Java.servlet.\*

Compiled from "Servlet.java"[5]

public interface Servlet

--------------------------------------------

public abstract void init( ServletConfig) throws ServletException;

public abstract void service( ServletRequest, ServletResponse) throws ServletException, java.io.IOException;

public abstract void destroy();

public abstract ServletConfig getServletConfig();

public abstract String getServletInfo();

Compiled from "ServletRequest.java"[29]

-------------------------------------------------------

public interfaceServletRequest

{

protocal://Server name:port

-------------------------------------------

public abstract String getProtocol();

public abstract String getServerName();

Port

--------------

public abstract int getServerPort();

public abstract int getRemotePort();

public abstract int getLocalPort();

Attributes

-------------

public abstract Object getAttribute(String);

public abstract void setAttribute(String, Object);

public abstract Enumeration getAttributeNames();

public abstract void removeAttribute(String);

Parameters

--------------------

public abstract String getParameter(String);

public abstract Enumeration getParameterNames();

public abstract String[] getParameterValues(String);

public abstract Map getParameterMap();

Content Type/Encoding

-----------------

public abstract String getContentType();

public abstract int getContentLength();

public abstract String getCharacterEncoding();

public abstract void setCharacterEncoding(String) throws UnsupportedEncodingException;

Remote - Host Address

-----------------------------------

public abstract String getRemoteAddr();

public abstract String getRemoteHost();

public abstract String getLocalName();

public abstract String getLocalAddr();

public abstract Locale getLocale();

public abstract Enumeration getLocales();

public abstract String getScheme();

public abstract String getRealPath(String);

public abstractRequestDispatcher getRequestDispatcher(String);

public abstractServletInputStream getInputStream() throws IOException;

public abstract BufferedReader getReader() throws IOException;

public abstract boolean isSecure();

Compiled from "ServletResponse.java"[15]

public interface javax.servlet.ServletResponse

{

public abstract void setContentType( String);

public abstract String getContentType();

public abstract void setContentLength(int);

public abstract void setCharacterEncoding( String);

public abstract String getCharacterEncoding();

public abstract void reset()

public abstract void setBufferSize(int);

public abstract int getBufferSize();

public abstract void flushBuffer() throws IOException;

public abstract void resetBuffer();

public abstract boolean isCommitted();

public abstract void setLocale( Locale);

public abstract Locale getLocale();

public abstract ServletOutputStream getOutputStream() throws IOException;

public abstract PrintWriter getWriter() throws IOException;

Compiled from "ServletContext.java"[24]

public interface ServletContext

{

Attribute

------------------

public abstract Object getAttribute( String);

public abstract void setAttribute( String, Object);

public abstract void removeAttribute( String);

public abstract Enumeration getAttributeNames();

Parameters

------------------------

public abstract String getInitParameter( String);

public abstract Enumeration getInitParameterNames();

public abstract ServletContext getContext( String);

public abstract String getMimeType( String);

ServerInfo

-----------------------

public abstract Servlet getServlet( String) throws ServletException;

public abstract Enumeration getServlets();

public abstract Enumeration getServletNames();

public abstract String getServerInfo();

public abstract String getServletContextName();

Version

--------------

public abstract int getMajorVersion();

public abstract int getMinorVersion();

public abstract RequestDispatcher getRequestDispatcher( String);

public abstract RequestDispatcher getNamedDispatcher( String);

public abstract void log( String);

public abstract void log( Exception, String);

public abstract void log( String, Throwable);

public abstract Set getResourcePaths( String);

public abstract java.net.URL getResource( String) throws java.net.MalformedURLException;

public abstract InputStream getResourceAsStream( String);

public abstract String getRealPath( String);

Compiled from "ServletConfig.java"

public interface javax.servlet.ServletConfig

{

public abstract java.lang.String getServletName();

public abstract javax.servlet.ServletContext getServletContext();

public abstract java.lang.String getInitParameter(java.lang.String);

public abstract java.util.Enumeration getInitParameterNames();

}

Compiled from "GenericServlet.java"[12]

public abstract class GenericServlet extends Object implements Servlet, ServletConfig,java.io.Serializable

{

public GenericServlet();

public void init( ServletConfig) throws ServletException;

public void init() throws ServletException;

public abstract void service( ServletRequest, ServletResponse) throws ServletException, java.io.IOException;

public void destroy();

public String getServletName();

public String getServletInfo();

public ServletConfig getServletConfig();

public ServletContext getServletContext();

public String getInitParameter( String);

public Enumeration getInitParameterNames();

public void log( String);

public void log( String, Throwable);

}

Compiled from "FilterChain.java"

public interface FilterChain

{

public abstract void doFilter( ServletRequest, ServletResponse) throws java.io.IOException, ServletException;

}

Compiled from "Filter.java"

public interface Filter

{

public abstract void init( FilterConfig) throws ServletException;

public abstract void doFilter( ServletRequest, ServletResponse, FilterChain) throws java.io.IOException, ServletException;

public abstract void destroy();

}

Compiled from "FilterConfig.java"

public interface FilterConfig

{

public abstract String getFilterName();

public abstract ServletContext getServletContext();

public abstract String getInitParameter( String);

public abstract Enumeration getInitParameterNames();

}

Compiled from "ServletRequestListener.java"

public interface ServletRequestListener extends java.util.EventListener

{

public abstract void requestDestroyed( ServletRequestEvent);

public abstract void requestInitialized( ServletRequestEvent);

}

Compiled from "ServletContextListener.java"

public interface ServletContextListener extends java.util.EventListener

{

public abstract void contextInitialized( ServletContextEvent);

public abstract void contextDestroyed( ServletContextEvent);

}

Compiled from "ServletContextAttributeListener.java"

public interface ServletContextAttributeListener extends java.util.EventListener

{

public abstract void attributeAdded( ServletContextAttributeEvent);

public abstract void attributeRemoved( ServletContextAttributeEvent);

public abstract void attributeReplaced( ServletContextAttributeEvent)

}

Compiled from "ServletRequestAttributeListener.java"

public interface ServletRequestAttributeListener extends java.util.EventListener

{

public abstract void attributeAdded( ServletRequestAttributeEvent);

public abstract void attributeRemoved( ServletRequestAttributeEvent);

public abstract void attributeReplaced( ServletRequestAttributeEvent);

}

Compiled from "ServletRequestEvent.java"

public class ServletRequestEvent extends java.util.EventObject

{

public ServletRequestEvent( ServletContext, ServletRequest);

public ServletRequest getServletRequest();

public ServletContext getServletContext();

}

Compiled from "ServletContextEvent.java"

public class ServletContextEvent extends java.util.EventObject

{

public ServletContextEvent( ServletContext);

public ServletContext getServletContext();

}

Compiled from "ServletContextAttributeEvent.java"

public class ServletContextAttributeEvent extends ServletContextEvent

{

public ServletContextAttributeEvent( ServletContext, String, Object);

public String getName();

public Object getValue();

}

Compiled from "ServletRequestAttributeEvent.java"

public class ServletRequestAttributeEvent extends ServletRequestEvent

{

public ServletRequestAttributeEvent( ServletContext, ServletRequest, String, Object);

public String getName();

public Object getValue();

Compiled from "ServletInputStream.java"

public abstract class javax.servlet.ServletInputStream extends InputStream

{

protected javax.servlet.ServletInputStream();

public int readLine(byte[], int, int) throws IOException;

}

Compiled from "ServletOutputStream.java"

public abstract class javax.servlet.ServletOutputStream extends OutputStream

{

protected javax.servlet.ServletOutputStream();

public void print(java.lang.String) throws IOException;

public void print(boolean) throws IOException;

public void print(char) throws IOException;

public void print(int) throws IOException;

public void print(long) throws IOException;

public void print(float) throws IOException;

public void print(double) throws IOException;

public void println() throws IOException;

public void println(java.lang.String) throws IOException;

public void println(boolean) throws IOException;

public void println(char) throws IOException;

public void println(int) throws IOException;

public void println(long) throws IOException;

public void println(float) throws IOException;

public void println(double) throws IOException;

static {};

}

Compiled from "ServletRequestWrapper.java"

public class ServletRequestWrapper extends Object implements ServletRequest

{

public ServletRequestWrapper( ServletRequest);

public ServletRequest getRequest();

------------------------------------------------------------------------------------

All are "ServletRequest" methods

--------------------------------------------------------

public void setRequest( ServletRequest);

public Object getAttribute( String);

public java.util.Enumeration getAttributeNames();

public String getCharacterEncoding();

public void setCharacterEncoding( String) throws UnsupportedEncodingException;

public int getContentLength();

public String getContentType();

public ServletInputStream getInputStream() throws IOException;

public String getParameter( String);

public java.util.Map getParameterMap();

public java.util.Enumeration getParameterNames();

public String[] getParameterValues( String);

public String getProtocol();

public String getScheme();

public String getServerName();

public int getServerPort();

public BufferedReader getReader() throws IOException;

public String getRemoteAddr();

public String getRemoteHost();

public void setAttribute( String, Object);

public void removeAttribute( String);

public java.util.Locale getLocale();

public java.util.Enumeration getLocales();

public boolean isSecure();

public RequestDispatcher getRequestDispatcher( String);

public String getRealPath( String);

public int getRemotePort();

public String getLocalName();

public String getLocalAddr();

public int getLocalPort();

}

Compiled from "ServletResponseWrapper.java"

public class ServletResponseWrapper extends Object implements ServletResponse

{

public ServletResponseWrapper( ServletResponse);

public ServletResponse getResponse();

public void setResponse( ServletResponse);

--------------------------------------------------------------------

All are "ServletResponse" methods

--------------------------------------------------------------------

public void setCharacterEncoding( String);

public String getCharacterEncoding();

public ServletOutputStream getOutputStream() throws java.io.IOException;

public java.io.PrintWriter getWriter() throws java.io.IOException;

public void setContentLength(int);

public void setContentType( String);

public String getContentType();

public void setBufferSize(int);

public int getBufferSize();

public void flushBuffer() throws java.io.IOException;

public boolean isCommitted();

public void reset();

public void resetBuffer();

public void setLocale( Locale);

public Locale getLocale();

}

Compiled from "RequestDispatcher.java"

public interface RequestDispatcher

{

public abstract void forward( ServletRequest, ServletResponse) throws ServletException, IOException;

public abstract void include( ServletRequest, ServletResponse) throws ServletException, IOException;

}

**javax.servlet.http.\***

Compiled from "HttpServlet.java"[10]

public abstract class HttpServlet extends GenericServlet implements Serializable

{

public HttpServlet();

public void service( ServletRequest, ServletResponse) throws ServletException, IOException;

protected void doGet( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doPost( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void service( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doHead( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doPut( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doDelete( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doOptions( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected void doTrace( HttpServletRequest, HttpServletResponse) throws ServletException, IOException;

protected long getLastModified( HttpServletRequest);

Compiled from "HttpServletRequest.java"

public interface HttpServletRequest extends ServletRequest

{

public static final String BASIC\_AUTH;

public static final String FORM\_AUTH;

public static final String CLIENT\_CERT\_AUTH;

public static final String DIGEST\_AUTH;

Header

------------

public abstract String getHeader( String);

public abstract Enumeration getHeaders( String);

public abstract Enumeration getHeaderNames();

public abstract int getIntHeader( String);

public abstract long getDateHeader( String);

Path

-------------

public abstract String getPathInfo();

public abstract String getPathTranslated();

public abstract String getServletPath();

public abstract String getContextPath();

URL

-----------------

public abstract String getRequestURI();

public abstract StringBuffer getRequestURL();

Session

--------------

public abstract HttpSession getSession(boolean);

public abstract HttpSession getSession();

public abstract String getRequestedSessionId();

public abstract boolean isRequestedSessionIdValid();

public abstract boolean isRequestedSessionIdFromCookie();

public abstract boolean isRequestedSessionIdFromURL();

public abstract boolean isRequestedSessionIdFromUrl();

Cookie

-------------

public abstract Cookie[] getCookies();

User

---------

public abstract String getRemoteUser();

public abstract String getAuthType();

public abstract boolean isUserInRole( String);

public abstract java.security.Principal getUserPrincipal();

public abstract String getMethod();

public abstract String getQueryString();

Compiled from "HttpServletResponse.java"

public interface javax.servlet.http.HttpServletResponse extends javax.servlet.ServletResponse

{

Header

------------------

public abstract void setHeader( String, String);

public abstract void addHeader( String, String);

public abstract void setIntHeader( String, int);

public abstract void addIntHeader( String, int);

public abstract void setDateHeader( String, long);

public abstract void addDateHeader( String, long);

public abstract boolean containsHeader( String);

URL

--------

public abstract String encodeURL( String);

public abstract String encodeUrl( String);

public abstract String encodeRedirectURL( String);

public abstract String encodeRedirectUrl( String);

public abstract void sendRedirect( String) throws IOException;

Cookie

-----------

public abstract void addCookie(javax.servlet.http.Cookie);

Error

-----------

public abstract void sendError(int, String) throws IOException;

public abstract void sendError(int) throws IOException;

Status

---------

public abstract void setStatus(int);

public abstract void setStatus(int, String);

}

Compiled from "HttpSession.java"

public interface HttpSession

{

Attribute

----------------

public abstract void setAttribute( String, Object);

public abstract Object getAttribute( String);

public abstract java.util.Enumeration getAttributeNames();

public abstract void removeAttribute( String);

Values

------------

public abstract void putValue( String, Object);

public abstract Object getValue( String);

public abstract String[] getValueNames();

public abstract void removeValue( String);

Time

--------

public abstract long getCreationTime();

public abstract long getLastAccessedTime();

public abstract void setMaxInactiveInterval(int);

public abstract int getMaxInactiveInterval();

public abstract HttpSessionContext getSessionContext();

public abstract ServletContext getServletContext();

public abstract void invalidate();

public abstract boolean isNew();

public abstract String getId();

}

Compiled from "Cookie.java"

public class Cookie extends Object implements Cloneable

{

public Cookie( String, String);

public void setComment( String);

public String getComment();

public void setDomain( String);

public String getDomain();

public void setMaxAge(int);

public int getMaxAge();

public void setPath( String);

public String getPath();

public void setSecure(boolean);

public boolean getSecure();

public String getName();

public void setValue( String);

public String getValue();

public int getVersion();

public void setVersion(int);

public Object clone()

Compiled from "HttpSessionListener.java"

public interface HttpSessionListener extends java.util.EventListener

{

public abstract void sessionCreated( HttpSessionEvent);

public abstract void sessionDestroyed( HttpSessionEvent);

}

Compiled from "HttpSessionAttributeListener.java"

public interface HttpSessionAttributeListener extends java.util.EventListener

{

public abstract void attributeAdded( HttpSessionBindingEvent);

public abstract void attributeRemoved( HttpSessionBindingEvent);

public abstract void attributeReplaced( HttpSessionBindingEvent);

}

Compiled from "HttpSessionActivationListener.java"

public interface HttpSessionActivationListener extends java.util.EventListener

{

public abstract void sessionWillPassivate( HttpSessionEvent);

public abstract void sessionDidActivate( HttpSessionEvent);

}

Compiled from "HttpSessionBindingListener.java"

public interface HttpSessionBindingListener extends java.util.EventListener

{

public abstract void valueBound( HttpSessionBindingEvent);

public abstract void valueUnbound( HttpSessionBindingEvent);

}

Compiled from "HttpSessionContext.java"

public interface javax.servlet.http.HttpSessionContext

{

public abstract HttpSession getSession(String);

public abstract Enumeration getIds();

}

Compiled from "HttpSessionEvent.java"

public class HttpSessionEvent extends java.util.EventObject

{

public HttpSessionEvent( HttpSession);

public HttpSession getSession();

}

Compiled from "HttpSessionBindingEvent.java"

public class HttpSessionBindingEvent extends HttpSessionEvent

{

public HttpSessionBindingEvent( HttpSession, String);

public HttpSessionBindingEvent( HttpSession, String, Object);

public HttpSession getSession();

public String getName();

public Object getValue();

Compiled from "HttpServletRequestWrapper.java"

public class HttpServletRequestWrapper extends ServletRequestWrapper implements HttpServletRequest

{

public HttpServletRequestWrapper( HttpServletRequest);

------------------------------------------------------------------------------------

All "HttpServletRequest"methods

------------------------------------------------------------------------------------

public String getAuthType();

public Cookie[] getCookies();

public long getDateHeader( String);

public String getHeader( String);

public Enumeration getHeaders( String);

public Enumeration getHeaderNames();

public int getIntHeader( String);

public String getMethod();

public String getPathInfo();

public String getPathTranslated();

public String getContextPath();

public String getQueryString();

public String getRemoteUser();

public boolean isUserInRole( String);

public java.security.Principal getUserPrincipal();

public String getRequestedSessionId();

public String getRequestURI();

public StringBuffer getRequestURL();

public String getServletPath();

public HttpSession getSession(boolean);

public HttpSession getSession();

public boolean isRequestedSessionIdValid();

public boolean isRequestedSessionIdFromCookie();

public boolean isRequestedSessionIdFromURL();

public boolean isRequestedSessionIdFromUrl();

}

Compiled from "HttpServletResponseWrapper.java"

public class HttpServletResponseWrapper extends javax.servlet.ServletResponseWrapper implements HttpServletResponse

{

public HttpServletResponseWrapper( HttpServletResponse);

-------------------------------------------------------------------

All are "HttpServletResponse" methods

-------------------------------------------------------------------

public void addCookie( Cookie);

public boolean containsHeader(String);

public String encodeURL(String);

public String encodeRedirectURL(String);

public String encodeUrl(String);

public String encodeRedirectUrl(String);

public void sendError(int, String) throws IOException;

public void sendError(int) throws IOException;

public void sendRedirect(String) throws IOException;

public void setDateHeader(String, long);

public void addDateHeader(String, long);

public void setHeader(String, String);

public void addHeader(String, String);

public void setIntHeader(String, int);

public void addIntHeader(String, int);

public void setStatus(int);

public void setStatus(int, String);

}

Compiled from "HttpUtils.java"

public class HttpUtils extends java.lang.Object

{

public HttpUtils();

public static java.util.Hashtable parseQueryString(java.lang.String);

public static java.util.Hashtable parsePostData(int, javax.servlet.ServletInputStream);

public static java.lang.StringBuffer getRequestURL( HttpServletRequest);

}

public static final int SC\_CONTINUE;

public static final int SC\_SWITCHING\_PROTOCOLS;

public static final int SC\_OK;

public static final int SC\_CREATED;

public static final int SC\_ACCEPTED;

public static final int SC\_NON\_AUTHORITATIVE\_INFORMATION;

public static final int SC\_NO\_CONTENT;

public static final int SC\_RESET\_CONTENT;

public static final int SC\_PARTIAL\_CONTENT;

public static final int SC\_MULTIPLE\_CHOICES;

public static final int SC\_MOVED\_PERMANENTLY;

public static final int SC\_MOVED\_TEMPORARILY;

public static final int SC\_FOUND;

public static final int SC\_SEE\_OTHER;

public static final int SC\_NOT\_MODIFIED;

public static final int SC\_USE\_PROXY;

public static final int SC\_TEMPORARY\_REDIRECT;

public static final int SC\_BAD\_REQUEST;

public static final int SC\_UNAUTHORIZED;

public static final int SC\_PAYMENT\_REQUIRED;

public static final int SC\_FORBIDDEN;

public static final int SC\_NOT\_FOUND;

public static final int SC\_METHOD\_NOT\_ALLOWED;

public static final int SC\_NOT\_ACCEPTABLE;

public static final int SC\_PROXY\_AUTHENTICATION\_REQUIRED;

public static final int SC\_REQUEST\_TIMEOUT;

public static final int SC\_CONFLICT;

public static final int SC\_GONE;

public static final int SC\_LENGTH\_REQUIRED;

public static final int SC\_PRECONDITION\_FAILED;

public static final int SC\_REQUEST\_ENTITY\_TOO\_LARGE;

public static final int SC\_REQUEST\_URI\_TOO\_LONG;

public static final int SC\_UNSUPPORTED\_MEDIA\_TYPE;

public static final int SC\_REQUESTED\_RANGE\_NOT\_SATISFIABLE;

public static final int SC\_EXPECTATION\_FAILED;

public static final int SC\_INTERNAL\_SERVER\_ERROR;

public static final int SC\_NOT\_IMPLEMENTED;

public static final int SC\_BAD\_GATEWAY;

public static final int SC\_SERVICE\_UNAVAILABLE;

public static final int SC\_GATEWAY\_TIMEOUT;

public static final int SC\_HTTP\_VERSION\_NOT\_SUPPORTED;

Javax.servlet.\*

**Interfaces**

Compiled from "Servlet.java"

public interface Servlet

{

public abstract void init( ServletConfig) throws ServletException;

public abstract void service( ServletRequest, ServletResponse) throws ServletException,IOException;

public abstract void destroy();

public abstract ServletConfig getServletConfig();

public abstract getServletInfo();

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Servlet : Defines methods that all servlets must implement

A servlet is a small Java program that runs within a Web server.

Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.

To implement this interface, you can write a generic servlet that extends javax.servlet.GenericServlet or javax.servlet.http.HttpServlet.

This interface defines methods to initialize a servlet, to service requests, and to remove a servlet from the server.

These are known as life-cycle methods and are called in the following sequence:

The servlet is constructed, then initialized with the init() method.

Any calls from clients to the service() method are handled.

The servlet is taken out of service, then destroyed() with the destroy method, then garbage collected and finalized.

In addition to the life-cycle methods, this interface provides the

getServletConfig : which the servlet can use to get any startup information,

getServletInfo : which allows the servlet to return basic information about itself, such as author, version, and copyright.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

init():

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Called by the servlet container servlet is being placed into service.

The servlet container calls the init method exactly once after instantiating the servlet.

The init method must complete successfully before the servlet can receive any requests.

The servlet container cannot place the servlet into service if the init method Throws a ServletException

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void service(ServletRequest req, ServletResponse res) throws ServletException, java.io.IOException

Called by the servlet container to allow the servlet to respond to a request.

This method is only called after the servlet's init() method has completed successfully.

Parameters:

req - the ServletRequest object that contains the client's request

res - the ServletResponse object that contains the servlet's response

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void destroy()

Called by the servlet container servlet is being taken out of service.

This method is only called once all threads within the servlet's service method have exited or after a timeout period has passed.

After the servlet container calls this method, it will not call the service method again on this servlet.

This method gives the servlet an opportunity to clean up any resources that are being held (for example, memory, file handles, threads)

and make sure that any persistent state is synchronized with the servlet's current state in memory.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public ServletConfig getServletConfig()

Returns a ServletConfig object, which contains initialization and startup parameters for this servlet.

The ServletConfig object returned is the one passed to the init method.

Returns:

the ServletConfig object that initializes this servlet

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public java.lang.String getServletInfo()

Returns information about the servlet, such as author, version, and copyright.

The string that this method returns should be plain text and not markup of any kind (such as HTML, XML, etc.).

Returns:

a String containing servlet information

**javax.servlet .Filter[i]**

A filter is an object that performs filtering tasks on either the request to a resource (a servlet or static content),

or on the response from a resource, or both.

Filters are configured in the deployment descriptor of a web application

Examples that have been identified for this design are

1) Authentication Filters

2) Logging and Auditing Filters

3) Image conversion Filters

4) Data compression Filters

5) Encryption Filters

6) Tokenizing Filters

7) Filters that trigger resource access events

8) XSL/T filters

9) Mime-type chain Filter

Since:

Servlet 2.3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void init(FilterConfig) throws ServletException

Called by the web container to indicate to a filter that it is being placed into service.

The servlet container calls the init method exactly once after instantiating the filter.

The init method must complete successfully before the filter is asked to do any filtering work.

The web container cannot place the filter into service if the init method either

1.Throws a ServletException

2.Does not return within a time period defined by the web container

Throws:

ServletException

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void doFilter (ServletRequest ,ServletResponse , FilterChain ) throws IOException, ServletException

The doFilter method of the Filter is called by the container each time a request/response pair is passed

through the chain due to a client request for a resource at the end of the chain.

The FilterChain passed in to this method allows the Filter to pass on the request and response to the next entity in the chain.

A typical implementation of this method would follow the following pattern:-

1. Examine the request

2. Optionally wrap the request object with a custom implementation to filter content or headers for input filtering

3. Optionally wrap the response object with a custom implementation to filter content or headers for output filtering

4. a) Either invoke the next entity in the chain using the FilterChain object (chain.doFilter()),

4. b) or not pass on the request/response pair to the next entity in the filter chain to block the request processing

5. Directly set headers on the response after invocation of the next entity in the filter chain.

Throws:

java.io.IOException

ServletException

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void destroy()

Called by the web container to indicate to a filter that it is being taken out of service.

This method is only called once all threads within the filter's doFilter method have exited or after a timeout period has passed.

After the web container calls this method, it will not call the doFilter method again on this instance of the filter.

This method gives the filter an opportunity to clean up any resources that are being held (for example, memory, file handles, threads)

and make sure that any persistent state is synchronized with the filter's current state in memory.

**javax.servlet .FilterChain[i]**

A FilterChain is an object provided by the servlet container to the developer giving a view into the invocation chain

of a filtered request for a resource.

Filters use the FilterChain to invoke the next filter in the chain, or last filter in the chain, to invoke the resource at the end of the chain.

Since:

Servlet 2.3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

doFilter

public void doFilter(ServletRequest, ServletResponse response) throws java.io.IOException, ServletException

Filters use the FilterChain to invoke the next filter in the chain, or last filter in the chain, to invoke the resource at the end of the chain.

Parameters:

request - the request to pass along the chain.

response - the response to pass along the chain.

Throws:

java.io.IOException

ServletException

Since:

2.3

javax.servlet . FilterConfig[i]

A filter configuration object used by a servlet container to pass information to a filter during initialization.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getFilterName()

Returns the filter-name as defined in the deployment descriptor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public ServletContext getServletContext()

Returns a reference to the ServletContext in which the caller is executing.

Returns:

a ServletContext object, used by the caller to interact with its servlet container

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getInitParameter( String name)

Returns a String containing the value of the initialization parameter, or null if the parameter does not exist.

Parameters:

name - a String specifying the name of the initialization parameter

Returns:

a String containing the value of the initialization parameter

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public java.util.Enumeration getInitParameterNames()

Returns the names of the filter's initialization parameters as an Enumeration of String objects,

or an empty Enumeration if the filter has no initialization parameters.

Returns:

an Enumeration of String objects containing the names of the filter's initialization parameters

**javax.servlet .RequestDispatcher[i]**

Defines an object that receives requests from the client and sends them to any resource (such as a servlet, HTML file, or JSP file) on the server.

The servlet container creates the RequestDispatcher object, which is used as a Cover around a server resource

located at a particular path or given by a particular name.

This interface is intended to Cover servlets, but a servlet container can create RequestDispatcher objects to Cover any type of resource.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void forward(ServletRequest , ServletResponse ) throws ServletException, java.io.IOException

Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.

This method allows one servlet to do preliminary processing of a request and another resource to generate the response.

For a RequestDispatcher obtained via getRequestDispatcher(), the ServletRequest object has its path elements and parameters

adjusted to match the path of the target resource.

forward should be called before the response has been committed to the client (before response body output has been flushed).

If the response already has been committed, this method throws an IllegalStateException.

Uncommitted output in the response buffer is automatically cleared before the forward.

Parameters:

request - a ServletRequest object that represents the request the client makes of the servlet

response - a ServletResponse object that represents the response the servlet returns to the client

Throws:

ServletException - if the target resource throws this exception

java.io.IOException - if the target resource throws this exception

java.lang.IllegalStateException - if the response was already committed

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void include(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException

Includes the content of a resource (servlet, JSP page, HTML file) in the response.

In essence, this method enables programmatic server-side includes.

The ServletResponse object has its path elements and parameters remain unchanged from the caller's.

The included servlet cannot change the response status code or set headers; any attempt to make a change is ignored.

The request and response parameters must be either the same objects as were passed to the calling

servlet's service method or be subclasses of the ServletRequestCover or ServletResponseCover classes that Cover them.

Parameters:

request - a ServletRequest object that contains the client's request

response - a ServletResponse object that contains the servlet's response

Throws:

ServletException - if the included resource throws this exception

java.io.IOException - if the included resource throws this exception

**javax.servlet .ServletConfig[i]**

|

Implementing

|

GenericServlet

A servlet configuration object used by a servlet container to pass information to a servlet during initialization.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getInitParameter(String name)

Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getInitParameterNames()

Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration

if the servlet has no initialization parameters.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletContext getServletContext()

Returns a reference to the ServletContext in which the caller is executing.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServletName()

Returns the name of this servlet instance.

**javax.servlet .ServletContext[i]**

Defines a set of methods that a servlet uses to communicate with its servlet container.

for example, to get the MIME type of a file, dispatch requests, or write to a log file.

There is one context per "web application" per Java Virtual Machine.

(A "web application" is a collection of servlets and content installed under a specific subset of the server's URL namespace

such as /catalog and possibly installed via a .war file.)

In the case of a web application marked "distributed" in its deployment descriptor,

there will be one context instance for each virtual machine.

In this situation, the context cannot be used as a location to share global information (because the information won't be truly global).

Use an external resource like a database instead.

The ServletContext object is contained within the ServletConfig object, which the Web server provides the servlet when the servlet is initialized.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getAttribute( String name)

Returns the servlet container attribute with the given name, or null if there is no attribute by that name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getAttributeNames()

Returns an Enumeration containing the attribute names available within this servlet context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletContext getContext( String uripath)

Returns a ServletContext object that corresponds to a specified URL on the server.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getInitParameter( String name)

Returns a String containing the value of the named context-wide initialization parameter, or null if the parameter does not exist.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getInitParameterNames()

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 getMajorVersion()

Returns the major version of the Java Servlet API that this servlet container supports.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getMimeType( String file)

Returns the MIME type of the specified file, or null if the MIME type is not known.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getMinorVersion()

Returns the minor version of the Servlet API that this servlet container supports.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServerInfo()

Returns the name and version of the servlet container on which the servlet is running.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Servlet getServlet( String name)

Deprecated. As of Java Servlet API 2.1, with no direct replacement.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void log( Exception exception, String msg)

Deprecated. As of Java Servlet API 2.1, use log(String message, Throwable throwable) instead.

This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void log( String msg)

Writes the specified message to a servlet log file, usually an event log.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void log( String message, Throwable throwable)

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void removeAttribute( String name)

Removes the attribute with the given name from the servlet context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setAttribute( String name, Object object)

Binds an object to a given attribute name in this servlet context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Set getResourcePaths( 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.

Paths indicating subdirectory paths end with a '/'.

The returned paths are all relative to the root of the web application and have a leading '/'.

For example, for a web application containing

/welcome.html

/catalog/index.html

/catalog/products.html

/catalog/offers/books.html

/catalog/offers/music.html

/customer/login.jsp

/WEB-INF/web.xml

/WEB-INF/classes/com.acme.OrderServlet.class,

getResourcePaths("/")

returns

{

"/welcome.html",

"/catalog/",

"/customer/",

"/WEB-INF/"

}

getResourcePaths("/catalog/")

returns

{

"/catalog/index.html",

"/catalog/products.html",

"/catalog/offers/"

}.

Parameters:

path - the partial path used to match the resources, which must start with a /

Returns:

a Set containing the directory listing, or null if there are no resources in the web application whose path begins with the supplied path.

Since:

Servlet 2.3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getResource

public URL getResource( String path)

Returns a URL to the resource that is mapped to a specified path.

The path must begin with a "/" and is interpreted as relative to the current context root.

This method allows the servlet container to make a resource available to servlets from any source.

Resources can be located on a local or remote file system, in a database, or in a .war file.

The servlet container must implement the URL handlers and URLConnection objects that are necessary to access the resource.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public InputStream getResourceAsStream( String path)

Returns the resource located at the named path as an InputStream object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public RequestDispatcher getRequestDispatcher( String path)

Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.

A RequestDispatcher object can be used to forward a request to the resource or to include the resource in a response.

The resource can be dynamic or static.

The pathname must begin with a "/" and is interpreted as relative to the current context root.

Use getContext to obtain a RequestDispatcher for resources in foreign contexts.

This method returns null if the ServletContext cannot return a RequestDispatcher.

Parameters:

path - a String specifying the pathname to the resource

Returns:

a RequestDispatcher object that acts as a wrapper for the resource at the specified path, or null if the ServletContext cannot return a RequestDispatcher

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public RequestDispatcher getNamedDispatcher( String name)

Returns a RequestDispatcher object that acts as a wrapper for the named servlet.

Servlets (and JSP pages also) may be given names via server administration or via a web application deployment descriptor. A servlet instance can determine its name using ServletConfig.getServletName().

This method returns null if the ServletContext cannot return a RequestDispatcher for any reason.

Parameters:

name - a String specifying the name of a servlet to wrap

Returns:

a RequestDispatcher object that acts as a wrapper for the named servlet, or null if the ServletContext cannot return a RequestDispatcher

**javax.servlet .ServletContextAttributeListener**

|

Superinterfaces

|

java.util.EventListener

public interface ServletContextAttributeListener

extends

|

EventListener

Implementations of this interface receive notifications of changes to the attribute list on the servlet context of a web application.

To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeAdded(ServletContextAttributeEvent scab)

Notification that a new attribute was added to the servlet context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeRemoved(ServletContextAttributeEvent scab)

Notification that an existing attribute has been removed from the servlet context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeReplaced(ServletContextAttributeEvent scab)

Notification that an attribute on the servlet context has been replaced.

**Interface ServletContextListener[i]**

|

All Superinterfaces

|

EventListener

public interface ServletContextListener

|

extends

|

java.util.EventListener

Implementations of this interface receive notifications about changes to the servlet context of the web application they are part of. To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void contextDestroyed(ServletContextEvent sce)

Notification that the servlet context is about to be shut down.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void contextInitialized(ServletContextEvent sce)

Notification that the web application initialization process is starting.

**javax.servlet .ServletRequest[i]**

|

Subinterfaces

|

HttpServletRequest

|

Implementing Classes:

|

HttpServletRequestWrapper, ServletRequestWrapper

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletRequest

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Defines an object to provide client request information to a servlet.

The servlet container creates a ServletRequest object and passes it as an argument to the servlet's service method.

A ServletRequest object provides data including parameter name and values, attributes, and an input stream.

Interfaces that extend ServletRequest can provide additional protocol-specific data (for example, HTTP data is provided by HttpServletRequest.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getAttribute( String name)

Returns the value of the named attribute as an Object, or null if no attribute of the given name exists.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getAttributeNames()

Returns an Enumeration containing the names of the attributes available to this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getCharacterEncoding()

Returns the name of the character encoding used in the body of this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getContentLength()

Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getContentType()

Returns the MIME type of the body of the request, or null if the type is not known.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletInputStream getInputStream()

Retrieves the body of the request as binary data using a ServletInputStream.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getLocalAddr()

Returns the Internet Protocol (IP) address of the interface on which the request was received.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Locale getLocale()

Returns the preferred Locale that the client will accept content in, based on the Accept-Language header.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getLocales()

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.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getLocalName()

Returns the host name of the Internet Protocol (IP) interface on which the request was received.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getLocalPort()

Returns the Internet Protocol (IP) port number of the interface on which the request was received.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getParameter( String name)

Returns the value of a request parameter as a String, or null if the parameter does not exist.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getParameterNames()

Returns an Enumeration of String objects containing the names of the parameters contained in this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String[] getParameterValues( String name)

Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getProtocol()

Returns the name and version of the protocol the request uses in the form protocol/majorVersion.minorVersion,

for example, HTTP/1.1.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getScheme()

Returns the name of the scheme used to make this request, for example, http, https, or ftp.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServerName()

Returns the host name of the server to which the request was sent.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getServerPort()

Returns the port number to which the request was sent.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isSecure()

Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void removeAttribute( String name)

Removes an attribute from this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setAttribute( String name, Object o)

Stores an attribute in this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setCharacterEncoding( String env)

Overrides the name of the character encoding used in the body of this request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getRemoteAddr

public String getRemoteAddr()

Returns the Internet Protocol (IP) address of the client or last proxy that sent the request.

For HTTP servlets, same as the value of the CGI variable REMOTE\_ADDR.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getRemoteHost()

Returns the fully qualified name of the client or the last proxy that sent the request. If the engine cannot or chooses not to resolve the hostname (to improve performance), this method returns the dotted-string form of the IP address. For HTTP servlets, same as the value of the CGI variable REMOTE\_HOST.

Returns:

a String containing the fully qualified name of the client

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public RequestDispatcher getRequestDispatcher( String path)

Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.

A RequestDispatcher object can be used to forward a request to the resource or to include the resource in a response.

The resource can be dynamic or static.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getRemotePort

public int getRemotePort()

Returns the Internet Protocol (IP) source port of the client or last proxy that sent the request.

**javax.servlet .ServletRequestAttributeListener**

|

Superinterfaces

|

EventListener

A ServletRequestAttributeListener can be implemented by the developer interested in being notified of request attribute changes.

Notifications will be generated while the request is within the scope of the web application in which the listener is registered.

A request is defined as coming into scope when it is about to enter the first servlet or filter in each web application,

as going out of scope when it exits the last servlet or the first filter in the chain.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeAdded(ServletRequestAttributeEvent srae)

Notification that a new attribute was added to the servlet request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeRemoved(ServletRequestAttributeEvent srae)

Notification that an existing attribute has been removed from the servlet request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeReplaced(ServletRequestAttributeEvent srae)

Notification that an attribute was replaced on the servlet request.

**javax.servlet .ServletRequestListener**

|

Superinterfaces

|

java.util.EventListener

A ServletRequestListener can be implemented by the developer interested in being notified of requests coming in

and out of scope in a web component. A request is defined as coming into scope when it is about to enter the first servlet

or filter in each web application, as going out of scope when it exits the last servlet or the first filter in the chain.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void requestDestroyed(ServletRequestEvent sre)

The request is about to go out of scope of the web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void requestInitialized(ServletRequestEvent sre)

The request is about to come into scope of the web application.

**javax.servlet .ServletResponse**

|

Subinterfaces

|

HttpServletResponse

|

Implementing Classes

|

HttpServletResponseWrapper

ServletResponseWrapper

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletResponse

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The servlet container creates a ServletResponse object and passes it as an argument to the servlet's service method.

To send binary data in a MIME body response, use the ServletOutputStream returned by getOutputStream().

To send character data, use the PrintWriter object returned by getWriter().

To mix binary and text data, for example, to create a multipart response, use a ServletOutputStream and

manage the character sections manually.

The charset for the MIME body response can be specified explicitly using the setCharacterEncoding( String) and setContentType( String)

or implicitly using the setLocale( Locale) method. If no charset is specified, ISO-8859-1 will be used.

The setCharacterEncoding, setContentType, or setLocale method must be called before getWriter and

before committing the response for the character encoding to be used.

See the Internet RFCs such as RFC 2045 for more information on MIME.

Protocols such as SMTP and HTTP define profiles of MIME, and those standards are still evolving.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void flushBuffer()

Forces any content in the buffer to be written to the client.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getCharacterEncoding()

Returns the name of the character encoding (MIME charset) used for the body sent in this response.

The character encoding may have been specified explicitly using the setCharacterEncoding( String) or

setContentType( String) methods, or implicitly using the setLocale( Locale) method.

Returns:

a String specifying the name of the character encoding, for example, UTF-8

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getContentType()

Returns the content type used for the MIME body sent in this response.

The content type proper must have been specified using setContentType( String) before the response is committed.

Returns:

a String specifying the content type, for example, text/html; charset=UTF-8, or null

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public ServletOutputStream getOutputStream() throws IOException

Returns a ServletOutputStream suitable for writing binary data in the response.

Calling flush() on the ServletOutputStream commits the response.

Either this method or getWriter() may be called to write the body, not both.

Throws:

IllegalStateException - if the getWriter method has been called on this response

IOException - if an input or output exception occurred

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public PrintWriter getWriter() throws IOException

Returns a PrintWriter object that can send character text to the client.

The PrintWriter uses the character encoding returned by getCharacterEncoding().

If the response's character encoding has not been specified as described in

getCharacterEncoding (use default value ISO-8859-1), getWriter updates it to ISO-8859-1.

Either this method or getOutputStream() may be called to write the body, not both.

Returns:

a PrintWriter object that can return character data to the client

Throws:

UnsupportedEncodingException - if the character encoding returned by getCharacterEncoding cannot be used

IllegalStateException - if the getOutputStream method has already been called for this response object

IOException - if an input or output exception occurred

setCharacterEncoding

public void setCharacterEncoding( String charset)

Sets the character encoding (MIME charset) of the response being sent to the client,

for example, to UTF-8. If the character encoding has already been set by setContentType( String) or setLocale( Locale),

this method overrides it. Calling setContentType( String) with the String of text/html and calling this method with the String of UTF-8 is

equivalent with calling setContentType with the String of text/html; charset=UTF-8.

Parameters:

charset - a String specifying only the character set defined by IANA Character Sets (http://www.iana.org/assignments/character-sets)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public 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.

Parameters:

len - an integer specifying the length of the content being returned to the client; sets the Content-Length header

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setContentType( String type)

Sets the content type of the response being sent to the client, if the response has not been committed yet.

The given content type may include a character encoding specification, for example, text/html;charset=UTF-8.

Parameters:

type - a String specifying the MIME type of the content

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setBufferSize(int size)

Sets the preferred buffer size for the body of the response. The servlet container will use a buffer at least as large as the size requested.

The actual buffer size used can be found using getBufferSize.

A larger buffer allows more content to be written before anything is actually sent, thus providing the servlet with more time to set appropriate

status codes and headers.

A smaller buffer decreases server memory load and allows the client to start receiving data more quickly.

Parameters:

size - the preferred buffer size

Throws:

IllegalStateException - if this method is called after content has been written

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public int getBufferSize()

Returns the actual buffer size used for the response. If no buffering is used, this method returns 0.

Returns:

the actual buffer size used

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void flushBuffer() throws IOException

Forces any content in the buffer to be written to the client.

A call to this method automatically commits the response, meaning the status code and headers will be written.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void resetBuffer()

Clears the content of the underlying buffer in the response without clearing headers or status code.

If the response has been committed, this method throws an IllegalStateException.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isCommitted()

Returns a boolean indicating if the response has been committed.

A committed response has already had its status code and headers written.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void reset()

Clears any data that exists in the buffer as well as the status code and headers.

If the response has been committed, this method throws an IllegalStateException.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setLocale( Locale loc)

Sets the locale of the response, if the response has not been committed yet.

It also sets the response's character encoding appropriately for the locale,

if the character encoding has not been explicitly set using setContentType( String) or setCharacterEncoding( String),

getWriter hasn't been called yet, and the response hasn't been committed yet.

If the deployment descriptor contains a locale-encoding-mapping-list element, and that element provides a

mapping for the given locale, that mapping is used. Otherwise, the mapping from locale to character encoding is container dependent.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Locale getLocale()

Returns the locale specified for this response using the setLocale( Locale) method.

Calls made to setLocale after the response is committed have no effect.

If no locale has been specified, the container's default locale is returned.

javax.servlet.SingleThreadModel

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public interface SingleThreadModel

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ensures that servlets handle only one request at a time. This interface has no methods.

If a servlet implements this interface, you are guaranteed that no two threads will execute concurrently in the servlet's service method.

The servlet container can make this guarantee by synchronizing access to a single instance of the servlet,

or by maintaining a pool of servlet instances and dispatching each new request to a free servlet.

Note that SingleThreadModel does not solve all thread safety issues.

For example, session attributes and static variables can still be accessed by multiple requests on multiple threads at the same time,

even when SingleThreadModel servlets are used.

It is recommended that a developer take other means to resolve those issues instead of implementing this interface,

such as avoiding the usage of an instance variable or synchronizing the block of the code accessing those resources.

**Classes**

**javax.servlet.GenericServlet[c]**

javax.servlet.GenericServlet

|

Implemented Interfaces:

|

Serializable, Servlet, ServletConfig

|

Subclasses

|

HttpServlet

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public abstract class GenericServlet extends Objectimplements Servlet, ServletConfig, Serializable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

-Defines a generic, protocol-independent servlet.

-To write an HTTP servlet use HttpServlet

GenericServlet implements the Servlet and ServletConfig interfaces.

GenericServlet may be directly extended by a servlet, although it's more common to extend a protocol-specific subclass such as HttpServlet.

GenericServlet makes writing servlets easier. It provides simple versions of the lifecycle methods init and destroy and of the methods

in the ServletConfig interface. GenericServlet also implements the log method, declared in the ServletContext interface.

To write a generic servlet, you need only override the abstract service method.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

\_\_\_\_\_\_\_\_\_

GenericServlet()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

\_\_\_\_\_\_\_\_\_\_\_

public void destroy()

Called by the servlet container to indicate to a servlet that the servlet is being taken out of service. See Servlet.destroy().

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getInitParameter( String name)

Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist.

See ServletConfig.getInitParameter( String).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Enumeration getInitParameterNames()

Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration

if the servlet has no initialization parameters. See ServletConfig.getInitParameterNames().

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getServletConfig

public ServletConfig getServletConfig()

Returns this servlet's ServletConfig object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletContext getServletContext()

Returns a reference to the ServletContext in which this servlet is running.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServletInfo()

Returns information about the servlet, such as author, version, and copyright.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServletName()

Returns the name of this servlet instance.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void init()

A convenience method which can be overridden so that there's no need to call super.init(config).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void init(ServletConfig config)

Called by the servlet container to indicate to a servlet that the servlet is being placed into service.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void log( String msg)

Writes the specified message to a servlet log file, prepended by the servlet's name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void log( String message, Throwable t)

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file, prepended by the servlet's name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

abstract void service(ServletRequest req, ServletResponse res)

Called by the servlet container to allow the servlet to respond to a request.

**javax.servlet .ServletContextAttributeEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is the event class for notifications about changes to the attributes of the servlet context of a web application.

Field

---------

Fields inherited from class java.util.EventObject

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

------------------

ServletContextAttributeEvent(ServletContext source, String name, Object value)

Construct a ServletContextAttributeEvent from the given context for the given attribute name and attribute value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-----------------

String getName()

Return the name of the attribute that changed on the ServletContext.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getValue()

Returns the value of the attribute that has been added, removed, or replaced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods inherited from class javax.servlet.ServletContextEvent :getServletContext

Methods inherited from class java.util.EventObject :getSource, toString

Methods inherited from class Object :clone, equals, finalize, getClass, hashCode, notify, notifyAll,

wait, wait, wait

**javax.servlet .ServletContextEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is the event class for notifications about changes to the servlet context of a web application.

Field

-------

Fields inherited from class java.util.EventObject

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

-------------

ServletContextEvent(ServletContext source)

Construct a ServletContextEvent from the given context.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

---------------

ServletContext getServletContext()

Return the ServletContext that changed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods inherited from class java.util.EventObject :getSource, toString

Methods inherited from class java.lang.Object :clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet .ServletInputStream**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides an input stream for reading binary data from a client request, including an efficient readLine method

for reading data one line at a time. With some protocols, such as HTTP POST and PUT, a ServletInputStream

object can be used to read data sent from the client.

A ServletInputStream object is normally retrieved via the ServletRequest.getInputStream() method.

This is an abstract class that a servlet container implements.

Subclasses of this class must implement the java.io.InputStream.read() method.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

--------------

protected ServletInputStream()

Does nothing, because this is an abstract class.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-------------------

int readLine(byte[] b, int off, int len)

Reads the input stream, one line at a time.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods inherited from class java.io.InputStream :available, close, mark, markSupported, read, read, read, reset, skip

Methods inherited from class java.lang.Object :clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString,

wait, wait, wait

**javax.servlet .ServletOutputStream**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides an output stream for sending binary data to the client.

A ServletOutputStream object is normally retrieved via the ServletResponse.getOutputStream() method.

This is an abstract class that the servlet container implements.

Subclasses of this class must implement the java.io.OutputStream.write(int) method.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

--------------

protected ServletOutputStream()

Does nothing, because this is an abstract class.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-------------

void print(boolean b)

Writes a boolean value to the client, with no carriage return-line feed (CRLF) character at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(char c)

Writes a character to the client, with no carriage return-line feed (CRLF) at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(double d)

Writes a double value to the client, with no carriage return-line feed (CRLF) at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(float f)

Writes a float value to the client, with no carriage return-line feed (CRLF) at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(int i)

Writes an int to the client, with no carriage return-line feed (CRLF) at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(long l)

Writes a long value to the client, with no carriage return-line feed (CRLF) at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void print(java.lang.String s)

Writes a String to the client, without a carriage return-line feed (CRLF) character at the end.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println()

Writes a carriage return-line feed (CRLF) to the client.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(boolean b)

Writes a boolean value to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(char c)

Writes a character to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(double d)

Writes a double value to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(float f)

Writes a float value to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(int i)

Writes an int to the client, followed by a carriage return-line feed (CRLF) character.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(long l)

Writes a long value to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void println(java.lang.String s)

Writes a String to the client, followed by a carriage return-line feed (CRLF).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods inherited from class java.io.OutputStream :close, flush, write, write, write

Methods inherited from class java.lang.Object :clone, equals, finalize, getClass, hashCode, notify, notifyAll,

toString, wait, wait, wait

**javax.servlet .ServletRequestAttributeEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is the event class for notifications of changes to the attributes of the servlet request in an application.

Constructor

--------------

ServletRequestAttributeEvent(ServletContext sc, ServletRequest request, String name, Object value)

Construct a ServletRequestAttributeEvent giving the servlet context of this web application,

the ServletRequest whose attributes are changing and the name and value of the attribute.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

------------------

String getName()

Return the name of the attribute that changed on the ServletRequest.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getValue()

Returns the value of the attribute that has been added, removed or replaced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods inherited from class javax.servlet.ServletRequestEvent :getServletContext, getServletRequest

Methods inherited from class java.util.EventObject :getSource, toString

Methods inherited from class Object :clone, equals, finalize, getClass, hashCode,

notify, notifyAll, wait, wait, wait

**javax.servlet .ServletRequestEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Subclasses:

|

ServletRequestAttributeEvent

public class ServletRequestEvent

Events of this kind indicate lifecycle events for a ServletRequest.

The source of the event is the ServletContext of this web application.

Constructor

------------------

ServletRequestEvent(ServletContext sc, ServletRequest request)

Construct a ServletRequestEvent for the given ServletContext and ServletRequest.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

----------

ServletContext getServletContext()

Returns the ServletContext of this web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ServletRequest getServletRequest()

Returns the ServletRequest that is changing.

**javax.servlet . ServletRequestWrapper**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides a convenient implementation of the ServletRequest interface that can be subclassed

by developers wishing to adapt the request to a Servlet.

This class implements the Wrapper or Decorator pattern.

Methods default to calling through to the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

ServletRequestWrapper(ServletRequest request)

Creates a ServletRequest adaptor wrapping the given request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

--------------

Object getAttribute( String name)

The default behavior of this method is to call getAttribute(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getAttributeNames()

The default behavior of this method is to return getAttributeNames() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getCharacterEncoding()

The default behavior of this method is to return getCharacterEncoding() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getContentLength()

The default behavior of this method is to return getContentLength() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getContentType()

The default behavior of this method is to return getContentType() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletInputStream getInputStream()

The default behavior of this method is to return getInputStream() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getLocalAddr()

The default behavior of this method is to return getLocalAddr() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Locale getLocale()

The default behavior of this method is to return getLocale() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getLocales()

The default behavior of this method is to return getLocales() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getLocalName()

The default behavior of this method is to return getLocalName() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getLocalPort()

The default behavior of this method is to return getLocalPort() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getParameter( String name)

The default behavior of this method is to return getParameter(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Map getParameterMap()

The default behavior of this method is to return getParameterMap() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getParameterNames()

The default behavior of this method is to return getParameterNames() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String[] getParameterValues( String name)

The default behavior of this method is to return getParameterValues(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getProtocol()

The default behavior of this method is to return getProtocol() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

BufferedReader getReader()

The default behavior of this method is to return getReader() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRealPath( String path)

The default behavior of this method is to return getRealPath(String path) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRemoteAddr()

The default behavior of this method is to return getRemoteAddr() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRemoteHost()

The default behavior of this method is to return getRemoteHost() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getRemotePort()

The default behavior of this method is to return getRemotePort() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletRequest getRequest()

Return the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

RequestDispatcher getRequestDispatcher( String path)

The default behavior of this method is to return getRequestDispatcher(String path) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getScheme()

The default behavior of this method is to return getScheme() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServerName()

The default behavior of this method is to return getServerName() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getServerPort()

The default behavior of this method is to return getServerPort() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isSecure()

The default behavior of this method is to return isSecure() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void removeAttribute( String name)

The default behavior of this method is to call removeAttribute(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setAttribute( String name, Object o)

The default behavior of this method is to return setAttribute(String name, Object o) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setCharacterEncoding( String enc)

The default behavior of this method is to set the character encoding on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setRequest(ServletRequest request)

Sets the request object being wrapped.

**javax.servlet . ServletResponseWrapper**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides a convenient implementation of the ServletResponse interface

that can be subclassed by developers wishing to adapt the response from a Servlet.

This class implements the Wrapper or Decorator pattern.

Methods default to calling through to the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

--------------------

ServletResponseWrapper(ServletResponse response)

Creates a ServletResponse adaptor wrapping the given response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-------------------

void flushBuffer()

The default behavior of this method is to call flushBuffer() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getBufferSize()

The default behavior of this method is to return getBufferSize() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getCharacterEncoding()

The default behavior of this method is to return getCharacterEncoding() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getContentType()

The default behavior of this method is to return getContentType() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Locale getLocale()

The default behavior of this method is to return getLocale() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletOutputStream getOutputStream()

The default behavior of this method is to return getOutputStream() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ServletResponse getResponse()

Return the wrapped ServletResponse object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PrintWriter getWriter()

The default behavior of this method is to return getWriter() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isCommitted()

The default behavior of this method is to return isCommitted() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void reset()

The default behavior of this method is to call reset() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void resetBuffer()

The default behavior of this method is to call resetBuffer() on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setBufferSize(int size)

The default behavior of this method is to call setBufferSize(int size) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setCharacterEncoding( String charset)

The default behavior of this method is to call setCharacterEncoding(String charset) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setContentLength(int len)

The default behavior of this method is to call setContentLength(int len) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setContentType( String type)

The default behavior of this method is to call setContentType(String type) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setLocale( Locale loc)

The default behavior of this method is to call setLocale(Locale loc) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setResponse(ServletResponse response)

Sets the response being wrapped.

**END**

**Javax.servlet.http.\***

**Interfaces**

javax.servlet.http. **HttpServletRequest**

|

ServletRequest[i]

|

HttpServletRequestWrapper[c]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public interface HttpServletRequest extends ServletRequest

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The servlet container creates an HttpServletRequest object and passes it as an argument to the servlet's service methods (doGet, doPost, etc).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Field

----------------

public static final String BASIC\_AUTH

String identifier for Basic authentication. Value "BASIC"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public static final String FORM\_AUTH

String identifier for Form authentication. Value "FORM"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public static final String CLIENT\_CERT\_AUTH

String identifier for Client Certificate authentication. Value "CLIENT\_CERT"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public static final String DIGEST\_AUTH

String identifier for Digest authentication. Value "DIGEST"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

--------

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getAuthType()

Returns the name of the authentication scheme used to protect the servlet.

If the servlet is not authenticated null is returned.

Same as the value of the CGI variable AUTH\_TYPE.

Returns:

one of the static members BASIC\_AUTH, FORM\_AUTH, CLIENT\_CERT\_AUTH, DIGEST\_AUTH

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Cookie[] getCookies()

Returns an array containing all of the Cookie objects the client sent with this request.

This method returns null if no cookies were sent.

Returns:

an array of all the Cookies included with this request, or null if the request has no cookies

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public long getDateHeader( String name)

Returns the value of the specified request header as a long value that represents a Date object.

The date is returned as the number of milliseconds since January 1, 1970 GMT. The header name is case insensitive.

If the request did not have a header of the specified name, this method returns -1.

If the header can't be converted to a date, the method throws an IllegalArgumentException.

Parameters:

name - a String specifying the name of the header

Throws:

IllegalArgumentException - If the header value can't be converted to a date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getHeader( String name)

Returns the value of the specified request header as a String.

If the request did not include a header of the specified name, this method returns null.

If there are multiple headers with the same name, this method returns the first head in the request.

The header name is case INSensitive. You can use this method with any request header.

Parameters:

name - a String specifying the header name

Returns:

a String containing the value of the requested header, or

null if the request does not have a header of that name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Enumeration getHeaders( String name)

Returns all the values of the specified request header as an Enumeration of String objects.

Parameters:

name - a String specifying the header name

Returns:

an Enumeration containing the values of the requested header.

If the request does not have any headers of that name return an empty enumeration.

If the container does not allow access to header information, return null

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public Enumeration getHeaderNames()

Returns an enumeration of all the header names this request contains.

Returns:

an enumeration of all the header names sent with this request;

if the request has no headers, an empty enumeration;

if the servlet container does not allow servlets to use this method, null

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public int getIntHeader( String name)

Returns the value of the specified request header as an int.

Parameters:

name - a String specifying the name of a request header

Returns:

an integer expressing the value of the request header or -1 if the request doesn't have a header of this name

Throws:

NumberFormatException - If the header value can't be converted to an int

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getMethod()

Returns the name of the HTTP method with which this request was made,

for example, GET, POST, or PUT.

Same as the value of the CGI variable REQUEST\_METHOD.

Returns:

a String specifying the name of the method with which this request was made

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getPathInfo()

Returns any extra path information associated with the URL the client sent when it made this request.

The extra path information follows the servlet path but precedes the query string and will start with a "/" character.

Same as the value of the CGI variable PATH\_INFO.

Returns:

a String, specifying extra path information that comes after the servlet path but before the query string in the request URL;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getPathTranslated()

Returns any extra path information after the servlet name but before the query string, and translates it to a real path.

Same as the value of the CGI variable PATH\_TRANSLATED.

Returns:

a String specifying the real path, or null if the URL does not have any extra path information

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getContextPath()

Returns the portion of the request URI that indicates the context of the request.

The context path always comes first in a request URI.

The path starts with a "/" character but does not end with a "/" character.

Returns:

a String specifying the portion of the request URI that indicates the context of the request

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getQueryString()

Returns the query string that is contained in the request URL after the path.

This method returns null if the URL does not have a query string.

Same as the value of the CGI variable QUERY\_STRING.

Returns:

a String containing the query string or null if the URL contains no query string.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getRemoteUser()

It returns the name of the USER who's Authentication is Verified, If not Authorized User It returns NULL

Same as the value of the CGI variable REMOTE\_USER.

Returns:

a String specifying the login of the user making this request, or null if the user login is not known

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isUserInRole( String role)

Returns a boolean indicating whether the authenticated user is included in the specified logical "role".

Roles defined in deployment descriptors. If the user has not been authenticated, the method returns false.

Parameters:

role - a String specifying the name of the role

Returns:

a boolean indicating whether the user making this request belongs to a given role; false if the user has not been authenticated

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public java.security.Principal getUserPrincipal()

Returns a java.security.Principal object containing the name of the current authenticated user.

If the user has not been authenticated, the method returns null.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getRequestedSessionId()

Returns the session ID specified by the client.

Returns:

a String specifying the session ID, or null if the request did not specify a session ID

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getRequestURI()

Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.

POST /some/path.html HTTP/1.1 /some/path.html

GET http://foo.bar/a.html HTTP/1.0 /a.html

HEAD /xyz?a=b HTTP/1.1 /xyz

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public StringBuffer getRequestURL()

Reconstructs the URL the client used to make the request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String getServletPath()

Returns the part of this request's URL that calls the servlet.

This path starts with a "/" character and includes either the servlet name or a path to the servlet,

but does not include any extra path information or a query string.

Same as the value of the CGI variable SCRIPT\_NAME.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public HttpSession getSession(boolean create)

Returns the current HttpSession associated with this request.

if there is no current session and create is true, returns a new session.

If create is false and the request has no valid HttpSession, this method returns null.

To make sure the session is properly maintained, you must call this method before the response is committed.

Parameters:

create - true to create a new session for this request if necessary; false to return null if there's no current session

Returns:

the HttpSession associated with this request or null if create is false and the request has no valid session

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public HttpSession getSession()

Returns the current session associated with this request, or if the request does not have a session, creates one.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isRequestedSessionIdValid()

Checks whether the requested session ID is still valid.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isRequestedSessionIdFromCookie()

Checks whether the requested session ID came in as a cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isRequestedSessionIdFromURL()

Checks whether the requested session ID came in as part of the request URL.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean isRequestedSessionIdFromUrl()

use isRequestedSessionIdFromURL() instead.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

javax.servlet.http .**HttpServletResponse[i]**

|

ServletResponse[i]

|

HttpServletResponseWrapper[c]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public interface HttpServletResponse extends ServletResponse

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Extends the ServletResponse interface to provide HTTP-specific functionality in sending a response.

For example, it has methods to access HTTP headers and cookies.

The servlet container creates an HttpServletResponse object and passes it as an argument to the servlet's service methods (doGet, doPost, etc).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Field

-------------------------------

See

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void addCookie(Cookie cookie)

Adds the specified cookie to the response. This method can be called multiple times to set more than one cookie.

Parameters:

cookie - the Cookie to return to the client

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public boolean containsHeader( String name)

Returns a boolean indicating whether the named response header has already been set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String encodeURL( String url)

Encodes the specified URL by including the session ID in it.

Parameters:

url - the url to be encoded.

Returns:

the encoded URL if encoding is needed; the unchanged URL otherwise.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String encodeRedirectURL( String url)

Encodes the specified URL for use in the sendRedirect method .

Parameters:

url - the url to be encoded.

Returns:

the encoded URL if encoding is needed; the unchanged URL otherwise.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String encodeUrl( String url)

use encodeURL(String url) instead

Parameters:

url - the url to be encoded.

Returns:

the encoded URL if encoding is needed; the unchanged URL otherwise.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public String encodeRedirectUrl( String url)

use encodeRedirectURL(String url) instead

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void sendError(int sc, String msg) throws IOException

Sends an error response to the client using the specified status.

The server defaults to creating the response to look like an HTML

Parameters:

sc - the error status code

msg - the descriptive message

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void sendError(int sc) throws IOException

Sends an error response to the client using the specified status code and clearing the buffer.

Parameters:

sc - the error status code

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void sendRedirect( String location) throws IOException

Sends a temporary redirect response to the client using the specified redirect location URL.

Parameters:

location - the redirect location URL

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setDateHeader( String name, long date)

Sets a response header with the given name and date-value.

Parameters:

name - the name of the header to set

date - the assigned date value

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void addDateHeader( String name, long date)

Adds a response header with the given name and date-value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setHeader( String name, String value)

Sets a response header with the given name and value.

Parameters:

name - the name of the header

value - the header value If it contains octet string, it should be encoded according to RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void addHeader( String name, String value)

Adds a response header with the given name and value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setIntHeader( String name, int value)

Sets a response header with the given name and integer value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void addIntHeader( String name, int value)

Adds a response header with the given name and integer value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setStatus(int sc)

Sets the status code for this response.

Parameters:

sc - the status code

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void setStatus(int sc, String sm)

Sets the status code and message for this response.

Parameters:

sc - the status code

sm - the status message

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http . HttpSession**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.

The servlet container uses this interface to create a session between an HTTP client and an HTTP server.

The session persists for a specified time period, across more than one connection or page request from the user.

A session usually corresponds to one user, who may visit a site many times.

The server can maintain a session in many ways such as using cookies or rewriting URLs.

This interface allows servlets to View and manipulate information about a session,

such as the session identifier, creation time, and last accessed time

When an application stores an object in or removes an object from a session,

Session information is scoped only to the current web application (ServletContext),

so information stored in one context will not be directly visible in another.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

----------------------------

Object getAttribute( String name)

Returns the object bound with the specified name in this session, or null if no object is bound under the name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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 received 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()

this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getValue( String name)

this method is replaced by getAttribute( String).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String[] getValueNames()

this method is replaced by getAttributeNames()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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( String name, Object value)

this method is replaced by setAttribute( String, Object)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void removeAttribute( String name)

Removes the object bound with the specified name from this session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void removeValue( String name)

this method is replaced by removeAttribute( String)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setAttribute( String name, Object value)

Binds an object to this session, using the name specified.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setMaxInactiveInterval(int interval)

Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http .HttpSessionActivationListener**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Objects that are bound to a session may listen to container events notifying them that sessions will be passivated and

that session will be activated.

A container that migrates session between VMs or persists sessions is required to notify all attributes bound to sessions implementing

HttpSessionActivationListener.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Methods

-----------------

public void sessionWillPassivate(HttpSessionEvent se)

Notification that the session is about to be passivated.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public void sessionDidActivate(HttpSessionEvent se)

Notification that the session has just been activated.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http.HttpSessionAttributeListener**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This listener interface can be implemented in order to get notifications of changes to the attribute

lists of sessions within this web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-----------------

void attributeAdded(HttpSessionBindingEvent se)

Notification that an attribute has been added to a session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeRemoved(HttpSessionBindingEvent se)

Notification that an attribute has been removed from a session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void attributeReplaced(HttpSessionBindingEvent se)

Notification that an attribute has been replaced in a session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http. HttpSessionBindingListener**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Causes an object to be notified when it is bound to or unbound from a session.

The object is notified by an HttpSessionBindingEvent object.

This may be as a result of a servlet programmer explicitly unbinding an attribute from a session,

due to a session being invalidated, or due to a session timing out.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method Summary

--------------------------

void valueBound(HttpSessionBindingEvent event)

Notifies the object that it is being bound to a session and identifies the session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void valueUnbound(HttpSessionBindingEvent event)

Notifies the object that it is being unbound from a session and identifies the session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http . HttpSessionContext**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public HttpSession getSession(java.lang.String sessionId)

getIds

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http. HttpSessionListener**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Implementations of this interface are notified of changes to the list of active sessions in a web application.

To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

------------------

void sessionCreated(HttpSessionEvent se)

Notification that a session was created.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void sessionDestroyed(HttpSessionEvent se)

Notification that a session is about to be invalidated.

**Classes**

**javax.servlet.http .Cookie[c]**

|

Cloneable[i]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class Cookie extends Object implements Cloneable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server.

A cookie's value can uniquely identify a client, so cookies are commonly used for session management.

A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers,

a maximum age, and a version number.

Some Web browsers have bugs in how they handle the optional attributes,

so use them sparingly to improve the interoperability of your servlets.

The servlet sends cookies to the browser by using the "HttpServletResponse.addCookie(javax.servlet.http.Cookie) method"

, which adds fields to HTTP response headers to send cookies to the browser, one at a time.

The browser is expected to support 20 cookies for each Web server, 300 cookies total, and may limit cookie size to 4 KB each.

The browser returns cookies to the servlet by adding fields to HTTP request headers.

Cookies can be retrieved from a request by using the "HttpServletRequest.getCookies() method".

Several cookies might have the same name but different path attributes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

---------------------

Cookie( String name, String value)

Constructs a cookie with a specified name and value.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

----------------------

Object clone()

Overrides the standard Object.clone method to return a copy of this cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getComment()

Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getDomain()

Returns the domain name set for this cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getMaxAge()

Returns the maximum age of the cookie, specified in seconds, By default, -1 indicating the cookie will persist until browser shutdown.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getName()

Returns the name of the cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getPath()

Returns the path on the server to which the browser returns this cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean getSecure()

Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getValue()

Returns the value of the cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getVersion()

Returns the version of the protocol this cookie complies with.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setComment( String purpose)

Specifies a comment that describes a cookie's purpose.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setDomain( String pattern)

Specifies the domain within which this cookie should be presented.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setMaxAge(int expiry)

Sets the maximum age of the cookie in seconds.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setPath( String uri)

Specifies a path for the cookie to which the client should return the cookie.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setSecure(boolean flag)

Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setValue( String newValue)

Assigns a new value to a cookie after the cookie is created.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setVersion(int v)

Sets the version of the cookie protocol this cookie complies with.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http . HttpServlet**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public abstract class HttpServlet extends GenericServlet implements Serializable

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.

A subclass of HttpServlet must override at least one method, usually one of these:

-doGet()---------> if the servlet supports HTTP GET requests

-doPost()--------> for HTTP POST requests

-doPut()---------> for HTTP PUT requests

-doDelete()-----> for HTTP DELETE requests

init and destroy, to manage resources that are held for the life of the servlet

getServletInfo, which the servlet uses to provide information about itself

There's almost no reason to override the service method.

service handles standard HTTP requests by dispatching them to the handler methods for each HTTP request type

(the doXXX methods listed above).

Likewise, there's almost no reason to override the doOptions and doTrace methods.

Servlets typically run on multithreaded servers, so be aware that a servlet must handle concurrent requests and

be careful to synchronize access to shared resources.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

---------------------

HttpServlet()

Does nothing, because this is an abstract class.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

------------------

protected void doGet(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a GET request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doPost(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a POST request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void service(HttpServletRequest req, HttpServletResponse resp)

Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void service(ServletRequest req, ServletResponse res)

Dispatches client requests to the protected service method.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doDelete(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a DELETE request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doHead(HttpServletRequest req, HttpServletResponse resp)

Receives an HTTP HEAD request from the protected service method and handles the request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doOptions(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a OPTIONS request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doPut(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a PUT request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected void doTrace(HttpServletRequest req, HttpServletResponse resp)

Called by the server (via the service method) to allow a servlet to handle a TRACE request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

protected long getLastModified(HttpServletRequest req)

Returns the time the HttpServletRequest object was last modified, in milliseconds since midnight January 1, 1970 GMT.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**public class HttpServletRequestWrapper** extends ServletRequestWrapper implements HttpServletRequest

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides a convenient implementation of the HttpServletRequest interface that can be subclassed by developers

wishing to adapt the request to a Servlet.

This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

-----------------------

HttpServletRequestWrapper(HttpServletRequest request)

Constructs a request object wrapping the given request.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

-----------------

String getAuthType()

The default behavior of this method is to return getAuthType() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getContextPath()

The default behavior of this method is to return getContextPath() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Cookie[] getCookies()

The default behavior of this method is to return getCookies() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

long getDateHeader( String name)

The default behavior of this method is to return getDateHeader(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getHeader( String name)

The default behavior of this method is to return getHeader(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getHeaderNames()

The default behavior of this method is to return getHeaderNames() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Enumeration getHeaders( String name)

The default behavior of this method is to return getHeaders(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

int getIntHeader( String name)

The default behavior of this method is to return getIntHeader(String name) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getMethod()

The default behavior of this method is to return getMethod() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getPathInfo()

The default behavior of this method is to return getPathInfo() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getPathTranslated()

The default behavior of this method is to return getPathTranslated() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getQueryString()

The default behavior of this method is to return getQueryString() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRemoteUser()

The default behavior of this method is to return getRemoteUser() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRequestedSessionId()

The default behavior of this method is to return getRequestedSessionId() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getRequestURI()

The default behavior of this method is to return getRequestURI() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

StringBuffer getRequestURL()

The default behavior of this method is to return getRequestURL() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String getServletPath()

The default behavior of this method is to return getServletPath() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HttpSession getSession()

The default behavior of this method is to return getSession() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HttpSession getSession(boolean create)

The default behavior of this method is to return getSession(boolean create) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

java.security.Principal getUserPrincipal()

The default behavior of this method is to return getUserPrincipal() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isRequestedSessionIdFromCookie()

The default behavior of this method is to return isRequestedSessionIdFromCookie() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isRequestedSessionIdFromUrl()

The default behavior of this method is to return isRequestedSessionIdFromUrl() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isRequestedSessionIdFromURL()

The default behavior of this method is to return isRequestedSessionIdFromURL() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isRequestedSessionIdValid()

The default behavior of this method is to return isRequestedSessionIdValid() on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean isUserInRole( String role)

The default behavior of this method is to return isUserInRole(String role) on the wrapped request object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http .HttpServletResponseWrapper**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Provides a convenient implementation of the HttpServletResponse interface that can be subclassed by

developers wishing to adapt the response from a Servlet.

This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Constructor

-----------------

HttpServletResponseWrapper(HttpServletResponse response)

Constructs a response adaptor wrapping the given response.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

--------------------

void addCookie(Cookie cookie)

The default behavior of this method is to call addCookie(Cookie cookie) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void addDateHeader( String name, long date)

The default behavior of this method is to call addDateHeader(String name, long date) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void addHeader( String name, String value)

The default behavior of this method is to return addHeader(String name, String value) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void addIntHeader( String name, int value)

The default behavior of this method is to call addIntHeader(String name, int value) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

boolean containsHeader( String name)

The default behavior of this method is to call containsHeader(String name) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String encodeRedirectUrl( String url)

The default behavior of this method is to return encodeRedirectUrl(String url) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String encodeRedirectURL( String url)

The default behavior of this method is to return encodeRedirectURL(String url) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String encodeUrl( String url)

The default behavior of this method is to call encodeUrl(String url) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

String encodeURL( String url)

The default behavior of this method is to call encodeURL(String url) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void sendError(int sc)

The default behavior of this method is to call sendError(int sc) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void sendError(int sc, String msg)

The default behavior of this method is to call sendError(int sc, String msg) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void sendRedirect( String location)

The default behavior of this method is to return sendRedirect(String location) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setDateHeader( String name, long date)

The default behavior of this method is to call setDateHeader(String name, long date) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setHeader( String name, String value)

The default behavior of this method is to return setHeader(String name, String value) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setIntHeader( String name, int value)

The default behavior of this method is to call setIntHeader(String name, int value) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setStatus(int sc)

The default behavior of this method is to call setStatus(int sc) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http .HttpSessionBindingEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

public class HttpSessionBindingEvent extends HttpSessionEvent

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Events of this type are either sent to an object that implements HttpSessionBindingListener when it is bound or unbound from a session,

or to a HttpSessionAttributeListener that has been configured in the deployment descriptor when any attribute is bound, unbound or

replaced in a session.

Constructor

----------------------------

HttpSessionBindingEvent(HttpSession session, String name)

Constructs an event that notifies an object that it has been bound to or unbound from a session.

HttpSessionBindingEvent(HttpSession session, String name, Object value)

Constructs an event that notifies an object that it has been bound to or unbound from a session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

--------------

String getName()

Returns the name with which the attribute is bound to or unbound from the session.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

HttpSession getSession()

Return the session that changed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Object getValue()

Returns the value of the attribute that has been added, removed or replaced.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

void setStatus(int sc, String sm)

The default behavior of this method is to call setStatus(int sc, String sm) on the wrapped response object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http . HttpSessionEvent**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is the class representing event notifications for changes to sessions within a web application.

Constructor

------------------

HttpSessionEvent(HttpSession source)

Construct a session event from the given source.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

--------------------

HttpSession getSession()

Return the session that changed.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**javax.servlet.http . HttpUtils**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

javax.servlet.http.HttpUtils

Deprecated. As of Java(tm) Servlet API 2.3.

These methods were only useful with the default encoding and have been moved to the request interfaces.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Method

---------------

static java.lang.StringBuffer getRequestURL(HttpServletRequest req)

Deprecated. Reconstructs the URL the client used to make the request, using information in the HttpServletRequest object.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

static java.util.Hashtable parsePostData(int len, ServletInputStream in)

Deprecated. Parses data from an HTML form that the client sends to the server using the HTTP POST method and the application/x-www-form-urlencoded MIME type.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

static java.util.Hashtable parseQueryString(java.lang.String s)

Deprecated. Parses a query string passed from the client to the server and builds a HashTable object with key-value pairs.