Request Logicsheet (2.1 legacy document)

Table of contents

Table of contents

Warning:
This document was copied as is from the Cocoon 2.1 documentation, but has not yet been fully reviewed or moved to its new home.

1. Description

The Request logicsheet (taglib) is an XSP logicsheet that wraps XML tags around standard request operations. Specifically, the Request logicsheet provides an XML interface to most methods of the HttpServletRequest object (see the Java Servlet API Specification, version 2.2) for more information.

The Request tags provide information about all aspects of the current request, such as the request method (GET, POST, etc.), what protocol is in use (http, https), cookie information, preferred locale, and so forth. The Request logicsheet is probably used mostly, however, for obtaining field values from a submitted HTML form. See xsp-request:get-parameter below for more information on this topic.

2. Usage

As an XSP logicsheet, the Request logicsheet can only be used in an XSP page. It may be helpful to be familiar with XSP before working with this (or any) logicsheet.

To use the Request logicsheet, you must first declare the *xsp-request* namespace, mapping it to the uri *http://apache.org/xsp/request/2.0*. This is typically done like this:

```
<xsp:page xmlns:xsp="http://apache.org/xsp"
xmlns:xsp-request="http://apache.org/xsp/request/2.0" > ... </xsp:page>
```

You may then use any of the elements in the *request* namespace described in the Elements Reference section below.

3. Example Code

The following code shows a typical example of using the Request logicsheet. The output elements display the request method and the value of the query parameter "fruit". Of course, rather than displaying these values as we've done, you might instead store them in elements and process them further through an XSLT stylesheet, for instance.

```
<?xml version="1.0"?> <xsp:page xmlns:xsp="http://apache.org/xsp"
xmlns:xsp-request="http://apache.org/xsp/request/2.0" > <html> <b>Request method:</b>
<xsp-request:get-method/> <br/> <b>Fruit requested:</b> <xsp-request:get-parameter
name="fruit"/> </html> </xsp:page>
```

If your server is www.mydomain.com and you save this XML in your Cocoon document tree as /cocoon/request.xml, then you can see the effect of the *request* elements by opening your browser with the URL http://www.mydomain.com/cocoon/request.xml?fruit=apple

The output should look something like:

Request Method: GET

Fruit requested: apple

4. XSP Interactions

The Request logicsheet tags may be used interchangeably with XSP code that directly uses the request object. The request object is an instance of the HttpServletRequest class, and is available inside the

user element in an XSP page. The Request logicsheet itself uses this object. Therefore, the following code snippets function essentially the same:

Using the Request logicsheet: <xsp:page xmlns:xsp="http://apache.org/xsp" xmlns:xsp-request="http://apache.org/xsp/request/2.0" > <page> <fruit><xsp-request:get-parameter name="fruit"/></fruit> </page> </xsp:page> **Using the request object:** <xsp:page xmlns:xsp="http://apache.org/xsp" > <page> <fruit><xsp:expr>request.getParameter("fruit")</xsp:expr></fruit> </page> </xsp:page>

You may freely mix Request elements with other XSP Java code. Many, if not most, Request elements result in String objects. Thus, the following code fragment is valid:

<xsp:logic> String fruit = <xsp-request:get-parameter name="fruit"/>; if (fruit != null) { fruit = fruit.toUpperCase(); } </xsp:logic> <fruit><xsp:expr>fruit</xsp:expr></fruit>

5. Elements Reference

All Request elements which require or allow for additional information allow you to provide the information as either an attribute or a child element. These attributes/child elements are listed in the "Attributes/Child Elements" column of the table below. Unless noted, these are required for the given element; their absence will result in Java compilation errors or exceptions.

The following fragments are equivalent:

<xsp-request:get-parameter name="fruit"/> <xsp-request:get-parameter> <xsp-request:name>fruit</xsp-request:name> </xsp-request:get-parameter>

All Request elements which get data from the request can output the data in two ways. The as attribute of the element is used to switch between the different output options. The choice is always between some default value for as and the value "node". Using the default value for as (which is most easily achieved by leaving out the attribute altogether), the Request element will put the result of its operation in an <xsp:expr> node. This allows you to use the result in a Java expression, or converts it to text in the output DOM tree. If you use as="node", however, the output is embedded in a node or nodes, as appropriate. For instance, the following code fragment:

<xsp-request:get-parameter as="xml" name="fruit"/>

results in output similar to:

<xsp-request:parameter name="fruit">apple</xsp-request:parameter>

This is especially useful with elements that return multiple pieces of information, such as xsp-request:get-parameter-values. Without using as="node", the returned values are written out end to end without separation. If node output is requested, however, each value is written out in a separate node, which may then be referenced separately.

The elements which provide for node output are marked with a "yes" in the "Node?" column of the table below. Unlike the other attributes used in Request elements, as cannot be supplied as a child element; it must be supplied as an attribute, if it is used at all.

Since these elements are primarily wrappers around HttpServletRequest methods, the HttpServletRequest documentation in the Java Servlet API Specification, version 2.2 is also helpful in understanding the behavior and usage of these elements.

Element Name	Attributes/Child Elements	Node?	Description
xsp-request:get-attribute	name	yes	Gets a named attribute set by the servlet container or by a

			previous xsp-request:set-attribute operation.
xsp-request:get-attribute		yes	Gets the names of all available request attributes.
xsp-request:get-auth-typ		yes	Gets the name of the authentication scheme used to protect this request location, if used, e.g., BASIC or SSL.
xsp-request:get-characte		yes	Gets the name of the character encoding used in the body of this request.
xsp-request:get-content-		yes	Gets the length, in bytes, of the request body, or -1 if the length is unknown.
xsp-request:get-content-		yes	Gets the MIME type of the body of the request.
xsp-request:get-context-		yes	Gets the portion of the request URI that indicates the context of the request.
xsp-request:get-cookies		yes	Gets all cookie objects supplied by the client with this request.
xsp-request:get-date-hea	name, format *(optional)*	yes	Gets the value of the named request header that represents a date. Use this method with headers that contain dates, such as If-Modified-Since. The as attribute for this element may be "long" (default), "string", "date", or "node". If "long", the returned value is a Java long that represents a Java Date value. If "date", the returned value is a Java Date object. If "string" or "node", the optional format attribute may be used to supply a format string for a Java SimpleDateFormat to

			format the resulting string.
xsp-request:get-header	name	yes	Gets the string value of the named request header.
xsp-request:get-headers	name	yes	Gets all values of the named request header.
xsp-request:get-header-		yes	Gets the names of all available request headers.
xsp-request:get-int-head	name	yes	Gets the value of the named request header which represents an integer, or -1 if the named header doesn't exist. The as attribute may be set to "int" (default), "string", or "node".
xsp-request:get-locale		yes	Gets the preferred locale for the client browser, or the default server locale if not provided by the client.
xsp-request:get-locales		yes	Gets the locales accepted by the client in order of preference.
xsp-request:get-method		yes	Gets the name of the method associated with this request, e.g., GET, POST, or PUT.
xsp-request:get-paramet	name	yes	Gets the value of the named request parameter. This is a value from the request string (e.g., ?fruit=apple) or from POSTed form data. If the parameter has more than one value, (e.g, ?fruit=apple&fruit=orange), then this gets the first value. See xsp-request:get-parameter-values. Possible attributes: form-encoding (depends on the encoding of the page which sends the form data) and container-encoding (default per servlet

			spec: ISO-8859-1 but if your servlet container uses another one you can adjust)
xsp-request:get-paramet		yes	Gets the names of all the request parameters.
xsp-request:get-paramet	name	yes	Gets all values for the named request parameter.
xsp-request:get-path-info		yes	Gets any additional path information supplied by the client with this request.
xsp-request:get-path-tra		yes	Gets any additional path information after the servlet name but before the query string, translated to a real path.
xsp-request:get-protocol		yes	Gets the name and version of the protocol the request uses, for example, HTTP/1.1.
xsp-request:get-query-st		yes	Gets the query string for this request (e.g., "?fruit=apple&bread=rye").
xsp-request:get-remote-		yes	Gets the IP address of the requesting client.
xsp-request:get-remote-		yes	Gets the fully-qualified name of the requesting client, or the IP address if the name cannot be determined.
xsp-request:get-remote-		yes	Gets the login name of the user making the request, if a user has been authenticated.
xsp-request:get-requeste		yes	Gets the session id contained in the request.
xsp-request:get-sitemap		yes	Gets the part of the request URL that was matched in the sitemap.
xsp-request:get-requeste		yes	Returns the full request URL including server name and port.
xsp-request:get-scheme		yes	Gets the name of the

			scheme used in this request, e.g., http or https.
xsp-request:get-server-n		yes	Gets the name of the server that received the request.
xsp-request:get-server-p		yes	Gets the port on which the request was received, e.g., 80 or 443.
xsp-request:get-servlet-p		yes	Gets the part of the request URL that calls the servlet.
xsp-request:get-session-		no	Gets a given attribute of a session.
xsp-request:get-session-		yes	Gets the id of the session.
xsp-request:get-user-pri		yes	Gets a java.security.Principal object containing the name of the user, if a user has been authenticated.
xsp-request:get-uri		yes	Gets the part of the request URL after the server address and port, up to the query string.
xsp-request:is-secure		yes	Indicates whether the request was made using a secure protocol such as HTTPS.
xsp-request:is-user-in-ro	role	yes	Indicates whether the authenticated user is a member in the named role.
xsp-request:remove-attri	name	no	Removes the named attribute from the request.
xsp-request:set-attribute	name	no	Sets the named attribute to the value represented by any children of the element. If the element has a text node as its child, the attribute will be set to the String containing the text.

1. Comments

add your comments