import com.microsoft.playwright.Locator;

import com.microsoft.playwright.Page;

public class LoginPageLocators {

public Locator usernameInput;

public Locator passwordInput;

public Locator loginButton;

public LoginPageLocators(Page page) {

usernameInput = page.locator("//input[@name='username']");

passwordInput = page.locator("//input[@name='password']");

loginButton = page.locator("//button[@type='submit']");

}

}

import pages.locators.LoginPageLocators;

public class LoginPageActions {

LoginPageLocators locators;

public LoginPageActions(LoginPageLocators locators) {

this.locators = locators;

}

public void login(String username, String password) {

locators.usernameInput.fill(username);

locators.passwordInput.fill(password);

locators.loginButton.click();

}

}

—-----------------------------------------------------------------------------------------------------------------

import com.microsoft.playwright.Locator;

import com.microsoft.playwright.Page;

public class DashboardPageLocators {

public Locator pimMenu;

public Locator userDropdown;

public Locator logoutLink;

public DashboardPageLocators(Page page) {

pimMenu = page.locator("a[href\*='pim/viewPimModule']");

userDropdown = page.locator("span[class='oxd-userdropdown-tab']");

logoutLink = page.locator("//a[text()='Logout']");

}

}

package pages.actions;

import pages.locators.DashboardPageLocators;

public class DashboardPageActions {

DashboardPageLocators locators;

public DashboardPageActions(DashboardPageLocators locators) {

this.locators = locators;

}

public void navigateToPIM() {

locators.pimMenu.hover();

locators.pimMenu.click();

}

public void logout() {

locators.userDropdown.click();

locators.logoutLink.click();

}

}

—----------------------------------------------------------------------------------------------------------------------------

package pages.locators;

import com.microsoft.playwright.Locator;

import com.microsoft.playwright.Page;

public class PIMPageLocators {

public Locator addEmployeeButton;

public Locator employeeListLink;

public PIMPageLocators(Page page) {

addEmployeeButton = page.locator("//a[text()='Add Employee']");

employeeListLink = page.locator("//a[text()='Employee List']");

}

}

import pages.locators.PIMPageLocators;

public class PIMPageActions {

PIMPageLocators locators;

public PIMPageActions(PIMPageLocators locators) {

this.locators = locators;

}

public void clickAddEmployee() {

locators.addEmployeeButton.click();

}

public void goToEmployeeList() {

locators.employeeListLink.click();

}

}

—-------------------------------------------------------------------------------------------------------------

import com.microsoft.playwright.Locator;

import com.microsoft.playwright.Page;

public class AddEmployeePageLocators {

public Locator firstNameField;

public Locator lastNameField;

public Locator saveButton;

public AddEmployeePageLocators(Page page) {

firstNameField = page.locator("//input[@name='firstName']");

lastNameField = page.locator("//input[@name='lastName']");

saveButton = page.locator("//button[@type='submit']");

}

}

package pages.actions;

import com.microsoft.playwright.Page;

import pages.locators.AddEmployeePageLocators;

public class AddEmployeePageActions {

AddEmployeePageLocators locators;

Page page;

public AddEmployeePageActions(AddEmployeePageLocators locators, Page page) {

this.locators = locators;

this.page = page;

}

public void addEmployee(String firstName, String lastName) {

locators.firstNameField.fill(firstName);

locators.lastNameField.fill(lastName);

locators.saveButton.click();

}

public void verifyEmployeeInList(String fullName) {

page.click("a[href\*='viewEmployeeList']");

page.waitForTimeout(2000);

boolean found = false;

for (int i = 0; i < 5; i++) {

if (page.isVisible("text=" + fullName)) {

System.out.println(fullName + " - Name Verified");

found = true;

break;

}

page.keyboard().press("PageDown");

}

if (!found) {

System.out.println(fullName + " - NOT FOUND");

}

}

}

—-----------------------------------------------------------------------------------------------------------------

public class PageActionsFactory {

private Page page;

private LoginPageActions loginPageActions;

private DashboardPageActions dashboardPageActions;

private PIMPageActions pimPageActions;

private AddEmployeePageActions addEmployeePageActions;

public PageActionsFactory(Page page) {

this.page = page;

loginPageActions = new LoginPageActions(page);

dashboardPageActions = new DashboardPageActions(page);

pimPageActions = new PIMPageActions(page);

addEmployeePageActions = new AddEmployeePageActions(page);

}

public LoginPageActions getLoginPageActions() {

return loginPageActions;

}

public DashboardPageActions getDashboardPageActions() {

return dashboardPageActions;

}

public PIMPageActions getPIMPageActions() {

return pimPageActions;

}

public AddEmployeePageActions getAddEmployeePageActions() {

return addEmployeePageActions;

}

}

—-------------------------------------------------------------------------------------------------------------------------

package tests;

import com.microsoft.playwright.\*;

import org.testng.annotations.\*;

import dataproviders.EmployeeDataProvider;

import factory.PageActionsFactory;

import pages.actions.\*;

public class OrangeHRMTest {

Playwright playwright;

Browser browser;

BrowserContext context;

Page page;

// Factory to get all page actions

PageActionsFactory actionsFactory;

// Page actions references

LoginPageActions loginActions;

DashboardPageActions dashboardActions;

PIMPageActions pimActions;

AddEmployeePageActions addEmpActions;

@BeforeClass

public void setup() {

playwright = Playwright.create();

browser = playwright.chromium().launch(new BrowserType.LaunchOptions().setHeadless(false));

context = browser.newContext();

page = context.newPage();

// Initialize factory with page

actionsFactory = new PageActionsFactory(page);

// Get page actions from factory

loginActions = actionsFactory.getLoginPageActions();

dashboardActions = actionsFactory.getDashboardPageActions();

pimActions = actionsFactory.getPIMPageActions();

addEmpActions = actionsFactory.getAddEmployeePageActions();

// Login and navigate to PIM once

page.navigate("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

loginActions.login("Admin", "admin123");

dashboardActions.navigateToPIM();

}

@Test(dataProvider = "employeeData", dataProviderClass = EmployeeDataProvider.class, priority = 1)

public void addEmployeeTest(String firstName, String lastName) {

pimActions.clickAddEmployee();

addEmpActions.addEmployee(firstName, lastName);

}

@Test(dataProvider = "employeeData", dataProviderClass = EmployeeDataProvider.class, priority = 2)

public void verifyEmployeeTest(String firstName, String lastName) {

String fullName = firstName + " " + lastName;

addEmpActions.verifyEmployeeInList(fullName);

}

@AfterClass

public void tearDown() {

dashboardActions.logout();

context.close();

browser.close();

playwright.close();

}

}

import org.testng.annotations.DataProvider;

public class EmployeeDataProvider {

@DataProvider(name = "employeeData")

public Object[][] getEmployeeData() {

return new Object[][] {

{"John", "Doe"},

{"Jane", "Smith"},

{"Alice", "Cooper"},

{"Bob", "Brown"}

};

}

}