Windows Application Automation

Contents

[A. Installation 2](#_Toc9673006)

[1. Python 2](#_Toc9673007)

[2. Windows Application Driver 2](#_Toc9673008)

[3. WhiteAppLibrary project 3](#_Toc9673009)

[1. WhiteLibrary 3](#_Toc9673010)

[2. SikuliLibrary 5](#_Toc9673011)

[3. AppiumLibrary 6](#_Toc9673012)

[B. Customization 10](#_Toc9673013)

[1. SikuliLibrary 10](#_Toc9673014)

[C. Common Question 11](#_Toc9673015)

[1. How to add a path into Path environment variable? 11](#_Toc9673016)

[2. How to create a new Interpreter in Pycharm? 12](#_Toc9673017)

# Installation

## 1. Python

Step 1: Download and install Python latest version (ver. 3.7.3 as May 2019) at: <https://www.python.org/downloads/>



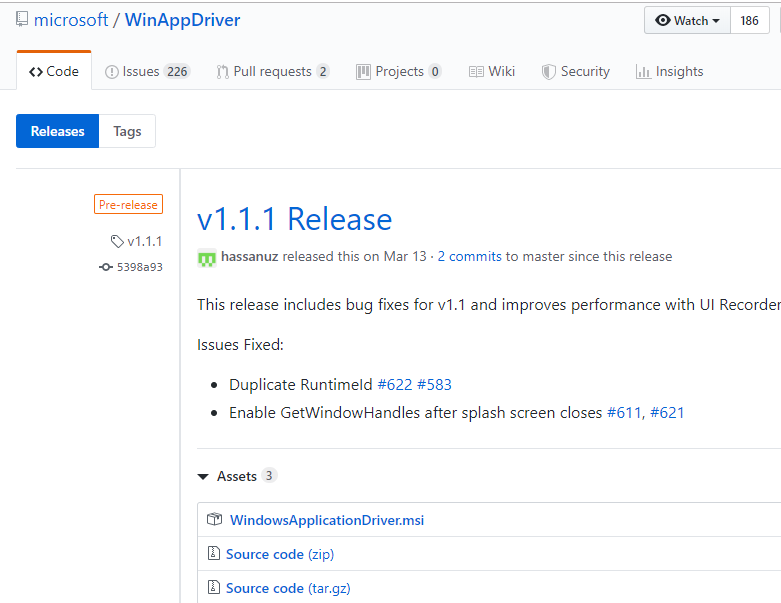
Step 2: Add its installation path into PATH environment variable. Go to **Common Question** section to know how to “[Add Path environment variable](#_1._How_to)”.

## 2. Windows Application Driver

Windows Application Driver (WinAppDriver) is a service to support Selenium-like UI Test Automation on Windows Applications. This service supports testing Universal Windows Platform (UWP), Windows Forms (WinForms), Windows Presentation Foundation (WPF), and Classic Windows (Win32) apps on Windows 10 PCs. WinAppDriver complies to the JSON Wire Protocol standard and some application management functionalities defined by Appium.

More information should be read from its GitHub: <https://github.com/microsoft/WinAppDriver>

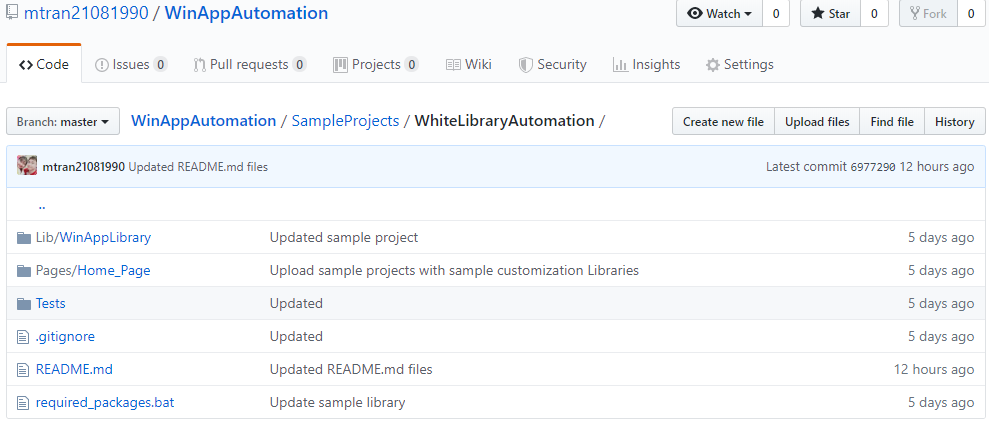
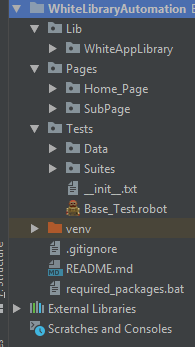
Step 1: Download and install WinAppDriver latest version (ver. 1.1.1 as May 2019). WinAppDriver can be downloaded at: <https://github.com/Microsoft/WinAppDriver/releases>



## 3. WhiteAppLibrary project

Step 1: Pull (or download) this project from GitHub:

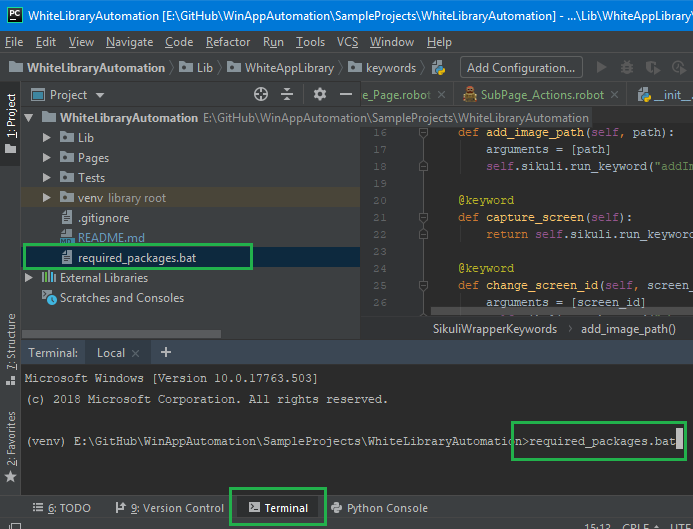
<https://github.com/mtran21081990/WinAppAutomation/tree/master/SampleProjects/WhiteLibraryAutomation>

**Project Details**:

* Lib > WinAppLibrary: contains the customized library which is extended from WhiteLibrary
* Pages: contains resource files for each module of an application. With a desktop application, each Window should be considered a module.
* Tests > Data: contains test data for each test suites.
* Tests > Suites: contains test suites.

Step 2: Open a Python IDE (prefered PyCharm), then open the WhiteLibraryAutomation project. Create a new Interpreter for this project in File > Settings > Project Interpreter. Restart IDE (if need). Go to **Common Question** section to know how to “[Create a new Interpreter on Pycharm](#_How_to_create)”.

Step 3: Install all required packages for the project:

- Open **Terminal**

- Input “required\_package.bat”

- Press ENTER.

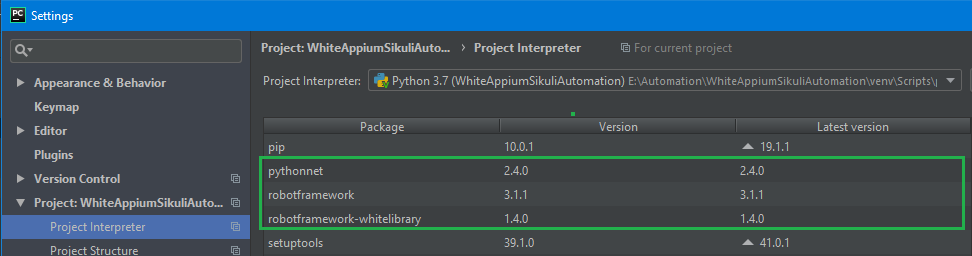
🡺 This will install the following libraries: robotframework, robotframework-SikuliLibrary, robotframework-appiumlibrary and pyyaml.

## 1. WhiteLibrary

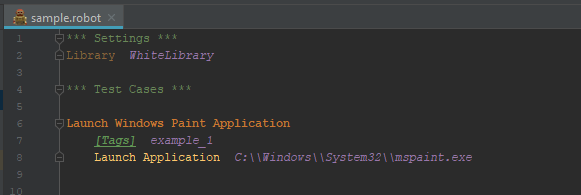
Step 1: In an existing or new python project, open Terminal and execute following command:

pip install --upgrade robotframework-whitelibrary

Step 2: Verify installed packages in Files > Settings: pythonnet, robotframework (if not) and robotframework-whitelibrary.



Step 3: Verify WhiteLibrary by composing and running a simple test case as follow:

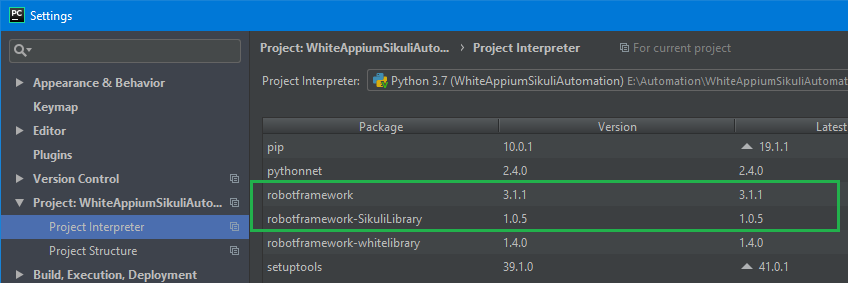


## 2. SikuliLibrary

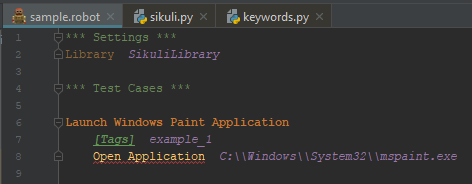
Step 1: In an existing or new python project, open Terminal and execute following command:

pip install robotframework-SikuliLibrary

Step 2: Verify installed packages in Files > Settings: robotframework (if not) and robotframework-SikuliLibrary.



Step 3: Verify SikuliLibrary by composing and running a simple test case as follow:

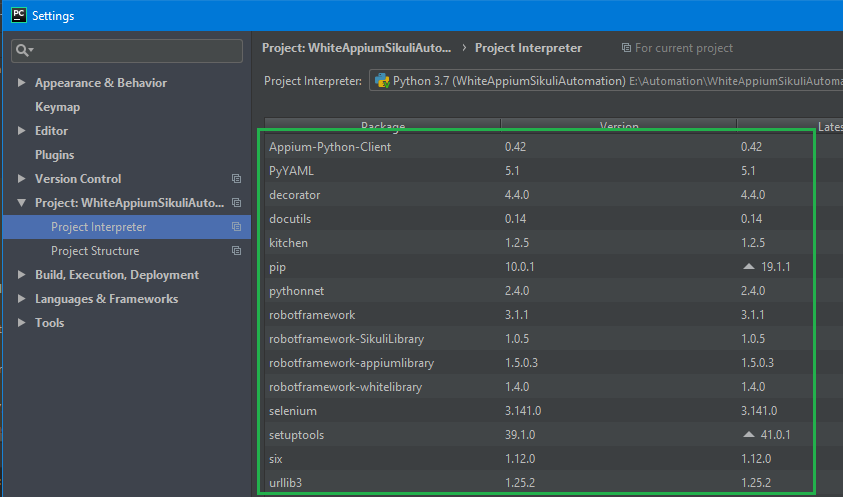


## 3. AppiumLibrary

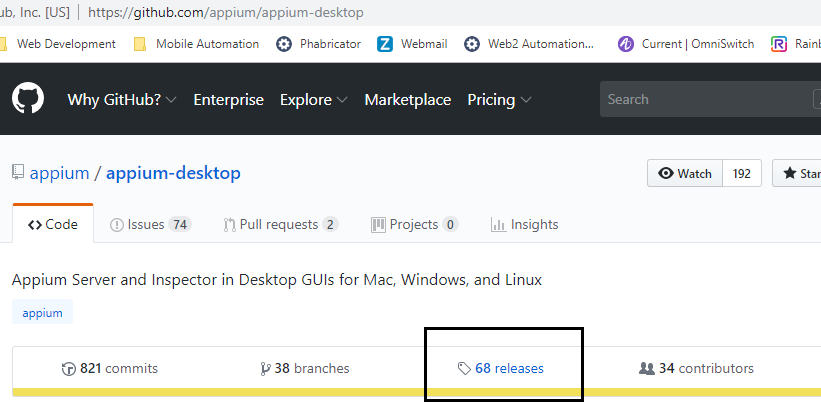
Step 1: In an existing or new python project, open Terminal and execute following command:

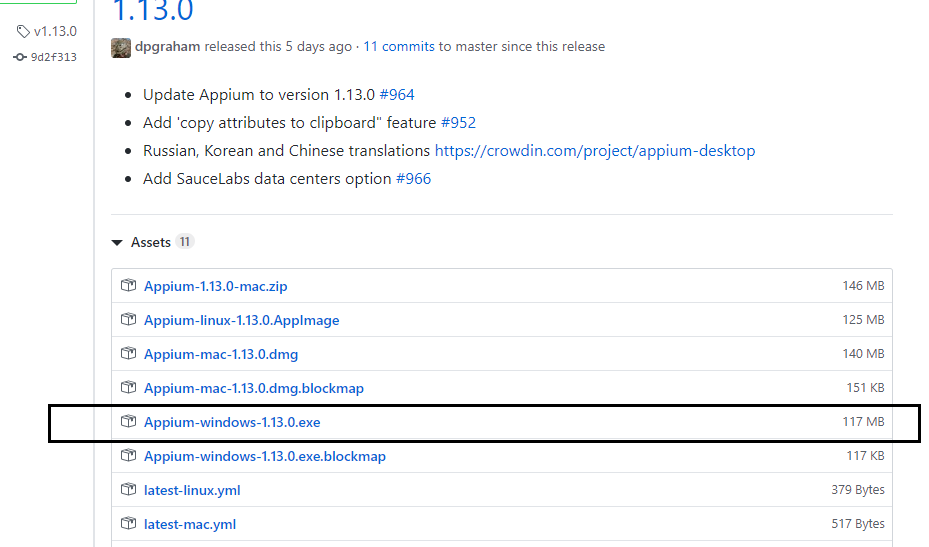
pip install robotframework-appiumlibrary

Step 2: Verify installed packages in Files > Settings: robotframework (if not), kitchen, decorator, urllib3, selenium, six, docutils, Appium-Python-Client and robotframework-appiumlibrary.

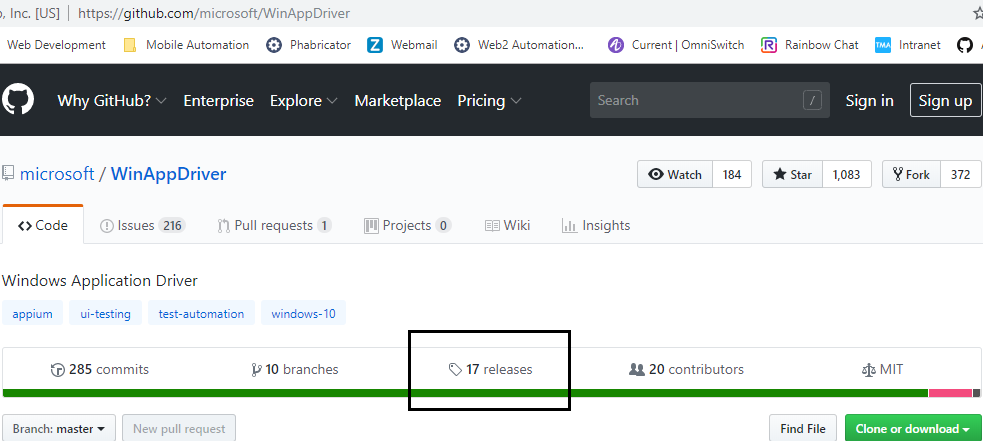


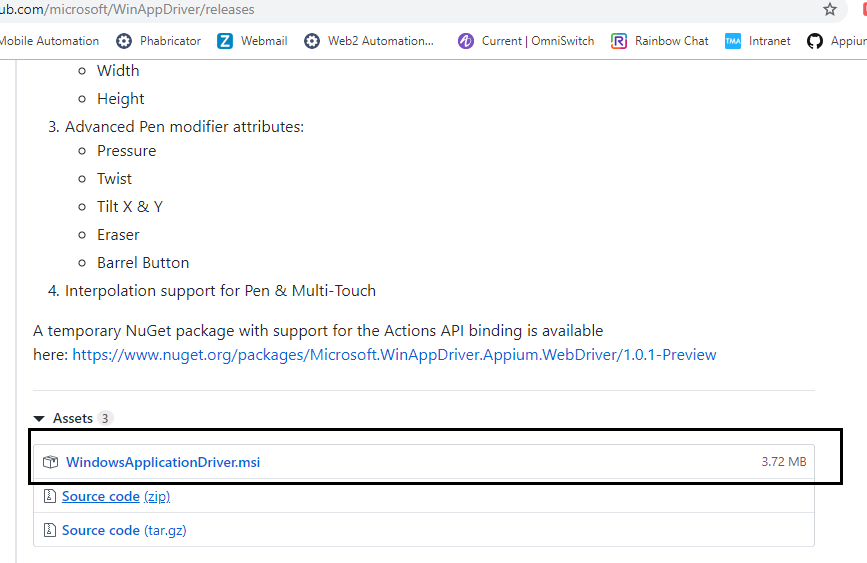
Step 3: Install Appium Desktop application by going to <https://github.com/appium/appium-desktop> and click on **Release** tab to download the latest version of Appium.



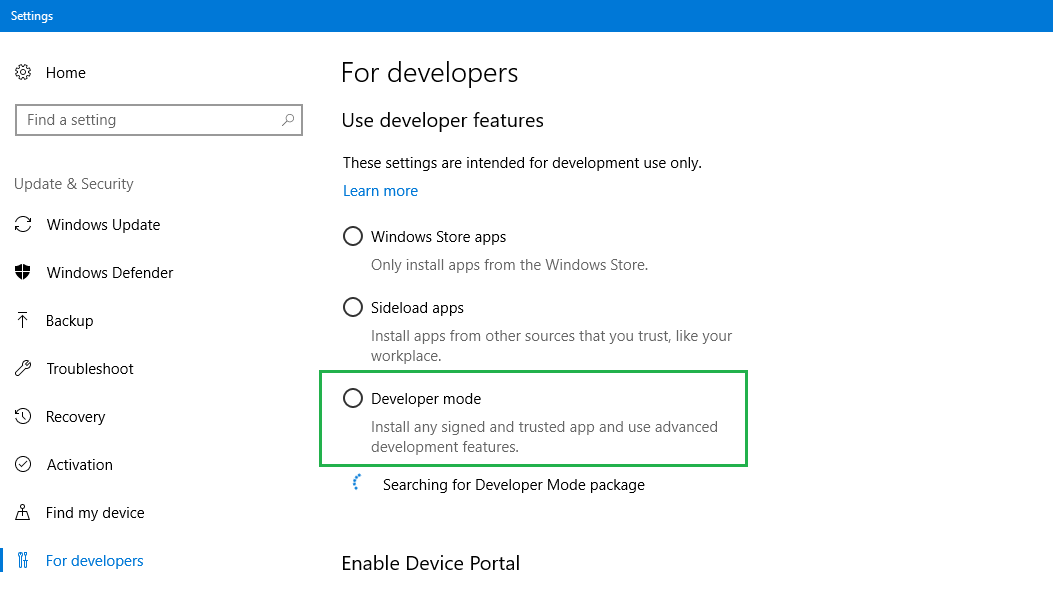


Step 4: Install Windows Application Driver (WinAppDriver) by going to <https://github.com/microsoft/WinAppDriver> and click on **Release** tab to download the latest version of WinAppDriver.

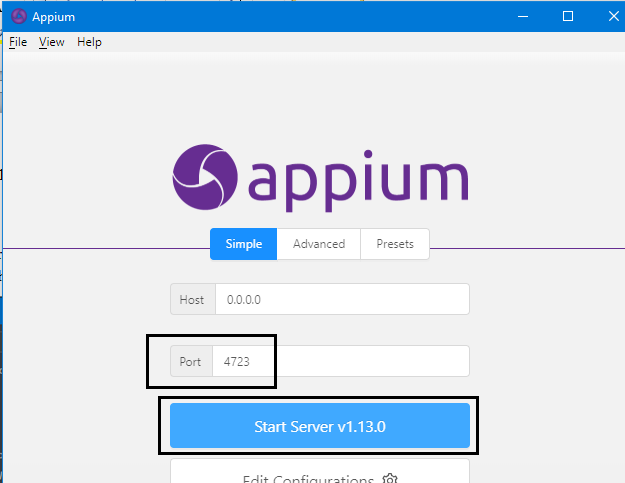




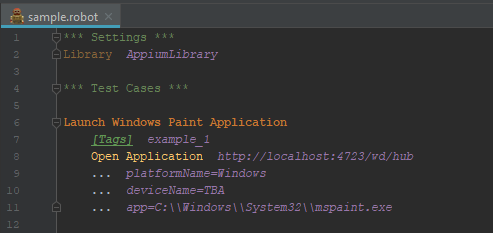
Step 5: Enable Developer mode on Windows by going to Settings:



Step 6: Install Appium Desktop & WinAppDriver applications. Run Appium Desktop application and Start Server (default port is 4723).



Step 7: Verify AppiumLibrary by composing and running a simple test case as follow:



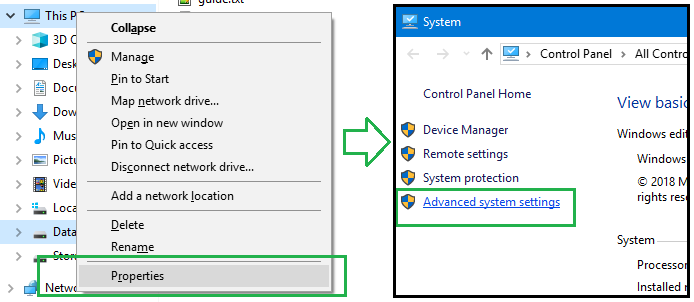
# Customization

## 1. SikuliLibrary

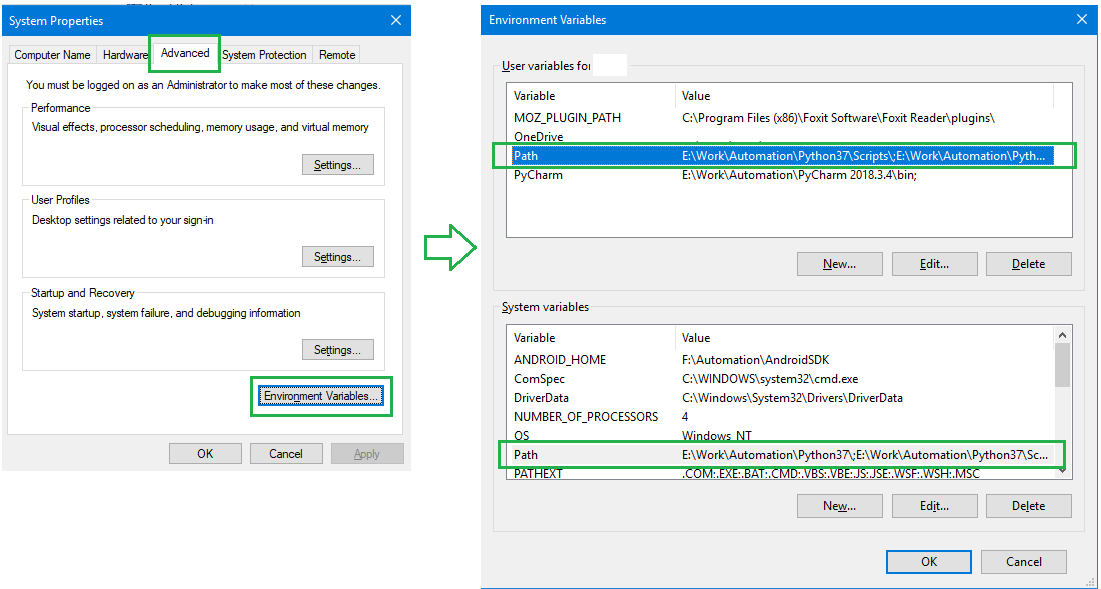
# Common Question

## How to add a path into Path environment variable?

Step 1: Right click on **My Computer** and select “Properties” to open **System** dialog. On **System** dialog, click “Advanced system settings” to open **System Properties** dialog.

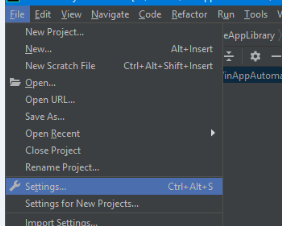


Step 2: On **System Properties** dialog, open “Advanced” tab, click “Environment Variables” button to open **Environment Variables** dialog. Add the new path into a ‘Path” variable (one is enough).

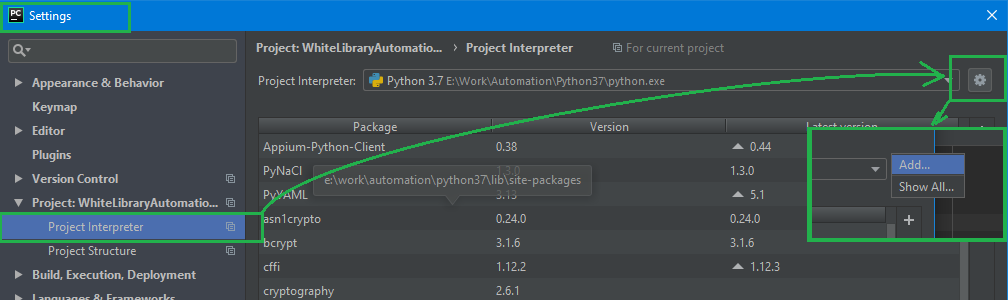


## How to create a new Interpreter in Pycharm?

Step 1: Open “File” on top menu and click “Settings” to open **Settings** dialog.



Step 2: On **Settings** dialog, open “Project Interpreter”, on the right section click on the **Gear** icon and select “Add” to open **Add Python Interpreter** dialog.



Step 3: On **Add Python Interpreter** dialog, select “New environment” radio button and uncheck any checkboxes. Click **OK** button.

