

Pengantar : Objek Implisit

Chocolove Mic

chocolove_mic@yahoo.co.uk

<http://www.mycgiserver.com/~chocolove2003>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Biasanya pada Java objek dibuat dari class-nya. Pada JSP ada beberapa objek dimana user tidak perlu membuatnya secara manual dari class-nya yang disebut Objek Implisit. Objek-objek tersebut adalah :

- Objek request
- Objek response
- Objek pageContext
- Objek session
- Objek application
- Objek out
- Objek config
- Objek page
- Objek exception

Objek-objek ini hanya dapat digunakan di antara tag scriptlet dan ekspresi.

Objek request

Objek ini adalah hasil cetakan dari class javax.servlet.ServletRequest. Otomatis objek request akan memiliki properti dan method yang dimiliki oleh class ServletRequest. Berikut adalah daftar method-methodnya :

Method Summary	
java.lang.Object	getAttribute (java.lang.String name) Returns the value of the named attribute as an Object, or null if no attribute of the given name exists.
java.util.Enumeration	getAttributeNames () Returns an Enumeration containing the names of the attributes available to this request.
java.lang.String	getCharacterEncoding () Returns the name of the character encoding used in the body of this request.

int	<code>getContentLength()</code> Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known.
java.lang.String	<code>getContentType()</code> Returns the MIME type of the body of the request, or null if the type is not known.
ServletInputStream	<code>getInputStream()</code> Retrieves the body of the request as binary data using a ServletInputStream.
java.util.Locale	<code>getLocale()</code> Returns the preferred Locale that the client will accept content in, based on the Accept-Language header.
java.util.Enumeration	<code>getLocales()</code> Returns an Enumeration of Locale objects indicating, in decreasing order starting with the preferred locale, the locales that are acceptable to the client based on the Accept-Language header.
java.lang.String	<code>getParameter(java.lang.String name)</code> Returns the value of a request parameter as a String, or null if the parameter does not exist.
java.util.Map	<code>getParameterMap()</code> Returns a java.util.Map of the parameters of this request.
java.util.Enumeration	<code>getParameterNames()</code> Returns an Enumeration of String objects containing the names of the parameters contained in this request.
java.lang.String[]	<code>getParameterValues(java.lang.String name)</code> Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.
java.lang.String	<code>getProtocol()</code> Returns the name and version of the protocol the request uses in the form <i>protocol/majorVersion.minorVersion</i> , for example, HTTP/1.1.
java.io.BufferedReader	<code>getReader()</code> Retrieves the body of the request as character data using a BufferedReader.
java.lang.String	<code>getRealPath(java.lang.String path)</code> Deprecated. <i>As of Version 2.1 of the Java Servlet API, use <code>ServletContext.getRealPath(java.lang.String)</code> instead.</i>
java.lang.String	<code>getRemoteAddr()</code> Returns the Internet Protocol (IP) address of the client that sent the request.
java.lang.String	<code>getRemoteHost()</code> Returns the fully qualified name of the client that sent

	the request.
<code>RequestDispatcher</code>	<code>getRequestDispatcher</code> (java.lang.String path) Returns a <code>RequestDispatcher</code> object that acts as a wrapper for the resource located at the given path.
java.lang.String	<code>getScheme</code> () Returns the name of the scheme used to make this request, for example, http, https, or ftp.
java.lang.String	<code>getServerName</code> () Returns the host name of the server that received the request.
int	<code>getServerPort</code> () Returns the port number on which this request was received.
boolean	<code>isSecure</code> () Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.
void	<code>removeAttribute</code> (java.lang.String name) Removes an attribute from this request.
void	<code>setAttribute</code> (java.lang.String name, java.lang.Object o) Stores an attribute in this request.
void	<code>setCharacterEncoding</code> (java.lang.String env) Overrides the name of the character encoding used in the body of this request.

Contoh :

```

objek_request.jsp
IP Address : <%= request.getRemoteAddr() %>
<br>
Host : <%= request.getRemoteHost() %>
<br>
Server : <%= request.getServerName() %>
<br>
<%
out.println("Port : "+request.getServerPort());
%>
    
```

Sumber : -

Objek response

Objek ini adalah hasil cetakan dari class `javax.servlet.ServletResponse`. Otomatis objek request akan memiliki properti dan method yang dimiliki oleh class `ServletResponse`. Berikut adalah daftar method-methodnya :

Method Summary	
void	<code>flushBuffer</code> () Forces any content in the buffer to be written to the client.
int	<code>getBufferSize</code> ()

	Returns the actual buffer size used for the response.
java.lang.String	getCharacterEncoding() Returns the name of the charset used for the MIME body sent in this response.
java.util.Locale	getLocale() Returns the locale assigned to the response.
ServletOutputStream	getOutputStream() Returns a ServletOutputStream suitable for writing binary data in the response.
java.io.PrintWriter	getWriter() Returns a <code>PrintWriter</code> object that can send character text to the client.
boolean	isCommitted() Returns a boolean indicating if the response has been committed.
void	reset() Clears any data that exists in the buffer as well as the status code and headers.
void	resetBuffer() Clears the content of the underlying buffer in the response without clearing headers or status code.
void	setBufferSize(int size) Sets the preferred buffer size for the body of the response.
void	setContentLength(int len) Sets the length of the content body in the response In HTTP servlets, this method sets the HTTP Content-Length header.
void	setContentType(java.lang.String type) Sets the content type of the response being sent to the client.
void	setLocale(java.util.Locale loc) Sets the locale of the response, setting the headers (including the Content-Type's charset) as appropriate.

Objek pageContext

Objek ini merupakan cetakan dari class javax.servlet.jsp.PageContext.

Berikut adalah properti yang dimiliki oleh objek ini :

Field Summary	
static java.lang.String	APPLICATION Name used to store ServletContext in PageContext name table.
static int	APPLICATION_SCOPE Application scope: named reference remains available in the ServletContext until it is reclaimed.

<code>static java.lang.String</code>	CONFIG Name used to store ServletConfig in PageContext name table.
<code>static java.lang.String</code>	EXCEPTION Name used to store uncaught exception in ServletRequest attribute list and PageContext name table.
<code>static java.lang.String</code>	OUT Name used to store current JspWriter in PageContext name table.
<code>static java.lang.String</code>	PAGE Name used to store the Servlet in this PageContext's nametables.
<code>static int</code>	PAGE_SCOPE Page scope: (this is the default) the named reference remains available in this PageContext until the return from the current Servlet.service() invocation.
<code>static java.lang.String</code>	PAGECONTEXT Name used to store this PageContext in it's own name table.
<code>static java.lang.String</code>	REQUEST Name used to store ServletRequest in PageContext name table.
<code>static int</code>	REQUEST_SCOPE Request scope: the named reference remains available from the ServletRequest associated with the Servlet until the current request is completed.
<code>static java.lang.String</code>	RESPONSE Name used to store ServletResponse in PageContext name table.
<code>static java.lang.String</code>	SESSION Name used to store HttpSession in PageContext name table.
<code>static int</code>	SESSION_SCOPE Session scope (only valid if this page participates in a session): the named reference remains available from the HttpSession (if any) associated with the Servlet until the HttpSession is invalidated.

Sedangkan method yang dimiliki oleh objek ini adalah :

Method Summary	
<code>abstract java.lang.Object</code>	findAttribute (java.lang.String name) Searches for the named attribute in page, request, session (if valid), and application scope(s) in order and returns the value associated or null.
<code>abstract void</code>	forward (java.lang.String relativeUrlPath)

	This method is used to re-direct, or "forward" the current ServletRequest and ServletResponse to another active component in the application.
abstract java.lang.Object	getAttribute (java.lang.String name) Return the object associated with the name in the page scope or null if not found.
abstract java.lang.Object	getAttribute (java.lang.String name, int scope) Return the object associated with the name in the specified scope or null if not found.
abstract java.util.Enumeration	getAttributeNamesInScope (int scope) Enumerate all the attributes in a given scope
abstract int	getAttributeScope (java.lang.String name) Get the scope where a given attribute is defined.
abstract java.lang.Exception	getException () The current value of the exception object (an Exception).
abstract JspWriter	getOut () The current value of the out object (a JspWriter).
abstract java.lang.Object	getPage () The current value of the page object (a Servlet).
abstract ServletRequest	getRequest () The current value of the request object (a ServletRequest).
abstract ServletResponse	getResponse () The current value of the response object (a ServletResponse).
abstract ServletConfig	getServletConfig () The ServletConfig instance.
abstract ServletContext	getServletContext () The ServletContext instance.
abstract HttpSession	getSession () The current value of the session object (an HttpSession).
abstract void	handlePageException (java.lang.Exception e) This method is intended to process an unhandled "page" level exception by redirecting the exception to either the specified error page for this JSP, or if none was specified, to perform some implementation dependent action.
abstract void	handlePageException (java.lang.Throwable t) This method is identical to the handlePageException(Exception), except that it accepts a Throwable.
abstract void	include (java.lang.String relativeUrlPath) Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse

	being processed by the calling Thread.
abstract void	<p>initialize(Servlet servlet, ServletRequest request, ServletResponse response, java.lang.String errorPageURL, boolean needsSession, int bufferSize, boolean autoFlush)</p> <p>The initialize method is called to initialize an uninitialized PageContext so that it may be used by a JSP Implementation class to service an incoming request and response within it's _jspService() method.</p>
JspWriter	<p>popBody()</p> <p>Return the previous JspWriter "out" saved by the matching pushBody(), and update the value of the "out" attribute in the page scope attribute namespace of the PageContext</p>
BodyContent	<p>pushBody()</p> <p>Return a new BodyContent object, save the current "out" JspWriter, and update the value of the "out" attribute in the page scope attribute namespace of the PageContext</p>
abstract void	<p>release()</p> <p>This method shall "reset" the internal state of a PageContext, releasing all internal references, and preparing the PageContext for potential reuse by a later invocation of initialize().</p>
abstract void	<p>removeAttribute(java.lang.String name)</p> <p>Remove the object reference associated with the given name, look in all scopes in the scope order.</p>
abstract void	<p>removeAttribute(java.lang.String name, int scope)</p> <p>Remove the object reference associated with the specified name in the given scope.</p>
abstract void	<p>setAttribute(java.lang.String name, java.lang.Object attribute)</p> <p>Register the name and object specified with page scope semantics.</p>
abstract void	<p>setAttribute(java.lang.String name, java.lang.Object o, int scope)</p> <p>register the name and object specified with appropriate scope semantics</p>

Objek session

Objek ini merupakan cetakan dari class javax.servlet.http.HttpSession. Method yang dimiliki oleh objek ini adalah sebagai berikut :

Method Summary

java.lang.Object	getAttribute (java.lang.String name) Returns the object bound with the specified name in this session, or null if no object is bound under the name.
java.util.Enumeration	getAttributeNames () Returns an Enumeration of String objects containing the names of all the objects bound to this session.
long	getCreationTime () Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
java.lang.String	getId () Returns a string containing the unique identifier assigned to this session.
long	getLastAccessedTime () Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container recieved the request.
int	getMaxInactiveInterval () Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
ServletContext	getServletContext () Returns the ServletContext to which this session belongs.
HttpSessionContext	getSessionContext () Deprecated. <i>As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.</i>
java.lang.Object	getValue (java.lang.String name) Deprecated. <i>As of Version 2.2, this method is replaced by getAttribute(java.lang.String).</i>
java.lang.String[]	getValueNames () Deprecated. <i>As of Version 2.2, this method is replaced by getAttributeNames()</i>
void	invalidate () Invalidates this session then unbinds any objects bound to it.
boolean	isNew () Returns true if the client does not yet know about the session or if the client chooses not to join the session.
void	putValue (java.lang.String name, java.lang.Object value) Deprecated. <i>As of Version 2.2, this method is replaced by setAttribute(java.lang.String, java.lang.Object)</i>
void	removeAttribute (java.lang.String name) Removes the object bound with the specified name from this session.

void	<code>removeValue</code> (java.lang.String name) Deprecated. As of Version 2.2, this method is replaced by removeAttribute(java.lang.String)
void	<code>setAttribute</code> (java.lang.String name, java.lang.Object value) Binds an object to this session, using the name specified.
void	<code>setMaxInactiveInterval</code> (int interval) Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.

Objek application

Objek ini merupakan hasil cetakan dari class javax.servlet.ServletContext. method-method yang dimilikinya adalah :

Method Summary	
java.lang.Object	<code>getAttribute</code> (java.lang.String name) Returns the servlet container attribute with the given name, or null if there is no attribute by that name.
java.util.Enumeration	<code>getAttributeNames</code> () Returns an Enumeration containing the attribute names available within this servlet context.
ServletContext	<code>getContext</code> (java.lang.String uripath) Returns a ServletContext object that corresponds to a specified URL on the server.
java.lang.String	<code>getInitParameter</code> (java.lang.String name) Returns a String containing the value of the named context-wide initialization parameter, or null if the parameter does not exist.
java.util.Enumeration	<code>getInitParameterNames</code> () Returns the names of the context's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the context has no initialization parameters.
int	<code>getMajorVersion</code> () Returns the major version of the Java Servlet API that this servlet container supports.
java.lang.String	<code>getMimeType</code> (java.lang.String file) Returns the MIME type of the specified file, or null if the MIME type is not known.
int	<code>getMinorVersion</code> () Returns the minor version of the Servlet API that this servlet container supports.
RequestDispatcher	<code>getNamedDispatcher</code> (java.lang.String name) Returns a RequestDispatcher object that acts as a wrapper for the named servlet.

java.lang.String	getRealPath (java.lang.String path) Returns a String containing the real path for a given virtual path.
RequestDispatcher	getRequestDispatcher (java.lang.String path) Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.
java.net.URL	getResource (java.lang.String path) Returns a URL to the resource that is mapped to a specified path.
java.io.InputStream	getResourceAsStream (java.lang.String path) Returns the resource located at the named path as an InputStream object.
java.util.Set	getResourcePaths (java.lang.String path) Returns a directory-like listing of all the paths to resources within the web application whose longest sub-path matches the supplied path argument.
java.lang.String	getServerInfo () Returns the name and version of the servlet container on which the servlet is running.
Servlet	getServlet (java.lang.String name) Deprecated. <i>As of Java Servlet API 2.1, with no direct replacement.</i> <i>This method was originally defined to retrieve a servlet from a ServletContext. In this version, this method always returns null and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i> <i>In lieu of this method, servlets can share information using the ServletContext class and can perform shared business logic by invoking methods on common non-servlet classes.</i>
java.lang.String	getServletContextName () Returns the name of this web application corresponding to this ServletContext as specified in the deployment descriptor for this web application by the display-name element.
java.util.Enumeration	getServletNames () Deprecated. <i>As of Java Servlet API 2.1, with no replacement.</i> <i>This method was originally defined to return an Enumeration of all the servlet names known to this context. In this version, this method always returns an empty Enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i>

java.util.Enumeration	<p>getServlets() Deprecated. As of Java Servlet API 2.0, with no replacement.</p> <p><i>This method was originally defined to return an Enumeration of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</i></p>
void	<p>log(java.lang.Exception exception, java.lang.String msg) Deprecated. As of Java Servlet API 2.1, use log(String message, Throwable throwable) instead.</p> <p><i>This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.</i></p>
void	<p>log(java.lang.String msg) Writes the specified message to a servlet log file, usually an event log.</p>
void	<p>log(java.lang.String message, java.lang.Throwable throwable) Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file.</p>
void	<p>removeAttribute(java.lang.String name) Removes the attribute with the given name from the servlet context.</p>
void	<p>setAttribute(java.lang.String name, java.lang.Object object) Binds an object to a given attribute name in this servlet context.</p>

Objek out

Objek ini merupakan cetakan dari class javax.servlet.jsp.JspWriter. Berikut ini adalah properti-properti yang dimilikinya :

Field Summary	
protected boolean	autoFlush
protected int	bufferSize
static int	<p>DEFAULT_BUFFER constant indicating that the Writer is buffered and is using the implementation default buffer size</p>

static int	NO_BUFFER constant indicating that the Writer is not buffering output
static int	UNBOUNDED_BUFFER constant indicating that the Writer is buffered and is unbounded; this is used in BodyContent

Dan method-method yang dimilikinya adalah sebagai berikut :

Method Summary	
abstract void	clear() Clear the contents of the buffer.
abstract void	clearBuffer() Clears the current contents of the buffer.
abstract void	close() Close the stream, flushing it first
abstract void	flush() Flush the stream.
int	getBufferSize() This method returns the size of the buffer used by the JspWriter.
abstract int	getRemaining() This method returns the number of unused bytes in the buffer.
boolean	isAutoFlush() This method indicates whether the JspWriter is autoFlushing.
abstract void	newLine() Write a line separator.
abstract void	print(boolean b) Print a boolean value.
abstract void	print(char c) Print a character.
abstract void	print(char[] s) Print an array of characters.
abstract void	print(double d) Print a double-precision floating-point number.
abstract void	print(float f) Print a floating-point number.
abstract void	print(int i) Print an integer.
abstract void	print(long l) Print a long integer.
abstract void	print(java.lang.Object obj) Print an object.
abstract void	print(java.lang.String s) Print a string.
abstract void	println()

	Terminate the current line by writing the line separator string.
abstract void	<code>println(boolean x)</code> Print a boolean value and then terminate the line.
abstract void	<code>println(char x)</code> Print a character and then terminate the line.
abstract void	<code>println(char[] x)</code> Print an array of characters and then terminate the line.
abstract void	<code>println(double x)</code> Print a double-precision floating-point number and then terminate the line.
abstract void	<code>println(float x)</code> Print a floating-point number and then terminate the line.
abstract void	<code>println(int x)</code> Print an integer and then terminate the line.
abstract void	<code>println(long x)</code> Print a long integer and then terminate the line.
abstract void	<code>println(java.lang.Object x)</code> Print an Object and then terminate the line.
abstract void	<code>println(java.lang.String x)</code> Print a String and then terminate the line.

Objek config

Objek ini di dibuat dari class `javax.servlet.ServletConfig`. Method-method yang dimilikinya adalah :

Method Summary	
<code>java.lang.String</code>	<code>getInitParameter(java.lang.String name)</code> Returns a <code>String</code> containing the value of the named initialization parameter, or <code>null</code> if the parameter does not exist.
<code>java.util.Enumeration</code>	<code>getInitParameterNames()</code> Returns the names of the servlet's initialization parameters as an <code>Enumeration</code> of <code>String</code> objects, or an empty <code>Enumeration</code> if the servlet has no initialization parameters.
<code>ServletContext</code>	<code>getServletContext()</code> Returns a reference to the ServletContext in which the caller is executing.
<code>java.lang.String</code>	<code>getServletName()</code> Returns the name of this servlet instance.

Objek page

Objek page dibuat dari class `java.lang.Object`.

Objek exception

Objek ini dibuat dari class `java.lang.Throwable`.

Cacatan Penulis :

Penulis adalah pemula dalam mempelajari Java, jadi mohon masukannya bagi pembaca yang menemukan kesalahan konsep atau asumsi yang digunakan penulis.
--