Robot Framework with Python

Table of Contents

[Introduction: 2](#_Toc31979304)

[Diagram: 2](#_Toc31979305)

[Softwares: 3](#_Toc31979306)

[Installation Process: 3](#_Toc31979307)

[Install Python 3](#_Toc31979308)

[Install Chrome Driver 8](#_Toc31979309)

[Install PyCharm 11](#_Toc31979310)

[Install Intellibot Plugin 12](#_Toc31979311)

[Project Creation: 13](#_Toc31979312)

[Scripts Execution: 15](#_Toc31979313)

[Advantages: 17](#_Toc31979314)

[Limitations: 22](#_Toc31979315)

|  |  |  |  |
| --- | --- | --- | --- |
| Author(s) | Version | Published Date | Modified |
| Venkateshwarlu Bhupathi | 1.0 | 02/07/2020 | Initial version |

# Introduction:

Robot Framework is an opensource automation framework which is developed by Peckka Klarck & Janne Harkonen in 2008.

Robot framework is a test automation framework for acceptance testing and Acceptance test-driven development. It follows different test case styles – keyword-driven, behavior-driven and data-driven for writing test cases. This feature makes it very easy to understand. 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.

# Diagram:

PyCharm

Robot Framework

Built-In-Library

Selenium- Library

DB Library

Etc

Python Interpreter

PIP

Web Drivers (Chrome, IE, Firefox, etc)

# Softwares:

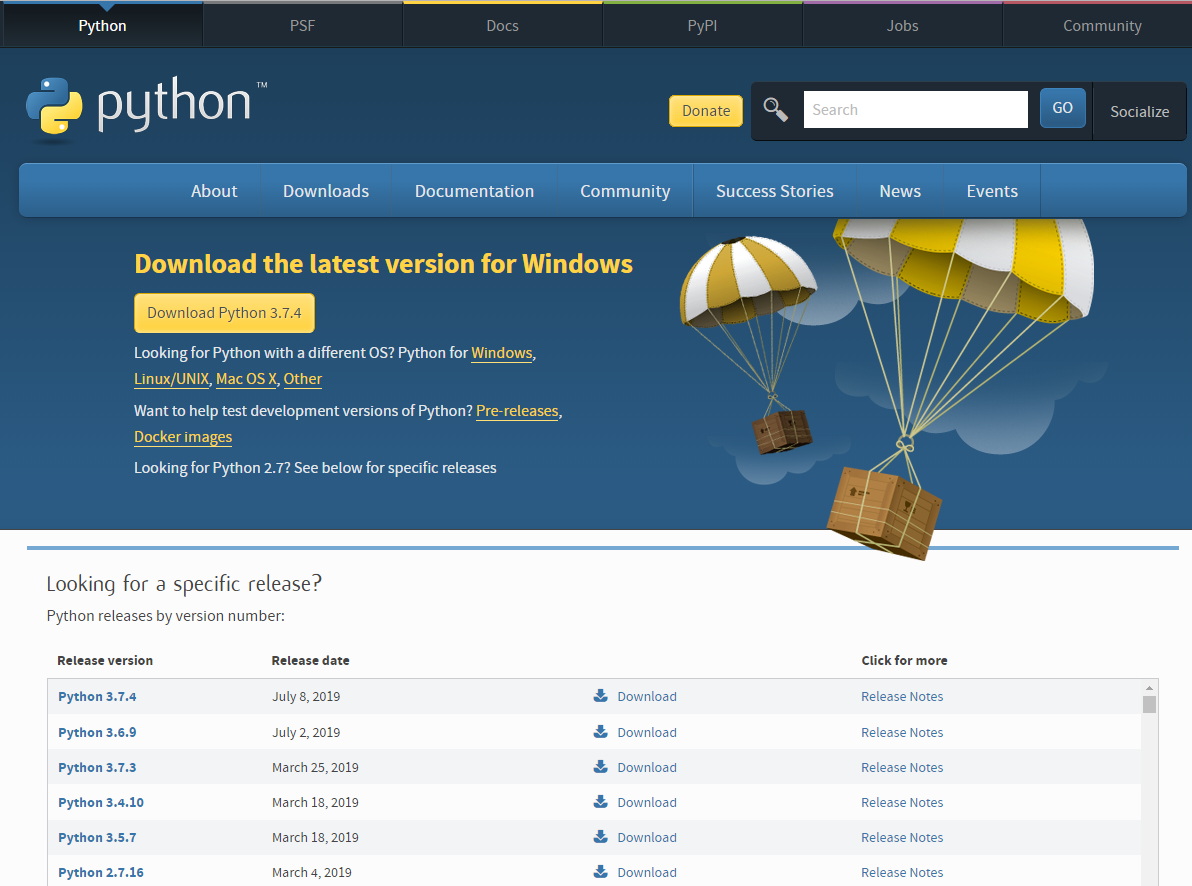
Below are the softwares are required before using Robot Framework with Python.

1. Python
2. Robot Framework Libraries
3. Browsers (Chrome, IE,etc)
4. Selenium Libraries
5. PyCharm
6. Intellibot Plugin(in PyCharm)

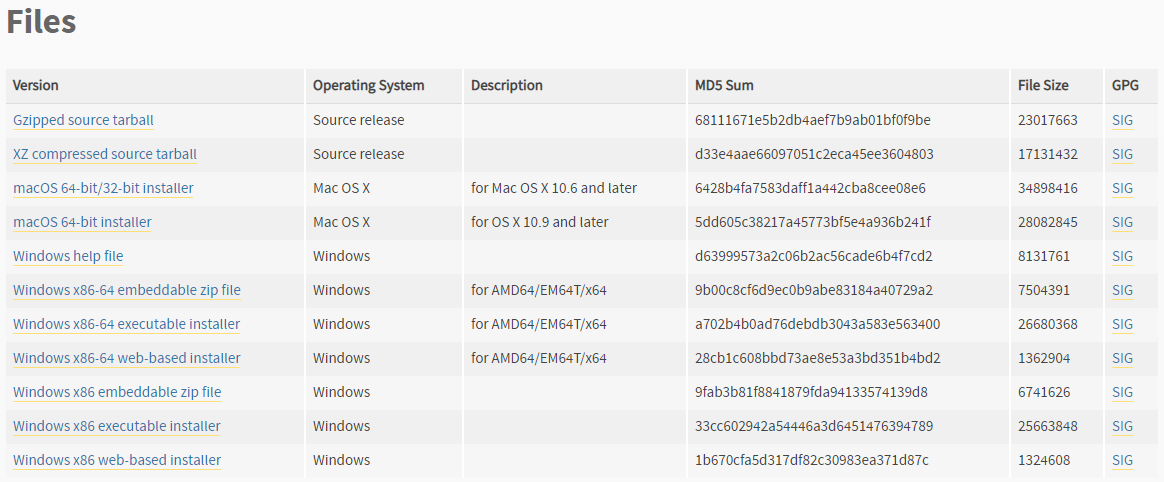
# Installation Process:

# Install Python

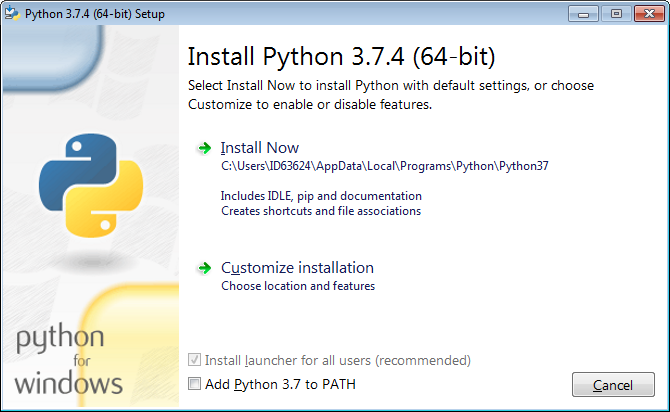
Use the following link - <https://www.python.org/downloads/> to download Python. The current latest version is 3.7.4. Go to the highlighted section and click on **Download** link to select the correct version in terms of operating system.



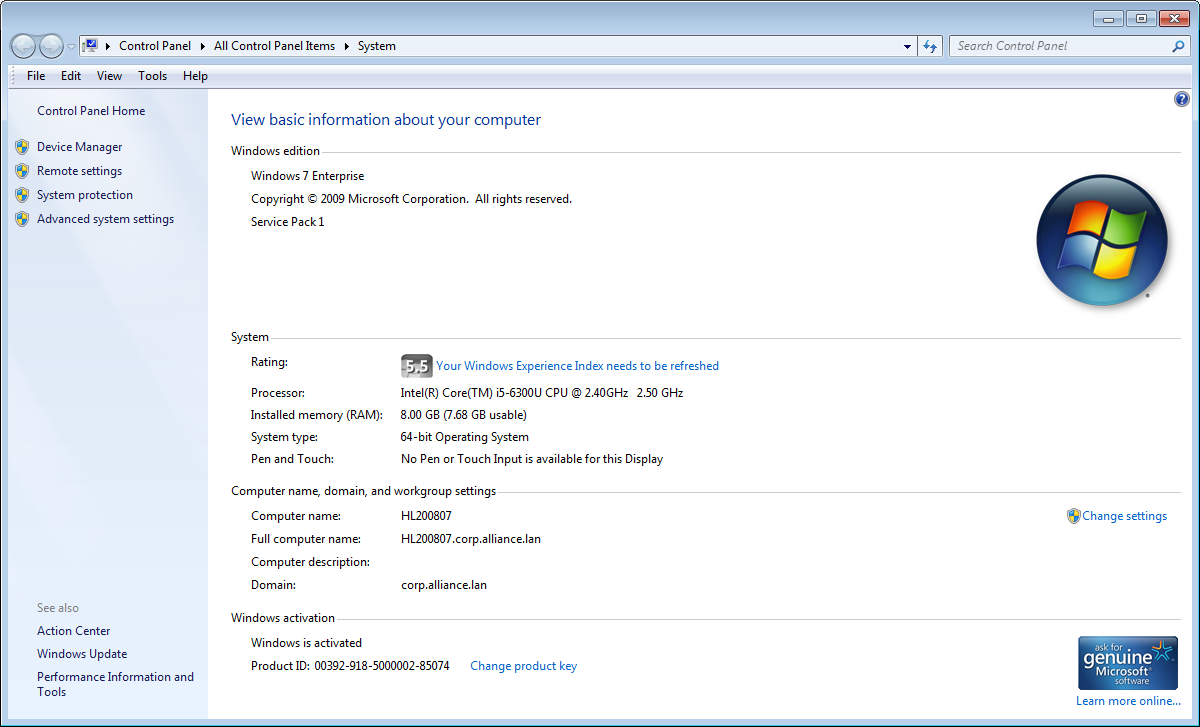
Below are versions for Python 3.7.4. System type can be found in above screen shots for computer name. For Windows 64-bit Operating System, please download **Windows x86-64 executable installer**.



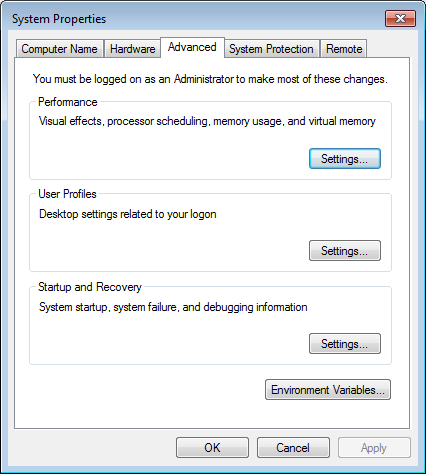
Run the executable file “python-3.7.4-amd64.exe” to install Python. In below popup, please check the box **Add Python 3.7 to PATH** before clicking on **Install Now** option. This step will add all required paths in Environment Variables.



In case this box is not checked during installation, please go to System again and click on “Advanced system settings”



Click on “Environment Variables…”



Find the variable “**Path**” in user variables section. Click on “**Edit…**” and add below two paths in Path variable. **The paths can be different based on where Python is installed in your machine.**

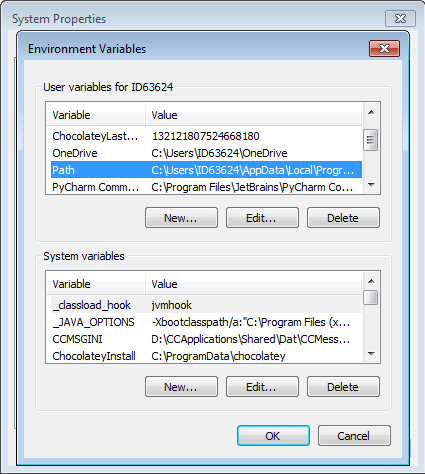
Ex:

C:\Users\ID\*\*\*\*\AppData\Local\Programs\Python\Python37\Scripts\

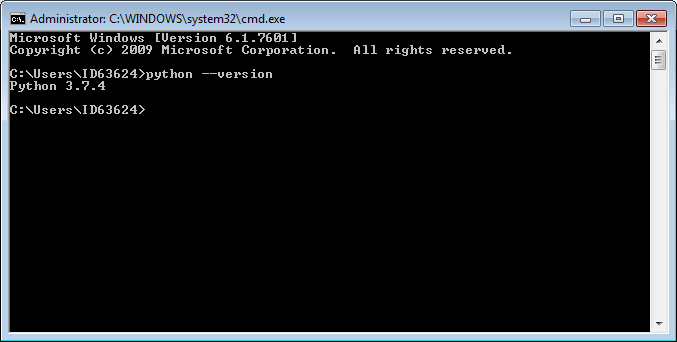
C:\Users\ID\*\*\*\*\AppData\Local\Programs\Python\Python37\

**Note: Paths are required to be separated by semicolon if multiple paths in Path variable.**

If the Path variable is not available in user variables section, click on “New…” to add Path variable and enter path values.



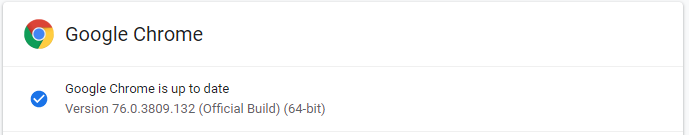
Go to Windows Start Menu -> All Programs -> Accessories and click on **Command Prompt**. In pop screen, enter **python --version** in the command line and hit <Enter>. If the Python is installed properly, the correct version information will be displayed as Python 3.7.4.



# Install Chrome Driver

Use the following link - <http://chromedriver.chromium.org/downloads> to download ChromeDriver.

For Google Chrome, please check your Chrome browser version through Help -> About Google Chrome



If you are using Chrome version 76, please **download** [ChromeDriver 76.0.3809.126](https://chromedriver.storage.googleapis.com/index.html?path=76.0.3809.126/).

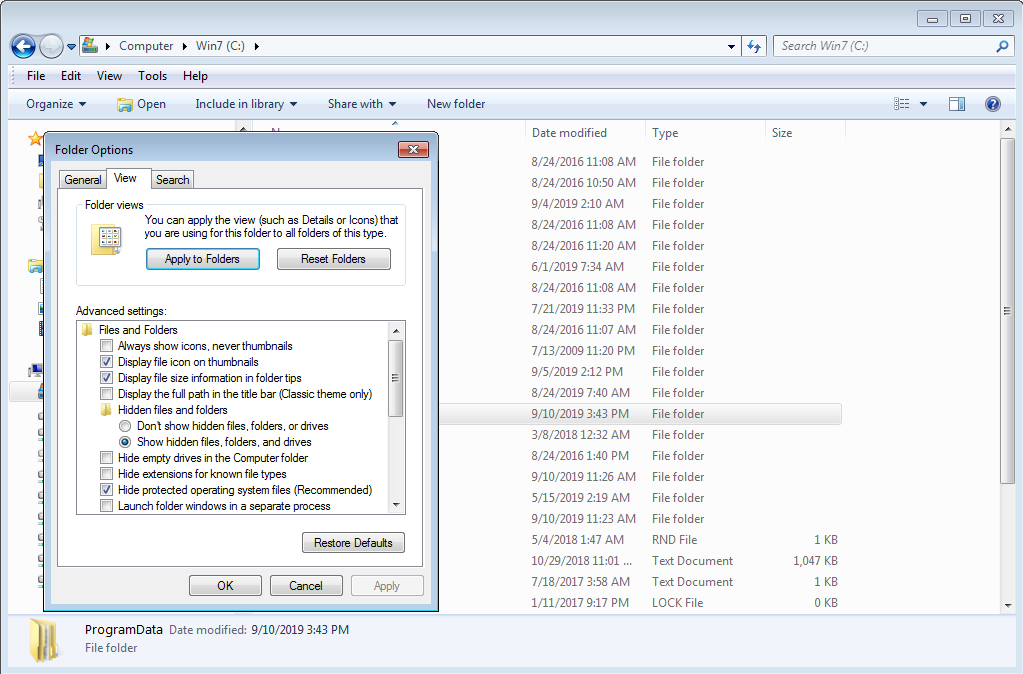
Once it’s downloaded, please keep this file in your local drive and add the folder **path** to Environment Variables. **This path can be different based on where ChromeDriver is stored in your machine.**

Ex:

C:\Users\ID63624\AppData\Local\Programs\Chromedriver

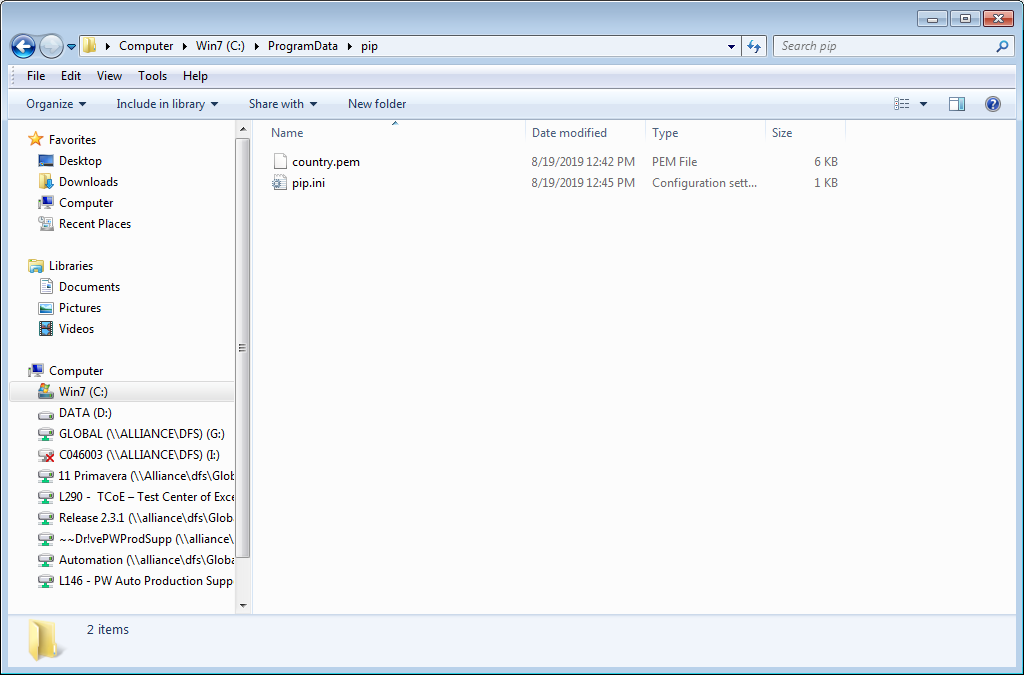
Please find more information related to ChromeDriver in the following link - <http://chromedriver.chromium.org/getting-started>.Install Robot Framework and Selenium Library

Go to Tools -> Folder options… -> View and change the Advanced settings for **Hidden files and folders** to **Show hidden files, folders, and drivers**. Then hidden folder C:\ProgramData can be displayed.



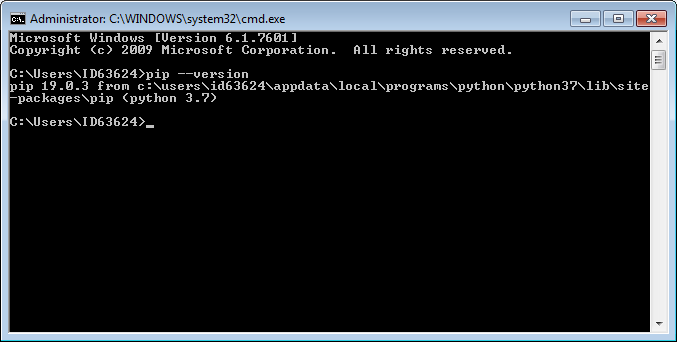
Create a new pip folder under the existing folder **C:\ProgramData** and copy the below two files in pip folder.

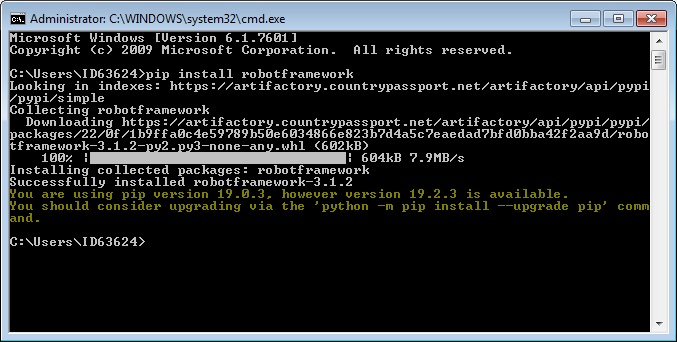


Go to Command Prompt, enter **pip --version** and hit <Enter> to check if pip command is installed and configured properly.

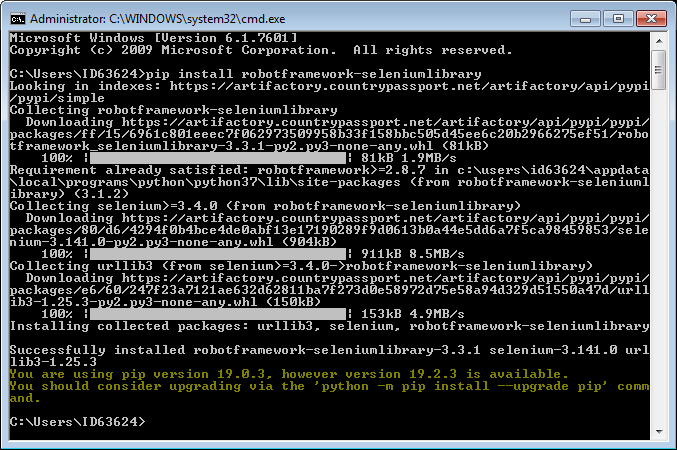
The current pip version is 19.0.3



Enter **pip install robotframework** and hit <Enter> to install Robot Framework.



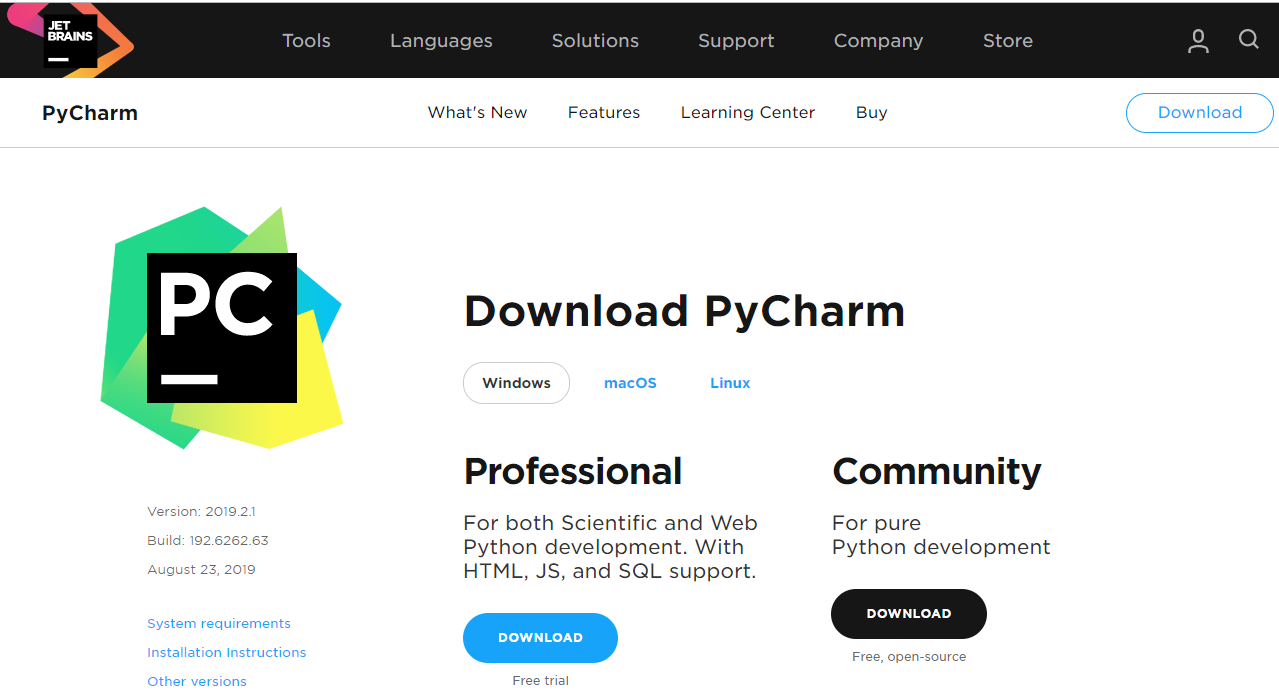
Enter **pip install robotframework-seleniumlibrary** and hit <Enter> to install Robot Framework Selenium Library.



# Install PyCharm

Use the following link - <https://www.jetbrains.com/pycharm/download/#section=windows> to download PyCharm which is one of integrated development environment (IDE) tools for Python and Robot Framework.

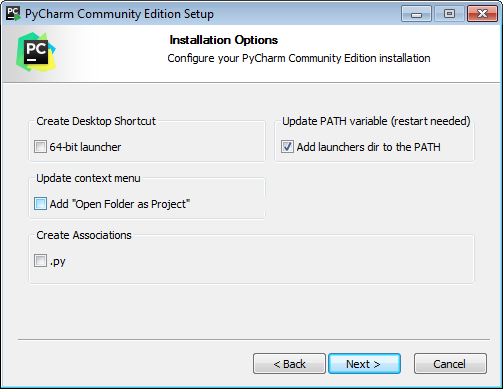
Select **Community version** to download. The current version is 2019.2.1.



During PyCharm installation, please check the box for **Add launchers dir to the PATH** and below information will be added into Environment Variables.

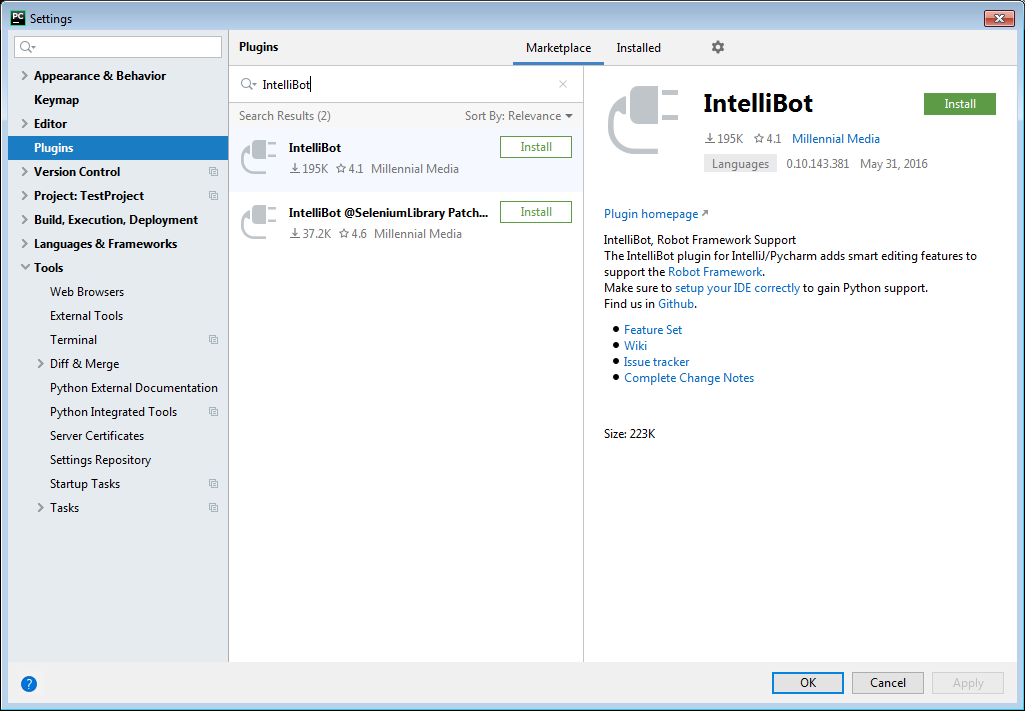
**%PyCharm Community Edition%**

Restart your machine after the installation.



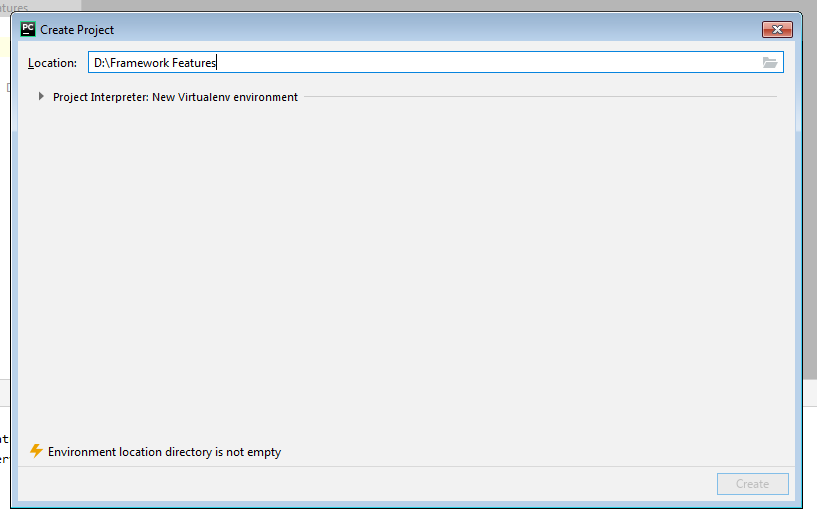
# Install Intellibot Plugin

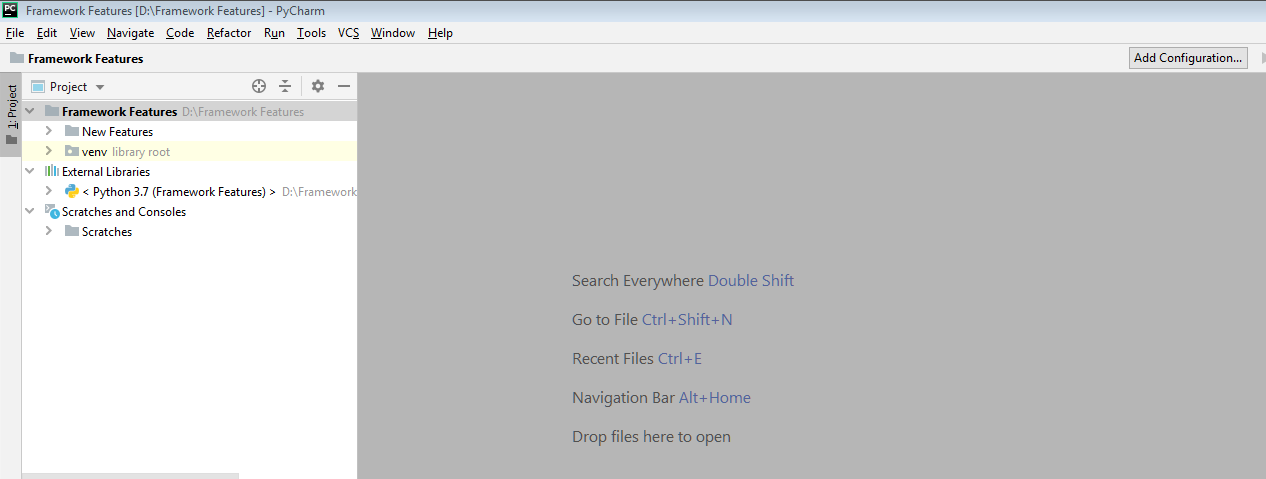
Go to **Plugins** in the same Settings window and search for **Intellibot** and click on Install button to install this plugin. Restart IDE.



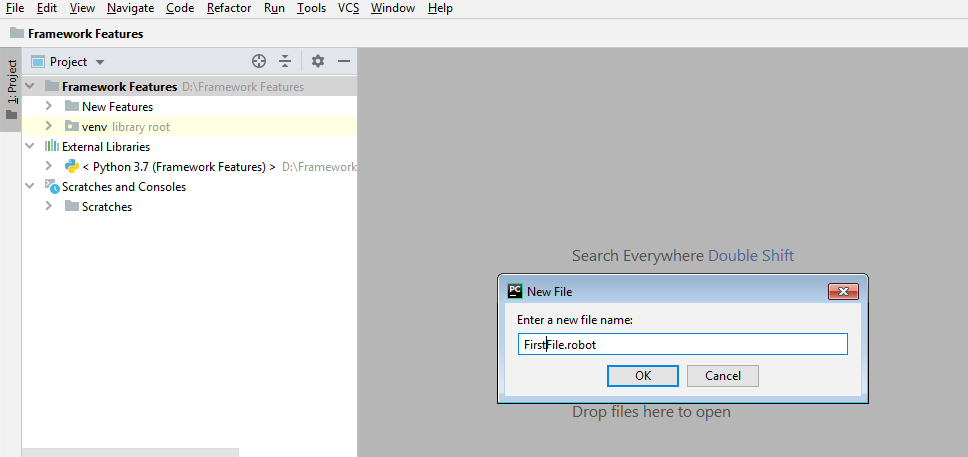
# Project Creation:

Go to File-> Create Project



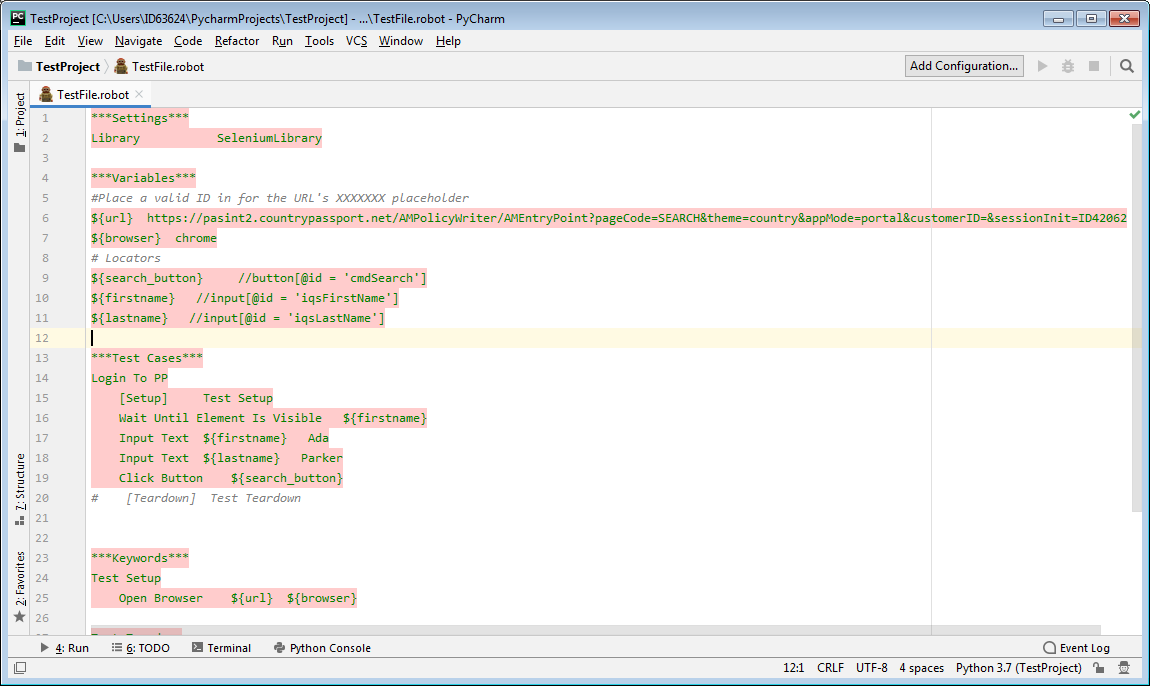


Right click on Framework Features, in the cascading menu, select New -> File. Create a test file “ **FirstFile.robot**



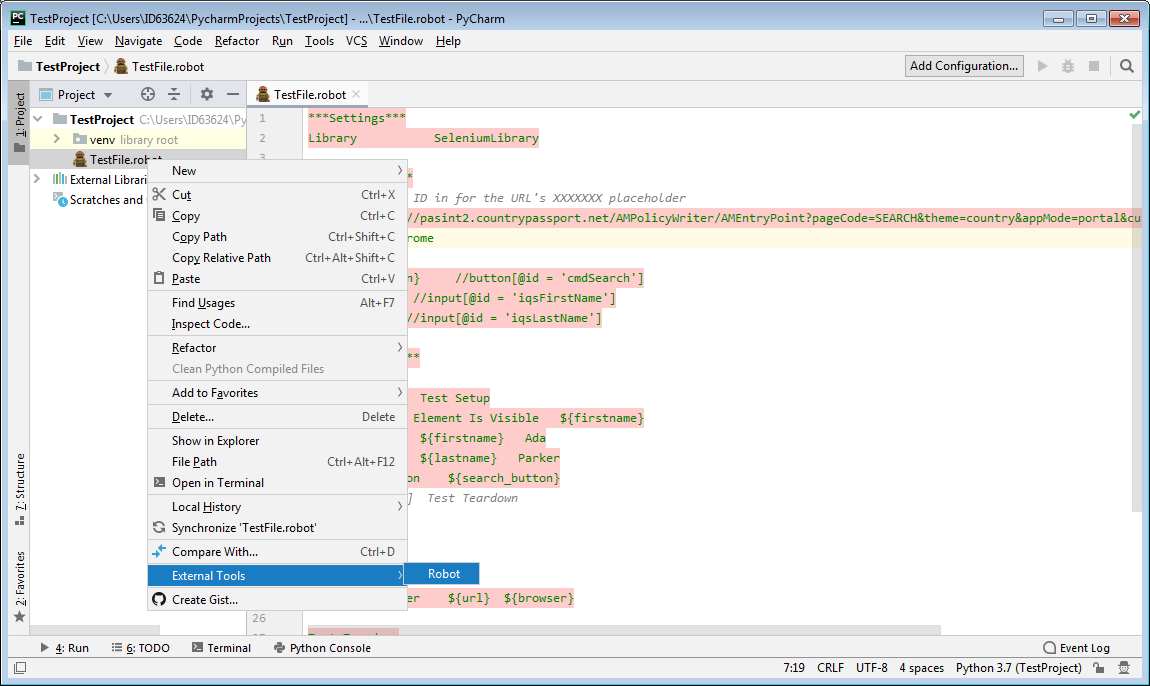
Write the script in .robot file .Robot file consists fout different modules like below.

1. **Settings** - In this section libraries will be defined
2. **Varaibles**– Test case related variables will be declared in this section
3. **Test cases**- Test cases will be defined in this section
4. **Keywords** – Keywords will be defined in this section

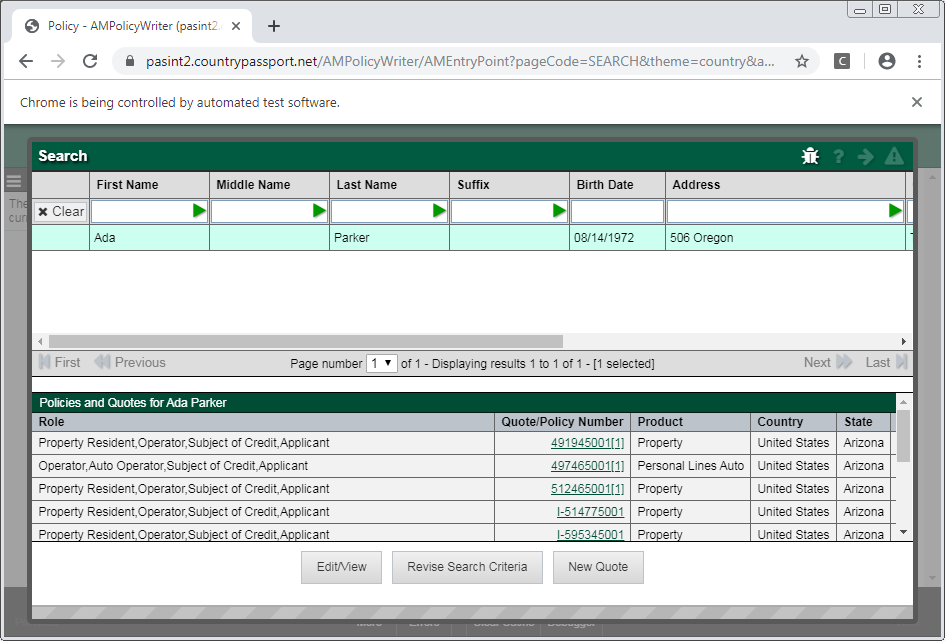


# Scripts Execution:

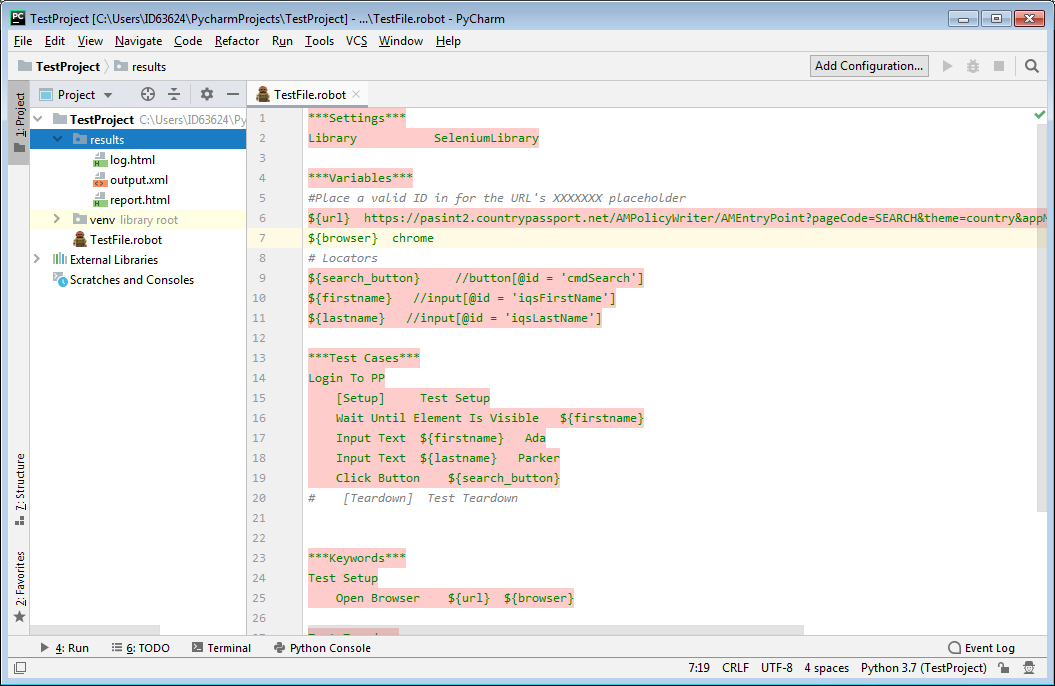
Right click on TestFile.robot ->External Tools ->**Robot**. Click on Robot menu and execute this robot script.

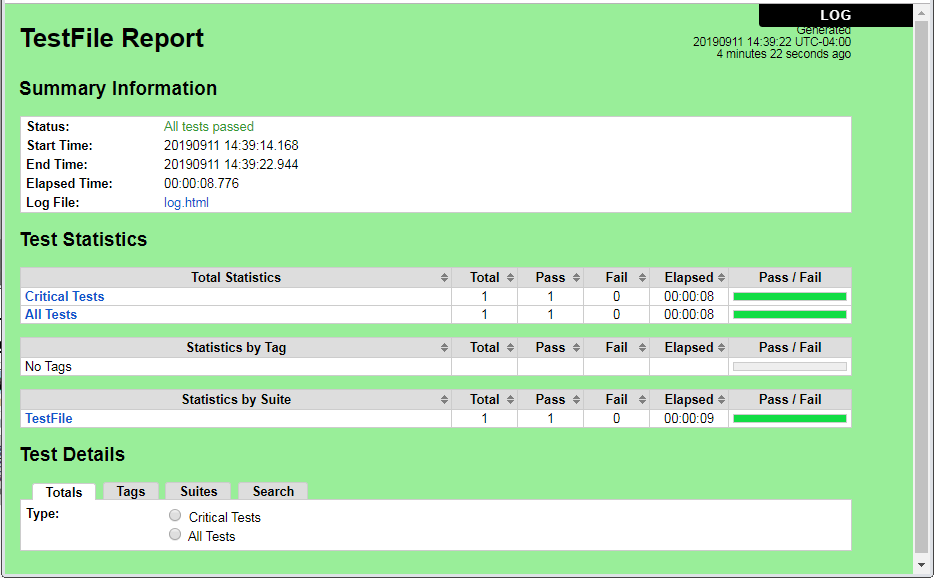


PW Direct Link should be launched and search for a customer. Below is the screen shot of last step for this test file.



Once the execution is complete, a results folder with log file, output file and report file will be created. Right click on report.html -> **Open in in Brower** -> Chrome.



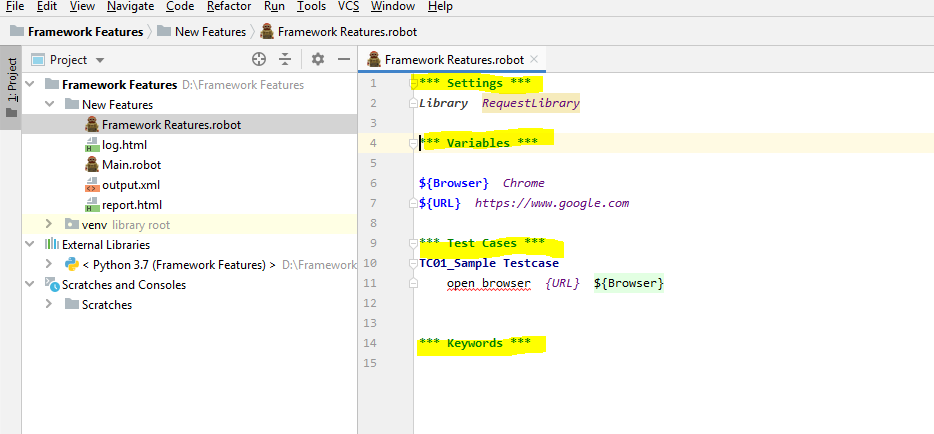


# Advantages:

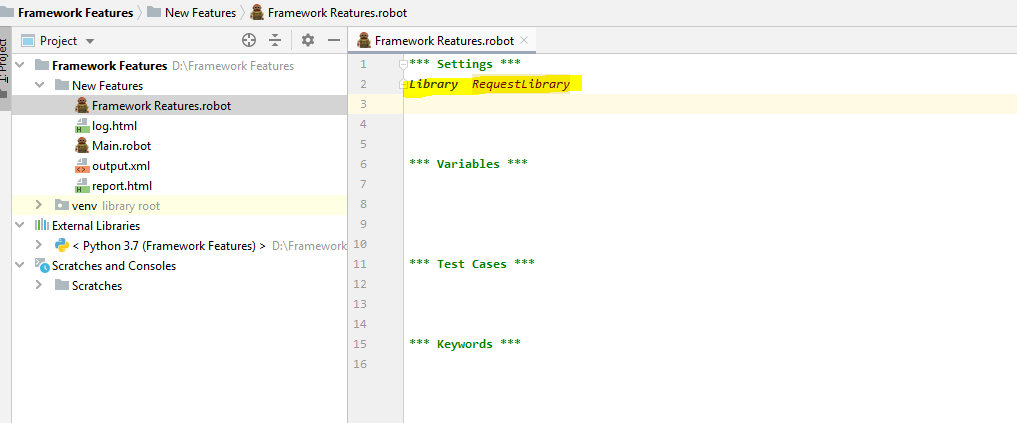
1. Robot Framework is very easy to install and helps in creating and executing test cases. Any new comer can easily understand and does not need any high level knowledge of testing to get started.
2. Robot framework will support keyword-driven, behavior-driven and data-driven style of writing test cases.
3. Robot framework will support for external libraries. Most used is Selenium Library, which is easy to install and use in robot framework.
4. Robot framework will support API testing
5. Robot Framework will support Mobile testing
6. Robot Framework will support Parallel Execution
7. Robot Framework will support continuous execution (CI/CD Pipeline)
8. Robot Framework will support windows applications automation
9. Robot Framework will support Headless browser testing
10. Robot Framework is Platform and Application Independent

**Easy to write the Scripts:** Robot framework scripts are in simple English language this can easy to understand by everyone.

Script is containing in different modules like Settings, Variables declaration, Test cases declaration, Keywords declaration.

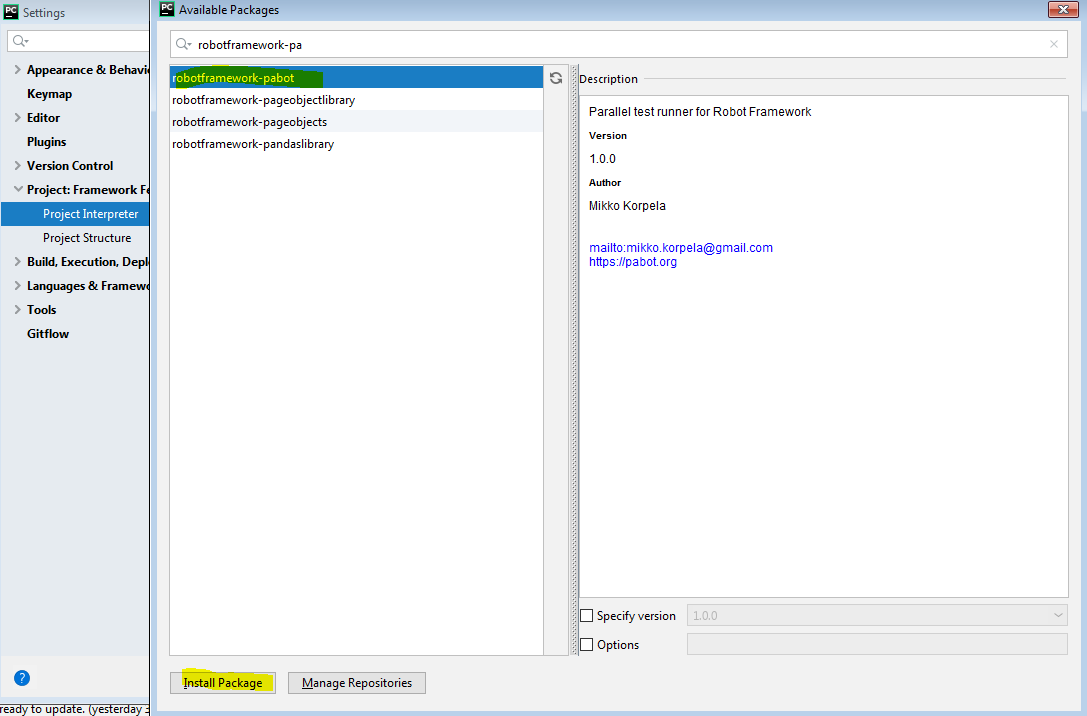


**API Testing:** Robot framework will support API testing by using Request library.

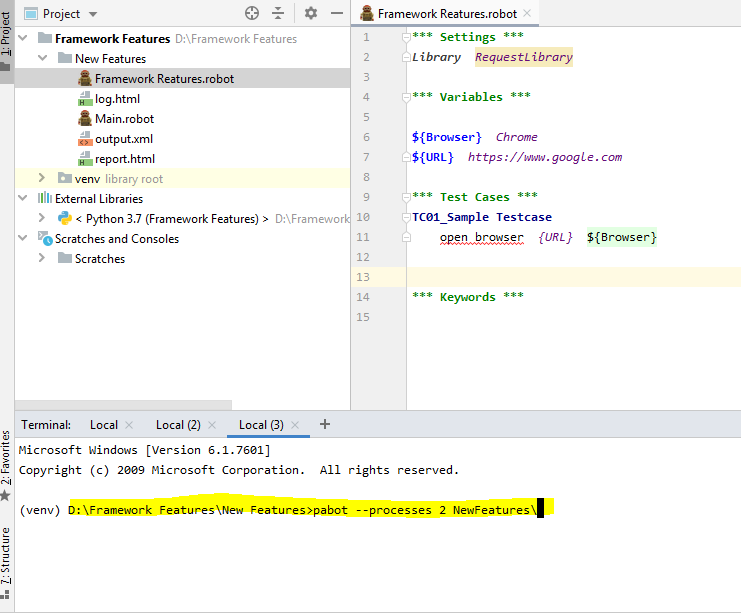


**Parallel Execution:** Robot framework will support Parallel execution by installing Pabot Plugin

In PyCharm Go to File->Settings-> Project Interpreter install Robotframework-pabot Plugin.



Once Plugin installation done then execute the script by using below commands in Terminal

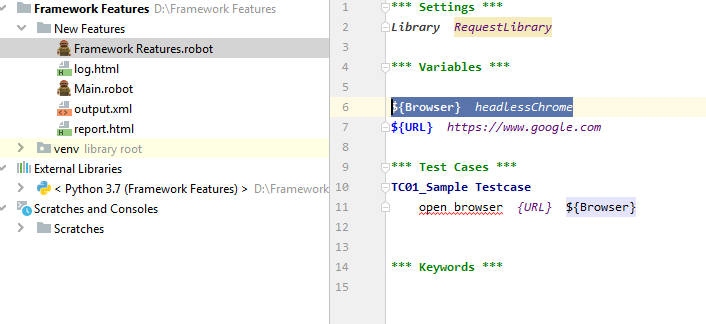


Syntax: **pabot –processes 3(no of scripts you want to execute) Folder Name( scripts location)\**

In the above example I want to execute three scripts parallelly and these three scripts are in New Features folder.

**Headless browser Testing:** (Execution will be done in backend) Robot framework will support headless browser testing.

**Syntax: ${Browser}** *headlessChrome*



In the above example just, we have to declare chrome details as headlessChrome. Headless browser execution is fast when compare to normal execution.

**Mobile Testing:** Robot Framework will support mobile testing by using Appium library.

Robot

Library

Test

Suite

HTML

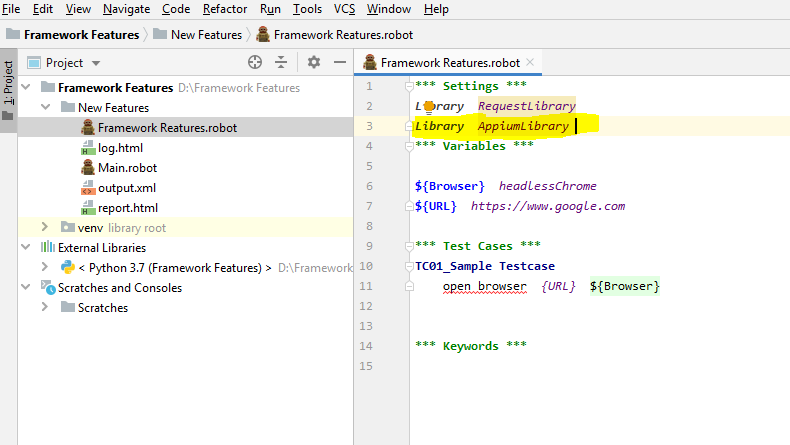
Report

Appium Library

ROBOT

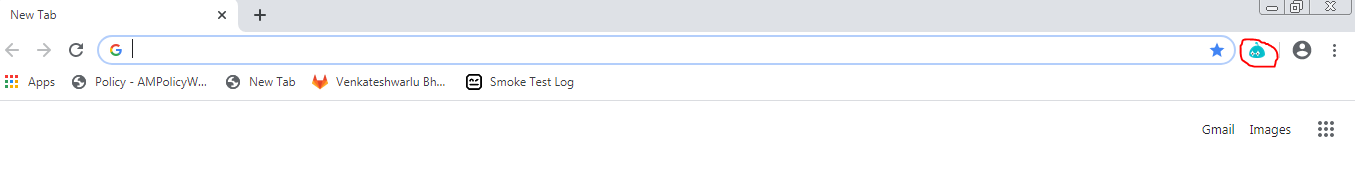
Appium Python Client

Mobile

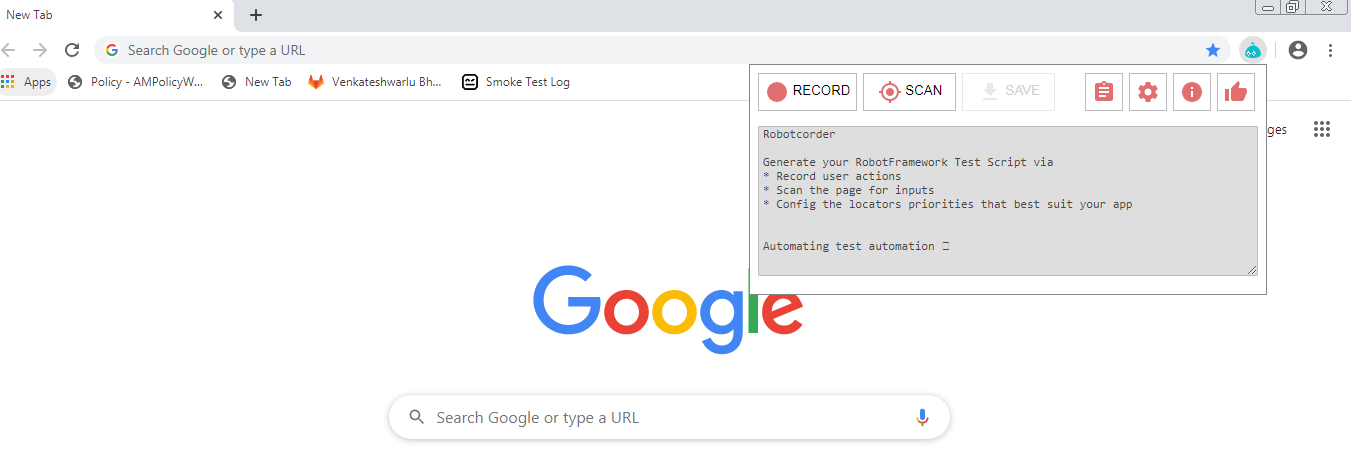


**Recording:** We are able to record the scripts in chrome browser by Robot Corder plugin

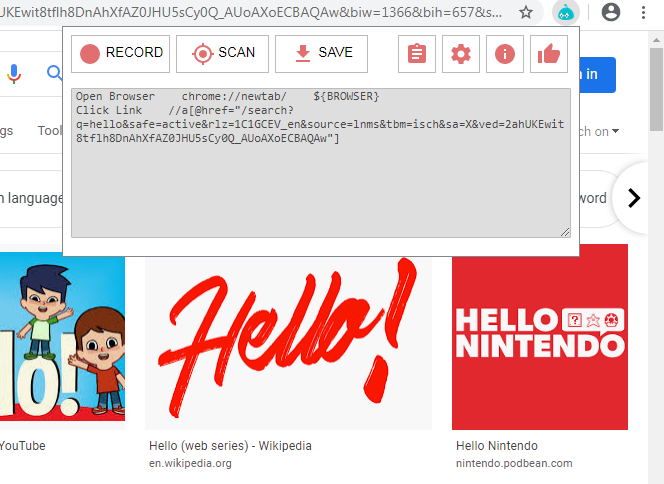
Robot Corder should be install in Chrome browser.



Click on highlighted icon in chrome browser.



Click on Record Button to record the steps, once recorded then copy the script and that can be use in Pycharm.



# Limitations:

1. Robot lacks support for if-else, nested loops, which are required when the code gets complex
2. HTML reports customization is tricky