FTCE

Faculdade de Tecnologia e Ciências Exatas

USJT - 2017 - Práticas de Programação

Professores Teoria: Bonato, Hamilton
Laboratório: Calvetti, Hamilton, Keity e Rodrigo

Aula: 5
Assunto: JSP - Introdução

Conceitos básicos

Primeira coisa. Um JSP, no final, vira um Servlet. O JSP é apenas uma maneira para se escrever código Java dentro de um HTML, e não HTML dentro do código Java.
Existem 4 tipos diferentes de expressões um um JSP:

- Scriptlet <% %>
- Diretiva <%@ %>
- Expressão <%= %>
- Declarações <%! %>

Em um Scriptlet você escreve código Java, as diretivas você usa para fazer os imports. As expressões são parecidas com os scriptlets, mas não precisam do out.println(), e as declarações servem para criar métodos e variáveis de instância. Veja os exemplos abaixo:

Scriptlet e Diretiva

```
<%@page import="utilidades.*" %>
<html>
<body>
O contador de páginas é:
<% out.println("Contador.getContagem()); %>
</body>
</html>
```

Expressão e Diretiva

```
<%@page import="utilidades.*" %>
<html>
<body>
O contador de páginas é:
<%= Contador.getContagem() %>
</body>
</html>
```

Note que não se usa ponto e vírgula em expressões. Você também não pode usar uma classe feita por você em um JSP se ela não estiver um uma package e esta package não houver sido informada ao JSP por meio de uma diretiva import.

Declaração

```
<html>
<body>
<%! int dobraContador() {
        contador = contador*2;
        return contador;
    } %>
<% int contador=1; %>
O contador de páginas é:
<%= dobraContador() %>
</body>
</html>
```

O método acima define o método *int dobraContador()* e a variável de instância *int Contador*.
Quando o JSP é transformado em servlet, as expressões e os scriptlets são colocados dentro de um método do servlet gerado. Já as declarações são colocadas fora deste método principal, pois são outros métodos. E as diretivas, é claro, são colocadas fora da classe, local onde os imports são colocados.

Os JSP tem alguns objetos implícitos, isto é, objetos que não é você que instancia mas que estão lá para você usar. São eles:

API	**Objeto Implícito**
JspWriter	out
HttpServletRequest	request
HttpServletResponse	response
HttpSession	session
ServletContext	application
ServletConfig	config
JspException	exception
PageContext	pageContext
Object	page

Exemplo:

O código abaixo mostra um CRUD de clientes, baseado na tabela de cliente com a chave primária id auto incremental. Basicamente o código que vimos usando como exemplo até agora.
O formulário fornecido na semana passada foi alterado para conter todas as ações do CRUD e o servlet também for alterado neste sentido. Além disso, agora o servlet não mais gera o código HTML e dá a resposta HTTP, mas despacha esta ação para um JSP.

Exemplo: formulário HTML

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Cadastro de Clientes</title>
</head>
<body>
<h3>Cadastro de Clientes</h3>
<form action="ManterCliente.do" method="get">

nome: <input type="text" name="nome"><br>
fone: <input type="text" name="fone"><br>
e-mail: <input type="text" name="email"><br>
<input type="submit">
</form>
</body>
</html>
```

Exemplo: Servlet

```
package controller;

import java.io.IOException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import model.Cliente;
import service.ClienteService;

@WebServlet("/ManterCliente.do")
public class ManterClienteController extends HttpServlet {
    private static final long serialVersionUID = 1L;

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

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

```
        String pNome = request.getParameter("nome");
        String pFone = request.getParameter("fone");
        String pEmail = request.getParameter("email");

        //instanciar o javabean
        Cliente cliente = new Cliente();
        cliente.setNome(pNome);
        cliente.setFone(pFone);
        cliente.setEmail(pEmail);

        //instanciar o service
        ClienteService cs = new ClienteService();
        cs.criar(cliente);
        cliente = cs.carregar(cliente.getId());

        //enviar para o jsp
        request.setAttribute("cliente", cliente);

        RequestDispatcher view =
        request.getRequestDispatcher("Cliente.jsp");
        view.forward(request, response);
    }
}
```

Exemplo: JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@page import="model.Cliente" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Cliente</title>
</head>
<body>
       <%Cliente cliente = (Cliente)request.getAttribute("cliente"); %>
       Id: <%=cliente.getId() %><br>
       Nome: <%=cliente.getNome() %><br>
       Fone: <%=cliente.getFone() %><br>
       E-mail: <%=cliente.getEmail() %><br>
</body>
</html>
```

Exercícios

1. Revisão da Teoria

a. Importe o projeto do cadastro de clientes criado na semana passada.

b. Substitua o servlet e o index.html pelo novo código fornecido.

c. Copie o arquivo Cliente.jsp para o WebContents do seu projeto.

d. Faça o deployment no Tomcat e dê restart.

e. Abra a aplicação no Chrome ou no Firefox.

f. Exporte o código e salve no seu pendrive.

2. Exercício para Nota (correção em aula)

a. Crie um arquivo JSP. Clique com o botão direito em WebContents, New > JSP File. Chame de Pais.jsp, next, escolha New JSP File (html), finish.

b. Altere o servlet para fazer dispatch do JSP. Não se esqueça de passar o javabean via request.

c. Clique sobre o Tomcat em Servers, dê um restart.

d. Volte para o index.html e cadastre outro país.

e. Exporte o código e salve no seu pendrive.

Bibliografia:

DEITEL, Harvey M.; DEITEL, Paul J; FURMANKIEWICZ, Edson. **Java: como programar.** 6. ed. atual. São Paulo: Pearson, 2005. xl, 1110 p. ISBN 8576050196 (Broch.)
nota: as outras edições não abordam JSP

BASHAM, Bryan; SIERRA, Kathy; BATES, Bert. **Use a cabeça!: Servlets & JSP**. 2. ed. Rio de Janeiro: Alta Books, 2008-2010. xxxii, 879 p. ISBN 9788576082941 (broch.)