**Automation framework User Guide**

HCL Automation framework is an integrated tool for web and API automation. We can develop automation scripts rapidly with minimal knowledge of Java and Selenium. This framework supports IE, Chrome, firefox, safari, headless browsers.

Our Automation framework is divided into two projects

1. ATnTWebAutomation
2. UniversalFramework-master

ATnTWebAutomation uses UniversalFramework-master project as reference in term of Jar file. Team need to write the test cases in ATnTWebAutomation project and all below mentioned steps must be done in ATnTWebAutomation project. Before start, you should have following software in your system.

1. Eclipse
2. Maven tool , dependencies
3. TestNG dependencies
4. **Test Environment setup and other configurable parameter**

We need to define test url, browser details etc. in env.properties file location under \src\test\resources.

This file is having following entries.

1. url : mention the url on which we want to run the test cases
2. browser: mention the browser name on which we want to run our test cases. Possible values are “chrome”, “firefox”, “ie”, “safari”, “qnx”,”headless”
3. Islocale= mention true/false. If true ie test case will run on locale machine otherwise test cases will run on remote machine using grid url

Note: For other parameters values are remain same for now.

1. **How to write a test cases for new page:**
2. Identify all the required controls which are required for test case automation for new page and add those into *Element.java* available in your project folder (*com.atnt.<your project name>* )

Note:: New page entries must be separated out with proper comments including page name.

1. Add new class file with proper page name which can be easily correlated with web page. This class file will have page factory initialization for its controls and all its major operation
2. Create test case class file in src\test\<respective test folder>. Each test case class file will have following things.
3. Test case class should extend WebTestCase class
4. Each test cases must have following declarations

*private Common common = new Common(driver);*

*private WebDriver driver2 = driver;*

1. There must be before class and after class methods with following entries.

@BeforeClass

**public** **void** initialize(ITestContext testContext) {

testContext.setAttribute("driver", driver2);

getDriver().manage().deleteAllCookies();

common.impicitWait(20);

getDriver().manage().window().maximize();

getDriver().get(getSession().getEnv().getUrl());

getDriver().manage().timeouts().pageLoadTimeout(200, TimeUnit.*SECONDS*);

}

@AfterClass

**public** **void** closeBrowser() {

**try** {

driver.quit();

} **catch** (Exception e) {}

}

1. **Enable screenshot in test report**
2. Each test case must have below mentioned entry with test case java class name

*@Listeners({ Screenshot.class})*

1. **Make test cases data driven**
2. Add one entry in your test case java file for excel file path which is having your data. For Ex:

***private*** *ExcelManager xls =* ***new*** *ExcelManager(System.getProperty("user.dir")*

*+ "//src//test//resources//testdata.xlsx");*

1. Add one data provider function which will read data from one particular sheet in mentioned excel file. For Ex:

*@DataProvider(name = "inputdata")*

***public*** *Object[][] readExcelData() {*

*Object[][] sheetData =* ***null****;*

*sheetData = xls.getSheetData("address", "validAddress");*

***return*** *sheetData;*

*}*

In above example, address is sheet name of excel and valid is particular row name. in this case, it will fetch the data from the row which is having keyword validAddress. In case, If we want to run the test case for all the data set available in Excel sheet then remove Second parameter from getSheetData Function in above example.

1. In Test cases, Now Add one TestNG attribute “Data Provider” with the same name given in data provider function under @ test annotation. In this example, data provider name would be “*inputdata”*
2. Add parameters in test case. Parameter count would be equal to number of columns available in excel sheet. all must be declared as String type.
3. **Enable Extent Reporting**
4. To enable the extent reports, please add the following entry under Suite tag of testNG file.

<suite name="ATnTDaily" > <listeners>

        <listener class-name=*"com.atnt.framework.utility.ExtentReporter"* />

</listeners>

1. If Someone want to run one particular test cases without TestNG suite file then please add ExtentReporter class file name explicitly in listener annotation under test case class file. For Ex:

@Listeners({ Screenshot.class ,com.atnt.framework.utility.ExtentReporter.class})