

USP SECURE ENTRY SERVER®



UNITED SECURITY PROVIDERS

Documentation series

Secure Login Service

JSP Taglib Guide

Version 5.19.0.4



United Security Providers AG
www.united-security-providers.ch
info@united-security-providers.ch

Headquarter Stauffacherstrasse 65/15 CH-3014 Bern Tel. +41 31 959 02 02
Baslerpark Mürtschenstrasse 27 CH-8048 Zürich Tel. +41 44 496 61 11



UNITED SECURITY PROVIDERS

Copyright © 2024 United Security Providers AG

This document is protected by copyright under the applicable laws and international treaties. No part of this document may be reproduced in any form and distributed to third parties by any means without prior written authorization of United Security Providers AG.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESSED OR IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED TO THE EXTENT PERMISSIBLE UNDER THE APPLICABLE LAWS.



Contents

1 JSP Tag Library	1
1.1 All tags	1
1.2 <sls:button>	2
1.3 <sls:doJexl>	2
1.4 <sls:doScript>	3
1.5 <sls:form>	3
1.6 <sls:getCred>	3
1.7 <sls:getJexl>	4
1.8 <sls:getScript>	4
1.9 <sls:ifDebug>	4
1.10 <sls:ifHasGlobalError>	4
1.11 <sls:ifJexl>	5
1.12 <sls:ifModel>	5
1.13 <sls:ifNotSesCredential>	5
1.14 <sls:ifScript>	6
1.15 <sls:ifSesCredential>	6
1.16 <sls:inputField>	6
1.17 <sls:logoutHsp>	7
1.18 <sls:mandatoryFooter>	8
1.19 <sls:mandatoryHead>	8
1.20 <sls:message>	8
1.21 <sls:metaRefresh>	9
1.22 <sls:pwdRules>	9
1.23 <sls:reqInfoMsg>	9
1.24 <sls:selection>	9
1.25 <sls:whileJexl>	10
1.26 <sls:whileScript>	10



Chapter 1

JSP Tag Library

1.1 All tags

The attributes listed here are available in most tags of this tag library (there may be some exceptions). In order to avoid documenting them for every tag again, they are documented once here.

- **parameter:** id

The id of this HTML element in the page.

- **parameter:** styleClass

The CSS style class for the button (or link).

- **parameter:** style

The style of a surrounding HTML span tag.

- **parameter:** title

The title of a surrounding HTML span tag. A span title can be used to trigger hover-pop-ups.

- **parameter:** surroundingTag

The name (e.g. "span" or "div") of a conditional surrounding tag. With this attribute, the resulting string can be forced to be a HTML tag like *span*, containing the output of the given SLS JSP tag as its content.

- **parameter:** eventhandlers

Set event type and java script function in same attribute, split by minus "-" character. For example, `eventhandler="onclick-myClickFunkction()`.

- **parameter:** additional

This optional attribute can be used to set any additional attribute for the rendered result tag. Quoting of the value of the result attribute must be done by hand, even no value is possible. Some `<code>example</code>` special stuff here.

- **since:** SLS 4.0.0



1.2 <sls:button>

Creates HTML buttons or HREF links for submitting forms, loading a URL, logging out etc.

- **parameter:** name

Adds a "name" attribute to the button tag.

- **parameter:** backFunction

If the "slsFunction" has been set to "back", this optional attribute can specify a JavaScript function that should be invoked instead of the pre-defined "back"-behaviour of the button tag.

- **parameter:** formName If

this button tag is used outside of an SLS form (aka a form generated with the "sls:form"-tag), this attribute must be set with the name of the surrounding form, or the button tag will display an error. The reason is that the tag will generate an invocation of a JavaScript function for submitting the form, and that JavaScript function requires the name of the form.

- **parameter:** redirectUrl

The url where the tag redirects to, if function="REDIRECT".

- **parameter:** value

The value attribute (for input tags). If it does not start with a backslash "\", it is interpreted as a key in the language resource files. So, to display a localized text label on a button, define a message with that label in the message resource file (sls-resources_*.properties), and set the key of that message as the "value" attribute of this tag.

- **parameter:** buttonType

The type of the resulting HTML tag: "button"|"link"|"input"

- **parameter:** function

Defines the functionality of the button (or link). Note that some functions are only available for either a button or a link; if the tag is configured in an invalid way, it will either use the appropriate button type automatically, or display an error message. <p>

1.3 <sls:doJexl>

DEPRECATED: Use "doScript" instead.

- **parameter:** expression

The JEXL/Groovy expression.

- **since:** SLS 4.0.0



1.4 <sls:doScript>

Tag which evaluates a JEXL/Groovy expression and returns nothing (no output). This tag is useful for executing JEXL functions without having any impact on the page, like setting custom response headers etc.

- **parameter:** expression

The JEXL/Groovy expression.

- **since:** SLS 4.0.0

1.5 <sls:form>

Tag which creates a HTML form. This tag creates a HTML form which includes some hidden fields required by the SLS in most cases. Also, it defines a form name that is in turn required by some of the tags included in the form.

- **parameter:** name

The name of the form. Any SLS tag included in this form will automatically know the name of this form, which eliminates the need to define the form name in all those tags separately.

- **parameter:** action

The name of the corresponding form action. If this attribute is not set, the URI by which the current model was triggered will be used automatically. Also, if the URI which triggered the current model is different than the value entered in this attribute, the model URI will be used. This allows to use the same JSP in different models that are triggered by different URIs. Otherwise, if the value was fixed, POSTing a request from the form would trigger the model defined by that URI, potentially falling from one model into another one. NOTE: The configuration property "form.use.dynamic.actions" must be set to "true" to enable this dynamic URI generation.

- **parameter:** forceAction

If this attribute is set instead of the regular action attribute, the action of the form will get this value, no matter what URI the model in the current session was triggered with. The regular "action" attribute, on the other hand, will be influenced by the current model.

- **since:** SLS 4.0.0

1.6 <sls:getCred>

Tag which returns the String value of a valid credential, or the empty String. (if there was no valid credential of the given type)

- **parameter:** credentialType

The semantic type of the credential.

- **since:** SLS 4.0.0



1.7 <sls:getJexl>

DEPRECATED: Use "getScript" instead.

- **parameter:** expression

The JEXL/Groovy expression.

- **since:** SLS 4.0.0

1.8 <sls:getScript>

Tag which evaluates a JEXL/Groovy expression and returns a result string.

- **parameter:** expression

The JEXL/Groovy expression.

- **since:** SLS 5.3.0.0

1.9 <sls:ifDebug>

Tag which evaluates the body only if the configuration property "debug.info" is enabled.

- **since:** SLS 4.0.0

1.10 <sls:ifHasGlobalError>

Show nested content only if there is a global error message to display. This allows to wrap the global error message into some elaborate HTML code which will not show up as long as there is no error message at all.

- **parameter:** invert

This inverts the result. Set *invert=true* to display the tag body only when there is no global error (in other words, this works like the "else" case).

- **parameter:** skipIfEmpty

If the message key of the global error message is not mapped to a message or mapped to an empty message, this tag normally evaluates the body. Set *skipIfEmpty=true* to skip the body in this case. Usage of this attribute is not recommended. Better use error mapping to the special key "do.not.display" .

- **since:** SLS 4.0.0



1.11 <sls:ifJexl>

DEPRECATED: Use "ifScript" instead.

- **parameter:** condition

The JEXL/Groovy expression, e.g. `${...}` or `#{...}`. Automatically wrapped in `${...}` if contains neither JEXL nor Groovy expressions.

- **since:** SLS 4.0.0

1.12 <sls:ifModel>

The body of the tag is evaluated only if the current state of the SLS login model accords to all specified attributes.

- **parameter:** state

The name of the current model state.

- **parameter:** stateNumber

The numeric value of the current model state.

- **parameter:** property

The value of the current model state property.

- **parameter:** modelName

The name of the current model.

- **since:** SLS 4.0.0

1.13 <sls:ifNotSesCredential>

The "SES session" refers to the session of the HSP reverse proxy (and, by that, to the actual connection between the browser client and the HSP). That session always has a certain state, like "valid" after a successful authentication. This tag allows to display HTML content in a page depending on the "SES session" NOT being in a certain state.

- **parameter:** state

Defines the value that the SES session state must NOT be in, in order to display the tag contents. Example: "valid"

- **since:** SLS 4.0.0



1.14 <sls:ifScript>

Evaluate the tag body depending on a JEXL/Groovy expression result.

- **parameter:** condition

The JEXL/Groovy expression, e.g. `${...}` or `#{...}`. Automatically wrapped in `${...}` if contains neither JEXL nor Groovy expressions.

- **since:** SLS 4.0.0

1.15 <sls:ifSesCredential>

The "SES session" refers to the session of the HSP reverse proxy (and, by that, to the actual connection between the browser client and the HSP). That session always has a certain state, like "valid" after a successful authentication. This tag allows to display HTML content in a page depending on the "SES session" being in a certain state.

- **parameter:** state

Defines the value that the SES session state must be in, in order to display the tag contents. Example: "valid"

- **since:** SLS 4.0.0

1.16 <sls:inputField>

This tag displays the input field for a credential. It enables special behaviour when the credential already exists, or when the user did not fill in the field last time.

- **parameter:** placeholder

The optional *placeholder* input-field attribute (HTML 5).

- **parameter:** credentialType

The semantic type of the credential, like *username*, *password* or *secret*.

- **parameter:** displayError

Defines if an error message is displayed next to the input field, when user enters an invalid value. Must evaluate to either *true* or *false*. Defaults to *true*.

- **parameter:** size

The size of the input field. Must evaluate to a number.

- **parameter:** maxLength

The maximum length for input entered in the field. Must evaluate to a number.



- **parameter:** hideTyping

If set to *true*, the input is not shown to user when typing, but only stars *. (HTML input type *password*). If set to *false*, the input is shown to the user when typing. Default, if not set: the input is only hidden for *credentialType=password*.

- **parameter:** errorStyle

The style-class of the error message. (Shown if the field was not filled correct.)

- **parameter:** presetStyle

The CSS style-class if the credential was preset and its value is written.

- **parameter:** hidelfPreset

If set to *true*, and the credential is still set, the field is totally skipped. We return an empty String in this case. Default: *false* → We show the credential's string value.

- **parameter:** overwritelPreset

If set to *true*, an already existing credential of the same type is removed from the SLS session when this tag is rendered, even if it was already verified. This feature cannot be used together with *hidelfPreset*.

- **parameter:** ifModelState

If set, the tag only returns a result if the current model state equals the one defined in this attribute. The check for this string value is not case-sensitive, so *get.cred* equals *Get.Cred*.

- **parameter:** ifModelProperty

If set, the tag only returns a result if the current model state has a property whose value is equal to the given string. e.g. In model state 4 *get.cred.token* you may define the property "doit". You can map the state to the *Login.jsp* and configure the third input field with *ifModelProperty="doit"*. This field is then only shown in the 3rd model state, but not in the first *get.cred* step, which uses the same jsp. The check for this string value is not case-sensitive, so *doit* equals *DOit*.

- **parameter:** parameterName

The name of the post parameter. If set, this value overwrites the default from *CredentialSemanticType*.

- **parameter:** tabindex

The html option tab index of the created field. Must evaluate to a number. Not set by default.

- **since:** SLS 4.0.0

1.17 <sls:logoutHsp>

Logs out a user from the current or all sessions by invalidating the HSP session.

- **parameter:** scope

If the scope attribute is set to "all", the user will be logged out of all active sessions, else only of the current session (in an SSO environment).

- **since:** SLS 4.0.0



1.18 <sls:mandatoryFooter>

Inserts table(s) with either debug information about the last processed request, some SLS internal state information etc., if "error.details" is set to "false". If it is set to "true", the tag instead checks if there are production-critical configuration settings and displays a table with appropriate warnings if necessary. Furthermore, the tag also inserts any SLS custom JavaScript functions etc. that may be needed at the bottom of the page. For this reason, this tag must ALWAYS be present in all the JSPs.

- **parameter:** formName

The name of the form in the page (defaults to "LoginForm"). If BID check functionality is used (configuration property "bid.check.enable" is true in "sls.properties"), this name of the form in the JavaScript code rendered by this tag must be equal to the actual name of the form in the current JSP. If that is not "LoginForm", set the name with this attribute.

- **since:** SLS 4.0.0

1.19 <sls:mandatoryHead>

This tag generates various header elements in the page required by the SLS for some features to work correctly (like BID-check functionality). It does not, however, generate any actual HTML in the page, so it is recommended to always include this tag in the "HEAD"-section of all SLS pages (except for the "SystemError.jsp").

- **parameter:** noScriptUrl

Allows to define a URL that should automatically be loaded through a META REFRESH tag if the current browser client does not support JavaScript. That URL should point to a HTML page explaining to the user that JavaScript is required to access the application.

- **parameter:** bidJsPath

If BID check is enabled, this attribute allows to define the URI path of the JavaScript file containing the BID-related functions. Defaults to "/sil-bid-check/ses.js".

- **since:** SLS 4.0.0

1.20 <sls:message>

Displays a localized message, based on the SLS resource files and the current session locale. There are also special keys, however (see "key" attribute for details).

- **parameter:** key

Usually the key of the resource property with the message to display. There are two special key values, however: "browser.blacklist.msg" to display the browser-blacklist description text, and "global.error.msg" to display the global error (the one to be displayed below the form input fields).

- **parameter:** innerReplacement

The message stored under "key" can have inner Multiple values can be provided by separating them with a pipe character ("|"). Automatically wrapped in \${...} if contains neither JEXL nor Groovy expressions.

- **since:** SLS 4.0.0



1.21 <sls:metaRefresh>

Creates a META refresh tag in the HEAD section, either in a general fashion with configurable reload URL, or specifically for the slowdown JSP.

- **parameter:** path

Defines the reload path. May be either a relative path or a fully qualified URL. Either this attribute or *slsPath* must be set.

- **parameter:** slsPath

Defines the SLS action to load, such as */auth*. Either this attribute or *path* must be set.

- **parameter:** time

Defines the time to wait before the reload of the page is performed. Usually this should be a numeric value, defining the number of seconds for the timeout. NOTE: For the Slowdown.jsp, this value must be set to *slowdown*. This is the only non-numerical value allowed for this attribute.

- **since:** SLS 4.0.0

1.22 <sls:pwdRules>

Tag for rendering the password policy requirements.

- **since:** SLS 4.0.0

1.23 <sls:reqInfoMsg>

Tag for rendering the optional "infomsg" field content from the "req" parameter, as sent during processes like transaction verification etc.

- **since:** SLS 4.0.0

1.24 <sls:selection>

This tag generates a selection for the available tokens. In the body can be button tags, that are inserted at the end.

- **parameter:** name

The name of the input selection.

- **parameter:** listKey

The key with them the list is stored in the SLS session.

- **parameter:** type



Valid types: radio, list, links. "radio": a list of radio buttons (submit button needed), "list": a standard selection list (submit button needed), "links": a list of links (no submit button needed)

- **parameter:** innerStyleClass

The CSS style class of a selectable item.

- **parameter:** innerStyle

The CSS style of a selectable item.

- **parameter:** innerEventHandlers

Eventhandlers of a selectable item.

- **since:** SLS 4.0.0

1.25 <sls:whileJexl>

DEPRECATED: Use "whileScript" instead.

- **parameter:** condition

The JEXL/Groovy expression, e.g. `${...}` or `#{...}`. Automatically wrapped in `${...}` if contains neither JEXL nor Groovy expressions.

- **since:** SLS 4.0.0

1.26 <sls:whileScript>

The body of the tag is evaluated as long as the JEXL/Groovy expression evaluates to *true*.

- **parameter:** condition

The JEXL/Groovy expression, e.g. `${...}` or `#{...}`. Automatically wrapped in `${...}` if contains neither JEXL nor Groovy expressions.

- **since:** SLS 4.0.0