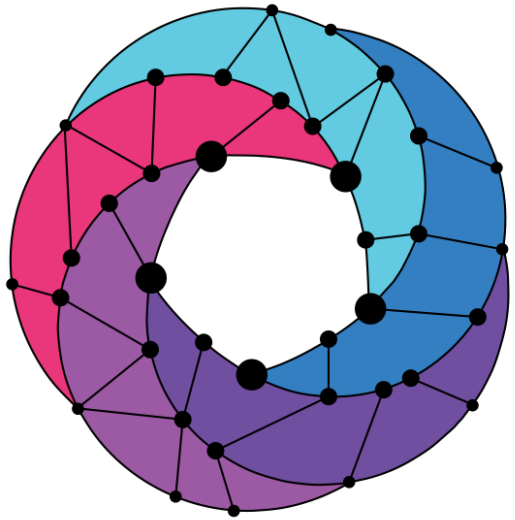


Open Source
MANO

Robot Framework- Hands-on

Jayant Madavi & Mrityunjay Yadav
[Tech Mahindra]

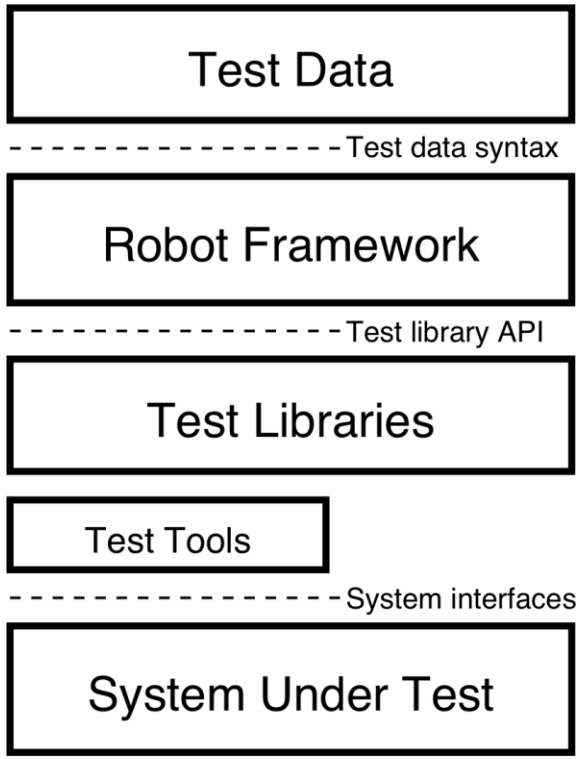
ETSI

What we will be learning

- Problem statement
- Crash course for robot
- Learn robot basics
- Write robot hello World
- Write simple test to test osm
- Write simple test to test OSM GUI
- How to Contribute…

What is Robot Framework?

- Generic open source test automation framework.
- Suitable for both end-to-end acceptance testing and acceptance-test-driven development (ATDD).
- The test syntax follows a tabular style and plain text format which makes writing test cases more user-friendly and easy to read.



Why Robot Framework?

- ❑ Keyword driven, tabular and easy to understand syntax for test case development.
- ❑ Allows creation of reusable keyword.
- ❑ Allows creation of custom keywords.
- ❑ Platform and application independence.
- ❑ Support for standard and external libraries for test automation.
- ❑ Tagging to categorize and select test cases to be executed.
- ❑ Easy-to-read reports and logs in HTML format.

Installation

wget -qO - https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX/OSM%20ETSI%20Release%20Key.gpg | sudo apt-key add -

deb [arch=amd64] https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX stable osmclient IM
or sudo DEBIAN_FRONTEND=noninteractive add-apt-repository -y "$1"
sudo apt-get update
sudo apt-get install python
sudo apt install python-pip

pip install robotframework robotframework-requests requests robotframework-seleniumlibrary
pip install python-magic pathlib pyvcloud pyangbind haikunator

pip list [check the packages installed ?]

sudo apt-get install python-osm-im python-osmclient python-openstackclient
python-osm-devops
robot --help

Write your first code…Hello World!!

mkdir robot

create robot file
vi robot/HelloRobot.robot

```
*** Test Cases ***
 My Test
    Log    "Hello World !!"
```

run robot robot/HelloRobot.robot

sudo apt-get install lighttpd
sudo vi /etc/lighttpd/lighttpd.conf [change the default port in case needed/ html folder to expose]

systemctl restart lighttpd.service
robot -d /home/ubuntu/jam/output/ ./robot/HelloRobot.robot

Test script architecture

Different sections in data:

Section	Used For
Settings	1) Importing test libraries, resource files and variable files. 2) Defining metadata for test suites and test cases.
Variables	Defining variables that can be used elsewhere in the test data.
Test Cases	Creating test cases from available keywords.
Keywords	Creating user keywords from existing lower-level keywords

Test script architecture: Settings

***** Settings *****
1) Importing test libraries, resource files and variable files.
2) Defining metadata for test suites and test cases.

```
*** Settings ***
Documentation     Example using the space separated plain text format.
Library           OperatingSystem
```

Import libraries
Library libraryName arg1 arg2...

Import External Keyword resources
Resource ../../keywords/myKWords.robot

Setup and Teardown
Suite Setup My Suite Setup Keyword
Suite Teardown My Suite Setup Keyword
Test Setup My Test Setup Keyword
Test Teardown My Test Setup Keyword variables.

Tags
Force Tags TAG1 TAG2
Default tags TAG

Test script architecture: Variables

***** Variables *****

Define variables at a "tests suite scope". Variables declared here are accessible from every test cases, keywords or settings

```
*** Variables ***
${MESSAGE}        Hello, world!
```

Creating scalar variables
This is done by giving the variable name (including ${}) in the first column of the Variable table and the value in the second one. If the second column is empty, an empty string is set as a value. Also an already defined variable can be used in the value.

Creating list variables
A list variable can have any number of values, starting from zero, and if many values are needed, they can be split into several rows.

Creating dictionary variables
Dictionary variables can be created in the variable table similarly as list variables. The difference is that items need to be created using name=value syntax or existing dictionary variables.

```
*** Variables ***
@{NAMES}        Matti       Teppo
@{NAMES2}       @{NAMES}    Seppo
&{MANY}         first=1     second=${2}     ${3}=third
&{EVEN MORE}    &{MANY}     first=override  empty=
```

Test script architecture: Test Cases

***** Test Cases *****

List of test cases with each test steps inside. Settings of a test cases are :

[Documentation] Used for specifying the test documentation

[Tags] Used tagging test cases

[Setup], **[Teardown]** Specify test setup (executed before the test) and teardown (executed after the test, even if test failed)

[Template] Specify the template keyword to use for each step

[Timeout] Set the test case execution timeout (Test fails if timeout is reached)

```
*** Test Cases ***
My Test
    [Documentation]    Example test
    Log    ${MESSAGE}
    My Keyword    /tmp

Another Test
    Should Be Equal    ${MESSAGE}    Hello, world!
```

Test script architecture: Keywords

***** Keywords *****
Contains keywords commons to your test suite. Keywords declared here can be used anywhere in the suite, even in setup and teardown calls. Keywords settings are:
[Documentation] Used for specifying the keyword documentation
[Arguments] Specify the keyword arguments
[Return] Specify the keyword return value
[Timeout] Set the keyword execution timeout (Test fails if timeout is reached)

```
*** Keywords ***
My Keyword
    [Arguments]    ${path}
    Directory Should Exist    ${path}
```

Robot Framework: Hello World enhancement

```
*** Settings ***
Documentation    Example Hello World.
Library          OperatingSystem

*** Variables ***
${MESSAGE}       Hello, world!

*** Test Cases ***
My Test
   [Documentation]    Jam Test 1
   [Tags]    Test1
   Log    ${MESSAGE}
   My Keyword    /tmp

Another Test
   [Documentation]    Jam Test 2
   [Tags]    Test2
   Should Be Equal    ${MESSAGE}    Hello, world!

*** Keywords ***
My Keyword
   [Arguments]    ${path}
   Directory Should Exist    ${path}
```

Test Script: VIM Test

```
*** Settings ***
Documentation    Test Suite to create and delete vim account
Library     Collections
Library     RequestsLibrary
Library     OperatingSystem
Suite Setup     Get Auth Token
Suite Teardown  Delete All Sessions

*** Variables ***
&{HEADERS}     Content-Type=application/json       Accept=application/json
&{data}       username=admin      password=admin      project-id=admin
@{success_status_code_list}    200    201    202    204
${descriptor_content_type_gzip}   application/gzip
${auth_token_uri}   /osm/admin/v1/tokens
${create_vim_uri}   /osm/admin/v1/vim_accounts
[Below details are something that needs to be filled based on the local environment]
${OSM_HOSTNAME}                          172.21.248.59
${vim_name}    API-TEST-VIM
${account_type}    openstack
${auth_url}    http://127.0.0.1:5000/v3
${user}    admin
${password}    admin
${tenant}    admin
${description}    Test OpenStack Vim Account

${vim_id}   ${EMPTY}
${token}   ${EMPTY}
${HOST}   ${EMPTY}
```

Test Script: VIM Test…

```
*** Test Cases ***
Create Vim Account
   [Tags]   comprehensive   api_vim_test
   Create Vim   ${vim_name}   ${account_type}   ${auth_url}   ${user}   ${password}   ${tenant}   ${description}

Delete Vim Account
   [Tags]   comprehensive   api_vim_test
   Delete Vim   ${vim_id}

*** Keywords ***
Get Auth Token
   Set Suite Variable   ${HOST}   https://${OSM_HOSTNAME}:9999
   Create Session   osmhit   ${HOST}   verify=${FALSE}   debug=1   headers=${HEADERS}
   ${resp}=   Post Request   osmhit   ${auth_token_uri}   data=${data}
   Pass Execution If   ${resp.status_code} in ${success_status_code_list}   Get Auth Token completed
   ${content}=   To Json   ${resp.content}
   ${t}=   Get From Dictionary                ${content}        _id
   Set Suite Variable   ${token}   ${t}

Create Vim
   [Arguments]  ${vim_name}   ${account_type}   ${auth_url}   ${user}   ${password}   ${tenant}   ${description}
   &{request_data}=   Create Dictionary   vim_user=${user}   vim_password=${password}   vim_url=${auth_url}
vim_tenant_name=${tenant}   vim_type=${account_type}   description=${description}   name=${vim_name}
   &{headers}=   Create Dictionary   Authorization=Bearer ${token}   Content-Type=application/json   Accept=application/json
   Create Session   osmvim   ${HOST}   verify=${FALSE}   headers=${headers}
   ${res}=   Post Request   osmvim  ${create_vim_uri}   data=${request_data}
   log   ${res.content}
   Pass Execution If   ${res.status_code} in ${success_status_code_list}   Create Vim Request completed
   Get Vim ID   ${res.content}
```

Test Script: VIM Test…

```
Delete Vim
    [Arguments]  ${vim_id}
    ${uri} =   Catenate    SEPARATOR=/     ${create_vim_uri}      ${vim_id}
    ${resp}=    Delete Request    osmvim    ${uri}
    log   ${resp.content}
    Pass Execution If      ${resp.status_code} in ${success_status_code_list}   Delete Vim Request
completed

Get Vim ID
    [Arguments]  ${res}
    ${content}=     To Json     ${res}
    ${id}=      Get From Dictionary     ${content}                  id
    Set Suite Variable     ${vim_id}   ${id}
```

Installation for GUI testing

Open Source MANO

```
sudo apt-get update
sudo apt-get install -y unzip xvfb libxi6 libgconf-2-4 curl
sudo apt-get install default-jdk

curl -sS -o - https://dl-ssl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
sudo DEBIAN_FRONTEND=noninteractive add-apt-repository -y "deb [arch=amd64]
http://dl.google.com/linux/chrome/deb/ stable main"
sudo apt-get -y update

sudo apt-get -y install google-chrome-stable

wget https://chromedriver.storage.googleapis.com/2.41/chromedriver_linux64.zip
unzip chromedriver_linux64.zip

sudo mv chromedriver /usr/bin/chromedriver
sudo chown root:root /usr/bin/chromedriver
sudo chmod +x /usr/bin/chromedriver
```

Test Script: GUI Login Test

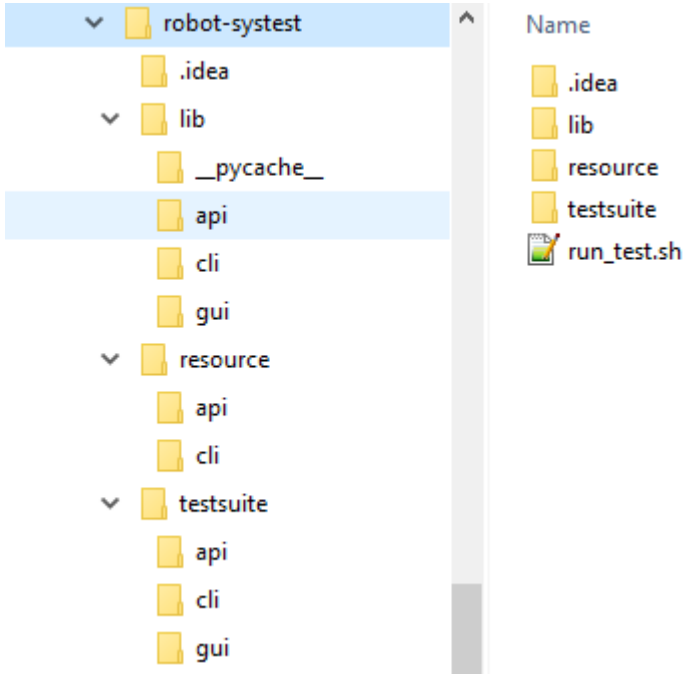
```
*** Settings ***
Documentation    Suite description
Library          SeleniumLibrary
Library          OperatingSystem

*** Variables ***
${OSM_HOST}    172.21.248.59
${DESIRED_CAPABILITIES}    desired_capabilities
${BROWSER}       Chrome
${DELAY}         0
${VALID USER}    admin
${VALID PASSWORD}    admin
${LOGIN URL}     /auth/
${WELCOME URL}    /projects/
${NS LIST URL}    /packages/ns/list
${VNF LIST URL}    /packages/vnf/list

*** Test Cases ***
Valid Login
    [Tags]    comprehensive   gui_login_test
    [Setup]    Set Server URL
    Open Browser To Login Page
    Enter Credentials    admin    admin
    Submit Credentials
    Home Page Should Be Open
    [Teardown]    Close Browser
```

Test Script: GUI Login Test…

```
*** Keywords ***
Set Server URL
   Set Suite Variable    ${SERVER}   http://${OSM_HOST}

Open Browser To Login Page
   ${chrome_options} =    Evaluate    sys.modules['selenium.webdriver'].ChromeOptions()    sys, selenium.webdriver
   Call Method    ${chrome_options}   add_argument    headless
   Call Method    ${chrome_options}   add_argument    disable-gpu
   Call Method    ${chrome_options}   add_argument    no-sandbox
   ${options}=    Call Method    ${chrome_options}    to_capabilities
   Open Browser    ${SERVER}${LOGIN URL}    ${BROWSER}    desired_capabilities=${options}
   Maximize Browser Window
   Set Selenium Speed    ${DELAY}
   Login Page Should Be Open

Login Page Should Be Open
   Element Text Should Be    //*[@id="main_content"]/div/div[2]/p    Sign in to start your session

Enter Credentials
   [Arguments]    ${username}    ${password}
   Input Text    name:username    ${username}
   Input Password    name:password    ${password}

Submit Credentials
   Click Button    //*[@id="main_content"]/div/div[2]/form/div[3]/div[2]/button

Home Page Should Be Open
   Location Should Be    ${SERVER}${WELCOME URL}
   Element Attribute Value Should Be    //*[@id="main_content"]/div/div[2]/div[1]/div[1]/div/a    href    ${SERVER}${NS LIST URL}
   Element Attribute Value Should Be    //*[@id="main_content"]/div/div[2]/div[1]/div[2]/div/a    href    ${SERVER}${VNF LIST URL}
```

Robot current status: Feature 7829

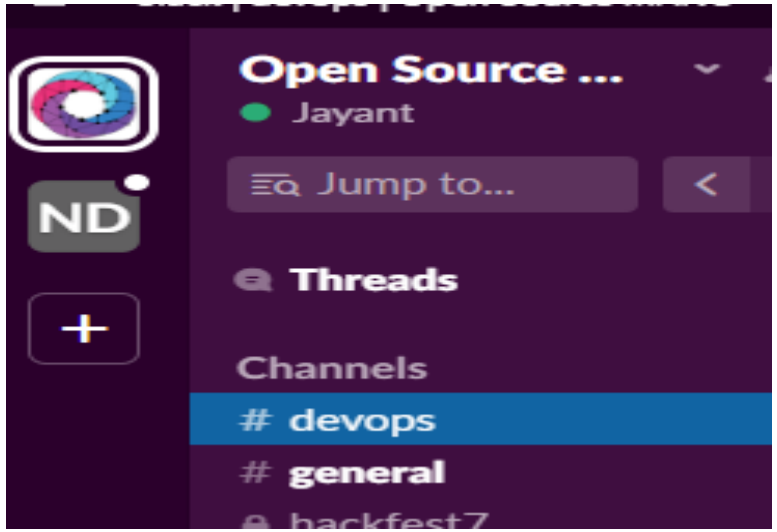
https://osm.etsi.org/gerrit/#/c/osm/devops/+/7829
https://osm.etsi.org/gerrit/#/c/osm/Features/+/7829/1/Release7/Robot_Integration_Tests.md



Join community #Devops

https://osm.etsi.org/wiki/index.php/Release_SIX_Integration_(DEVOPS)

36	034. Feature 7326 - Disable port security at network level		PASSED
37	035. Feature 7366 - Eclipse fog05	Not applicable	Not applicable
38	036. Feature 1417 - Support of PDUs		PASSED
39	037. Feature 1420 - VNF SW upgrade (Adam)		PASSED
40	038. Feature 638 - Service chaining	FAILED	Not applicable
41	039. Feature 1413 - OSM platform resiliency to single component failure		PASSED
42	040. Feature 1412 - OSM platform recovery after major failure		PASSED
43	041. Feature 5650 - Allow to specify management IP addresses as parameters at instantiation time	PASSED	PASSED
44	042. Feature 5945 - Enable dynamic connectivity setup in multi-site Network Services (only CRUD over WIM)		PASSED
45	043. Feature - Control of LCM operations over a NS instance		PASSED



#devops on slack
#bi-weekly meeting

OSM TECH – DevOps bi-weekly calls

Wednesdays @ 16:00 CEST

To join : https://www.gotomeet.me/OSMTECH

Access Code: 119-703-237

References

Develops/User guide:

http://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html

https://github.com/robotframework/QuickStartGuide/blob/master/QuickStart.rst

https://twiki.cern.ch/twiki/bin/view/EMI/RobotFrameworkAdvancedGuide\

https://bulkan.github.io/robotframework-requests/

Editor:

https://pypi.org › project › robotframework-ride
https://macromates.com/
https://github.com/nokia/RED
http://www.jetbrains.com/pycharm/

Thanks

Open Source MANO

Q & A

Thank You