**Pride Automation Startup Guide**

The Pride Automation project also known as IAutomate automation project uses POM(Page Object Model) design pattern with Test NG and Hybrid Framework(Test data will is stored in excel sheet template & will be passed in the script for execution).

**Prerequisite:**

1.JDK 1.8 & above

2.Eclipse IDE

3.SONICWALL NetExtender VPN

Raise a ticket in HyFi for installing SONICWALL NetExtender VPN in your system and with the help of Sysnet team install it.

You need to ask Sysnet team to add the below entries in your hosts file located inside C: drive.

192.168.20.8 xorbi.xoriant.com 10.20.1.11 ebizdev.xoriant.com 10.20.1.14 ebizdevords.xoriant.com10.20.1.19 ebizuatords.xoriant.com10.20.0.13 ADFSTest-PUN.ADTEST.COM10.20.1.20 ebizqa.xoriant.com 10.20.1.21 ebizqaords.xoriant.com

(You can verify the hosts file by traversing the folder structure C:\Windows\System32\drivers\etc)

4. Application Link:

<http://ebizdevords.xoriant.com:8060/ords/pride/r/pinnacle/login_desktop>

It is preferred to open the link in **Chrome Browser in incognito mode**.(Ctrl+Shift+n)

User needs to connect to SONICWALL NetExtender VPN to access the application.

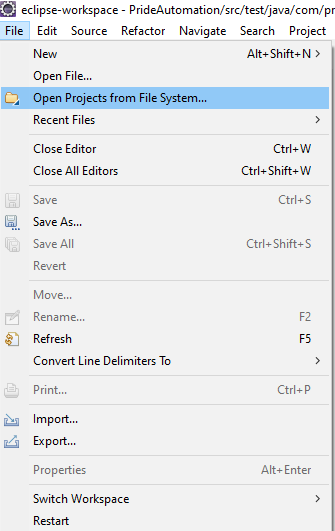
**Project Setup**

**Step1: Copy the PrideAutomation project in your local system**

Once you receive the PrideAutomation project in zipped format, copy it in your local folder & later unzip it.

**Step2: Import the project**

Open Eclipse IDE and click on File-> Open Projects from File System, a window will be displayed on screen for importing your project.

****

Inside the window as shown below, select the Pride Automation folder by clicking on the “Directory” button as highlighted below. Select the project once it is imported and click on “Finish”.

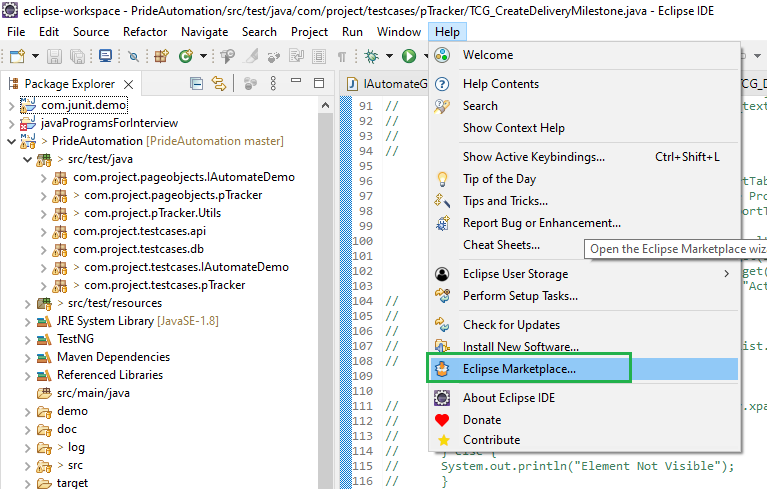
(Note- It might take few seconds/minutes to import the project)

Graphical user interface, text, application

Description automatically generated

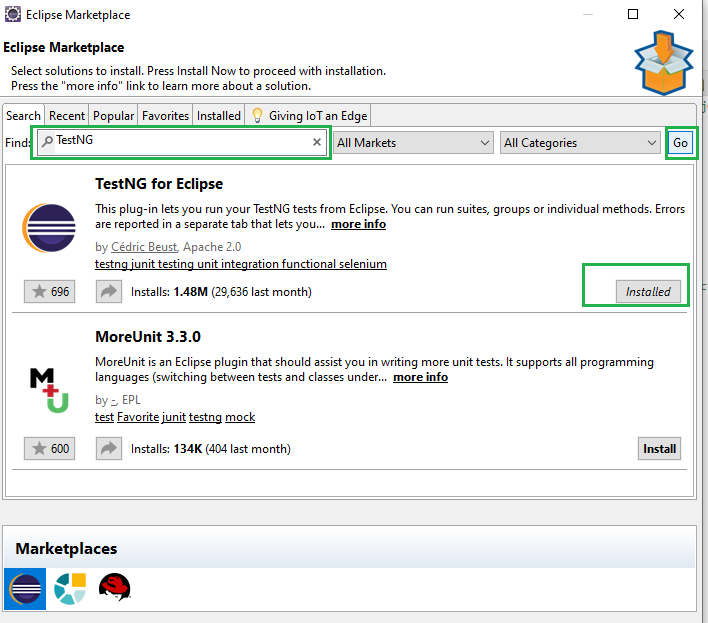
**3. Step3: Install Test NG:**

Inside Eclipse navigate to Help->Eclipse Marketplace



Enter TestNG in the “Find” search box and click on Go button as shown below. “TestNG for Eclipse” will be fetched & shown. Click on the “Install” button to install the software at your end.

(Note- Since TestNG is already installed at my end it is showing as Installed.)



**PrideAutomation Folder Structure**

**There are 3 class files for each page/screen.**

For example for Login screen 2 files have been created under com.project.pageobjects.pTracker package**.**

1. **PTracketLoginConstants.java –** This class consists of all the elements required for the Login feature.

**public** **static** **final** String ***USERNAME\_TXT*** = "//\*[@id='P9999\_USERNAME']";

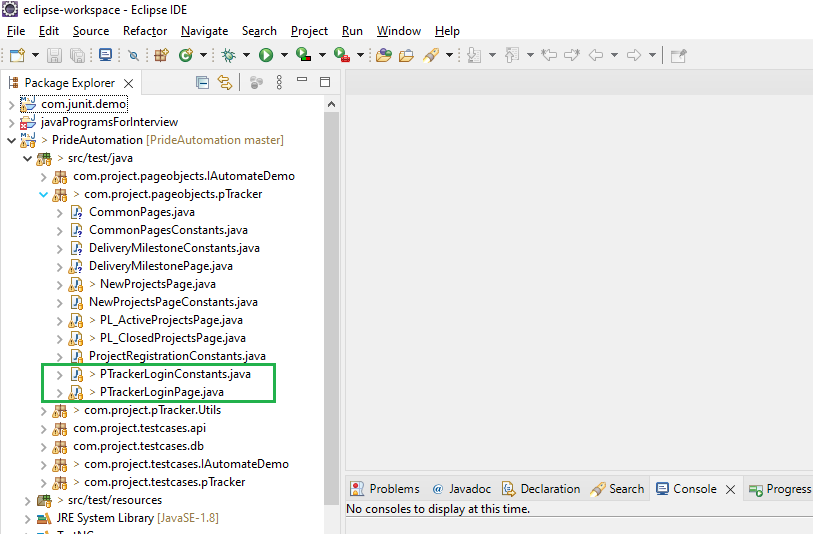
**public** **static** **final** String ***PASSWORD\_TXT*** = "//\*[@id='P9999\_PASSWORD']";

**public** **static** **final signifies** as below

public- It can be accesses anywhere inside & outside the package. No restriction.

Static- We can access it without creating the object.

Final- It can’t be changed

****

**Snippet of PTracketLoginConstants.java**

Graphical user interface, text, application

Description automatically generated

**Snippet of PTrackerLoginPage.java**

This consists of all the methods which has been created for the functionalities present on the page in addition it also has some common methods.

Methods related to the screen/feature:

**public** **boolean** isDashboardIconVisibleOnPage()

**public** **boolean** selectEmployee(String eName)

**public** **boolean** goToPTrack()

Common Methods:

**1.public** **boolean** isElementDisplayed(WebElement webElement) for verifying element displayed or not.

2. **public** **boolean** waitUntilElementPresent

We have used PageFactory to initialize webelements.

PageFactory.*initElements*(driver, **this**);

Graphical user interface, text, application

Description automatically generated

**Snippet of TCG\_PTrackerLogin.java**

All the test cases are located inside com.project.testcases.pTracker package.

We need to execute the test cases to verify our functionality,

**The execution flow works like below:**

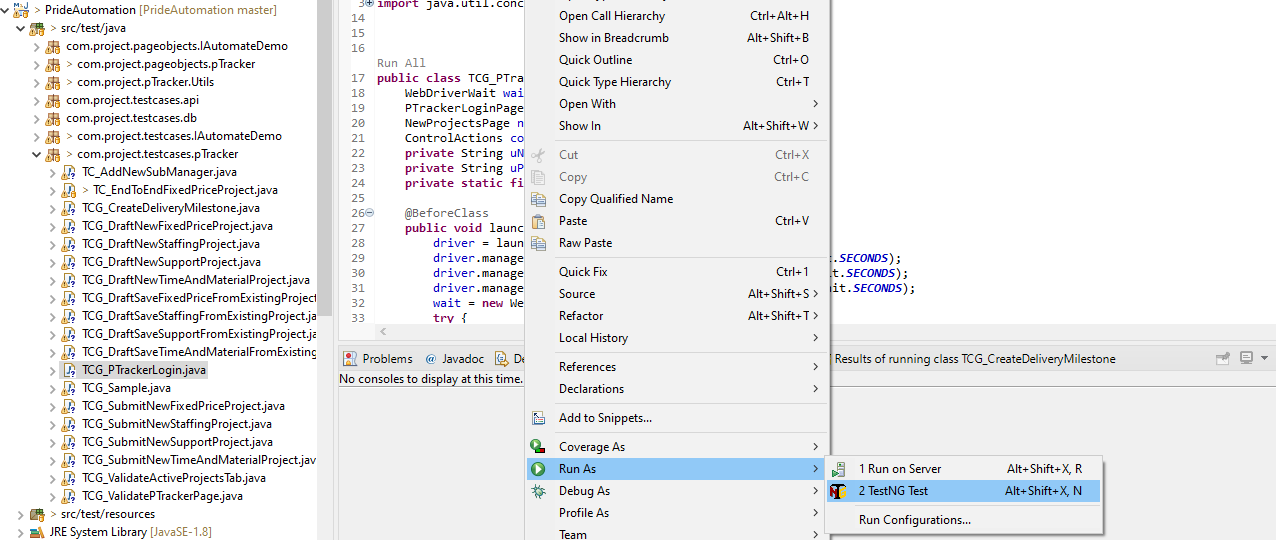
Execute **TCG\_PTrackerLogin.java**->It calls the methods present inside **PTracketLoginPage.java->**It calls the elements stored inside **PTrackerLoginConstants.java**

This is the driver script. We need to execute this script to test the feature.

**Graphical user interface, text, application

Description automatically generated**

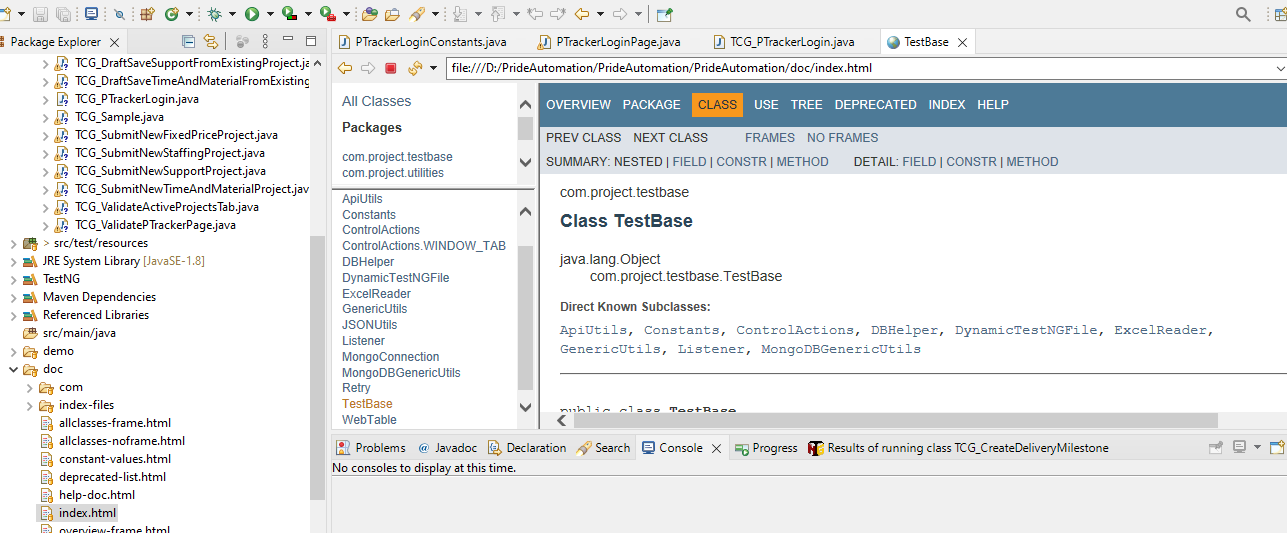
Perform Right Click->Run As->TestNG Test

****

**Q) Where can I get the methods in TestBase, ControlActions & other classes.**

Inside Eclipse IDE navigate to doc->open index.html in a browser

Click on the Packages, Calssses, Methods etc for better understanding.

****

**Q)Show me one example where we are passing the data from excel file to script.**

File name: TCG\_DraftNewFixedPriceProject.java

Line no 56:

String workspace = System.*getProperty*("user.dir");

The above line will show your project directory.

Right click on project->Click on Properties. You can see your project location.

Graphical user interface, text, application

Description automatically generated

Line no 57:

String datapoolPath = workspace+"\\test-data-files\\UI-TestData\\TC\_CreateFixedPriceNewProject.xls";

//tcRowNum = 1;

Inside //Project//test-data-files//UI-TestData// folder the excel sheet TC\_CreateFixedPriceNewProject.xls is located from which data will be passed to your script.

Line No:59

tcRowNum = ExcelUtils.*getRowNum*(datapoolPath,"Automation","testCase","TC\_DRAFT\_NEW\_FIXED\_PRICE\_PROJECT");

Inside TC\_CreateFixedPriceNewProject.xls excel sheet one row ","TC\_DRAFT\_NEW\_FIXED\_PRICE\_PROJECT"is located for your respective test case.

Graphical user interface, text, application

Description automatically generated