Tutoriais

Objetos Implícitos
Parte 3: Objetos Contextuais

por: Ulisses Telemaco Neto
ulisses@jspbrasil.com.br

Os objetos implícitos dessa categoria fornece à página JSP acesso ao contexto dentro do qual ela está respondendo. Os quatro objetos implícitos contextuais são:
request;
session;
application;
pageContext.

Todos eles têm a capacidade de armazenar e recuperar valores de atributos arbitrários. Os atributos de página, armazenados no objeto pageContext, duram apenas enquanto o processamento de uma única página ocorre. Os atributos de solicitação, armazenados no objeto request, também tem pouca duração, mas podem ser transferidos entre páginas quando for transferido o controle. Os atributos de sessão, armazenados no objeto session, duram enquanto o usuário continuar a interagir com o servidor da web. Os atributos de aplicação, armazenados no objeto application, são mantidos enquanto o container JSP mantiver uma ou mais páginas de uma aplicação carregada na memória - enquanto o container JSP estiver rodando. A tabela a seguir traz os métodos comuns aos quatro objetos e que são usados para armazenar e recuperar valores de atributos:

Tabela 1: Os métodos comuns a esses quatro objetos e que são usados para armazenar e recuperar valores de atributos:

Métodos	Descrição
void setAttribute(String key, Object value)	Associa um valor de atributo com um nome.
Enumeration getAttributeNames()	Recupera os nomes de todos os atributos associados com o objeto.
Object getAttribute(String key)	Recupera o valor de atributo associado com a chave.
void removeAttribute(String key)	Remove o valor de atributo associado com a chave.

Objeto session

Este objeto representa a sessão atual de um usuário individual. Todas as solicitações feitas por um usuário são consideradas parte de uma sessão. Desde que novas solicitações por aqueles usuários continuem a ser recebidas pelo servidor, a sessão persiste. Se, no entanto, um certo período de tempo passar sem que qualquer nova solicitação do usuário seja recebida, a sessão expira.

O objeto session armazena informações a respeito da sessão. Um dos principais usos para o objeto session é armazenar e recuperar valores de atributos, a fim de transmitir as informações específicas de usuários entre as páginas. Abaixo segue um exemplo que armazena dados na sessão, na forma de um objeto que é instância de uma classe hipotética "Usuario":

```
<%
Usuario u = new Usuario(nome, senha);
session.setAttribute("usuario", u);
%>
```

Uma vez que um objeto tenha sido armazenado através do método setAttibute(), ele pode ser recuperado - na mesma página ou em outra acessada pelo usuário. O código abaixo ilustra a recuperação do objeto armazenado no código anterior:

```
<%
Usuario u = (Usuario)session.getAttribute("usuario");
.....
%>
```

Perceba que o método getAttribute() retorna um objeto da classe Object, portanto, é necessário fazermos um cast para converter o objeto retornado em uma instância da classe desejada.

O objeto session implementa a interface javax.servlet.http.HttpSession. A tabela abaixo traz os principais métodos utilizados por esse objeto, além daqueles descritos anteriormente na tabela 1:

Métodos	Descrição
Object getAttibute(String nome)	Recupera o objeto identificado por "nome".
String getId()	Retorna o Id da sessão.
long getCreationTime()	Retorna a hora na qual a sessão foi criada.
long getLastAccessedTime()	Retorna a última vez que uma solicitação associada com a sessão foi recebida.
int getMaxInactiveInterval()	Retorna o tempo máximo (em segundos) entre solicitações pelo qual a sessão será mantida.
void setMaxInactiveInterval(int time)	Define o tempo máximo (em segundos) entre solicitações pelo qual a sessão será mantida.
boolean isNew()	Retorna se o navegador do usuário ainda não tiver confirmado o ID de sessão.
boolean invalidate()	Descarta a sessão, liberando quaisquer objetos armazenados como atributos.

O link abaixo exibe a API completa da interface que implementa o objeto session: HttpSession

Objeto application

Este objeto representa a aplicação à qual a página JSP pertence. Ele é uma instância da interface javax.servlet.ServletContext. Os containers JSP tipicamente tratam do primeiro nome de diretório em um URL como uma aplicação. Exemplo:

```
http://server/jspbrasil/index.jsp
http://server/jspbrasil/principal.jsp
http://server/jspbrasil/forum/forum.jsp
http://server/jspbrasil/tutorial/tutorial.jsp
```

São todos considerados parte da mesma aplicação jspbrasil.

Além dos métodos descritos na tabela 1, os métodos do objeto application podem ser divididos em quatro funcionalidades:

Recuperar informações de versão do container servlet:

Método	Descrição
String getServerInfo()	Retorna o nome e versão do container servlet.
int getMajorVersion()	Retorna a versão principal da API do servlet para o container servlet.
int getMinorVersion()	Retorna a versão secundária da API do servlet para o container servlet.

Interagir com arquivos e caminhos no servidor:

Método	Descrição
String getMimeType(String filename)	Retorna o tipo MIME para o arquivo indicado, se conhecido pelo servidor.
URL getResource(String path)	Traduz uma cadeia especificando um URL em um objeto que acessa os conteúdos dos URLs, localmente ou na rede.
InputStream getResourceAsStream(String path)	Traduz uma cadeia especificando um URL em um fluxo de entrada para ler seu conteúdo.
String getRealPath(String path)	Traduz um URL local em um nome de caminho no sistema de arquivo local.
ServletContext getContext(String path)	Retorna o contexto de aplicação para o URL local especificado.
RequestDispatcher getRequestDispatcher(String path)	Cria um dispacher de solicitação para o URL local indicado.

Suporte para log de mensagens:

Método	Descrição
void log(String message)	Grava a mensagem o arquivo de log.
void log(String message, Exception e)	Grava a mensagem no aqrquivo de log, junto com a trilha de pilha para a exceção especificada.

Acessar parâmetros de inicialização:

Método	Descrição
Enumerations getInitParameterNames()	Recupera os nomes de todos os parâmetros de inicialização.
String getInitParameter(String name)	Recupera o valor do parâmetro de inicilização como o nome dado.

O link abaixo exibe a API completa da interface que implementa o objeto application: ServletContext

Objeto pageContext

O objeto pageContext fornece várias facilidades como gerenciamento de sessões, atributos, páginas de erro, inclusões e encaminhamento de requisições de fluxo de resposta.
O objeto pageContext é uma instância da classe javax.servlet.jsp.PageContext. Além dos métodos descritos na tabela 1, os principais métodos desse objeto podem ser divididos em três categorias:

Acessar outros objetos implícitos de JSP:

Método	Descrição
Object getPage()	Retorna a instância de servlet para a página atual (objeto implícito page).
ServletRequest getRequest()	Retorna a solicitação que iniciou o processamento da página (objeto implícito request).
ServletResponse	Retorna a respota para a página (objeto implícito response).
JspWriter getOut	Retorna o fluxo de saída atual para a página (objeto implícito out).
HttpSession getSession()	Retorna a sessão associada com a solicitação da página atual, se houver alguma (objeto implícito session).
ServletConfig getServletConfig()	Retorna o objeto de configuração de servlet (objeto implícito config).
ServletContext getServletContext()	Retorna o contexto no qual o servlet da página roda (objeto implícito application).
Exception getException()	Para páginas de erro, retorna a exceção passada para a página (objeto implícito exception).

Envio de solicitações de uma página JSP para outra:

Método	Descrição
void forward(String path)	Encaminha o processamento para um outro URL local dado pela String path.
void include(String path)	Inclui o output do processamento de um outro URL local.

Acessar atributos através de múltiplos escopos:

Método	Descrição
void setAttribute(String key, Object obj, int scope)	Associa o valor do atributo "obj" com a chave "key" no escopo "scope".
Enumeration getAttributeNamesInScope(int scope)	Recupera os nomes de todos os atributos associado com "key" no escopo "scope".
Object getAttribute(String name, int scope)	Recupera o valor de tributo associado com "name" no escopo "scope"
removeAttribute(String name, int scope)	Remove o valor de atributo associado com "name" no escope "scope"
Object findAttribute(String name)	Procura em todos os escopos pelo atributo associado com "name".
int getAttributesScope(String name)	Retorna o escopo no qual o atributo associado com "name" está armazenado.

Os últimos dois métodos listados na tabela anterior permitem a procura, através de todos os escopos definidos, por um atributo associado com uma String passada como parâmetro. Nos dois casos, o objeto pageContext irá realizar uma busca através dos escopos na seguinte ordem: pageContext, request, session e application.
A tabela anterior traz métodos que recebe parâmetros para especificar o escopo. A classe javax.servlet.jsp.PageContext fornece variáveis estáticas para representar estes quatro escopos diferentes. A tabela a seguir resume estas variáveis:

Variável	Descrição
PAGE_SCOPE	Escopo para atributos armazenados no objeto pageContext.
REQUEST_SCOPE	Escopo para atributos armazenados no objeto request.
SESSION_SCOPE	Escopo para atributos armazenados no objeto session.
APPLICATION_SCOPE	Escopo para atributos armazenados no objeto application.

O exemplo abaixo ilustra o uso do método "getAttribute" e das variáveis estáticas descritas na tabela anterior:

```
<%
User uPag=(User)pageContext.getAttibute("user",pageContext.PAGE_SCOPE)
//Recupera o object "usuario" do escopo pageContext

User uReq=(User)pageContext.getAttibute("user",pageContext.REQUEST_SCOPE)
//Recupera o object "usuario" do escopo request

User uSes=(User)pageContext.getAttibute("user",pageContext.SESSION_SCOPE)
//Recupera o object "usuario" do escopo session

User uApp=(User)pageContext.getAttibute("user",pageContext.APPLICATION_SCOPE)
//Recupera o object "usuario" do escopo application
%>
```

O link abaixo exibe a API completa da interface que implementa o objeto pageContext: PageContext