**Why should we use Selenium with python as Automation Framework?**

Compared to other languages, Python take less time to run script and complete execution. It’s very   
easy to learn. Python has very good support of IDE and good industry adoption also. For Web application, Mobile application and API automation Python is good choice because it provides faster development and also achieve good memory management. Selenium python also provide easy environment setup process for applications.

**What is CT Automation Framework?**

The CT Automation Framework provides a starter kit for automation of web applications using WebDriver 2.0 and common testing frameworks like Unit test. The code base consists of classes which allows and assist anyone to start automation of web-based application using Python as a base language.

It can be used to test any applications running on browsers like Chrome, Firefox, Internet Explore etc.

**Getting Started:**

**Prerequisites:**

* Firefox 56, Internet Explorer 11 or Chrome 75 browser installed on the machine where the scripts will be executed. Browser version and drivers need to be updated as per the requirement.
* Knowledge of Python and selenium web driver.
* An IDE like Eclipse or in case anyone is comfortable coding in PyCharm, that is all what is required.
* Unittest framework in used to run scripts.
* The basic Python program like adding two numbers should be executed on IDE.
* Installed selenium on machine by using pip command pip install selenium from command prompt.
* Access to Git repo of CitiusTech.

**Test Data and Setup:**

* Download Eclipse Refer below link:

<https://eclipse.org/downloads/>

* Download the project from GitHub and import into the IDE
* Download the latest version of Python and Install it on machine Refer Below Link: <https://www.python.org/downloads/>
* Make sure System Environment variables are updated class path for python.

E.g. Under the System variables I created new variable as “PythonPath” and in this variable I have added path for python folder which is installed on machine.

**Download PyDev in Eclipse:**

* Open Eclipse IDE and Go to Help menu and click on Install New Software.
* Enter http://pydev.org/updates in the Work with:  field.
* After several seconds, two options should appear. Select the PyDev for Eclipse option.
* Click "Next" and "OK" to continue installing PyDev.
* Select "I accept the terms of the license agreement", then click "Finish". The installer will begin to download the plug-in.
* When the installation is complete, you will be asked if you want to restart Eclipse. Select "Yes".

**Configure PyDev:**

* Python must be installed to configure PyDev.
* Go to Window -> Preferences. In the Preferences window, expand PyDev and select Interpreter-Python.
* Click “New” and type interpreter name. For Interpreter executable, browse to your copy of python.exe and press open.
* Click ok and the selection Needed window will appear.
* Select all and click OK as many times as necessary to exit the preferences. The default selection should be fine.
* The Interpreter is now set up so that the code you write can be interpreted for the computer to run. You are now ready to start running code.

**Executing a Sample Project:**

By default, the starter kit comprises of one test case that will help the users to understand the framework better. The template scenario dose the following:

* Launch the browser and navigate to Amazon.com
* Sign in to the Amazon
* Search for the Product and Logout from the website

To execute the tests, do the following as per the test framework chosen-

* Open the Tests.py file and click on Run menu and select Run As -> Python Unittest

**Writing Custom Tests:**

The following changes will have to be done to use the code as per your needs-

* Config.py – This file has all the variable required to customize the code as per your requirement.
  + HTML\_REPORT\_DIR – path where HTML report can be stored.
  + CHROME\_DRIVER\_PATH – path for chrome driver
  + FIREFOX\_DRIVER\_PATH – path for gecko driver
  + IE\_DRIVER\_PATH – path for IE driver
  + APPLICATION\_URL
* Locators – Create unique file for each page in Locators folder and add page objects by creating the variable in python.
* Pages – This folder have unique file for each page and it contains the functions a page does.
* Tests – Sample Login.py file has all the login page test cases and Tests.py file has unittest framework setup and all testes would be called in this file.
* Reports – This folder contain all the HTML reports
* Utilities – This folder contain all the generic functions.

**What’s in CT automation Framework:**

To run the test scripts, open the **Tests.py** file from Test Folder and click on run and select run option. It will execute the test cases for Amazon website.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.No. | Module Name | File Name 1 | Description |
| 1 | Comparator | Comparator.py | This file contains functions for comparing objects and primitive data type |
| 2 | Config | Config.py  Driver.py  DBDetails.py | This file contains all the constant which are going to use in code base. |
| 3 | DateAndTimeConversion | DateAndTimeConversion.py | This file contains function for different date formats and time conversion. |
| 4 | DBConnection | DBConnection.py | This file contains the sample code for DB Connection |
| MySqlConnection.py | This file contains the MySQL db connection and DB operations code like insert, update, delete, select |
| 5 | Drivers |  | This module contains chrome, Firefox and IE drivers. |
| 6 | BrowserType | BrowserType.py | This file contains function for setup the drivers for different type of browsers |
| 7 | Exceptions | UrlNavigationException.py | All the user defined exceptions classes go under this module |
| 8 | Images |  | All the images stored in this folder |
| 9 | Locators | Locators.py | Classes with locators of the web objects are placed under this file. |
| 10 | Pages |  | This framework follows Page Object Pattern; hence a separate module has been provided for each page in the application. Any such page can be placed under this module for example- LoginPage.py |
| 11 | Reports |  | All the generated HTML report are stored in this folder |
| 12 | Tests | Login.py  Test.py | This module contains tests file which include all the test and unittest framework setup |
| 13 | Utilities | CsvHandler.py  ExcleHandler.py  GenericFunctions.py  Log.py  Screenshot.py  Utilities.py | All the utilities classes which are placed under this package. They provide common functionalities like Logging, Reporting, Read Excel, etc. |
| 14 | Waits | waits.py | This class is used for synchronizing the automation scripts with the application. It has methods like wait\_for\_element\_presence  wait\_for\_element\_visiblity |

**Features yet to covered:**

* HL7 Message Parsing
* DICOM image comparison
* Json file comparison
* Parallel Execution
* Cloud Services
* Mobile Driver
* Continues Integration Jenkins

**Python Libraries Used so far:**

* Pandas - This library is used to work with data, it is used to read or write data from csv or excel file. It has many other purposes also.
* Logging -This module is intended to provide a standard error logging mechanism in Python.
* Time – This module allows us to handle various operations regarding time.
* datetime-This module allows us to handle various operations regrading date and time.
* urllib3- urllib3 is a powerful, friendly HTTP client for Python. This module used to provide feature like Thread safety, connection pooling, client-side SSL/TLS verification, File uploads with multiple encoding.
* Mss- This module is used to capture the screenshot.
* Random- This module is used to generate the random numbers.
* Mysql.connector – This module is used to implements DB operations by using MySQL db.

**Feature Not working Currently:**

* Application not working in IE browser.