Test Objective:

The objective is to automate the below testcase using selenium WebDriver and framework concepts.

Testcase: TC03

Process: Edit a new push notification under Marketing

Steps to Execute:

Step1: Login to admin portal

Step2: Edit a new notification by filling only the mandatory fields.

Expected Results: Validate the presence of the modified notification

Important note: Currently admin portal is navigating to the screen ‘Edit Coupon’, when the user tries to edit the notification, which is unexpected. Hence the mentor advised to work for the below objective, as it verifies the similar functionality.

Process: Add new notification under Marketing

Steps to Execute:

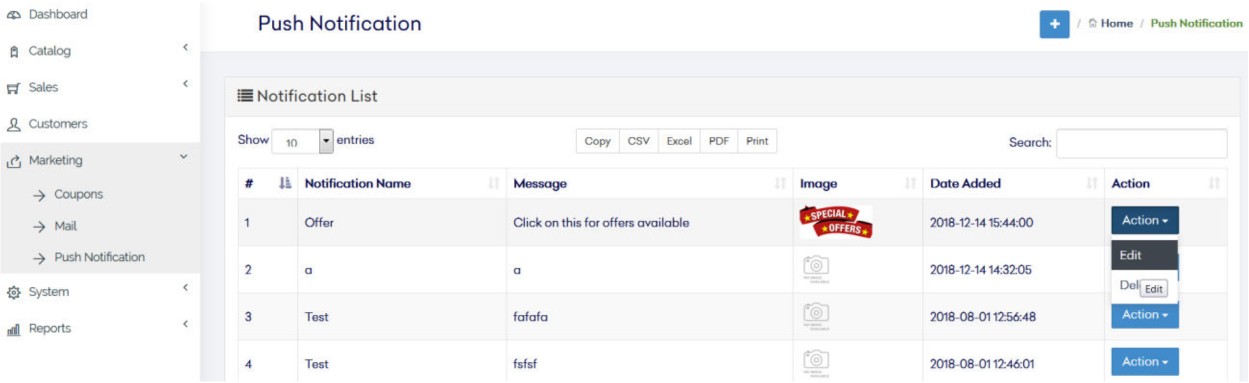
Step1: Login to admin portal

Step2: Add a new notification by filling mandatory fields.

Expected Result: Validate the presence of the added notification.

Below is for information:

User is trying to edit notification



Push Notification

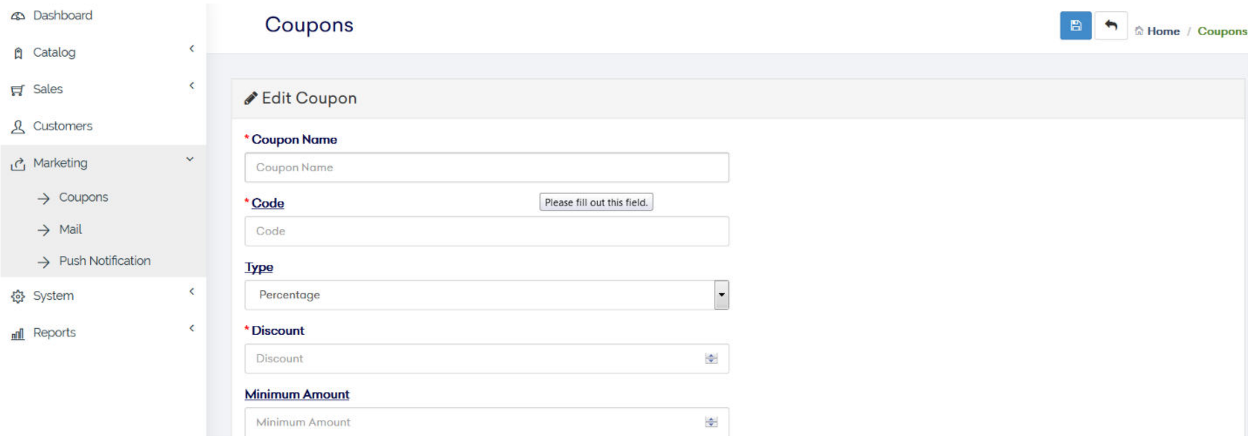
Notification List

Show 10 entries

Search:

#	Notification Name	Message	Image	Date Added	Action
1	Offer	Click on this for offers available	SPECIAL OFFERS	2018-12-14 15:44:00	Action
2	a	a		2018-12-14 14:32:05	
3	Test	fafafa		2018-08-01 12:56:48	Action
4	Test	fsfsf		2018-08-01 12:46:01	Action

But the page for edit coupons is displayed.



Source code:

File1: `AddNotiification.java`

```java
package com.ibm.groceries;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Date;
import java.util.HashMap;
import java.util.Properties;
import java.util.Set;
import java.util.concurrent.TimeUnit;
import org.apache.commons.io.FileUtils;
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 org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import com.ibm.groceriespages.PageLogin;
import com.ibm.groceriespages.AddNotificationPage;
import com.ibm.groceriespages.PageDashboard;
import com.ibm.groceriespages.PageProducts;
import com.ibm.groceriespages.PushNotificatoinPage;
import com.ibm.initialization.WebDriverLaunch;
import com.ibm.utilities.PropertiesFileHandler;
import jdk.nashorn.internal.runtime.PrototypeObject;

public class AddNotiification extends WebDriverLaunch {

//To add new push notification
        @Test(priority = 1, testName = "AddNotification", groups = "low")
        public void addPushNotificatoin() {

                //Getting the values defined in properties file
                String url = data.get("url");
                String userName = data.get("username");
                String password = data.get("password");
                String expMessage = data.get("expNotificationMessage");
                String notificaitonName = data.get("notifyName").toString();
                String notificatoinMessage = data.get("notifyMessage").toString();
                String imagePath=data.get("imagePath");

                // Launching the web site for atozgroceries
                driver.get(url);

                PageLogin login = new PageLogin(driver, wait);
                // To enter email address and password and click on login button
                login.enterEmailAddress(userName);
                login.enterPassword(password);
                login.clickOnLogin();

        Assert.assertTrue(driver.findElement(By.partialLinkText("Logout")).isDisplayed
());

                PageDashboard dashboard = new PageDashboard(driver, wait);
                dashboard.clickOnMarketing();
                dashboard.clickOnPushNotification();

                PushNotificatoinPage notification = new PushNotificatoinPage(driver,
wait);
                // calling method to click on add notification button
                notification.clickonAddNotification();

                AddNotificationPage add = new AddNotificationPage(driver, wait);
                String pageSource = add.addNotification(notificaitonName,
notificatoinMessage,imagePath);

                // Verifying whether the expected message displayed after adding
notification

                if (pageSource.contains(expMessage))
                        System.out.println(expMessage);
        Assert.assertTrue(pageSource.contains(expMessage), "Assertion on Adding 
Notificaiton");

        //Verifying the notification name and notification message added
        if (pageSource.contains(notificaitonName))
            System.out.println(notificaitonName);
        Assert.assertTrue(pageSource.contains(notificaitonName), "Assertion on 
added notification name");

        if (pageSource.contains(notificaitonName))
            System.out.println(notificatoinMessage);
        Assert.assertTrue(pageSource.contains(notificatoinMessage), "Assertion 
on added notification message");

    }

}
```

File2: PageLogin.java

```
package com.ibm.groceriespages;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

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 PageLogin {


    @FindBy(name="email")

    WebElement emailEle;


    @FindBy(name="pword")

    WebElement passEle;


    //@FindBy(className="btn btn-labeled btn-info m-b-5")
```

```
@FindBy(xpath="//button[@class='btn btn-labeled btn-info m-b-5']")
WebElement loginEle;

WebDriverWait wait;
WebDriver driver;

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

//To enter email address
public void enterEmailAddress(String userName)
{
	emailEle.sendKeys(userName);
}

//To enter apssword
public void enterPassword(String password)
{
	passEle.sendKeys(password);
}

//To click on Login button
public void clickOnLogin()
{
	loginEle.click();
	}

}
```

File3: PageDashboard.java

```
package com.ibm.groceriespages;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
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 PageDashboard {

	//@FindBy(xpath="//a[@class='material-ripple']")
	@FindBy(xpath="//a[text()=' Catalog']")
	WebElement catalogEle;

	@FindBy(xpath="//a[text()=' Products']")
	WebElement productEle;

	By prodElt=By.xpath("//a[text()=' Products']");
	WebDriverWait wait;
	WebDriver driver;

	//Locating elements for Marketing link
```

```
@FindBy(xpath="//a[text()='  Marketing']")
WebElement marketEle;

//Locating Pushnotification link
@FindBy(xpath="//a[text()=' Push Notification']")
WebElement notificatonEle;

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

//To click on Catalog
public void clickOnCatalog()
{
        catalogEle.click();
        wait.until(ExpectedConditions.presenceOfElementLocated(prodElt));

        }
//To click on Products
public void clickOnProducts()
        {
        productEle.click();

}


//To click on Marketing link
```

```
        public void clickOnMarketing()

        {


                marketEle.click();


        }


        //To click on Push Notificatoin link

        public void clickOnPushNotification()

        {

                notificatonEle.click();


        }


}
```

File4: PushNotificatoinPage.java

```
package com.ibm.groceriespages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class PushNotificatoinPage {

        // Locating element for Add new link
        @FindBy(xpath = "//a[@title='Add New']")
        WebElement addnewEle;

        WebDriverWait wait;
        WebDriver driver;

        public PushNotificatoinPage(WebDriver driver, WebDriverWait wait) {
                PageFactory.initElements(driver, this);
                this.driver = driver;
                this.wait = wait;
        }
        // TO click on Add new notification link
        public void clickonAddNotification() {

                addnewEle.click();
        }

}
```

File5: AddNotificationPage.java

```
package com.ibm.groceriespages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class AddNotificationPage {

        // Locating element for Notification name field
        @FindBy(xpath = "//input[@name='name']")
        WebElement nameEle;

        // Locating element for Message field
        @FindBy(xpath = "//textarea[@name='message']")
        WebElement messageEle;

        // Locating element for image
        @FindBy(xpath = "//input[@id='files']")
        WebElement imageEle;

        // Locating element for Save button
        @FindBy(xpath = "//button[@title='Save']")
        WebElement saveEle;

        WebDriverWait wait;
        WebDriver driver;

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

        //To add push notification
        public String addNotification(String notification, String message,String 
imagepath) {
                nameEle.sendKeys(notification);
                messageEle.sendKeys(message);
                //imageEle.sendKeys("C:\\Users\\IBM_ADMIN\\eclipse-
workspace\\TestCases_DataDrivenFramework\\TestData\\globe.jpg");
                imageEle.sendKeys(imagepath);
```

```java
                saveEle.click();
                return driver.getPageSource();
        }

}
```

File6: WebDriverLaunch.java

```java
package com.ibm.initialization;

import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import com.ibm.utilities.PropertiesFileHandler;

public class WebDriverLaunch {
        public WebDriver driver;
        public WebDriverWait wait;
        public PropertiesFileHandler propFileHandler;
        public HashMap<String, String> data;

        //Getting the keys from properties file
        //@BeforeSuite
        @BeforeSuite(groups= {"high","low"})
                public void preSetForTest() throws IOException {
                String file = "./TestData/groceries.properties";
                propFileHandler = new PropertiesFileHandler();
                data = propFileHandler.getPropertiesAsMap(file);
        }


        //@BeforeTest
        @BeforeMethod(groups= {"high","low"})
        @Parameters({"browser"})
        public void Initialization(@Optional("ff")String browser) {
                browserInitialization(browser);
                wait = new WebDriverWait(driver, 60);
                driver.manage().window().maximize();
                driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        }

        //Closing driver
        //@AfterMethod
```

```
    @AfterMethod(groups= {"high","low"})
    public void closeBrowser() {
        driver.quit();
    }

    //Setting path for webdriver
    public void browserInitialization(String browser)
    {
        switch (browser.toLowerCase()) {
        case "ff":
        case "firefox":
            System.setProperty("webdriver.gecko.driver",
"./drivers/geckodriver.exe");
            driver = new FirefoxDriver();
            break;
        case "ch":
            System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
            driver = new ChromeDriver();
            break;
        case "ie":
            System.setProperty("webdriver.ie.driver",
"./drivers/IEDriverServer.exe");
            driver = new InternetExplorerDriver();
            break;
        default:
            System.out.println("No browser Available "+browser);
            break;
        }
    }
}
```

File7: PropertiesFileHandler.java

```
package com.ibm.utilities;


import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.HashMap;

import java.util.Properties;

import java.util.Set;


public class PropertiesFileHandler {
    }

    public HashMap<String, String> getPropertiesAsMap(String file) throws IOException {
        HashMap<String, String> magentoMap = new HashMap<String, String>();

        FileInputStream fileIn = new FileInputStream(file);
        Properties prop = new Properties();
        prop.load(fileIn);

        Set<Object> keysProp = prop.keySet();
        for (Object key : keysProp) {
            magentoMap.put(key.toString(), prop.getProperty(key.toString()));
        }
        prop.clear();
        return magentoMap;
    }

    public void setKeyAndValue(String file,String key,String value) throws IOException
    {
        FileInputStream fileIn = new FileInputStream(file);
        Properties prop = new Properties();
        prop.load(fileIn);

        prop.setProperty(key, value);

        FileOutputStream fOut=new FileOutputStream(file);
        prop.store(fOut, "Test Result");
        fOut.close();
        fileIn.close();
    }
}
```

File8:testing_notificaiton.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
<groups>
             <run>
                    <exclude name="medium"></exclude>
             </run>
       </groups>
  <test thread-count="5" name="AddingNotification">
  <parameter name="browser" value="firefox"></parameter>
    <classes>
      <class name="com.ibm.groceries.AddNotiification"/>
    </classes>
  </test> <!-- AddingNotification -->
</suite> <!-- Suite -->
```

File9: groceries.properties

url=https://atozgroceries.com/admin

username=demo@atozgroceries.com

password=456789

expectedMessage=Success: You have successfully deleted data!

notifyName=DemoNotification

notifyMessage=Adding Notification

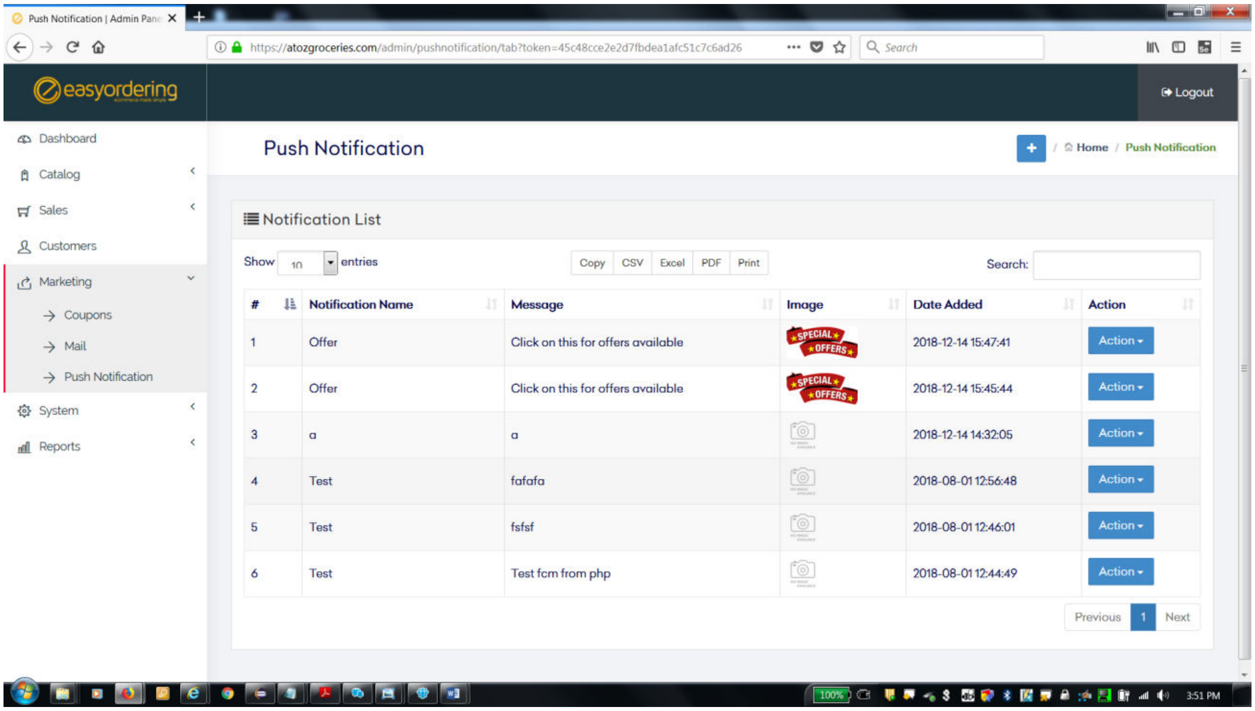
expNotificationMessage=Success: You have successfully sent push notification to your app!

imagePath=C:\\Users\\IBM_ADMIN\\eclipse-workspace\\TestCases_DataDrivenFramework\\TestData\\globe.jpg

Test Result: A new push notification is successfully added and assertion is done on presence of added notification**.**

Screenshots:

Existing notifications:



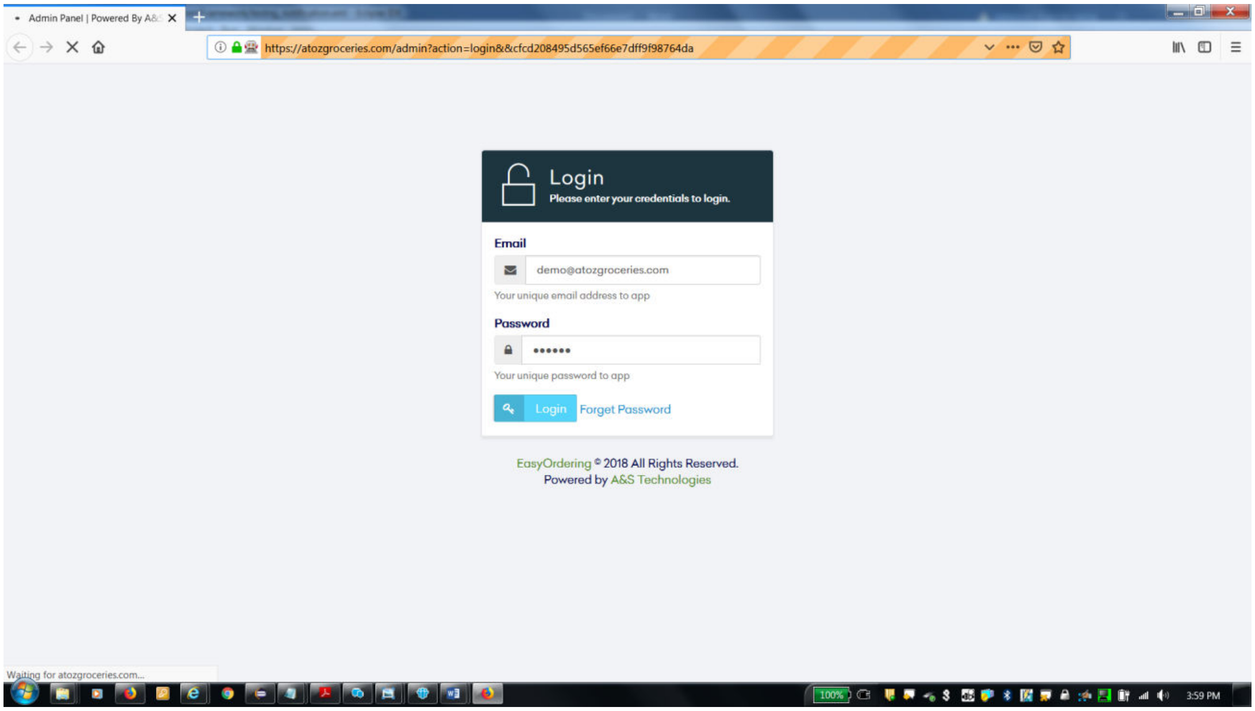
Push Notification

Notification List

Show 10 entries

#	Notification Name	Message	Image	Date Added	Action
1	Offer	Click on this for offers available	SPECIAL OFFERS	2018-12-14 15:47:41	Action
2	Offer	Click on this for offers available	SPECIAL OFFERS	2018-12-14 15:45:44	Action
3	a	a		2018-12-14 14:32:05	Action
4	Test	fafafa		2018-08-01 12:56:48	Action
5	Test	fsfsf		2018-08-01 12:46:01	Action
6	Test	Test fcm from php		2018-08-01 12:44:49	Action

Login to admin portal



https://atozgroceries.com/admin?action=login&&cfcd208495d565ef66e7dff9f98764da

Login

Please enter your credentials to login.

Email

demo@atozgroceries.com

Your unique email address to app

Password

••••••

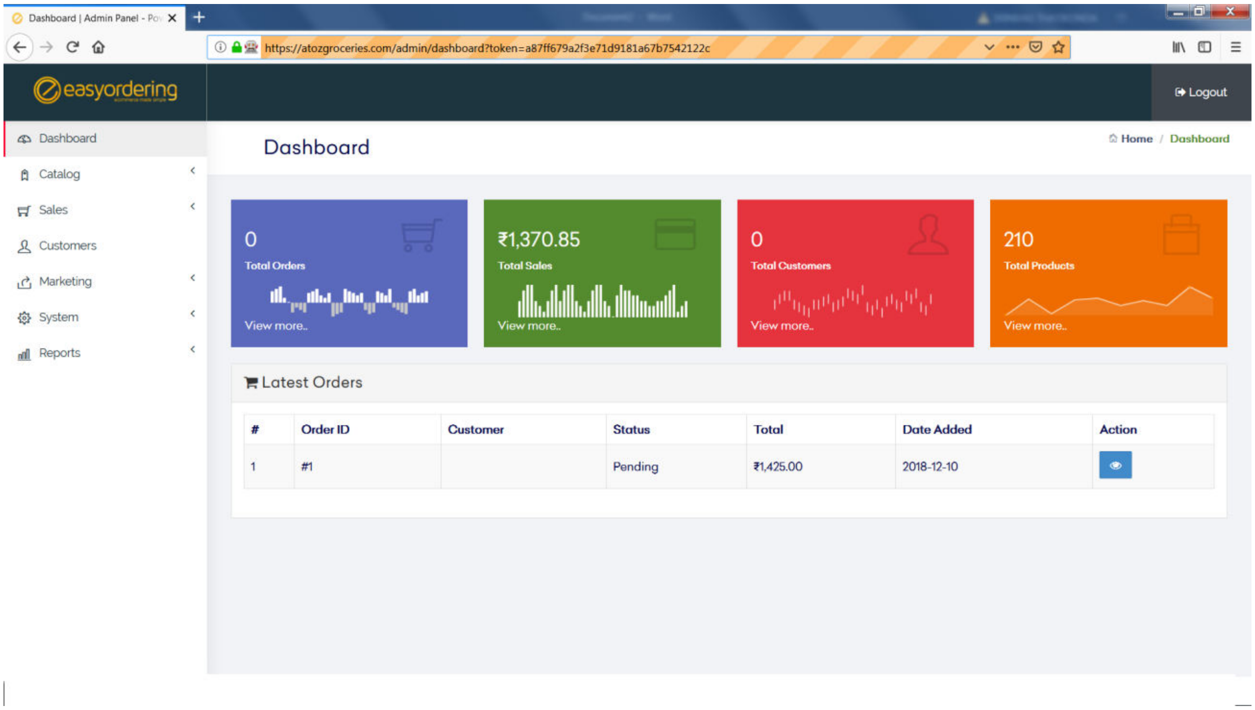
Your unique password to app

Login Forget Password

EasyOrdering © 2018 All Rights Reserved.
Powered by A&S Technologies

Waiting for atozgroceries.com...

Dashboard page is displayed



Dashboard | Admin Panel - Po
https://atozgroceries.com/admin/dashboard?token=a87ff679a2f3e71d9181a67b7542122c
easyordering
Logout
Dashboard
Catalog
Sales
Customers
Marketing
System
Reports
Home / Dashboard

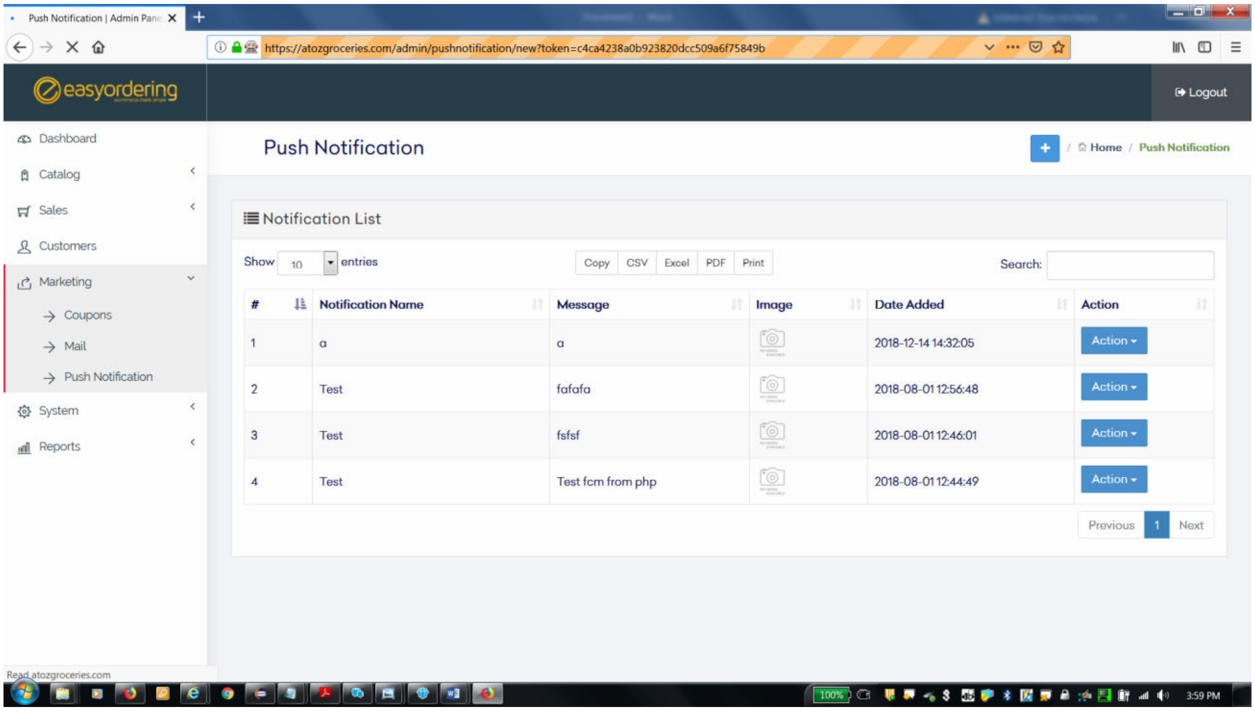
Dashboard

0 Total Orders View more..	₹1,370.85 Total Sales View more..	0 Total Customers View more..	210 Total Products View more..

Latest Orders

#	Order ID	Customer	Status	Total	Date Added	Action
1	#1		Pending	₹1,425.00	2018-12-10	

Push notification page is displayed after clicking Marketing->Push Notification



Push Notification

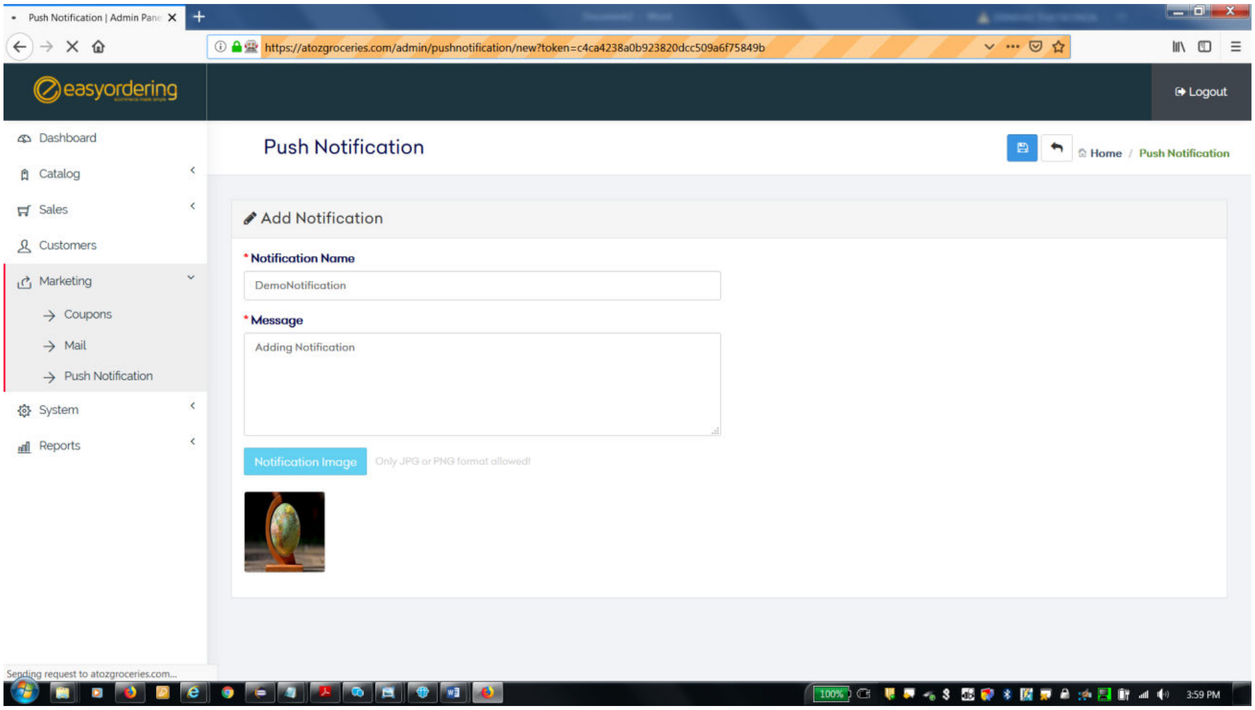
Notification List

Show 10 entries · Copy · CSV · Excel · PDF · Print · Search:

#	Notification Name	Message	Image	Date Added	Action
1	a	a		2018-12-14 14:32:05	Action
2	Test	fafafa		2018-08-01 12:56:48	Action
3	Test	fsfsf		2018-08-01 12:46:01	Action
4	Test	Test fcm from php		2018-08-01 12:44:49	Action

Previous 1 Next

After clicking Add Notification link, the below page is displayed and mandatory fields are valued.


Push Notification | Admin Panel
https://atozgroceries.com/admin/pushnotification/new?token=c4ca4238a0b923820dcc509a6f75849b
easyordering
Logout
Dashboard
Catalog
Sales
Customers
Marketing
Coupons
Mail
Push Notification
System
Reports
Home / Push Notification
Sending request to atozgroceries.com...
3:59 PM

Push Notification

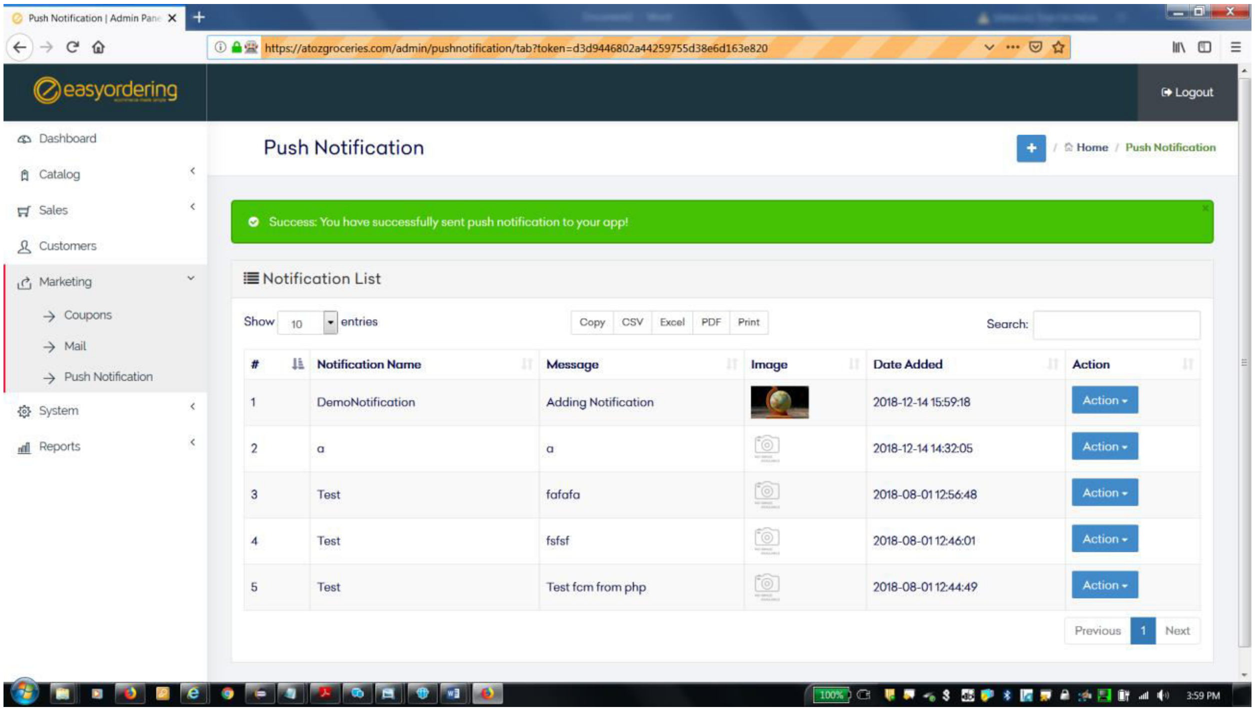
Add Notification

* Notification Name: DemoNotification

* Message: Adding Notification

Notification Image — Only JPG or PNG format allowed!

The confirmation message is displayed after adding notification and added field values are displayed on page.



Push Notification

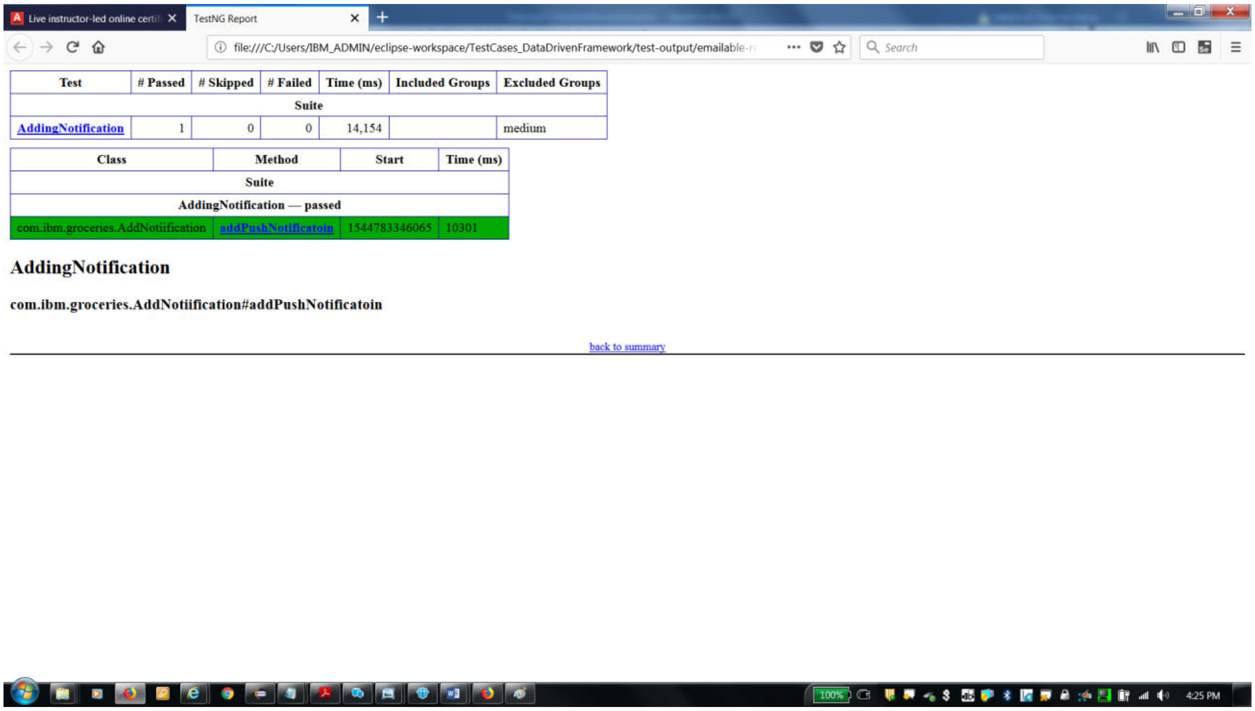
Success: You have successfully sent push notification to your app!

Notification List

Show 10 entries Copy CSV Excel PDF Print Search:

#	Notification Name	Message	Image	Date Added	Action
1	DemoNotification	Adding Notification		2018-12-14 15:59:18	Action
2	a	a		2018-12-14 14:32:05	Action
3	Test	fafafa		2018-08-01 12:56:48	Action
4	Test	fsfsf		2018-08-01 12:46:01	Action
5	Test	Test fcm from php		2018-08-01 12:44:49	Action

Previous 1 Next



Live instructor-led online certi
TestNG Report
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/emailable-r
Search

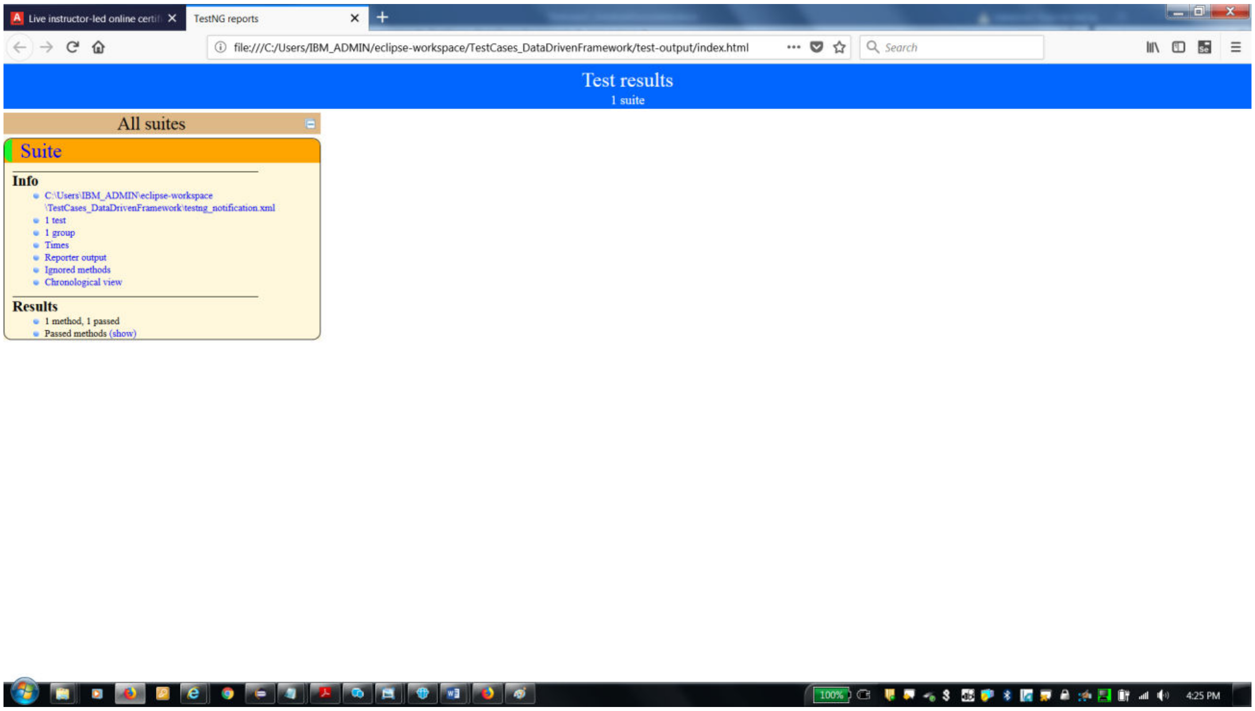
Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
AddingNotification	1	0	0	14,154		medium

Class	Method	Start	Time (ms)
Suite			
AddingNotification — passed			
com.ibm.groceries.AddNotiification	addPushNotificatoin	1544783346065	10301

AddingNotification

com.ibm.groceries.AddNotiification#addPushNotificatoin

back to summary



Live instructor-led online certi
TestNG reports
file:///C:/Users/IBM_ADMIN/eclipse-workspace/TestCases_DataDrivenFramework/test-output/index.html
Search
4:25 PM

Test results

1 suite

All suites

Suite

Info

- C:\Users\IBM_ADMIN\eclipse-workspace \TestCases_DataDrivenFramework\testng_notification.xml
- 1 test
- 1 group
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 1 method, 1 passed
- Passed methods (show)

Flow Information:

Initially a property file ‘groceries.properties’ is created in Test Data folder of the project and list of key, values for url, username,password ,expected validation message,notification name,notficaiton message,path for image file etc. are stored in this property file.

Under the class ‘WebDriverLaunch’ ,a Test NG annotation ‘Before Suite’ is used to invoke the method ‘preSetFortTest’ which is used to instantiate object for class PropertiesFileHandler to return the list of these key,values in to a HashMap.

The Test NG annotation ‘Before Method’ is used by passing the parameter for browser to select and to invoke the method ‘Initialization’ which is used to call another method and to create webdriver object for the firefoxdriver or chromedriver or internetexplorer by using switch statement and setting path for webdriver exe.

The PageLogin class is defined with methods to locate the web elements email,password and login buttons. PageFactory method is used in the in the constructor of this class. The method sendKeys is used to enter the email address and password. The method click is used to click on login button

The class ‘PageDashboard’ is defined with methods to locate the web elements marketing, Pushnotification. PageFactory method is used in the in the constructor of this class.The method click is used to click on Marketing and PushNotification links.

The class ‘PushNotificationPage’ is defined with method to locate ‘Add New’ link the using ‘Find By’and click on it. PageFactory method is used in the in the constructor of this class.

The class ‘AddNotificationPage’ locates notification name,notification message,image file, and save button using annotation FindBy.The method addNotification is used with sendKeys to enter the values for notification name, notification message and to add image file and then clicking on save button using click method. The method returns the pagesource.

The annotation @Test is used with method addPushNotification in the class ‘AddNotification. This method is used to get the values for username,password ,notification name,message,image path and the expected message to validate. The get method is called to launch the web url for admin portal. The objects for classes PageLogin,PageDashboard,PushNotificationPage,AddNotificationPage are created and the above necessary methods are called on these objects. Finally the Assert is used to verify the expected message for successful addition of notification and presence of added notification name and message.The testng_notification xml file is having the parameter ‘firefox’ to launch chrome browser.