**Project path:**

UAT: /home/fincash-marketing-automation

LIVE: /opt/Marketing-Automation

**Below are the Test command to execute the project**

**To execute the bhavcopy automation**

**Execution Steps:**

1. Maven command (to run suite file)
2. Goes to suitefile (this is fixed (bhavcopy.xml))
3. Class BhavcopyDownload ( main class to run automation)

mvn test -Dbrowser=headlessChrome -DsuiteFile=bhavcopy.xml -Ddate="28-01-2019" -DdownloadedPath="D:\download" -Duri="http://10.1.0.103:8500/bhavcopy/protected/processbhavcopydata" -Dusername=marketing -Dpassword=test123

**description:**

**-**Dbrowser **:** Browser (headless/chrome)

-DsuiteFile : this is fixed (bhavcopy.xml)

-Ddate: date can be pass in DD-MM-YYYY format

-DdownloadedPath: path to save the file

-Duri: URL of post request for node api.

-Dusername: basic authorization for post request (marketing)

-Dpassword: basic authorization for post request (test123)

**To execute the Screener automation which takes ticker as input**

mvn test -Dbrowser=headlesschrome -DsuiteFile=screener.xml -Dticker="5PAISA" -Duri="http://localhost:8500/screener/protected/processscreenerdata" -Dusername=marketing -Dpassword=test123

**description:**

**-**Dbrowser **:** Browser (headlesschrome/chrome)

-DsuiteFile : this is fixed (screener.xml)

- Dticker: registered company ticker on nse can be fetched from bhavcopy file

-Duri: URL of post request for node api.

-Dusername: basic authorization for post request (marketing)

-Dpassword: basic authorization for post request (test123)

**To execute the AMFI automation which takes city as input**

mvn test -Dbrowser=chrome -DsuiteFile=amfi.xml -Dcity="WARRANGAL" -Duri="http://localhost:8500/amfi/protected/saveAmfidata" -Dusername=marketing -Dpassword=test123

**description:**

**-**Dbrowser **:** Browser (headless/chrome)

-DsuiteFile : this is fixed (bhavcopy.xml)

- Dcity: city name for which the distributor data is required

-Duri: URL of post request for node api.

-Dusername: basic authorization for post request (marketing)

-Dpassword: basic authorization for post request (test123)

**To execute the citylist automation to get all city**

mvn test -Dbrowser=chrome -DsuiteFile=postcitylist.xml -Duri="http://localhost:8500/amfi/protected/getcitylist" -Dusername=marketing -Dpassword=test123

**description:**

**-**Dbrowser **:** Browser (headless/chrome)

-DsuiteFile : this is fixed (bhavcopy.xml)

-Duri: URL of post request for node api.

-Dusername: basic authorization for post request (marketing)

-Dpassword: basic authorization for post request (test123)

**Project Structure : We have followed POM with Pagefactory**

Page Factory is an inbuilt Page Object Model concept for Selenium WebDriver but it is very optimized.

Here as well, we follow the concept of separation of Page Object Repository and Test Methods. Additionally, with the help of PageFactory class, we use annotations **@FindBy** to find WebElement. We use initElements method to initialize web elements

**@FindBy** can accept **tagName, partialLinkText, name, linkText, id, css, className, xpath**as attributes.

These page object repository can be found in **Src/main/java/uiActionWebsiteName** package.

Ex: \src\main\java\uiActionBhavcopy\Bhavcopy.java

WebDriver driver;  
  
public Bhavcopy(WebDriver driver)   
{  
 this.driver =driver;  
 PageFactory.*initElements*(driver,this); //web Element initialization   
}  
  
  
**@FindBy** (xpath = **"//\*[@id='h\_filetype']"**) //locator to find Web Element  
private WebElement selectReport;  
  
**@FindBy** (xpath = **"//\*[@id='date']"**)  
private WebElement date;

Method to interact with Web Element can also be found on same page.

public void chooseDate(String date)  
{  
 waitTillElementToBeVisibleAndCliable(this.date);   
 this.date.clear();  
 this.date.sendKeys(date+Keys.*ENTER*+Keys.*TAB*);  
}

**Pom.xml**

It contains the project configuration details used by Maven like property, dependencies, plugin.

Like system property can be set from here.

1. Declare the properties as below.

<properties>  
 <tickers></tickers>  
 <fromDate></fromDate>  
 <userName></userName>  
 <passWord></passWord>  
</properties>

1. System Varible name/ parameter name can be set like ${ticker} for the property.

<systemPropertyVariables>  
 <tickers>${ticker}</tickers>  
 <fromDate>${date}</fromDate>  
 <downloadPath>${downloadedPath}</downloadPath>  
 <csvFile>${csvFilePath}</csvFile>  
 <URI>${uri}</URI>  
 <userName>${username}</userName>  
 <passWord>${password}</passWord>  
 <cityname>${city}</cityname>  
</systemPropertyVariables>

1. Property can be read by Property name.

String SYMBOL = System.*getProperty*(**"**fromDate**"**); //By property name

1. This can be used in maven command as below, These are caseSensitive.

-Ddate=”” –Dticker=”” –Duri=”” –Dusername=””

**Suitefile.xml**

POM.xml file runs the parameterised Suite file. Suite file instruct helps in running the Test. There can be n number of Test in suite file, and n number of classes in that Test. Below is the example.

<?xml version=**"1.0"** encoding=**"UTF-8"**?>  
<!DOCTYPE suite SYSTEM **"http://testng.org/testng-1.0.dtd"** >  
  
  
  
<suite name=**"Morning Suite"** preserve-order=**"false"** parallel=**"methods"** thread-count=**"10"** verbose=**"1"**>  
  
  
 <test name=**"kamsFileDownloadTest"**>  
 <classes>  
 <class name=**"cams.CamsFileDownload"**></class>  
 </classes>  
 </test>  
  
</suite>

In this suite file, We have written one Test ie **kamsFileDownloadTest** And it runs the one class which is under **cams** package and the class is **CamsFileDownload.**

<class name=**" packagename.classname"**></class>

**Src/test/java >>**

**AMFI:**

router.post('/protected/saveAmfidata', controller.saveAmfidata);

router.post('/protected/saveCityListdata', controller.saveCityListdata); router.post('/protected/runamfiautomation', controller.runamfiautomation);

router.post('/protected/runcitylistautomation', controller.runcitylistautomation); router.all('/protected/automate/:from/:to', controller.automate);

**Bhavcopy:**

router.post('/protected/processbhavcopydata', controller.processbhavcopydata); router.post('/protected/runbhavcopyautomation', controller.processbhavcopyautomation);

**Screener:**

router.post('/protected/processscreenerdata', controller.processscreenerdata); router.post('/protected/runscreenerautomation', controller.processscreenerautomation); router.all('/protected/automate', controller.automate); router.all('/protected/automateBatch/:from/:to', controller.automate);