**Implementing Modular Framework:**

Since Modular Driven framework is separating the commonly used functions and write them as separate scripts lets identify the commonly used functionalities under admin area.

1. Login functionality – Login is a commonly used functionality as you have to login to Admin area if you want to do any thing.
2. Posting Vehicle – Posting vehicle can be a commonly used functionality.

There can other commonly used functionalities like posting brands etc., but let’s implement a modular framework for the following test case initially.

The test case is

Login to admin -> Navigate to Vehicles and then Post Vehicle -> Post a vehicle

Since we identified Login and Post a vehicle sections are commonly used functions, we will write separate test scripts for them.

Create a new project ModularDriven and create a package modules under it and create two classes under this and those classes are

* AdminLoging.java
* AdminVehiclePost.java

AdminLogin.java:

**package** modules;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**public** **class** AdminLogin {

WebDriver driver;

**public** AdminLogin(WebDriver driver) {

**this**.driver = driver;

}

**public** **void** login() {

driver.get("http://localhost/carrental/admin/");

driver.findElement(By.*xpath*("//input[@name='username']")).sendKeys("admin");

driver.findElement(By.*xpath*("//input[@name='password']")).sendKeys("Test@12345");

driver.findElement(By.*xpath*("//button[@name='login']")).click();

}

}

In the above program, we pass the driver that we create in our test when we create the object of the class which will be invoked by the constructor of the class.

Then we have implemented a separate function login() which will does the steps for login.

1. It will open the admin home page
2. It will enter the user name
3. It will enter the password
4. It will click on login button.