WSSO scripting language

Scripting language

The scripting language is fully compatible with ECMAScript 3rd edition plus a small set of new objects and methods.

Before running the script, Soffid WSSO will create some global variables referring to the request of the user agent and content generated by the application server.

Here are the functions and classes implemented:

Global Functions

debug	text: string	Sends a message to the apache log file
env	text: string returns string	Gets the value of an environment variable
logout		Log out of Web Single Sign on session. Any further access to a protected resource will ask the user for credentials.

Global objects

request	Contains a **Request** object referring the request made by the user agent.
response	Contains a **Response** object referring the data generated by the application server
secretStore	Contains the secrets of the user, including account names and passwords.
document	Contains a **Document** object referring the generated page.

secretStore object

The object is available from authenticated requests and provides access to the secrets of the user. Any usage of secretStore object on an anonymous request will produce an undefined value

getSecret	text: string returns string	Gets the value of a secret
getAccounts	system: string return: array	Returns the list of accounts that the user is allowed to use on a particular system
getAccounts	system: string returns string	Returns the first account that the user is allowed to use on a particular system.
getPassword	system: string account: string returns string	Returns the actual password for the specifieds account and system.

Request class

The Request class encapsulates the request made by the browser and forwarded to the application server.

Attributes

method	string	Used method (POST, GET or others)
url	string	Contains the requested URL, including scheme, host and port
headers	Array	Headers sent with the request.
content	String	Data posted by the user agent on POST requests. Due to memory usage limits, its content may not be complete.
params	Array	Array containing the request parameters.

Methods

clone	returns Request	Create a new request from the current. The current response will be canceled and a new request will be processed

Response class

The class encapsulates the request response sent by the server application.

Attributes

document	Document	Document generated by the server
headers	Array	Vector header with the request

Document class

Contains the page generated by the application server. Thus, scripts can access the website and its DOM tree in runtime. The document object implements a subset of the standard DOM HtmlDocument.

Attributes

url	string	Full URL of the document.
title	string	Title of document.
cookie	string	Contains the cookies sent by the user agent
anchors	Collection	Contains elements with tag A.
forms	Collection	Contains elements with tag FORM.
images	Collection	Contains elements with tag IMG.
links	Collection	Contains elements with tags A and AREA.
documentElement	Item	Contains the root element, usually and HTML tag element.
content	string	Full content of the document in string format

Methods

getElementById	id: string Element returns	Find the first element with the specified ID
getElementsByTagName	tag: string returns Collection	Find all elements with the specified tag

Element class

Obtained from the document object, implements a subset of the DOM class HtmlElement.

Attributes

childNodes	Collection	The vector containing children elements
disabled	boolean	Indicator of whether the element is disabled or not
id	string	The id attribute of the element
tagName	string	Tag of the element
parentNode	Item	Parent element

Methods

getAttribute	name: string returns string	Returns the value of the specified attribute name.
setAttribute	name: string value: string	Ses the value of the specified attribute.
removeAttribute	name: string	Remove the specified attribute
setText	value: string	Changes the inner content of the element
getText	returns string	Gets the element inner content.
getElementsByTagName	tag: string returns Collection	Find all children's elements with the specified tag
addChild	tag: string before: element (opt) returns element	Creates a new item with the specified tag. If a node is specified, the new element will be ordered just before the specified item in the child list

remove		Removes the node from the document

Collection class

The document object implements a subset of the standard DOM HTMLCollection

Attributes

length	Long	Number of items in the collection

Methods

item	id: long returns Element	Gets the n-position element within the collection. The first element is numbered as 0.
namedItem	id: string returns Element	Search for an item with the given name. First, search for an element with a matching id attribute. If none is found, search for an element with a matching name attribute.

Revision #2
Created 7 June 2022 14:26:22
Updated 17 October 2024 09:27:15