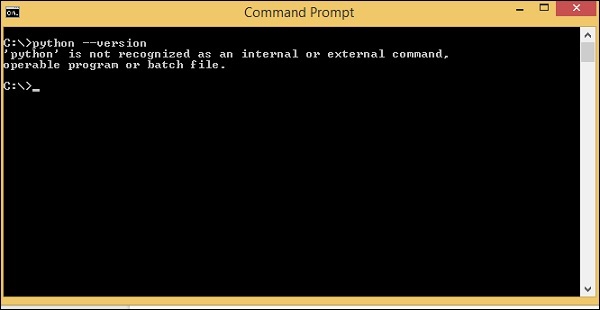
**ROBOT FRAMEWORK**

**Course Contents**

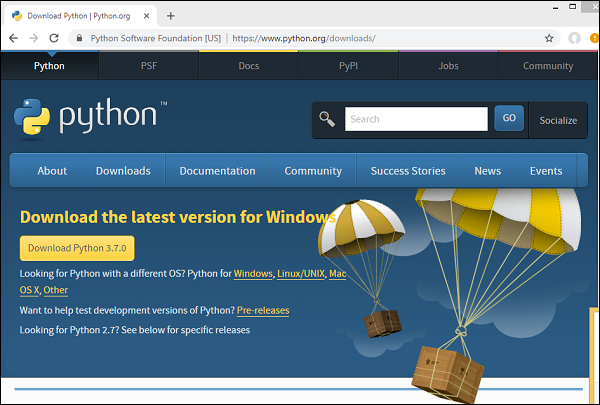
* Introduction
* Installation Steps
* How to work with RIDE tool
* How to import python code in RIDE tool
* Advantages
* Limitations
* **Introduction:**
* **Robot Framework** is an open source test automation framework for acceptance testing and acceptance test-driven development
* Test cases are written using keyword style in a tabular format
* Robot Framework provides good support for external libraries, tools that are open source and can be used for automation.
* Test cases are written using keyword style in a tabular format. We can use any text editor or Robot Integrated Development Environment (RIDE) for writing test cases.
* Robot framework works fine on all the Operating Systems available. The framework is built on Python and runs on Jython (JVM) and IronPython (.NET)
* Robot framework is built using python
* **Installation Steps:**
* Robot framework is built using python. To work with Robot Framework, we need to install the following –
* Python
* pip
* Robot Framework
* wxPython for Ride IDE
* Robot Framework Ride
* **Install python -**
* Before you download python, it is recommended you check your system if python is already present by running the following command in the command line –

**Windows Installation**

* python --version



* If python is not available ,download python using link <https://www.python.org/downloads/>
* Here is the screenshot of the python download site −

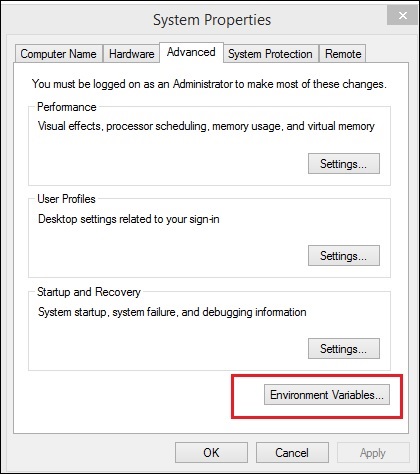


* The latest version available as per release dates are as follows −

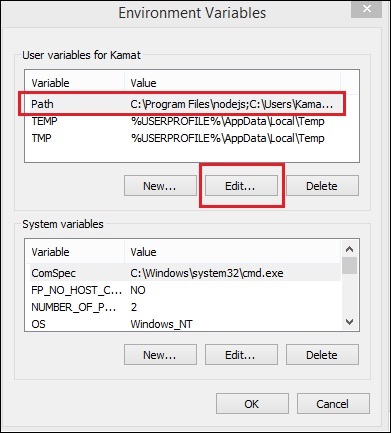


* **Setting path for Windows**

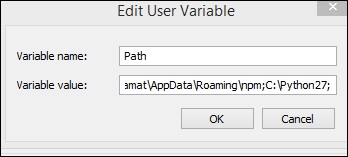
Right-click on My Computer icon and select properties. Click on Advanced System setting and the following screen will be displayed.



Click on *Environment Variables* button highlighted above and it will show you the screen as follows −

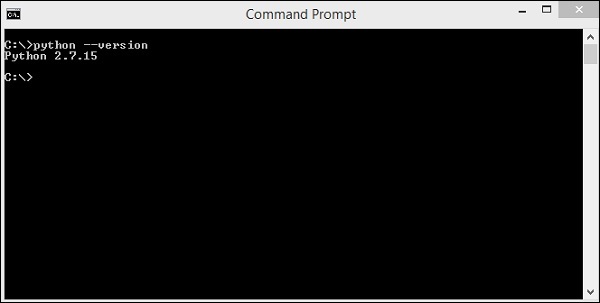


Select the *Variable Path* and click the *Edit* button.



Get the path where python is installed and add the same to Variable value at the end as shown above

Once this is done, you can check if python is installed from any path or directory as shown below −



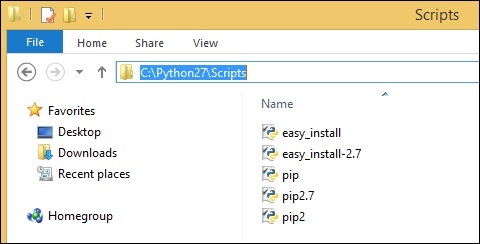
* **Install PIP**
* Now, we will check for the next step, which is pip installation for python. PIP is a package manager to install modules for python.
* PIP gets installed along with python and you can check the same in command line as follows –

**Command**

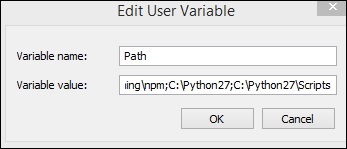
pip --version



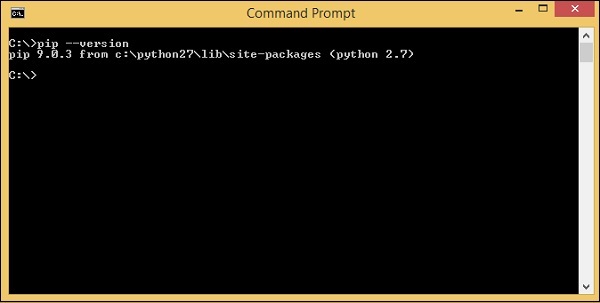
Here we are still not getting the version for pip. We need to add the pip path to Environment variables so that we can use it globally. PIP will be installed in Scripts folder of python as shown below −



Go to environment variables and add the path of pip to the variables list. Add C:\Python27\SCripts to environment variables as follows −



Now open your command line and check the version of pip installed −



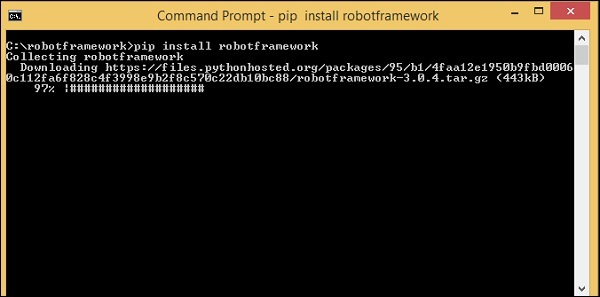
So now, we have python and pip installed.

* **Install Robot Framework**

We will now use pip – python package manager to install the robot framework and the command for it is as follows –

**Command**

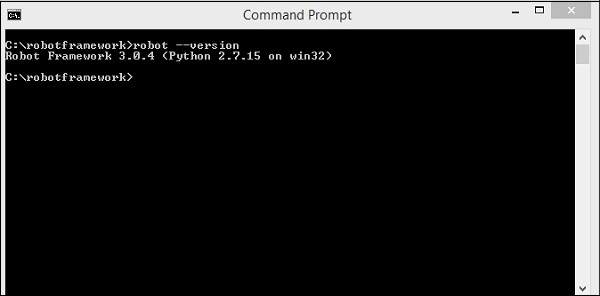
* pip install robotframework



Once the installation is done, you can check the version of robot framework installed as shown below –

**Command**

robot --version



So, we can see Robot Framework 3.0.4 is installed.

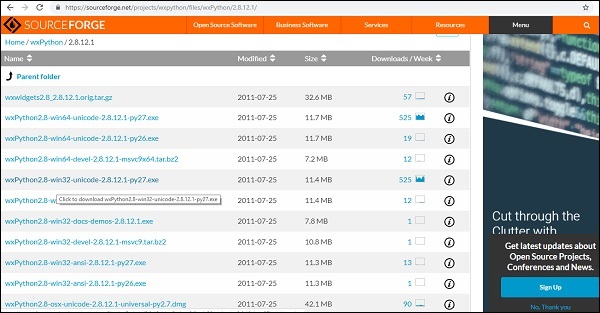
* **Install wxPython**

We need wxPython for Robot Framework Ride, which is an IDE for Robot Framework.

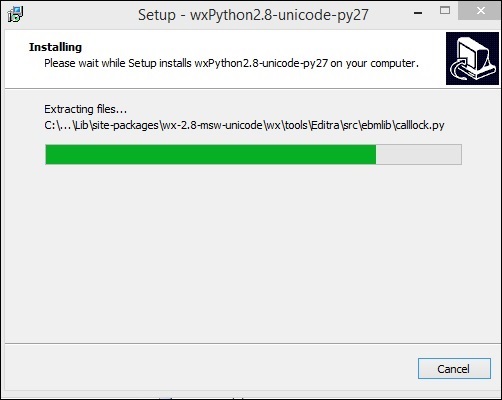
* **For windows** to get the required download for wxPython, go to the following URL –

<https://sourceforge.net/projects/wxpython/files/wxPython/2.8.12.1/>

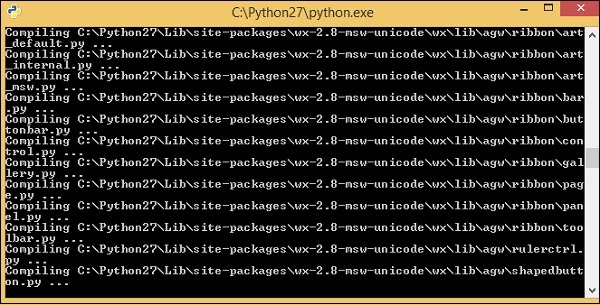
And, download 32 or 64-bit wxpython for windows as per your Windows Operating system.



Download the 32-bit wxPython and install the same.



Once the installation is done, it opens the command line and auto runs some commands as shown below −



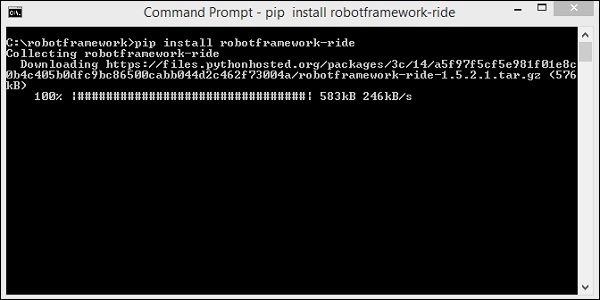
wxPython is now installed

* **Install Ride**

Ride is Robot Framework IDE. We can use pip to install it as shown below.

### **Command**

pip install robotframework-ride



Once the installation is done, open the command prompt and type the following command to open the Ride-IDE.

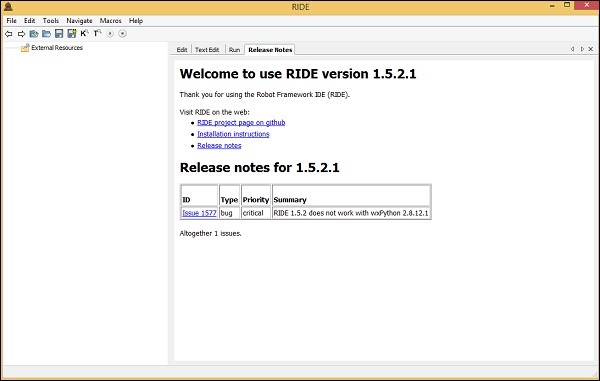
* **Robot Framework - Introduction to Ride**

Ride is a testing editor for Robot Framework. Further, we will write test cases in Ride. To start Ride, we need to run the command shown below.

**Command**

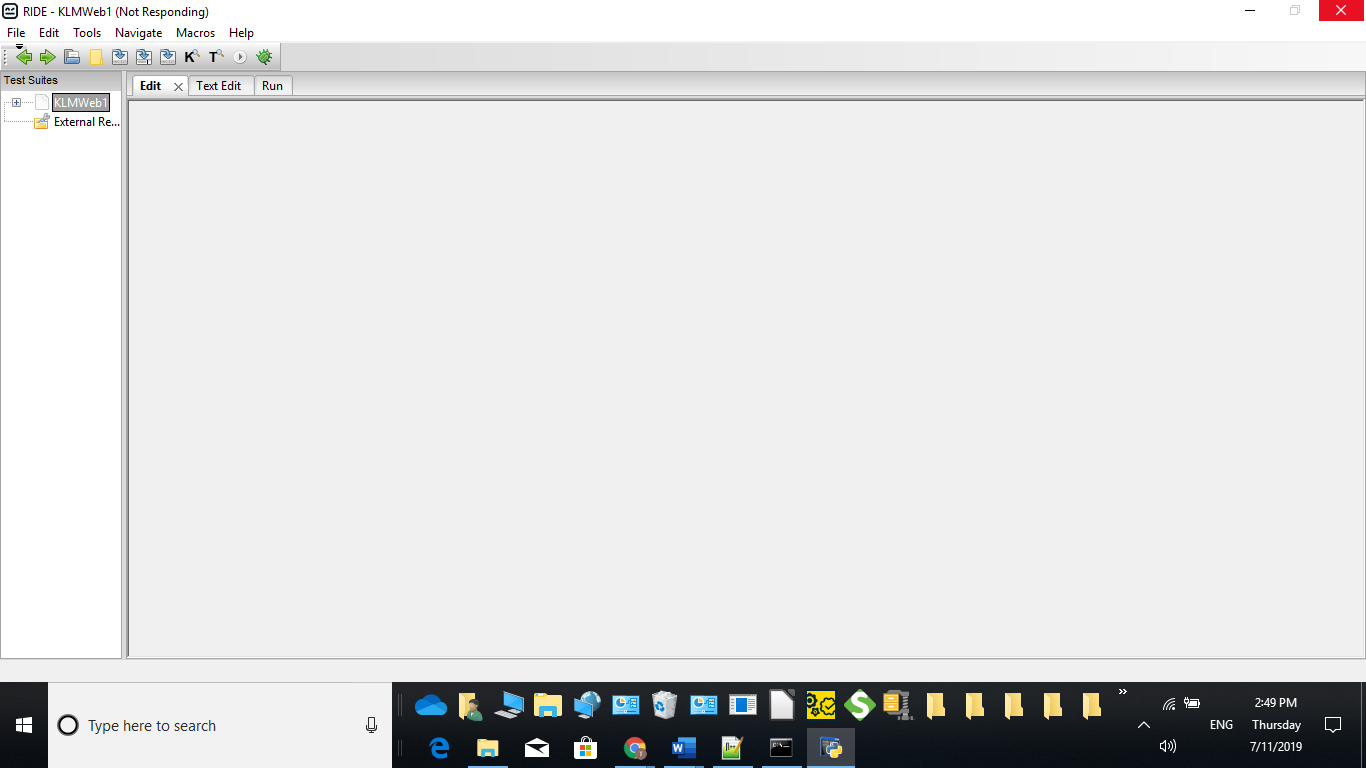
ride.py

The above command opens the IDE as follows −



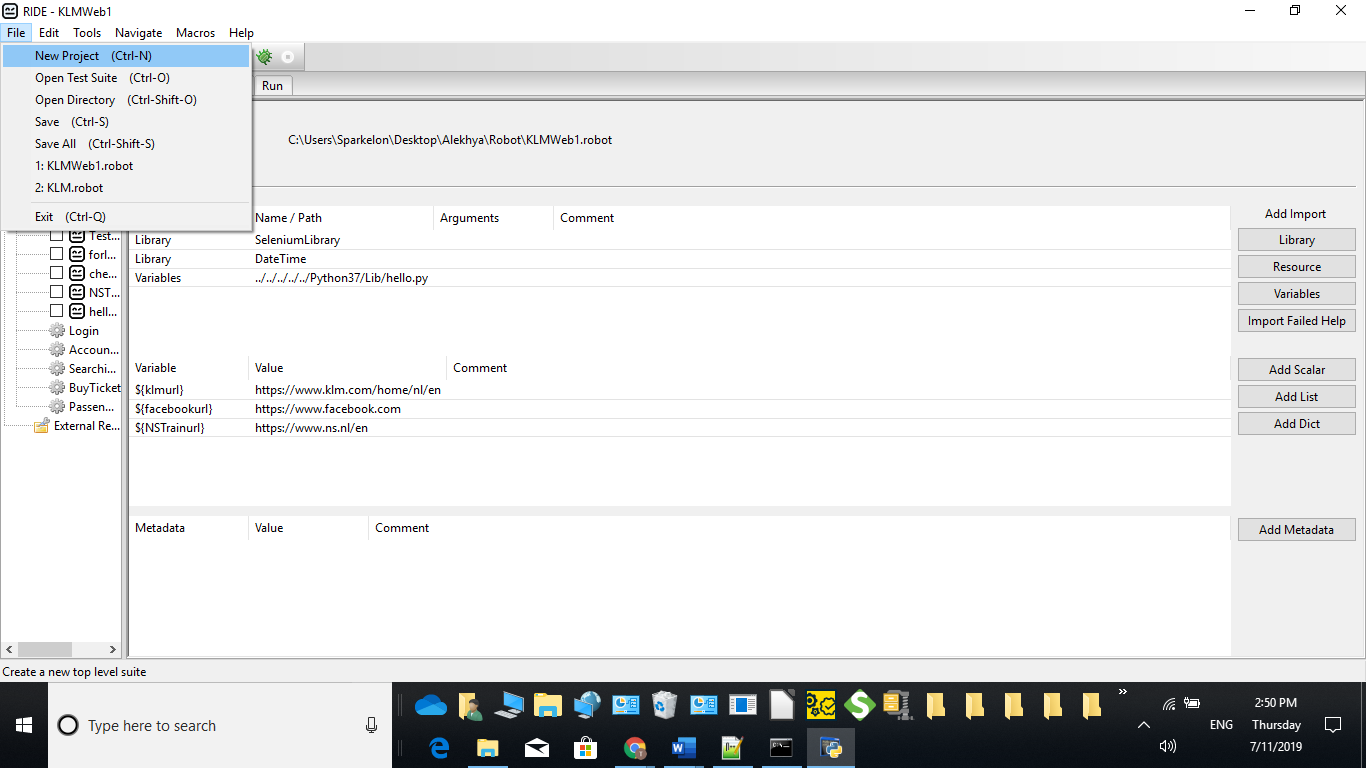
* **How to work with RIDE tool :**

Ride tool editor will be as shown below

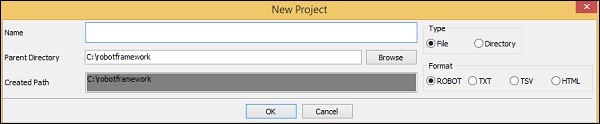


* **Create New Project:**

File🡪New project

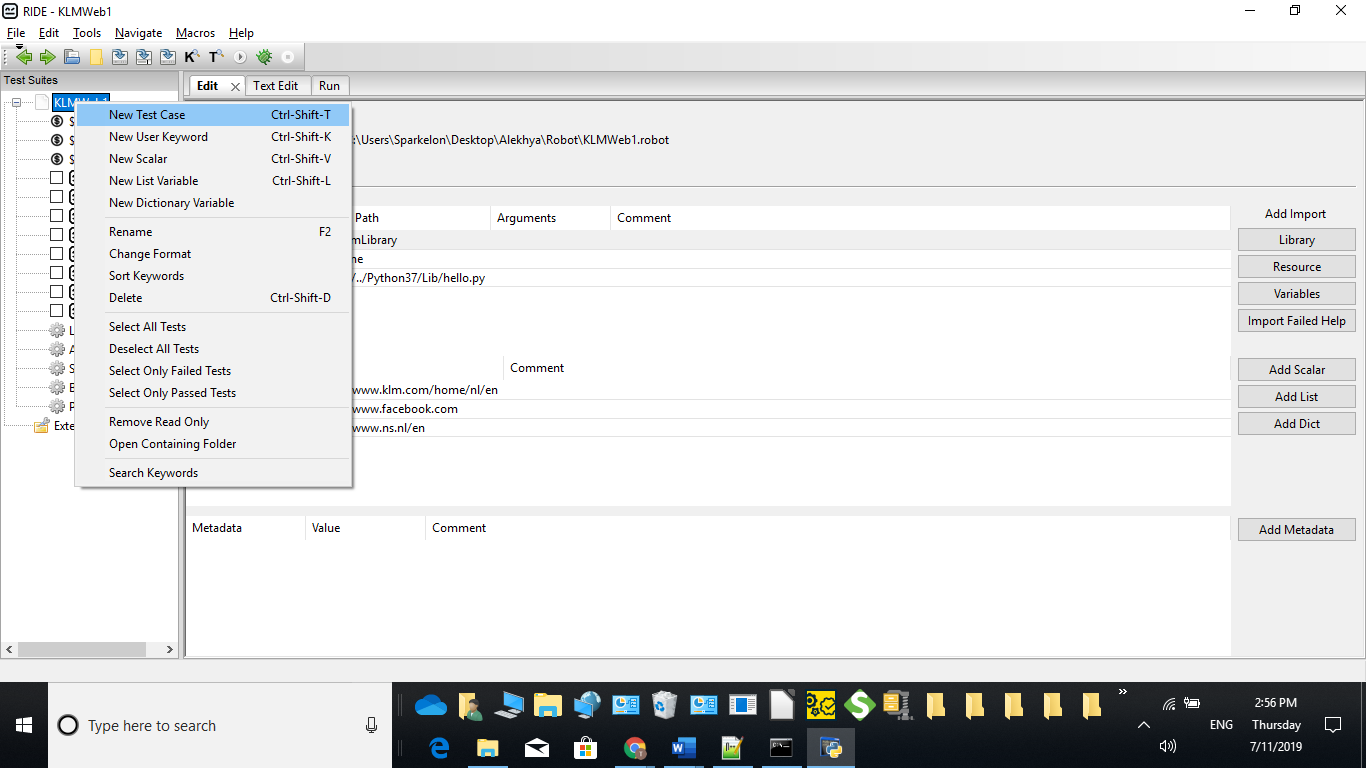


The following screen will appear when you click New Project.



Give project name : Klmweb1

Right click on project 🡪Create New Test Case



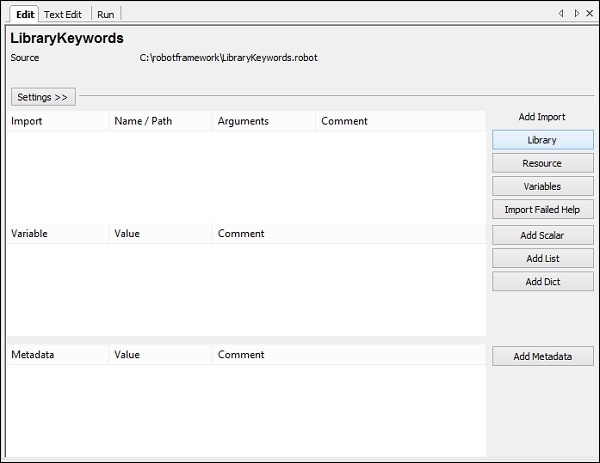
**To work with selenium**

In command prompt run the below command:

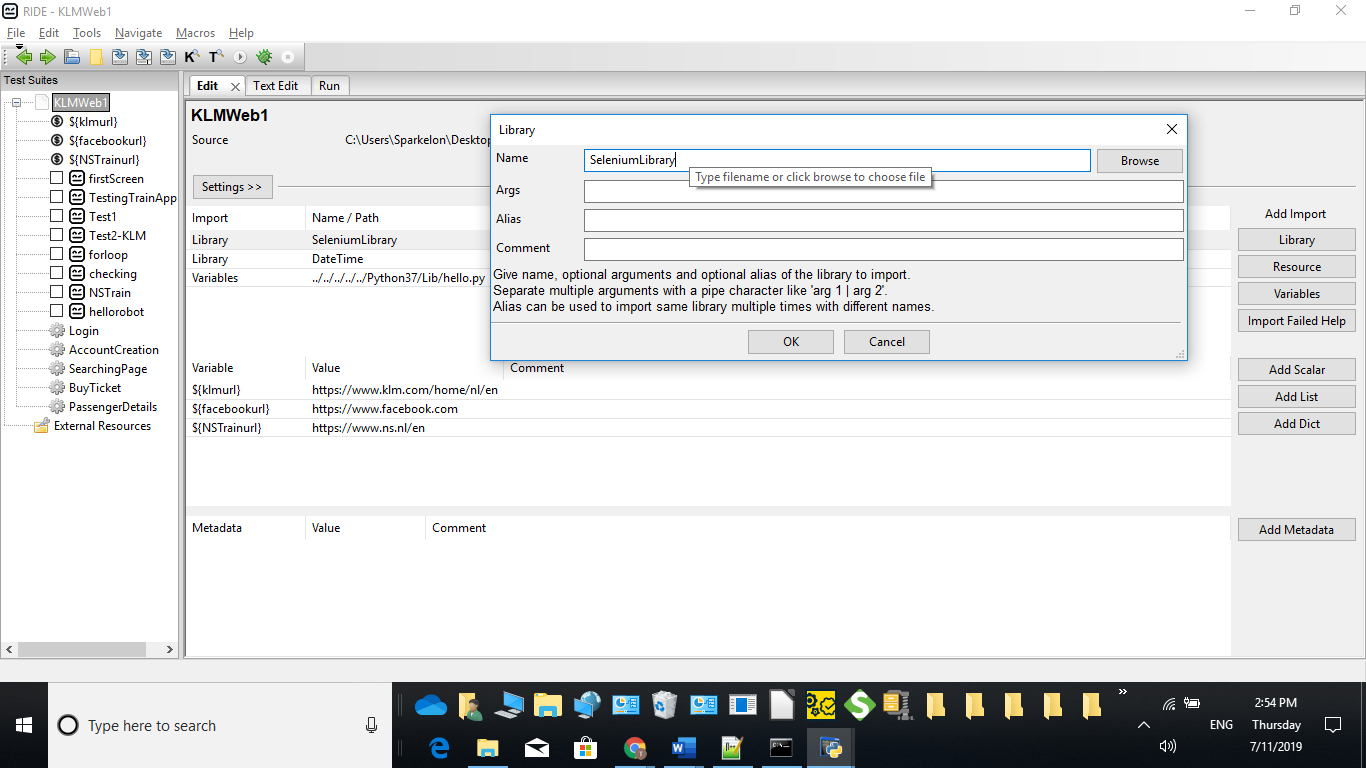
* pip install --upgrade robotframework-seleniumlibrary
* **Library Keywords**

Library Keywords are keywords that come from the library we import in Robot Framework. We will now take a look at the Selenium library, which helps us interact with the browser

Click on your project on the left side and click Library.



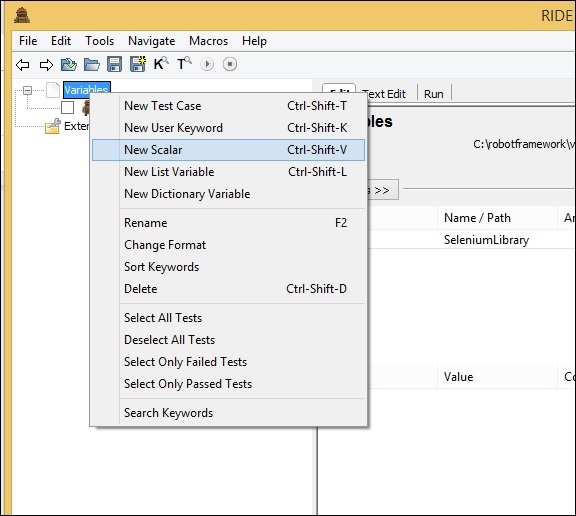
To import external libraries 🡪 click on Library



* **Test Case for Scalar Variable:**

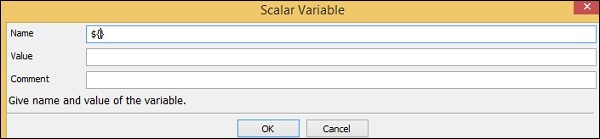
To create scalar variable, right-click on your project and click on *New Scalar* as shown below

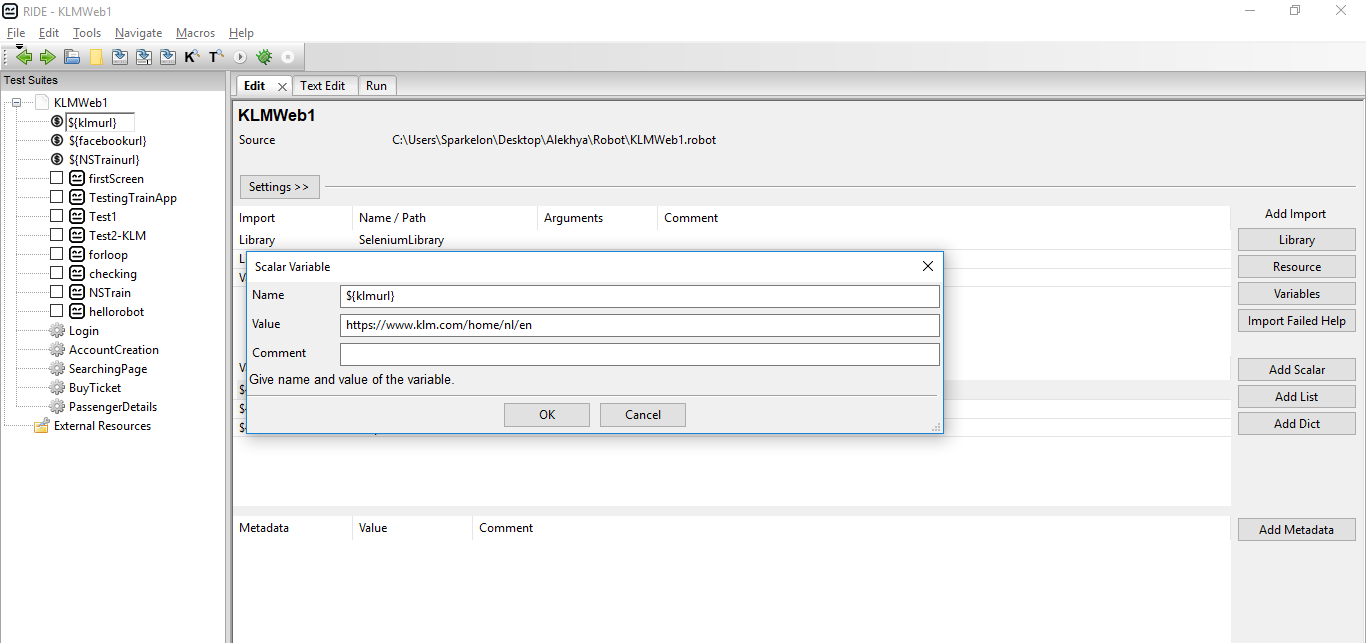
Right click on project 🡪create scalar



Clicking on New Scalar will open the following screen to create the variable and the value we need to replace with when the variable in used inside test cases.

We get ${} for the Name field.

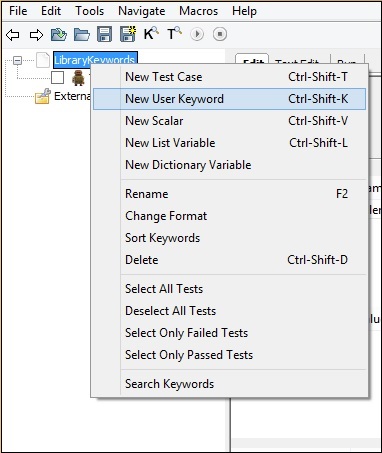




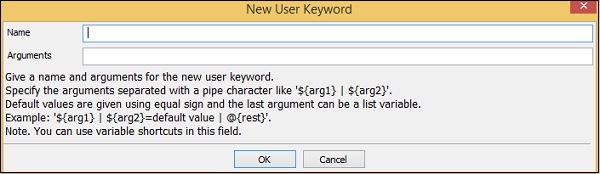
* **User-defined Keywords**

User-defined keywords can be created to perform a particular action in the test case or it can also be created using the library keywords and built-in keywords in robot framework.

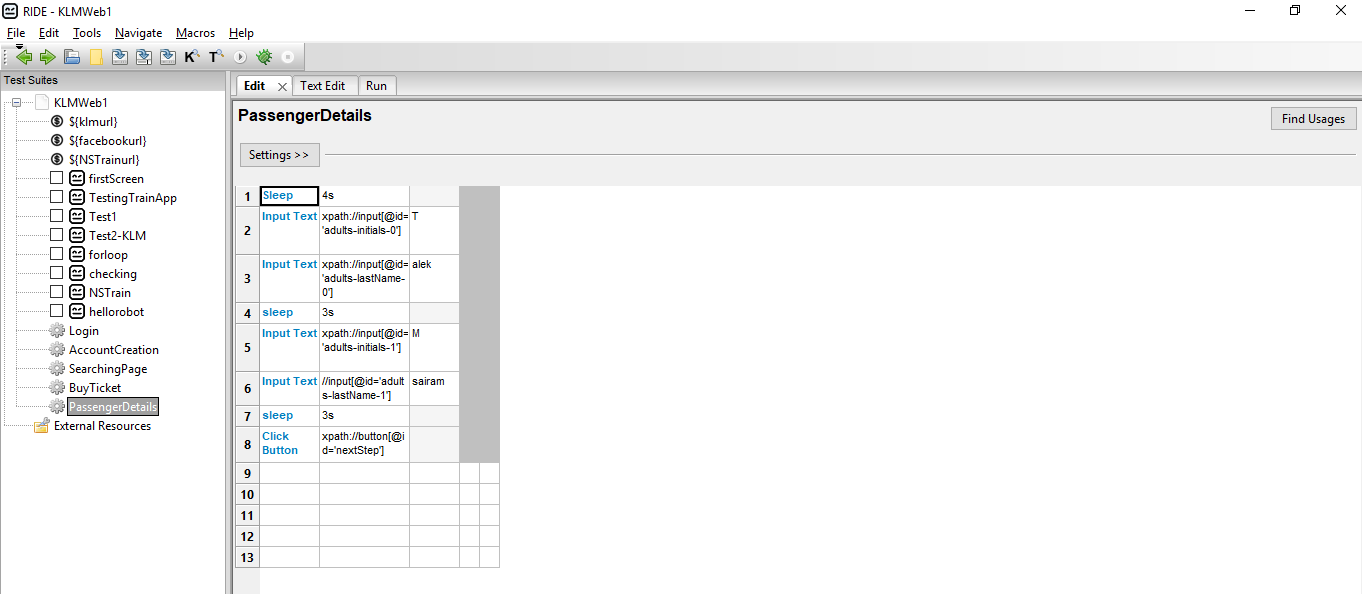
To create keyword in Ride, right-click on your project and click on New User Keyword as shown below −



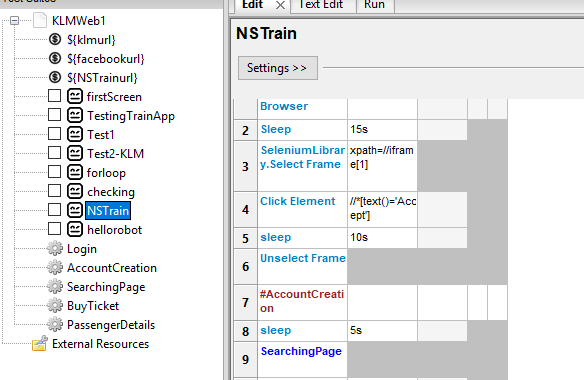
Upon clicking New User Keyword, a screen appears as shown below −



Enter keyword name as **PassengerDetails**

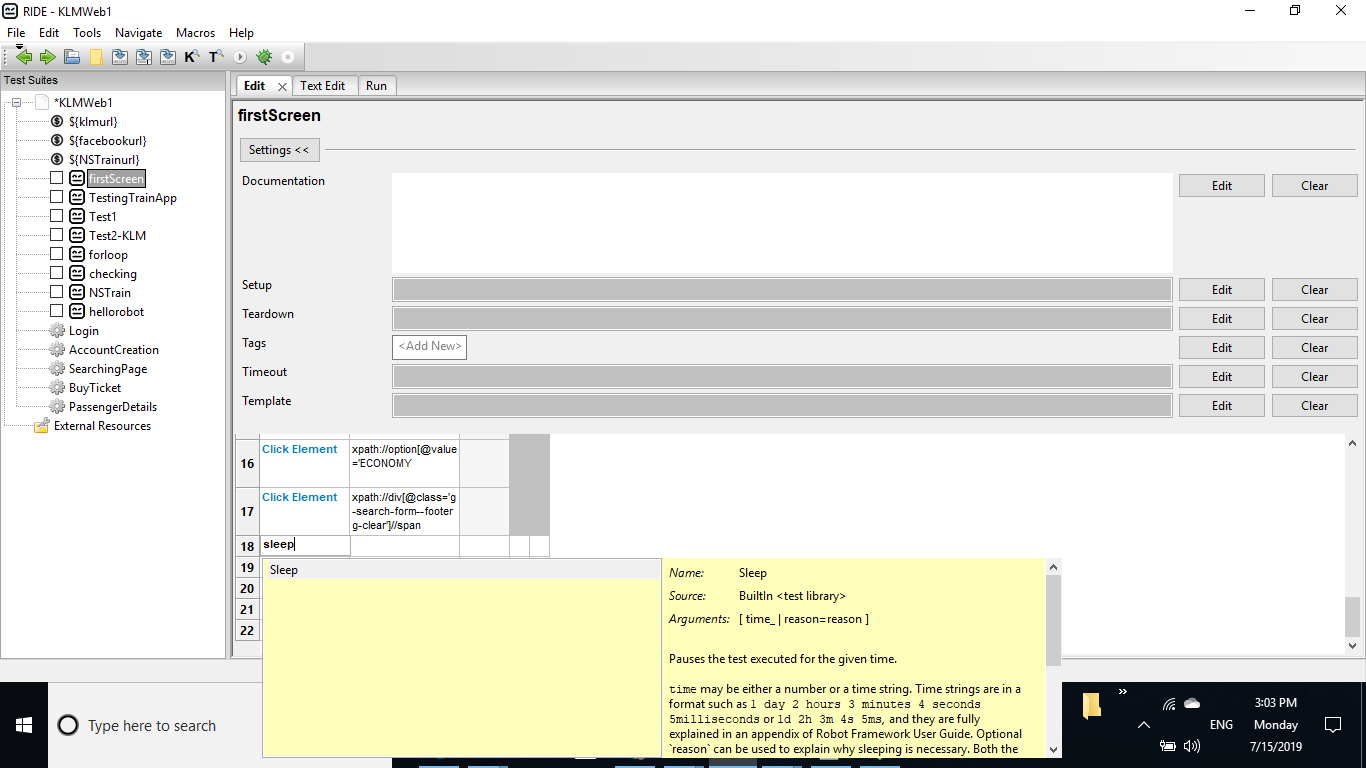


When we refer user keyword in testcase it will highlights blue colour as shown below



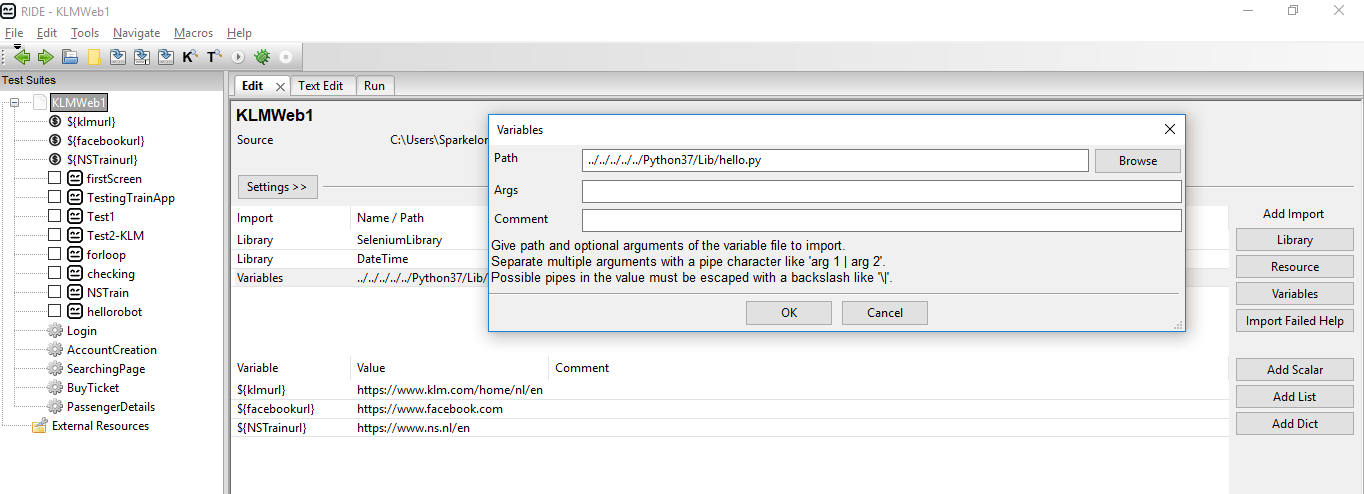
TestCase 🡪Edit

Enter the keyword and ctrl+alt+space then builtin keywords will display



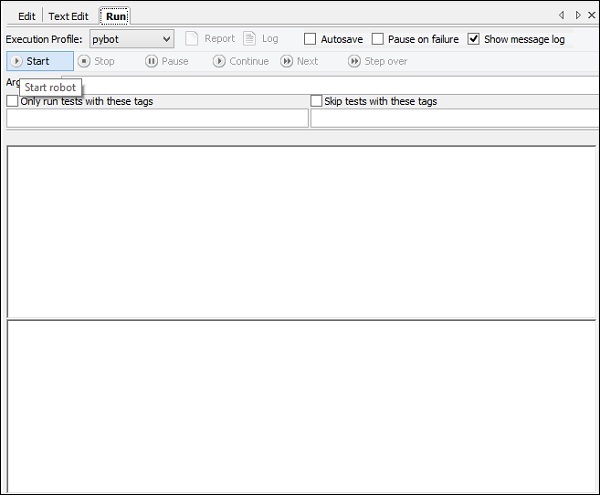
* **How to import python code in RIDE tool:**
* Create python custom code and save as .py extension
* To access this in robot framework ride tool place the file library folder. i.e C:\Python37\Lib
* Go to RIDE 🡪project -Settings🡪Add import🡪Variables

Add path to the python file

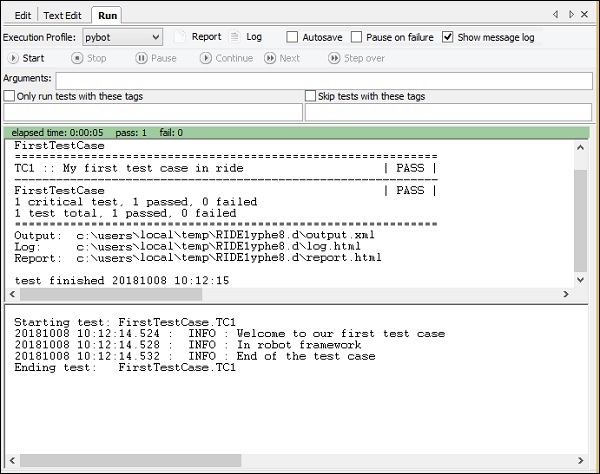


* **How to run Testcase in RIDE:**

To run the test case, we need to click on Start as shown below −



Click on start and here are is the output of the test case −

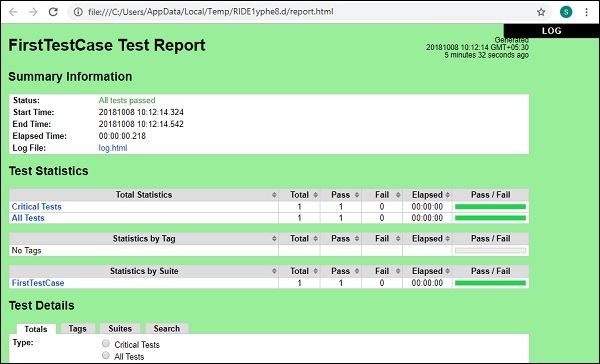


Our test case has executed successfully and the details are as shown above. It gives the status as *PASS*.

We can also see the details of the test case execution in Report and Log as highlighted below.

case execution

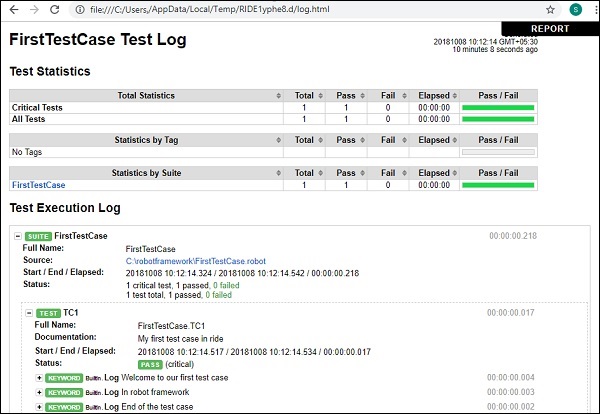
Click on Report and it opens the details in a new tab as follows



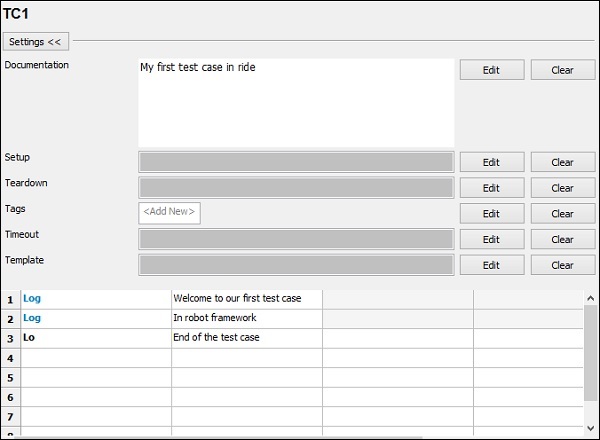
In Report, it gives the details like the start time, end time, path to the log file, status of the test case, etc.

Click on Log at the top right corner in report or from the Run screen.

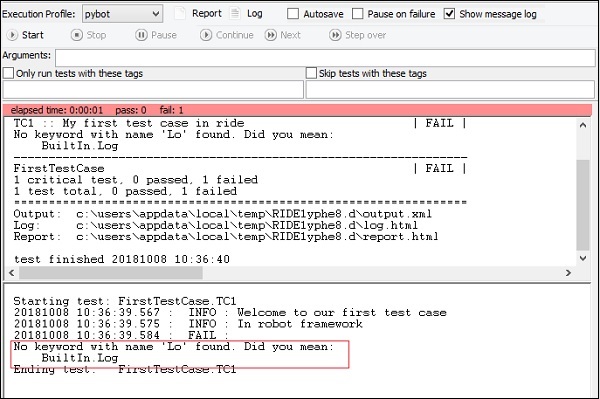
Here are the details of the log file −



* The Log file gives the details of the test execution and the details of keywords we gave for the test case.
* In the report and the log file, we get green color for the status.
* Let us now make some changes that will lead to the failure of the test case fail and see the output.

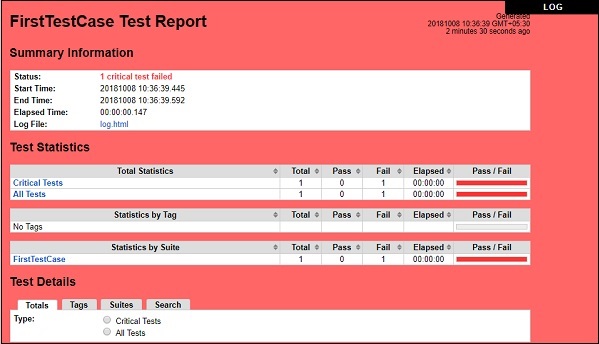


In the above test case, the Log keyword is wrong. We will run the test case and see the output −

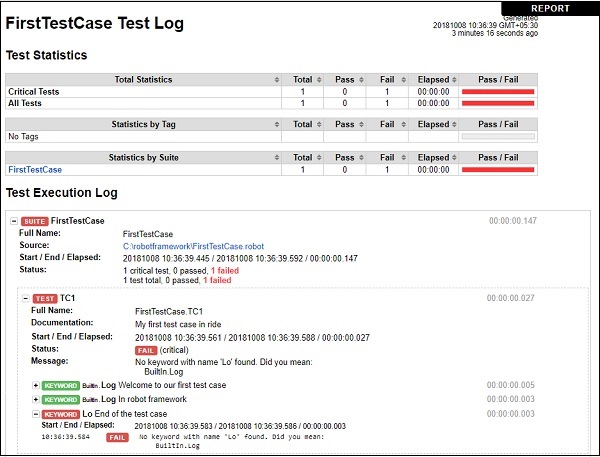


We see that the test case has failed. I have highlighted the error that it tells about the test case.

Now will see the report and log output.From Report −



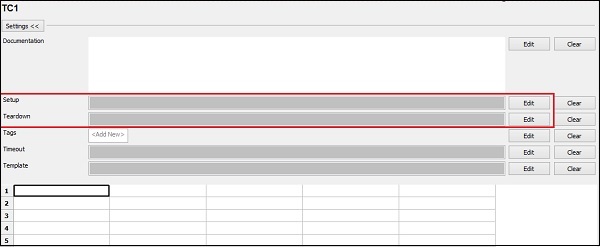
### **From Log**



When the test case fails, the color is changed to Red as shown above.

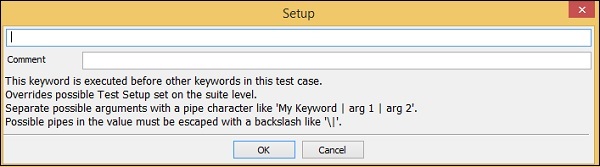
* **Set Up and Tear down scripts:**

Open Test Case 🡪click Settings

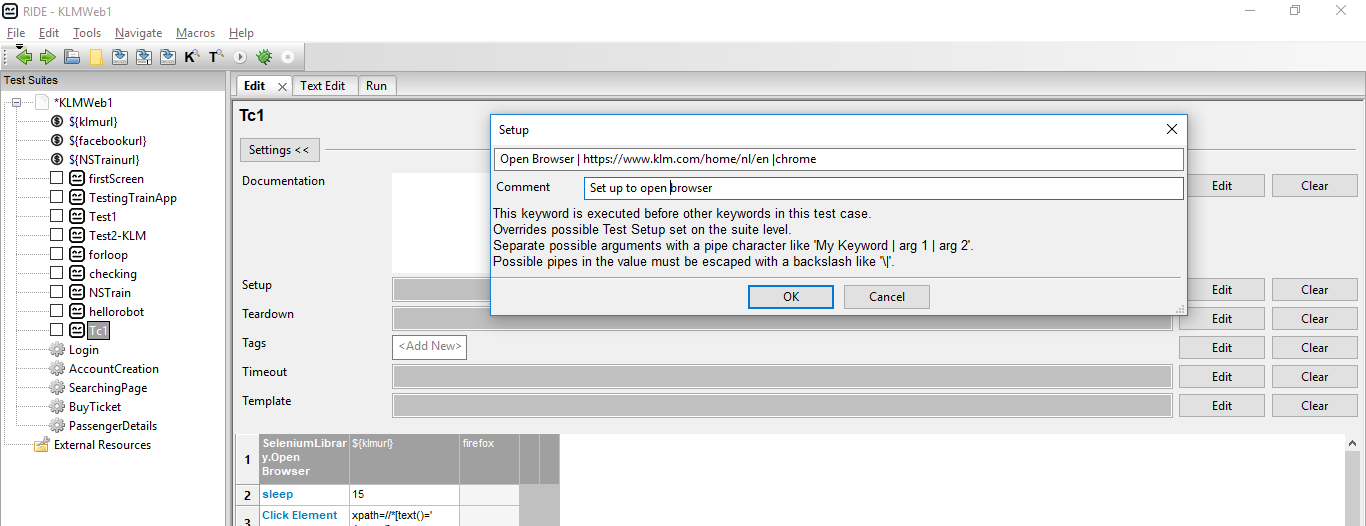


In the above screenshot, the Settings section has *Setup and Teardown* options. For Setup, click **Edit** to enter the keyword.

Now, enter the Keyword −



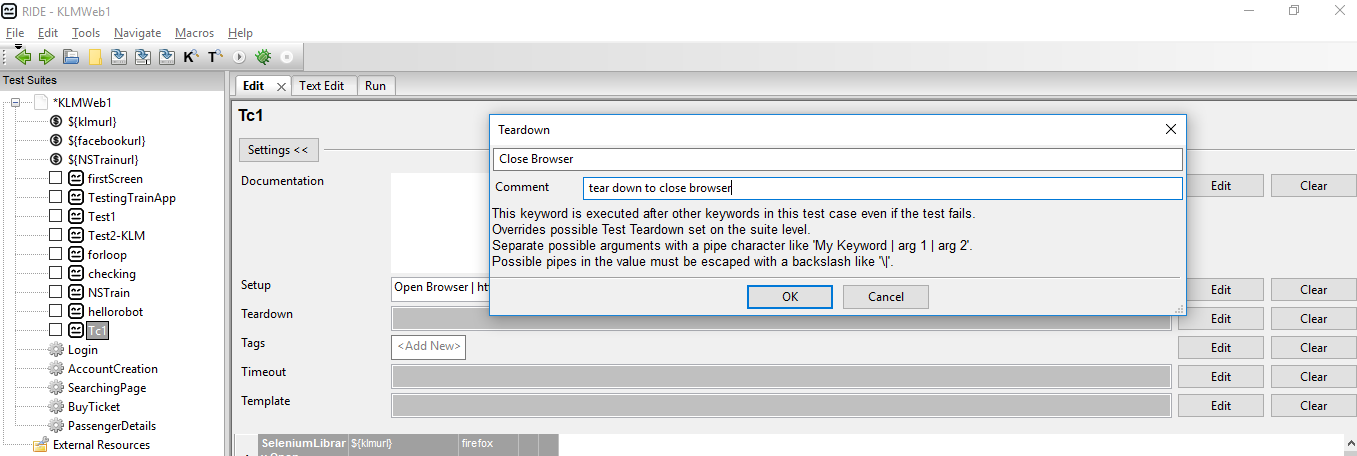
Arguments have to be separated with the pipe character (|).



Click OK to save the Setup. We have entered the Keyword **Open browser** and also added the arguments as shown above.

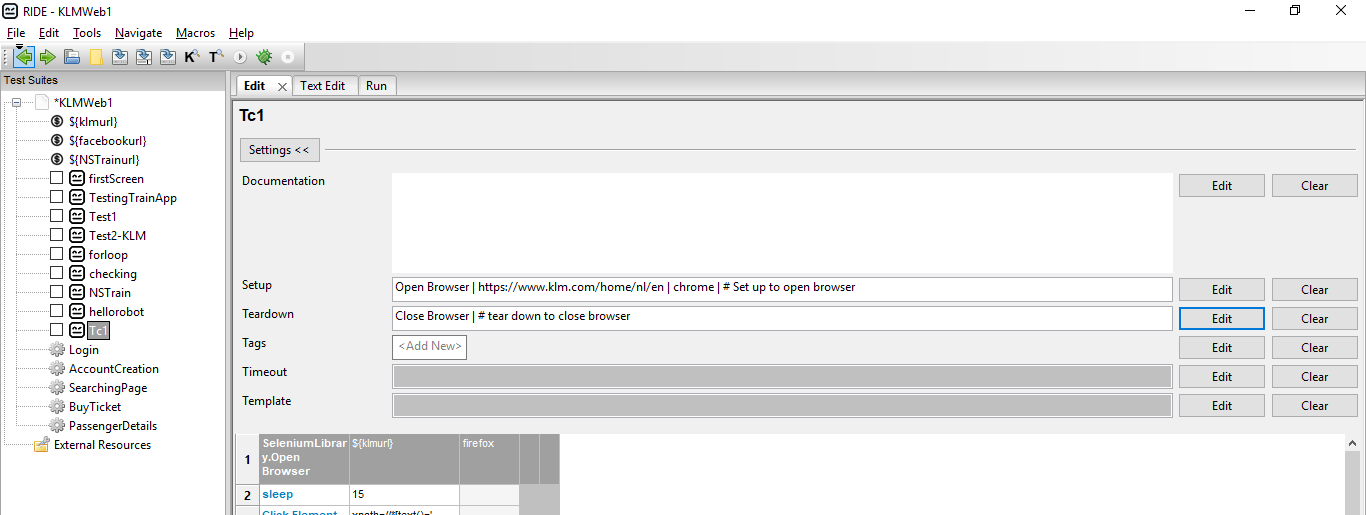
Now, we will enter the teardown case.

Click Edit for Teardown and enter the keyword.



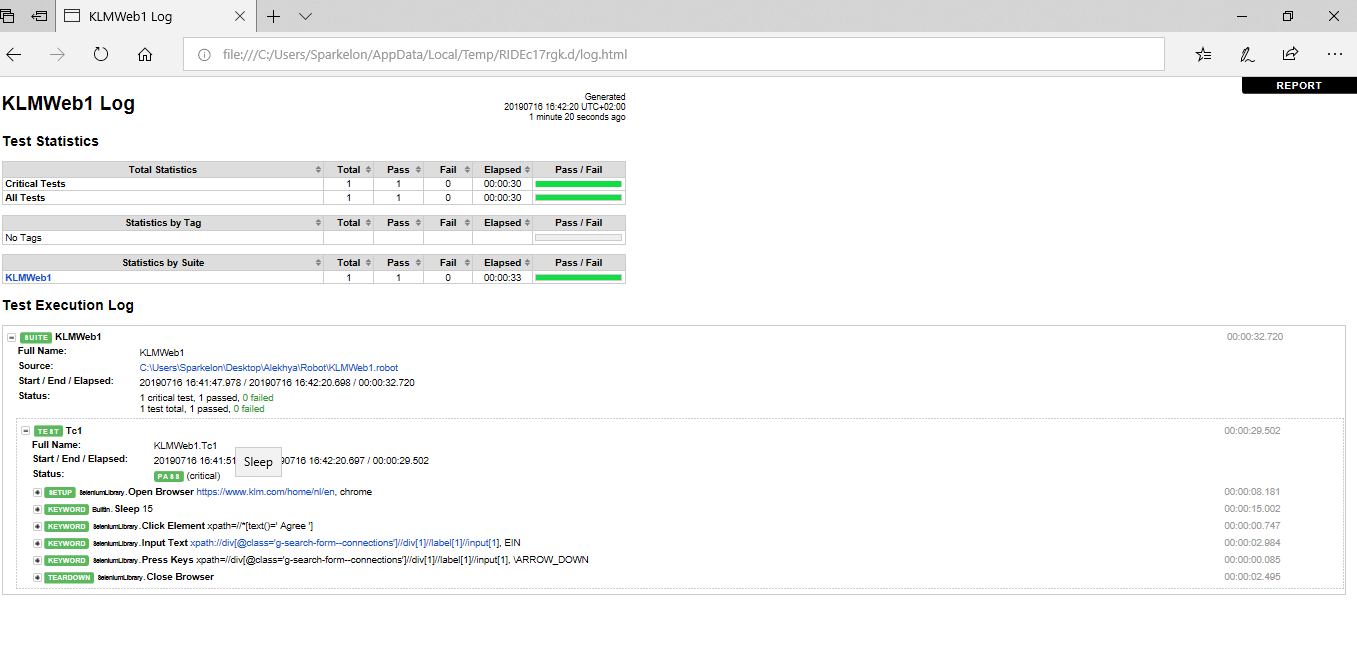
Click OK to save the teardown.

Now, we will enter the keywords for test case.



We have only Input Text in the test case. The opening and closing of the browser is done from Setup and Teardown Settings.

### **Test Execution Details**



* **Advantages:**
* Open source
* Very easy to install
* RF is application and platform independent
* User doesn't need a programming language to write a Robot Framework test case & run it
* Supports Keyword driven, Data-driven and Behaviour-driven (BDD) approaches
* The use of Selenium2 library in RF will help a person to automate the existing manual test cases of a project in a Rapid Phase
* Outstanding Reporting system: The RF automatically generates a report and html log after executing each build
* Robot Framework provides lots of libraries to test different application like Appium Library for Mobile Automation, Database Library for DB testing, Android Library etc.
* **Limitations:**
* Less flexibility - No features like author tag, skip test etc
* Robot lacks support for if-else, nested loops, which are required when the code gets complex