Baseclass.java

|  |
| --- |
| package com.inetbankingv1.testcases;  import java.io.File;  import java.io.IOException;  import java.util.concurrent.TimeUnit;  import org.apache.commons.io.FileUtils;  import org.apache.commons.lang3.RandomStringUtils;  //import org.apache.log4j.PropertyConfigurator;  import org.apache.log4j.Logger;  import org.apache.log4j.PropertyConfigurator;  import org.openqa.selenium.OutputType;  import org.openqa.selenium.TakesScreenshot;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.edge.EdgeDriver;  import org.openqa.selenium.firefox.FirefoxDriver;  import org.openqa.selenium.firefox.GeckoDriverService;  import org.testng.annotations.AfterClass;  import org.testng.annotations.BeforeClass;  //import org.testng.log4testng.Logger;  import org.testng.annotations.Parameters;  import com.inetbankingv1.utilities.ReadConfig;  import net.bytebuddy.implementation.FieldAccessor.PropertyConfigurable;  public class BaseClass {  ReadConfig readConfig = new ReadConfig();    public String baseURL = readConfig.getApplicationURL() ;  public String userName =readConfig.getUserName() ;  public String password =readConfig.getPassword();    public static WebDriver driver;  public static Logger logger;    @Parameters("browser")  @BeforeClass  public void setUp(String br) {  logger = Logger.getLogger("iNetBanking");  PropertyConfigurator.configure("log4j.properties");    if(br.equals("chrome")) {  System.setProperty("webdriver.chrome.driver", readConfig.getChromepath());  driver= new ChromeDriver();  }  else if(br.equals("firefox")) {  System.setProperty("webdriver.gecko.driver", readConfig.getFirefoxpath());  driver= new FirefoxDriver();  }  else if(br.equals("edge")) {  System.out.print("Calling from edge browser");  System.setProperty("webdriver.edge.driver", readConfig.getEdgepath());  System.out.print("opened with edge browser");  driver= new EdgeDriver();  }  driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  driver.get(baseURL);    }    @AfterClass  public void tearDown() {  //driver.quit();  }    public void captureScreen(WebDriver driver, String tname) throws IOException {  TakesScreenshot ts = (TakesScreenshot) driver;  File source = ts.getScreenshotAs(OutputType.FILE);  File target = new File(System.getProperty("user.dir")+"/Screenshots/"+tname +".png");  FileUtils.copyFile(source, target);  System.out.println("Screenshot tken");  }    public String randomString() {  String generatedString =RandomStringUtils.randomAlphabetic(8);  return generatedString;  }    public static String RandomNum() {  String gereratedNum = RandomStringUtils.randomNumeric(5);  return gereratedNum;  }  } |

AddCustomerPage.java

|  |
| --- |
| package com.inetbankingv1.pageobjects;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.support.FindBy;  import org.openqa.selenium.support.How;  import org.openqa.selenium.support.PageFactory;  public class AddCustomerPage {  WebDriver lDriver;  public AddCustomerPage(WebDriver rDriver){  lDriver=rDriver;  PageFactory.initElements(rDriver, this);  }    @FindBy(how =How.XPATH, using ="/html/body/div[3]/div/ul/li[2]/a")  WebElement lnkAddNewCustomer;    @FindBy(how =How.NAME, using ="name")  WebElement txtCustomerName;    @FindBy(how =How.NAME, using ="rad1")  WebElement rdGender;    @FindBy(how =How.NAME, using ="dob")  WebElement txtdob;    @FindBy(how =How.NAME, using ="addr")  WebElement txtaddr;    @FindBy(how =How.NAME, using ="city")  WebElement txtcity;    @FindBy(how =How.NAME, using ="state")  WebElement txtstate;    @FindBy(how =How.NAME, using ="pinno")  WebElement txtpinno;    @FindBy(how =How.NAME, using ="telephoneno")  WebElement txtMobile;    @FindBy(how =How.NAME, using ="emailid")  WebElement txtemailid;    @FindBy(how =How.NAME, using ="password")  WebElement txtpassword;    @FindBy(how =How.NAME, using ="sub")  WebElement btnSubmit;    @FindBy(how =How.NAME, using ="res")  WebElement btnReset;      public void clickAddNewCustomer() {  lnkAddNewCustomer.click();  }    public void SetCustName(String cName) {  txtCustomerName.sendKeys(cName);  }    public void setGender(String cGender) {  rdGender.click();;  }    public void setDOB(String mm, String dd, String yy) {  txtdob.sendKeys(mm);  txtdob.sendKeys(dd);  txtdob.sendKeys(yy);  }    public void setAddress(String cAddress) {  txtaddr.sendKeys(cAddress);  }    public void setCity(String cCity) {  txtcity.sendKeys(cCity);  }  public void setState(String cState) {  txtstate.sendKeys(cState);  }    public void setPin(String cPin) {  txtpinno.sendKeys(String.valueOf(cPin));  }    public void setMobile(String cMobile) {  txtMobile.sendKeys(cMobile);  }    public void setEmail(String cEmail) {  txtemailid.sendKeys(cEmail);  }    public void setPassword(String cPassword) {  txtpassword.sendKeys(cPassword);  }    public void clickSubmit() {  btnSubmit.click();;  }  } |

LoginPage.java

|  |
| --- |
| package com.inetbankingv1.pageobjects;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.support.CacheLookup;  import org.openqa.selenium.support.FindBy;  import org.openqa.selenium.support.PageFactory;  public class LoginPage {  WebDriver lDriver;    public LoginPage(WebDriver rDriver){  lDriver=rDriver;  PageFactory.initElements(rDriver, this);  }    @FindBy(name="uid")  @CacheLookup  WebElement txtUserName;    @FindBy(name ="password")  @CacheLookup  WebElement txtPassword;    @FindBy(name ="btnLogin")  @CacheLookup  WebElement btnLogin;    //Logout link  @FindBy(xpath = "/html/body/div[3]/div/ul/li[15]/a")  @CacheLookup  WebElement lnkLogout;    public void setUserName(String uname) {  txtUserName.sendKeys(uname);  }    public void setPassword(String pwd) {  txtPassword.sendKeys(pwd);  }    public void clickSubmit() {  btnLogin.click();  }    public void clickLogout() {  lnkLogout.click();  }  } |

TC\_Login\_001.java

|  |
| --- |
| package com.inetbankingv1.testcases;  import java.io.IOException;  import org.junit.Assert;  //import org.openqa.selenium.WebDriver;  import org.testng.annotations.Test;  import com.inetbankingv1.pageobjects.LoginPage;  public class TC\_Login\_001 extends BaseClass {  @Test  public void loginTest() throws InterruptedException, IOException {    logger.info("Login-Loggin page");  Thread.sleep(5000);  LoginPage lp = new LoginPage(driver);    lp.setUserName(userName);  logger.info("Login-Entered username");    lp.setPassword(password);  logger.info("Login-Entered password");    lp.clickSubmit();  System.out.println("Submit clicked");  logger.info("Login-Clicked the submit");  System.out.println(driver.getTitle());    if(driver.getTitle().equals("Guru99 Bank Manager HomePage")) {  Assert.assertTrue(true);  logger.info("Login-Login Test passed");  }  else {  captureScreen(driver, "loginTest");  System.out.println("Test case failed. Screenshot taken");  Assert.assertTrue(false);  logger.info("Login-Login Test failed");  }    }  } |

TC\_LoginDDT\_002

|  |
| --- |
| package com.inetbankingv1.testcases;  import java.io.IOException;  import org.junit.Assert;  import org.openqa.selenium.NoAlertPresentException;  import org.testng.annotations.DataProvider;  import org.testng.annotations.Test;  import com.inetbankingv1.pageobjects.LoginPage;  import com.inetbankingv1.utilities.XLUtils;  public class TC\_LoginDDT\_002 extends BaseClass {    @Test(dataProvider="LoginData")  public void loginDDT(String user, String pwd) throws IOException {  LoginPage lp = new LoginPage(driver);    lp.setUserName(user);  logger.info("'TC\_LoginDDT\_002'- Username provided");    lp.setPassword(pwd);  System.out.println("User = " + user +" Password = " + pwd);  logger.info("'TC\_LoginDDT\_002'- Password provided");  lp.clickSubmit();    if (isAlertPresent()==true) {  captureScreen(driver, "loginDDT");  driver.switchTo().alert().accept();  driver.switchTo().defaultContent();  Assert.assertTrue(false);  logger.warn("'TC\_LoginDDT\_002'- DDT Test failed");  }  else {  Assert.assertTrue(true);  logger.info("'TC\_LoginDDT\_002'- DDT Login passed");  lp.clickLogout();  driver.switchTo().alert().accept();  driver.switchTo().defaultContent();  }  }      public boolean isAlertPresent() {  try {  driver.switchTo().alert();  System.out.println("Alert is present-- Invalid user");  return true;  }  catch(NoAlertPresentException e) {  System.out.println("Inside exception. Alert is not present-- Valid user");  return false;  }  }    @DataProvider(name ="LoginData")  String[][] getData() throws IOException{  String path = System.getProperty("user.dir")+ "/src/test/java/com/inetbankingv1/testdata/LoginData1.xlsx";  int rownum = XLUtils.getRowCount(path, "Sheet1");  int colcount = XLUtils.getCellCount(path, "Sheet1", 1);  String logindata[][] = new String[rownum][colcount];  for (int i =1; i<=rownum ; i++) {  for(int j=0; j<colcount;j++) {  logindata[i-1][j]= XLUtils.getCellData(path, "Sheet1", i, j);  }  }  return logindata;  }  } |

TC\_AddNewCustomerTest\_003.java

|  |
| --- |
| package com.inetbankingv1.testcases;  import java.io.IOException;  import org.apache.commons.lang3.RandomStringUtils;  import org.junit.Assert;  import org.testng.annotations.Test;  import com.inetbankingv1.pageobjects.AddCustomerPage;  import com.inetbankingv1.pageobjects.LoginPage;  public class TC\_AddNewCustomerTest\_003 extends BaseClass{  @Test  public void addNewCustomer() throws InterruptedException, IOException {  //Create loginpage instance  LoginPage lp= new LoginPage(driver);  Thread.sleep(3000);  lp.setUserName(userName);  lp.setPassword(password);  lp.clickSubmit();  logger.info("AddNewCustomer-- Logged in successfully");  Thread.sleep(3000);    //Create AddCustomerPage instance  AddCustomerPage addCustomer = new AddCustomerPage(driver);  addCustomer.clickAddNewCustomer();  logger.info("AddNewCustomer-- Clicked New Customer");  Thread.sleep(5000);    logger.info("AddNewCustomer-- Setting Customer details");  addCustomer.SetCustName("Simba");  addCustomer.setGender("male");  addCustomer.setDOB("05", "15", "2022");  Thread.sleep(2000);  addCustomer.setAddress("My new Ddress");  addCustomer.setCity("Fremont");  addCustomer.setState("CA");  addCustomer.setPin("123456");  addCustomer.setMobile("22334455");    String email = randomString()+"@gmail.com";  addCustomer.setEmail(email);  logger.info("AddNewCustomer-- Generated Random email");  addCustomer.setPassword("simba");  addCustomer.clickSubmit();    boolean res = driver.getPageSource().contains("Customer Registered Successfully");  try {  if (res==true) {  logger.info("AddNewCustomer--Test case passed");  Assert.assertTrue(true);  }  else {  logger.info("AddNewCustomer-- Test case failed");  captureScreen(driver, "addNewCustomer");  Assert.assertTrue(false);  }  } catch (IOException e) {  e.printStackTrace();  }  }  } |

LoginData1.xls

|  |  |
| --- | --- |
| username | password |
| aaa | gYqAbEz |
| bbb | gYqAbEz |
| mngr398351 | gYqAbEz |
| test | fail |
| test2 | fail2 |

ReadConfig.java

|  |
| --- |
| package com.inetbankingv1.utilities;  import java.io.File;  import java.io.FileInputStream;  import java.util.Properties;  public class ReadConfig {  Properties pro;    public ReadConfig() {  File src = new File("./Configrations/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("baseURL");  return url;  }    public String getUserName() {  String userName = pro.getProperty("userName");  return userName;  }    public String getPassword() {  String password = pro.getProperty("password");  return password;  }    public String getFirefoxpath() {  String firefoxpath = pro.getProperty("firefoxpath");  return firefoxpath;  }    public String getChromepath() {  String chromepath = pro.getProperty("chromepath");  return chromepath;  }    public String getEdgepath() {  String edgepath = pro.getProperty("edgepath");  return edgepath;  }  } |

Reporting.java

|  |
| --- |
| package com.inetbankingv1.utilities;  import java.io.File;  import java.io.IOException;  import java.text.SimpleDateFormat;  import java.util.Date;  import org.testng.ITestContext;  import org.testng.ITestResult;  import org.testng.TestListenerAdapter;  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;  public class Reporting extends TestListenerAdapter{  public ExtentHtmlReporter htmlReporter;  public ExtentReports extent;  public ExtentTest extentLogger;      public void onStart(ITestContext testContext) {  String timeStamp = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new Date());  String repName ="San\_Test-Report"+timeStamp+".html";  htmlReporter = new ExtentHtmlReporter (System.getProperty("user.dir")+"/test-output/"+repName);  htmlReporter.loadXMLConfig(System.getProperty("user.dir")+"/extent-config.xml");    System.out.println("RepName =" +repName);  System.out.println("here Report location = " + System.getProperty("user.dir")+"/test-output/"+repName);  //System.out.println("here Report location = " + System.getProperty("user.dir")+"/test-output/"+ repName);    extent = new ExtentReports();  extent.attachReporter(htmlReporter);  extent.setSystemInfo("Host name", "localhost");  extent.setSystemInfo("Environment", "QA");  extent.setSystemInfo("User", "Simba");    htmlReporter.config().setDocumentTitle("INetBanking project");  htmlReporter.config().setReportName("Functional Test Report");  htmlReporter.config().setTestViewChartLocation(ChartLocation.TOP);  htmlReporter.config().setTheme(Theme.DARK);  }    public void onTestSuccess(ITestResult tr) {  System.out.println("Inside onTestSuccess");  extentLogger= extent.createTest(tr.getName());  extentLogger.log(Status.PASS, MarkupHelper.createLabel(tr.getName(),ExtentColor.GREEN ));  System.out.println("Created the report onTestSuccess");  }    public void onTestFailure(ITestResult tr) {  extentLogger= extent.createTest(tr.getName());  extentLogger.log(Status.FAIL, MarkupHelper.createLabel(tr.getName(),ExtentColor.RED ));    String screenshotPath= System.getProperty("user.dir")+"\\Screenshots\\"+tr.getName()+".png";  File f = new File(screenshotPath);  if (f.exists()) {  try {  extentLogger.fail("Srreenshot is below:"+ extentLogger.addScreenCaptureFromPath(screenshotPath) );  }  catch(IOException e) {  e.printStackTrace();  }  }  }      public void onTestSkipped(ITestResult tr) {  extentLogger= extent.createTest(tr.getName());  extentLogger.log(Status.SKIP, MarkupHelper.createLabel(tr.getName(),ExtentColor.ORANGE ));  }    public void onFinish(ITestContext testContext) {  System.out.println("Inside onFinish");  extent.flush();  }  } |

XLUtils.java

|  |
| --- |
| package com.inetbankingv1.utilities;  import java.io.FileInputStream;  import java.io.FileNotFoundException;  import java.io.FileOutputStream;  import java.io.IOException;  import org.apache.poi.ss.usermodel.DataFormatter;  import org.apache.poi.xssf.usermodel.XSSFCell;  import org.apache.poi.xssf.usermodel.XSSFRow;  import org.apache.poi.xssf.usermodel.XSSFSheet;  import org.apache.poi.xssf.usermodel.XSSFWorkbook;  public class XLUtils {  public static FileInputStream fi;  public static FileOutputStream fo;  public static XSSFWorkbook wb;  public static XSSFSheet ws;  public static XSSFRow row;  public static XSSFCell cell;      public static int getRowCount(String xlFile, String xlSheet) throws IOException {  fi=new FileInputStream(xlFile);  wb = new XSSFWorkbook(fi);  ws = wb.getSheet(xlSheet);  int rowcount = ws.getLastRowNum();  wb.close();  fi.close();  return rowcount;  }    public static int getCellCount(String xlFile, String xlSheet, int rownum) throws IOException {  fi=new FileInputStream(xlFile);  wb = new XSSFWorkbook(fi);  ws = wb.getSheet(xlSheet);  row = ws.getRow(rownum);  int cellcount = row.getLastCellNum();  wb.close();  fi.close();  return cellcount;  }    public static String getCellData(String xlFile, String xlSheet, int rownum, int colnum) throws IOException {  fi=new FileInputStream(xlFile);  wb = new XSSFWorkbook(fi);  ws = wb.getSheet(xlSheet);  row = ws.getRow(rownum);  cell= row.getCell(colnum);  String data;  try {  DataFormatter formatter = new DataFormatter();  String cellData = formatter.formatCellValue(cell);  return cellData;  }  catch(Exception e){  //e.printStackTrace();  data="";  }  wb.close();  fi.close();  return data;  }      public void setCellData(String xlFile, String xlSheet, int rownum, int colnum, String data) throws IOException {  fi=new FileInputStream(xlFile);  wb = new XSSFWorkbook(fi);  ws = wb.getSheet(xlSheet);  row = ws.getRow(rownum);  cell = row.createCell(colnum);  cell.setCellValue(data);  fo = new FileOutputStream(xlFile);  wb.write(fo);  wb.close();  fi.close();  fo.close();  }  } |

TestNG.xml

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >  <suite name="inetBankingV1">  <listeners>  <listener class-name="com.inetbankingv1.utilities.Reporting"></listener>  </listeners>  <test name="BankingApp Test">  <parameter name="browser" value="chrome"/>  <classes>  <class name="com.inetbankingv1.testcases.TC\_Login\_001"/>  <!-- <class name="com.inetbankingv1.testcases.TC\_LoginDDT\_002"/> -->  <class name="com.inetbankingv1.testcases.TC\_AddNewCustomerTest\_003"/>  </classes>  </test>  </suite> |

Run.bat

|  |
| --- |
| cd C:\Users\TISHAN1KA\eclipse-workspace\iNetBankingV1  mvn clean install |

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

CucumberFramework

TestRunner.java

|  |
| --- |
| package runners;  package runners;  import java.io.\*;  import org.junit.AfterClass;  import org.junit.runner.RunWith;  import com.cucumber.listener.Reporter;  //import cucumber.api.CucumberOptions;  //import cucumber.api.junit.Cucumber;  import managers.FileReaderManager;  import io.cucumber.junit.Cucumber;  import io.cucumber.junit.CucumberOptions;  @RunWith(Cucumber.class)  @CucumberOptions(  features ="src\\test\\resources\\functionalTests"  ,glue= {"stepDefinitions"}  //,plugin = {"pretty", "com.cucumber.listener.ExtentCucumberFormatter:target/cucumber-reports/report.html"}  ,plugin = {"pretty"  , "junit:target/cucumber-reports/cucumber.xml"  , "json:target/cucumber-reports/cucumber\_report.json"  , "html:target/cucumber-reports/cucumber\_report.html"  }    ,monochrome = true  )    public class TestRunner {  @AfterClass  public static void writeExtentReport() {  //System.out.println("TestRunner -- writeExtentReport stsrt");  //Reporter.loadXMLConfig(new File(FileReaderManager.getInstance().getConfigReader().getReportConfigPath()));  //System.out.println("TestRunner -- writeExtentReport end");  }  } |

End2EndTest.feature

|  |
| --- |
| Feature: Automated End2End Tests  Description: To test End2End automation Tests  Scenario Outline: Customer places an order by purchasing an item from search  Given User is on home page  When he searhes for "dress"  And choose to buy first item  And move to checkout from mini cart  And enter "<customer>" personal details on checkout page  And enters shipping details  And select payment method as "check" payment  And place the order  Then verify the order details  Examples:  |customer|  |Lakshay| |

ProductListingPage.java

|  |
| --- |
| package pageObjects;  import java.util.List;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.support.FindAll;  import org.openqa.selenium.support.FindBy;  import org.openqa.selenium.support.How;  import org.openqa.selenium.support.PageFactory;  import org.openqa.selenium.support.ui.Select;  //import xxx\_dataProviders.ConfigFileReader;  public class ProductListingPage {    public ProductListingPage(WebDriver driver) {  System.out.println("ProductListingPage -- running constructor");  PageFactory.initElements(driver, this);  System.out.println("Inside ProductListingPage ");  }      @FindBy(how=How.XPATH, using ="//\*[@id=\"pa\_color\"]" )  private WebElement dropdown\_color;    @FindBy(how=How.XPATH, using ="//\*[@id=\"pa\_size\"]" )  private WebElement dropdown\_size;  // "//\*[@id=\"product-1467\"]/div[1]/div[2]/form/div/div[2]/button"  // "//\*[@id=\"product-1467\"]/div[1]/div[2]/div[2]/div/a"  //button.single\_add\_to\_cart\_button    @FindBy(how = How.CSS, using = "button.single\_add\_to\_cart\_button")  private WebElement btn\_AddToCart;    //@FindBy(how = How.CSS, using = "button.single\_add\_to\_cart\_button")  //private WebElement btn\_AddToCart;    //original  //@FindBy(how = How.XPATH, using ="//\*[@id=\"product-1473\"]/div[1]/div[2]/form/div/div[2]/button")  //private WebElement btn\_AddToCart;    @FindAll(@FindBy(how = How.XPATH, using ="//div[@class=\"noo-product-inner\"]"))  private List<WebElement> prd\_List;    public void select\_Color() throws InterruptedException {  try {  System.out.println("inside Select\_color-- getText()=" + dropdown\_color.getText() );  Select clr = new Select(dropdown\_color);  //System.out.println("Select\_color -- Creating Select object");  clr.selectByIndex(1);  //System.out.println("Select\_color -- SeelectByIndex1");  System.out.println("color selected");  }  catch(Exception e) {  System.out.println("Error Occured" );  System.out.println(e);    }  }    public void select\_Size() {  System.out.println("inside select\_Size");  Select sze = new Select(dropdown\_size);  sze.selectByIndex(1);  System.out.println("size Selected");  }    public void clickOn\_AddToCart() throws InterruptedException {  System.out.println("inside clickOn\_AddToCart. size prd\_List=" + prd\_List.size());  btn\_AddToCart.click();  Thread.sleep(5000);  System.out.println("Added to cart");  }    public void select\_Product(int productNumber) {  System.out.println("inside select\_Product. prd\_List size = " + prd\_List.size());  prd\_List.get(productNumber).click();  }    public String getProductname(int productNumber) {  //return prd\_List.get(productNumber).getText();  System.out.println("ProductListingPage -- getProductname =" + prd\_List.get(productNumber).findElement(By.cssSelector("h3")).getText());  return prd\_List.get(productNumber).findElement(By.cssSelector("h3")).getText();  }  } |

Hooks.java

|  |
| --- |
| package stepDefinitions;  import cucumber.TestContext;  import io.cucumber.java.After;  import io.cucumber.java.Before;  //import cucumber.api.java.Before;  public class Hooks {  TestContext testContext;    public Hooks(TestContext context) {  testContext = context;  }      @Before  public void BeforeSteps() {  System.out.println("Hooks -- Executing BeforeSteps");  /\*What all you can perform here  Starting a webdriver  Setting up DB connections  Setting up test data  Setting up browser cookies  Navigating to certain page  or anything before the test  \*/  }    @Before  public void BeforeSteps\_second\_method() {  System.out.println("Hooks -- Executing BeforeSteps\_second\_method");  }    @After  public void AfterSteps() {  System.out.println("Close the driver here");  //testContext.getWebDriverManager().closeDriver();  }      @After  public void AfterSteps\_second\_method() {  System.out.println("Hooks -- AfterSteps\_second\_method");  //testContext.getWebDriverManager().closeDriver();  }  } |

ProductPageSteps.java

|  |
| --- |
| package stepDefinitions;  import cucumber.TestContext;  import enums.Context;  import io.cucumber.java.en.When;  import pageObjects.ProductListingPage;  public class ProductPageSteps {  TestContext testContext;  ProductListingPage productListingPage;      public ProductPageSteps(TestContext context) {  System.out.println("ProductPageSteps TestRunner initiated");  testContext = context;  productListingPage = testContext.getPageObjectManager().getProductListingPage();  }    @When("^choose to buy first item$")  public void choose\_to\_buy\_first\_item() throws Throwable {    String productName = productListingPage.getProductname(1);  testContext.getscenarioContext().setContext(Context.PRODUCT\_NAME, productName);    productListingPage.select\_Product(1);  productListingPage.select\_Color();  productListingPage.select\_Size();        productListingPage.clickOn\_AddToCart();  }      } |

Customer.json

|  |
| --- |
| [  {  "firstName": "Lakshay",  "lastName": "Sharma",  "age": 35,  "emailAddress": "Lakshay@Gmail.com",  "address": {  "streetAddress": "Shalimar Bagh",  "city": "Delhi",  "postCode": "110088",  "state": "Delhi",  "country": "India",  "county": "Delhi"  },  "phoneNumber": {  "home": "012345678",  "mob": "0987654321"  }  },  {  "firstName": "Virender",  "lastName": "Singh",  "age": 35,  "emailAddress": "Virender@Gmail.com",  "address": {  "streetAddress": "Palam Vihar",  "city": "Gurgaon",  "postCode": "122345",  "state": "Haryana",  "country": "India",  "county": "Delhi"  },  "phoneNumber": {  "home": "012345678",  "mob": "0987654321"  }  }  ] |

Steps.java

|  |
| --- |
| package stepDefinitions;  import java.util.List;  import java.util.concurrent.TimeUnit;  import org.openqa.selenium.By;  import org.openqa.selenium.JavascriptExecutor;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.WebElement;  import org.openqa.selenium.chrome.ChromeDriver;  import org.openqa.selenium.support.ui.Select;  import io.cucumber.java.en.Given;  import io.cucumber.java.en.When;  //import cucumber.api.PendingException;  //import cucumber.api.java.en.Given;  //import cucumber.api.java.en.When;  public class Steps {  public WebDriver driver;    @Given("^User is on home page$")  public void user\_is\_on\_home\_page() throws Throwable {  System.out.println("Inside ");  System.setProperty("webdriver.chrome.driver", "C:\\Drivers\\chromedriver\_win32\\chromedriver.exe");    driver = new ChromeDriver();  driver.manage().window().maximize();  driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS );  driver.get("https://shop.demoqa.com");    }  @When("^he searhes for \"([^\"]\*)\"$")  public void he\_searhes\_for(String arg1) throws Throwable {  System.out.println("Inside he\_searhes\_for ");    driver.navigate().to("https://shop.demoqa.com/?s="+"dress" + "&post\_type=product");    }  @When("^choose to buy first item$")  public void choose\_to\_buy\_first\_item() throws Throwable {  System.out.println("Inside choose\_to\_buy\_first\_item ");    List <WebElement> items = driver.findElements(By.xpath("//div[@class=\"noo-product-inner\"]"));  if(items.size()>0) {  items.get(0).click();    }else {  System.out.println("No products were found matching your selection.");    }  }  @When("^move to checkout from mini cart$")  public void move\_to\_checkout\_from\_mini\_cart() throws Throwable {    System.out.println("Inside move\_to\_checkout\_from\_mini\_cart ");    WebElement addToCart = driver.findElement(By.xpath("//button[@class=\"single\_add\_to\_cart\_button button alt disabled wc-variation-selection-needed\"]"));    WebElement color = driver.findElement(By.xpath("//\*[@id=\"pa\_color\"]"));    System.out.print("Selecting color");  Select clr = new Select(color);  clr.selectByIndex(1);    System.out.println("Selecting Size");  WebElement size = driver.findElement(By.xpath("//\*[@id=\"pa\_size\"]"));  Select sze = new Select(size);  sze.selectByIndex(1);  addToCart.click();    //click on view cart  System.out.println("View cart");  driver.findElement(By.xpath("//\*[@id=\"nav-menu-item-cart\"]/a/span/i")).click();    //click on checkout  System.out.println("Proceed to Checking out");  driver.findElement(By.xpath("//\*[@id=\"post-6\"]/div/div/div[2]/div[2]/div/a")).click();    }    @When("^enters shipping details$")  public void enters\_shipping\_details() throws Throwable {  // Write code here that turns the phrase above into concrete actions  System.out.println("Inside enters\_shipping\_details");  }  @When("^select payment method as \"([^\"]\*)\" payment$")  public void select\_payment\_method\_as\_payment(String arg1) throws Throwable {  System.out.println("Inside select\_payment\_method\_as\_payment");    //Adding customer details  System.out.println("FirstName");  driver.findElement(By.xpath("//\*[@id=\"billing\_first\_name\"]")).sendKeys("Ketaki");    System.out.println("LastName");  driver.findElement(By.xpath("//\*[@id=\"billing\_last\_name\"]")).sendKeys("Phadnis");    System.out.println("Selecting Country");  WebElement country =driver.findElement(By.name("billing\_country"));    Select drpCountry = new Select(country);  drpCountry.selectByVisibleText("Iceland");  Thread.sleep(3000);  System.out.println("Selected Country");    System.out.println("Entering street address");  WebElement streetAddress= driver.findElement(By.xpath("//\*[@id=\"billing\_address\_1\"]"));  streetAddress.sendKeys("1111 Central Avenue");      System.out.println("Enter Town or City");  driver.findElement(By.xpath("//\*[@id=\"billing\_city\"]")).sendKeys("Santa Clara");      System.out.println("Enter Zipcode");  driver.findElement(By.xpath("//\*[@id=\"billing\_postcode\"]")).sendKeys("580031");    System.out.println("Enter Phone");  driver.findElement(By.xpath("//\*[@id=\"billing\_phone\"]")).sendKeys("9998887771");    System.out.println("Enter email address");  driver.findElement(By.xpath("//\*[@id=\"billing\_email\"]")).sendKeys("ketaki@rumba.com");    System.out.println("Check terms and conditions");  WebElement terms = driver.findElement(By.xpath("//input[@id='terms']"));    Thread.sleep(1000);  JavascriptExecutor js =(JavascriptExecutor) driver;  js.executeScript("arguments[0].click()", terms);  System.out.println("Checked");    }  @When("^place the order$")  public void place\_the\_order() throws Throwable {  System.out.println("Inside Place order");    //Place order  System.out.println("Place order");  driver.findElement(By.xpath("//\*[@id=\"place\_order\"]")).click();    }  } |

jsonDataReader.java

|  |
| --- |
| package dataProviders;  import java.io.BufferedReader;  import java.io.FileNotFoundException;  import java.io.FileReader;  import java.io.IOException;  import java.util.Arrays;  import java.util.List;  import com.google.gson.Gson;  import managers.FileReaderManager;  import testDataTypes.Customer;  public class JsonDataReader {  //private final String customerFilePath = FileReaderManager.getInstance().getConfigReader().getTestDataResourcePath() + "Customer.json";  private final String customerFilePath = FileReaderManager.getInstance().getConfigReader().getTestDataResourcesPath() + "\\Customer.json";  private List <Customer> customerList;    public JsonDataReader() {  customerList= getCustomerData();  }    private List<Customer> getCustomerData() {  System.out.println("JsonDataReader -- executing getCustomerData");  Gson gson = new Gson();  BufferedReader bufferReader = null;    try {  System.out.println("JsonDataReader -- customerFilePath " + customerFilePath);  bufferReader = new BufferedReader(new FileReader(customerFilePath));  Customer[] customers = gson.fromJson(bufferReader, Customer[].class);  System.out.println("JsonDataReader -- Customer length = " + customers.length);  return Arrays.asList(customers);  } catch (FileNotFoundException e) {  throw new RuntimeException("Json file not found at path : " + customerFilePath);  //e.printStackTrace();  }  finally {  try { if(bufferReader != null) bufferReader.close();}  catch (IOException ignore) {}  }  }    /\*  public final Customer getCustomerByName(String customerName) {  return customerList.stream().filter(x -> x.firstName.equalsIgnoreCase(customerName)).findAny().get();  }  \*/    public final Customer getCustomerByName(String customerName){  System.out.println("JsonDataReader -- getCustomerByName =" + customerName);  for(Customer customer : customerList) {  System.err.println("JsonDataReader -- getCustomerByName -- customer = " + customer.firstName + "," + customer.lastName);  if(customer.firstName.equalsIgnoreCase(customerName)) return customer;  }  return null;  }  } |

ConfitFileReader.java

|  |
| --- |
| package dataProviders;  import java.io.BufferedReader;  import java.io.FileNotFoundException;  import java.io.FileReader;  import java.io.IOException;  import java.util.Properties;  import enums.DriverType;  import enums.EnvironmentType;  public class ConfigFileReader {    private Properties properties;  private final String propertyFilePath= "C:\\Users\\TISHAN1KA\\eclipse-workspace\\CucumberFramework\_new\\configs\\Configuation.properties";    public ConfigFileReader() {  BufferedReader reader;  //System.out.println("propertyFilePath = " + propertyFilePath);  try {  reader = new BufferedReader(new FileReader (propertyFilePath));  properties = new Properties();  try {  properties.load(reader);  reader.close();  } catch(IOException e) {  e.printStackTrace();  }  }catch(FileNotFoundException e) {  e.printStackTrace();  throw new RuntimeException("Configuration property not found at " + propertyFilePath);  }  }      public String getDriverPath() {  String driverPath= properties.getProperty("DriverPath");  if (driverPath != null )  return driverPath;  else  {  throw new RuntimeException("DriverPath not specified in the configuration property file " );  }  }    public long getImplicitWait() {  String implicitWait= properties.getProperty("ImplicitlyWait");  if (implicitWait != null )  return Long.parseLong(implicitWait) ;  else  {  throw new RuntimeException("ImplicitWait not specified in the configuration property file " );  }  }    public String getApplicationURL() {  String url= properties.getProperty("URL");  if (url != null )  return url;  else  {  throw new RuntimeException("URL not specified in the configuration property file " );  }  }    public DriverType getBrowser() {  String browserName = properties.getProperty("browser");  if (browserName==null || browserName.equals("chrome")) return DriverType.CHROME;  else if (browserName.equalsIgnoreCase("firefox"))  return DriverType.FIREFOX;  else if (browserName.equalsIgnoreCase("msedge"))  return DriverType.MSEDGE;  else throw new RuntimeException("Browser Name is not spscified in the configuration file");  }    public EnvironmentType getEnvironment() {  String environmentName = properties.getProperty("environment");  if (environmentName == null || environmentName.equals("local")) return EnvironmentType.LOCAL;  else if(environmentName.equals("remote")) return EnvironmentType.REMOTE;  else throw new RuntimeException("Environment is not spscified in the configuration file");  }    public Boolean getBrowserWindowSize() {  String windowSize = properties.getProperty("windowMaximize");  if (windowSize != null)  return Boolean.valueOf(windowSize) ;  else  return true;  }    public String getTestDataResourcesPath() {  String testDataResourcesPath = properties.getProperty("testDataResourcesPath");  System.out.println("ConfigFileReader -- testDataResourcesPath =" + testDataResourcesPath);  if (testDataResourcesPath != null) return testDataResourcesPath;  else throw new RuntimeException("TestDataResourcesPath is not specified in the configuration file");  }  public Long getImplicitlyWait() {  // TODO Auto-generated method stub  return (long) 10;  }    public String getReportConfigPath() {  String reportConfigPath = properties.getProperty("reportConfigPath");  System.out.println("getReportConfigPath = " + reportConfigPath );  if (reportConfigPath != null)  return reportConfigPath;  else throw new RuntimeException("Report Config Path not specified in the Configuration.properties file for the Key:reportConfigPath");  }  } |