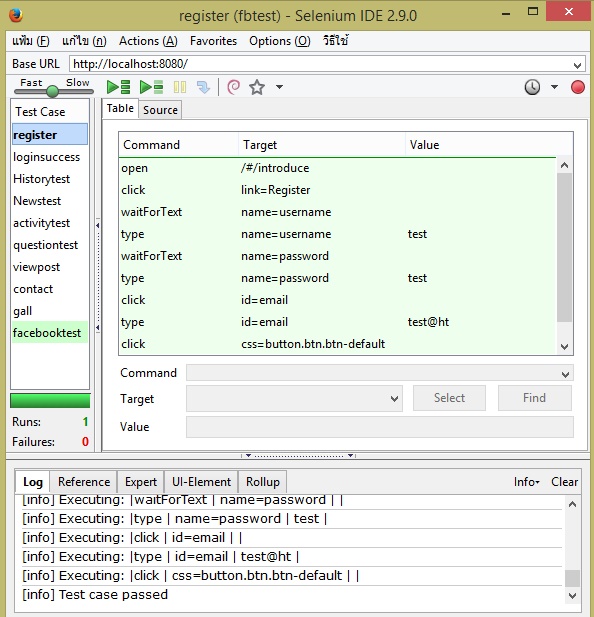
**Integration Testing: Using Selenium IDE**

**TC-01**

**User can register account with validate user name and password.**



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 Registercode {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testRegistercode() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.linkText("Register")).click();

for (int second = 0;; second++) {

if (second >= 60) fail("timeout");

try { if ("".equals(driver.findElement(By.name("username")).getText())) break; } catch (Exception e) {}

Thread.sleep(1000);

}

driver.findElement(By.name("username")).clear();

driver.findElement(By.name("username")).sendKeys("test");

for (int second = 0;; second++) {

if (second >= 60) fail("timeout");

try { if ("".equals(driver.findElement(By.name("password")).getText())) break; } catch (Exception e) {}

Thread.sleep(1000);

}

driver.findElement(By.name("password")).clear();

driver.findElement(By.name("password")).sendKeys("test");

driver.findElement(By.id("email")).click();

driver.findElement(By.id("email")).clear();

driver.findElement(By.id("email")).sendKeys("test@ht");

driver.findElement(By.cssSelector("button.btn.btn-default")).click();

}

@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;

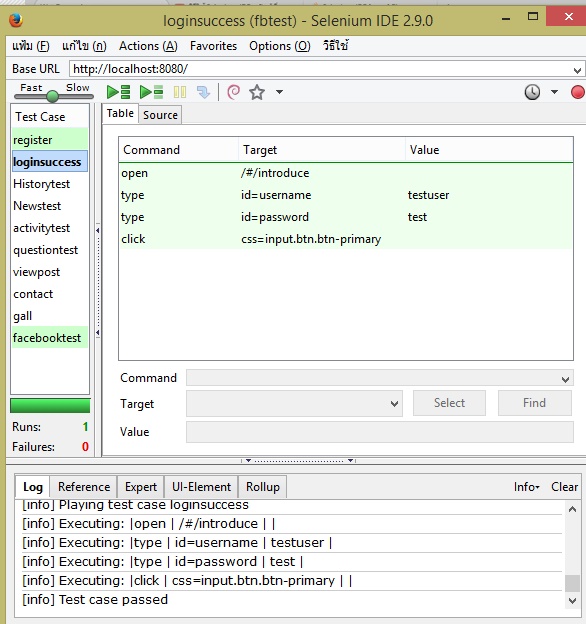
}

}

}

**TC-02**

**User can login to the website as a member.**



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 Loginsuccess2 {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testLoginsuccess2() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

}

@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;

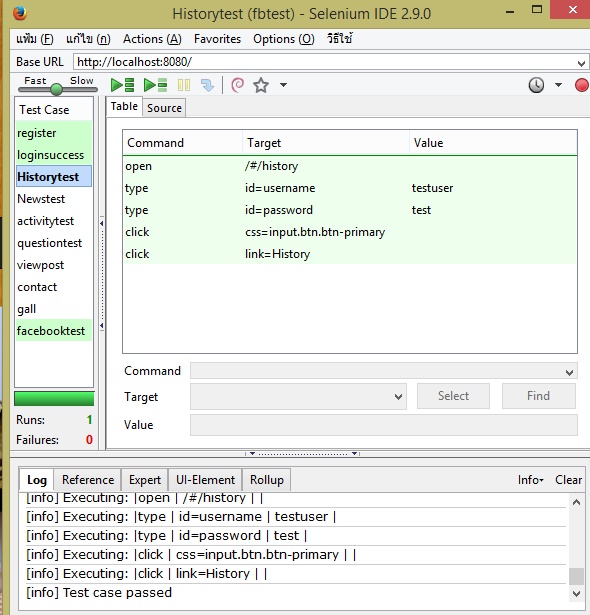
}

}

}

**TC-03**

**User can read history of temple**



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 History {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testHistory() throws Exception {

driver.get(baseUrl + "/#/history");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("History")).click();

}

@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;

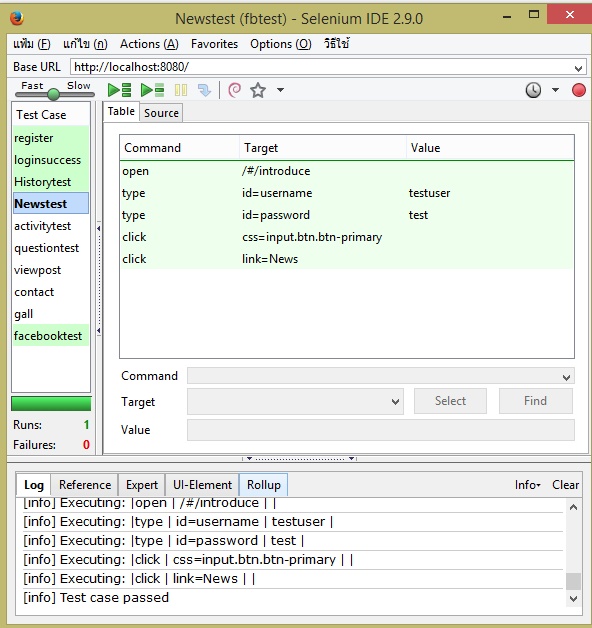
}

}

}

**TC-04**

**User can read news of the temple**



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 Neescode {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testNeescode() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("News")).click();

}

@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;

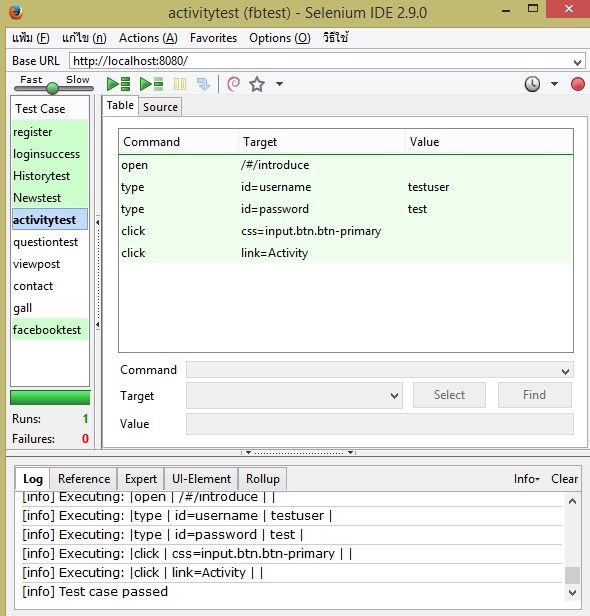
}

}

}

**TC-05**

**User can view activities of the temple**



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 Activity {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testActivity() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("Activity")).click();

}

@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;

}

}

}

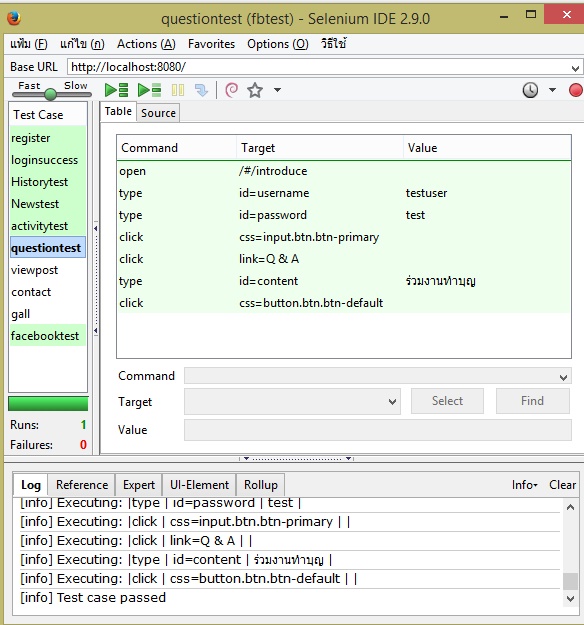
**TC-06**

**User can receive the news and activities via email.**

**-**

**TC-07**

**User can post question to ask the guru.**



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 Questioncodee {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testQuestioncodee() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("Q & A")).click();

driver.findElement(By.id("content")).clear();

driver.findElement(By.id("content")).sendKeys("เธฃเนเธงเธกเธเธฒเธเธ—เธณเธเธธเธ");

driver.findElement(By.cssSelector("button.btn.btn-default")).click();

}

@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;

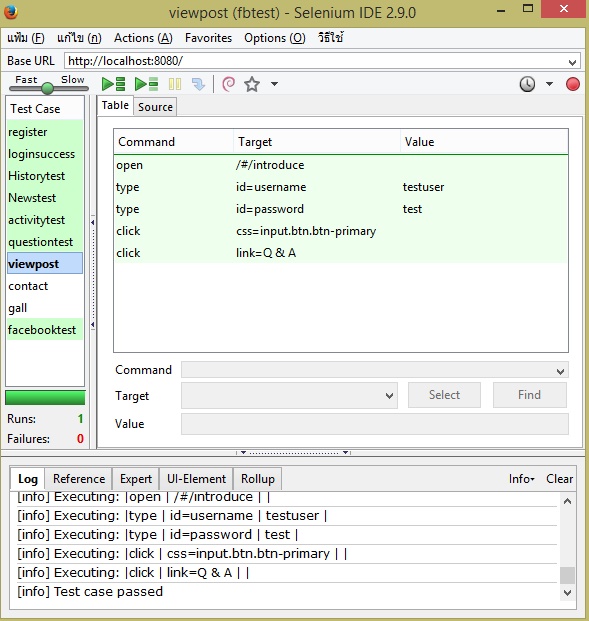
}

}

}

**TC-08**

**User can view others’ posts.**



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 Viewotherpostcode {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testViewotherpostcode() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("Q & A")).click();

}

@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;

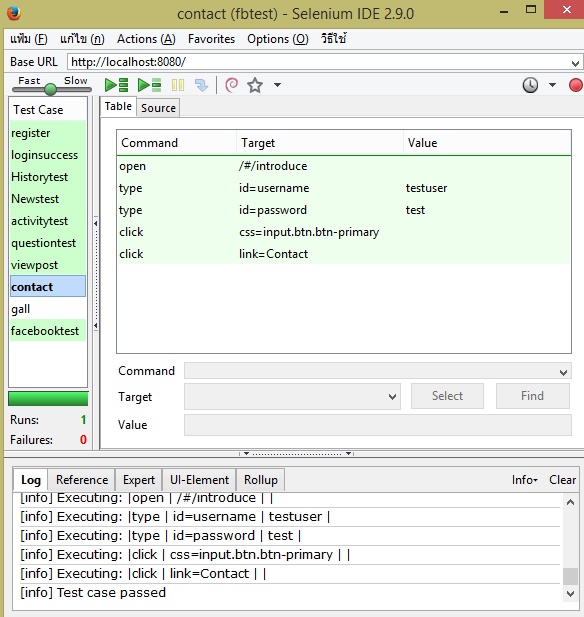
}

}

}

**TC-09**

**User can view contact details.**



package com.example.tests;

import com.thoughtworks.selenium.\*;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

import java.util.regex.Pattern;

public class contactcode {

private Selenium selenium;

@Before

public void setUp() throws Exception {

selenium = new DefaultSelenium("localhost", 4444, "\*chrome", "http://localhost:8080/");

selenium.start();

}

@Test

public void testContactcode() throws Exception {

selenium.open("/#/introduce");

selenium.type("id=username", "testuser");

selenium.type("id=password", "test");

selenium.click("css=input.btn.btn-primary");

selenium.click("link=Contact");

}

@After

public void tearDown() throws Exception {

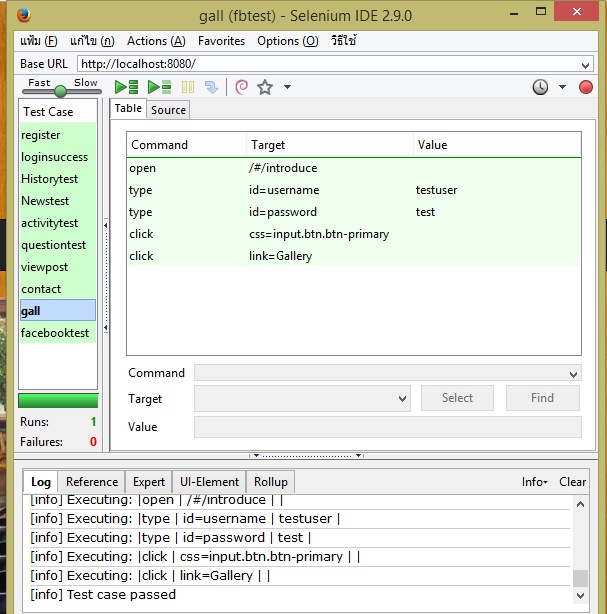
selenium.stop();

}

}

**TC-10**

**User can see pictures in gallery.**



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 Galll {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testGalll() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("Gallery")).click();

}

@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;

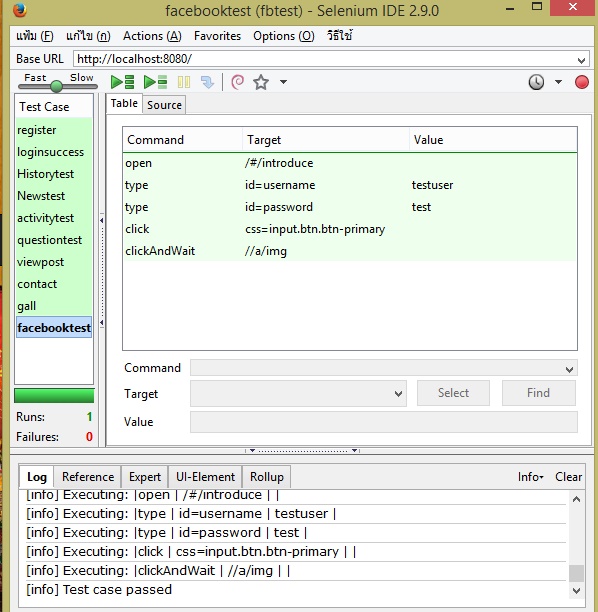
}

}

}

**TC-11**

**User can open temple’s Facebook page.**



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 Facebooktest {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testFacebook() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.xpath("//a/img")).click();

}

@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;

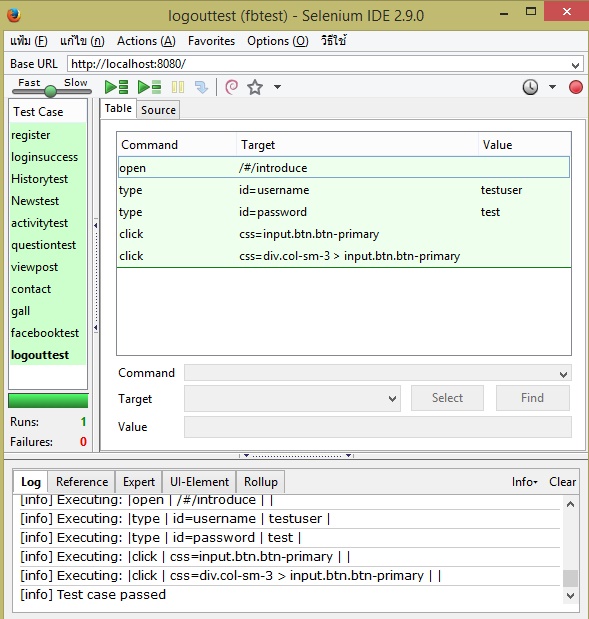
}

}

}

**TC-12**

**User can logout their account.**



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 Logcode {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testLogcode() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("testuser");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("test");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.cssSelector("div.col-sm-3 > input.btn.btn-primary")).click();

}

@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;

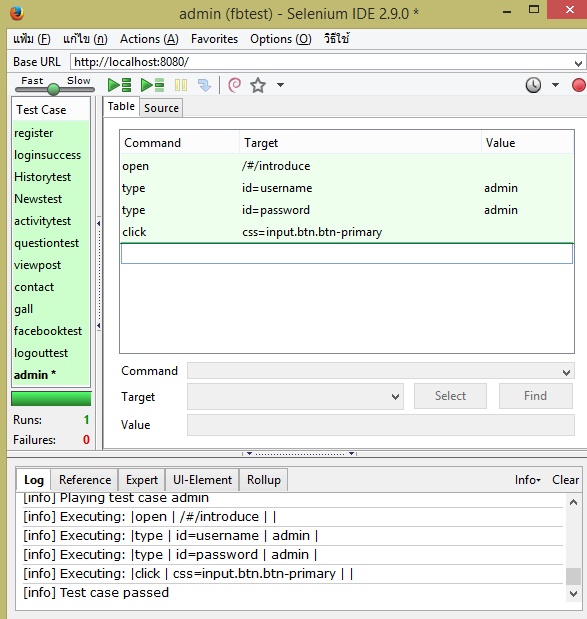
}

}

}

**TC-13**

**Admin can login as administrator.**



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 13 {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void test13() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("admin");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("admin");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

}

@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;

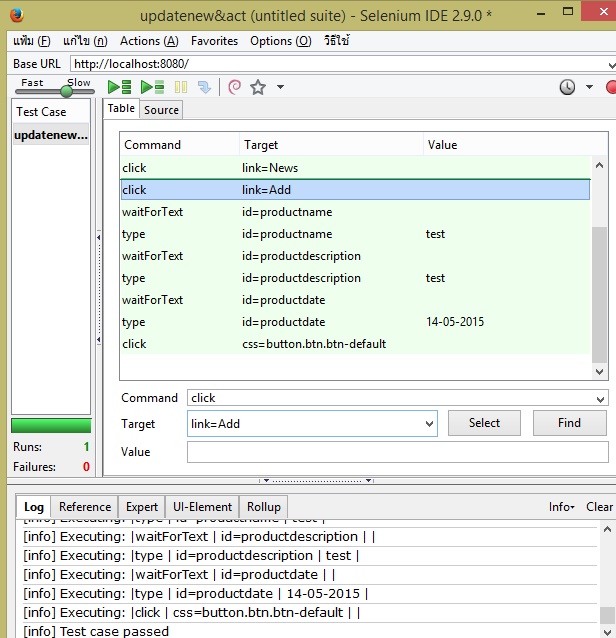
}

}

}

**TC-14**

**Admin can update news/activities.**



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 Update {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testUpdate() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("admin");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("admin");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("News")).click();

driver.findElement(By.linkText("Add")).click();

for (int second = 0;; second++) {

if (second >= 60) fail("timeout");

try { if ("".equals(driver.findElement(By.id("productname")).getText())) break; } catch (Exception e) {}

Thread.sleep(1000);

}

driver.findElement(By.id("productname")).clear();

driver.findElement(By.id("productname")).sendKeys("test");

for (int second = 0;; second++) {

if (second >= 60) fail("timeout");

try { if ("".equals(driver.findElement(By.id("productdescription")).getText())) break; } catch (Exception e) {}

Thread.sleep(1000);

}

driver.findElement(By.id("productdescription")).clear();

driver.findElement(By.id("productdescription")).sendKeys("test");

for (int second = 0;; second++) {

if (second >= 60) fail("timeout");

try { if ("".equals(driver.findElement(By.id("productdate")).getText())) break; } catch (Exception e) {}

Thread.sleep(1000);

}

driver.findElement(By.id("productdate")).clear();

driver.findElement(By.id("productdate")).sendKeys("14-05-2015");

driver.findElement(By.cssSelector("button.btn.btn-default")).click();

}

@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;

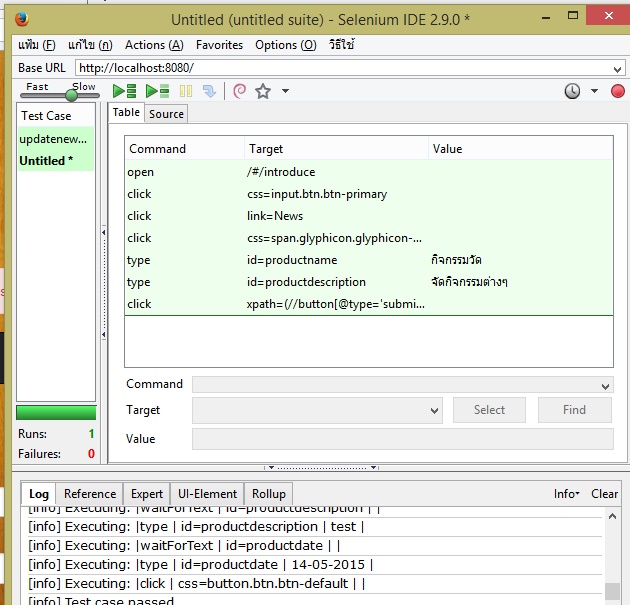
}

}

}

**TC-15**

**Admin can edit news/activities.**



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 Editcode {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void testEditcode() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("News")).click();

driver.findElement(By.cssSelector("span.glyphicon.glyphicon-edit")).click();

driver.findElement(By.id("productname")).clear();

driver.findElement(By.id("productname")).sendKeys("กิจกรรมวัด");

driver.findElement(By.id("productdescription")).clear();

driver.findElement(By.id("productdescription")).sendKeys("จัดกิจกรรมต่างๆ");

driver.findElement(By.xpath("(//button[@type='submit'])[2]")).click();

}

@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;

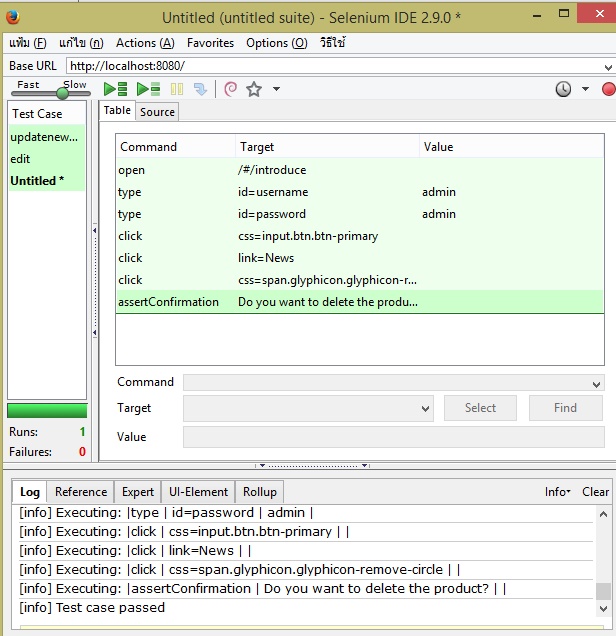
}

}

}

**TC-16**

**Admin can delete news/activities.**



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 166 {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void test166() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("admin");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("admin");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("News")).click();

driver.findElement(By.cssSelector("span.glyphicon.glyphicon-remove-circle")).click();

assertTrue(closeAlertAndGetItsText().matches("^Do you want to delete the product[\\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;

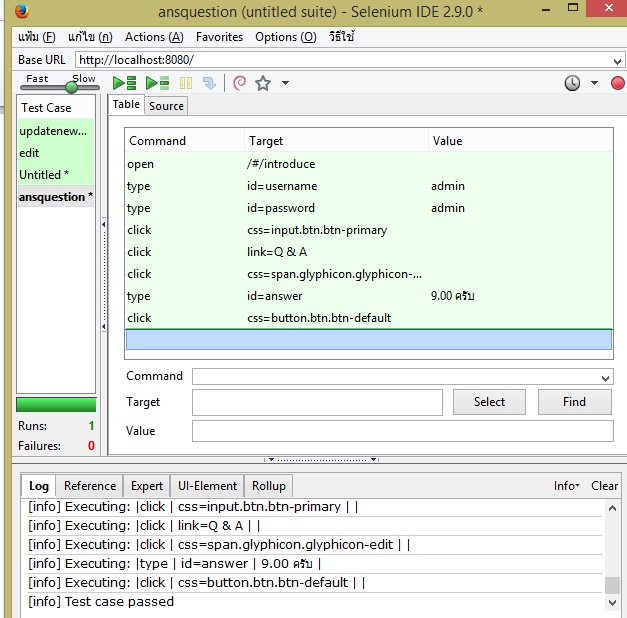
}

}

}

**TC-17**

**Admin can answer the question from member.**



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 177 {

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://localhost:8080/";

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@Test

public void test177() throws Exception {

driver.get(baseUrl + "/#/introduce");

driver.findElement(By.id("username")).clear();

driver.findElement(By.id("username")).sendKeys("admin");

driver.findElement(By.id("password")).clear();

driver.findElement(By.id("password")).sendKeys("admin");

driver.findElement(By.cssSelector("input.btn.btn-primary")).click();

driver.findElement(By.linkText("Q & A")).click();

driver.findElement(By.cssSelector("span.glyphicon.glyphicon-edit")).click();

driver.findElement(By.id("answer")).clear();

driver.findElement(By.id("answer")).sendKeys("9.00 ครับ");

driver.findElement(By.cssSelector("button.btn.btn-default")).click();

}

@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;

}

}

}