

Tecnologias Web - JSP

Gustavo Wagner - gugawag@gmail.com



RETROSPECTIVA

- Na ultima aula vimos uma loja com carrinho de compras utilizando Servlets;
- Trabalhar com Servlets eh interessante, mas *webdesigners* nao sabem como programar em Java, nem programadores Java sao especializados em gerarem paginas web elegantes;
- Precisamos de uma alternativa para gerarmos menos codigo, realizando o mesmo trabalho;

THE BIG PICTURE

Programacao em Servlet eh no ultimo deck, proximo aos negoes que colocam madeira na fornalha!! (analogia feita pelo professor Fred)

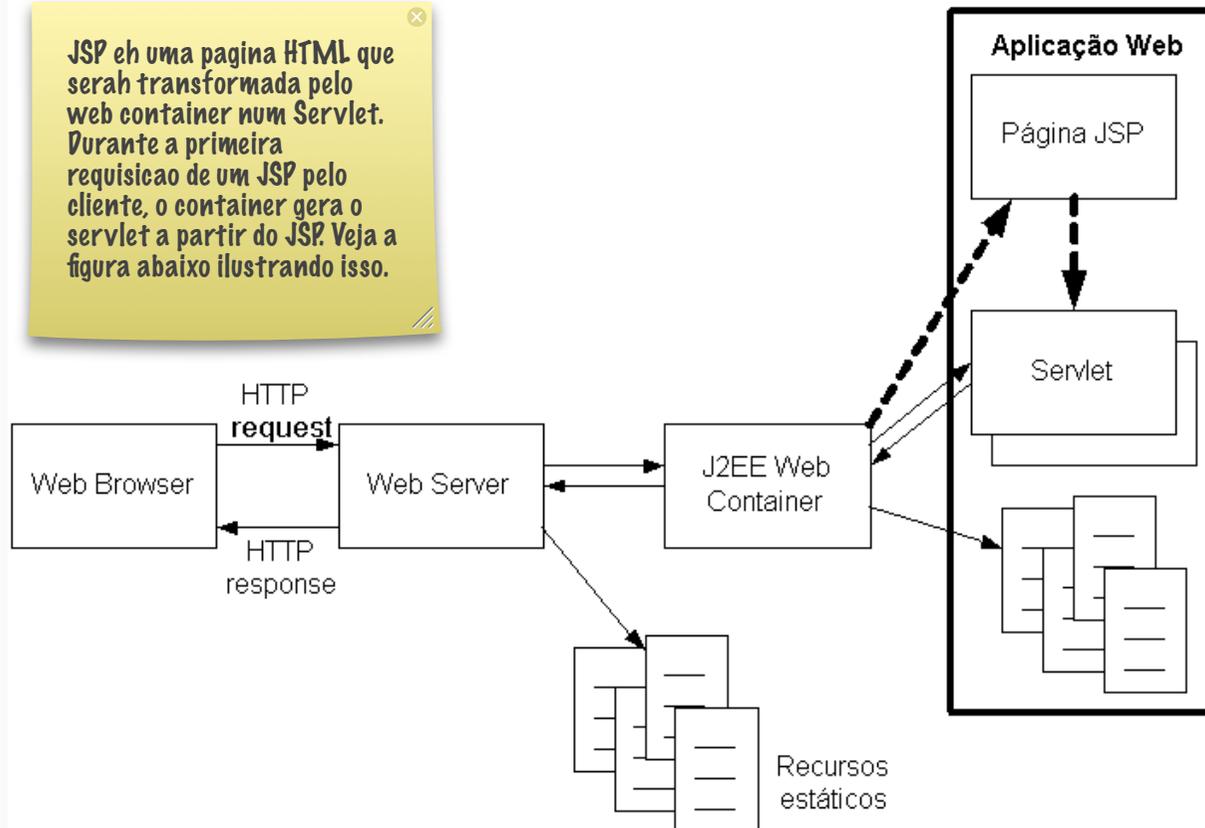


Programacao com JSF eh no ultimo deck, tomando Amarula e proximo a piscina. Mas ainda nao chegamos lah.

Vamos programar num deck intermediario, com JSP

O QUE É JSP?

JSP é uma página HTML que será transformada pelo web container num Servlet. Durante a primeira requisição de um JSP pelo cliente, o container gera o servlet a partir do JSP. Veja a figura abaixo ilustrando isso.



POSSIBILIDADES DE TRABALHO COM JSP

```
<%@ page import="java.util.*" %>
```

diretiva

Usada para passar informacao ao web container na hora da traducao JSP -> Servlet

```
<%! int y=3; %>
```

declaracao

Usada para declarar metodos ou atributos no servlet gerado pelo container.

```
email: ${applicationScope.mail}
```

expressao EL

Usada para mostrar informacoes de atributos ou parametros no JSP.

```
<%= pageContext.getAttribute("foo") %>
```

scriptlet

Usado para retornar ao html do JSP a expressao.

```
<jsp:include file="foo.html" />
```

acao

Usada para executar alguma acao no JSP. Algumas acoes serao mostradas nos proximos slides.

Boa pratica de programacao: o JSP deve ter menos scriptlets possivel. Para isso, usaremos EL, acoes e Custom Tags. Lembre-se que quem desenvolve JSP potencialmente eh um webdesigner, nao um programador Java.

USANDO BEANS

```
<jsp:useBean id="pessoa" class="modelo.Pessoa">  
    <jsp:setProperty name="pessoa" property="*">  
</jsp:useBean>
```

- Cria um objeto chamado pessoa da classe modelo.Pessoa e altera suas propriedades de acordo com os valores do request.

DIFERENÇA ENTRE REQUEST E REQUESTSCOPE

- Em EL, *request* é diferente de *requestScope*;
- *requestScope* é usado para pegar atributos (atributos das requisições);
- *request* é usado para pegar propriedades. *request* é um objeto de *pageContext*;
- ex.1: `${pageContext.request.method}` [retornará GET, POST, ou outro método HTTP]
- ex.2: `${requestScope.pessoa.nome}` [retornará o nome do objeto pessoa que está no escopo de request]
-

OBJETOS IMPLICITOS DE EL

- pageScope;
- requestScope;
- sessionScope;
- applicationScope;
- param e paramValues;
- header e headerValues;
- cookie;
- initParam;
- pageContext;

CUSTOM TAGS

- Para se usar custom tags (jstl 1.1), deve-se colocar os jars jstl.jar e standard.jar no diretorio de sua aplicacao WEB-INF/lib. Essas bibliotecas estao no tomcat;
- Alem disso, deve-se acrescentar nas suas paginas JSP a seguinte diretiva:
 - `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`

CUSTOM TAGS

```
<c:forEach var="filme" items="{filmes}">
```

```
    Filme: ${filme.nome}
```

```
</c:forEach>
```

```
<c:if test="{empty param.nome}">
```

```
    Faltou parametro nome
```

```
</c:if>
```