

DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET

Parte VI

Prof. Msc. Felipe Diniz Dallilo



Contexto do projeto

Não é comum utilizar o endereço estático do servidor, o mesmo deve ser coletado dinamicamente através de variáveis iniciadas pelo Servlet.

Uma opção é o E.L. abaixo:

- **`${appContext}`**

Utilização:

- `<script language="JavaScript"
src="${appContext}/engine.js"></script>`

Variáveis pré-definidas JSP

- Algumas variáveis já estão pré-definidas implicitamente no JSP por padrão, sendo elas:

Variável	Classe	Descrição
application	<code>javax.servlet.ServletContext</code>	O contexto do servlet da página JSP e de qualquer Web Component contidos na mesma aplicação, os dados são compartilhados por todas as páginas da aplicação
config	<code>javax.servlet.ServletConfig</code>	Informação de inicialização para o servlet da página JSP
exception	<code>java.lang.Throwable</code>	Acessível apenas a partir de uma página de erro
out	<code>javax.servlet.jsp.JspWriter</code>	O stream de saída para o conteúdo da página. Envia dados de saída de volta para o usuário. É utilizado apenas dentro de scriptlets.

Variáveis pré-definidas JSP

Variável	Classe	Descrição
page	java.lang.Object	A instância do servlet da página JSP processando o pedido atual. Raramente usado por autores de páginas JSP
pageContext	javax.servlet.jsp.PageContext	O contexto de uma página JSP. Provê uma API única para gerenciar atributos com escopo. Esta API é muito usada ao implementar "tag handlers". Oferece métodos para recuperação dos outros objetos implícitos.
request	subtipo de javax.servlet.ServletRequest	O pedido gatilhando a execução da página JSP. Dados de solicitação incluindo parâmetros. Permite a obtenção de informações sobre o request.
response	subtipo de javax.servlet.ServletResponse	A resposta retornada ao cliente. Raramente usado por autores de páginas JSP
session	javax.servlet.http.HttpSession	O objeto de sessão com o cliente. Dados de sessão específica de usuário.

Recapitulando...

Diretivas são instruções processadas quando a página JSP é compilada em Servlet:

Include - Adiciona um arquivo estático (outro JSP).

```
<%@include file="page.jsp"%>
```

Page – Define atributos que são aplicados a todo o arquivo JSP.

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

```
<%@ page extends="com.altieres.ocewcd.MyClass"%>
```

Tag Lib – Tag's JSP Customizadas

- ```
<%@ taglib uri = "http://www.example.com/custlib" prefix = "mytag" %>
```



# Diretiva page

- Contém **11 atributos** “chave/valor”:

## Info

- Usado para inserir informações sumarizadas da página, não havendo restrições ao seu tamanho. Exemplo:
- `<%@ page info="Estudo sobre Diretivas Page" %>`

## Language

- Usado, em geral, para especificar Java como a linguagem de criação de script para a página. Exemplo:
- `<%@ page language="java" %>`

## ContentType

- Este atributo indica qual o tipo MIME (Multipurpose Internet Mail Extensions) da resposta está sendo gerada pela JSP. Os tipos mais comuns são: “text/plain”, “text/html”, “text/xml”. Exemplo:
- `<%@ page contentType="text/html" %>`

# Diretiva page

## Extends

Serve para indicar a super classe que será usada pelo container JSP no momento de tradução da página em um Servlet Java. Exemplo:

```
<%@ page extends="com.taglib.jsp.primeirapagina" %>
```

## Import

- Como o nome já diz, importa classes Java que poderão ser usadas nas páginas JSPs. Exemplo:
- ```
<%@ page import="java.util.List" %>
```

Diretiva page

Session

- Session é do tipo boolean, indica se a página está participando do gerenciamento de sessão. Exemplo:
- `<%@ page session="true" %>`

Buffer

- Controla o buffer de saída da página JSP. Se for ajustado para "none" o conteúdo de uma JSP é passado instantaneamente à resposta HTTP. Exemplo:
- `<%@ page buffer="12kb" %>` ou `<%@ page buffer="none" %>`

Diretiva page

AutoFlush

- Responsável por controlar o comportamento do container JSP quando já está cheio o Buffer de saída. Neste caso é esvaziado automaticamente o Buffer de saída e o conteúdo enviado para o servidor HTTP que transmite para o Browser responsável pela solicitação. Exemplo:
- `<%@ page autoFlush="true" %>`

Atributo isThreadSafe

- Quando uma página JSP é compilada em um Servlet, ela deve ser capaz de atender a múltiplas solicitações. Para isso devemos utilizar o atributo isThreadSafe. Exemplo:
- `<%@ page isThreadSafe="false" %>`

Diretiva page

Atributo `errorPage`

- `errorPage` indica uma página alternativa que será exibida caso aconteça um erro não previsto durante o processamento de uma página JSP no container. Exemplo:
- `<%@ page errorPage="/trabalho/paginas/erro.jsp" %>`

Atributo `isErrorPage`

- Responsável por definir uma página JSP que servirá como a página de erro padrão para um grupo de páginas JSP. Sendo do tipo boolean, sua sintaxe é descrita abaixo:
- `<%@ page isErrorPage="true"%>`

Página de erro

Em um JSP, você pode definir uma página que será chamada caso ocorra algum erro no fluxo de execução Java sem precisar fazer a tratativa com try/catch e o redirecionamento.

Definindo uma página de erro

- Crie um JSP com a seguinte sintaxe:
- `<form action = "index.jsp" >` -- JSP de retorno
- `<%@ page isErrorPage = "true" %>` -- Diretiva para página de erro
- `<h3>Desculpe, um erro ocorreu </h3>` -- Mensagem HTML de Erro
- `Exception is: <%= exception %>
` -- Exception Java (Não é uma boa prática)
- `<input type="submit" value="retornar"/>` -- Botão HTML
- `</form>`

Definindo um JSP de teste

- Crie um jsp chamado index.jsp com o seguinte bloco:

```
<%@ page errorPage = "error.jsp" %> -- Página de erro
<% -- Scriptlet
    String num1 =request.getParameter( "n1" );
    int a = Integer.parseInt(num1);
    out.print( "O número é: " +a );
%>
```


Definindo o erro para todos

- No Web.xml é possível definir o redirecionamento a seguir:

```
<error-page>
```

```
    <exception-type>java.lang.Exception</exception-type>    --
```

Caso ocorra este erro

```
    <location>/error.jsp</location>    -- Redireciona para este JSP.
```

```
</error-page>
```

Erro específico (código)

- É possível identificar o código do erro e redirecionar para o JSP adequado:

```
<error-page>
```

```
    <error-code>500</error-code> -- Código do erro
```

```
    <location>/error.jsp</location> -- JSP adequado.
```

```
</error-page>
```

Definindo mais de um

- No WEB.xml pode ser definido mais de uma página de erro (por código de erro ou exception)

```
<error-page>
```

```
    <error-code>403</error-code> -- Código do erro
```

```
    <location>/error1.jsp</location> -- JSP adequado.
```

```
</error-page>
```

```
<error-page>
```

```
    <error-code>500</error-code> -- Código do erro
```

```
    <location>/error2.jsp</location> -- JSP adequado.
```

```
</error-page>
```

Código de erros (Relembrando)

- **1xx Informativa**
 - 100 Continuar
 - 101 Mudando protocolos.
 - 122 Pedido-URI muito longo
- **2xx Sucesso**
 - 200 OK
 - 201 Criado
 - 202 Aceito
 - 203 não-autorizado (desde HTTP/1.1)
 - 204 Nenhum conteúdo
 - 205 Reset
 - 206 Conteúdo parcial
 - 207-Status Multi (WebDAV) (RFC 4918)
- **3xx Redirecionamento**
 - 300 Múltipla escolha
 - 301 Movido
 - 302 Encontrado
 - 304 Não modificado
 - 305 Use Proxy (desde HTTP/1.1)
 - 306 Proxy Switch
- 307 Redirecionamento temporário (desde HTTP/1.1)
- **4xx Erro de cliente**
 - 400 Requisição inválida
 - 401 Não autorizado
 - 402 Pagamento necessário
 - 403 Proibido
 - 404 Não encontrado
 - 405 Método não permitido
 - 406 Não Aceitável
 - 407 Autenticação de proxy necessária
 - 408 Tempo de requisição esgotou (Timeout)
 - 409 Conflito
 - 410 Gone
 - 411 comprimento necessário
 - 412 Pré-condição falhou
 - 413 Entidade de solicitação muito grande •
 - 414 Pedido-URI Too Long
 - 415 Tipo de mídia não suportado
 - 416 Solicitada de Faixa Não Satisfatória
 - 417 Falha na expectativa
 - 418 Eu sou um bule de chá
 - 422 Entidade improcessável (WebDAV) (RFC 4918)
 - 423 Fechado (WebDAV) (RFC 4918)
 - 424 Falha de Dependência (WebDAV) (RFC 4918)
 - 425 coleção não ordenada (RFC 3648)
 - 426 Upgrade Obrigatório (RFC 2817)
 - 450 bloqueados pelo Controle de Pais do Windows
 - 499 cliente fechou Pedido (utilizado em ERPs/VPsA)
- **5xx outros erros**
 - 500 Erro interno do servidor (Internal Server Error)
 - 501 Não implementado (Not implemented)
 - 502 Bad Gateway
 - 503 Serviço indisponível (Service Unavailable)
 - 504 Gateway Time-Out
 - 505 HTTP Version not supported

Erro padrão

HTTP Status 404 – Not Found

Type Status Report

Message /BookStoreWebsite/view_books

Description The origin server did not find a current representation for the target resource or is not willing to disclose that one exists.

Apache Tomcat/9.0.16

Erros tratados



400. That's an error.

Your client has issued a malformed or illegal request.
That's all we know.



facebook

Sorry, something went wrong.

We're working on it and we'll get it fixed as soon as we can

[Go Back](#)

Facebook © 2014 · [Help Center](#)

Referências

- Deitel, Paul J.; Deitel, Harvey M. Java: como programar - 8ª edição. Editora Pearson. ISBN: 9788576055631
- Barnes, David.; Kölling, Michael. Programação Orientada a Objetos com Java: uma introdução prática usando o Blue J - 1ª edição. 2013. Editora Pearson. ISBN: 9788576050124
- Puga, Sandra; Rissetti, Gerson. Lógica de Programação e Estruturas de Dados: com aplicações em Java. 1ª edição. 2013. Editora Pearson. ISBN: 9788587918826
- Lemay, Laura; Colburn, Rafe; Tyler, Denise. Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias. Editora Pearson. 2013. 1ª Edição. ISBN: 9788534614283
- Chak, Andrew. Como Criar Sites Persuasivos. Editora Pearson. 2012. 1ª Edição. ISBN: 9788534615112
- Fábio Flatschart; Clécio Bachini; Cesar Cusin. Open Web Platform. Editora Pearson. 2013. 1ª Edição. ISBN: 9788574526140