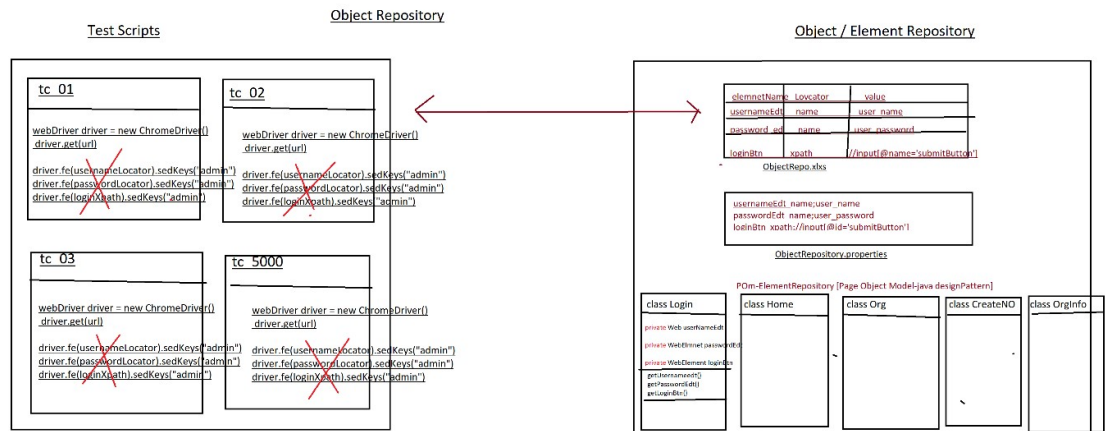


## 1. What is Object/Elements/POMRepository

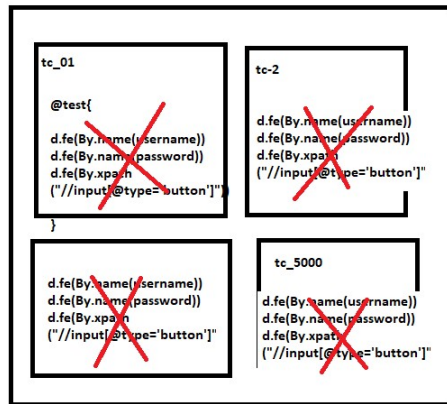
Its collection of elements locators & business libraries in one place & its developed using POM design pattern



## 2. Why Object repository ?

As per the rule of the automation, we should not ~~hardcode~~ [fixed] elements with in test Scripts, instead we should get elements from Object Repository , because in Agile process due to frequent requirement changes , modification & maintenance of elements is tedious job

EG : below is the example of Gmail application GUI changes

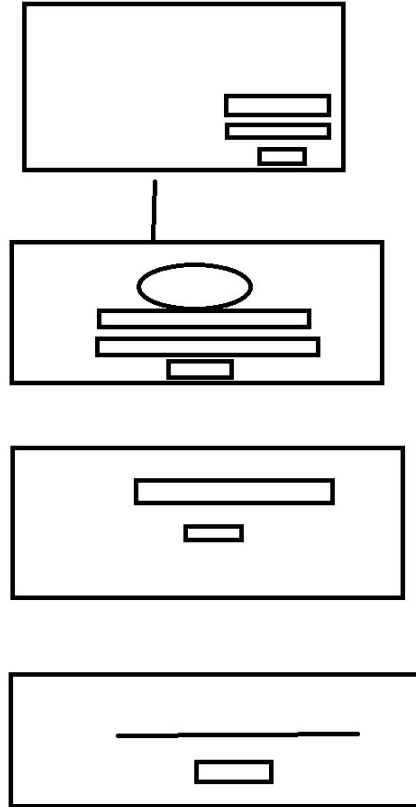


2010

GUI Changes

2014

2017



### 3. What is the advantages repository?

- Reusability of elements, no need to write xpath& otherlocators again & again
- Modification in Repository is easy, when GUI changes frequently
- Maintenance is easy, because all the elements we kept in one place
- Test Script Code Optimized via business reusable libraries
- More Readability
- Test Script development is faster due to business lib
- Test Script is more robust
- Handle Stale Elements Exception.

### 4. What is POM?

POM is a java design pattern preferred by google to develop object repository.

### 5. Why POM ?

It's a well-organized structured design pattern, where we can maintain all the web elements in page wise, due to POM design pattern maintains & modification is easy & faster.

### 6. Advantages of POM:

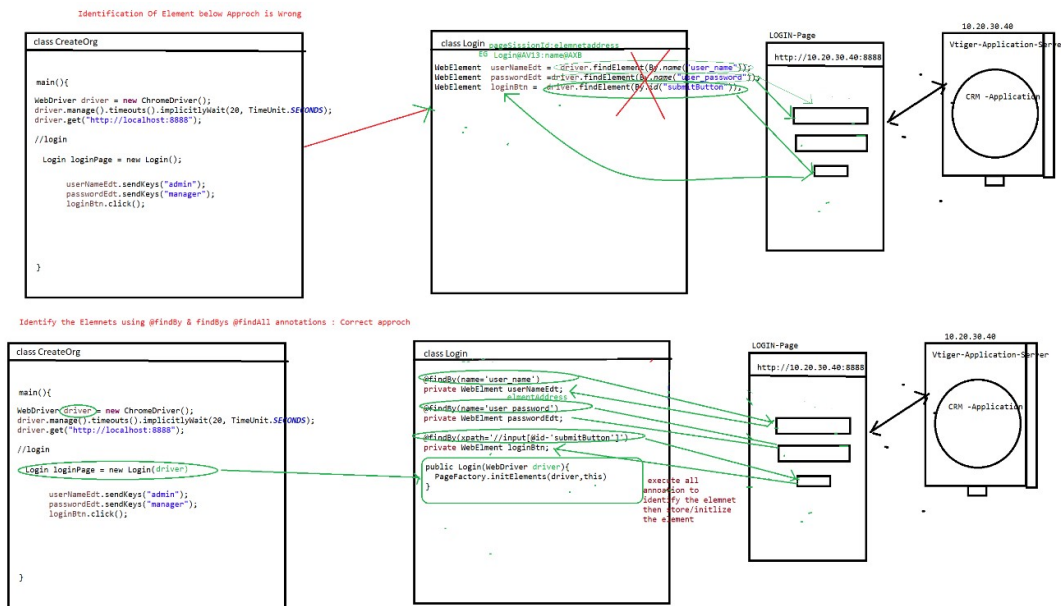
- Well organized structure
- Handle stale element exception.
- maintains & modification of element is easy
- We can directly store Web Elements in java class
- Better fit for Agile processes
- Support Auto heal feature

## 7. Why @FindBy(locator) annotation instead of driver.findElement("locator")

Ans : to avoid staleElementReferenceException

## 8. What is staleElementReferenceException ?

It's one of selenium Exception , whenever webdriver try to identify an element , element was available in GUI, but at time of performing an action on the elements element was not recognized due to **page got refreshed** or **elements may become old** or **element not attached to page** in such case we get **staleElementReferenceException**



## 9. Rules of POM

**Rule 1** : create separate java class for every page in a application & class name should be same page name

**Rule 2** : Identify all the elements using @findBy & @findAll , @findbys annotations & store them in specific pom / java class (**Element declaration**)

**Rule 3** : For Every POM class create Constructor to get an Object of the class & initialize the Page Elements , in order to initialize all the page Elements we should use Pagefactory.initElement() (**Element initialization**)

**Rule 4** : declare all the WebElements as private & provide getters methods to access elements in testScripts class [**this processes is called Encapsulation**]

**Note** : to create getters methods inside the java class follow below steps

→ place cursor inside the class → Right click → source → generate getters & setters → select the getters check box → click on ok button

**Rule 5** : Go to every page & identify the reusable business libraries & implement them in same POM class

## 10. Difference between POM &PageFactory design pattern?

POM is java design pattern, where will maintain all the Web element locator in well-organized manner

Pagefactory it's an extended design pattern of POM , which is used to create an Object to POM classes , &at the time of object creation it will execute all @findBy @findbys annotation then initialize all the elements

### Difference between @findBy , @findAll&@findBys annotation

All annotation available in Selenium webdriver, its traditional ways to identify the elements in GUI.

@findBy : used to identify the element using one locator or one condition

@findAll : it contains multiple @findBy annotation , it mean we can identify the same element using multiple locator (multiple conditions) , it will use OR condition during execution of locator

```
@findAll({ @findBy(@id='username') , @findBy(name='user')})
```

```
Private WebelementsuserNAmeEdt;
```

Note : using above concepts we can achieve **Autohealing** technique

**AutoHealing**: during execution , if one locator fails to identify the element , it will retry to identify the same element using another locator

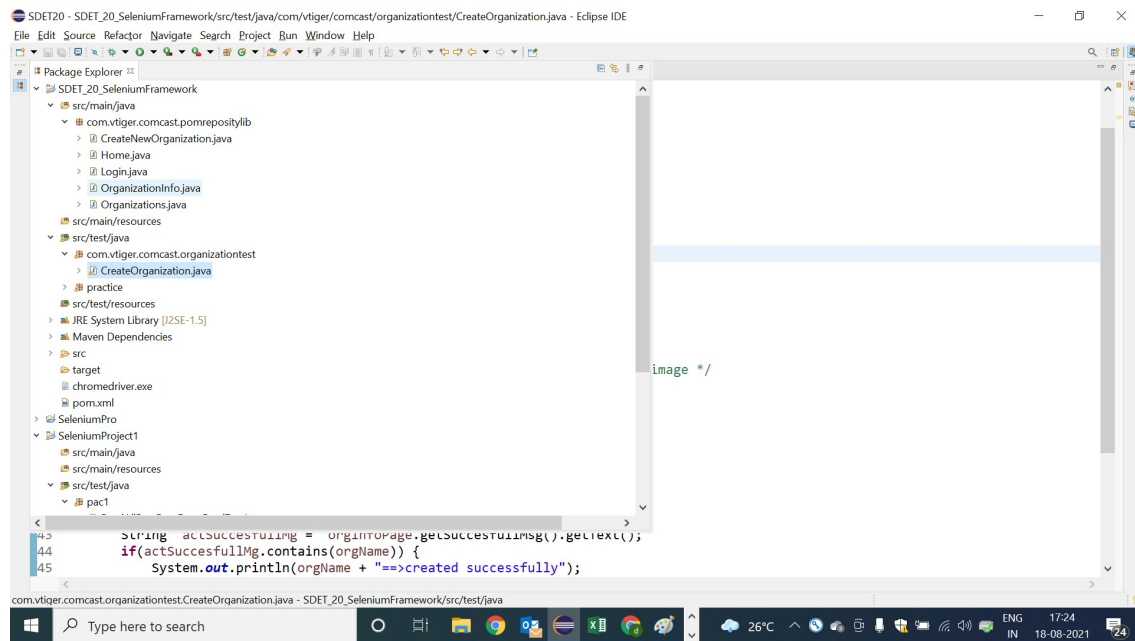
@FindBys :it contains multiple @findBy annotation , it mean we can identify the elements using multiple locator (multiple conditions) , it will use AND condition to during execution of locator

```
@findBys({ @findBy(@id='username') , @findBy(name='user')})
```

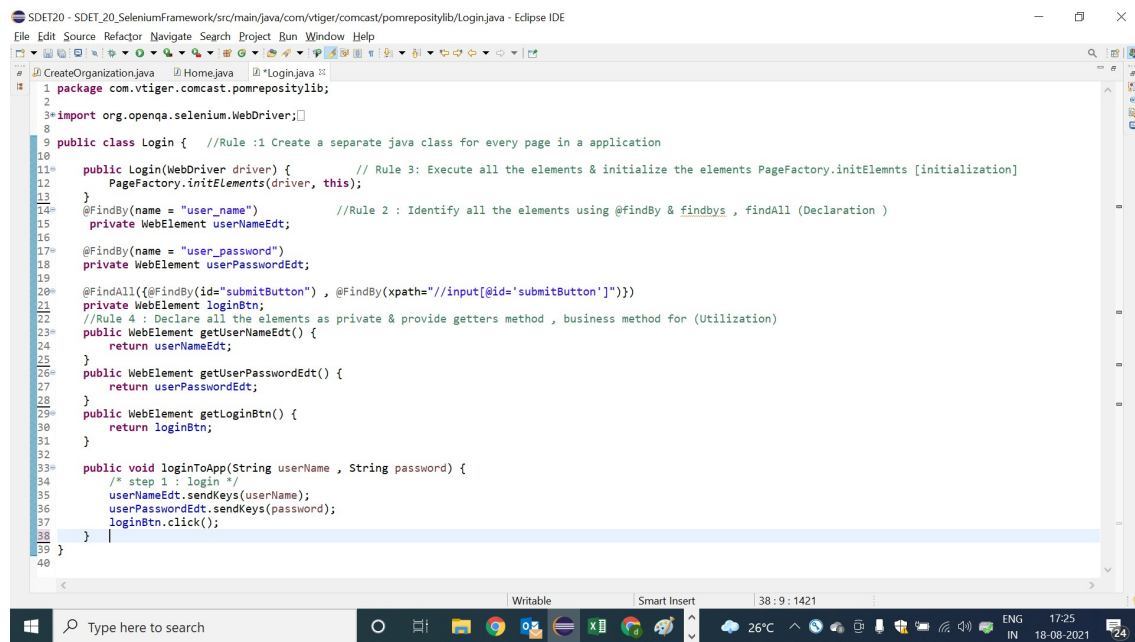
```
Private WebelementsuserNAmeEdt;
```

---

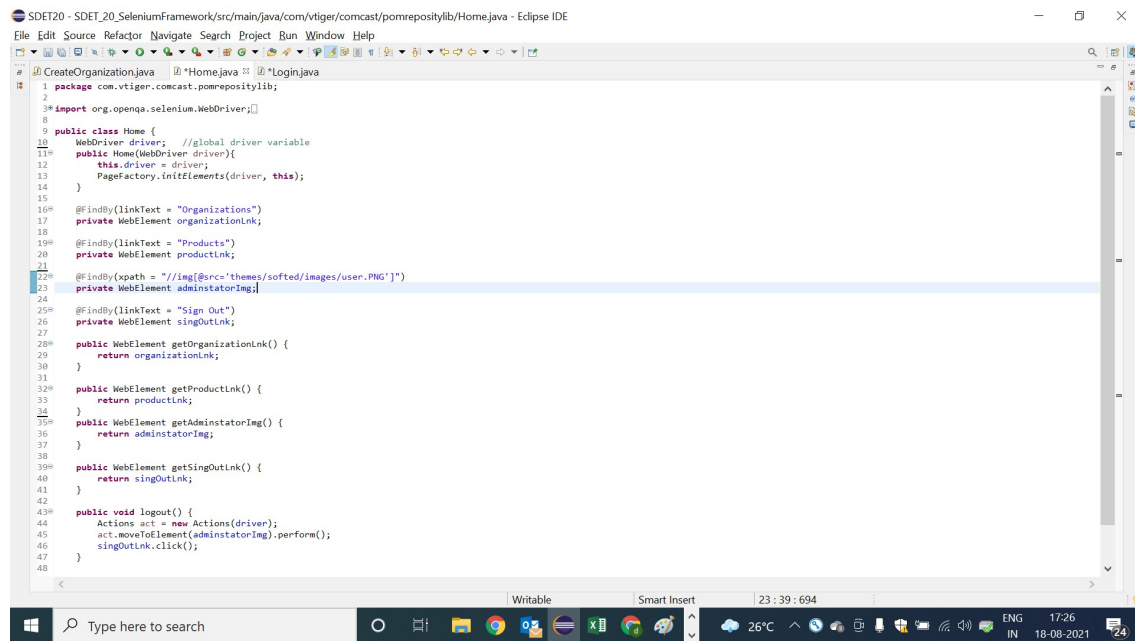
## Project Structure



## PomClasses : Login

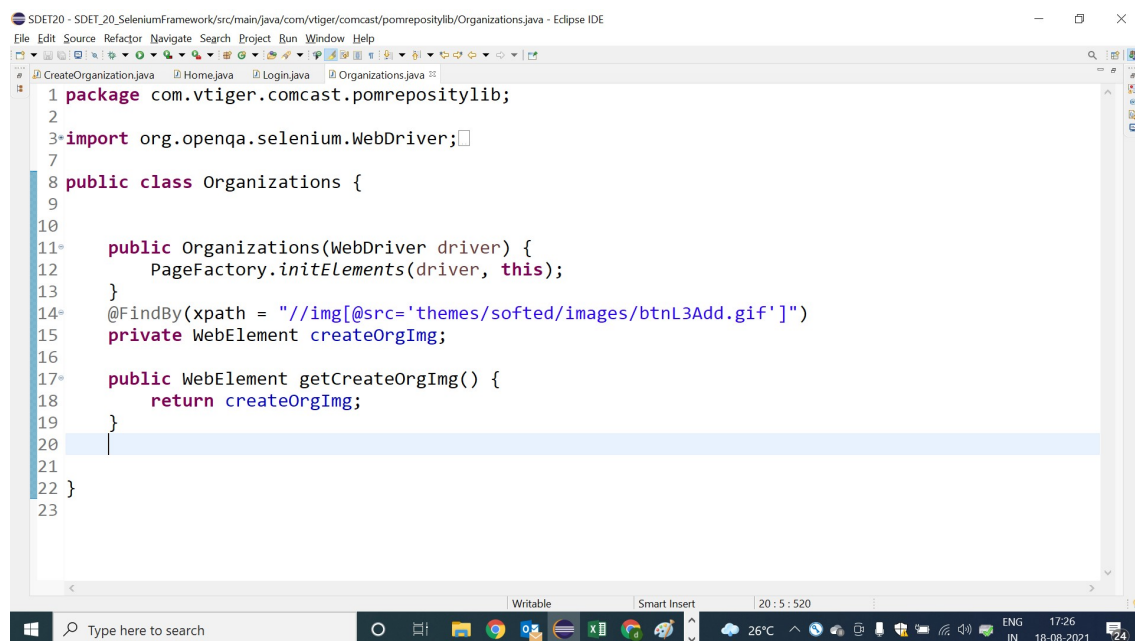


## PomClasses : Home



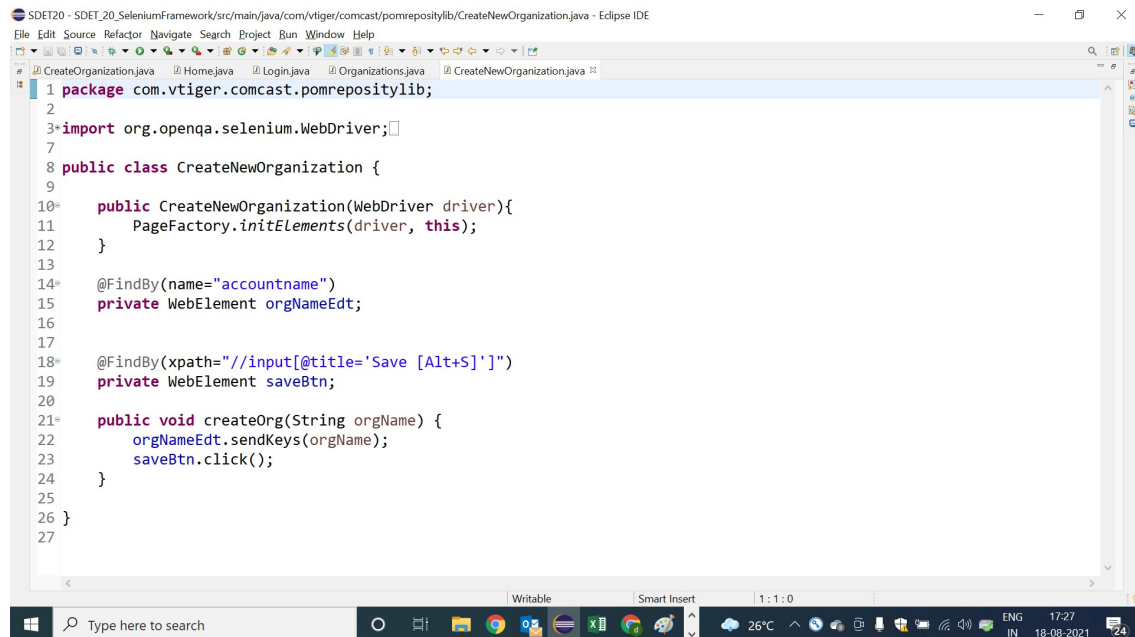
```
1 package com.vtiger.comcast.pomrepositorylib;
2
3 import org.openqa.selenium.WebDriver;
4
5 public class Home {
6     WebDriver driver; //global driver variable
7     public Home(WebDriver driver){
8         this.driver = driver;
9         PageFactory.initElements(driver, this);
10    }
11
12    @FindBy(linkText = "Organizations")
13    private WebElement organizationLink;
14
15    @FindBy(linkText = "Products")
16    private WebElement productLink;
17
18    @FindBy(xpath = "//img[@src='themes/softed/images/user.PNG']")
19    private WebElement administratorImg;
20
21    @FindBy(linkText = "Sign Out")
22    private WebElement singOutLink;
23
24    public WebElement getOrganizationLink() {
25        return organizationLink;
26    }
27
28    public WebElement getProductLink() {
29        return productLink;
30    }
31
32    public WebElement getAdministratorImg() {
33        return administratorImg;
34    }
35
36    public WebElement getSingOutLink() {
37        return singOutLink;
38    }
39
40    public void logout() {
41        Actions act = new Actions(driver);
42        act.moveToElement(administratorImg).perform();
43        singOutLink.click();
44    }
45
46 }
47
48
```

## POM class : Organization



```
1 package com.vtiger.comcast.pomrepositorylib;
2
3 import org.openqa.selenium.WebDriver;
4
5 public class Organizations {
6
7     public Organizations(WebDriver driver) {
8         PageFactory.initElements(driver, this);
9     }
10
11    @FindBy(xpath = "//img[@src='themes/softed/images/btnL3Add.gif']")
12    private WebElement createOrgImg;
13
14    public WebElement getCreateOrgImg() {
15        return createOrgImg;
16    }
17
18 }
19
20
21
22
23
```

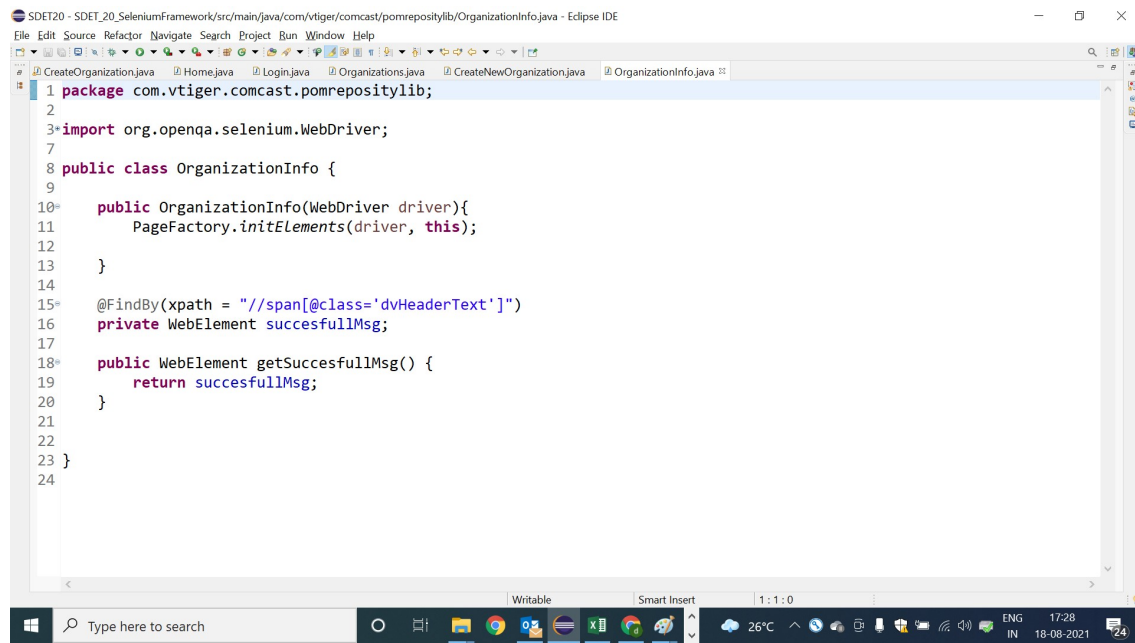
## POM class : Create new Organization Page



The screenshot shows the Eclipse IDE with the file `CreateNewOrganization.java` open. The code is as follows:

```
1 package com.vtiger.comcast.pomrepositorylib;
2
3 import org.openqa.selenium.WebDriver;
4
5 public class CreateNewOrganization {
6
7     public CreateNewOrganization(WebDriver driver){
8         PageFactory.initElements(driver, this);
9     }
10
11     @FindBy(name="accountname")
12     private WebElement orgNameEdt;
13
14     @FindBy(xpath="//input[@title='Save [Alt+S]']")
15     private WebElement saveBtn;
16
17     public void createOrg(String orgName) {
18         orgNameEdt.sendKeys(orgName);
19         saveBtn.click();
20     }
21 }
22
23
24
25
26
27
```

## POMclass :OrganizationInfo



The screenshot shows the Eclipse IDE with the file `OrganizationInfo.java` open. The code is as follows:

```
1 package com.vtiger.comcast.pomrepositorylib;
2
3 import org.openqa.selenium.WebDriver;
4
5 public class OrganizationInfo {
6
7     public OrganizationInfo(WebDriver driver){
8         PageFactory.initElements(driver, this);
9     }
10
11     @FindBy(xpath = "//span[@class='dvHeaderText']")
12     private WebElement succesfullMsg;
13
14     public WebElement getSuccessfullMsg() {
15         return succesfullMsg;
16     }
17 }
18
19
20
21
22
23
24
```



## Test Scripts using POM class

```
SDET22 - SDET_22_ComcastFrameWork/src/test/java/com/crm/comcast/orgTest/CreateOrgTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CreateOrgTest.java CreateNewOrg.java Home.java WebDriverUtility.java CommonData.properties OrgInfoPage.java Login.java CreateContactTest.java

22 23 24
25 public class CreateOrgTest {
26     /*create Objects */
27     ExcelUtility elib = new ExcelUtility();
28     FileUtils flib = new FileUtils();
29     JavaUtility jLib = new JavaUtility();
30     WebDriverUtility wLib = new WebDriverUtility();
31     WebDriver driver = null;
32
33     /*read common data*/
34     String BROWSER = flib.getPropertyKeyValue("browser");
35     String URL = flib.getPropertyKeyValue("url");
36     String USERNAME = flib.getPropertyKeyValue("username");
37     String PASSWORD = flib.getPropertyKeyValue("password");
38
39     /*read test data*/
40     String orgName = elib.getExcelData("org", 1, 2) + "_" + jLib.getRandomData();
41
42     /*step 1 : launch the Browser*/
43     if(BROWSER.equals("firefox")) {
44         driver = new FirefoxDriver();
45     } else if(BROWSER.equals("chrome")) {
46         driver = new ChromeDriver();
47     } else if(BROWSER.equals("ie")) {
48         driver = new InternetExplorerDriver();
49     } else {
50         driver = new ChromeDriver();
51     }
52
53     /*step 2 : login to app*/
54     driver.get(URL);
55     Login lp = new Login(driver);
56     lp.loginToApp(USERNAME, PASSWORD, URL);
57
58     /*step 3 : navigate to Org*/
```

```
SDET22 - SDET_22_ComcastFrameWork/src/test/java/com/crm/comcast/orgTest/CreateOrgTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
CreateOrgTest.java CreateNewOrg.java Home.java WebDriverUtility.java CommonData.properties OrgInfoPage.java Login.java CreateContactTest.java

55 Login lp = new Login(driver);
56 lp.loginToApp(USERNAME, PASSWORD, URL);
57
58 /*step 3 : navigate to Org*/
59 Home hp = new Home(driver);
60 hp.getOrgLnk().click();
61
62 /*step 4 : navigate to create Org Page*/
63 Organizations op = new Organizations(driver);
64 op.getCreateOrgBtn().click();
65
66 /*step 5 : create org*/
67 CreateNewOrg cnop = new CreateNewOrg(driver);
68 cnop.createOrg(orgName);
69
70 /*step 6 : verify */
71 OrgInfoPage oinfo = new OrgInfoPage(driver);
72 wLib.waitForElementVisibility(driver, oinfo.getOrgHeaderSucInfo());
73 String actSucMsg = oinfo.getOrgHeaderSucInfo().getText();
74 if(actSucMsg.contains(orgName)) {
75     System.out.println("org is created successfully==>PASS");
76 } else {
77     System.out.println("org is not created successfully==>FAIL");
78 }
79
80 /*step 7 : logout */
81 hp.logout();
82
83 /*step 8 : close the Browser */
84 driver.close();
85
86
87 }
88
89 }
90
91
```