Feature file

Feature: Test the IIQ Roles functionality

Scenario Outline: Verifying the Role information

Given User launch the ANZ IIQ application

And enter Userid and Password and click on Submit button

When User clicks on Setup tab and choose Roles option

And enter <RoleName> in the search field

Then verify the <Jobcode>,<Org\_Hrchy\_Lvl6>,<Org>,<Bus\_Unit> and <RBI> details for the <RoleName> in RoleInformation section

#And User clicks on Logout button

Examples:-

|RoleName |Jobcode|Org\_Hrchy\_Lvl6 |Org |Bus\_Unit|RBI |

|ABN - Branch Manager (RBI Tier 2)|"ttttt"|"Aust Branch Network"|"AU"|"ANZAU" |"true"|

# |ABN - Branch Manager (LOCAM) |hhhhhh |Aust Branch Network |kk |ANZAU |true|

Step definition file

package stepdefinition;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.HashMap;

import java.util.Iterator;

import java.util.List;

import java.util.Properties;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import com.google.common.io.Files;

import cucumber.api.java.en.And;

import cucumber.api.java.en.Given;

import cucumber.api.java.en.Then;

import cucumber.api.java.en.When;

import junit.framework.Assert;

//@SuppressWarnings("deprecation")

@SuppressWarnings("deprecation")

public class iiqsd {

WebDriver driver;

Properties prop = new Properties();

@Given("^User launch the ANZ IIQ application$")

public void userLaunchtheANZIIQApplication() throws Throwable {

FileInputStream fileInput = new FileInputStream("C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\iiqprop.properties");

prop.load(fileInput);

//System.setProperty("webdriver.chrome.driver","C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\chromedriver\\chromedriver.exe");

System.setProperty("webdriver.chrome.driver",prop.getProperty("ChromeDriver"));

driver = new ChromeDriver();

driver.manage().deleteAllCookies();

driver.get(prop.getProperty("URL"));

driver.manage().timeouts().implicitlyWait(10000, TimeUnit.SECONDS);

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

Assert.assertEquals("Welcome to IdentityIQ", driver.findElement(By.xpath("//header/span")).getText());

}

@And("^enter Userid and Password and click on Submit button$")

public void theUserEntersUseridAndPassword() throws Throwable {

driver.findElement(By.xpath("//input[@placeholder='Username']")).sendKeys(prop.getProperty("Userid"));

driver.findElement(By.xpath("//input[@placeholder='Password']")).sendKeys(prop.getProperty("Password"));

driver.findElement(By.xpath("//input[@value='Login']")).click();

}

@When("^User clicks on Setup tab and choose Roles option$")

public void userClicksOnSetupTabAndChooseRoles() throws Throwable {

boolean setupflag = selectitem("Setup");

if(setupflag == true)

{

boolean rolesflag = selectitem("Roles");

if(rolesflag != true)

{

driver.quit();

}

}

Assert.assertEquals("Role Management", driver.findElement(By.xpath("//h1[contains(text(),'Role Management')]")).getText());

}

@And("^enter (.\*) in the search field$")

public void enterTheRoleName(String rolename) throws Throwable {

driver.findElement(By.xpath("//input[@placeholder='Enter a Role Name']")).sendKeys(rolename);

WebElement stritem = driver.findElement(By.xpath("//div[@class='sectionHeader']"));

if(stritem.getText().contains(rolename))

{

stritem.click();

Thread.sleep(3000);

}

}

@Then("^verify the (.\*),(.\*),(.\*),(.\*) and (.\*) details for the (.\*) in RoleInformation section$")

public void verifyTheJobcodeOrg\_Hrchy\_LvlOrgBus\_UnitRBITier\_AndLending\_FundamentalsDetailsForTheRoleNameInTheRoleInformationSection(

String strJobcode,String strOrg\_Hrchy\_Lvl6, String strOrganization, String strBusinessUnit,

String strRBI, String strrolename) throws Throwable {

System.out.println("\n");

System.out.println("================== Role Name : "+strrolename+" ==================");

String strRolepath = "//div[@id='roleAssignmentRule']";

String strmatchlisttext = fetchmatchlist(strRolepath);

//System.out.println("final app text :" +strmatchlisttext);

String[] keyValuePairs = strmatchlisttext.split(" AND ");

for(int i=0;i<keyValuePairs.length;i++)

{

//System.out.println("pairs :"+keyValuePairs[i]);

String str1 = keyValuePairs[i].substring(1, keyValuePairs[i].length()-1);

//System.out.println("str1 : "+str1);

HashMap<String,String> map = new HashMap<String,String>();

String[] abc = str1.split("=");

map.put(abc[0].trim(), abc[1].trim());

//System.out.println("final :" +map.put(abc[0], abc[1]));

try

{

if(strJobcode.equals(map.put(abc[0].trim(), abc[1].trim())))

{

System.out.println("The given Job code 2 : "+strJobcode+" is matched with Job code "+map.put(abc[0].trim(), abc[1].trim())+" in the system");

}

else if(strOrg\_Hrchy\_Lvl6.equals(map.put(abc[0].trim(), abc[1].trim())))

{

System.out.println("The given Org Hrchy Lvl 6 : "+strOrg\_Hrchy\_Lvl6+" is matched with Org Hrchy Lvl 6 "+map.put(abc[0].trim(), abc[1].trim())+" in the system");

}

else if(strOrganization.equals(map.put(abc[0].trim(), abc[1].trim())))

{

System.out.println("The given Organization : "+strOrganization+" is matched with Organization "+map.put(abc[0].trim(), abc[1].trim())+" in the system");

}

else if(strBusinessUnit.equals(map.put(abc[0].trim(), abc[1].trim())))

{

System.out.println("The given Business Unit : "+strBusinessUnit+" is matched with Business Unit "+map.put(abc[0].trim(), abc[1].trim())+" in the system");

}

else if(strRBI.equals(map.put(abc[0].trim(), abc[1].trim())))

{

System.out.println("The given RBI : "+strRBI+" is matched with RBI "+map.put(abc[0].trim(), abc[1].trim())+" in the system");

}

}

catch(Exception e)

{

System.out.println("The given parameter is NOT matched with the system");

}

}

System.out.println("=================================================================");

System.out.println("\n");

}

/\*@Then("^verify the (.\*),(.\*),(.\*),(.\*),(.\*),(.\*) and (.\*) details for the (.\*) in RoleInformation section$")

public void verifyTheJobcodeOrg\_Hrchy\_LvlOrgBus\_UnitRBITier\_AndLending\_FundamentalsDetailsForTheRoleNameInTheRoleInformationSection(

String strJobcode,String strOrg\_Hrchy\_Lvl6, String strOrganization, String strBusinessUnit,

String strRBI, String strTier2, String strLndgfmtls, String strrolename) throws Throwable {

String strRolepath = "//div[@id='roleAssignmentRule']";

String strmatchlisttext = fetchmatchlist(strRolepath);

System.out.println("final app text :" +strmatchlisttext);

String strjc="Job Code";

String strohl = "Organization Hierarchy Level6";

String strorg = "Organization";

String strbunit = "Business Unit";

String strrbi = "RBI";

String strtier2 = "Tier 2";

String strlf = "Lending Fundamentals";

System.out.println("\n");

System.out.println("================== Role Name : "+strrolename+" ==================");

matchlistcheck(strmatchlisttext, strjc, strJobcode);

matchlistcheck(strmatchlisttext, strohl, strOrg\_Hrchy\_Lvl6);

matchlistcheck(strmatchlisttext, strorg, strOrganization);

matchlistcheck(strmatchlisttext, strbunit, strBusinessUnit);

matchlistcheck(strmatchlisttext, strrbi, strRBI);

matchlistcheck(strmatchlisttext, strtier2, strTier2);

matchlistcheck(strmatchlisttext, strlf, strLndgfmtls);

System.out.println("=================================================================");

System.out.println("\n");

}\*/

@And("^User clicks on Logout button$")

public void userClicksOnLogoutButton() throws Throwable {

WebElement maintab = driver.findElement(By.xpath("//div[@id='usernameMenu']"));

if(maintab.isDisplayed())

{

maintab.click();

boolean logoutflag = selectitem("Logout");

if(logoutflag != true)

{

driver.quit();

}

else if(logoutflag == true)

{

driver.quit();

}

}

}

public String fetchmatchlist(String strRoledtlspath)

{

String strmatchlisttext = null;

List<WebElement> ltmatchlist = driver.findElements(By.xpath(strRoledtlspath));

Iterator<WebElement> itrmatch = ltmatchlist.iterator();

while (itrmatch.hasNext())

{

WebElement webmatchtext = itrmatch.next();

List<WebElement> ltmatchlisttext = webmatchtext.findElements(By.tagName("td"));

Iterator<WebElement> itrmatchtext = ltmatchlisttext.iterator();

while (itrmatchtext.hasNext())

{

WebElement webmatchlisttext = itrmatchtext.next();

webmatchlisttext.click();

strmatchlisttext = webmatchlisttext.getText();

}

}

return strmatchlisttext;

}

@SuppressWarnings("unused")

public void matchlistcheck(String strmatchtext1,String strcode, String strmatchitem) throws IOException, InterruptedException

{

for(int i=0; i<=strmatchtext1.length(); i++)

{

if(strmatchtext1.contains(strmatchitem))

{

System.out.println("The given "+strcode+" : "+strmatchitem+" in schema matched with the "+strmatchtext1+"");

break;

}

else

{

Thread.sleep(1000);

System.out.println("The given "+strcode+" : "+strmatchitem+" in schema NOT MATCHED with the"+strmatchtext1+"");

screenshot(strmatchitem);

break;

}

}

}

public boolean selectitem(String selecttab)

{

List<WebElement> litem = driver.findElements(By.tagName("a"));

Iterator<WebElement> iitem = litem.iterator();

while(iitem.hasNext())

{

WebElement witem = iitem.next();

String stritem = witem.getText();

if(stritem.contains(selecttab))

{

witem.click();

break;

}

}

return true;

}

public void screenshot(String strerror) throws IOException {

DateFormat dateformat = new SimpleDateFormat("MM-dd-yy-hhmm");

Date date = new Date();

String tstamp = dateformat.format(date).toString();

TakesScreenshot scrshot = ((TakesScreenshot)driver);

File scrfile = scrshot.getScreenshotAs(OutputType.FILE);

//File destfile = new File("C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\screenshots\\"+strerror+"\_"+tstamp+".png");

File destfile = new File(prop.getProperty("Screenshot")+strerror+"\_"+tstamp+".png");

Files.copy(scrfile,destfile);

}

}

package cucumberTests;

import cucumber.api.CucumberOptions;

import cucumber.api.testng.AbstractTestNGCucumberTests;

@CucumberOptions(features="src/features",glue= {"stepdefinition"})

public class TestRunner extends AbstractTestNGCucumberTests{

}

Textng.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">

<suite name=*"IIQ Suite"*>

<test name=*"IIQ Role Verification"*>

<classes>

<class name=*"cucumberTests.TestRunner"*/>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

Working files

Feature file

Feature: Test the IIQ Roles functionality

Scenario Outline: Verifying the Role information

Given User launch the ANZ IIQ application

And enter Userid and Password and click on Submit button

When User clicks on Setup tab and choose Roles option

And enter <RoleName> in the search field

Then verify the <Jobcode>,<Org\_Hrchy\_Lvl6>,<Org>,<Bus\_Unit>,<RBI>,<Tier\_2> and <Lending\_Fundamentals> details for the <RoleName> in the RoleInformation section

And User clicks on Logout button

Examples:-

|RoleName |Jobcode|Org\_Hrchy\_Lvl6 |Org|Bus\_Unit|RBI |Tier\_2|Lending\_Fundamentals|

|ABN - Branch Manager (RBI Tier 2)|J00019 |Aust Branch Network |AU |ANZAU |true|true |false |

|ABN - Branch Manager (LOCAM) |hhhhhh |Aust Branch Network |kk |ANZAU |true|u65t |true |

Step definition

**package** stepdefinition;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.IOException;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.Properties;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.OutputType;

**import** org.openqa.selenium.TakesScreenshot;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**import** com.google.common.io.Files;

**import** cucumber.api.java.en.And;

**import** cucumber.api.java.en.Given;

**import** cucumber.api.java.en.Then;

**import** cucumber.api.java.en.When;

**import** junit.framework.~~Assert~~;

//@SuppressWarnings("deprecation")

@SuppressWarnings("deprecation")

**public** **class** iiqsd {

WebDriver driver;

Properties prop = **new** Properties();

@Given("^User launch the ANZ IIQ application$")

**public** **void** userLaunchtheANZIIQApplication() **throws** Throwable {

FileInputStream fileInput = **new** FileInputStream("C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\iiqprop.properties");

prop.load(fileInput);

//System.setProperty("webdriver.chrome.driver","C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\chromedriver\\chromedriver.exe");

System.*setProperty*("webdriver.chrome.driver",prop.getProperty("ChromeDriver"));

driver = **new** ChromeDriver();

driver.manage().deleteAllCookies();

driver.get(prop.getProperty("URL"));

driver.manage().timeouts().implicitlyWait(10000, TimeUnit.***SECONDS***);

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

~~Assert~~.~~assertEquals~~("Welcome to IdentityIQ", driver.findElement(By.*xpath*("//header/span")).getText());

}

@And("^enter Userid and Password and click on Submit button$")

**public** **void** theUserEntersUseridAndPassword() **throws** Throwable {

driver.findElement(By.*xpath*("//input[@placeholder='Username']")).sendKeys(prop.getProperty("Userid"));

driver.findElement(By.*xpath*("//input[@placeholder='Password']")).sendKeys(prop.getProperty("Password"));

driver.findElement(By.*xpath*("//input[@value='Login']")).click();

}

@When("^User clicks on Setup tab and choose Roles option$")

**public** **void** userClicksOnSetupTabAndChooseRoles() **throws** Throwable {

**boolean** setupflag = selectitem("Setup");

**if**(setupflag == **true**)

{

**boolean** rolesflag = selectitem("Roles");

**if**(rolesflag != **true**)

{

driver.quit();

}

}

~~Assert~~.~~assertEquals~~("Role Management", driver.findElement(By.*xpath*("//h1[contains(text(),'Role Management')]")).getText());

}

@And("^enter (.\*) in the search field$")

**public** **void** enterTheRoleName(String rolename) **throws** Throwable {

driver.findElement(By.*xpath*("//input[@placeholder='Enter a Role Name']")).sendKeys(rolename);

WebElement stritem = driver.findElement(By.*xpath*("//div[@class='sectionHeader']"));

**if**(stritem.getText().contains(rolename))

{

stritem.click();

Thread.*sleep*(3000);

}

}

@Then("^verify the (.\*),(.\*),(.\*),(.\*),(.\*),(.\*) and (.\*) details for the (.\*) in the RoleInformation section$")

**public** **void** verifyTheJobcodeOrg\_Hrchy\_LvlOrgBus\_UnitRBITier\_AndLending\_FundamentalsDetailsForTheRoleNameInTheRoleInformationSection(

String strJobcode,String strOrg\_Hrchy\_Lvl6, String strOrganization, String strBusinessUnit,

String strRBI, String strTier2, String strLndgfmtls, String strrolename) **throws** Throwable {

String strRolepath = "//div[@id='roleAssignmentRule']";

String strmatchlisttext = fetchmatchlist(strRolepath);

String strjc="Job Code";

String strohl = "Organization Hierarchy Level6";

String strorg = "Organization";

String strbunit = "Business Unit";

String strrbi = "RBI";

String strtier2 = "Tier 2";

String strlf = "Lending Fundamentals";

System.***out***.println("\n");

System.***out***.println("================== Role Name : "+strrolename+" ==================");

matchlistcheck(strmatchlisttext, strjc, strJobcode);

matchlistcheck(strmatchlisttext, strohl, strOrg\_Hrchy\_Lvl6);

matchlistcheck(strmatchlisttext, strorg, strOrganization);

matchlistcheck(strmatchlisttext, strbunit, strBusinessUnit);

matchlistcheck(strmatchlisttext, strrbi, strRBI);

matchlistcheck(strmatchlisttext, strtier2, strTier2);

matchlistcheck(strmatchlisttext, strlf, strLndgfmtls);

System.***out***.println("=================================================================");

System.***out***.println("\n");

}

@And("^User clicks on Logout button$")

**public** **void** userClicksOnLogoutButton() **throws** Throwable {

WebElement maintab = driver.findElement(By.*xpath*("//div[@id='usernameMenu']"));

**if**(maintab.isDisplayed())

{

maintab.click();

**boolean** logoutflag = selectitem("Logout");

**if**(logoutflag != **true**)

{

driver.quit();

}

**else** **if**(logoutflag == **true**)

{

driver.quit();

}

}

}

**public** String fetchmatchlist(String strRoledtlspath)

{

String strmatchlisttext = **null**;

List<WebElement> ltmatchlist = driver.findElements(By.*xpath*(strRoledtlspath));

Iterator<WebElement> itrmatch = ltmatchlist.iterator();

**while** (itrmatch.hasNext())

{

WebElement webmatchtext = itrmatch.next();

List<WebElement> ltmatchlisttext = webmatchtext.findElements(By.*tagName*("td"));

Iterator<WebElement> itrmatchtext = ltmatchlisttext.iterator();

**while** (itrmatchtext.hasNext())

{

WebElement webmatchlisttext = itrmatchtext.next();

webmatchlisttext.click();

strmatchlisttext = webmatchlisttext.getText();

}

}

**return** strmatchlisttext;

}

@SuppressWarnings("unused")

**public** **void** matchlistcheck(String strmatchtext1,String strcode, String strmatchitem) **throws** IOException, InterruptedException

{

**for**(**int** i=0; i<=strmatchtext1.length(); i++)

{

**if**(strmatchtext1.contains(strmatchitem))

{

System.***out***.println("The given "+strcode+" : "+strmatchitem+" in schema matched with the "+strmatchtext1+"");

**break**;

}

**else**

{

Thread.*sleep*(1000);

System.***out***.println("The given "+strcode+" : "+strmatchitem+" in schema NOT MATCHED with the"+strmatchtext1+"");

screenshot(strmatchitem);

**break**;

}

}

}

**public** **boolean** selectitem(String selecttab)

{

List<WebElement> litem = driver.findElements(By.*tagName*("a"));

Iterator<WebElement> iitem = litem.iterator();

**while**(iitem.hasNext())

{

WebElement witem = iitem.next();

String stritem = witem.getText();

**if**(stritem.contains(selecttab))

{

witem.click();

**break**;

}

}

**return** **true**;

}

**public** **void** screenshot(String strerror) **throws** IOException {

DateFormat dateformat = **new** SimpleDateFormat("MM-dd-yy-hhmm");

Date date = **new** Date();

String tstamp = dateformat.format(date).toString();

TakesScreenshot scrshot = ((TakesScreenshot)driver);

File scrfile = scrshot.getScreenshotAs(OutputType.***FILE***);

//File destfile = new File("C:\\Users\\moharas2\\eclipse-workspace\\iiqbdd\\screenshots\\"+strerror+"\_"+tstamp+".png");

File destfile = **new** File(prop.getProperty("Screenshot")+strerror+"\_"+tstamp+".png");

Files.*copy*(scrfile,destfile);

}

}

Miscellaneous

**package** resources;

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** test {

**public** **static** **void** main(String[] args)

{

/\* String value = "{first\_name = naresh,last\_name = kumar,gender = male}";

value = value.substring(1, value.length()-1); //remove curly brackets

System.out.println(value);

String[] keyValuePairs = value.split(","); //split the string to creat key-value pairs

//System.out.println(keyValuePairs);

Map<String,String> map = new HashMap<>();

for(String pair : keyValuePairs) //iterate over the pairs

{

String[] entry = pair.split("="); //split the pairs to get key and value

map.put(entry[0].trim(), entry[1].trim()); //add them to the hashmap and trim whitespaces

}

\*/

/\*String sinp = "(Job Code 2 = 'J00019') AND (Organisation Hierarchy Level 6 = 'AUST BRANCH')";

//sinp = sinp.substring(1, sinp.length()-1);

System.out.println("sinp"+sinp);

String[] keyValuePairs = sinp.split(" AND ");

for(int i=0;i<keyValuePairs.length;i++)

{

//System.out.println("pairs :"+keyValuePairs[i]);

String str1 = keyValuePairs[i].substring(1, keyValuePairs[i].length()-1);

//System.out.println("str1 : "+str1);

HashMap<String,String> map = new HashMap<String,String>();

String[] abc = str1.split("=");

map.put(abc[0], abc[1]);

System.out.println("final :" +map.put(abc[0], abc[1]));

}\*/

HashMap<String, String> myMap = **new** HashMap<String, String>();

String s = "SALES:0,SALE\_PRODUCTS:1,EXPENSES:2,EXPENSES\_ITEMS:3";

String[] pairs = s.split(",");

**for** (**int** i=0;i<pairs.length;i++) {

String pair = pairs[i];

String[] keyValue = pair.split(":");

myMap.put(keyValue[0], (keyValue[1]));

System.***out***.println("final :" +myMap.put(keyValue[0], (keyValue[1])));

}

}

}