```java
package com.ibm.test;

import static org.testng.Assert.assertTrue;

import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
import com.ibm.pages.AdminPage;
import com.ibm.pages.LoginPage;
import com.ibm.utilities.ExcelUtil;
import com.ibm.utilities.PropertiesFileHandler;


public class BaseTest {


	//initializing the driver, wait , profilehandler and HashMap

	WebDriver driver;
	WebDriverWait wait;
	PropertiesFileHandler propFileHandler;
	HashMap<String, String> data;


	//  beforesuit method is used to declare the statement that are going to
execute before the running of  testcase

	@BeforeSuite

	public void preSetForTest() throws IOException {

		String file = "./TestData/data.properties";
		propFileHandler = new PropertiesFileHandler();
		data = propFileHandler.getPropertiesAsMap(file);

	}

	//beforemethod  is used to declare the statement that are going to run at the
starting of  all test cases
	@BeforeMethod

	public void Initialization() {

		System.setProperty("webdriver.chrome.driver",
"./driver/chromedriver.exe");
```

```
			driver = new ChromeDriver();//  opening the chrome browser
			wait = new WebDriverWait(driver, 60);// giving explicit wait
			driver.manage().window().maximize();// maximizing the window
			driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
//declaring implicit wait

	}

	// aftermethod is used to run once all the test cases are run

	@AfterMethod
	public void closeBrowser() {
			driver.quit();//  close the browser
	}

	// Use properties to solve the simple test cases.
	@Test
	public void testcase1() throws IOException, InterruptedException {

			String url = data.get("url");
			String username = data.get("user");
			String password = data.get("password");
			String expectedmsg = data.get("requiredmsg");

			// open the url https://atozgroceries.com/admin
			driver.get(url);


			// calling login page method  to enter the user name, password and click
on log in button
			LoginPage login = new LoginPage(driver, wait);


			login.enterEmailAddress(username);
			login.enterPassword(password);
			login.clickOnLogin();

			//  calling admin page method to perform the task in admin page
			AdminPage admin = new AdminPage(driver, wait);

			admin.clickOnCatalog();
			admin.clickOnTabs();
			admin.clickOnAction();
			admin.clickOnDelete();
			Thread.sleep(5000);
			admin.clickOnDeleteConfirmbox();
			Thread.sleep(10000);

			// Assertion to verify the delete message to verify the test case is
passed .
			String actualmsg = admin.getTextToVerifyDeleteTabMsg();

			Assert.assertTrue(actualmsg.contains(expectedmsg), "Assertion on Deleted
Data");
```

```
        }
}
```

Admin Page :

```
package com.ibm.pages;

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.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class AdminPage {

        WebDriverWait wait;
        WebDriver driver;


        @FindBy(xpath="//*[@id=\"side-menu\"]/li[2]/a/span")
        WebElement catalogEle;
```

```
@FindBy(xpath="//*[@id=\"side-menu\"]/li[2]/ul/li[1]/a")
WebElement tabEle;

@FindBy(xpath="//*[@id=\"dataTableExample2\"]/tbody/tr[1]/td[5]/div/button")
WebElement actionEle;

@FindBy(xpath="//*[@id=\"dataTableExample2\"]/tbody/tr[1]/td[5]/div/ul/li[2]/a")
WebElement deleteEle;

@FindBy(xpath="//button[@class='confirm']")
WebElement delConfirmationEle;

@FindBy(xpath="//*[contains(text(),'You have successfully deleted data')]")
WebElement deleteMsgEle;



public AdminPage(WebDriver driver, WebDriverWait wait)  {
        PageFactory.initElements(driver, this);
        this.driver=  driver;
        this.wait=  wait;

}

//  method to click on the  catalog tab

public void clickOnCatalog(){
        catalogEle.click();
```

```
	}
	//  click on Tab
	public void clickOnTabs() {

		tabEle.click();
	}

	// Click on Action  on the Tab window
	public void clickOnAction() {

		actionEle.click();
	}

	// Select Delete
	public void clickOnDelete() {

		deleteEle.click();
	}

	//  confirmation to delete the Tab
	public void clickOnDeleteConfirmbox() {
		delConfirmationEle.click();
	}
	//  this string is  declared to verify the text we are getting after  deleting the tab
		public String getTextToVerifyDeleteTabMsg() {

	wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//*[contains(text(),'You have successfully deleted data')]")));
			String actualmsg = deleteMsgEle.getText();
}			return actualmsg;
```

After running the script