Getting started with Robot Framework

note: RIDE might not work with python 3

Today we will learn:

================

1. How to setup Robot Framework from scratch

2. How to uninstall Robot Framework

3. Helpful Tips

=================================

How to setup Robot Framework

-------------------------------------------

Step 1 : Pre-check

Check if python is already installed

python --version

pip --version

Check if robot framework is already installed

robot --version

Step 2 : Install Python

check on cmd : python --version

pip --version

Step 3 : Set Python in environment variables

C:\Python27;C:\Python27\Scripts;

Check:

python --version

pip --version

Step 4 : Install robot framework

pip install robotframework

to uninstall : pip uninstall robotframework

Step 5 : Check on cmd :

robot --version

pybot --version

Step 6 : Download and install wxPython

https://sourceforge.net/projects/wxpy...

check with : pip freeze

wxpython should be available

Step 7 : Install RIDE

pip install robotframework-ride

https://github.com/robotframework/RID...

For Mac OS try these commands:

pip install -U -r https://raw.githubusercontent.com/rob...

pip install -U robotframework-ride==1.7.4b1

Step 8 : On cmd goto folder where ride.py is (C:\Python27\Scripts)

run on cmd : ride.py

This should open RIDE

======================================================

How to uninstall Robot Framework

-------------------------------------------------------

pip uninstall robotframework-ride

pip uninstall robotframework

======================================================

Helpful TIPS:

1. Always install wxPython before installing RIDE.

wxPython is a wrapper for getting the GUI of RIDE.

So it is important to install wxPython before you install RIDE

2. Always check your python version and install wxPython for the same version

\*\*\*32 bit - check your python ver by running command python and install same bit wxpython

3. pip cache folder is located at - C:\Users\Raghav Pal\AppData\Local\pip\cache

If you uninstall a lib and install it again, it will use the zip from cache.

to install fresh

- you can empty contents of cache folder

- pip install --no-cache-dir robotframework-ride

4. Versions prior to Robot Framework 3.0 did not have the robot script. Instead they had pybot, jybot and ipybot scripts that executed tests using Python, Jython and IronPython, respectively. These scripts are still installed, but the plan is to deprecate and remove them in the future.

5. Robot Framework UserGuide -

http://robotframework.org/robotframew...

https://github.com/robotframework/Qui...

Keyboard Shortcuts -

https://github.com/robotframework/RID...

RF GitHub page -

<https://github.com/robotframework>

# How to create first Selenium Test with RIDE

Step 1 : Open RIDE

You can open ride from cmd by command ride.py

You can also create a desktop shortcut

Step 2 : RIDE - Very Simple UI

Step 3 : Create New Project - Test Suite - TestCase

Step 4 : Download and add SeleniumLibrary

Goto - http://robotframework.org/ - LIBRARIES - External -

SeleniumLibrary

Step 5 : In Project

Add Import - Library - SeleniumLibrary

If it shows in Black that means its successfully imported

else not (in red)

Step 6 : Now in the TC if you start typing Selenium and ctrl+space it

will show you all the keywords of SeleniumLibrary

You can use full name - SeleniumLibrary.Open Browser

OR

only action - OpenBrowser

Step 7 : Add keywords and run

Step 8 : Add browser drivers

Step 9 : Run and see reports

Step 10 : Settings - Tags - Setup | Teardown

# How to use VARIABLES

What are Variables

Variables are elements used to store values that can be referenced by other elements

RobotFramework has 3 types of variables

SCALAR

LIST

DICTIONARY

we can also use

ENVIRONMENT variables

Step 1 : Open RIDE

Step 2 : Create a Test Case for login

http://opensource.demo.orangehrmlive....

Step 3 : Create a SCALAR variable for url and refer in TestCase

syntax : ${VariableName}

Step 4 : Create a LIST variable for username and password and refer in TestCase

syntax : @{VariableName}

Step 5 : Create a DICTIONARY variable for username and password and refer inTestCase

syntax : &{VariableName}

Environment

%{variable name}

Useful TIPS

=========

1. If a non-existent variable is used in test-data, the keyword using it will fail

2. We can set variables from command line

3. Variables are case-insensitive

4. As best practice use uppercase variable names for global variables

And lowercase variable names for local variables

5. Built in variables - <http://robotframework.org/robotframew>...

# How to use KEYWORDS

USEFUL TIPS

2 Types:

LIBRARY KEYWORDS : low level keywords

USER KEYWORDS : high level keywords

custom keywords

combination of library keywords/actions

Step 1 : Open RIDE

Step 2 : Right click on Project / Test Suite - New User Keyword

Step 3 : Create a keyword for Login

Step 4 : Refer the keyword in a TestCase

Useful Tips

User Keyword syntax is similar to TestCase syntax

Search Keyword - F5

Find Usage

# How to use SETUP and TEARDOWN

SETUP - keyword executed before running any lower level component/keyword

TEARDOWN - keyword executed after running any lower level component/keyword

Step 1 : Open RIDE

Step 2 : Create TestSuite Setup and Teardown and run it

# How to use TAGS

Sanity, Smoke, Regression, Critical

Tags are used to classify/categorize test cases

Useful TIP

command line option : --settag

Step 1 : Goto cmd and cd to your robot framework project

Step 2 : syntax

robot -t TestName --settag=TagName TestSuite.txt

# SOME USEFUL TIPS | Commands

robot --version

robot --help

pip freeze

pip list

pip show

pip check

# How to run Tests from Command Line

Useful TIPS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 1 : Open cmd and goto the location of project

Step 2 : Run single test

robot -t TestName SuiteFile

Step 3 : Run multiple tests

robot -t Test1 -t Test2 SuiteFile

Step 4 : Run Tests with TAGS

Include TAGS

robot --include tagName SuiteFile

robot -i tagName SuiteFile

robot -i tagName -i tagName SuiteFile

robot -i S\* SuiteFile

Exclude TAGS

robot --exclude tagName SuiteFile

robot -e tagName SuiteFile

robot -e tagName -e tagName SuiteFile

robot -e S\* SuiteFile

Step 5 : Run all tests in a test suite

robot suiteFile

Step 6 : Send results to a folder

robot -t Test5 -d Results TestSuite1.txt

Useful TIPS

Using regular expressions

how to make tags critical / non-critical

e.g.

robot -i S\* --critical Smoke TestSuite1.txt

robot -i S\* -c Smoke TestSuite1.txt

robot -i S\* --noncritical Smoke TestSuite1.txt

robot -i S\* -n Smoke TestSuite1.txt

Command line options guide

<https://github.com/robotframework/rob>...

# How to run Robot tests from JENKINS

Useful TIPS

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 1 : Download Jenkins

Step 2 : Start Jenkins

Step 3 : Add robot framework plugin in jenkins

Step 4 : Create job to run robot framework tests

Step 5 : Run and validate

In case not able to see the html reports in jenkins

Goto Manage Jenkins - Script Console

Add this script and run

System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")

If you see Result, it means success

Now from next runs html reports will be accessible from jenkins

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Useful Tips

Goto Dashboard - You will find a new column Robot Results

You can use wild cards in name of output files of robot framework in Jenkins

Jenkins content security policy - to enable viewing html reports