# Samurai

| | |
|---|---|
| **Library version:** | RENAT 0.1.8 |
| **Library scope:** | test suite |
| **Named arguments:** | supported |

## Introduction

A library provides functions to control Samurai application

The library utilize *Selenium2Library* and adds more functions to control Samurai application easily. Without other furthur mentions, all of the concepts of `user`, `user group` are Samurai concepts. By default, RENAT will try to connec to all Samurai nodes defined in active `local.yaml` at the beginning of the test and disconnect from them at the end of the test automatically. Usually user does not need to use `Connect All` and `Close` explicitly.

Currently, this module supposed that Samurai is used in Japanese locale. When Samurai module has error, it tried to make the last snapshot in `result/selenium-screenshot-x.png`. Checking this capture will help to understand the reason of the error.

Some keywords of Samurai is using `xpath` to identify elements. See *Selenium2Library* for more details about xpath.

See WebApp for common keywords of web applications and how to configure the `local.yaml` file.

*Selenium2Library* keywords still could be used together within this library. See Selenium2Library for more details.

## Shortcuts

**A**dd Policy · **A**dd Policy Group · **A**dd User · **C**apture Screenshot · **C**hange Policy View Group · **C**lick All Elements · **C**lose · **C**lose All · **C**lose Window · **C**onnect · **C**onnect All · **D**elete Policy · **D**elete Policy Group · **D**elete User · **E**dit Policy · **L**eft Menu · **L**ogin · **L**ogout · **M**ake Item Map · **R**eset Capture Counter · **S**elect Items In Table · **S**elect Window · **S**et Capture Counter · **S**et Capture Format · **S**how Detail Mitigation · **S**how Policy Basic · **S**how Policy Mitigation · **S**how Policy Mo · **S**how Policy Monitor · **S**tart Mitigation · **S**top Mitigation · **S**witch

## Keywords

<table>
<tr><th>Keyword</th><th>Arguments</th><th>Documentation</th></tr>
<tr><td>Add Policy</td><td>**policy</td><td>
Adds a new Samurai policy<br><br>
<code>policy</code> is a map containing the below information to create the new policy.
<table>
<tr><th>key</th><th>meaning</th><th>mandatory</th><th>sample</th></tr>
<tr><td>name</td><td>name of the policy</td><td>yes</td><td>test001</td></tr>
<tr><td>basic_alias</td><td>alias name of the policy</td><td></td><td>test001</td></tr>
<tr><td>basic_port_id</td><td>another alias</td><td></td><td></td></tr>
<tr><td>basic_facing</td><td><code>customer</code> or <code>backbone</code></td><td></td><td>customer</td></tr>
<tr><td>basic_intf_list</td><td>list of router and interface pair, separated by comma</td><td>yes</td><td>10.128.18.31:xe-0/0/0.1</td></tr>
<tr><td>basic_cidr_list</td><td>list of CIDR separate by comma</td><td></td><td></td></tr>
<tr><td>basic_option_filter</td><td>optinal filter</td><td></td><td></td></tr>
<tr><td>basic_direction</td><td>direction of the traffic (<code>incoming</code> or <code>outgoing</code>)</td><td></td><td>Incoming</td></tr>
<tr><td>traffic_enabled</td><td>Enable traffic monitoring or not</td><td>yes</td><td>${True}</i> or <i>${False}</td></tr>
<tr><td>detection_enabled</td><td>Enable detection or not</td><td>yes</td><td>${True}</i> or <i>${False}</td></tr>
<tr><td>mitigation_zone_name</td><td>Name of the zone for mitigation</td><td></td><td>zone001</td></tr>
<tr><td>mitigation_zone_prefix</td><td>Prefixes that could mitigate</td><td></td><td>1.1.1.1/32</td></tr>
<tr><td>mitigation_thr_bps</td><td>Upper limit (bps)</td><td></td><td>800,000,000</td></tr>
<tr><td>mitigation_thr_pps</td><td>Upper limit (pps)</td><td></td><td>54,000,000</td></tr>
<tr><td>mitigation_mo_enabled</td><td>Using Arbor TMS MO or not</td><td>yes</td><td>${True}</i> or <i>${False}</td></tr>
<tr><td>mitigation_device_list</td><td>Devices used for TMS, separated by comma</td><td></td><td>ArborSP-A</td></tr>
<tr><td>mitigation_mo_name</td><td>MO name, separated by comma</td><td></td><td>OCN12(ALU)_LOOSE</td></tr>
<tr><td>mitigation_comm_list</td><td>commna separated peer/community list</td><td>yes</td><td>1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777</td></tr>
<tr><td>nw_monitor_gre1</td><td>1st GRE address for NW monitor</td><td></td><td>210.0.1.1</td></tr>
<tr><td>nw_monitor_gre2</td><td>2nd GRE address for NW monitor</td><td></td><td>210.0.1.1</td></tr>
<tr><td>nw_monitor_ce1</td><td>1st CE address for NW monitor</td><td></td><td>210.0.1.2</td></tr>
<tr><td>nw_monitor_ce2</td><td>2nd CE address for NW monitor</td><td></td><td>210.0.1.2</td></tr>
<tr><td>nw_monitor_pe1</td><td>1st PE for NW monitor (list)</td><td></td><td>edge01hige-MX2020-15(118.23.176.244)</td></tr>
<tr><td>nw_monitor_pe2</td><td>2nd PE for NW monitor (list)</td><td></td><td>edge01hige-MX2020-15(118.23.176.244)</td></tr>
<tr><td>event_name</td><td>name of the message event to make</td><td></td><td>info1</td></tr>
<tr><td>event_addr</td><td>address to send the events</td><td></td><td>user@mail.com</td></tr>
<tr><td>view_group</td><td>user group that could view this policy, separated by comma</td><td>yes</td><td>SuperGroup,test_group_007</td></tr>
</table>
Example:
<table>
<tr><td></td><td></td><td></td></tr>
<tr><td>Samurai.Switch<br>Samurai.Add Policy</td><td>samurai-1<br>name=${POLICY_NAME}</td><td>basic_alias=${POLICY_NAME}</td></tr>
<tr><td>...</td><td>basic_facing=customer</td><td>basic_intf_list=10.128.18.31:xe-0/0/0.1</td></tr>
<tr><td>...</td><td>basic_cidr_list=1.1.1.0/24</td><td>basic_direction=incoming</td></tr>
<tr><td>...</td><td>traffic_enabled=${TRUE}</td><td></td></tr>
<tr><td>...</td><td>detection_enabled=${TRUE}</td><td></td></tr>
<tr><td>...</td><td>mitigation_zone_name=test_zone001</td><td>mitigation_zone_prefix=1.1.1.1/32</td></tr>
<tr><td>...</td><td>mitigation_device_list=ArborSP-A,ArborSP-B</td><td></td></tr>
</table>
</td></tr>
</table>

| | | |
|---|---|---|
| ... | mitigation_mo_enabled=${TRUE} | |
| ... | mitigation_mo_name=N000000012_LOOSE | |
| ... | mitigation_comm_list=1.10(180.0.1.10)/2914:666,1.11(180.0.1.11)/2914:777 | |
| ... | event_name=test | event_addr=user@mail.com |
| ... | view_group=SuperGroup | |

| | | |
|---|---|---|
| **Add Policy Group** | *group_name*, *policy_list=**, *limit_bps=4000000000*, *limit_pps=2700000* | Add a new policy group<br>`group_name` is the name of the new group. `policy_list` is a comma separated of existed policy that should be bound to this policy. An asterisk for this parameter (`*`) means *all of the existed policy*. `limit_bps` and `limit_pps` are the mitigation capacity threshold of this group. |
| **Add User** | *group*, ***user_info* | Adds user to the current group `user_info` is a dictionary contains user information that has following keys: `name`, `password`, `privilege` and `policy`<br>`privilege` is existed privilege that has been created (e.g: *system_admin*.<br>`policy` could be `*` for all current policies or a list of policy names that are binded to this user.<br>`group` is the user group. `Dot(.)` means current group<br>Examples:<br>Samurai.*Add User* OCNDDoS name=user000 password=Test12345678<br>... privilege=system_admin policy=*<br>Samurai.*Add User* OCNDDoS username=user001 password=Test12345678<br>... privilege=system_admin policy=OCN11,OCN12 |
| **Capture Screenshot** | *filename=None*, *extra=* | Captures the current screen to file<br>Using the internal counter for filename if `filename` is not specified. In this case, the filename is defined by a pre-set format. *Set Capture Format* could be used to change the current format.<br>An extra information will be add to the filename if `extra` is defined<br>Examples:<br>Samurai.*Capture Screenshot* # samurai_0000000001.png<br>Samurai.*Capture Screenshot* extra=_list # samurai_0000000002_*list*.png<br>Arbor.*Capture Screenshot* # arbor_0000000001.png<br>Arbor.*Capture Screenshot* extra=_xxx # arbor_0000000001_*xxx*.png<br>Samurai.*Capture Screenshot* filename=1111.png # 1111.png |
| **Change Policy View Group** | *name*, **group_name* | Changes the groups that could see this policy<br>`name` is the policy name. `group_name` is a list of policies<br>Example:<br>Samurai.*Change Policy View Group* super_admin test_group001 |
| **Click All Elements** | *xpath* | Click all element in current page defined by `xpath`<br>Returns the number of elements that have been clicked |
| **Close** | | Closes the current active browser |
| **Close All** | | Closes all current opened applications |
| **Close Window** | | Closes the current window |
| **Connect** | *app*, *name* | Opens a web browser and connects to application and assigns a `name`.<br>If not defined in `local.yaml` those following key will have defaut values:<br>browser firefox optional<br>login_url / optiona<br>proxy: optional<br>http: 10.128.8.210:8080 optional<br>ssl: 10.128.8.210:8080 optional<br>socks: 10.128.8.210:8080 optional<br>profile_dir ./config/samurai.profile optional |
| **Connect All** | | Connects to all applications defined in `local.yaml`<br>The name of the connection will be the same of the *webapp* name |
| **Delete Policy** | **policy_names* | Deletes poilcies by their names<br>Returned the number of deleted users<br>**Notes:** If the policy does not exists, the system will not report any error.<br>Examples:<br>Samurai.*Delete Policy* test001 test002 |
| **Delete Policy Group** | **group_list* | Deletes policy groups<br>Returns the number of deleted policy groups Example:<br>Samurai.*Delete Policy Group* test_group001 test_group002 |
| **Delete User** | *group*, **user_list* | Deletes user from the user group<br>`group` is the user group. And `.` means current group Returns the number of deleted users<br>Examples:<br>Samurai.*Delete User* SuperGroup user001 user002<br>Samurai.*Delete User* . user002 |
| **Edit Policy** | ***policy* | Edits a Samurai policy |

| | | |
|---|---|---|
| | | `policy` contains information about the policy. See *Add Policy* for more details about `policy` format |
| **Left Menu** | *menu*, *locator=None*, *ignore_first_element=True* | Chooses the left panel menu by its displayed name<br>When `locater` is not null, the keyword will return a list of text attribute of all elements specified by the `locator`. `locator` could be a xpath or a predefined string.<br>`locator` predefined strings are: `MITIGATE_REALTIME`, `MITIGATE_LIST`, `DETECT_LIST`<br>For example, a xpath `//div[@id='infoareain2']/*//td[1]/a` means the list of *link* of all elements in a 1st column of a table insides a `div` with id `infoareain2`.<br>Examples:<br>Samurai.*Left Menu* \| Traffic<br>Samurai.*Left Menu* \| Detection<br>Samurai.*Left Menu* \| ポリシー管理<br>@{LIST}= \| Samurai.*Left Menu* \| Active Mitigation \| //div[@id='infoareain2']/*//td[1]/a |
| **Login** | | Logs-in into the application<br>User and password is set by the template and authentication methods in the master files |
| **Logout** | | Logs-out the current application, the browser remains |
| **Make Item Map** | *xpath* | Makes a item/webelement defined *xpath*<br>The map is a dictionary from *item* to the *WebElement* Items name found by `xpath` are used as keys |
| **Reset Capture Counter** | | Resets the counter of the screen capture |
| **Select Items In Table** | *xpath*, *xpath2*, **item_list* | Checks items in Samurai table by xpath<br>`xpath` points to the column that used as key and `xpath2` is the relative xpath contains the checkbox column.<br>`item_list` is a list of item that need to check. Item in the list could be a regular expresion with the format `reg=<regular expression>`.<br>The keyword is called with assuming that the table is already visible.<br>Returns the tupple of all items and selected items<br>**Note:** Non-width-space (\u200b) will be take care by the keyword.<br>**Note:** if the first item_list is * then the keywork will try to click a link named *すべてを選択*. |
| **Select Window** | *title* | Selects a window by its title |
| **Set Capture Counter** | *value=0* | Sets the counter of the screen capture to `value` |
| **Set Capture Format** | *format* | Sets the format for the screen capture file<br>The format does not include the default prefix .png The default format is `<mod>_%010d`. `mod` could be `samurai` or `arbor`<br>See https://docs.python.org/2/library/string.html#format-specification-mini-language for more details about the format string.<br>Examples:<br>Samurai.*Set Capture Format* \| ${case}_%010d \| # ${case} is a predefined variable |
| **Show Detail Mitigation** | *id* | Shows detail information of a mitigation |
| **Show Policy Basic** | *policy_name* | Makes the virtual browser show basic setting of the policy *name*.<br>A following Samurai.*Capture Screenshot* is necessary to capture the result. |
| **Show Policy Mitigation** | *policy_name* | Make the virtual browser show the mitigation setting of a policy<br>A following Samurai.*Capture Screenshot* is necessary to capture the result. |
| **Show Policy Mo** | *policy_name* | Make the virtual browser show the MO setting of a policy<br>Automatically expand the MO section of other devices if necessary.<br>A following Samurai.*Capture Screenshot* is necessary to capture the result. |
| **Show Policy Monitor** | *policy_name* | Make a virtual browser show the mitigation setting of a policy<br>A following Samurai.*Capture Screenshot* is necessary to capture the result. |
| **Start Mitigation** | *policy*, *prefix*, *comment=mitigation started by RENAT*, *device=None*, *force=False* | Starts a mitigation with specific `prefix`<br>`device` is used for matching real device name configured by Samurai If `force` is `TRUE` then the keyword will fail if selected device does not contain `device`<br>Returns mitigation `id` and selected `arbor device`<br>Example:<br>${id} \| ${device}= \| Samurai.*Start Mitigation* \| 211.1.12.1/32 \| mitigation by RENAT \| SP-A \| ${TRUE} |
| **Stop Mitigation** | *id* | Stops a mitigation by its ID<br>Example:<br>Samurai.*Stop Mitigation* \| 700 |
| **Switch** | *name* | Switches the current browser to `name` |

Altogether 32 keywords.
Generated by Libdoc on 2018-06-17 21:39:29.