Configuración de una aplicación para poder ejecutar las tareas de construcción de pruebas con Framework de Pruebas de Dextra (DTF)

Contents

[Consideraciones iniciales 2](#_Toc330289081)

[Configuración del POM 2](#_Toc330289082)

[Propiedades 2](#_Toc330289083)

[Dependencias 2](#_Toc330289084)

[dtf-generator-maven-plugin 2](#_Toc330289085)

[maven-surefire-plugin 3](#_Toc330289086)

[maven-surefire-report-plugin 3](#_Toc330289087)

[selenium-maven-plugin 4](#_Toc330289088)

[Archivos de configuración 5](#_Toc330289089)

# Consideraciones iniciales

* Este documento explica los pasos necesario para configurar la ejecución de pruebas en una aplicación Web Java.
* Se espera que la aplicación Java a probar tenga una estructura Maven estándar.
* El proyecto petclinic contiene las configuraciones necesarias para la utilización del plugin de pruebas. Se puede usar este proyecto como una guía de configuración.
* Para más información sobre las distintas opciones de configuración se puede consultar el documento DTF-Arquitectura-de-la-solucion.docx y la documentación HTML en el módulo dtf-generator-maven-plugin-site del repositorio TESTING.

# Configuración del POM

Para agregar a un proyecto la capacidad de ejecutar las pruebas Web usando el plugin, solo es necesario agregar la siguiente configuración en el pom.xml.

## Propiedades

Se debe definer las siguientes propiedades:

<properties>

<selenium.browser>\*chrome</selenium.browser>

<selenium.suite>src/test/selenium/testSuite1.html</selenium.suite>

<selenium.startURL>http://google.com.mx</selenium.startURL>

</properties>

## Dependencias

Son necesarias la siguientes dependencias.

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.8</version>

</dependency>

<dependency>

<groupId>org.seleniumhq.selenium</groupId>

<artifactId>selenium-java</artifactId>

<version>2.4.0</version>

<scope>test</scope>

</dependency>

## dtf-generator-maven-plugin

<plugin>

<groupId>com.dextratech.dtf</groupId>

<artifactId>dtf-generator-maven-plugin</artifactId>

<version>1.0-SNAPSHOT</version>

<configuration>

<verbose>true</verbose>

<htmlFile>C:\jesfre\dev\projects\DEXTRA-testing-framework\petclinic-trunk\src\test\resources\layouts\htmlGenerado.html</htmlFile>

<finalName>testSuitePrueba.html</finalName>

<baseUrl>http://localhost:8484/</baseUrl>

<timeout>4500</timeout>

<testSuitesBasePackage>com.dextratech.dtf</testSuitesBasePackage>

</configuration>

</plugin>

## maven-surefire-plugin

Se encargará de ejecutar las pruebas usando las clases Java generadas automáticamente por el dtf-generator-maven-plugin.

<plugin>

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

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

<configuration>

<includes>

<include>\*\*/test/temp/\*\*/Suite\*.java</include>

</includes>

<excludes>

<exclude>\*\*/\*.java</exclude>

</excludes>

</configuration>

<executions>

<execution>

<id>integration-tests</id>

<phase>integration-test</phase>

<goals>

<goal>test</goal>

</goals>

<configuration>

<skip>false</skip>

<excludes>

<exclude>\*\*/samples/\*Tests.java</exclude>

</excludes>

<includes>

<include>\*\*/test/temp/\*\*/Suite\*.java</include>

</includes>

</configuration>

</execution>

</executions>

</plugin>

## maven-surefire-report-plugin

Con este plugin se pueden generar reportes HTML de los resultados de las pruebas ejecutadas por el plugin anterior.

<plugin>

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

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

<version>2.12</version>

<configuration>

<showSuccess>true</showSuccess>

<linkXRef>true</linkXRef>

</configuration>

</plugin>

## selenium-maven-plugin

Provee soporte para la utilización de Selenium con la capacidad de poder iniciar el servidor Selenium usando Remote Control. Utiliza también el Selenium Java Client Driver. También tiene las funcionalidades para correr los scripts de prueba HTML.

<plugin>

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

<artifactId>selenium-maven-plugin</artifactId>

<version>2.3</version>

<configuration>

<browser>${selenium.browser}</browser>

<suite>${selenium.suite}</suite>

<startURL>${selenium.startURL}</startURL>

</configuration>

<executions>

<execution>

<id>start</id>

<phase>pre-integration-test</phase>

<goals>

<goal>start-server</goal>

</goals>

<configuration>

<skip>false</skip>

<background>true</background>

<logOutput>true</logOutput>

<multiWindow>true</multiWindow>

</configuration>

</execution>

<execution>

<id>stop</id>

<phase>post-integration-test</phase>

<goals>

<goal>stop-server</goal>

</goals>

<configuration>

<skip>true</skip>

</configuration>

</execution>

<execution>

<id>test-html</id>

<phase>integration-test</phase>

<goals>

<goal>selenese</goal>

</goals>

<configuration>

<skip>true</skip>

</configuration>

</execution>

</executions>

</plugin>

# Archivos de configuración

|  |  |
| --- | --- |
| Archivo | GlobalConfiguration.dtd |
| Descripción | Se trata del archivo de definición del XML que contendrá las configuraciones globales. |
| Configuraciones del archivo | <?xml version="1.0" encoding="UTF-8"?>  <!ELEMENT dtfConfig (host,tiempoEspera,navegador,url,puertoSelenium,usuario,password,listaCasosPrueba)>  <!ELEMENT host (#PCDATA)>  <!ELEMENT tiempoEspera (#PCDATA)>  <!ELEMENT navegador (#PCDATA)>  <!ELEMENT url (#PCDATA)>  <!ELEMENT puertoSelenium (#PCDATA)>  <!ELEMENT usuario (#PCDATA)>  <!ELEMENT password (#PCDATA)>  <!ELEMENT listaCasosPrueba (casoPrueba+)>  <!ELEMENT casoPrueba (nombreTest,archivoNombre,ejecutar)>    <!ELEMENT nombreTest (#PCDATA)>  <!ELEMENT archivoNombre (#PCDATA)>  <!ELEMENT ejecutar (#PCDATA)> |

|  |  |
| --- | --- |
| Archivo | globalConfiguration.xml |
| Descripción | Contiene las configuraciones globales para las pruebas, así como un listado de las pruebas que se han de ejecutar. |
| Configuraciones del archivo | <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <!DOCTYPE dtfConfig SYSTEM "GlobalConfiguration.dtd" >  <dtfConfig>  <host>localhost</host>  <tiempoEspera>5000</tiempoEspera>  <navegador>\*googlechrome</navegador>  <url>http://localhost:8484</url>  <puertoSelenium>4444</puertoSelenium>  <usuario>rsanchez</usuario>  <password>Rsanchez123</password>  <listaCasosPrueba>  <casoPrueba>  <nombreTest>htmlGenerado-html</nombreTest>  <archivoNombre>testSuitePrueba.html</archivoNombre>  <ejecutar>true</ejecutar>  </casoPrueba>  </listaCasosPrueba>  </dtfConfig> |

|  |  |
| --- | --- |
| Archivo | dataset.xls |
| Descripción | En este archivo se estructurarán los conjuntos de datos de prueba que se usarán para construir nuevos casos de prueba en base a una plantilla creada. La estructura de campos que tendrá este archivo dependerá de los datos que la plantilla puede recibir. |
| Configuraciones del archivo | |  |  |  |  | | --- | --- | --- | --- | | dataTable | s\_value\_1 | s\_value\_2 |  | |  | uno | 1 |  | |  | dos | 2 |  | |  | tres | 3 |  | |  | cuatro | 4 |  | |  | cinco | 5 |  | |  |  |  | dataTable | |

# Ejecución de las pruebas

Para ejecutar las pruebas se deben seguir los siguientes pasos:

1. Crear los archivos de configuración en las carpetas correctas:
   1. \src\test\resources\config\GlobalConfiguration.dtd
   2. \src\test\resources\config\globalConfiguration.xml
   3. \src\test\resources\datasets\
   4. \src\test\resources\layouts\<generado\_selenium\_ide>.html
2. Agregar la configuración en el pom.xml como se muestra en las configuraciones del POM.
3. Correr **>mvn dtf-generator:build-layout** para modificar el html generador por el IDE de Selenium y crear los wildcards para agregar datos de prueba. Esto habrá generado el layout en \target\layout\<generado\_selenium\_ide>.html. También habrá generado los archivos Excel para los datos de prueba en \src\test\resources\datasets\.
4. Correr **>mvn dtf-generator:generate-tests** para generar un test suite de Selenium en HTML a partir del layout generado en el paso anterior. Habrá creado los archivos HTML en \target\testsuites.
5. Correr **>mvn dtf-generator:html2java** para generar las clases Java usando algún driver de Selenium y tomando en cuenta los datos del dataset.xls. Habrá creado las nuevas clases en \src\test\java\com\dextratech\dtf\test\temp\\*.
6. Levantar en una ventana de comandos aparte el servidor de Selenium. Desde la carpeta del proyecto correr **>mvn selenium:start-server**.
7. Nuevamente en el proyecto correr **>mvn dtf-generator:run-tests** para correr las pruebas HTML.
8. Para correr las pruebas Java generadas, **>mvn surefire:test.**
9. Para generar los reportes **>mvn surefire-report:report-only**. Habrá generado el reporte en \target\site\surefire-report.html.