**Webdriver Page Object Framework User Guide**

|  |  |  |  |
| --- | --- | --- | --- |
| Author Name | Date | Version | Comments |
| Pradeep Kote | 11/18/2010 | 1.0 | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**System requirements:**

1. JDK 1.6 or higher
2. Eclipse 3.5 or higher
3. Apache Ant 1.6 or higher
4. Webdriver jar file (selenium-server-2.0a5.jar) included in lib
5. TestNG jdk15 or higher(included in lib)

**Webdriver Framework package Structure**

com.ironmountain.pageobject.pageobjectrunner.framework.webdriver

drivers

pages

tests

utils

**How to start with the Webdriver Page Object Framework?**

**1. Defining the classes:**

1. The users should define a page class for each of the web pages in the application. For example LoginPage, etc.

The Page class should be a sub class of BaseWebDriverPage from framework.

1. The test case class should extend BaseWebDriverTest class from framework. BaseWebDriverTest can take WebDriver object in constructor where Project specific driver can be passed. By default it will pick up the driver corresponding to the browser specified in environment file.

The getWebDriver() method can give the WebDriver object which can be used for further operations.

1. Inside drivers package, currently we have CommonInternetExplorerDriver (which will only work for IE browser)

Here we can add common drivers for other browsers also.

1. All Popup pages should extend PopUpWerbDriverPage. This page is already extending BaseWebDriverPage. The PopUpWerbDriverPage keeps the reference of parent window(page) which helps in navigating across the windows.

**2. Defining the locators files**

Currently framework has not defined the place for locators. But it can be properties file instead of xml file. So we don’t need to parse the xml and create some kind of property pool. In case of properties file we can easily load the values with minimum code writing.

**3. Customizing the Framework using Config files**.

Every Project which is using this framework needs to implement getProjectFolder() abstract method which is defined in BaseWebDriverTest class. This will be used to load the Project specific config parameters. Framework assumes that every project has ‘config’ directory and environment.properties file in it. The properties can be execution environment, test case folder names, locations, browser etc.

**4. How the Framework works**

1. First the environment (Project specific) file (config/ environment.properties) is loaded.

2. Then common properties (module specific) file is loaded and appended to environment file

3. The Test case specific properties file is loaded at the end which then appended to common properties files.

4. Once all the parameters are collected in one properties file, test case execution starts and wherever required the parameters are fetched from it.

5. The test case specific properties file can be named as per Test case Id (defined in HP QC or any other test case repository) e.g. TC1001.properties

**5. Utilty classes**

All utility classes and constants (Interfaces) files are kept under utils package. TestCaseUtils is used for loading the config file.

**Reference** :

Confluence page

<http://confluence.englab.local/confluence/display/DAENG/Page+Object+Framework+for+WebDriver>

**Sample testng.xml**

<suite name=*"Search Assets Tests"* verbose=*"2"* >

<parameter name=*"testType"* value=*"search"*/>

<parameter name=*"commonParametersFile"* value=*"searchCommon.properties"*/>

<test name=*"Search By Subject"*>

<parameter name=*"execute"* value=*"true"*/>

<parameter name=*"testID"* value=*"0001"*/>

<parameter name=*"testName"* value=*"Search By Subject"*/>

<parameter name=*"parametersFile"* value=*"TC0001.properties"*/>

<classes>

<class name=*"com.ironmountain.digital.pageobject.drccm.tests.search.SearchBySubject"*>

<methods>

<include name=*"testBySubject"* />

</methods>

</class>

</classes>

</test>

</suite>