

Session Logicsheet (2.1 legacy document)

Table of contents

1 Comments.....	6
-----------------	---

Table of contents

1 Description.....	3
2 Usage.....	3
3 Example Code.....	3
4 XSP Interactions.....	4
5 Elements Reference.....	4

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 Session logicsheet (taglib) is an XSP logicsheet that wraps XML tags around standard session operations. Specifically, the Session logicsheet provides an XML interface to most methods of the `HttpSession` object (see the [Java Servlet API Specification, version 2.2](#)) for more information.

Each client gets its own unique session, which is created the first time it accesses a page which creates or retrieves a session (see Usage, below). This session is identified by a unique id, which is generated by the servlet server and is passed back and forth to the client either as a cookie or as a parameter in the request query string.

The Session logicsheet may be used to retrieve information about the session itself, such as its creation time, and it may be used to store and retrieve information in the session, such as a login name. The session will typically become invalid after some length of time, releasing the stored information. You can query whether a session is new and how long it will remain valid between client requests, and you can also set how long the session should remain valid.

2. Usage

As an XSP logicsheet, the Session 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 Session logicsheet, you must first declare the *session* namespace, mapping it to the uri `http://apache.org/xsp/session/2.0`. Also, to ensure that you have a session to work with, you must set the create-session attribute in the `xsp:page` element to true. This will retrieve the existing session, or create a new one if the current one is invalid or doesn't exist. These steps will result in code like this:

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

You may then use any of the elements in the *session* namespace described in the [Elements Reference](#) section below.

3. Example Code

The following code shows an example of using the Session logicsheet. This code stores a value in the session under the name "fruit", then retrieves it into the output. It also retrieves the creation time of the session as a String. Of course, rather than displaying the retrieved 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-session="http://apache.org/xsp/session/2.0" create-session="true"> <html>
<xsp-session:set-attribute name="fruit">Apple</xsp-session:set-attribute> <b>Your fruit
is:</b> <xsp-session:get-attribute name="fruit"/> <br/> <b>Your session was created:</b>
<xsp-session:get-creation-time as="string"/> </html> </xsp:page>
```

The output of this page should look something like:

Your fruit is: Apple

Your session was created: Wed Jun 13 17:42:44 EDT 2001

4. XSP Interactions

The Session logicsheet tags may be used interchangeably with XSP code that directly uses the session object. The session object is an instance of the HttpSession class, and is available inside the user element in an XSP page, if the create-session attribute of the xsp:page element has been set to true. The Session logicsheet itself uses this object. Therefore, the following code snippets function essentially the same:

Using the Session logicsheet: `<xsp:page xmlns:xsp="http://apache.org/xsp" xmlns:xsp-session="http://apache.org/xsp/session/2.0" create-session="true"> <page> <xsp-session:set-attribute name="fruit">Apple</xsp-session:set-attribute> <fruit><xsp-session:get-attribute name="fruit"/></fruit> </page> </xsp:page>` **Using the session object:** `<xsp:page xmlns:xsp="http://apache.org/xsp" xmlns:xsp-session="http://apache.org/xsp/session/2.0" create-session="true"> <page> session.setAttribute("fruit", "Apple"); <fruit><xsp:expr>session.getAttribute("fruit")</xsp:expr></fruit> </page> </xsp:page>`

You may freely mix Session elements with other XSP Java code, thus:

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

5. Elements Reference

All Session 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-session:get-attribute name="fruit"/>
<xsp-session:get-attribute><xsp-session:name>fruit</xsp-session:name></xsp-session:get-attribute>
```

All Session elements which get data from the session 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 Session 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-session:get-attribute as="node" name="fruit"/>
```

results in output similar to:

```
<xsp-session:attribute name="fruit">apple</xsp-session:attribute>
```

This is especially useful with elements that return multiple pieces of information, such as xsp-session:get-attribute-names. 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 Session 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 HttpSession methods, the HttpSession 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-session:get-attribute	name	yes	Gets the value of the named attribute stored in the session.
xsp-session:get-attribute		yes	Gets the names of all attributes stored in the session.
xsp-session:get-creation		yes	Gets the time when this session was created. The as attribute for this element may be "long" (default), "string", or "node". If "long", the returned value is a Java long that represents a Java Date value. If "string", the value is converted to a String representation of the date, e.g., "Wed Jun 13 15:57:06 EDT 2001". If "node", the long value is output in the output node.
xsp-session:get-id		yes	Gets the session id, generally a randomly generated string (server dependent).
xsp-session:get-last-acc		yes	Gets the last time this session was accessed (the last time a page was requested using this session id). The as attribute for this element may be "long" (default), "string", or "node". If "long", the returned value is a Java long that represents a Java Date value. If "string", the value is converted to a String representation of the date, e.g., "Wed Jun 13 15:57:06 EDT 2001". If "node", the long value is output in the output node.
xsp-session:get-max-ina		yes	Gets the minimum

			time, in seconds, that the server will maintain this session between client requests.
xsp-session:invalidate		no	Invalidates the current session. Any attributes stored in the session are lost.
xsp-session:is-new		yes	Indicates whether this session was just created.
xsp-session:remove-attr	name	no	Removes the named attribute from the session.
xsp-session:set-attribute	name	no	Stores a named attribute in the session. Place the value to be stored as the text contents of this element, e.g., <xsp-session:set-attribute name="fruit">apple</xsp-session:set-attribute>
xsp-session:set-max-ina	interval	no	Set the minimum time, in seconds, that the server should maintain the current session between client requests.
ignorethisitisjusttoprevent			

1. Comments

add your comments