

Objetos para manipulação de páginas jsp

Objetos implícito

O container disponibiliza nove objetos implícitos para manipulação em suas páginas jsp. Esses objetos são instanciados automaticamente. Esses nove objetos são divididos em duas categorias:

Servlet API: application, config, request, response e session

JSP API: exception, out, page, pageContext

Objeto exception

Esse objeto `java.lang.Throwable`. É importante salientar que esse objeto somente estará disponível naquelas páginas em que o atributo `isErrorPage` for definido como `"true"`. Esse objeto visa permitir o desenvolvedor tratar de forma apropriada a ocorrência de exceções.

gerarErro.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page errorPage="erro.jsp" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Gera uma exceção</title>
</head>
<body>
<%
String a[] = {"a", "b"};
%>
<%=a[2] %>
</body>
</html>
```

erro.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page isErrorPage="true" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Tratamento da exceção</title>
</head>
<body>
<%
if(exception!=null){
    %>
    <h1 style="color: red;">Ocorreu uma exceção!</h1>
    <h3 style="color: red;"><%=exception.toString() %></h3>
    %>
} else{
    %>
    <h1 style="color: blue;">Não ocorreu exceção!</h1>
    %>
}
}
```

```
%>
</body>
</html>
```

Objeto application

Representa o contexto da aplicação, oferecendo métodos que trazem informações sobre o ambiente em que o Servlet gerado é executado, podendo ainda armazenar informações(objetos), que podem ser compartilhados por todos os Servlets da aplicação. O objeto implementa a interface javax.Servlet.ServletContext.

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Objeto application</title>
</head>
<body>
<%
//String compartilhada

application.setAttribute("texto", "Posso compartilhar com todas as páginas da
minha aplicação!");

%>
Servidor e versão:
<%=application.getServerInfo()%>
</body>
</html>
```

index2.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Objeto application 2</title>
</head>
<body>
Texto compartilhado através do objeto application:
<%=application.getAttribute("texto")%>
</body>
</html>
```

Objeto out

Instância da classe java.Servlet.jsp.JspWriter. Esse objeto é usado para escrever caracteres na página HTML de resposta;

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
        pageEncoding = "ISO-8859-1"%>
<%@page import = "java.util. *" %>
<%@page import = "java.text. *" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset=ISO-8859-1">
<title>Objeto out</title>
</head>
<body>
    <%
        Date d = new Date();
        Calendar c = Calendar.getInstance();
        c.setTime(d);
        DateFormat df = DateFormat.getDateInstance(DateFormat.FULL);
        //Escreve na saída a data atual
        out.println(df.format(c.getTime()));
    %>
</body>
</html>
```

Obejto request

Instância da classe javax.Servlet.http.HttpServletRequest. Objeto responsável por modelar as requisições enviadas às páginas jsp tanto por get como por post. Sua manipulação é feita de forma semelhante à feita em Servlet.

Nesse exemplo, faremos o mesmo sistema de geração de números para loteria feito na aula anterior, contudo usaremos jsp e não Servlet.

A página que contem o formulário pode continuar sendo a mesma, alterando-se apenas o atributo action do formulário para apontar para nossa nova página jsp.

formulario.html

```
<html>
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset=ISO-8859-1">
<title>Loteria</title>
</head>
<body>
    <h3>Sistema de geração de números para loteria</h3>
    <form action = "loteria.jsp">
        <table>
            <tr>
                <td>Nome: </td>
                <td><input type = "text" name = "nome" />
                </td>
            </tr>
            <tr>
                <td>Quantos números você quer jogar: </td>
                <td><input type = "text" name = "quantidadeDeNumeros" />
                </td>
            </tr>
            <tr>
                <td>Quantas apostas você quer fazer: </td>
                <td><input type = "text" name = "quantidadeDeApostas" />
                </td>
            </tr>
        </table>
    </form>
</body>
```

```
        </tr>
      <tr>
        <td colspan="2"><input type="submit" value="Enviar" />
      </td>
    </tr>
  </table>
</form>
</body>
</html>
loteria.jsp

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Loteria</title>
</head>
<body>
<%
//Pegamos os parâmetros através do objeto request
String nome = request.getParameter("nome");
String quantidadeDeNumeros = request.getParameter("quantidadeDeNumeros");
String quantidadeDeApostas = request.getParameter("quantidadeDeApostas");

int quantidadeDeNumerosInt = Integer.parseInt(quantidadeDeNumeros);
int quantidadeDeApostasInt = Integer.parseInt(quantidadeDeApostas);

String resposta = "<h1> "+nome+" suas apostas</h1>";

for(int i = 0; i < quantidadeDeApostasInt; i++){
    resposta += "<b>Aposta "+ (i+1) + "</b>: ";
    for(int j = 0; j < quantidadeDeNumerosInt; j++){
        resposta += ((int) (Math.random()*59))+1; ;
        if(j==quantidadeDeNumerosInt-1){
            resposta += "<br/>";
        }else{
            resposta += ";";
        }
    }
}
}
out.println(resposta);
%>
</body>
</html>
```

Objeto response

Semelhante ao objeto request, porém implementa a interface `javax.Servlet.http.HttpServletResponse`.

Objeto session

É uma instância da classe `javax.Servlet.http.HttpSession`, podendo controlar a sessão do usuário. Cada vez que um usuário acessa uma página/Servlet existente no container é

criado um objeto session, que é capaz de manter informações sobre aquele usuário. Vale lembrar que a sessão é criada entre o container e o navegador do cliente, sendo assim, se um usuário abrir a mesma página em dois navegadores diferentes em uma mesma máquina, serão criadas duas sessões, uma para cada navegador. Cada objeto HttpSession tem um tempo de vida chamado de time-out. Uma vez que esse time-out, **XXXXXXXXXXXX** a sessão será interrompida e as informações nela contida descartadas. Esse time-out pode ser alterado através do objeto session, ou no arquivo de configuração web.xml, tanto do contexto, como do container.

Objeto config

O objeto config interface javax.Servlet.ServletConfig, desta forma fornece vários métodos que visam manipular as configurações de inicialização do Servlet gerado. Os parâmetros de inicialização são inseridos através do arquivo web.xml.

Objeto page

O objeto referencia a própria página(*this*). Na prática, não é utilizado, tendo seu uso destinado ao futuro, caso o container passe a suportar outras linguagens de programação além da Java.

Objeto pageContext

Referencia um objeto javax.Servlet.jsp.PageContext. Esse objeto que mantém as referências aos outros objetos implícitos. Cabe a ele provê acesso aos diferentes escopos existentes.