Test Factory

Base Framework using Selenium Java

Version 1.0

Stand 21.12.2018

Status Draft

Author Kaushal Potdar & Vanshika Chauhan

Protection category: internal

Introduction

Framework Capabilities

1. The framework is driven by a single user config file which drives the way the whole test suite should execute and the reporting configurations.
2. View the offline Extent Reports along with a centralized web application to view the historical reports and trend analysis across the multiple build test runs.

Setup

1. Java(Skip the step if Java is already installed)
   1. Download and install the JDK from [here](https://www.oracle.com/technetwork/java/javase/downloads/index.html).
   2. Once installed open cmd prompt and type ‘java -version’. If you don’t see the installed version then there is an error in the installation or the system environment variable is not set. Please set the system path variable in the environment variable section then,
      1. Output for the above command on windows

“ java version "<<version\_installed>>"

Java(TM) SE Runtime Environment (build <<version>>)

Java HotSpot(TM) 64-Bit Server VM (build <<version>>, mixed mode) ”

1. Eclipse IDE(Skip the step if Eclipse is already installed)
   1. Download the latest version of ‘Eclipse IDE for Java Developers’ from [here](http://www.eclipse.org/downloads/).
2. Maven Setup(Skip the step if Maven is already installed)
   1. Download the Maven binary zip archive file from [here](https://maven.apache.org/download.cgi).
   2. Extract the archive to the directory you wish to install Maven.
   3. Set the environment variable using system properties.
      1. M2\_HOME=<<path to the extracted maven folder until the folder containing the bin subfolder>>
      2. M2=%M2\_HOME%\bin
      3. Append the Maven bin directory to the system path. Append the string %M2% to the end of the system variable path.
      4. Verify the maven installation. Open cmd prompt and run ‘mvn --version’.
         1. Output for the above command on windows:

“ Apache Maven <<version>>

Java version: <<version>>

Java home: C:\Program Files\Java\jdk<<version>> \jre “

1. TestNG Framework Installation
   1. In eclipse click on ‘Help’ menu -> ‘Install New Software’ option.
   2. Enter the URL (http://beust.com/eclipse/) at Work with field and click on "Add" button. TestNG Plugin

JavaDoc

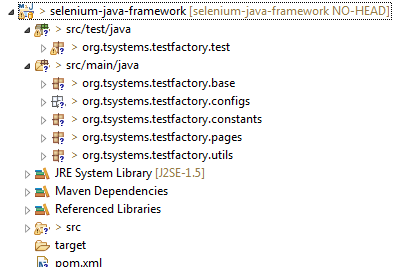
UseCase(AmazonEx)

Report

* How to view the report.
* Configurations

Common Errors

Folder Structure



**src/test/java –** All the user developed project’s automation test cases should be written here.

**src/main/java -**

**<<packagename/groupid>>/base** – All the function wrappers for the repeating selenium methods, driver initialization methods reside here.

**<<packagename/groupid>>/config** – Property files per every page reside here. All the locators for the elements on the page are defined in the property files.

**<<packagename/groupid>>/constants** – All the file paths to the external resources like driver, excel files and also path to the config folder resides here.

**<<packagename/groupid>>/pages** – All the actions that can be performed per page resides in individual page .java files.

**<<packagename/groupid>>/utils –** All the common utilities, property file reading, excel I/O operations, database connection configurations code reside here.

Setup

1. Setup the Maven Project
2. Generate Java Doc

Architecture Description

* The framework is based on Maven Structure.

|  |  |  |  |
| --- | --- | --- | --- |
| **Group Id** | **Artifact Id** | **Version** | **Purpose** |
| commons-io | commons-io | 2.6 | Library contains utility classes, stream implementations, file filters, file comparators and much more. |
| com.aventstack | extentreports | 4.0.5 | Extent Framework |
| com.relevantcodes | extentreports | 2.41.2 |  |
| org.apache.httpcomponents | httpclient | 4.5.1 | Apache Http Components Client |
| log4j | log4j | 1.2.17 | Log4j Framework for logging purpose. |
| org.apache.poi | poi | 3.17 | Java API to access microsoft format files. |
| org.apache.poi | poi-ooxml | 3.17 |
| org.apache.poi | poi-ooxml-schemas | 3.17 |
| org.seleniumhq.selenium | selenium-ie-driver | 3.14.0 |  |
| org.seleniumhq.selenium | selenium-java | 3.14.0 | Browser automation framework and ecosystem |
| org.testng | testng | 6.11 | Testing framework for the JVM |

Dependency Description

Prerequisites

**Requirement documents**: All applicable requirements and performance relevant agreements must be accessible.

**Architecture specification:** The hard- and software architecture is well-documented and the information is accessible.

**Test cases**: The project provides test cases that are supposed to be executable and to have passed functional testing when the scripting phase starts. The test data needed per single execution is fully specified.

**Contact person:** The project provides a contact person available to answer open issues, e.g. related to scenarios, technical issues, or work orders on a daily basis.