SELENIUM MIGRATION TO ASCENTIALTEST

BY Matryxsoft Tech LLP

DATE : 15th Sept 2020

Background and Introduction:

Until recently, there were many adherent's to Selenium, an open source automated testing tool and organizations hired consulting services on build test automation suites. Consulting companies encouraged their customers to adopt the open source tool, shifting the cost of licensing to services, which made for profitable business. In the initial phase, there was a good rate of success, but over time and with an increase in application features, tests began to fail and maintaining test projects became a major problem as frameworks started to crumble Ultimately Selenium test automation projects became too costly. Consequently, clients have started look for ways to migrate their years of efforts to a stable and reliable test automation tool.

Matryxsoft Tech, already well known for building reliable tests, analysed the requirement from their customers along with the need from the IT market arena on test automation migration of Selenium. Being the distributors of **AscentialTest (AT),** we came up with a plan and worked with Zeenyx Software, Inc (Product Owner) to migrate Selenium to **AT**. The purpose of this paper is to provide a detailed understanding on how the migration was successfully completed from Selenium Java to **AT** w.r.t to Object Repository, Steps/Functions, Tests. The approach applies to other Selenium languages.

Setup:

AscentialTest Tool and a testing project built on Selenium are the basic requirement for migration. The Selenium project includes the object repository, functions, tests, plans and input data reading

from external databases. We segregated all the files according to the conversion requirements. While it is possible to execute the conversion with the utilities that are prepared for migration, a better conversion rate can be achieved by developing an understanding of the Selenium architecture and the framework built around for the application under test.

Application Knowledge and the manual flow testcase document will help in fine tuning the conversion at a faster rate.

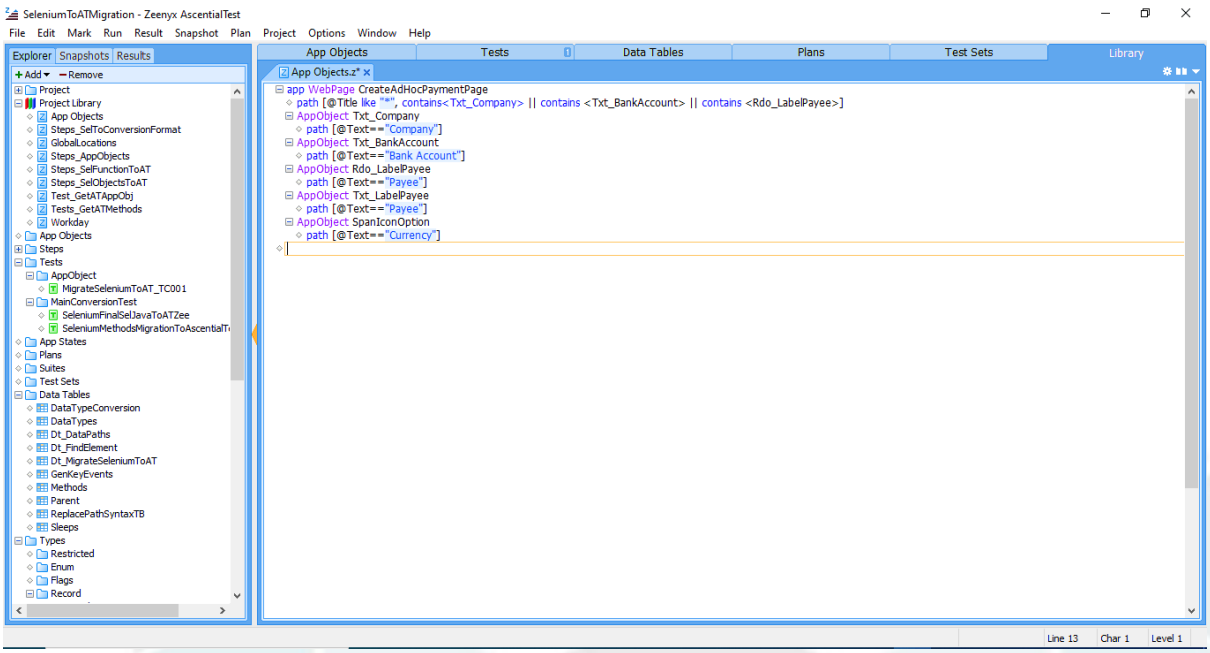
AppObjects to AppObjects:

Selenium Objects are recognized by the path statements like id, name, linktext, xpath, CSS and so on. At first, all the java files related to the app objects are copied and converted to the AscentialTest AppObjects through the conversion utilities that we created. The conversion utilities handle all the necessary indentation and the syntax changes required by AscentialTest. The challenges are seen on the driver.FindElement and driver.FindElements which are handled through our utilities and brings them to the AscentialTest AppObjects format. Please find the example of AppObjects conversion as shown in the AT Screenshot.

Selenium Screenshot

```
// Company
@FindBy(xpath = "(//label[text()='Company']/parent::div/following-sibling::div//input)")
WebElement Txt_Company;

// Bank Account
@FindBy(xpath = "(//label[text()='Bank Account']/parent::div/following-sibling::div//input)")
WebElement Txt_BankAccount;

// label payee ----------------------
@FindBy(xpath = "//label[text()='Payee']/../parent::div//input[@type=\"radio\"]")
WebElement Rdo_LabelPayee;

// Text payee
@FindBy(xpath = "(//label[text()='Payee']/../parent::div//input)[2]")
WebElement Txt_LabelPayee;

// Currency
@FindBy(xpath = "(//label[text()=\"Currency\"]/../following-sibling::*//span[@data-automation-id=\"promptIcon\"])[1]")
WebElement SpanIconOption;
```

AscentialTest Migrated Screenshot



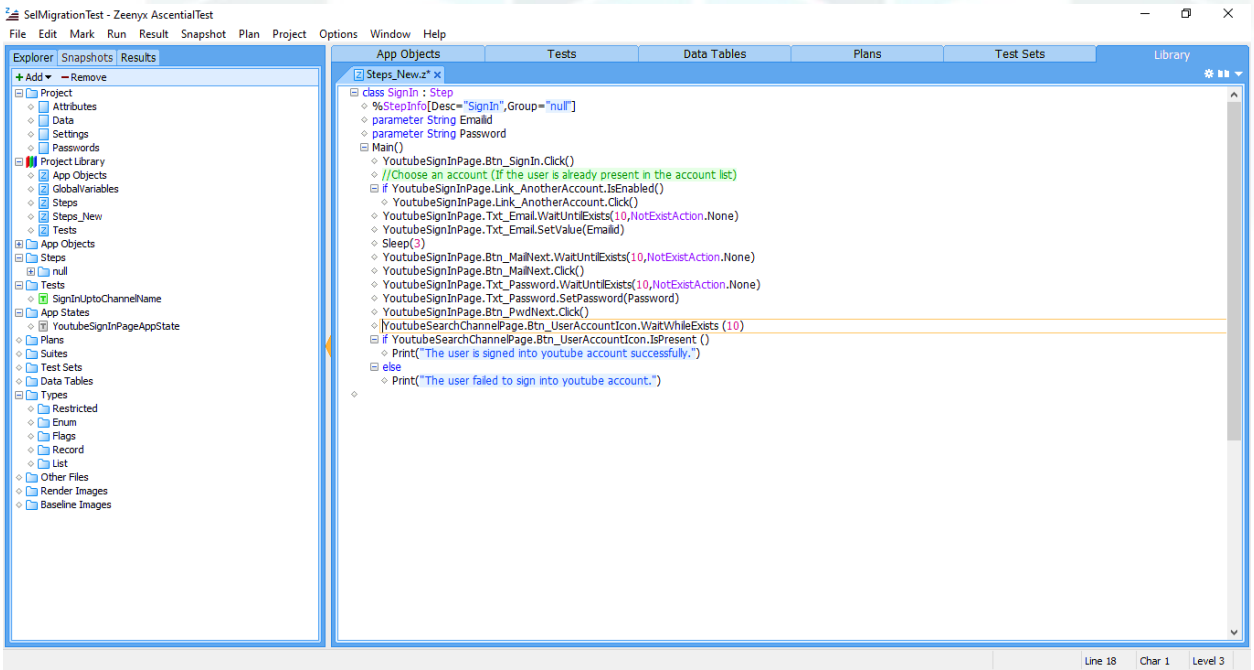
Functions to Steps/Functions:

Selenium functions are written on multiple java files that are created within classes referencing using interfaces and extend classes. There could be ‘n’ number of functions with ‘n’ number of parameters which may or may not have return data type. For AscentialTest the references (Interfaces and Extend Classes) are of no importance and also does not depend on the object creation to access the functions present in different files. All the functions which return more than one datatype are kept as a function and the ones with no return type are converted to steps in AscentialTest. Numerous challenges were faced while migrating the functions on the datatypes, parameters, function calls, subfunction calls, indentation, return datatypes and reading input data. All these challenges were successfully handled using the steps utilities created by Matryxsoft. There are multiple utilities that were written to handle the conversion techniques. With the help of these utilities, we converted Selenium functions to AT Steps/Functions. A Sample of a function written in Selenium that is migrated to AT is shown in the below pictures

Selenium Function

```
      public void SignIn(String Emailid, String Password) throws
InterruptedException
      {
            Btn_SignIn.click();
            //Choose an account (If the user is already present in
the account list)
if(driver.findElements(By.xpath("//div[contains(text(),'Use another
account')]")).size() > 0==true)
          {
            Link_AnotherAccount.click();
          }
            WebDriverWait wait = new WebDriverWait(driver,10);
          wait.until(ExpectedConditions.visibilityOf(Txt_Email));
            Txt_Email.sendKeys(Emailid);
            Thread.sleep(3000);
            Btn_MailNext.click();
     wait.until(ExpectedConditions.visibilityOf(Txt_Password));
            Txt_Password.sendKeys(Password);
            Btn_PwdNext.click();
            if(Btn_UserAccountIcon.isDisplayed())
            {
                  System.out.println("The user is signed into
youtube account successfully.");

            }
            else
            {
                  System.out.println("The user failed to sign into
youtube account.");
            }
      `
```

AscentialTest StepConversion



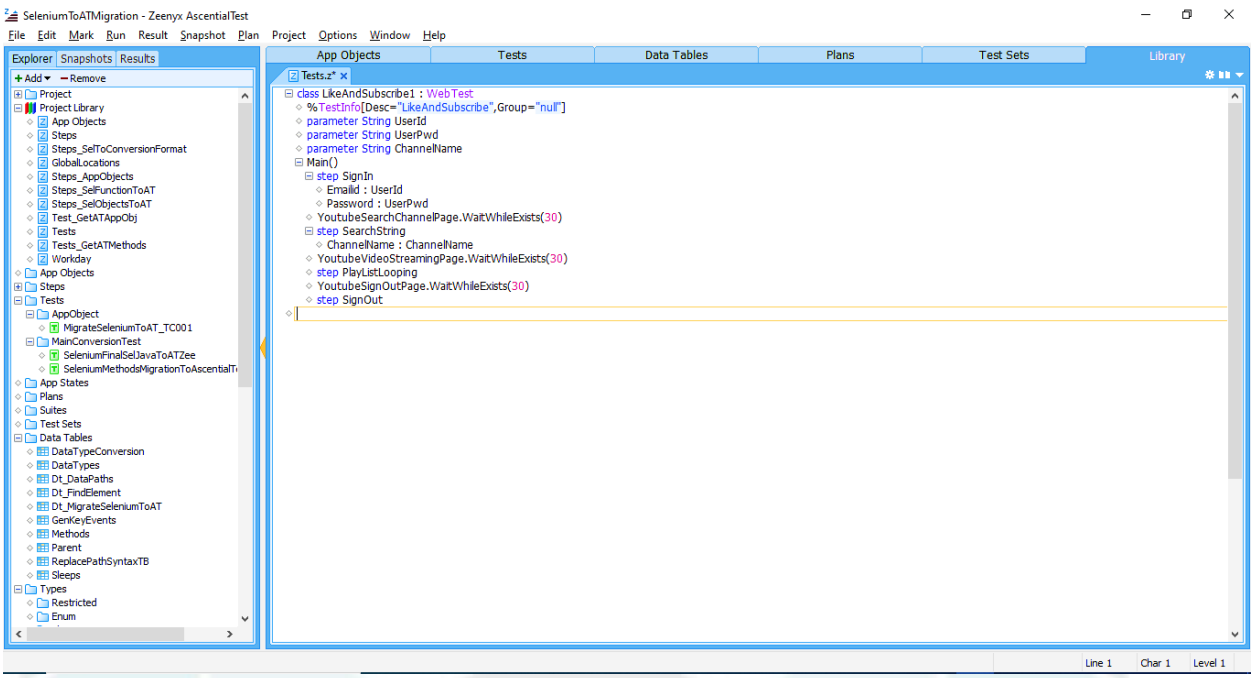
Tests to Tests

Selenium testcases are written in a separate test java files which calls 'n' number of functions and every test is associated with @BeforeTest which calls the base state of the application as a prerequisite and @AfterTest closes the application at the end of the testcase. In AscentialTest, we have a concept of AppState that has OnStart and OnFinish, which works similar way of Selenium @BeforeTest and @AfterTest. The challenges faced in the test were related to function call, indentation and parametrization. All these were handled by the test utilities that were written by Matryxsoft in order to convert the tests from Selenium to AscentialTest. The successful transition of the tests is as shown in the below pictures.

Selenium Test

```
      public void LikeAndSubscribe(String UserId,String UserPwd,
String ChannelName) throws InterruptedException
            {
            //Navigate to Youtube User Account
            YoutubeSignInPage SignInElements=
PageFactory.initElements(driver, YoutubeSignInPage.class);
            SignInElements.SignIn(UserId,UserPwd);
            driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);

            //Search and Select the Channel
            YoutubeSearchChannelPage ToSearchChannel=
PageFactory.initElements(driver, YoutubeSearchChannelPage.class);
            ToSearchChannel.SearchString(ChannelName);
            driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);

            //Like and Subscribe the channel
            YoutubeVideoStreamingPage PlayList =
PageFactory.initElements(driver, YoutubeVideoStreamingPage.class);
            PlayList.PlayListLooping();
            driver.manage().timeouts().implicitlyWait(30,
TimeUnit.SECONDS);

            //SignOut from the User Account
          YoutubeSignOutPage SignOutElements=
PageFactory.initElements(driver, YoutubeSignOutPage.class);
            SignOutElements.SignOut();
            driver.manage().timeouts().implicitlyWait(25,
TimeUnit.SECONDS);
```

AT Test Conversion



Conclusion:

Organizations have seen years of investment in Selenium tests becoming unstable, unreliable and difficult to maintain. Matryxsoft Tech conversion utilities makes it easy to convert those tests to AscentialTest resulting in improved reliability and maintainability of the tests and in turn delivers best software quality.

For more information on migrating Selenium to **AT** test automation tool

Visit our Website : https://www.matryxsoft.com/selenium-to-ascentialtest/