package bilTechnical

import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint

import static com.kms.katalon.core.testcase.TestCaseFactory.findTestCase

import static com.kms.katalon.core.testdata.TestDataFactory.findTestData

import static com.kms.katalon.core.testobject.ObjectRepository.findTestObject

import org.junit.After

import com.kms.katalon.core.annotation.Keyword

import com.kms.katalon.core.checkpoint.Checkpoint

import com.kms.katalon.core.checkpoint.CheckpointFactory

import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords

import com.kms.katalon.core.model.FailureHandling

import com.kms.katalon.core.testcase.TestCase

import com.kms.katalon.core.testcase.TestCaseFactory

import com.kms.katalon.core.testdata.TestData

import com.kms.katalon.core.testdata.TestDataFactory

import com.kms.katalon.core.testobject.ObjectRepository

import com.kms.katalon.core.testobject.TestObject

import com.kms.katalon.core.util.KeywordUtil

import com.kms.katalon.core.webservice.keyword.WSBuiltInKeywords

import com.kms.katalon.core.webui.keyword.WebUiBuiltInKeywords

import com.kms.katalon.core.webui.keyword.builtin.ScrollToElementKeyword

import internal.GlobalVariable

import io.appium.java\_client.MobileDriver

import io.appium.java\_client.MobileElement

import WSBuiltInKeywords as WS

import WebUiBuiltInKeywords as WebUI

import com.kms.katalon.core.mobile.keyword.MobileBuiltInKeywords as Mobile

public class MOBILE {

@Keyword

def ScrollToElement(TestObject tmpTestObject){

int centerOfTheMobileX = (Mobile.getDeviceWidth()/2).intValue()

int centerOfTheMobileY = (Mobile.getDeviceHeight()/2).intValue()

int endCoordinateY = (Mobile.getDeviceHeight()/4).intValue()

int relativePositionY = endCoordinateY - centerOfTheMobileY;

int nbrOfSwipesAllowed = 4

KeywordUtil.logInfo("Center of Y : "+centerOfTheMobileY + " End Coordinates Y : "+endCoordinateY+" Device OS : "+ Mobile.getDeviceOS().toString())

KeywordUtil.logInfo("Center of Mobile X : "+centerOfTheMobileX + " Center of Mobile Y : "+centerOfTheMobileY)

Mobile.delay(1, FailureHandling.STOP\_ON\_FAILURE)

KeywordUtil.logInfo("Checking on the current page")

if(Mobile.verifyElementVisible(tmpTestObject, 1, FailureHandling.OPTIONAL)){

KeywordUtil.markPassed("Scroll was not needed - Element found")

return

}

KeywordUtil.logInfo("Scrolling on the page to find it")

int tmpNbrOfSwipes=0;

while (true) {

if(GlobalVariable.currentOs.equals("iOS"))

Mobile.swipe(centerOfTheMobileX, centerOfTheMobileY,centerOfTheMobileX, -1 \* endCoordinateY, FailureHandling.STOP\_ON\_FAILURE)

else{

Mobile.swipe(centerOfTheMobileX, centerOfTheMobileY,centerOfTheMobileX, endCoordinateY , FailureHandling.STOP\_ON\_FAILURE)

KeywordUtil.logInfo("Android")

}

tmpNbrOfSwipes++

//Mobile.verifyElementExist(tmpTestObject,1,FailureHandling.OPTIONAL)

if(Mobile.verifyElementVisible(tmpTestObject, 1, FailureHandling.OPTIONAL)){

break

}

if(tmpNbrOfSwipes==nbrOfSwipesAllowed){

KeywordUtil.markFailed("Element not found after - "+tmpNbrOfSwipes+"/"+nbrOfSwipesAllowed+" tentatives")

return

}

}

KeywordUtil.logInfo("End of the process")

KeywordUtil.markPassed("Element found after "+tmpNbrOfSwipes+"/"+nbrOfSwipesAllowed+" tentatives")

}

@Keyword

def ScrollToElement(TestObject tmpTestObject, int nbrOfSwipesAllowed){

int centerOfTheMobileX = (Mobile.getDeviceWidth()/2).intValue()

int centerOfTheMobileY = (Mobile.getDeviceHeight()/2).intValue()

int endCoordinateY = (Mobile.getDeviceHeight()/4).intValue()

Mobile.delay(1, FailureHandling.STOP\_ON\_FAILURE)

KeywordUtil.logInfo("Checking on the current page")

if(Mobile.verifyElementExist(tmpTestObject,1,FailureHandling.OPTIONAL)){

KeywordUtil.markPassed("Scroll was not needed - Element found")

return

}

if(nbrOfSwipesAllowed==0)

{

KeywordUtil.markWarning("No Scroll Requested")

return

}

KeywordUtil.logInfo("Scrolling on the page to find it")

int tmpNbrOfSwipes= 0;

while (true) {

Mobile.swipe(centerOfTheMobileX, centerOfTheMobileY,centerOfTheMobileX, endCoordinateY , FailureHandling.STOP\_ON\_FAILURE)

tmpNbrOfSwipes++

if(Mobile.verifyElementExist(tmpTestObject,1,FailureHandling.OPTIONAL)){

break

}

if(tmpNbrOfSwipes==nbrOfSwipesAllowed){

KeywordUtil.markFailed("Element not found after - "+tmpNbrOfSwipes+"/"+nbrOfSwipesAllowed+" tentatives")

return

}

}

KeywordUtil.logInfo("End of the process")

KeywordUtil.markPassed("Element found after "+tmpNbrOfSwipes+"/"+nbrOfSwipesAllowed+" tentatives")

}

@Keyword

def setTextAndHideKeyboard(TestObject to, String stringValue, int timeout){

Mobile.setText(to, stringValue, timeout, FailureHandling.STOP\_ON\_FAILURE)

Mobile.hideKeyboard(FailureHandling.CONTINUE\_ON\_FAILURE)

}

@Keyword

def hideKeyboardByClickingOnTheScreen(){

int whereToTapY = (Mobile.getDeviceHeight()/2).intValue()

KeywordUtil.logInfo("Where to Tap Y : "+whereToTapY)

Mobile.tapAtPosition(10, whereToTapY)

}

}