

Test Summary for Docler Holding

author: József Kereszturi

1. Introduction

This is a test summary based on the specification handed over by Docler Holding.

The GitHub repository with the test suite and the test cases available at:

<https://github.com/kereszturi87/Docler-Test>

The source code uploaded in both Selenium and JAVA as well.

Test cases were designed and run in Selenium IDE, later converted into JAVA code. The test can be run as a test suite or one by one. The design steps and both the Selenium and JAVA code is included in this documentation as well.

If there were any relevant addition to the results, a short description can be found at the Evaluation part describing it. The summary with the results can be found at the end.

2. Specification, test design and evaluation

2.1. REQ-UI-01

Requirement specification:

The Title should be "UI Testing Site" on every site

Basic flow:

Scope: test title of every site

- step 1: open site
- step 2: assert title
- step 3: click „UI Testing“
- step 4: assert title
- step 5: click „Home“
- step 6: assert title
- step 7: click „Form“
- step 8: assert title
- step 9: input „a“ in the input field
- step 10: click „Go!“
- step 11: assert title
- step 12: open home page
- step 13: click „Error“
- step 14: assert title

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertTitle	UI Testing Site	
clickAndWait	id=site	
assertTitle	UI Testing Site	
clickAndWait	id=home	
assertTitle	UI Testing Site	
clickAndWait	id=form	
assertTitle	UI Testing Site	
type	id=hello-input	a
clickAndWait	id=hello-submit	
assertTitle	UI Testing Site	
open	http://uitest.duodecadits.com/	
clickAndWait	id=error	
assertTitle	UI Testing Site	

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-01</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-01</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>UI Testing Site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>UI Testing Site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>UI Testing Site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td>assertTitle</td>
<td>UI Testing Site</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>id=hello-input</td>
<td>a</td>
</tr>
<tr>
<td>clickAndWait</td>
```

```
<td>id=hello-submit</td>
<td></td>
</tr>
<tr>
  <td>assertTitle</td>
  <td>UI Testing Site</td>
  <td></td>
</tr>
<tr>
  <td>open</td>
  <td>http://uitest.duodecadits.com/</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=error</td>
  <td></td>
</tr>
<tr>
  <td>assertTitle</td>
  <td>UI Testing Site</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI01 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI01() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("UI Testing Site", driver.getTitle());
        driver.findElement(By.id("site")).click();
        assertEquals("UI Testing Site", driver.getTitle());
        driver.findElement(By.id("home")).click();
        assertEquals("UI Testing Site", driver.getTitle());
        driver.findElement(By.id("form")).click();
        assertEquals("UI Testing Site", driver.getTitle());
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("a");
        driver.findElement(By.id("hello-submit")).click();
        assertEquals("UI Testing Site", driver.getTitle());
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("error")).click();
        assertEquals("UI Testing Site", driver.getTitle());
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }

    private boolean isElementPresent(By by) {
        try {
            driver.findElement(by);
            return true;
        } catch (NoSuchElementException e) {
            return false;
        }
    }
}
```

```
private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-01
- [info] Executing: |open | http://uittest.duodecadits.com/ | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [info] Executing: |type | id=hello-input | a |
- [info] Executing: |clickAndWait | id=hello-submit | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [info] Executing: |open | http://uittest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=error | |
- [info] Executing: |assertTitle | UI Testing Site | |
- [error] Actual value '404 Error: File not found :-)' did not match 'UI Testing Site'
- [info] Test case failed

Evaluation

Test case failed as the error page has different Title ('404 Error: File not found :-)').

The screenshot displays the Selenium IDE 2.9.1 interface for test case REQ-UI-01. The base URL is set to http://uittest.duodecadits.com/. The test case steps are listed in the left pane, and the right pane shows the execution details for the 'assertTitle' command. The error message indicates that the actual value '404 Error: File not found :-)' did not match the expected value 'UI Testing Site'.

Command	Target	Value
assertTitle	UI Testing Site	
clickAndWait	id=home	
assertTitle	UI Testing Site	
clickAndWait	id=form	
assertTitle	UI Testing Site	
type	id=hello-input	a
clickAndWait	id=hello-submit	
assertTitle	UI Testing Site	
open	http://uittest.duodecadits.com/	
clickAndWait	id=error	
assertTitle	UI Testing Site	

Command: assertTitle
Target: UI Testing Site
Value:

Log: [info] Executing: |assertTitle | UI Testing Site | |
[info] Executing: |type | id=hello-input | a |
[info] Executing: |clickAndWait | id=hello-submit | |
[info] Executing: |assertTitle | UI Testing Site | |
[info] Executing: |open | http://uittest.duodecadits.com/ | |
[info] Executing: |clickAndWait | id=error | |
[info] Executing: |assertTitle | UI Testing Site | |
[error] Actual value '404 Error: File not found :-)' did not match 'UI Testing Site'
[info] Test case failed

2.2. REQ-UI-02

Requirement specification:

The Company Logo should be visible on every site

Basic flow:

Scope: test logo on every site

- step 1: open site
- step 2: assert logo element is present
- step 3: click „UI Testing“
- step 4: assert logo element is present
- step 5: click „Home“
- step 6: assert logo element is present
- step 7: click „Form“
- step 8: assert logo element is present
- step 9: input „a“ in the input field
- step 10: click „Go!“
- step 11: assert logo element is present
- step 12: open home page
- step 13: click „Error“
- step 14: assert logo element is present

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertElementPresent	id=dh_logo	
clickAndWait	id=site	
assertElementPresent	id=dh_logo	
clickAndWait	id=home	
assertElementPresent	id=dh_logo	
clickAndWait	id=form	
assertElementPresent	id=dh_logo	
type	id=hello-input	a
clickAndWait	id=hello-submit	
assertElementPresent	id=dh_logo	
open	http://uitest.duodecadits.com/	
clickAndWait	id=error	
assertElementPresent	id=dh_logo	

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-02</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-02</td></tr>
</thead><tbody>
<tr>
<td><td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>assertElementPresent</td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><td>assertElementPresent</td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><td>assertElementPresent</td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td><td>assertElementPresent</td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><td>type</td>
<td>id=hello-input</td>
<td>a</td>
</tr>
<tr>
<td><td>clickAndWait</td>
</tr>
</tbody>
</table>
</body>
</html>
```

```
<td>id=hello-submit</td>
<td></td>
</tr>
<tr>
  <td>assertElementPresent</td>
  <td>id=dh_logo</td>
  <td></td>
</tr>
<tr>
  <td>open</td>
  <td>http://uitest.duodecadits.com/</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=error</td>
  <td></td>
</tr>
<tr>
  <td>assertElementPresent</td>
  <td>id=dh_logo</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI02 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI02() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("site")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("home")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("form")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("a");
        driver.findElement(By.id("hello-submit")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("error")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }

    private boolean isElementPresent(By by) {
        try {
            driver.findElement(by);
            return true;
        } catch (NoSuchElementException e) {
            return false;
        }
    }
}
```

```
private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-02
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |type | id=hello-input | a |
- [info] Executing: |clickAndWait | id=hello-submit | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=error | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [error] false
- [info] Test case failed

Evaluation

Test case failed as the error page has no logo present.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-02 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The "Test Case" pane on the left lists several test cases, with "REQ-UI-02" highlighted. The main pane shows a table of commands and their results:

Command	Target	Value
clickAndWait	id=home	
assertElementPresent	id=dh_logo	
clickAndWait	id=form	
assertElementPresent	id=dh_logo	
type	id=hello-input	a
clickAndWait	id=hello-submit	
assertElementPresent	id=dh_logo	
open	http://uitest.duodecadits.com/	
clickAndWait	id=error	
assertElementPresent	id=dh_logo	

Below the table, the "Command" field is set to "assertElementPresent", the "Target" field is "id=dh_logo", and the "Value" field is empty. The "Runs:" counter shows 1, and the "Failures:" counter shows 1.

The "Log" pane at the bottom shows the execution steps and the final error message:

```
[info] Executing: |assertElementPresent | id=dh_logo | |
[info] Executing: |type | id=hello-input | a |
[info] Executing: |clickAndWait | id=hello-submit | |
[info] Executing: |assertElementPresent | id=dh_logo | |
[info] Executing: |open | http://uitest.duodecadits.com/ | |
[info] Executing: |clickAndWait | id=error | |
[info] Executing: |assertElementPresent | id=dh_logo | |
[error] false
[info] Test case failed
```

2.3. REQ-UI-03

Requirement specification:

When I click on the Home button, I should get navigated to the Home page

Basic flow:

Scope: basic Home button navigation

- step 1: open site
- step 2: assert home page
- step 3: verify home page text (for safety reasons)
- step 4: click „UI Testing”
- step 5: click „Home”
- step 6: assert home page
- step 7: verify home page text (for safety reasons)
- step 8: click „Home”
- step 9: assert home page
- step 10: verify home page text (for safety reasons)
- step 11: click „Form”
- step 12: click „Home”
- step 13: assert home page
- step 14: verify home page text (for safety reasons)

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-03</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-03</td></tr>
</thead><tbody>
<tr>
<td><open></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><verifyText></td>
<td><css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><verifyText></td>
<td><css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><verifyText></td>
```



```
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=form</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=home</td>
    <td></td>
</tr>
<tr>
    <td>assertLocation</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
<tr>
    <td>verifyText</td>
    <td>css=h1</td>
    <td>Welcome to the Docler Holding QA Department</td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI03 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI03() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
    }

    @After
    public void tearDown() throws Exception {
```

```
driver.quit();
String verificationErrorString = verificationErrors.toString();
if (!"".equals(verificationErrorString)) {
    fail(verificationErrorString);
}
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-03
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Test case passed

Evaluation

Test case was successful.

There are requirement specification / design problems:

- no back button from the filled „Form” screen, test case does not handle it
- no back button from „Error” screen, test case does not handle it
- the purpose of the „UI Testing” button is not specified

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-03 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The menu bar includes "Fájl", "Szerkesztés", "Actions", "Options", and "Súgó". The "Base URL" is set to "http://uittest.duodecadits.com/". The "Test Case" list on the left shows "REQ-UI-03" selected. The main table lists the following commands and targets:

Command	Target	Value
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...

Below the table, there are input fields for "Command", "Target", and "Value", along with "Select" and "Find" buttons. The "Log" tab at the bottom shows the following messages:

```
[info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
[info] Executing: |clickAndWait | id=home | |
[info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
[info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |clickAndWait | id=home | |
[info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
[info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
[info] Test case passed
```

2.4. REQ-UI-04

Requirement specification:

When I click on the Home button, it should turn to active status

Basic flow:

Scope: Home button active status is present

- step 1: open site
- step 2: assert home page
- step 3: verify home page text (for safety reasons)
- step 4: assert active status
- step 5: click „UI Testing“
- step 6: click „Home“
- step 7: assert home page
- step 8: verify home page text (for safety reasons)
- step 9: assert active status
- step 10: click „Home“
- step 11: assert home page
- step 12: verify home page text (for safety reasons)
- step 13: assert active status
- step 14: click „Form“
- step 15: click „Home“
- step 16: assert home page
- step 17: verify home page text (for safety reasons)
- step 18: assert active status

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertLocation	http://uitest.duodecadits.com/	
assertAttribute	css=a#home@href	*/*
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
assertAttribute	css=a#home@href	*/*
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	

verifyText	css=h1
assertAttribute	css=a#home@href
clickAndWait	id=form
clickAndWait	id=home
assertLocation	http://uitest.duodecadits.com/
verifyText	css=h1
assertAttribute	css=a#home@href

Welcome to the Docler Holding QA Depart...
/

Welcome to the Docler Holding QA Depart...
/

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-04</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-04</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertAttribute</td>
<td>css=a#home@href</td>
<td>*/*</td>
</tr>
<tr>
<td>verifyText</td>
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>verifyText</td>
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td>assertAttribute</td>
<td>css=a#home@href</td>
<td>*/*</td>
</tr>
<tr>
<td>clickAndWait</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
</body>
</html>
```



```

        <td>id=home</td>
        <td></td>
    </tr>
    <tr>
        <td>assertLocation</td>
        <td>http://uitest.duodecadits.com/</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyText</td>
        <td>css=h1</td>
        <td>Welcome to the Docler Holding QA Department</td>
    </tr>
    <tr>
        <td>assertAttribute</td>
        <td>css=a#home@href</td>
        <td>*/*</td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=home</td>
        <td></td>
    </tr>
    <tr>
        <td>assertLocation</td>
        <td>http://uitest.duodecadits.com/</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyText</td>
        <td>css=h1</td>
        <td>Welcome to the Docler Holding QA Department</td>
    </tr>
    <tr>
        <td>assertAttribute</td>
        <td>css=a#home@href</td>
        <td>*/*</td>
    </tr>
</tbody></table>
</body>
</html>

```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI04 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI04() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertTrue(driver.findElement(By.cssSelector("a#home")).getAttribute("href").matches("^\\s\\S*/[\\s\\S]*$"));
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("a#home")).getAttribute("href").matches("^\\s\\S*/[\\s\\S]*$"));
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("a#home")).getAttribute("href").matches("^\\s\\S*/[\\s\\S]*$"));
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        try {
            assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("a#home")).getAttribute("href").matches("^\\s\\S*/[\\s\\S]*$"));
    }
}
```

```

}

@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}

```

Result

- [info] Playing test case REQ-UI-04
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Test case passed

Evaluation

Test case was successful.

REQ-UI-03 test case updated with one extra assertion parameter examining active status.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-04 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The "Test Case" list on the left includes REQ-UI-01 through REQ-UI-12 and an additional test case. REQ-UI-04 is selected. The "Table" view shows the following commands and their results:

Command	Target	Value
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
assertAttribute	css=a#home@href	*/*
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
assertAttribute	css=a#home@href	*/*
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
verifyText	css=h1	Welcome to the Docler Holding QA Depart...
assertAttribute	css=a#home@href	*/*

Below the table, the "Command", "Target", and "Value" fields are empty. The "Runs" counter shows 1, and the "Failures" counter shows 0.

The "Log" tab at the bottom shows the following execution log:

- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |verifyText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |assertAttribute | css=a#home@href | */* |
- [info] Test case passed

2.5. REQ-UI-05

Requirement specification:

When I click on the Form button, I should get navigated to the Form page

Basic flow:

Scope: basic Form button navigation

- step 1: open site
- step 2: click „UI Testing”
- step 3: click „Form”
- step 4: assert current location page
- step 5: verify form text (for safety reasons)
- step 6: click „Home”
- step 7: click „Form”
- step 8: assert current location page
- step 9: verify form text (for safety reasons)
- step 10: click „Form”
- step 11: click „Form”
- step 12: assert current location page
- step 13: verify form text (for safety reasons)

Command	Target	Value
open	http://uitest.duodecadits.com/	
clickAndWait	id=site	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
clickAndWait	id=home	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
clickAndWait	id=form	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-05</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-05</td></tr>
</thead><tbody>
<tr>
<td><td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td><td>assertLocation</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td><td>verifyText</td>
<td>css=h1</td>
<td>Simple Form Submission</td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td><td>assertLocation</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td><td>verifyText</td>
<td>css=h1</td>
<td>Simple Form Submission</td>
</tr>
<tr>
<td><td>clickAndWait</td>
</tr>
</tbody>
</table>
```

```
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>assertLocation</td>
        <td>http://uitest.duodecadits.com/form.html</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyText</td>
        <td>css=h1</td>
        <td>Simple Form Submission</td>
    </tr>
</tbody></table>
</body>
</html>
```


JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI05 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI05() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("home")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }
}
```

```
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-05
- [info] Executing: |open | http://uittest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Test case passed

Evaluation

Test case was successful.

Same remarks as REQ-UI-05.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-05 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The "Base URL" is set to "http://uittest.duodecadits.com/". The "Test Case" list on the left shows "REQ-UI-05" selected. The "Table" view on the right lists the commands and their targets/values:

Command	Target	Value
clickAndWait	id=form	
assertLocation	http://uittest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
clickAndWait	id=home	
clickAndWait	id=form	
assertLocation	http://uittest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
clickAndWait	id=form	
clickAndWait	id=form	
assertLocation	http://uittest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission

The "Log" tab at the bottom shows the execution details:

```
[info] Executing: |clickAndWait | id=home | |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
[info] Executing: |verifyText | css=h1 | Simple Form Submission |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
[info] Executing: |verifyText | css=h1 | Simple Form Submission |
[info] Test case passed
```

The "Runs" counter shows 1, and the "Failures" counter shows 0.

2.6. REQ-UI-06

Requirement specification:

When I click on the Form button, it should turn to active status

Basic flow:

Scope: Form button active status is present

- step 1: open site
- step 2: click „UI Testing”
- step 3: click „Form”
- step 4: assert current location page
- step 5: verify form text (for safety reasons)
- step 6: assert active status
- step 7: click „Home”
- step 8: click „Form”
- step 9: assert current location page
- step 10: verify form text (for safety reasons)
- step 11: assert active status
- step 12: click „Form”
- step 13: click „Form”
- step 14: assert current location page
- step 15: verify form text (for safety reasons)
- step 16: assert active status

Command	Target	Value
open	http://uitest.duodecadits.com/	
clickAndWait	id=site	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
assertAttribute	css=a#form@href	*/form.html*
clickAndWait	id=home	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
assertAttribute	css=a#form@href	*/form.html*
clickAndWait	id=form	
clickAndWait	id=form	
assertLocation	http://uitest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
assertAttribute	css=a#form@href	*/form.html*

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-06</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-06</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td>verifyText</td>
<td>css=h1</td>
<td>Simple Form Submission</td>
</tr>
<tr>
<td>assertAttribute</td>
<td>css=a#form@href</td>
<td>*/form.html*</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td>verifyText</td>
```

```

        <td>css=h1</td>
        <td>Simple Form Submission</td>
    </tr>
    <tr>
        <td>assertAttribute</td>
        <td>css=a#form@href</td>
        <td>*/form.html*</td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>assertLocation</td>
        <td>http://uitest.duodecadits.com/form.html</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyText</td>
        <td>css=h1</td>
        <td>Simple Form Submission</td>
    </tr>
    <tr>
        <td>assertAttribute</td>
        <td>css=a#form@href</td>
        <td>*/form.html*</td>
    </tr>
</tbody></table>
</body>
</html>

```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI06 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI06() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }

        assertTrue(driver.findElement(By.cssSelector("a#form")).getAttribute("href").matches("^\\|s\\|S\\|*/form\\.html/[\\|s\\|S\\|*$"));
        driver.findElement(By.id("home")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }

        assertTrue(driver.findElement(By.cssSelector("a#form")).getAttribute("href").matches("^\\|s\\|S\\|*/form\\.html/[\\|s\\|S\\|*$"));
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("form")).click();
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
        try {
            assertEquals("Simple Form Submission", driver.findElement(By.cssSelector("h1")).getText());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }

        assertTrue(driver.findElement(By.cssSelector("a#form")).getAttribute("href").matches("^\\|s\\|S\\|*/form\\.html/[\\|s\\|S\\|*$"));
    }
}
```

```

}

@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}

```


Result

- [info] Playing test case REQ-UI-06
- [info] Executing: |open | http://uittest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Executing: |assertAttribute | css=a#form@href | */form.html* |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Executing: |assertAttribute | css=a#form@href | */form.html* |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |verifyText | css=h1 | Simple Form Submission |
- [info] Executing: |assertAttribute | css=a#form@href | */form.html* |
- [info] Test case passed

Evaluation

Test case was successful.

REQ-UI-05 test case updated with one extra assertion parameter examining active status.

The screenshot displays the Selenium IDE 2.9.1 interface for test case REQ-UI-06. The main window shows a table of test steps with columns for Command, Target, and Value. The test case is currently running, as indicated by the 'Runs' counter showing 1 and 'Failures' showing 0. The log at the bottom shows the execution of the test case, including assertions and clicks, and confirms that the test case passed.

Command	Target	Value
open	http://uittest.duodecadits.com/	
clickAndWait	id=site	
clickAndWait	id=form	
assertLocation	http://uittest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
assertAttribute	css=a#form@href	*/form.html*
clickAndWait	id=home	
clickAndWait	id=form	
assertLocation	http://uittest.duodecadits.com/form.html	
verifyText	css=h1	Simple Form Submission
assertAttribute	css=a#form@href	*/form.html*

Log:

```
[info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
[info] Executing: |verifyText | css=h1 | Simple Form Submission |
[info] Executing: |assertAttribute | css=a#form@href | */form.html* |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |assertLocation | http://uittest.duodecadits.com/form.html | |
[info] Executing: |verifyText | css=h1 | Simple Form Submission |
[info] Executing: |assertAttribute | css=a#form@href | */form.html* |
[info] Test case passed
```

2.7. REQ-UI-07

Requirement specification:

When I click on the Error button, I should get a 404 HTTP response code

Basic flow:

Scope: 404 error code is present

- step 1: open site
- step 2: click „UI Testing”
- step 3: click „Error”
- step 4: verify current location page
- step 5: assert text
- step 6: assert title
- step 7: open site
- step 8: click „Home”
- step 9: click „Error”
- step 10: verify current location page
- step 11: assert text
- step 12: assert title
- step 13: open site
- step 14: click „Form”
- step 15: click „Error”
- step 16: verify current location page
- step 17: assert text
- step 18: assert title

Command	Target	Value
open	http://uitest.duodecadits.com/	
clickAndWait	id=site	
clickAndWait	id=error	
verifyLocation	http://uitest.duodecadits.com/error	
assertText	css=h1	*404*
assertTitle	*404*	
open	http://uitest.duodecadits.com/	
clickAndWait	id=home	
clickAndWait	id=error	
verifyLocation	http://uitest.duodecadits.com/error	
assertText	css=h1	*404*

assertTitle	*404*	
open	http://uitest.duodecadits.com/	
clickAndWait	id=form	
clickAndWait	id=error	
verifyLocation	http://uitest.duodecadits.com/error	
assertText	css=h1	*404*
assertTitle	*404*	

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-07</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-07</td></tr>
</thead><tbody>
<tr>
<td><td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=error</td>
<td></td>
</tr>
<tr>
<td><td>verifyLocation</td>
<td>http://uitest.duodecadits.com/error</td>
<td></td>
</tr>
<tr>
<td><td>assertText</td>
<td>css=h1</td>
<td>*404*</td>
</tr>
<tr>
<td><td>assertTitle</td>
<td>*404*</td>
<td></td>
</tr>
<tr>
<td><td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=error</td>
<td></td>
</tr>
<tr>
<td><td>verifyLocation</td>
```

```
<td>http://uitest.duodecadits.com/error</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
<td>css=h1</td>
<td>*404*</td>
</tr>
<tr>
<td>assertTitle</td>
<td>*404*</td>
<td></td>
</tr>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=error</td>
<td></td>
</tr>
<tr>
<td>verifyLocation</td>
<td>http://uitest.duodecadits.com/error</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
<td>css=h1</td>
<td>*404*</td>
</tr>
<tr>
<td>assertTitle</td>
<td>*404*</td>
<td></td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI07 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI07() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("error")).click();
        try {
            assertEquals("http://uitest.duodecadits.com/error", driver.getCurrentUrl());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("h1")).getText().matches("^\\s\\S*404\\s\\S*$"));
        assertTrue(driver.getTitle().matches("^\\s\\S*404\\s\\S*$"));
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("home")).click();
        driver.findElement(By.id("error")).click();
        try {
            assertEquals("http://uitest.duodecadits.com/error", driver.getCurrentUrl());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("h1")).getText().matches("^\\s\\S*404\\s\\S*$"));
        assertTrue(driver.getTitle().matches("^\\s\\S*404\\s\\S*$"));
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("error")).click();
        try {
            assertEquals("http://uitest.duodecadits.com/error", driver.getCurrentUrl());
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertTrue(driver.findElement(By.cssSelector("h1")).getText().matches("^\\s\\S*404\\s\\S*$"));
        assertTrue(driver.getTitle().matches("^\\s\\S*404\\s\\S*$"));
    }

    @After
    public void tearDown() throws Exception {
```

```
driver.quit();
String verificationErrorString = verificationErrors.toString();
if (!"".equals(verificationErrorString)) {
    fail(verificationErrorString);
}
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-07
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=error | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/error | |
- [info] Executing: |assertText | css=h1 | *404* |
- [info] Executing: |assertTitle | *404* | |
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |clickAndWait | id=error | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/error | |
- [info] Executing: |assertText | css=h1 | *404* |
- [info] Executing: |assertTitle | *404* | |
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=error | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/error | |
- [info] Executing: |assertText | css=h1 | *404* |
- [info] Executing: |assertTitle | *404* | |
- [info] Test case passed

Evaluation

Test case was successful.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-07 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The "Base URL" is set to "http://uittest.duodecadits.com/". The "Test Case" list on the left includes REQ-UI-01 through REQ-UI-12, with REQ-UI-07 selected. The "Table" view shows the following commands and their targets/values:

Command	Target	Value
clickAndWait	id=home	
clickAndWait	id=error	
verifyLocation	http://uittest.duodecadits.com/error	
assertText	css=h1	*404*
assertTitle	*404*	
open	http://uittest.duodecadits.com/	
clickAndWait	id=form	
clickAndWait	id=error	
verifyLocation	http://uittest.duodecadits.com/error	
assertText	css=h1	*404*
assertTitle	*404*	

Below the table, the "Command", "Target", and "Value" fields are empty. The "Runs" counter shows 1, and the "Failures" counter shows 0. The "Log" panel at the bottom shows the following messages:

- [info] Executing: |assertText | css=h1 | *404* |
- [info] Executing: |assertTitle | *404* |
- [info] Executing: |open | http://uittest.duodecadits.com/ |
- [info] Executing: |clickAndWait | id=form |
- [info] Executing: |clickAndWait | id=error |
- [info] Executing: |verifyLocation | http://uittest.duodecadits.com/error |
- [info] Executing: |assertText | css=h1 | *404* |
- [info] Executing: |assertTitle | *404* |
- [info] Test case passed

2.8. REQ-UI-08

Requirement specification:

When I click on the Logo, I should get navigated to the Home page

Basic flow:

Scope: logo is functional as Home button

- step 1: open site
- step 2: assert location
- step 3: assert logo element is present
- step 4: click logo element and wait for page to load
- step 5: assert home page location
- step 6: click „UI Testing”
- step 7: assert logo element is present
- step 8: click logo element and wait for page to load
- step 9: assert home page location
- step 10: click „Home”
- step 11: assert logo element is present
- step 12: click logo element and wait for page to load
- step 13: assert home page location
- step 14: click „Form”
- step 15: assert logo element is present
- step 16: click logo element and wait for page to load
- step 17: assert home page location
- step 18: click „Form”
- step 19: input „a” in the input field
- step 20: click „Go!”
- step 21: assert logo element is present
- step 22: click logo element and wait for page to load
- step 23: assert home page location
- step 24: open home page
- step 25: click „Error”
- step 26: assert logo element is present
- step 27: click logo element and wait for page to load
- step 28: assert home page location

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertLocation	http://uitest.duodecadits.com/	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	
clickAndWait	id=site	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	
clickAndWait	id=home	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	
clickAndWait	id=form	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	
clickAndWait	id=form	
type	id=hello-input	a
clickAndWait	id=hello-submit	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	
open	http://uitest.duodecadits.com/	
clickAndWait	id=error	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uitest.duodecadits.com/	

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-08</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-08</td></tr>
</thead><tbody>
<tr>
<td><open></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><assertElementPresent></td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><assertElementPresent></td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=dh_logo</td>
<td></td>
</tr>
<tr>
<td><assertLocation></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
</tr>

```

	<td>id=home</td>
	<td></td>
</tr>	
<tr>	
	<td>assertElementPresent</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	
<tr>	
	<td>assertLocation</td>
	<td>http://uitest.duodecadits.com/</td>
	<td></td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=form</td>
	<td></td>
</tr>	
<tr>	
	<td>assertElementPresent</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	
<tr>	
	<td>assertLocation</td>
	<td>http://uitest.duodecadits.com/</td>
	<td></td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=form</td>
	<td></td>
</tr>	
<tr>	
	<td>type</td>
	<td>id=hello-input</td>
	<td>a</td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=hello-submit</td>
	<td></td>
</tr>	
<tr>	
	<td>assertElementPresent</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	
<tr>	
	<td>clickAndWait</td>
	<td>id=dh_logo</td>
	<td></td>
</tr>	

```
<tr>
    <td>assertLocation</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
<tr>
    <td>open</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=error</td>
    <td></td>
</tr>
<tr>
    <td>assertElementPresent</td>
    <td>id=dh_logo</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=dh_logo</td>
    <td></td>
</tr>
<tr>
    <td>assertLocation</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI08 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI08() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        driver.findElement(By.id("site")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        driver.findElement(By.id("home")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        driver.findElement(By.id("form")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("a");
        driver.findElement(By.id("hello-submit")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        driver.get("http://uitest.duodecadits.com/");
        driver.findElement(By.id("error")).click();
        assertTrue(isElementPresent(By.id("dh_logo")));
        driver.findElement(By.id("dh_logo")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
    }
}
```

```
String verificationErrorString = verificationErrors.toString();
if (!"".equals(verificationErrorString)) {
    fail(verificationErrorString);
}
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```


Result

- [info] Playing test case REQ-UI-08
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=dh_logo | |
- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=dh_logo | |
- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=dh_logo | |
- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=dh_logo | |
- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [error] Actual value 'http://uitest.duodecadits.com/form.html' did not match 'http://uitest.duodecadits.com/'
- [info] Test case failed

Evaluation

Test case failed, the logo does not function as a home page link. The test did skip the part when the origination and destination are the same (UI Testing > Home, Home > Home) due to timeout, though if the logo is working it would give the correct result. This does not change anything, just a remark for future use.

The screenshot displays the Selenium IDE 2.9.1 interface for a test suite named 'REQ-UI-08 (DOCLER TEST SUITE)'. The base URL is 'http://uittest.duodecadits.com/'. The test case list on the left includes REQ-UI-01 through REQ-UI-12, with REQ-UI-08 highlighted. The main pane shows a table of test steps for REQ-UI-08, which includes commands like 'assertElementPresent', 'clickAndWait', and 'assertLocation' with targets such as 'id=dh_logo' and 'http://uittest.duodecadits.com/'. Below the table, there are input fields for 'Command', 'Target', and 'Value'. At the bottom, the log pane shows the execution details, including a red error message: '[error] Timed out after 30000ms' and '[error] Actual value 'http://uittest.duodecadits.com/form.html' did not match 'http://uittest.duodecadits.com/'', indicating the test failure.

Command	Target	Value
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uittest.duodecadits.com/	
clickAndWait	id=home	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uittest.duodecadits.com/	
clickAndWait	id=form	
assertElementPresent	id=dh_logo	
clickAndWait	id=dh_logo	
assertLocation	http://uittest.duodecadits.com/	

Log:

- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |assertElementPresent | id=dh_logo | |
- [info] Executing: |clickAndWait | id=dh_logo | |
- [error] Timed out after 30000ms
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
- [error] Actual value 'http://uittest.duodecadits.com/form.html' did not match 'http://uittest.duodecadits.com/'
- [info] Test case failed

2.9. REQ-UI-09

Requirement specification:

The following text should be visible on the Home page in <h1> tag:

"Welcome to the Docler Holding QA Department"

Basic flow:

Scope: the given text is present at the home page

- step 1: open site
- step 2: assert home page
- step 3: assert required text
- step 4: click „UI Testing“
- step 5: click „Home“
- step 6: assert home page
- step 7: assert required text
- step 8: click „Home“
- step 9: assert home page
- step 10: assert required text
- step 11: click „Form“
- step 12: click „Home“
- step 13: assert home page
- step 14: assert required text

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-09</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-09</td></tr>
</thead><tbody>
<tr>
<td><td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>assertText</td>
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>assertText</td>
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
<td><td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><td>assertText</td>
```

```
<td>css=h1</td>
<td>Welcome to the Docler Holding QA Department</td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=form</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=home</td>
    <td></td>
</tr>
<tr>
    <td>assertLocation</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
<tr>
    <td>assertText</td>
    <td>css=h1</td>
    <td>Welcome to the Docler Holding QA Department</td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI09 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI09() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("Welcome to the Docler Holding QA Department", driver.findElement(By.cssSelector("h1")).getText());
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }

    private boolean isElementPresent(By by) {
        try {
            driver.findElement(by);
            return true;
        } catch (NoSuchElementException e) {
            return false;
        }
    }
}
```

```
private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-09
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Test case passed

Evaluation

Test case was successful.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-09 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The menu bar includes "Fájl", "Szerkesztés", "Actions", "Options", and "Súgó". The "Base URL" is set to "http://uittest.duodecadits.com/".

On the left, the "Test Case" list shows several test cases, with "REQ-UI-09" highlighted in grey. Below the list, the status indicates "Runs: 1" and "Failures: 0".

The main area features a "Table" with "Source" and "Table" tabs. The "Table" tab is active, showing a list of commands and their targets/values:

Command	Target	Value
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=h1	Welcome to the Docler Holding QA Depart...

Below the table, there are input fields for "Command", "Target", and "Value", along with "Select" and "Find" buttons.

At the bottom, the "Log" tab is active, showing a series of log entries:

- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=home |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Executing: |clickAndWait | id=form |
- [info] Executing: |clickAndWait | id=home |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ |
- [info] Executing: |assertText | css=h1 | Welcome to the Docler Holding QA Department |
- [info] Test case passed

2.10. REQ-UI-10

Requirement specification:

The following text should be visible on the Home page in <p> tag:

"This site is dedicated to perform some exercises and demonstrate automated web testing."

Basic flow:

Scope: the given text is present at the home page

- step 1: open site
- step 2: assert home page
- step 3: assert required text
- step 4: click „UI Testing“
- step 5: click „Home“
- step 6: assert home page
- step 7: assert required text
- step 8: click „Home“
- step 9: assert home page
- step 10: assert required text
- step 11: click „Form“
- step 12: click „Home“
- step 13: assert home page
- step 14: assert required text

Command	Target	Value
open	http://uitest.duodecadits.com/	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=form	
clickAndWait	id=home	
assertLocation	http://uitest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-10</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-10</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
<td>css=p</td>
<td>This site is dedicated to perform some exercises and demonstrate automated web testing.</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
<td>css=p</td>
<td>This site is dedicated to perform some exercises and demonstrate automated web testing.</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td>assertLocation</td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
```

```
<td>css=p</td>
<td>This site is dedicated to perform some exercises and demonstrate automated web testing.</td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=form</td>
    <td></td>
</tr>
<tr>
    <td>clickAndWait</td>
    <td>id=home</td>
    <td></td>
</tr>
<tr>
    <td>assertLocation</td>
    <td>http://uitest.duodecadits.com/</td>
    <td></td>
</tr>
<tr>
    <td>assertText</td>
    <td>css=p</td>
    <td>This site is dedicated to perform some exercises and demonstrate automated web testing.</td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI10 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI10() throws Exception {
        driver.get("http://uitest.duodecadits.com/");
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("This site is dedicated to perform some exercises and demonstrate automated web testing.",
            driver.findElement(By.cssSelector("p")).getText());
        driver.findElement(By.id("site")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("This site is dedicated to perform some exercises and demonstrate automated web testing.",
            driver.findElement(By.cssSelector("p")).getText());
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("This site is dedicated to perform some exercises and demonstrate automated web testing.",
            driver.findElement(By.cssSelector("p")).getText());
        driver.findElement(By.id("form")).click();
        driver.findElement(By.id("home")).click();
        assertEquals("http://uitest.duodecadits.com/", driver.getCurrentUrl());
        assertEquals("This site is dedicated to perform some exercises and demonstrate automated web testing.",
            driver.findElement(By.cssSelector("p")).getText());
    }

    @After
    public void tearDown() throws Exception {
        driver.quit();
        String verificationErrorString = verificationErrors.toString();
        if (!"".equals(verificationErrorString)) {
            fail(verificationErrorString);
        }
    }

    private boolean isElementPresent(By by) {
        try {
            driver.findElement(by);
            return true;
        } catch (NoSuchElementException e) {
            return false;
        }
    }
}
```

```
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [info] Playing test case REQ-UI-10
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uitest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Test case passed

Evaluation

Test case was successful.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-10 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The menu bar includes "Fájl", "Szerkesztés", "Actions", "Options", and "Súgó". The Base URL is set to "http://uittest.duodecadits.com/".

On the left, the "Test Case" list shows several test cases, with "REQ-UI-10" highlighted in grey. Below the list, it indicates "Runs: 1" and "Failures: 0".

The main area is divided into "Table" and "Source" tabs. The "Table" tab is active, showing a table of commands and their values:

Command	Target	Value
open	http://uittest.duodecadits.com/	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=site	
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=home	
assertLocation	http://uittest.duodecadits.com/	
assertText	css=p	This site is dedicated to perform some exerci...
clickAndWait	id=form	

Below the table, there are input fields for "Command", "Target", and "Value", along with "Select" and "Find" buttons.

At the bottom, the "Log" tab is active, showing a list of execution messages:

- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |assertLocation | http://uittest.duodecadits.com/ | |
- [info] Executing: |assertText | css=p | This site is dedicated to perform some exercises and demonstrate automated web testing. |
- [info] Test case passed

2.11. REQ-UI-11

Requirement specification:

On the Form page, a form should be visible with one input box and one submit button

Basic flow:

Scope: input box and submit button are present at the Form page

- step 1: open site
- step 2: click „UI Testing“
- step 3: click „Form“
- step 4: verify current location page (for safety reasons)
- step 5: assert text box present
- step 6: assert button present
- step 7: click „Home“
- step 8: click „Form“
- step 9: verify current location page (for safety reasons)
- step 10: assert text box present
- step 11: assert button present
- step 12: click „Form“
- step 13: click „Form“
- step 14: verify current location page (for safety reasons)
- step 15: assert text box present
- step 16: assert button present

Command	Target	Value
open	http://uitest.duodecadits.com/	
clickAndWait	id=site	
clickAndWait	id=form	
verifyLocation	http://uitest.duodecadits.com/form.html	
assertElementPresent	id=hello-input	
assertElementPresent	id=hello-submit	
clickAndWait	id=home	
clickAndWait	id=form	
verifyLocation	http://uitest.duodecadits.com/form.html	
assertElementPresent	id=hello-input	
assertElementPresent	id=hello-submit	
clickAndWait	id=form	
clickAndWait	id=form	
verifyLocation	http://uitest.duodecadits.com/form.html	
assertElementPresent	id=hello-input	
assertElementPresent	id=hello-submit	

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-11</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-11</td></tr>
</thead><tbody>
<tr>
<td><open></td>
<td>http://uitest.duodecadits.com/</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=site</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td><verifyLocation></td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td><assertElementPresent></td>
<td>id=hello-input</td>
<td></td>
</tr>
<tr>
<td><assertElementPresent></td>
<td>id=hello-submit</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=home</td>
<td></td>
</tr>
<tr>
<td><clickAndWait></td>
<td>id=form</td>
<td></td>
</tr>
<tr>
<td><verifyLocation></td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td><assertElementPresent></td>
```

```

        <td>id=hello-input</td>
        <td></td>
    </tr>
    <tr>
        <td>assertElementPresent</td>
        <td>id=hello-submit</td>
        <td></td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=form</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyLocation</td>
        <td>http://uitest.duodecadits.com/form.html</td>
        <td></td>
    </tr>
    <tr>
        <td>assertElementPresent</td>
        <td>id=hello-input</td>
        <td></td>
    </tr>
    <tr>
        <td>assertElementPresent</td>
        <td>id=hello-submit</td>
        <td></td>
    </tr>
</tbody></table>
</body>
</html>

```

JAVA

```

package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI11 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

```

```

@Test
public void testREQUI11() throws Exception {
    driver.get("http://uitest.duodecadits.com/");
    driver.findElement(By.id("site")).click();
    driver.findElement(By.id("form")).click();
    try {
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
    } catch (Error e) {
        verificationErrors.append(e.toString());
    }
    assertTrue(isElementPresent(By.id("hello-input")));
    assertTrue(isElementPresent(By.id("hello-submit")));
    driver.findElement(By.id("home")).click();
    driver.findElement(By.id("form")).click();
    try {
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
    } catch (Error e) {
        verificationErrors.append(e.toString());
    }
    assertTrue(isElementPresent(By.id("hello-input")));
    assertTrue(isElementPresent(By.id("hello-submit")));
    driver.findElement(By.id("form")).click();
    driver.findElement(By.id("form")).click();
    try {
        assertEquals("http://uitest.duodecadits.com/form.html", driver.getCurrentUrl());
    } catch (Error e) {
        verificationErrors.append(e.toString());
    }
    assertTrue(isElementPresent(By.id("hello-input")));
    assertTrue(isElementPresent(By.id("hello-submit")));
}

```

```

@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

```

```

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

```

```

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

```

```

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
    }
}

```

```
if (acceptNextAlert) {  
    alert.accept();  
} else {  
    alert.dismiss();  
}  
return alertText;  
} finally {  
    acceptNextAlert = true;  
}  
}  
}
```

Result

- [info] Playing test case REQ-UI-11
- [info] Executing: |open | http://uitest.duodecadits.com/ | |
- [info] Executing: |clickAndWait | id=site | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/form.html | |
- [info] Executing: |assertElementPresent | id=hello-input | |
- [info] Executing: |assertElementPresent | id=hello-submit | |
- [info] Executing: |clickAndWait | id=home | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/form.html | |
- [info] Executing: |assertElementPresent | id=hello-input | |
- [info] Executing: |assertElementPresent | id=hello-submit | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |clickAndWait | id=form | |
- [info] Executing: |verifyLocation | http://uitest.duodecadits.com/form.html | |
- [info] Executing: |assertElementPresent | id=hello-input | |
- [info] Executing: |assertElementPresent | id=hello-submit | |
- [info] Test case passed

Evaluation

Test case was successful.

The screenshot displays the Selenium IDE 2.9.1 interface. The title bar reads "REQ-UI-11 (DOCLER TEST SUITE) - Selenium IDE 2.9.1". The "Test Case" pane on the left lists several test cases, with "REQ-UI-11" highlighted in grey. Below the list, it shows "Runs: 1" and "Failures: 0". The "Table" pane in the center shows a list of commands and their targets for the selected test case. The "Log" pane at the bottom shows the execution steps and their status.

Command	Target	Value
assertElementPresent	id=hello-submit	
clickAndWait	id=home	
clickAndWait	id=form	
verifyLocation	http://uitest.duodecadits.com/form.html	
assertElementPresent	id=hello-input	
assertElementPresent	id=hello-submit	
clickAndWait	id=form	
clickAndWait	id=form	
verifyLocation	http://uitest.duodecadits.com/form.html	
assertElementPresent	id=hello-input	
assertElementPresent	id=hello-submit	

Command: assertElementPresent
Target: id=hello-submit
Value:

Log: [info] Executing: |verifyLocation | http://uitest.duodecadits.com/form.html | |
[info] Executing: |assertElementPresent | id=hello-input | |
[info] Executing: |assertElementPresent | id=hello-submit | |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |clickAndWait | id=form | |
[info] Executing: |verifyLocation | http://uitest.duodecadits.com/form.html | |
[info] Executing: |assertElementPresent | id=hello-input | |
[info] Executing: |assertElementPresent | id=hello-submit | |
[info] Test case passed

2.12. REQ-UI-12

Requirement specification:

On the Form page, if you type <value> the input field and submit the form, you should get redirect to the Hello page, and the following text should appear: <result>

<value> | <result>

=====

John | Hello John!

Sophia | Hello Sophia!

Charlie | Hello Charlie!

Emily | Hello Emily!

Basic flow:

Scope: basic functionality of input Form

- step 1: open site
- step 2: input „John” in the input field
- step 3: click „Go!”
- step 4: verify current location page (for safety reasons)
- step 5: assert text
- step 6: open site
- step 7: input „Sophia” in the input field
- step 8: click „Go!”
- step 9: verify current location page (for safety reasons)
- step 10: assert text
- step 11: open site
- step 12: input „Charlie” in the input field
- step 13: click „Go!”
- step 14: verify current location page (for safety reasons)
- step 15: assert text
- step 16: open site
- step 17: input „Emily” in the input field
- step 18: click „Go!”
- step 19: verify current location page (for safety reasons)
- step 20: assert text

Command	Target	Value
open	http://uitest.duodecadits.com/form.html	
type	id=hello-input	John
clickAndWait	id=hello-submit	
verifyLocation	*http://uitest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello John!
open	http://uitest.duodecadits.com/form.html	
type	id=hello-input	Sophia
clickAndWait	id=hello-submit	
verifyLocation	*http://uitest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello Sophia!
open	http://uitest.duodecadits.com/form.html	
type	id=hello-input	Charlie
clickAndWait	id=hello-submit	
verifyLocation	*http://uitest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello Charlie!
open	http://uitest.duodecadits.com/form.html	
type	id=hello-input	Emily
clickAndWait	id=hello-submit	
verifyLocation	*http://uitest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello Emily!

Source code:

Selenium IDE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://uitest.duodecadits.com/" />
<title>REQ-UI-12</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">REQ-UI-12</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>id=hello-input</td>
<td>John</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=hello-submit</td>
<td></td>
</tr>
<tr>
<td>verifyLocation</td>
<td>*http://uitest.duodecadits.com/hello.html?myName=*</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
<td>id=hello-text</td>
<td>Hello John!</td>
</tr>
<tr>
<td>open</td>
<td>http://uitest.duodecadits.com/form.html</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>id=hello-input</td>
<td>Sophia</td>
</tr>
<tr>
<td>clickAndWait</td>
<td>id=hello-submit</td>
<td></td>
</tr>
<tr>
<td>verifyLocation</td>
<td>*http://uitest.duodecadits.com/hello.html?myName=*</td>
<td></td>
</tr>
<tr>
<td>assertText</td>
```

```

        <td>id=hello-text</td>
        <td>Hello Sophia!</td>
    </tr>
    <tr>
        <td>open</td>
        <td>http://uitest.duodecadits.com/form.html</td>
        <td></td>
    </tr>
    <tr>
        <td>type</td>
        <td>id=hello-input</td>
        <td>Charlie</td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=hello-submit</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyLocation</td>
        <td>*http://uitest.duodecadits.com/hello.html?myName=*</td>
        <td></td>
    </tr>
    <tr>
        <td>assertText</td>
        <td>id=hello-text</td>
        <td>Hello Charlie!</td>
    </tr>
    <tr>
        <td>open</td>
        <td>http://uitest.duodecadits.com/form.html</td>
        <td></td>
    </tr>
    <tr>
        <td>type</td>
        <td>id=hello-input</td>
        <td>Emily</td>
    </tr>
    <tr>
        <td>clickAndWait</td>
        <td>id=hello-submit</td>
        <td></td>
    </tr>
    <tr>
        <td>verifyLocation</td>
        <td>*http://uitest.duodecadits.com/hello.html?myName=*</td>
        <td></td>
    </tr>
    <tr>
        <td>assertText</td>
        <td>id=hello-text</td>
        <td>Hello Emily!</td>
    </tr>
</tbody></table>
</body>
</html>

```

JAVA

```
package com.example.tests;

import java.util.regex.Pattern;
import java.util.concurrent.TimeUnit;
import org.junit.*;
import static org.junit.Assert.*;
import static org.hamcrest.CoreMatchers.*;
import org.openqa.selenium.*;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.Select;

public class REQUI12 {
    private WebDriver driver;
    private String baseUrl;
    private boolean acceptNextAlert = true;
    private StringBuffer verificationErrors = new StringBuffer();

    @Before
    public void setUp() throws Exception {
        driver = new FirefoxDriver();
        baseUrl = "http://uitest.duodecadits.com/";
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
    }

    @Test
    public void testREQUI12() throws Exception {
        driver.get("http://uitest.duodecadits.com/form.html");
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("John");
        driver.findElement(By.id("hello-submit")).click();
        try {

assertTrue(driver.getCurrentUrl().matches("^\\|\\|s\\|S]*http://uitest\\.duodecadits\\.com/hello\\|\\.html\\|\\|s\\|S]*myName=[\\|\\|s\\|S]*$"));
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertEquals("Hello John!", driver.findElement(By.id("hello-text")).getText());
        driver.get("http://uitest.duodecadits.com/form.html");
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("Sophia");
        driver.findElement(By.id("hello-submit")).click();
        try {

assertTrue(driver.getCurrentUrl().matches("^\\|\\|s\\|S]*http://uitest\\.duodecadits\\.com/hello\\|\\.html\\|\\|s\\|S]*myName=[\\|\\|s\\|S]*$"));
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
        assertEquals("Hello Sophia!", driver.findElement(By.id("hello-text")).getText());
        driver.get("http://uitest.duodecadits.com/form.html");
        driver.findElement(By.id("hello-input")).clear();
        driver.findElement(By.id("hello-input")).sendKeys("Charlie");
        driver.findElement(By.id("hello-submit")).click();
        try {

assertTrue(driver.getCurrentUrl().matches("^\\|\\|s\\|S]*http://uitest\\.duodecadits\\.com/hello\\|\\.html\\|\\|s\\|S]*myName=[\\|\\|s\\|S]*$"));
        } catch (Error e) {
            verificationErrors.append(e.toString());
        }
    }
}
```

```

    }
    assertEquals("Hello Charlie!", driver.findElement(By.id("hello-text")).getText());
    driver.get("http://uitest.duodecadits.com/form.html");
    driver.findElement(By.id("hello-input")).clear();
    driver.findElement(By.id("hello-input")).sendKeys("Emily");
    driver.findElement(By.id("hello-submit")).click();
    try {

assertTrue(driver.getCurrentUrl().matches("^\\s*$http://uitest\\.duodecadits\\.com/hello\\.html/\\s*$myName=[\\s\\S]*$"));
    } catch (Error e) {
        verificationErrors.append(e.toString());
    }
    assertEquals("Hello Emily!", driver.findElement(By.id("hello-text")).getText());
}

@After
public void tearDown() throws Exception {
    driver.quit();
    String verificationErrorString = verificationErrors.toString();
    if (!"".equals(verificationErrorString)) {
        fail(verificationErrorString);
    }
}

private boolean isElementPresent(By by) {
    try {
        driver.findElement(by);
        return true;
    } catch (NoSuchElementException e) {
        return false;
    }
}

private boolean isAlertPresent() {
    try {
        driver.switchTo().alert();
        return true;
    } catch (NoAlertPresentException e) {
        return false;
    }
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}

```

Result

- [info] Playing test case REQ-UI-12
- [info] Executing: |open | http://uitest.duodecadits.com/form.html ||
- [info] Executing: |type | id=hello-input | John |
- [info] Executing: |clickAndWait | id=hello-submit ||
- [info] Executing: |verifyLocation |
http://uitest.duodecadits.com/hello.html?myName= ||
- [info] Executing: |assertText | id=hello-text | Hello John! |
- [info] Executing: |open | http://uitest.duodecadits.com/form.html ||
- [info] Executing: |type | id=hello-input | Sophia |
- [info] Executing: |clickAndWait | id=hello-submit ||
- [info] Executing: |verifyLocation |
http://uitest.duodecadits.com/hello.html?myName= ||
- [info] Executing: |assertText | id=hello-text | Hello Sophia! |
- [info] Executing: |open | http://uitest.duodecadits.com/form.html ||
- [info] Executing: |type | id=hello-input | Charlie |
- [info] Executing: |clickAndWait | id=hello-submit ||
- [info] Executing: |verifyLocation |
http://uitest.duodecadits.com/hello.html?myName= ||
- [info] Executing: |assertText | id=hello-text | Hello Charlie! |
- [info] Executing: |open | http://uitest.duodecadits.com/form.html ||
- [info] Executing: |type | id=hello-input | Emily |
- [info] Executing: |clickAndWait | id=hello-submit ||
- [info] Executing: |verifyLocation |
http://uitest.duodecadits.com/hello.html?myName= ||
- [info] Executing: |assertText | id=hello-text | Hello Emily! |
- [info] Test case passed

Evaluation

Test case was successful.

Though there are concerns with the specification as it does not cover special characters, numbers, etc. That is the reason extra test case is added to cover this as well.

The screenshot displays the Selenium IDE 2.9.1 interface for a test suite named 'REQ-UI-12 (DOCLER TEST SUITE)'. The Base URL is set to 'http://uittest.duodecadits.com/'. The Test Case list on the left shows 'REQ-UI-12' as the current test case. The Command/Target/Value table in the center lists the following commands and their targets and values:

Command	Target	Value
assertText	id=hello-text	Hello Sophia!
open	http://uittest.duodecadits.com/form.html	
type	id=hello-input	Charlie
clickAndWait	id=hello-submit	
verifyLocation	*http://uittest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello Charlie!
open	http://uittest.duodecadits.com/form.html	
type	id=hello-input	Emily
clickAndWait	id=hello-submit	
verifyLocation	*http://uittest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello Emily!

The Log at the bottom shows the execution of the test case, including the following messages:

- [info] Executing: |clickAndWait | id=hello-submit | |
- [info] Executing: |verifyLocation | *http://uittest.duodecadits.com/hello.html?myName=* | |
- [info] Executing: |assertText | id=hello-text | Hello Charlie! |
- [info] Executing: |open | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |type | id=hello-input | Emily |
- [info] Executing: |clickAndWait | id=hello-submit | |
- [info] Executing: |verifyLocation | *http://uittest.duodecadits.com/hello.html?myName=* | |
- [info] Executing: |assertText | id=hello-text | Hello Emily! |
- [info] Test case passed

2.13. REQ-UI-12-additional-tests

Requirement specifcation:

It is not in the specification, covers extreme cases for REQ-UI-12

Basic flow:

Scope: basic functionality of input Form (extreme cases)

- [illegible]

Source code:

Selenium IDE

[illegible]

[illegible]

```
<td>{'&quot;!%_#&amp;@</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=hello-submit</td>
  <td></td>
</tr>
<tr>
  <td>verifyLocation</td>
  <td>*http://uitest.duodecadits.com/hello.html?myName=*</td>
  <td></td>
</tr>
<tr>
  <td>assertText</td>
  <td>id=hello-text</td>
  <td>Hello {'&quot;!%_#&amp;@!</td>
</tr>
</tbody></table>
</body>
</html>
```

JAVA

[illegible]

[illegible]

```
try {
    driver.switchTo().alert();
    return true;
} catch (NoAlertPresentException e) {
    return false;
}
}

private String closeAlertAndGetItsText() {
    try {
        Alert alert = driver.switchTo().alert();
        String alertText = alert.getText();
        if (acceptNextAlert) {
            alert.accept();
        } else {
            alert.dismiss();
        }
        return alertText;
    } finally {
        acceptNextAlert = true;
    }
}
}
```

Result

- [illegible]

- [info] Executing: |open | http://uittest.duodecadits.com/form.html | |
- [info] Executing: |type | id=hello-input | ÁáüĐđß |
- [info] Executing: |clickAndWait | id=hello-submit | |
- [info] Executing: |verifyLocation |
http://uittest.duodecadits.com/hello.html?myName= | |
- [info] Executing: |assertText | id=hello-text | Hello ÁáüĐđß! |
- [error] Actual value 'Hello
%C3%81%C3%A1%C3%A4%C5%B1%C4%90%C4%91%C3%9F!' did not match
'Hello ÁáüĐđß!'
- [info] Test case failed

Evaluation

Test case was failed at special characters. UTF-8 is not sufficient for characters like ÁáüĐđß, LATIN-1 is preferred in this case. Make note it is not part of the specification.

The screenshot shows the Selenium IDE 2.9.1 interface. The test case 'REQ-UI-12-additional-test-cases' is selected. The test steps are as follows:

Command	Target	Value
assertText	id=hello-text	Hello 1234!
open	http://uittest.duodecadits.com/form.html	
type	id=hello-input	AAAAAAAAAAAAAAAAAAAAAAAAAAAA...
clickAndWait	id=hello-submit	
verifyLocation	*http://uittest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello AAAAAAAAAAAAAAAAAAAAAA...
open	http://uittest.duodecadits.com/form.html	
type	id=hello-input	ÁáüĐđß
clickAndWait	id=hello-submit	
verifyLocation	*http://uittest.duodecadits.com/hello.html?my...	
assertText	id=hello-text	Hello ÁáüĐđß!

The test case failed at the final assertion. The log shows the following error:

```
[error] Actual value 'Hello %C3%81%C3%A1%C3%A4%C5%B1%C4%90%C4%91%C3%9F!' did not match 'Hello ÁáüĐđß!'
```

The log also shows the following steps:

```
[info] Executing: |open | http://uittest.duodecadits.com/form.html | |
[info] Executing: |type | id=hello-input | ÁáüĐđß |
[info] Executing: |clickAndWait | id=hello-submit | |
[info] Executing: |verifyLocation | *http://uittest.duodecadits.com/hello.html?myName=* | |
[info] Executing: |assertText | id=hello-text | Hello ÁáüĐđß! |
[info] Test case failed
```


3. Test summary

Test case	Input values	Expected outcome	Actual outcome	Result
REQ-UI-01	specified above	the given title is at all pages	failed at Error page	FAIL
REQ-UI-02	specified above	the given logo is at all pages	failed at Error page	FAIL
REQ-UI-03	specified above	Home button navigation is working	Home button is navigation is working	PASS
REQ-UI-04	specified above	Home button is active if pressed	Home button is active if pressed	PASS
REQ-UI-05	specified above	Form button navigation is working	Form button navigation is working	PASS
REQ-UI-06	specified above	Form button is active if pressed	Form button is active if pressed	PASS
REQ-UI-07	specified above	404 response code is present	404 response code is present	PASS
REQ-UI-08	specified above	logo should work as a home page button	logo is not functioning as a home page button	FAIL
REQ-UI-09	specified above	the given text is present at the home page	the given text is present at the home page	PASS
REQ-UI-10	specified above	the given text is present at the home page	the given text is present at the home page	PASS
REQ-UI-11	specified above	input box and submit button are present at the Form page	input box and submit button are present at the Form page	PASS
REQ-UI-12	specified above	basic functionality of input Form	input form gave back the expected result	PASS
REQ-UI-12-additional-tests	specified above	basic functionality of input Form (extreme cases)	failed at special characters	FAIL