**Selenium visual framework help**

|  |  |  |  |
| --- | --- | --- | --- |
| **History** | | | |
| Date | Version | author | Comments |
| 2013-03-21 | 0.1 | Micky Hu  C5182992 | Initial version |
| 2013-09-27 | 0.2 | Micky Hu | Update maven profiles and sample test |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Background**

The test framework is initially developed for SAPUI5 product. It is used to check UI element render in different browsers, now it also can be used for any visual test in web application.

**Features introduction**

1. It is based on Junit4.8 and Selenium2.0

So you can get all benefits of Junit4.8 and Selenium2.0

1. Own User Action to extend Selenium2.0

As Selenium2.0 user action sometimes does not work well for new UI framework like UI5, we develop own user action

1. Take screen shot.

It not only takes full screen shot, but also can get snapshot for the target element.

1. Image Comparer

Compare the target element render with expected image, and then generate diff.

1. Parameterized test execution

Provide a profile in POM.xml, let user define and modify their parameters.

1. Page Object Model

Recommend Page Object pattern for tests design.

1. Closely integrated with Maven, so tests projects need be managed by maven.
2. Multi-Browser support: Firefox, Chrome, IE8, IE9, IE10.

**Getting start in 10 minutes**

1. Create a simple maven project in eclipse.
2. Edit POM.xml and add below contents between tag <project> and </project>.

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* POM SECTION: Repositories Settings

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<repositories>

<repository>

<id>repo.jenkins-ci.org</id>

<url>http://repo.jenkins-ci.org/public/</url>

</repository>

<repository>

<id>nexus</id>

<name>SNAPSHOTS</name>

<url>http://nexus.wdf.sap.corp:8081/nexus/content/groups/build.snapshots</url>

</repository>

</repositories>

<pluginRepositories>

<pluginRepository>

<id>repo.jenkins-ci.org</id>

<url>http://repo.jenkins-ci.org/public/</url>

</pluginRepository>

<pluginRepository>

<id>nexus</id>

<url>http://nexus.wdf.sap.corp:8081/nexus/content/groups/build.snapshots</url>

</pluginRepository>

</pluginRepositories>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* POM SECTION: Build Settings

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>org.eclipse.m2e</groupId>

<artifactId>lifecycle-mapping</artifactId>

<version>1.0.0</version>

<configuration>

<lifecycleMappingMetadata>

<pluginExecutions>

<pluginExecution>

<pluginExecutionFilter>

<groupId>org.codehaus.mojo</groupId>

<artifactId>properties-maven-plugin</artifactId>

<versionRange>[1.0-alpha-2,)</versionRange>

<goals>

<goal>write-project-properties</goal>

</goals>

</pluginExecutionFilter>

<action>

<execute>

<runOnIncremental>true</runOnIncremental>

</execute>

</action>

</pluginExecution>

</pluginExecutions>

</lifecycleMappingMetadata>

</configuration>

</plugin>

</plugins>

</pluginManagement>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.0</version>

<configuration>

<fork>true</fork>

<target>1.6</target>

<source>1.6</source>

</configuration>

</plugin>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>properties-maven-plugin</artifactId>

<version>1.0-alpha-2</version>

<executions>

<execution>

<phase>generate-resources</phase>

<goals>

<goal>write-project-properties</goal>

</goals>

<configuration>

<outputFile>

${env.USERPROFILE}/Automation/Test-Parameters/Config.properties

</outputFile>

</configuration>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.14.1</version>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>com.sap.ui5</groupId>

<artifactId>selenium-visual-framework</artifactId>

<version>0.0.1-SNAPSHOT</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.sap.ui5.selenium</groupId>

<artifactId>core</artifactId>

<version>RELEASE</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.sap.ui5.selenium</groupId>

<artifactId>commons</artifactId>

<version>RELEASE</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.sap.ui5.selenium</groupId>

<artifactId>table</artifactId>

<version>RELEASE</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.sap.ui5.selenium</groupId>

<artifactId>ux3</artifactId>

<version>RELEASE</version>

<scope>test</scope>

</dependency>

</dependencies>

<!--\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* POM SECTION: Profiles Settings

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-->

<profiles>

<profile>

<id>default</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<properties>

<!-- Run mode: DEV, TEST

DEV: If expected images are missing, the test will not be failed and save all new images

Test: If expected images are missing, the test will be failed and stopped. (TEST is default)

-->

<Run.Mode>TEST</Run.Mode>

<!-- Run Environment: Local, Remote

Local: The tests are executed in local machines.(Local is default)

Remote: The tests are executed in remote server by RemoteWebDriver. Grid need use it.

-->

<Run.Environment>Local</Run.Environment>

<!-- Target screen resolution and color depth-->

<Screen.Resolution.Width>1280</Screen.Resolution.Width>

<Screen.Resolution.Height>800</Screen.Resolution.Height>

<Screen.ColorDepth>16</Screen.ColorDepth>

<!-- Expected images root directory -->

<ImageRepositoryHome>C:\Automation\UI5\ImageRepo\master</ImageRepositoryHome>

<!-- OS: WindowsXP, Windows7, Windows8, Windows2008, Windows2012, -->

<Driver.OS.Name>Windows7</Driver.OS.Name>

<!-- OS Bits: 32bits, 64bits -->

<Driver.OS.Bits>64</Driver.OS.Bits>

<!-- Browser: Firefox, Chrome, IE

IE: IE8 32/64bits, IE9 32/64bits, IE10 32/64bits

-->

<Driver.Browser.Name>Firefox</Driver.Browser.Name>

<Driver.Browser.Version>21</Driver.Browser.Version>

<!-- Browser Bits: 32, 64

It is only for IE, so it is ignored by other browsers like chrome, firefox

-->

<Driver.Browser.Bits>32</Driver.Browser.Bits>

<!-- Target AUT base URL: http://veui5infra.dhcp.wdf.sap.corp:8080/uilib-sample-->

<URL.Protocol>http</URL.Protocol>

<URL.Host>veui5infra</URL.Host>

<URL.Domain>dhcp.wdf.sap.corp</URL.Domain>

<URL.Port>8080</URL.Port>

<URL.BasePath>/uilib-sample</URL.BasePath>

<!-- Remote WebDriver Setting, it will take effect if Run.Environment=Remote -->

<!-- If you need use Grid, modify it based on your Grid hub URL -->

<Driver.Remote.Protocol>http</Driver.Remote.Protocol>

<Driver.Remote.Host>veui5infra</Driver.Remote.Host>

<Driver.Remote.Domain>dhcp.wdf.sap.corp</Driver.Remote.Domain>

<Driver.Remote.Port>4444</Driver.Remote.Port>

<!-- sap-ui-theme: sap\_goldreflection, sap\_hcb, sap\_bluecrystal, sap\_platinum -->

<URL.Parameter.sap-ui-theme>sap\_goldreflection</URL.Parameter.sap-ui-theme>

<!-- sap-ui-rtl: false, true-->

<URL.Parameter.sap-ui-rtl>false</URL.Parameter.sap-ui-rtl>

<!-- sap-ui-jqueryversion: 1.7.1, 1.8.1-->

<URL.Parameter.sap-ui-jqueryversion>1.8.1</URL.Parameter.sap-ui-jqueryversion>

</properties>

</profile>

</profiles>

1. Modify the profile if you need.

You can directly modify the properties in profile of POM.xml, OR dynamically specify the value on maven run.

1. Create a package path for you tests.

We suggest that package style like “com.sap.ui5.modules. librarytests.commons.pages” and “com.sap.ui5.modules. librarytests.commons.tests”.

“com.sap.ui5” is your organization or product, you can change it by yourself.

“modules” is required as prefix, it is fixed and should not be changed.

“librarytests” is top module for your tests, you can name it by yourself.

“commons” is sub-module for your tests, you can name it by yourself.

2 level modules would be more flexible to handle possible situation.

“pages” and “tests” are packages by Page Object pattern, their name should not be changed.

1. Create your first test in “com.sap.ui5.modules. librarytests .commons.tests” package.

public class HelloWorld extends TestBase{

private String targetUrl = "/test-resources/sap/ui/commons/ visual /Button.html";

@Before

public void setUp(){

driver.get(getFullUrl(targetUrl));

}

/\*\* Verify full Page UI and all element initial UI, Check enable and visible all elements \*/

@Test

public void testAllElements(){

//Test all initial elements

verifyFullPageUI("full-initial");

}

}

1. If your test extends from TestBase class, your test could have lots of utility methods, like take screenshot, verify target element, Also you can use userAction to interactive UI elements, like click, mouse over.