post" role="form">
            <div class="form-group">
                <label class="col-sm-2 control-label">Nome</label>
                <div class="col-sm-4">
                    <input type="text" name="nome" class="form-control"
                        placeholder="Nome" autofocus>
                </div>
                <label class="col-sm-2 control-label">Telefone</label>
                <div class="col-sm-4">
                    <input type="text" name="telefone" class="form-control"
                        placeholder="(xx)0000-0000">
                </div>
            </div>
            <div class="form-group">
                <label class="col-sm-2 control-label">Endereço</label>
                <div class="col-sm-10">
                    <input type="text" name="endereco" class="form-control"
                        placeholder="Rua, Bairro, Cidade">
                </div>
            </div>
            <div class="form-group">
                <label class="col-sm-2 control-label">RG</label>
                <div class="col-sm-10">
                    <input type="text" name="rg" class="form-control"
                        placeholder="0000000">
                </div>
            </div>
            <div class="row">
                <p class="text-right" style="padding-right: 5%;">
                    <button type="reset" class="btn btn-primary">Limpar</button>
                    <button type="submit" class="btn btn-primary">Salvar</button>
                </p>
            </div>
        </form>
    </div>
</body>
```

action: atributo do componente <form>, deve ser mapeado o nome do servlet que tratará as requisições do formulário.

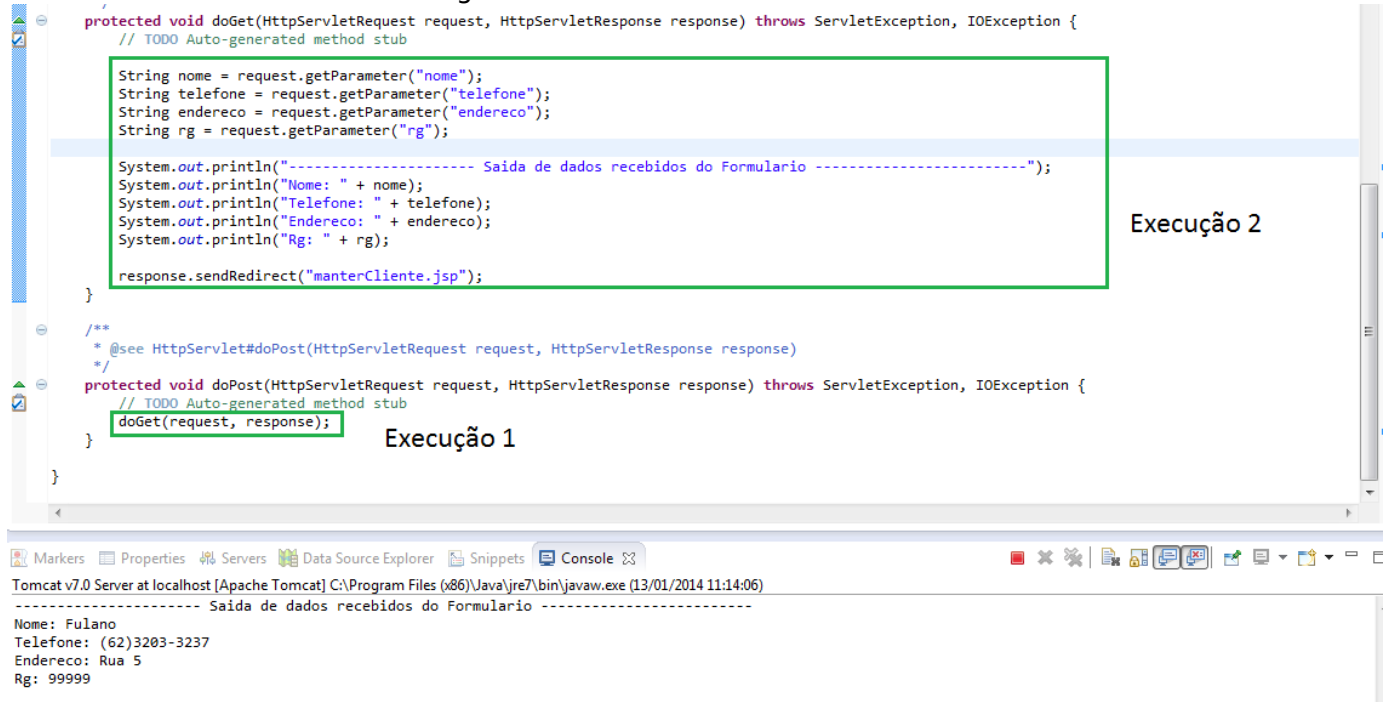
method: atributo do componente <form>, pode ser mapeado como get/post. Representa como os dados serão enviados para o servidor.

name: atributo do componente <input>, possibilita a recuperação do valor do componente pelo servlet.

type: Reset (limpa todos os dados do formulário) ou Submit (submete-dados do formulário).

6. Agora envie os dados pressionando o botão salvar.

Obs: a saída será conforme a imagem abaixo.



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saida de dados recebidos do Formulario -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);

    response.sendRedirect("manterCliente.jsp");
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}

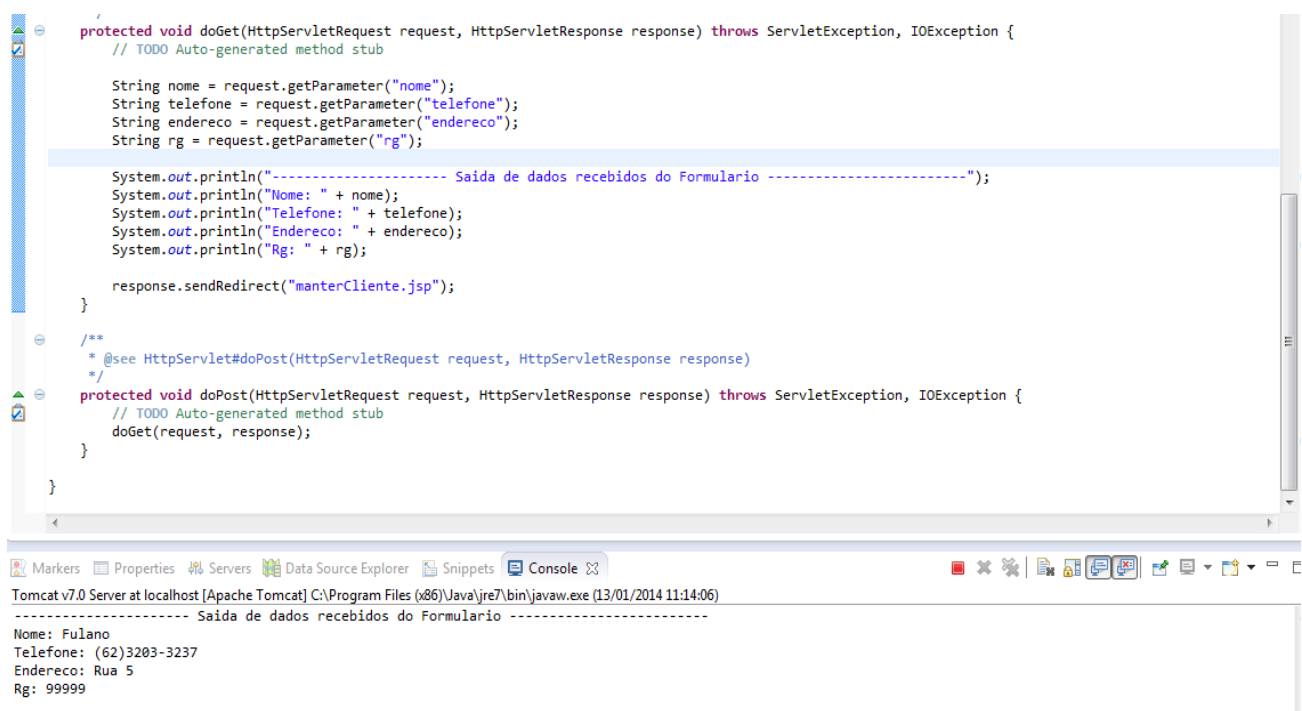
----- Saida de dados recebidos do Formulario -----
Nome: Fulano
Telefone: (62)3203-3237
Endereco: Rua 5
Rg: 99999
```

Execução 1

Execução 2

7. Exemplo de envio pela URL, nesse caso o método chamado será apenas o **doGet()**, copie o código selecionado abaixo e cole na URL do navegador, a saída deve ficar igual a imagem abaixo.

- [http://localhost:8080/Livraria-web/ManterCliente?nome=Fulano&telefone=\(62\)3203-3237&endereco=Rua+5&rg=99999](http://localhost:8080/Livraria-web/ManterCliente?nome=Fulano&telefone=(62)3203-3237&endereco=Rua+5&rg=99999)



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    String nome = request.getParameter("nome");
    String telefone = request.getParameter("telefone");
    String endereco = request.getParameter("endereco");
    String rg = request.getParameter("rg");

    System.out.println("----- Saida de dados recebidos do Formulario -----");
    System.out.println("Nome: " + nome);
    System.out.println("Telefone: " + telefone);
    System.out.println("Endereco: " + endereco);
    System.out.println("Rg: " + rg);

    response.sendRedirect("manterCliente.jsp");
}

/**
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
 */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);
}

----- Saida de dados recebidos do Formulario -----
Nome: Fulano
Telefone: (62)3203-3237
Endereco: Rua 5
Rg: 99999
```

Exercício 2 - Manipulando Cookies em um Servlet

1. Crie um novo **Servlet** com o nome **Login** conforme o passo um do Exercício um.
2. Realize o mapeamento no **web.xml** conforme foi feito no passo dois do Exercício um.
3. Realize o Override dos métodos doGet() e doPost(), trazendo os para a classe Login.
4. Altere o método doGet() de doPost() para que fique conforme a especificação abaixo.

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {

    final String login = request.getParameter("login");
    final String senha = request.getParameter("senha");

    final String cookieName = "usuario";
    final String cookieValue = login;

    //criando o objeto cookie
    Cookie cookieUser = new Cookie(cookieName, cookieValue);
    //vamos criar um cookie da senha apenas como exemplo para termos mais de um cookie
    Cookie cookiePsw = new Cookie("senha", senha);

    //duração de um dia
    cookieUser.setMaxAge(60*60*24);
    cookiePsw.setMaxAge(60*60*24);

    //adicionando o cookie ao navegador
    response.addCookie(cookieUser);
    response.addCookie(cookiePsw);

    //Esta funcionalidade será ensinada nas próximas aulas
    response.sendRedirect("login.jsp");

}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    doGet(request, response);
}
```

No construtor da classe Cookie é especificado o nome e valor do cookie que será criado no navegador, por enquanto não esta sendo validado login e senha apenas recuperando os dados do formulário de login e criando o cookie. Fique atento aos comentários no código.

5. No diretório WebContent, crie o arquivo **login.jsp** com o código abaixo.

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<jsp:useBean id="mensagem" scope="session" class="java.lang.String"/>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<link href="css/bootstrap.css" rel="stylesheet" type="text/css">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
</head>
<body>
    <div class="container">

        <div>
            <div class="col-md-4 col-md-offset-4" style="margin-top: 20%;">
                <div class="panel panel-default">
```

```

<div class="panel-heading">
  <span class="glyphicon glyphicon-lock"></span> Login</div>
<div class="panel-body">
  <form class="form-horizontal" method="post" action="Login" role="form">
    <div class="form-group">
      <label for="inputEmail3" class="col-sm-3 control-label">
        Login:</label>
      <div class="col-sm-9">
        <input type="text" class="form-control" name="login"
          placeholder="login" required="required" autofocus="autofocus">
      </div>
    </div>
    <div class="form-group">
      <label for="inputPassword3" class="col-sm-3 control-label">
        Password:</label>
      <div class="col-sm-9">
        <input type="password" class="form-control" name="senha"
          placeholder="senha" required>
      </div>
    </div>
    <div class="form-group last">
      <div class="col-sm-offset-3 col-sm-9">
        <button type="submit" class="btn btn-success btn-sm">
          Sign in</button>
        <button type="reset" class="btn btn-default btn-sm">
          Reset</button>
      </div>
    </div>
  </form>
</div>
<div class="panel-footer" style="text-align:center;font-size: 12px;">
  Livraria 3way - Fone:(99) 999-9999
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

6. Abra o arquivo web.xml e altere a **tag** `<welcome-file-list>` deixando conforme abaixo.

```

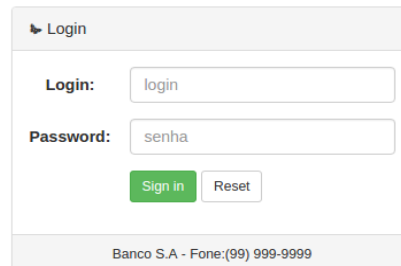
<display-name>Livraria-web</display-name>
<welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
</welcome-file-list>

```

7. Agora adicione o projeto ao container apache tomcat e inicie o serviço (google chrome ou firefox).

8. Usando o navegador acesse o projeto:

- <http://localhost:8080/Livraria-web/>
- Realize o login, por enquanto pode digitar qualquer coisa, sugestão (**login:**'admin', **senha:**'admin').
- Pressione a tecla F12, e selecione a aba console da "ferramenta para desenvolvedor" conforme a imagem abaixo, digite **document.cookie** e pressione enter.

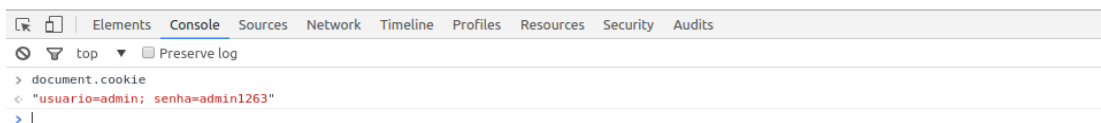


Login

Login:

Password:

Banco S.A - Fone:(99) 999-9999



```
> document.cookie
< "usuario=admin; senha=admin1263"
```

- Note que além do cookie criado marcado em verde, conseguimos recuperar outros cookies de outras aplicações, dessa forma aplicações maliciosas podem utilizar esses dados em cookies para outros fins, não é recomendado desenvolvermos aplicações que guardam informações ou dependem muito de cookies.

Lembrete: Atributos de um **Cookie**.

- **getDomain/setDomain** - Permite você especificar o **domínio** o qual o cookie se aplica. O **host** deve ser parte do domínio especificado.
- **getMaxAge/setMaxAge** - Retorna/atribui o **tempo de expiração** do cookie (em segundos). Se você omitir esta atribuição, o cookie se aplica apenas para a sessão atual do browser.
- **getName** - Retorna o nome do cookie. Não existe método setName; fornecemos o nome para o construtor. Para um array de cookies, utilizamos **getName** para encontrar o cookie de interesse.
- **getPath/setPath** - Retorna/atribui o caminho no qual o cookie se aplica. Se não for especificado, o cookie se aplica a URLs que estão dentro ou abaixo do diretório contendo a página atual.
- **getSecure/setSecure** - Retorna/atribui o **flag** indica se o cookie deve ser aplicado apenas a **conexões SSL** ou para todas conexões.
- **getValue/setValue** - Retorna/atribui o valor associado com o cookie. Para novos cookies, fornecemos o valor para o construtor, não através de **setValue**. Para uma array de cookies de entrada, utilizamos **getName** para encontrar o cookie de interesse, então chamamos **getValue** no resultado. Se atribuirmos um valor ao cookie de entrada, ainda necessitamos enviá-lo de volta com **response.addCookie**.