

```

package tests;
import java.util.regex.Pattern; import java.io.FileInputStream;          import java.util.Properties;      import java.util.concurrent.TimeUnit;
import org.junit.*;           import static org.junit.Assert.*;        import static org.hamcrest.CoreMatchers.*;  import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;      import org.openqa.selenium.support.ui.Select; import functions.HotelApp_BusinessFunctions;

public class DynamicUIObjectTest extends HotelApp_BusinessFunctions {
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        prop = new Properties();
        driver = new FirefoxDriver();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        //load the property files
        HA_BF_LoadPropertyFiles();
    }

    @Test
    public void testMyFirstWebDriver() throws Exception {
        String hotelUserId = "debugger";
        String hotelUserPwd = "SW516java!";

        //load AUT in browser
        driver.get(sAppURL);

        String spageTitle = driver.getTitle();          //validate page title - Arlen 5/9/16
        if (spageTitle.equalsIgnoreCase("AdactIn.com - Hotel Reservation System"))
            System.out.println("Page Title is correct. Actual title is: " + spageTitle);
        else
            System.out.println("Page Title is incorrect. Actual title is: " + spageTitle);

        ///## LOGIN PAGE
        HA_BF_Login(driver, hotelUserId, hotelUserPwd);

        String userWelcomeText = driver.findElement(By.xpath(prop.getProperty("Txt_SearchHotel_WelcomeText"))).getAttribute("value");
        assertEquals("User's Name Display", "Hello " + hotelUserId + "!", userWelcomeText);    //validate user login name appears with Assert statement

        driver.findElement(By.id(prop.getProperty("Btn_SearchHotel_Submit"))).click();          //validate error msg for missing location - Arlen 5/9/16
        String sLocationFieldError = driver.findElement(By.xpath(prop.getProperty("Txt_SearchHotel_LocationError"))).getText();
    }
}

```

```

if(sLocationFieldError.equalsIgnoreCase("Please Select a Location"))
    System.out.println("Location Error Message is correct. Actual message is: " + sLocationFieldError);
else
    System.out.println("Location Error Message is incorrect. Actual message is: " + sLocationFieldError);

//### HOTEL SEARCH PAGE
String strFile = "./DataPool/HA_HotelSearch.xls";          //read data from excel
String strLocation = HA_GF_readXL(1,"Location",strFile);
HA_BF_SearchHotel(driver, strLocation, "2 - Two", "2 - Two");

String slocation = driver.findElement(By.xpath(prop.getProperty("Lst_SelectHotel_LocationValue"))).getAttribute("value");
if (slocation.equalsIgnoreCase("Sydney"))                //validate hotel location in search results - Arlen 5/9/16
    System.out.println("The search location results are correct");
else
    System.out.println("The search location results are incorrect");

//### SELECT HOTEL PAGE
HA_BF_SelectHotel(driver);

//### BOOK HOTEL PAGE
HA_BF_BookHotel(driver, "John", "Doe", "55 Somestreet Lane\nCOS, CO 80919", "3434343434343434", "VISA", "January", "2020", "123");

//### GET ORDER NUMBER
Thread.sleep(10);
String strOrderNo = driver.findElement(By.id(prop.getProperty("Txt_BookHotel_OrderNo"))).getAttribute("value");
System.out.println("Order Number generated is " + strOrderNo);

//CANCEL THE ORDER
driver.findElement(By.id(prop.getProperty("Btn_BookItinerary_MyItinerary"))).click();
driver.findElement(By.id(prop.getProperty("Txt_BookItinerary_SearchOrderId"))).sendKeys(strOrderNo);
driver.findElement(By.id(prop.getProperty("Btn_BookItinerary_Go"))).click();
driver.findElement(By.xpath(prop.getProperty("Btn_BookItinerary_CancelOrderNo"))).click();
Alert javascriptAlert = driver.switchTo().alert();
System.out.println(javascriptAlert.getText());
javascriptAlert.accept();

//### LOGOUT PAGE
HA_BF_Logout(driver);

//### RETURN TO LOGIN PAGE
HA_BF_ReturnToLogin(driver);
}

```

```
@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
```

SHAREDUIMAP.PROPERTIES

```
#### LOGIN PAGE ####
Txt_Login_Username = username
Txt_Login_Password = password
Btn_Login_Login = login

#### SEARCH HOTEL PAGE ####
Txt_SearchHotel_WelcomeText = .//*[[@id='username_show']
Txt_SearchHotel_LocationError = .//*[[@id='location_span']
Txt_SearchHotel_Location = location
Txt_SearchHotel_RoomCount = room_nos
Txt_SearchHotel_AdultPerRoom = adult_room
Btn_SearchHotel_Submit = Submit

#### SELECT HOTEL PAGE ####
Lst_SelectHotel_LocationValue = .//*[[@id='location_1']
Rad_SelectHotel_Radio_Button_2 = radiobutton_2
Btn_SelectHotel_Continue = continue

#### BOOK HOTEL PAGE ####
Txt_BookHotel_FirstName = first_name
Txt_BookHotel_LastName = last_name
Txt_BookHotel_Address = address
Txt_BookHotel_CCNumber = .//*[[@id='cc_num']
Lst_BookHotel_CCType = cc_type
Lst_BookHotel_CCExpMonth = cc_exp_month
Lst_BookHotel_CCExpYear = cc_exp_year
Txt_BookHotel_CCCvvNum = .//*[[@id='cc_cvv']
Btn_BookHotel_BookNow = book_now
Txt_BookHotel_OrderNo = order_no

#### BOOKED ITINERARY ####
Lnk_BookItinerary_BookedItinerary = Booked Itinerary
Btn_BookItinerary_MyItinerary = my_itinerary
Txt_BookItinerary_SearchOrderId = order_id_text
Btn_BookItinerary_Go = search_hotel_id
Txt_BookItinerary_OrderResult = .//*[starts-with(@id,'order_id')] [@class="select_text"]
Btn_BookItinerary_CancelOrderNo = .//*[contains(@id,'btn_id')]

#### LOGOUT ELEMENTS ####
Lnk_BookHotel_Logout = Logout
Btn_BookHotel_Logout = logout

#### RETURN TO LOGIN PAGE ####
Lnk_LoginReturn_ReturnToLogin = Click here to login again
```

HOTELAPP_BUSINESSFUNCTIONS . JAVA

package functions;

```
import jxl.*;    import jxl.read.biff.BiffException;    import java.io.File;    import java.io.FileInputStream;    import java.util.List;
import java.util.Locale;    import java.util.NoSuchElementException;    import java.util.Properties;    import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;    import org.openqa.selenium.WebElement;    import org.openqa.selenium.support.ui.Select;
```

```
public class HotelApp_BusinessFunctions {
    public static WebDriver driver;
    public static Properties prop;
    public static String sAppURL;
    public static String sSharedUIMapPath;
    public static String sSharedHotelSearchXL;
    public static String sSharedHotelBookingXL;

    public void HA_BF_LoadPropertyFiles () throws Exception{
        prop.load(new FileInputStream("./Configuration/HA_Configuration.properties"));
        sAppURL = prop.getProperty("AppUrl");
        sSharedUIMapPath = prop.getProperty("SharedUIMap");
        prop.load(new FileInputStream(sSharedUIMapPath));
        sSharedHotelSearchXL = prop.getProperty("SharedHotelSearchXL");
        sSharedHotelBookingXL = prop.getProperty("SharedHotelBookingXL");
    }

    public void HA_BF_Login (WebDriver driver, String sUserName, String sPassword)throws InterruptedException{
        driver.findElement(By.id(prop.getProperty("Txt_Login_Username"))).clear();
        driver.findElement(By.id(prop.getProperty("Txt_Login_Username"))).sendKeys(sUserName);
        driver.findElement(By.id(prop.getProperty("Txt_Login_Password"))).clear();
        driver.findElement(By.id(prop.getProperty("Txt_Login_Password"))).sendKeys(sPassword);
        driver.findElement(By.id(prop.getProperty("Btn_Login_Login"))).click();
    }

    public void HA_BF_BookHotel (WebDriver driver, String sFirstName, String sLastName, String sAddress, String sCCNum, String sCCType, String sExpMth){
        driver.findElement(By.id(prop.getProperty("Txt_BookHotel_FirstName"))).clear();
        driver.findElement(By.id(prop.getProperty("Txt_BookHotel_FirstName"))).sendKeys(sFirstName);
        new Select(driver.findElement(By.id(prop.getProperty("Lst_BookHotel_CCType")))).selectByVisibleText(sCCType);
        new Select(driver.findElement(By.id(prop.getProperty("Lst_BookHotel_CCExpMonth")))).selectByVisibleText(sExpMth);
        new Select(driver.findElement(By.id(prop.getProperty("Lst_BookHotel_CCExpYear")))).selectByVisibleText(sExpYr);
        driver.findElement(By.id(prop.getProperty("Btn_BookHotel_BookNow"))).click();
    }
}
```

```

public void HA_BF_LogoutBtn (WebDriver driver) {
    driver.findElement(By.id(prop.getProperty("Btn_BookHotel_Logout"))).click();
}

public void HA_BF_ReturnToLogin (WebDriver driver) {
    driver.findElement(By.linkText(prop.getProperty("Lnk_LoginReturn_ReturnToLogin"))).click();
}

public static String HA_GF_readXL (int row, String column, String strFilePath) {
    Cell c= null;
    int reqCol=0;
    WorkbookSettings ws = null;
    Workbook workbook = null;
    Sheet sheet = null;
    FileInputStream fs = null;
try{
    fs = new FileInputStream(new File(strFilePath));
    ws = new WorkbookSettings();
    ws.setLocale(new Locale("en", "EN"));
    // opening the work book and sheet for reading data
    workbook = Workbook.getWorkbook(fs, ws);
    sheet = workbook.getSheet(0);
    // Sanitize given data
    String col = column.trim();
    //loop for going through the given row
    for(int j=0; j<sheet.getColumns(); j++)
    {
        Cell cell = sheet.getCell(j,0);
        if((cell.getContents().trim()).equalsIgnoreCase(col))
        {
            reqCol= cell.getColumn();
            //System.out.println("column No:"+reqCol);
            c = sheet.getCell(reqCol, row);
            fs.close();
            return c.getContents();
        }
    }
}
catch(BiffException be)
{

```

```

        System.out.println("The given file should have .xls extension.");
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
    System.out.println("NO MATCH FOUND IN GIVEN FILE: PROBLEM IS COMING FROM DATA FILE");
    return null;
}

public static int HA_GF_XLRowCount (String strFilePath, String sColumn) {
    int k;
    for (k=1; k<999; k++)
    {
        String sParamVal = HA_GF_readXL(k,sColumn, strFilePath);
        if (sParamVal.equals("ENDOFROW"))
        {
            break;
        }
    }
    return k;
}

//dynamically wait for element presence
public void HA_GF_WaitForElementPresent (WebDriver driver, By by, int iTimeOut) throws InterruptedException {
    int iTotal = 0;
    int iSleepTime = 5000;
    while(iTotal < iTimeOut)
    {
        List<WebElement> oWebElements = driver.findElements(by);
        if(oWebElements.size(>0)
            return;
        else
        {
            Thread.sleep(iSleepTime);
            iTotal = iTotal + iSleepTime;
            System.out.println(String.format("Waited for %d milliseconds.[%s]", iTotal, by));
        }
    }
}

```

//dynamically wait for element to achieve its property value

public void HA_GF_WaitForPropertyValue (WebDriver driver, By by, String ExpPropertyVal, String PropertName, int iTimeOut) throws InterruptedException {

int iTotal = 0;

int iSleepTime = 5000;

while(iTotal < iTimeOut)

{

List<WebElement> oWebElements = driver.findElements(by);

if(oWebElements.size()>0)

for(WebElement weOption : oWebElements)

{

if(weOption.getAttribute(PropertName).equalsIgnoreCase(ExpPropertyVal))

{

return;

}

else

{

Thread.sleep(iSleepTime);

iTotal = iTotal + iSleepTime;

System.out.println(String.format("Waited for %d milliseconds.[%s]", iTotal, by));

}

}

else

{

Thread.sleep(iSleepTime);

iTotal = iTotal + iSleepTime;

System.out.println(String.format("Waited for %d milliseconds.[%s]", iTotal, by));

}

}

}

public void HA_BF_SearchBookedItinerary (WebDriver driver, String OrderNo){

driver.findElement(By.linkText(prop.getProperty("Lnk_BookItinerary_BookedItinerary"))).click();

driver.findElement(By.id(prop.getProperty("Txt_BookItinerary_SearchOrderId"))).clear();

driver.findElement(By.id(prop.getProperty("Txt_BookItinerary_SearchOrderId"))).sendKeys(OrderNo);

driver.findElement(By.id(prop.getProperty("Btn_BookItinerary_Go"))).click();

}

}

HA_CONFIGURATION.PROPERTIES

APPLICATION URL

AppUrl = <http://www.adactin.com/HotelAppBuild2/>

SHARED UI MAP FILE PATH

SharedUIMap = [./SharedUIMap/SharedUIMap.properties](#)

SHARED EXCEL HOTEL SEARCH DATA FILE

SharedHotelSearchXL = [./DataPool/HA_HotelSearch.xls](#)

SHARED EXCEL HOTEL BOOKING DATA FILE

SharedHotelBookingXL = [./DataPool/HA_Booking.xls](#)