



Java – Servlet – JSP

Java Server Pages é um tipo de arquivo do java (que chegou junto com **Servlet**) que te **permite programar em um arquivo HTML**, é um jeito de você dar dinamismo a páginas **HTML** no lado do **servidor**. Basicamente o **servidor renderiza o HTML** antes de enviar pro navegador:

O código escrito dentro de uma JSP é chamado de **scriptlet**.

```
1 <%
2   String nomeEmpresa = "Balura";
3   System.out.println(nomeEmpresa);
4 %>
5
6 <html>
7 <body>
8   Empresa <%= nomeEmpresa %> Recebida
9 </body>
10 </html>
```

localhost:8080/gerenciador/novaEmpresaCriada.jsp
Empresa Balura Recebida

Lembrando o funcionamento, eu tinha um **formulário html** de cadastro que chamava um **HttpServlet**, esse **Servlet** fazia sua função e já **devolvia o HTML** na resposta, só que não faz sentido usar código estático de **HTML** dentro do java, por isso a abordagem foi usar uma **JSP** pra devolver a resposta:

```
1 <DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <form action="/gerenciador/novaEmpresa" method="post">
9   Nome: <input type="text" name="nome"/>
10  <input type="submit" />
11 </form>
12 </body>
13 </html>
```

localhost:8080/gerenciador/formNovaEmpresa.html
Nome: Enviar

```
1 @WebServlet(urlPatterns = {"/novaEmpresa"})
2 public class NovaEmpresaServlet extends HttpServlet {
3     private static final long serialVersionUID = 1L;
4
5     @Override
6     protected void doPost(HttpServletRequest request, HttpServletResponse response)
7     {
8         System.out.println("Nova empresa request chegou");
9         String nomeEmpresa = request.getParameter("nome");
10        Empresa empresa = new Empresa();
11        empresa.setNome(nomeEmpresa);
12        Banco DB = new Banco();
13        DB.adicionarEmpresa(empresa);
14    }
15 }
```

Mas dessa maneira minha **JSP** esta desacoplada do **código que recebe a requisição**, então como eu faço pro **Servlet** encontrar a **JSP**?

Na verdade é bem simples, a **própria requisição** tem um **objeto de "dispatche"**, podemos pegar **ele** e **pedir para que carregue algum atributo e dispatche para um local (no caso a JSP)**, então eu posso na **JSP** pegar essa **requisição (que contém o dispatche)** e **acessar as informações que ele carrega**:

```
1 Empresa empresa = new Empresa();
2 empresa.setNome(nomeEmpresa);
3
4 Banco DB = new Banco();
5 DB.adicionarEmpresa(empresa);
6
7 RequestDispatcher rd = request.getRequestDispatcher("/novaEmpresaCriada.jsp");
8 request.setAttribute("empresa", empresa.getNome());
9 rd.forward(request, response);
```

```
1 <%
2   String nomeEmpresa = (String) request.getAttribute("empresa");
3   System.out.println(nomeEmpresa);
4 %>
5
6 <html>
7 <body>
8   Empresa <%= nomeEmpresa %> Recebida
9 </body>
10 </html>
```

Java – Scriptlet

Scriptlets é como são chamados esses códigos de java que entram no meio do **JSP**, por mais que existam alguns atalhos ainda sim é um jeito bem esquisito de se intercalar **HTML** com código Java.

Mas basicamente a ideia é que tudo que está entre “<%><%>” pode ser escrito em **Java** e obviamente, gerar dinamismo nas páginas **HTML**, então o servidor processa e renderiza o HTML que vai ser devolvido ao cliente.

Não importa quantas vezes eu olhe isso, sempre acho esquisito um <%><%> no meio de tags, até porque eles vem acompanhado com códigos. Inclusive **imports** podem ser feitos nesses **JSP** :

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ page import="java.util.List, br.com.gerenciador.servelet.Empresa" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>

    <h1>Empresas cadastradas</h1>

    <ul>
        <%
            List<Empresa> empresas = (List<Empresa>) request.getAttribute("empresas");
            for(Empresa empresa : empresas){
        %>

        <li><%= empresa.getNome() %></li>

        <% } %>
    </ul>

</body>
</html>
```