

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* dataProvider.ConfigDataProvider\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package dataProvider;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.util.Properties;

public class ConfigDataProvider {

Properties pro;

public ConfigDataProvider(){

File src=new File("./Configuration/config.properties");

try {

FileInputStream fis=new FileInputStream(src);

pro=new Properties();

pro.load(fis);

}

catch (Exception e) {

System.out.println("Exception is " + e.getMessage());

}

}

public String getApplicationurl(){

String url=pro.getProperty("url");

return url;

}

public String getChromePath()

{

String ChromePath=pro.getProperty("chromePath");

return ChromePath;

}

public String getIEPath()

{

String IEPath=pro.getProperty("IEPath");

return IEPath;

}

public String getusername()

{

String Username=pro.getProperty("OrangeHRMUserName");

return Username;

}

public String getpassword()

{

String password=pro.getProperty("OrangeHRMPassword");

return password;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*factory.BrowserFactory\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package factory;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.openqa.selenium.ie.InternetExplorerDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import dataProvider.ConfigDataProvider;

public class BrowserFactory {

public static WebDriver driver;

public BrowserFactory(WebDriver driver){

this.driver=driver;

}

public static WebDriver getBrowser(String browserName)

{

if(browserName.equalsIgnoreCase("firefox"))

{

driver=new FirefoxDriver();

}

else if(browserName.equalsIgnoreCase("Chrome"))

{

//ConfigDataProvider config=new ConfigDataProvider();

//DesiredCapabilities capabilities = DesiredCapabilities.chrome();

//ChromeOptions options = new ChromeOptions();

//options.addArguments("test-type");

//capabilities.setCapability("chrome.binary", "<Path to binary>");

//capabilities.setCapability(ChromeOptions.CAPABILITY, options);

//System.setProperty("webdriver.chrome.driver", DataProviderFactory.getConfig().getChromePath());

//driver = new ChromeDriver(capabilities);

System.setProperty("webdriver.chrome.driver", DataProviderFactory.getConfig().getChromePath());

driver=new ChromeDriver();

}

else if(browserName.equalsIgnoreCase("IE"))

{

//ConfigDataProvider config=new ConfigDataProvider();

DesiredCapabilities cap = DesiredCapabilities.internetExplorer();

cap.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS, true);

System.setProperty("webdriver.ie.driver", DataProviderFactory.getConfig().getIEPath());

//System.setProperty("webdriver.ie.driver", "D:\\Selenium Drivers\\IEDriverServer.exe");

//WebDriver driver = new InternetExplorerDriver(cap);

driver = new InternetExplorerDriver(cap);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Browser Started\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//driver=new InternetExplorerDriver();

}

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

return driver;

}

public static void closeBrowser(WebDriver driver)

{

driver.close();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Factory.DataProviderFactory\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package factory;

import dataProvider.ConfigDataProvider;

import dataProvider.ExcelDataConfig;

public class DataProviderFactory {

public static ConfigDataProvider getConfig()

{

ConfigDataProvider config=new ConfigDataProvider();

return config;

}

public static ExcelDataConfig getExcel()

{

ExcelDataConfig excel=new ExcelDataConfig("F:\\Selenium Framework\\com.orangeHRM.hybridFramework\\TestData\\OrangeHRM.xlsx");

return excel;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*pages. OrangeHRMHomePage\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package pages;

import java.util.List;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.How;

import factory.BrowserFactory;

import utility.BaseExtentTest;

public class OrangeHRMHomePage{

final WebDriver driver;

public OrangeHRMHomePage(WebDriver driver)

{

this.driver=driver;

}

@FindBy(how = How.XPATH, using = "//a[@href='/index.php/admin/viewAdminModule']")

public static WebElement AdminLink;

@FindBy(how = How.LINK\_TEXT, using = "User Management")

public static WebElement UserManagement;

@FindBy(how = How.XPATH, using = "//a[@href='/index.php/admin/viewSystemUsers']")

public static WebElement users;

@FindBy(how = How.ID, using = "btnAdd")

public static WebElement addbutton;

@FindBy(how = How.ID, using = "systemUser\_userType")

public static WebElement UserRole;

@FindBy(how = How.ID, using = "systemUser\_employeeName\_empName")

public static WebElement EmployeeName;

@FindBy(how = How.ID, using = "systemUser\_userName")

public static WebElement username;

@FindBy(how = How.ID, using = "systemUser\_status")

public static WebElement status;

@FindBy(how = How.ID, using = "systemUser\_password")

public static WebElement password;

@FindBy(how = How.ID, using = "systemUser\_confirmPassword")

public static WebElement confirmpassword;

@FindBy(how = How.ID, using = "btnSave")

public static WebElement savebutton;

@FindBy(how = How.XPATH, using = "//a[text()='Welcome Admin']")

public static WebElement WelcomeAdmin;

@FindBy(how = How.XPATH, using = "//a[text()='Logout']")

public static WebElement LogoutButton;

@FindBy(how = How.XPATH, using = "//table[@id='resultTable']//tbody//tr")

public static List<WebElement> ResultsTable;

@FindBy(how = How.TAG\_NAME, using = "td")

public static List<WebElement> ResultsTDTable;

@FindBy(how = How.ID, using = "btnDelete")

public static WebElement deletebutton;

@FindBy(how = How.XPATH, using = "//input[@value='Ok']")

public static WebElement UserdelOk;

public static void Logout() throws Exception

{

WelcomeAdmin.click();

Thread.sleep(5000);

LogoutButton.click();

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*testcases.AddEmployee\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package testCases;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.support.PageFactory;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.Select;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeTest;

import org.testng.annotations.Test;

import dataProvider.ExcelDataConfig;

import factory.BrowserFactory;

import factory.DataProviderFactory;

import pages.AddEmployeePage;

import pages.OrangeHRMLogin;

import pages.PersonalDetailsPage;

import utility.BaseExtentTest;

import utility.GetScreenshot;

import utility.Highlight;

public class AddEmployee extends BaseExtentTest{

//static WebDriver driver;

static AddEmployeePage AddEmpPage;

static OrangeHRMLogin PageLogin;

static PersonalDetailsPage PersonalDTPage;

//AddEmployeePage addEmpPage;

static ExcelDataConfig excel = new ExcelDataConfig("F:\\Selenium Framework\\com.hybridFramework\\ApplicationTestData\\TestData.xlsx");

/\*

@SuppressWarnings("static-access")

@BeforeTest

public static void AppOrangeHRMLogin(){

logger = extent.createTest("AppOrangeHRMLogin");

//this will automatically opens the browser, maximize the browser

driver = BrowserFactory.getBrowser("Chrome");

//this will provide the URL from the config.properties file

driver.get(DataProviderFactory.getConfig().getApplicationurl());

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

PageLogin=PageFactory.initElements(driver, OrangeHRMLogin.class);

Highlight.highLightElement(driver, OrangeHRMLogin.txtbx\_UserName);

Highlight.highLightElement(driver, OrangeHRMLogin.txtbx\_Password);

PageLogin.OrangeLogin(DataProviderFactory.getConfig().getusername(), DataProviderFactory.getConfig().getpassword());

}

\*/

@SuppressWarnings("static-access")

@Test

public static void addEmployee() throws Exception{

AddEmpPage=PageFactory.initElements(driver, AddEmployeePage.class);

//AddEmployeePage.PIMLink.click();

AddEmpPage.PIMLink.click();

Thread.sleep(5000);

Highlight.highLightElement(driver, AddEmployeePage.AddEmployeeButton);

AddEmpPage.AddEmployeeButton.click();

Thread.sleep(5000);

Highlight.highLightElement(driver, AddEmployeePage.FirstName);

AddEmpPage.FirstName.sendKeys("Narayana14");

Highlight.highLightElement(driver, AddEmployeePage.LastName);

AddEmpPage.LastName.sendKeys("Barnana14");

Highlight.highLightElement(driver, AddEmployeePage.EmployeeId);

AddEmpPage.EmployeeId.sendKeys("133454");

Highlight.highLightElement(driver, AddEmployeePage.Photograph);

AddEmpPage.Photograph.click();

Runtime.getRuntime().exec("C:\\Users\\Narayana\\Desktop\\ORAddempFileUpload.exe");

Highlight.highLightElement(driver, AddEmployeePage.CreateLoginCheckBox);

AddEmpPage.CreateLoginCheckBox.click();

WebDriverWait wait=new WebDriverWait(driver,10);

wait.until(ExpectedConditions.visibilityOfElementLocated((By.id("user\_name"))));

Highlight.highLightElement(driver, AddEmployeePage.empUserName);

AddEmpPage.empUserName.sendKeys("Narayana14");

Highlight.highLightElement(driver, AddEmployeePage.empPassword);

AddEmpPage.empPassword.sendKeys("Test@009");

Highlight.highLightElement(driver, AddEmployeePage.empConfirmPassword);

AddEmpPage.empConfirmPassword.sendKeys("Test@009");

Highlight.highLightElement(driver, AddEmployeePage.empStatus);

Select statusdd=new Select(AddEmployeePage.empStatus);

statusdd.selectByVisibleText("Enabled");

Highlight.highLightElement(driver, AddEmployeePage.SaveEmployeeButton);

AddEmpPage.SaveEmployeeButton.click();

GetScreenshot.capture(driver, "AddEmplooyeePage");

//Edit the Personal details

PersonalDTPage=PageFactory.initElements(driver, PersonalDetailsPage.class);

Thread.sleep(5000);

//Highlight.highLightElement(driver, PersonalDTPage.PersonalDetailsLink);

//PersonalDTPage.PersonalDetailsLink.click();

Highlight.highLightElement(driver, PersonalDTPage.EditButton);

PersonalDTPage.EditButton.click();

WebDriverWait WaitNat=new WebDriverWait(driver,10);

WaitNat.until(ExpectedConditions.visibilityOfElementLocated((By.id("personal\_cmbNation"))));

Date myDate = new Date();

System.out.println(myDate);

//System.out.println(new SimpleDateFormat("MM/dd/yyyy").format(myDate));

String DateF=new SimpleDateFormat("yyyy-MM-dd").format(myDate);

System.out.println("The date format is " + DateF);

System.out.println(new SimpleDateFormat("yyyy-MM-dd").format(myDate));

System.out.println(myDate);

Highlight.highLightElement(driver, PersonalDTPage.LicenseExpiryDate);

//PersonalDTPage.LicenseExpiryDate.sendKeys(DateF);

PersonalDTPage.LicenseExpiryDate.clear();

PersonalDTPage.LicenseExpiryDate.sendKeys(DateF);

Highlight.highLightElement(driver, PersonalDTPage.GenderCheckBox);

PersonalDTPage.GenderCheckBox.click();

Highlight.highLightElement(driver, PersonalDTPage.MaritalStatus);

Select MaritalStatusdd=new Select(PersonalDTPage.MaritalStatus);

MaritalStatusdd.selectByVisibleText("Single");

Highlight.highLightElement(driver, PersonalDTPage.Nationality);

Select Nationalitydd=new Select(PersonalDTPage.Nationality);

Nationalitydd.selectByVisibleText("Indian");

String dob="1991-12-04";

Highlight.highLightElement(driver, PersonalDTPage.personalDOB);

PersonalDTPage.personalDOB.clear();

PersonalDTPage.personalDOB.sendKeys(dob);

Highlight.highLightElement(driver, PersonalDTPage.SaveButton);

PersonalDTPage.SaveButton.click();

GetScreenshot.capture(driver, "PersonalDetails");

Thread.sleep(5000);

//homepage.Logout();

//BrowserFactory.closeBrowser(driver);

}

/\*

@SuppressWarnings("static-access")

@Test(dependsOnMethods="addEmployee")

public static void EditPersonalDetails() throws Exception

{

PersonalDTPage=PageFactory.initElements(driver, PersonalDetailsPage.class);

Thread.sleep(5000);

//Highlight.highLightElement(driver, PersonalDTPage.PersonalDetailsLink);

//PersonalDTPage.PersonalDetailsLink.click();

Highlight.highLightElement(driver, PersonalDTPage.EditButton);

PersonalDTPage.EditButton.click();

WebDriverWait WaitNat=new WebDriverWait(driver,10);

WaitNat.until(ExpectedConditions.visibilityOfElementLocated((By.id("personal\_cmbNation"))));

Date myDate = new Date();

System.out.println(myDate);

//System.out.println(new SimpleDateFormat("MM/dd/yyyy").format(myDate));

String DateF=new SimpleDateFormat("yyyy-MM-dd").format(myDate);

System.out.println("The date format is " + DateF);

System.out.println(new SimpleDateFormat("yyyy-MM-dd").format(myDate));

System.out.println(myDate);

Highlight.highLightElement(driver, PersonalDTPage.LicenseExpiryDate);

//PersonalDTPage.LicenseExpiryDate.sendKeys(DateF);

PersonalDTPage.LicenseExpiryDate.clear();

PersonalDTPage.LicenseExpiryDate.sendKeys(DateF);

Highlight.highLightElement(driver, PersonalDTPage.GenderCheckBox);

PersonalDTPage.GenderCheckBox.click();

Highlight.highLightElement(driver, PersonalDTPage.MaritalStatus);

Select MaritalStatusdd=new Select(PersonalDTPage.MaritalStatus);

MaritalStatusdd.selectByVisibleText("Single");

Highlight.highLightElement(driver, PersonalDTPage.Nationality);

Select Nationalitydd=new Select(PersonalDTPage.Nationality);

Nationalitydd.selectByVisibleText("Indian");

String dob="1991-12-04";

Highlight.highLightElement(driver, PersonalDTPage.personalDOB);

PersonalDTPage.personalDOB.clear();

PersonalDTPage.personalDOB.sendKeys(dob);

Highlight.highLightElement(driver, PersonalDTPage.SaveButton);

PersonalDTPage.SaveButton.click();

GetScreenshot.capture(driver, "PersonalDetails");

homepage.Logout();

BrowserFactory.closeBrowser(driver);

}

\*/

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*utility.getScreenshot\*\*\*\*\*\*\*\*\*\*\*\*

package utility;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

public class GetScreenshot {

public static String capture(WebDriver driver, String screenshotName) throws Exception

{

TakesScreenshot ts=(TakesScreenshot)driver;

//File source=ts.getScreenshotAs(OutputType.FILE);

File source=ts.getScreenshotAs(OutputType.FILE);

String dest = "./Screenshots/"+screenshotName+".png";

File destination = new File(dest);

FileUtils.copyFile(source, destination);

return dest;

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*utility.baseclass\*\*\*\*\*\*\*\*\*\*\*

package utility;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.support.PageFactory;

import org.testng.ITestResult;

import org.testng.annotations.AfterClass;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.AfterSuite;

import org.testng.annotations.AfterTest;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;

import com.aventstack.extentreports.ExtentReports;

import com.aventstack.extentreports.ExtentTest;

//import com.aventstack.extentreports.Status;

import com.aventstack.extentreports.markuputils.ExtentColor;

import com.aventstack.extentreports.markuputils.MarkupHelper;

import com.aventstack.extentreports.reporter.ExtentHtmlReporter;

import com.aventstack.extentreports.reporter.configuration.ChartLocation;

import com.aventstack.extentreports.reporter.configuration.Theme;

import factory.BrowserFactory;

import factory.DataProviderFactory;

import pages.OrangeHRMHomePage;

import pages.OrangeHRMLogin;

public class BaseExtentTest {

public static WebDriver driver;

public static ExtentReports extent;

public static ExtentTest logger;

public static ExtentHtmlReporter htmlReporter;

public static OrangeHRMHomePage homepage;

@SuppressWarnings("static-access")

@BeforeClass

public static void AppOrangeHRMLogin(){

logger = extent.createTest("AppOrangeHRMLogin");

//this will automatically opens the browser, maximize the browser

driver = BrowserFactory.getBrowser("Chrome");

//this will provide the URL from the config.properties file

driver.get(DataProviderFactory.getConfig().getApplicationurl());

driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

OrangeHRMLogin PageLogin=PageFactory.initElements(driver, OrangeHRMLogin.class);

PageLogin.OrangeLogin(DataProviderFactory.getConfig().getusername(), DataProviderFactory.getConfig().getpassword());

//new BaseExtentTest();

}

@BeforeSuite

public void setUp()

{

htmlReporter = new ExtentHtmlReporter("./Reports/LoginPage.html");

extent = new ExtentReports();

extent.config();

extent.attachReporter(htmlReporter);

htmlReporter.config().setReportName("Report");

htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);

htmlReporter.config().setTheme(Theme.STANDARD);

}

@AfterMethod

public void getResult(ITestResult result) throws Exception{

if(result.getStatus()==ITestResult.FAILURE)

{

System.out.println(result.getName());

System.out.println(driver);

String screenshotPath=GetScreenshot.capture(driver, result.getName());

//logger.log(Status.FAIL, MarkupHelper.createLabel(result.getName() + "test case failed due to below issues", ExtentColor.RED));

logger.fail(MarkupHelper.createLabel(result.getName() + "test case failed due to below issues", ExtentColor.RED));

logger.fail(result.getThrowable());

//String screenshotPath=GetScreenshot.capture(driver, result.getName());

logger.fail("Screenshot below" + logger.addScreenCaptureFromPath(screenshotPath));

}

else if(result.getStatus()==ITestResult.SUCCESS)

{

logger.pass(MarkupHelper.createLabel(result.getName() + "test case Passed", ExtentColor.GREEN));

}

else

{

//logger.skip(Status.SKIP, MarkupHelper.createLabel(result.getName() + "test case Passed", ExtentColor.YELLOW));

logger.skip(MarkupHelper.createLabel(result.getName() + "test case Skipped", ExtentColor.YELLOW));

logger.skip(result.getThrowable());

}

//extent.flush();

}

@AfterClass

public static void CloseApp() throws InterruptedException{

homepage=PageFactory.initElements(driver, OrangeHRMHomePage.class);

homepage.WelcomeAdmin.click();

//WelcomeAdmin.click();

Thread.sleep(5000);

homepage.LogoutButton.click();

//LogoutButton.click();

//driver.close();

driver.quit();

}

@AfterSuite

public void teardown()

{

extent.flush();

//driver.quit();

}

}

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.orangeHRM.hybridFramework</groupId>

<artifactId>com.orangeHRM.hybridFramework</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>3.7.1</version>

</dependency>

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>6.8</version>

<scope>test</scope>

</dependency>

<!-- https://mvnrepository.com/artifact/com.aventstack/extentreports -->

<dependency>

<groupId>com.aventstack</groupId>

<artifactId>extentreports</artifactId>

<version>3.1.2</version>

</dependency>

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi</artifactId>

<version>3.17</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->

<dependency>

<groupId>org.apache.poi</groupId>

<artifactId>poi-ooxml</artifactId>

<version>3.17</version>

</dependency>

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.6</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.7.0</version>

<configuration>

<compilerVersion>3.7.0</compilerVersion>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.20.1</version>

<configuration>

<suiteXmlFiles>

<suiteXmlFile>testng.xml</suiteXmlFile>

</suiteXmlFiles>

</configuration>

</plugin>

</plugins>

</build>

</project>