ScrapperMin – Web Automation Guide

Language/Software/Documentation Author : John Kenedy

Syntax : SET

Example : SET('VariableName', 'Single Value');

Example : SET('VariableName', 'Array Value 1';'Array Value2');

SET allows you to set a value or array of value into variable, the first argument is the VariableName must be enclosed by single quote, the second parameter is the string or array of string which enclosed by single quote too. Array is created by joining string with semicolon.

Syntax : GET

Example : GET('VariableName');

GET allows you to get a value from a variable name, error will occur if the VariableName does not exist, the return result will be usually passed to another method for processing such as for logging using LOG statement

Syntax : LOG

Example : LOG('Text to display');

Example : SET('VariableA', 'My Value'); LOG(GET('VariableA'));

LOG allows you to display a value, just like normal function, it can be called by passing another function that return string or array of string. If array of string is passed, the method will loop and display all the string, if a string is passed it will display only that string, second example above will display 'My Value'

Syntax : EXIT

Example : EXIT('Message to display when exit');

EXIT allows you to display a message and terminate the script.

Syntax : PASS

Example : PASS()

PASS will return null, which basically means do nothing, this is useful for IF statement that will execute true or false, if we want only to execute any of it (true/false) and the other do nothing, simply use PASS() in that area.

Syntax : IF

Example : IF('A', '=', 'A', LOG('True');, PASS());

Above will test if A equal to A, if yes it will LOG true, if false it will PASS or do nothing. Notice that IF has 5 arguments separating with comma, the first is operand 1, follow by operator, then operand 1, statement to execute when true (can be more than one statement separated by semicolon), then statement to execute when false (can be more than one statement separated by semicolon.

Supported operator

= To test equality

!= To test inequal

> To test greater than (only if the variable hold a string that can be parse into number)

< To test lesser than (only if the variable hold a string that can be parse into number)

STARTSWITH To test whether a string starts with second operand

ENDSWITH To test whether a string ends with second operand

CONTAINS To test whether a string containing a substring

IN To test whether a string is inside an array of string in the third argument

Syntax : FOR

Example : FOR('A', '1';'2';'3';'4';'5', LOG(GET('A')));

For allows to loop when the second argument is an array (remember array is string enclosed with single quote and joined by semicolon). Each loop the value will be stored in a variable name at first argument which is A in above example. The third argument can contain one or more statement separating by semicolon, above is only one statement which is LOG that is passed with GET statement to get variable A which is the current loop value. So the final result will display 12345

Syntax : COUNT

Example : COUNT('A';'B';'C');

COUNT allows you to count the length or size of an array, above A,B,C is three string joining together by semicolon forming an array, and the length is 3. Different from LENGTH statement, LENGTH statement will check the length of a string, example 'hello' has length 5.

Syntax : LENGTH

Example : LENGTH('testing');

LENGTH allows you to count the length of a string, above will return 7, if the given variable is an array, it will count the length of first string in the array and ignore the rest.

Syntax : WHILE

Example : WHILE('1', '=', '1', LOG('a'));

Above statement will test operand 1 (first argument) with third argument with second operator, the operator list is the same with what IF has, since 1 is equal to 1, the third argument will be executed when the condition is true, because it is always true so it will keep displaying a infinitely.

Example :

```
SET('A', '1');

WHILE(COUNT(GET('A')), '<=', '5',

    LOG('My Value');

    SET('A', GET('A');'1');

);
```

Above will display My Value for 5 times. First it set variable A with value 1, then it check whether the count of variable A is lesser or equal than 5. Since variable A is firstly not an array just a string, COUNT return 1, then it display My Value, then it execute another statement which is to SET variable A with the current value (GET('A') adding with another string which is '1' (remember array is generated by appending semicolon), so variable A becomes an array that contains '1';'1' which has COUNT 2. Since 2 is still lesser or equal than 5 it display My Value again and continue adding with '1' which becomes '1';'1';'1' or has COUNT 3 and continue.

Syntax : JOIN

Example : JOIN(',', '1', '2', '3');

JOIN allows to join an array of string or string in second argument until the end of the argument. Above example has 4 argument, the first is the joining separator, second to fourth argument is the string to be joined. Instead of above it can be done like below

Example : JOIN (',', '1';'2';'3');

Above will have same effect with the first example. The second example uses an array instead of series of arguments which will be joined becomes 1,2,3 too. This is useful when generating a POST string like below

Example : JOIN('', 'username=', GET('{PARAM0}'), '&password=', GET('{PARAM1}'));

Above will join series of arguments with empty string which basically concatenating them together.

Syntax : FILTER

Example : FILTER('3', '0';'1';'2';'3';'4');

FILTER allows to get only a string in a certain index. Above having first argument 3 means getting index 3 of array in second argument which will return 3, since index starts with 0. FILTER will filter out unused string in an array and getting the useful one only.

Syntax : REPLACE

Example : REPLACE("I am John", " ", "-");

REPLACE will replace a sub string with another set of sub string, above will return I-am-John

Syntax : TRIM

Example : TRIM('   I am John   ');

TRIM will remove white space in front and ending of a string so above will return I am John without the spaces. Besides spaces it will also remove character such as line break if contains in front or ending of the string.

WEB AUTOMATION SYNTAX

Web related syntax starts with WC or stand for WebClient. It is categorized for easy understanding.

Syntax : WC_MethodPage

Example : WC_MethodPage('http://www.innosia.com', 'GET', '', '');

Above will visit www.innosia.com and uses GET method (which basically GET means visiting a site, while POST is to put information to a site). First argument is the URL to visit, second argument indicating the request method (GET/POST), if GET is used third argument will be empty, third argument is the POST string to post to the website. Usually logins uses POST and third argument will be the username and password other security information to pass. The fourth argument is the request header.

Example : WC_MethodPage('http://www.innosia.com', 'POST', 'username=john&password=123456', 'Referer=http://www.innosia.com');

Above sample won't work since innosia does not accept POST on that url. The third argument is separated by & for each data to be posted, the data to be posted is a key and value separated by equal sign. The fourth argument is the request header, which is separated by & and each header contains key and value separated by equal sign.

Usually you will need to use fiddler and a browser to analyze what a website url you want to automate. You run fiddler and hook it to your browser (default), then visit the website and do login (example), fiddler will capture what information is sent to the url which is the POST string, usually website generates security number called verification token in HTML to prevent cross site scripting and you will need to POST that security number to the website back along with the username and password.

Headers are important because some website checking where the last page you visit before visiting their site, which is called referrer. Also some website requires header to specify it is requested by XMLHttpRequest, so aside of setting referrer you set another header call X-Requested-With like below

Referer=http://www.innosia.com&X-Requested-With=XMLHttpRequest

Syntax : WC_MethodPageForm

Example : WC_MethodPageForm('http://www.innosia.com', 'POST', 'username;'password', 'john';'123456', '');

If you are posting to another site that the value containing equal sign, it will have issue when you use MethodPage because MethodPage treats equal sign as separator between key and value of each data to be posted. To post to such site use WC_MethodPageForm. Third argument is an array of keys and fourth argument is an array of values. The length of keys and values must be matched or problems will occur. The fifth argument is the header just like WC_MethodPage.

Syntax : WC_MethodUploadFileForm

Example : WC_MethodUploadFileForm('http://www.innosia.com', 'POST', 'filekey', 'filepath', 'key1';'key2', 'value1';'value2', '');

WC_MethodUploadFileForm allows you to upload a binary file to a website along with other information needed. The file usually has a name which must be given in third argument and the fourth argument is the file path. The fifth is array of other keys to be posted and the sixth is array of values matches in length with the keys forming key and value pair to be posted. The seventh argument is a string

Synxtax : WC_MethodDownloadFileForm

Example : WC_MethodDownloadFileForm("http://www.innosia.com/content/latest/scrappermin/scripts/downloadkumpulbagi.txt', "GET", '', 'downloadkumpulbagi.txt', '', '', "");

Above will download a file from url in first argument, second argument indicate the file is served by GET or POST, the third argument is the path. In ScrapperMin android default path is the ScrapperMin folder which resides in root directory of SDCard called ScrapperMin, if you pass another path, it will become sub path of ScrapperMin root folder. Fourth argument is the filename, fifth argument is array of string indicating keys to be passed, the sixth argument is array of values to pass, and seventh argument is the header.

Some files are served by POST which means you need to post some information before downloads such which sites implement this for added security. You can use fiddler to analyze the information browser posted to the site which usually is key value pair, the key is gathered and generated as array in fifth argument while the values is gathered and generated as array in sixth argument.

Syntax : WC_MethodDownloadFile

Example : WC_MethodDownloadFile('http://www.innosia.com', 'GET', '', 'page.txt', '', '');

WC_MethodDownloadFile will download a file from url in first argument, second argument indicated the file is served by GET/POST method (HTTP protocol), the third argument is the path, it has the same behavior with WC_MethodDownloadFileForm and fourth argument is the POST string which is a string only that has key value pair separated by equal sign and if the pairs are more than one the rest of joined by & (same with WC_MethodPage POST string behavior)

STRING OPERATIONS SYNTAX

String operation syntax is method start with SO_ which stand for StringOps.

Syntax : SO_TagMatch

Example : SO_TagMatch('<div>aaa</div><div>bbb</div>', '<div>',</div>');

SO_TagMatch allows you to generate array of string that is enclosed in between second and third argument. Above sample will return array of string with 2 length, first string is aaa, second is bbb.

Syntax : SO_TagMatchSingle

Example : SO_TagMatchSingle('<input type="text" value="my value" name="text1"', 'name="myurl"', 'value="', '"');

SO_TagMatchSingle allows you to search of a HTML tag that contains second argument which is name="myurl" the first tag that has that value is get then it will use find string in between third and fourth argument which is value=" and " so above it will return my value.

Syntax : SO_IsInFile

Example : SO_IsInFile('filename.txt', 'line to serach');

SO_IsInFile will read a text file in first argument line by line and finds the line that has the same value with its second argument, if found it will return '1' else it will return '0'. In ScrapperMin Android version the file is reside in ScrapperMin folder in root of the sdcard.

Syntax : SO_AppendToFile

Example : SO_AppendToFile('filename.txt', 'new line');

SO_AppendToFile will add a new line in its second argument to the file in the first argument. In ScrapperMin Android version the file is reside in ScrapperMin folder in root of the sdcard. If the file does not exist it will create new one, if it already exist it will a new line to it.

Syntax : SO_ConvertToBase64

Example : SO_ConvertToBase64('MyString');

SO_ConvertToBase64 will convert a string in its first argument using base 64 encoding. This is useful for certain website that receive base 64 input.

Syntax : SO_ConvertFromBase64

Example : SO_ConvertFromBase64('TXkgU3RyaW5n');

SO_ConvertFromBase64 will decode base 64 string in first argument and return the original string.

Syntax : SO_FileExist

Example : SO_FileExist('filename.txt');

SO_FileExist will check if the file is exist or not, default folder of ScrapperMin Android version is the ScrapperMin folder in the root of your sdcard.

Syntax : SO_FileDelete

Example : SO_FileDelete('filename.txt');

SO_FileDelete will delete the filename in the first argument. Default folder of ScrapperMin Android version is the ScrapperMin folder in the root of your sdcard.

Syntax : SO_StripHTML

Example : SO_StripHTML('<a>hi</a><b>hellow</b>');

SO_StripHTML will remove all HTML syntax and return string without HTML syntax, above will return hihellow.

Syntax : SO_GetMd5Hash

Example : SO_ GetMd5Hash('password');

SO_ GetMd5Hash will hash a string using MD5 algorithm. Normally website uses hashing on password by sending the hash instead of plain text to the server. Vbulletin and most forum uses MD5 to secure the password.

Syntax : SO_UrlEncode

Example : SO_UrlEncode('http://www.innosia.com/my path');

SO_UrlEncode will encode a string using url format, such as space will become %20. Above will return 'http://www.innosia.com/my%20path'

Syntax : SO_UrlDecode

Example : SO_UrlDecode('http://www.innosia.com/my%20path');

SO_UrlDecode will decode a string from url format, above will return 'http://www.innosia.com/my path'. Url encoding and decoding is important since some website requires the input to be encoded when posted.