

WsRequest

 Apr 05, 2024 11:35

Supported Clients

SmartClient WebClient NGClient

Property Summary

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

Methods Summary

Number	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 <code>null</code> if the type is not known.
Array	getContents()	Get raw the contents of the request.
String	getContextPath()	Returns the portion of the request URI that indicates the context of the request.
Array	getCookies()	Returns an array containing all of the <code>Cookie</code> objects the client sent with this request.
Number	getDateHeader(name)	Returns the value of the specified request header as a <code>long</code> value that represents a <code>Date</code> object.
String	getHeader(name)	Returns the value of the specified request header as a String.
Array	getHeaderNames()	Returns an enumeration of all the header names this request contains.
Array	getHeaders(name)	Returns all the values of the specified request header as an array of <code>String</code> objects.
Number	getIntHeader(name)	Returns the value of the specified request header as an <code>int</code> .
String	getLocalAddr()	Returns the Internet Protocol (IP) address of the interface on which the request was received.
String	getLocalName()	Returns the host name of the Internet Protocol (IP) interface on which the request was received.
Number	getLocalPort()	Returns the Internet Protocol (IP) port number of the interface on which the request was received.
String	getLocaleLanguageTag()	Returns the preferred <code>Locale</code> that the client will accept content in, based on the Accept-Language header.
Array	getLocalesLanguageTags()	Returns an array of <code>Locale</code> 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	getMethod()	Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT.
String	getParameter(name)	Returns the value of a request parameter as a <code>String</code> , or <code>null</code> if the parameter does not exist.
Object	getParameterMap()	Returns an object of the parameters of this request.
Array	getParameterNames()	Returns an array of <code>String</code> objects containing the names of the parameters contained in this request.
Array	getParameterValues(name)	Returns an array of <code>String</code> objects containing all of the values the given request parameter has, or <code>null</code> if the parameter does not exist.
String	getPathInfo()	Returns any extra path information associated with the URL the client sent when it made this request.
String	getPathTranslated()	Returns any extra path information after the servlet name but before the query string, and translates it to a real path.
String	getProtocol()	Returns the name and version of the protocol the request uses in the form <code><i>protocol/majorVersion</code> .
String	getQueryString()	Returns the query string that is contained in the request URL after the path.
String	getRealPath(path)	Gets the <code><i>real</i></code> path corresponding to the given <code><i>virtual</i></code> path.
String	getRemoteAddr()	Returns the Internet Protocol (IP) address of the client or last proxy that sent the request.
String	getRemoteHost()	Returns the fully qualified name of the client or the last proxy that sent the request.
Number	getRemotePort()	Returns the Internet Protocol (IP) source port of the client or last proxy that sent the request.
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.
String	getRequestURL()	Reconstructs the URL the client used to make the request.
String	getScheme()	Returns the name of the scheme used to make this request, for example, <code>http</code> , <code>https</code> , or <code>ftp</code> .
String	getServerName()	Returns the host name of the server to which the request was sent.
Number	getServerPort()	Returns the port number to which the request was sent.
String	getServletPath()	Returns the part of this request's URL that calls the servlet.
Boolean	isSecure()	Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.

Property Details

characterEncoding

Returns the name of the character encoding used in the body of this request. This method returns `null` if the request does not specify a character encoding

Returns

[String](#) a `String` containing the name of the character encoding, or `null` if the request does not specify a character encoding

Supported Clients

SmartClient, WebClient, NGClient

Sample

Methods Details

getLength()

Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known. For HTTP servlets, same as the value of the CGI variable CONTENT_LENGTH.

Returns

[Number](#) a long containing the length of the request body or -1L if the length is not known

Supported Clients

SmartClient, WebClient, NGClient

Sample

getContentType()

Returns the MIME type of the body of the request, or `null` if the type is not known. For HTTP servlets, same as the value of the CGI variable CONTENT_TYPE.

Returns

[String](#) a `String` containing the name of the MIME type of the request, or null if the type is not known

Supported Clients

SmartClient, WebClient, NGClient

Sample

getContents()

Get raw the contents of the request.

In case of multipart request, all uploaded items are listed separately.

In case of a request with a single body, the contents array consists of one item, the body.

This method returns an empty array if the request has no contents.

Returns

[Array](#) an array of WsContents objects.

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var request = plugins.rest_ws.getRequest();
var contents = request.getContents();
```

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. For servlets in the default (root) context, this method returns "". The container does not decode this string.

<p>It is possible that a servlet container may match a context by more than one context path. In such cases this method will return the actual context path used by the request and it may differ from the path returned by the `javax.servlet.ServletContext#getContextPath()` method. The context path returned by `javax.servlet.ServletContext#getContextPath()` should be considered as the prime or preferred context path of the application.

Returns

[String](#) a `String` specifying the portion of the request URI that indicates the context of the request

Supported Clients

SmartClient,WebClient,NGClient

Sample

getCookies()

Returns an array containing all of the `Cookie` objects the client sent with this request. This method returns an empty array if no cookies were sent.

Returns

[Array](#) an array of all the `Cookies` included with this request

Supported Clients

SmartClient,WebClient,NGClient

Sample

getDateHeader(name)

Returns the value of the specified request header as a `long` value that represents a `Date` object. Use this method with headers that contain dates, such as `If-Modified-Since`.

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

<p>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

[String](#) name a `String` specifying the name of the header

Returns

[Number](#) a `long` value representing the date specified in the header expressed as the number of milliseconds since January 1, 1970 GMT, or -1 if the named header was not included with the request

Supported Clients

SmartClient,WebClient,NGClient

Sample

getHeader(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

[String](#) name The name of the header to get

Returns[String](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var request = plugins.rest_ws.getRequest();
var header = request.getHeader('');
```

getHeaderNames()

Returns an enumeration of all the header names this request contains. If the request has no headers, this method returns an empty enumeration.

<p>Some servlet containers do not allow servlets to access headers using this method, in which case this method returns `null`</p>

Returns

[Array](#) 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`

Supported Clients

SmartClient,WebClient,NGClient

Sample**getHeaders(name)**

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

<p>Some headers, such as `Accept-Language` can be sent by clients as several headers each with a different value rather than sending the header as a comma separated list.

<p>If the request did not include any headers of the specified name, this method returns an empty array.
The header name is case insensitive. You can use this method with any request header.

Parameters

[String](#) name a `String` specifying the header name

Returns

[Array](#) an array 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

Supported Clients

SmartClient,WebClient,NGClient

Sample**getIntHeader(name)**

Returns the value of the specified request header as an `int`. If the request does not have a header of the specified name, this method returns -1. If the header cannot be converted to an integer, this method throws a `NumberFormatException`.

<p>The header name is case insensitive.

Parameters

[String](#) name a `String` specifying the name of a request header

Returns

[Number](#) an integer expressing the value of the request header or -1 if the request doesn't have a header of this name

Supported Clients

SmartClient,WebClient,NGClient

Sample
getLocalAddr()

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

Returns

[String](#) a `String` containing the IP address on which the request was received.

Supported Clients

SmartClient, WebClient, NGClient

Sample
getLocalName()

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

Returns

[String](#) a `String` containing the host name of the IP on which the request was received.

Supported Clients

SmartClient, WebClient, NGClient

Sample
getLocalPort()

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

Returns

[Number](#) an integer specifying the port number

Supported Clients

SmartClient, WebClient, NGClient

Sample
getLocaleLanguageTag()

Returns the preferred `Locale` that the client will accept content in, based on the Accept-Language header. If the client request doesn't provide an Accept-Language header, this method returns the default locale for the server.

Returns a well-formed IETF BCP 47 language tag representing this locale.

Returns

[String](#) the preferred `Locale` for the client

Supported Clients

SmartClient, WebClient, NGClient

Sample
getLocalesLanguageTags()

Returns an array 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.

If the client request doesn't provide an Accept-Language header, this method returns an array containing one `Locale`, the default locale for the server.

Returns well-formed IETF BCP 47 language tags representing the locales.

Returns

[Array](#) an array of preferred `Locale` objects for the client

Supported Clients

SmartClient,WebClient,NGClient

Sample

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

[String](#) a `String` specifying the name of the method with which this request was made

Supported Clients

SmartClient,WebClient,NGClient

Sample

getParameter(name)

Returns the value of a request parameter as a `String`, or `null` if the parameter does not exist. Request parameters are extra information sent with the request. For HTTP servlets, parameters are contained in the query string or posted form data.

<p>You should only use this method when you are sure the parameter has only one value. If the parameter might have more than one value, use `#getParameterValues(String)`.

<p>If you use this method with a multivalued parameter, the value returned is equal to the first value in the array returned by `#getParameterValues`.

<p>If the parameter data was sent in the request body, such as occurs with an HTTP POST request, then reading the body directly via `#getInputStream` or `#getReader` can interfere with the execution of this method.

Parameters

[String](#) name a `String` specifying the name of the parameter

Returns

[String](#) a `String` representing the single value of the parameter

Supported Clients

SmartClient,WebClient,NGClient

Sample

getParameterMap()

Returns an object of the parameters of this request.

<p>Request parameters are extra information sent with the request. For HTTP servlets, parameters are contained in the query string or posted form data.

Returns

[Object](#) an object containing parameter names as keys and parameter values as map values. The keys in the parameter map are of type `String`. The values in the parameter map are of type `String` array.

Supported Clients

SmartClient,WebClient,NGClient

Sample

getParameterNames()

Returns an array of `String` objects containing the names of the parameters contained in this request. If the request has no parameters, the method returns an empty array.

Returns

[Array](#) an array of `String` objects, each `String` containing the name of a request parameter; or an empty array if the request has no parameters

Supported Clients

SmartClient, WebClient, NGClient

Sample

getParameterValues(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.

If the parameter has a single value, the array has a length of 1.

Parameters

[String](#) name a `String` containing the name of the parameter whose value is requested

Returns

[Array](#) an array of `String` objects containing the parameter's values

Supported Clients

SmartClient, WebClient, NGClient

Sample

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.

This method returns `null` if there was no extra path information.

Same as the value of the CGI variable PATH_INFO.

Returns

[String](#) a `String`, decoded by the web container, specifying extra path information that comes after the servlet path but before the query string in the request URL; or `null` if the URL does not have any extra path information

Supported Clients

SmartClient, WebClient, NGClient

Sample

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.

If the URL does not have any extra path information, this method returns `null` or the servlet container cannot translate the virtual path to a real path for any reason (such as when the web application is executed from an archive).

The web container does not decode this string.

Returns

[String](#) a `String` specifying the real path, or `null` if the URL does not have any extra path information

Supported Clients

SmartClient, WebClient, NGClient

Sample**getProtocol()**

Returns the name and version of the protocol the request uses in the form `<i>protocol/majorVersion.minorVersion</i>`, for example, HTTP/1.1. For HTTP servlets, the value returned is the same as the value of the CGI variable `<code>SERVER_PROTOCOL</code>`.

Returns

[String](#) a `<code>String</code>` containing the protocol name and version number

Supported Clients

SmartClient, WebClient, NGClient

Sample**getQueryString()**

Returns the query string that is contained in the request URL after the path. This method returns `<code>null</code>` if the URL does not have a query string. Same as the value of the CGI variable `QUERY_STRING`.

Returns

[String](#) a `<code>String</code>` containing the query string or `<code>null</code>` if the URL contains no query string. The value is not decoded by the container.

Supported Clients

SmartClient, WebClient, NGClient

Sample**getRealPath(path)**

Gets the `<i>real</i>` path corresponding to the given `<i>virtual</i>` path.

Parameters

[String](#) path the `<i>virtual</i>` path to be translated to a `<i>real</i>` path

Returns

[String](#) the `<i>real</i>` path, or `<tt>null</tt>` if the translation cannot be performed

Supported Clients

SmartClient, WebClient, NGClient

Sample**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 `<code>REMOTE_ADDR</code>`.

Returns

[String](#) a `<code>String</code>` containing the IP address of the client that sent the request

Supported Clients

SmartClient, WebClient, NGClient

Sample**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 `<code>REMOTE_HOST</code>`.

Returns

[String](#) a `String` containing the fully qualified name of the client

Supported Clients

SmartClient, WebClient, NGClient

Sample**getRemotePort()**

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

Returns

[Number](#) an integer specifying the port number

Supported Clients

SmartClient, WebClient, NGClient

Sample**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. The web container does not decode this String. For example:

```
<table summary="Examples of Returned Values">
<tr align=left><th>First line of HTTP request      </th>
<th>      Returned Value</th>
<tr><td>POST /some/path.html HTTP/1.1<td><td>/some/path.html
<tr><td>GET http://foo.bar/a.html HTTP/1.0
<td><td>/a.html
<tr><td>HEAD /xyz?a=b HTTP/1.1<td><td>/xyz
</table>
```

Returns

[String](#) a `String` containing the part of the URL from the protocol name up to the query string

Supported Clients

SmartClient, WebClient, NGClient

Sample**getRequestURL()**

Reconstructs the URL the client used to make the request. The returned URL contains a protocol, server name, port number, and server path, but it does not include query string parameters.

If this request has been forwarded using `javax.servlet.RequestDispatcher#forward(ServletRequest, ServletResponse)`, the server path in the reconstructed URL must reflect the path used to obtain the `RequestDispatcher`, and not the server path specified by the client.

This method is useful for creating redirect messages and for reporting errors.

Returns

[String](#) a `StringBuffer` object containing the reconstructed URL

Supported Clients

SmartClient, WebClient, NGClient

Sample**getScheme()**

Returns the name of the scheme used to make this request, for example, `http`, `https`, or `ftp`. Different schemes have different rules for constructing URLs, as noted in RFC 1738.

Returns

[String](#) a `String` containing the name of the scheme used to make this request

Supported Clients

SmartClient,WebClient,NGClient

Sample**getServerName()**

Returns the host name of the server to which the request was sent. It is the value of the part before ":" in the `Host` header value, if any, or the resolved server name, or the server IP address.

Returns

[String](#) a `String` containing the name of the server

Supported Clients

SmartClient,WebClient,NGClient

Sample**getServerPort()**

Returns the port number to which the request was sent. It is the value of the part after ":" in the `Host` header value, if any, or the server port where the client connection was accepted on.

Returns

[Number](#) an integer specifying the port number

Supported Clients

SmartClient,WebClient,NGClient

Sample**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`.

This method will return an empty string ("") if the servlet used to process this request was matched using the "/" pattern.

Returns

[String](#) a `String` containing the name or path of the servlet being called, as specified in the request URL, decoded, or an empty string if the servlet used to process the request is matched using the "/" pattern.

Supported Clients

SmartClient,WebClient,NGClient

Sample**isSecure()**

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

Returns

[Boolean](#) a boolean indicating if the request was made using a secure channel

Supported Clients

SmartClient,WebClient,NGClient

Sample

