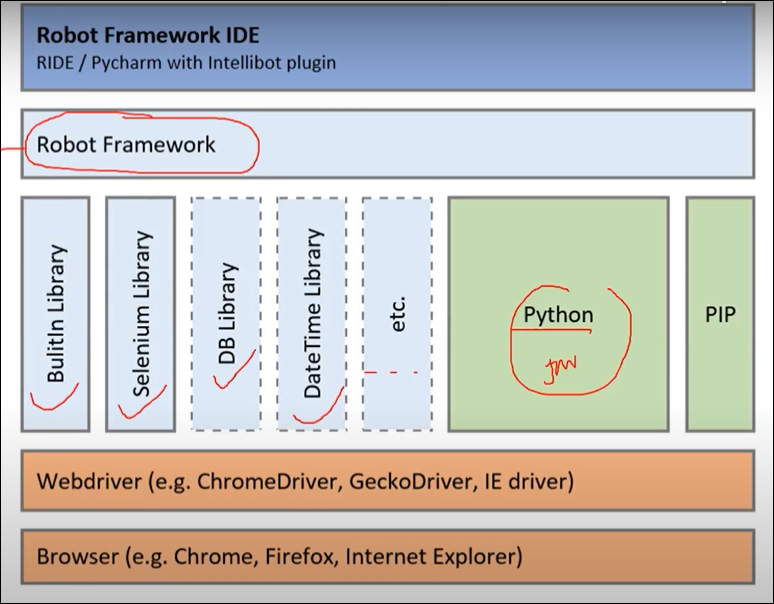
A screenshot of a computer program

Description automatically generated with low confidence

1. Robot framework is a generic test automation framework for software acceptance testing.
2. Acceptance test-driven development (ATDD).
3. It utilizes the keyword driven testing approach.
4. Provides test libraries implemented either with Python or java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating testcases.
5. Not Much Programming is required.





Setup

1. Install python.
2. Install PyCharm IDE
3. Install Selenium
4. Install robot framework.
5. Install robot framework selenium library.

Install Selenium

Command to install – pip install selenium

Command to Uninstall – pip uninstall selenium

A screenshot of a computer

Description automatically generated with medium confidence

Install robot framework.

Command to install – pip install robotframework

Command to Uninstall – pip uninstall robotframework

Command to upgrade – pip install --upgrade robotframework

Pip list

Pip show robotframework

Pip check robotframework

A picture containing text, screenshot, font, software

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidence

Install robot framework selenium library

Command to install – pip install robotframework-seleniumlibrary

Command to uninstall – pip uninstall robotframework-seleniumlibrary

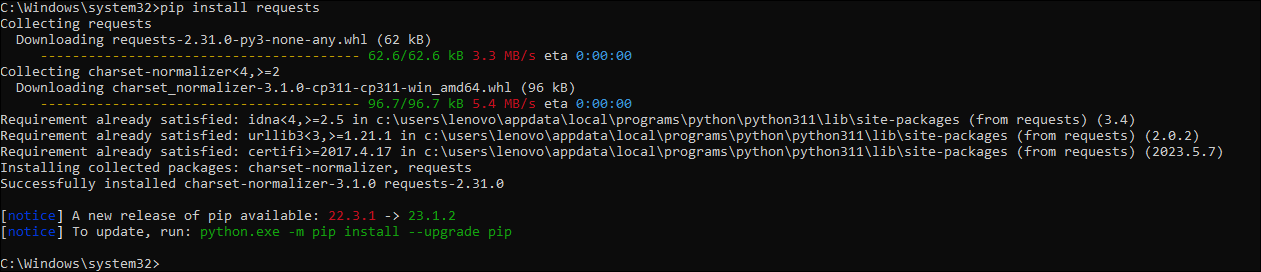
A screen shot of a computer program

Description automatically generated with low confidence

Install request library

Command to install – pip install requests

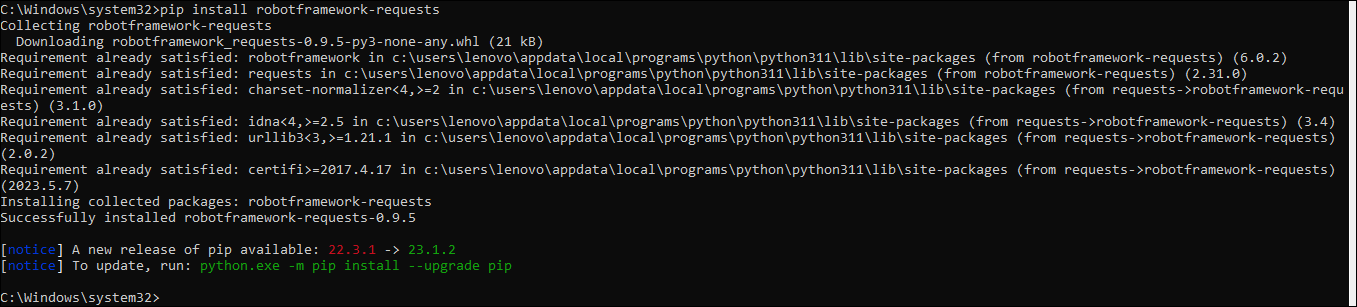
Command to uninstall – pip uninstall requests



Install request library.

Command to install – pip install robotframework-requests

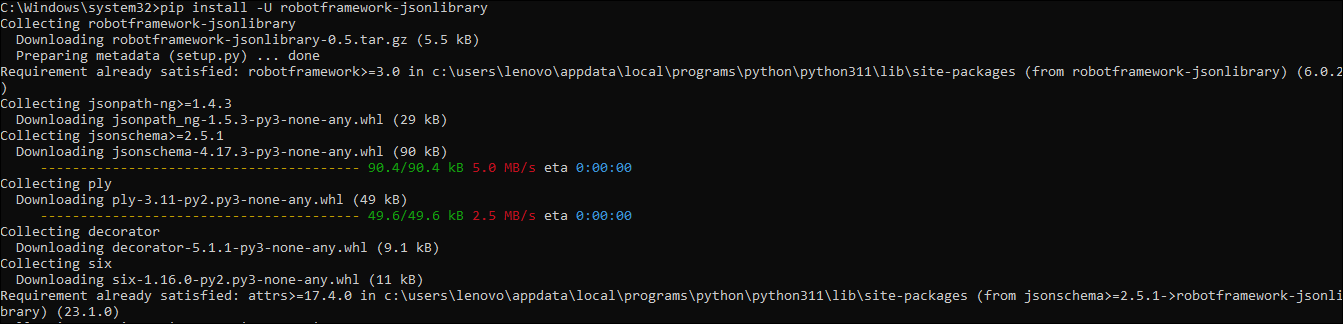
Command to uninstall – pip uninstall robotframework-requests



Install robotframework-jsonlibrary library.

Command to install – pip install -U robotframework-jsonlibrary

Command to uninstall – pip uninstall robotframework-jsonlibrary

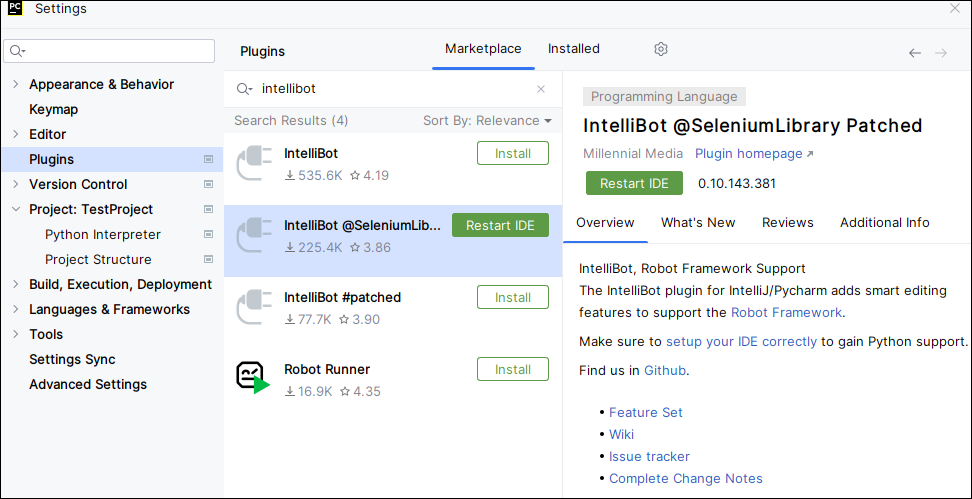


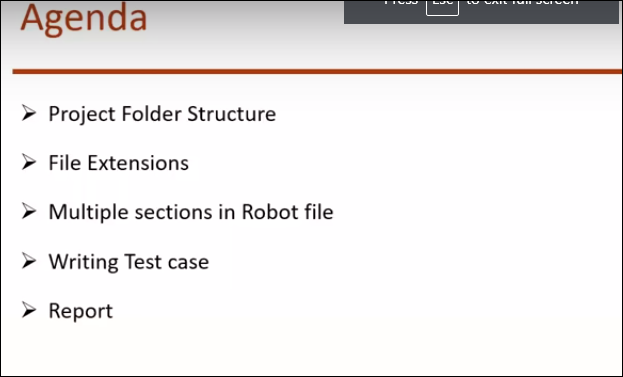
Finding – Now PyCharm don’t know where these library are installed.

Solution – So now we have to set these library in PyCharm.

Open Pycharm >Select Project > Go to file > Go to Settings > Project Test> Python interpreture> Click on + > Search for library > Click on install package.

Install Intellibot@seleniumlibrary Plugin ---so it can identify .robot extension file.





Extension of file

Testcase file ---- .robot

Excel - .xls

Notepad - .txt

Python - .py

Limitations: -

1. Robot can validate the value in string whereas value get extracted in integer.

Appium installation

pip install robotframework-appiumlibrary

A computer screen with text

Description automatically generated

Robotframwork Variables

1. **Scalar Variables**: These variables hold single values like strings or numbers. You can define them using the **Set Variable** keyword or by using the syntax **${variable\_name}**.
2. \*\*\* Variables \*\*\*
3. ${username} JohnDoe
4. ${password} password123

${url} <http://example.com>

1. **List Variables**: These variables hold a list of values. You can define them using the **@{variable\_name}** syntax.

\*\*\* Variables \*\*\*

@{browsers} Chrome Firefox Safari

1. **Dictionary Variables**: These variables hold key-value pairs. You can define them using the **&{variable\_name}** syntax.

\*\*\* Variables \*\*\*

&{user\_info} username=JohnDoe password=password123

1. **Environment Variables**: These are system-level variables that you can access within your Robot Framework tests. They can be set outside of Robot Framework and are accessed using the **%{variable\_name}** syntax.
2. \*\*\* Variables \*\*\*

%{HOME\_DIR} /home/user

Tagging

Robot –include=sanity <pathOfTestSuite>

-I 🡪 To include

-e 🡪 To Exclude

**Sequential Execution**

Robot </recentFolder/> 🡪 To Run all the Testsuite inside the recentFolder.

----Run testsuite using Regular expression ----

Robot </recentFolder/\*.robot> 🡪 To run all the testsuite name ends with “.robot” inside the </recentFolder/>

Robot </recentFolder/Po\*.robot> 🡪 To run all the testsuite name starts with “Po” inside the </recentFolder/>

**Parallel Execution**

* Using pabot we can run number of testsuit parallel

pip install -U robotframework-pabot ---Installation Command

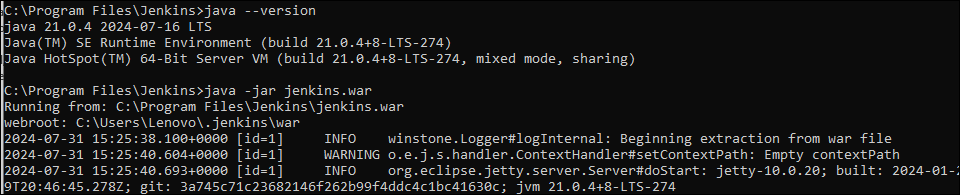
pabot –processes <no of test suite> –output results <recentFolder>\\*.robot – Not working in my PC.

Below one is working in my PC.

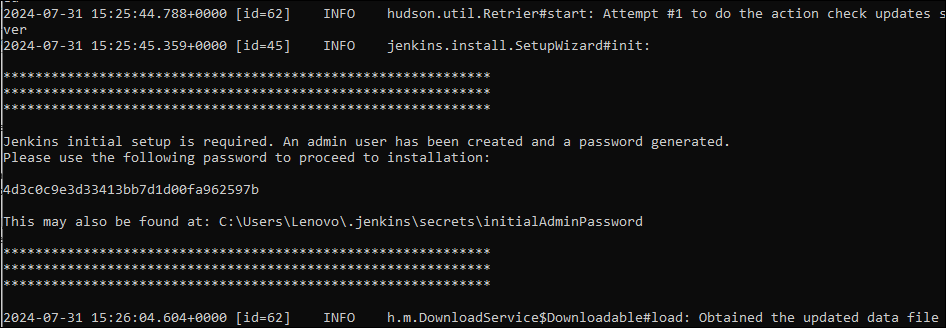
python -m pabot.pabot --processes 2 --outputdir Results TestSuit\TestCases\\*.robot

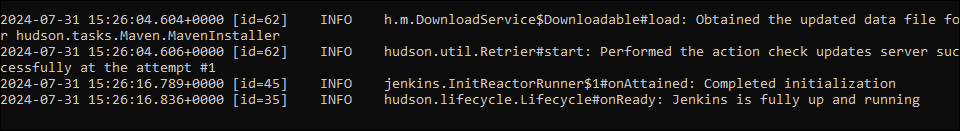
Bat file

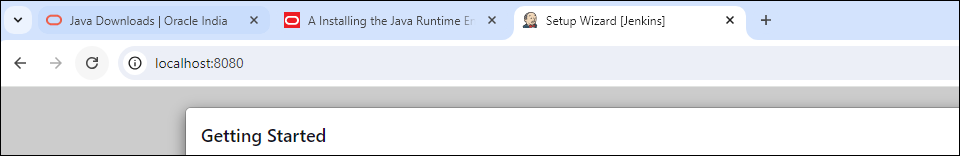
**Robot testcases in Jenkins**

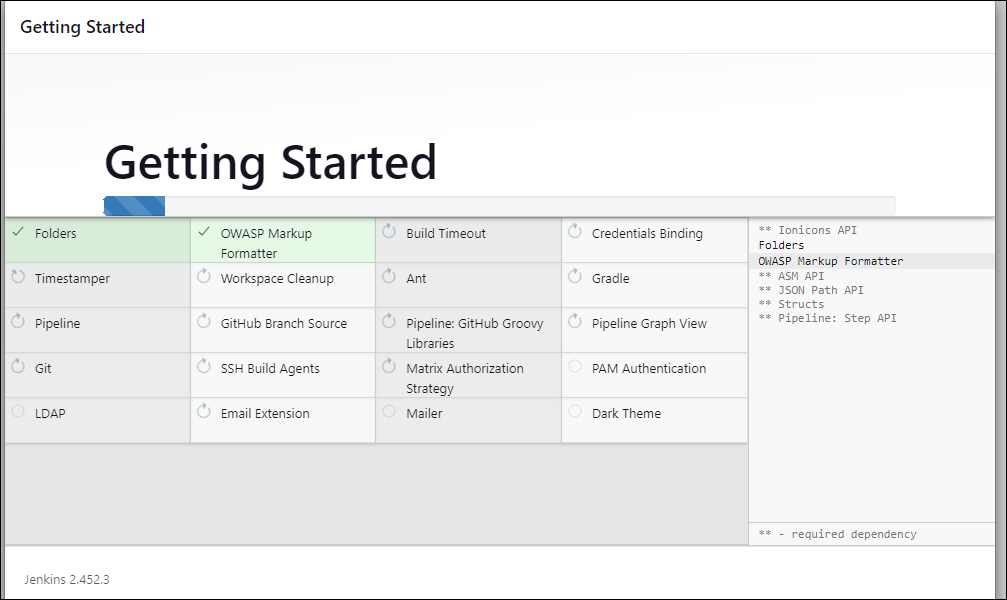


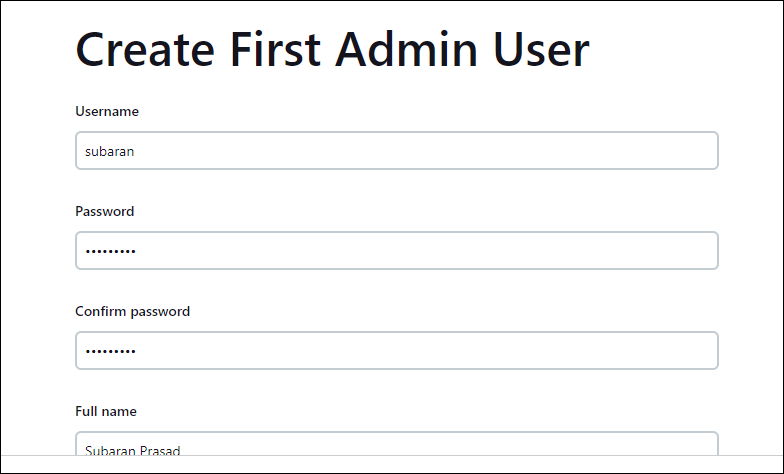
Jenkins Pas- 4d3c0c9e3d33413bb7d1d00fa962597b









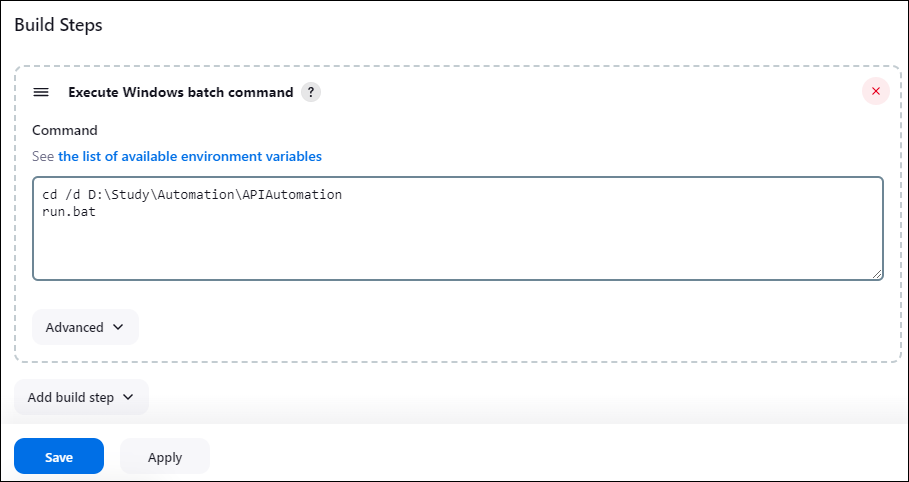


Username-Testlearnlab

Pass-Testlearnlab

Email-Testlearnlab24@gmail.com

Project Name:-RobotAPIAutomatioProject 🡪 FreeStyleProject 🡪 OK



Go To Services -🡪 Restart the Jenkins

