
HttpUnit

INDEX

HttpUnit

Prepared By - Nirmala Kumar Sahu

HttpUnit

- Unit testing tool for web application.
- Generally, after developing web application, we test it by using browser to send request and to get response (in Manual testing environment).
- To automate the unit testing of web application, we need stimulator for the browser software that can created using a Programming API i.e. HttpUnit.
- HttpUnit is developed on the top of JUnit.
- Maven dependency

```
<!-- https://mvnrepository.com/artifact/httpunit/httpunit -->
<dependency>
        <groupId>httpunit</groupId>
        <artifactId>httpunit</artifactId>
        <version>1.7</version>
        <scope>test</scope>
</dependency>
```

Application Development:

Step 1: Create maven project by taking maven-archetype-webapp as the archetype.

[open pom.xml and change java version to 1.8, Right click on the Project maven update the project]

File --> maven project --> next --> select maven-archetype-webapp --> next -->

group Id: nit
artifact Id: HttpUnit-LoginApp
--> finish.

Step 2: Add following jars in pom.xml as dependent by collecting from mvnrepository.com in pom.xml under <dependencies> tag.

- httpunit.1.7.3.jar
- junit-jupiter-api.5.7.0.jar
- javax.servlet.api.4.0.1.jar
- junit-jupiter-engine.5.7.0.jar

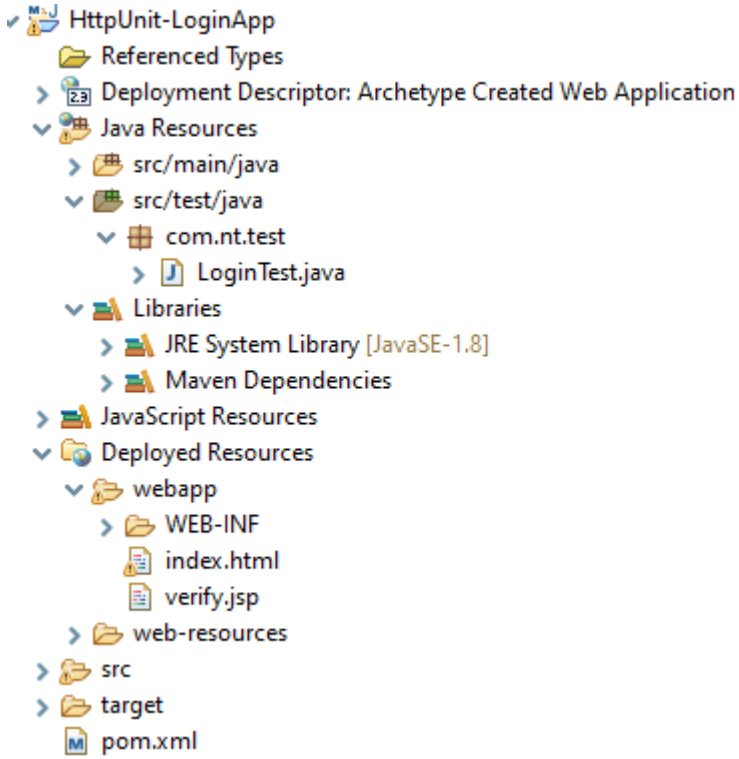
Step 3: Add index.html, verify.jsp as the web components in webapp folder having login application logics.

Step 4: Configure Tomcat server with Eclipse IDE.

Step 5: Add Junit Test class in src/test/java folder using HttpUnit API

Step 6: Run the LoginTest.java class as JUnit

Directory Structure of HttpUnit-LoginApp:



- Develop the above directory Structure and package, class, XML file and add the jar dependencies in pom.xml file then use the following code with in their respective file.

pom.xml

```
        <dependencies>
                <!-- https://mvnrepository.com/artifact/org.httpunit/httpunit -
->
                <dependency>
                        <groupId>org.httpunit</groupId>
                        <artifactId>httpunit</artifactId>
                        <version>1.7.3</version>
                        <scope>test</scope>
                </dependency>
```

```
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-api</artifactId>
            <version>5.7.0</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>javax.servlet-api</artifactId>
            <version>4.0.1</version>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter-engine</artifactId>
            <version>5.7.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

index.html

```
<form action="verify.jsp" method="POST">
    <table border="0" bgcolor="cyan" align="center">
        <tr>
            <td>Enter username: </td>
            <td><input type="text" name="uname"></td>
        </tr>
        <tr>
            <td>Enter password: </td>
            <td><input type="text" name="password"></td>
        </tr>

        <tr>
            <td colspan="2"><input type="submit"
value="Login"></td>
        </tr>

    </table>
</form>
```

verify.jsp

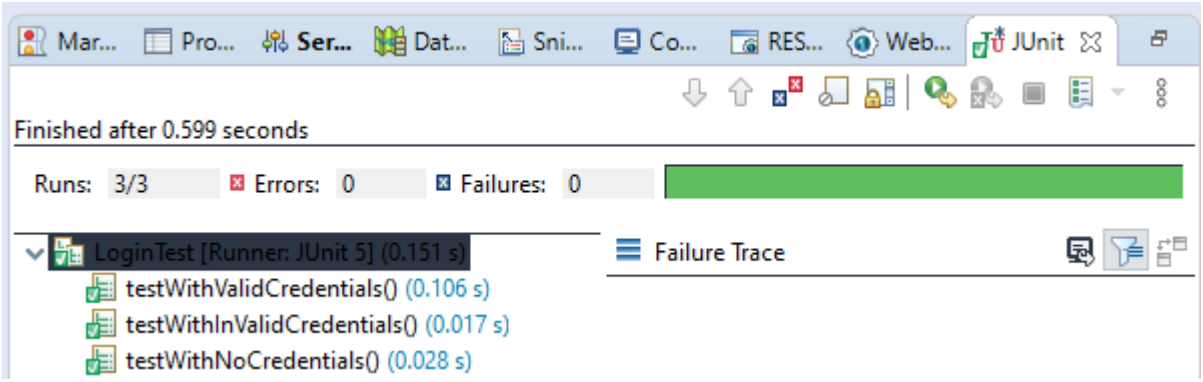
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%
        //read form data
        String user = request.getParameter("uname").trim();
        String pwd = request.getParameter("password").trim();

        if(user.length()==0||user.equals("")||pwd.length()==0||pwd.equals("
")) {
                out.print("provide credentials");
                return;
        }

        //write login/ authentication logic
        if(user.equalsIgnoreCase("raja")&&pwd.equalsIgnoreCase("rani"))
                out.print("valid credential");
        else
                out.print("invalid credential");
%>
```

LoginTest.java

```
package com.nt.test;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;

import com.meterware.httpunit.WebConversation;
import com.meterware.httpunit.WebForm;
import com.meterware.httpunit.WebResponse;

public class LoginTest {

        private static WebConversation conversation;

        @BeforeAll
        public static void setUpOnce() {
                conversation = new WebConversation();
        }

        @Test
```

```
	public void testWithValidCredentials() throws Exception {
		// get response by geneating request to index.html
		WebResponse response =
conversation.getResponse("http://localhost:2525/HttpUnit-
LoginApp/index.html");
		// get access to the form from the response
		WebForm form = response.getForms()[0];
		// set request param values to the form object
		form.setParameter("uname", "raja");
		form.setParameter("password", "rani");
		// submit the form and get the reponse
		WebResponse response1 = form.submit();
		// get actual output from response1 obj
		String actual = response1.getText().trim();
		// perform assertion (compare atual results with expected
results)
		assertEquals("valid credential", actual);
	}

	@Test
	public void testWithInValidCredentials() throws Exception {
		// get response by geneating request to index.html
		WebResponse response =
conversation.getResponse("http://localhost:2525/HttpUnit-
LoginApp/index.html");
		// get access to the form from the response
		WebForm form = response.getForms()[0];
		// set request param values to the form object
		form.setParameter("uname", "raja");
		form.setParameter("password", "rani1");
		// submit the form and get the reponse
		WebResponse response1 = form.submit();
		// get actual output from response1 obj
		String actual = response1.getText().trim();
		// perform assertion (compare atual results with expected
results)
		assertEquals("invalid credential", actual);
	}

	@Test
	public void testWithNoCredentials() throws Exception {
```

```
		// get response by geneating request to index.html
		WebResponse response =
conversation.getResponse("http://localhost:2525/HttpUnit-
LoginApp/index.html");
		// get access to the form from the response
		WebForm form = response.getForms()[0];
		// set request param values to the form object
		form.setParameter("uname", "");
		form.setParameter("password", "");
		// submit the form and get the reponse
		WebResponse response1 = form.submit();
		// get actual output from response1 obj
		String actual = response1.getText().trim();
		// perform assertion (compare atual results with expected
results)
		assertEquals("provide credentials", actual);
	}

	@AfterAll
	public static void cleanOnce() {
		conversation = null;
	}

}
```



---------- The END ----------