

Test Objective:

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

Testcase: Tc10**Process:** Add Email**Steps to Execute:**

Step1: Check for count of email table records in DB

Step2: Launch Admin portal and navigate on to email under Marketing tab

Step3: Add a new email record.

Expected Results:

Validate the presence newly added record and the increase in the table record count by 1

Sourcecode:

File1: EmailCount.java

```
package com.ibm.groceries;

import java.io.IOException;
import java.sql.SQLException;
import org.openqa.selenium.By;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;
import com.ibm.groceriespages.MailListPage;
import com.ibm.groceriespages.PageDashboard;
import com.ibm.groceriespages.PageLogin;
import com.ibm.groceriespages.SendMailPage;
import com.ibm.initialization.WebDriverLaunch;
import com.ibm.utilities.DatabaseConnection;
import com.ibm.utilities.GetScreenshot;

public class EmailCount extends WebDriverLaunch {

    @Test(priority = 1, testName = "VerifyEmailCount", groups = "low")

    public void verifyEmailCountDbase() throws IOException, InterruptedException,
SQLException {
        String url = data.get("url");
        String userName = data.get("username");
        String password = data.get("password");
        String email = data.get("customerEmail");
        String subject = data.get("subject");
        String message = data.get("message");
        String expMessage = data.get("emailMessage");
        String query = "select count(mail_id) from as_mail";
        String beforeMsg = data.get("beforeMsg");
        String afterMsg = data.get("afterMsg");
```

```

DatabaseConnection dbaseutil = new DatabaseConnection();
// Verifying the email count from database before adding email
int emailCountBefore = dbaseutil.countRecords(query);
System.out.println(beforemsg + emailCountBefore);
// Launching the web site for atozgroceries
driver.get(url);
GetScreenshot screen = new GetScreenshot();

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

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

PageDashboard dashboard = new PageDashboard(driver, wait);
// To click on Catalog
dashboard.clickOnMarketing();

// To click on Mail link
dashboard.clickOnMail();
screen.takeScreenshot(driver);

MailListPage maillistObj = new MailListPage(driver, wait);
// Calling method to click on Add email button
maillistObj.clickOnAdd();
screen.takeScreenshot(driver);
// Calling method to add email
SendMailPage addemailObj = new SendMailPage(driver, wait);
String pageSource = addemailObj.addMail(email, subject, message);

screen.takeScreenshot(driver);
if (pageSource.contains(expMessage)) {
    Reporter.Log(expMessage);

    // checking whether success message is displayed or not
    Assert.assertTrue(pageSource.contains(expMessage));

    // Checking whether added email record is displayed or not
    Assert.assertTrue(pageSource.contains(subject));

    // Verifying the email count from database after adding email
    int emailCountAfter = dbaseutil.countRecords(query);
    System.out.println(afterMsg + emailCountAfter);
    Assert.assertEquals(emailCountAfter, emailCountBefore + 1);
}
}
}

```

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.FindingBy;
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;
```

```
    // Locating System link
```

```
    @FindBy(xpath = "//a[text()=' System']")
```

```
    WebElement systemEle;
```

```
// Locating Returns link
```

```
@FindBy(xpath = "//a[text()=' Returns']")
```

```
// @FindBy(xpath("//a[contains(text(),'Returns')]"))
```

```
WebElement returnsEle;
```

```
// Locating Shipping link
```

```
@FindBy(xpath = "//a[text()=' Shipping']")
```

```
WebElement shippingEle;
```

```
// Locating ShippingLocations link
```

```
@FindBy(xpath = "//a[text()=' Shipping Locations']")
```

```
WebElement locatoinEle;
```

```
@FindBy(xpath = "//a[text()=' Return Actions']")
```

```
WebElement actionsEle;
```

```
@FindBy(xpath = "//a[text()=' Settings']")
```

```
WebElement settingsEle;
```

```
@FindBy(xpath = "//a[text()=' Mail']")
```

```
WebElement mailEle;
```

```
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();

}
```

```
// TO click on System link
public void clickOnsystem() {
    systemEle.click();
}
```

```
}
```

```
// To click on Returns link
```

```
public void clickOnReturns() {  
    driver.findElement(By.partialLinkText("Returns")).click();  
    // returnsEle.click();  
}
```

```
// To click on Return actions link
```

```
public void clickOnReturnActions()  
  
{  
    actionsEle.click();  
}
```

```
public void clickOnShipping() {  
    driver.findElement(By.partialLinkText("Shipping")).click();  
    // shippingEle.click();  
}
```

```
public void clickOnShippingLocations() {  
    locationEle.click();  
}
```

```
// To click on Settings link
```

```
public void clickOnSettings() {
```



```

        settingsEle.click();
    }

    public void clickOnMail() {
        mailEle.click();
    }
}

```

File4: MaillistPage.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 MaillistPage {

    @FindBy(xpath = "//a[@title='Add New']")
    WebElement addnewEle;

    WebDriverWait wait;
    WebDriver driver;

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

    // Method to add email
    public void clickOnAdd() {
        addnewEle.click();
    }
}

```

File5: SendMailPage.java

```

package com.ibm.groceriespages;

import org.openqa.selenium.JavascriptExecutor;

```

```

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.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class SendMailPage {

    // To locate To dropdown field
    @FindBy(xpath = "//select[@name='to']")
    WebElement toEle;

    // To locate Customer E-mail text box
    @FindBy(xpath = "//input[@id='email']")
    WebElement emailEle;

    // To locate Subject text box
    @FindBy(xpath = "//input[@name='subject']")
    WebElement subjectEle;

    /*
     * To locate message web element
     *
     * @FindBy(xpath="//div[@class='note-editable panel-body']") WebElement
     * messageEle;
     */

    // To locate Save button
    @FindBy(xpath = "//button[@title='Save']")
    WebElement saveEle;

    WebDriverWait wait;
    WebDriver driver;

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

    // Method to add email
    public String addMail(String customerEmail, String subject, String message)
    throws InterruptedException {

        JavascriptExecutor js = (JavascriptExecutor) driver;

        // To select the value for dropdown 'To'
        Select toSelect = new Select(toEle);
        toSelect.selectByIndex(2);

        // To enter customer email
        emailEle.sendKeys(customerEmail);
        // To enter subject
        subjectEle.sendKeys(subject);
    }
}

```

```

        // To enter message
        js.executeScript("document.getElementById('summernote').value='Welcome
All'");
        // To click on Save button
        saveEle.click();
        Thread.sleep(1000);
        return driver.getPageSource();
    }
}

```

File6: DatabaseConnection.java

```
package com.ibm.utilities;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import java.sql.Statement;
```

```
public class DatabaseConnection {
```

```
    public Statement connectDatabase() throws SQLException {
```

```
        Connection c =
```

```
        DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries",
```

```
        "foodsonfinger_atoz", "welcome@123");
```

```
        Statement stmt = c.createStatement();
```

```
        return stmt;
```

```
    }
```

```
    public int countRecords(String query) throws SQLException {
```

```
        int count = 0;
```

```

        Connection c =
DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgroceries",
        "foodsonfinger_atoz", "welcome@123");

        Statement st = c.createStatement();

        ResultSet rs = st.executeQuery(query);

        if (rs.next()) {
            count = rs.getInt(1);
        }

        return count;
    }

}

```

File7: GetScreenshot.java

```
package com.ibm.utilities;
```

```
import java.io.File;
```

```
import java.io.IOException;
```

```
import java.util.Date;
```

```
import org.apache.commons.io.FileUtils;
```

```
import org.openqa.selenium.OutputType;
```

```
import org.openqa.selenium.TakesScreenshot;
```

```
import org.openqa.selenium.WebDriver;
```

```
public class GetScreenshot {
```

```
    public void takeScreenshot(WebDriver driver)throws IOException, InterruptedException
```

```
{
```

```

        Thread.sleep(2000);

        TakesScreenshot ts=(TakesScreenshot)driver;

        File file=ts.getScreenshotAs(OutputType.FILE);

        Date date=new Date();

        String currentDate=date.toString().replaceAll(":", "-");

        FileUtils.copyFile(file,new File("./screenshots/Error_"+currentDate+".png"));

    }

}

```

File8: 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();
}

```

```

}

```

File9:WebDriverLaunch.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);
```

```
}
```

```
//@AfterMethod
```

```
//Closing driver
```

```
@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;

    }

}

}

```

File10: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
searchForKeyword=refund
searchDispalyMessage=Search with keyword is successful
noMatchDisplayMessage=No matching records found
searchForShipKeyword=Discount
prodNameNew=ProductNew100
modelNameNew=Model-New
expectedEditProductMessage=Success: You have successfully updated product!
productNotUpdated=Product update failed
userPageUrl=https://atozgroceries.com
userPageMsg=New Product found on User page
addressNew=Koramangala,Bengaluru
emailNew=info@atozgroceries.com
phoneNew=9797979797
expectedEditSettingMessage=successfully updated Store!
addressFoundMsg=new Address found on user page.
emailFoundMsg=new Email found on user page.
phoneFoundMsg=new Phone number found on user page.
fullName=srinivast

```

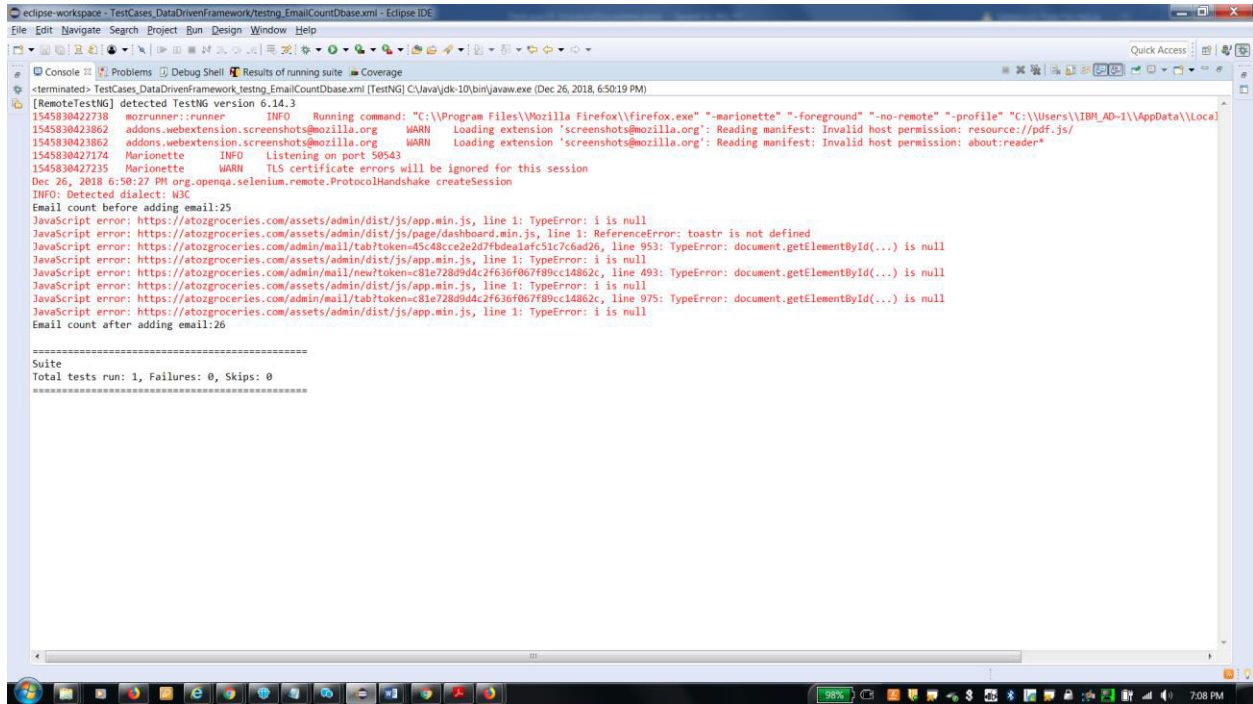
phoneNum=9701934935
passwd=dasara
confirmPassword=dasara
quantity=30
expectedShoppingCartMsg=You have successfully updated cart items!
titleNameNew=Title-New
tagDescriptionNew=Tag-New
keywordNew=keyword-New
hsnNew=Hsn-New
tableProducts=as_products
productName=Badam-p01
modifiedProductConsole=Modified details of product from DataBase:
customerEmail=srinivas123@gmail.com
subject=Adding email
message=Welcome to atoz groceries
emailMessage=Success: You have successfully sent mail to all customer!
beforeMsg=Email count before adding email:
afterMsg=Email count after adding email:

File11:testng_EmailCountDbase.xml

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

Screenshots:

Console output:

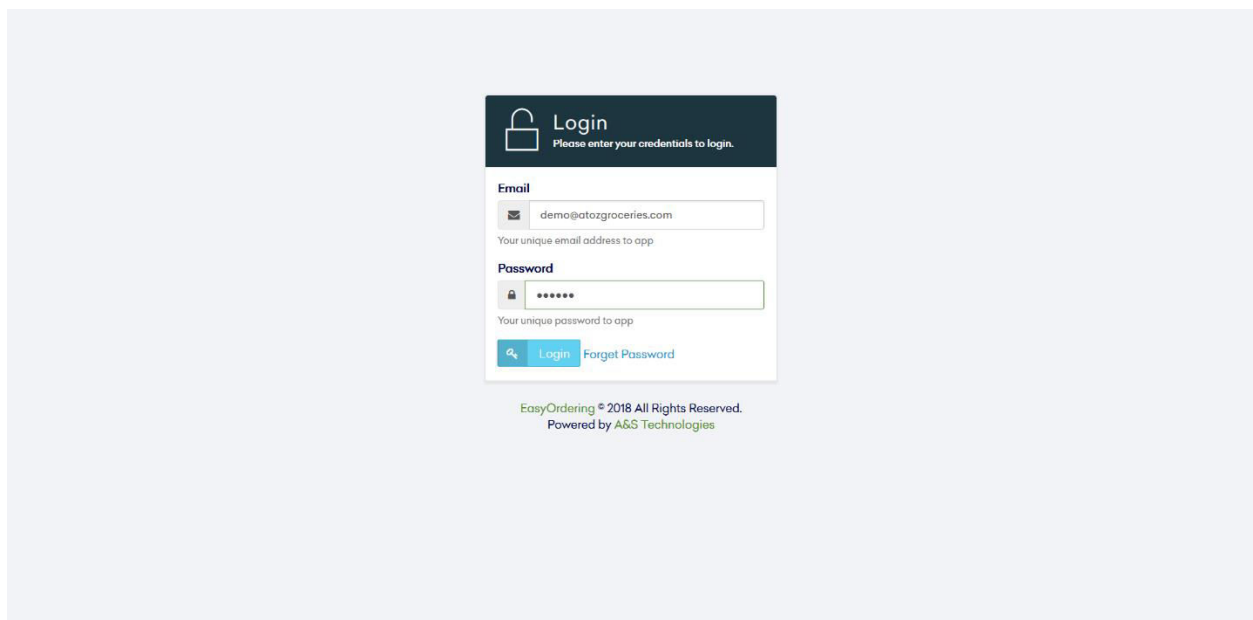


```
eclipse-workspace - TestCases_DataDrivenFramework/testing_EmailCountDbase.xml - Eclipse IDE
File Edit Navigate Search Project Run Design Window Help

[RemoteTestNG] detected TestNG version 6.14.3
1545830422738 mozrunner:runner INFO Running command: "C:\Program Files\Mozilla Firefox\Firefox.exe" "-marionette" "-foreground" "-no-remote" "-profile" "C:\Users\IBM_AD-1\AppData\Local
15458304223862 addons.webextension.screenshots.mozilla.org WARN Loading extension 'screenshots.mozilla.org': Reading manifest: Invalid host permission: resource://pdf.js/
15458304227174 Marionette INFO Listening on port 50943
15458304227235 Marionette WARN TLS certificate errors will be ignored for this session
Dec 26, 2018 6:50:27 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Email count before adding email:25
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/page/dashboard.min.js, line 1: ReferenceError: toastr is not defined
JavaScript error: https://atozgroceries.com/admin/mail/tab?token=45c48cce2e2d7fbdeda1afc51c7c6ad26, line 953: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/admin/mail/new?token=c81e728d9d4c2f636f067f89cc14862c, line 493: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
JavaScript error: https://atozgroceries.com/admin/mail/tab?token=c81e728d9d4c2f636f067f89cc14862c, line 975: TypeError: document.getElementById(...) is null
JavaScript error: https://atozgroceries.com/assets/admin/dist/js/app.min.js, line 1: TypeError: i is null
Email count after adding email:26

=====
Suite
Total tests run: 1, Failures: 0, Skips: 0
=====
```

Login page is displayed for admin portal



Mail List page is displayed.

The screenshot shows the 'Mail List' page in the 'easyordering' admin panel. The left sidebar contains navigation links: Dashboard, Catalog, Sales, Customers, Marketing (highlighted with a red border), System, and Reports. Under 'Marketing', there are links for Coupons, Mail, and Push Notification. The main content area is titled 'Mail' and features a 'Mail List' table. Above the table, there are options to 'Show 10 entries' and buttons for 'Copy', 'CSV', 'Excel', 'PDF', and 'Print'. A search bar is also present. The table has columns for '#', 'Mail Type', 'Subject', 'Date Added', and 'Action'. It lists 8 entries, all with 'All Customers' as the mail type. The subjects are 'demo', 'new', 'Test', 'testsubject', 'test', 'Welcome', 'hsdjh djdjh dhsj', and 'xxxxxddd'. Each entry has an 'Action' button.

#	Mail Type	Subject	Date Added	Action
1	All Customers	demo	2018-12-26 17:48:50	Action
2	All Customers	new	2018-12-26 17:34:16	Action
3	All Customers	Test	2018-12-26 15:12:05	Action
4	All Customers	testsubject	2018-12-26 12:53:16	Action
5	All Customers	test	2018-12-26 12:21:01	Action
6	All Customers	Welcome	2018-12-26 09:33:50	Action
7	All Customers	hsdjh djdjh dhsj	2018-12-19 17:14:17	Action
8	xxxxxddd	xxx	2018-12-18 13:02:20	Action

Send Mail page is displayed with entered details

The screenshot shows the 'Send Mail' page in the 'easyordering' admin panel. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Send Mail' and contains a form with the following fields: 'From' (set to 'Default'), 'To' (set to 'All Customers'), and a text input field containing 'srinivas123@gmail.com'. The 'Subject' field is labeled with an asterisk and contains the text 'Adding email'. The 'Message' field is labeled with an asterisk and contains the text 'Welcome All'. Below the message field is a rich text editor toolbar with various formatting options.

Success message is displayed after adding email

The screenshot shows the 'Mail' management page in the EasyOrdering system. A green success message banner at the top states: 'Success: You have successfully sent mail to all customer!'. Below this is a 'Mail List' table with 7 entries. The table columns are: #, Mail Type, Subject, Date Added, and Action. All mail types are 'All Customers'. The subjects are 'Adding email', 'demo', 'new', 'Test', 'testsubject', 'test', and 'Welcome'. Each entry has an 'Action' button. The left sidebar shows the navigation menu with 'Marketing' expanded, showing 'Coupons', 'Mail', and 'Push Notification'. The top right has a 'Logout' button.

#	Mail Type	Subject	Date Added	Action
1	All Customers	Adding email	2018-12-26 18:50:49	Action -
2	All Customers	demo	2018-12-26 17:48:50	Action -
3	All Customers	new	2018-12-26 17:34:16	Action -
4	All Customers	Test	2018-12-26 15:12:05	Action -
5	All Customers	testsubject	2018-12-26 12:53:16	Action -
6	All Customers	test	2018-12-26 12:21:01	Action -
7	All Customers	Welcome	2018-12-26 09:33:50	Action -

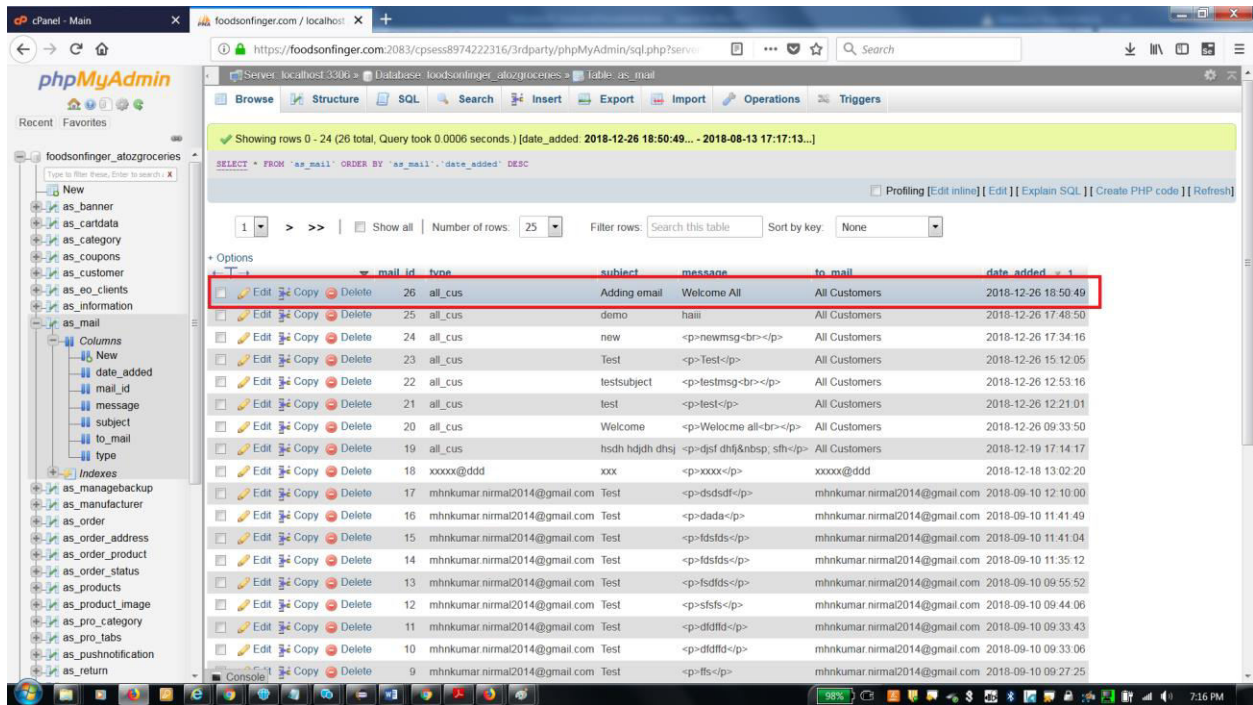
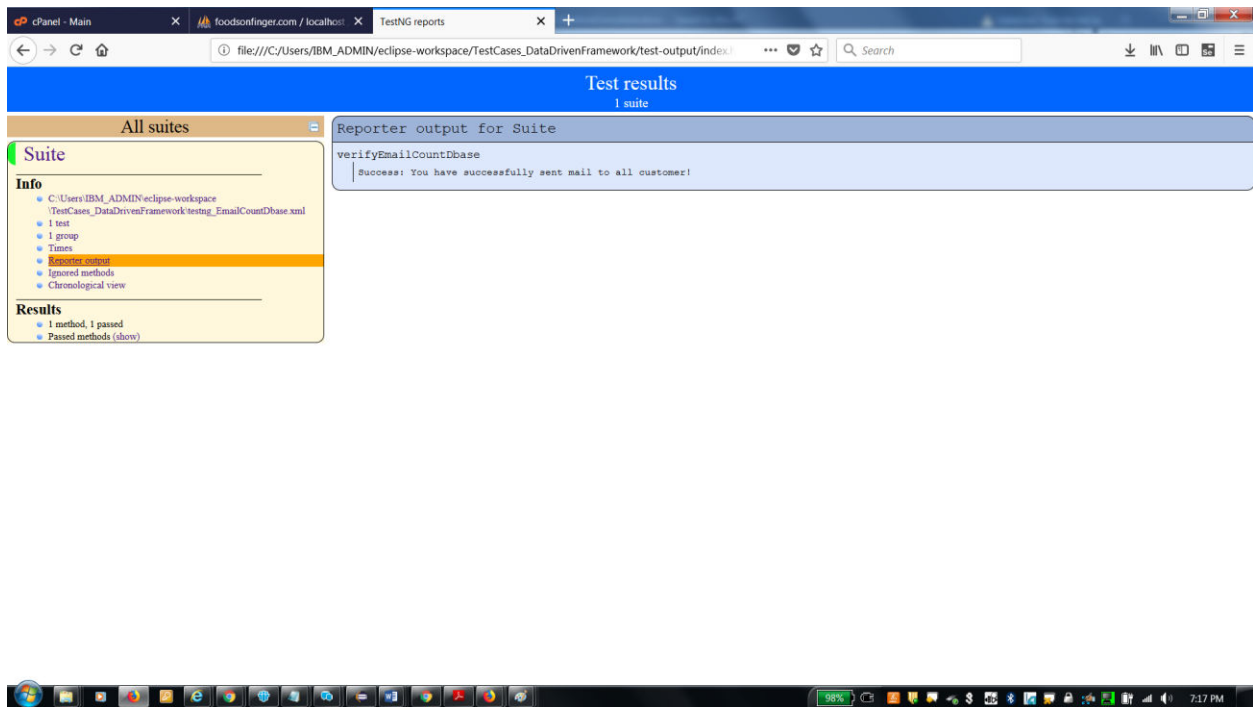
The screenshot shows a TestNG report in a web browser. The report is titled 'EmailCount' and shows a single test case that passed. The report includes a table with the following data:

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
com.ibm.groceries.EmailCount	1	0	0	29,299		

Below this table is a 'Messages' section with the following text:

Success: You have successfully sent mail to all customer!

At the bottom of the report, there is a link to 'back to summary'.



Flow Information:

Initially a property file 'groceries.properties' is created in Test Data folder of the project and list of key, values for admin page url,user name,password,customer email,subject and validation message 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 is used 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, Mail. PageFactory is used in the constructor of this class. The method click is used to click on Marketing, Mail links.

The class 'MailListPage' is defined with method clickOnAdd to click on 'Add New' link which are located using 'FindBy' annotation . PageFactory method is used in the in the constructor of this class.

The class 'SendMailPage' is used to locate the elements 'To',customer email,subject,message by using annotation FindBy.The method addMail is used to enter or select the values for web elements 'To',customer email,subject,message.Click method is used to click on Save button. PageFactory is used in the constructor of this class.

The class DatabaseConnection defines the method countRecords to connect with database and returns the count of records from as_mail table.

The testng_EmailCountDbase xml file is having the parameter 'firefox' to launch Firefox browser.

The annotation @Test is used with method verifyEmailCountDbase in the class 'EmailCount'. This method is used to get the values for username,password , customer email,subject and validation message etc. The get method is called to launch the web url for admin portal . The objects for classes PageLogin,PageDashboard, , MailListPage, SendMailPage, DatabaseConnection are created and the above necessary methods are called on these objects. The executeQuery method is used to execute the query to retrieve the count of the records from as_mail table and next method used on result set object.

Finally the Assert is used to verify presence of the record for added email, the increased count of records and expected message when email is added in admin page.

