UITestFramework

How to write tests for Reference Application UI Framework

Overview

The beginnings of a UI Test Framework are now in place. The project is here: https://github.com/openmrs/openmrs-contrib-uitestframework. The framework supports execution of tests that drive an OpenMRS installation via its (browser) User Interface. It uses Selenium to drive the UI and do validation. It uses REST to populate the database before a test (or a group of tests). It uses DbUnit (via JDBC) to clean up the database after. It relies on JUnit 5 annotations (@Before, @BeforeClass, @After). It also supports several ways to specify test settings such as the URL of the OpenMRS service.

Use of the framework can be seen in the Reference App tests, found here: https://github.com/openmrs/openmrs-distro-referenceapplication/tree/master/ui-tests.

Usage Guide

How to launch tests

Notes

You can use http://int-refapp.openmrs.org/ (or others sites) instead of localhost, if you cant or dont want to use Vagrant (https://wiki.openmrs.org/x/CIC3Ag)

You can use Chrome browser instead of Firefox browser. New versions of Firefox work incorrectly on Windows 7 and 8, so you must use old Firefox version, e.g. 15. Before committing not forget to ignore pom.xml

How to write tests

We use VisitTest as an example test (https://github.com/openmrs/openmrs-distro-referenceapplication/blob/master/ui-tests/src/test/java/org/openmrs/reference/StartVisitTest.java) and PatientDashboardPage as an example page (https://github.com/openmrs/openmrs-distro-referenceapplication/blob/master/ui-tests/src/test/java/org/openmrs/reference/page/HomePage.java)

Tests are basically composed of:

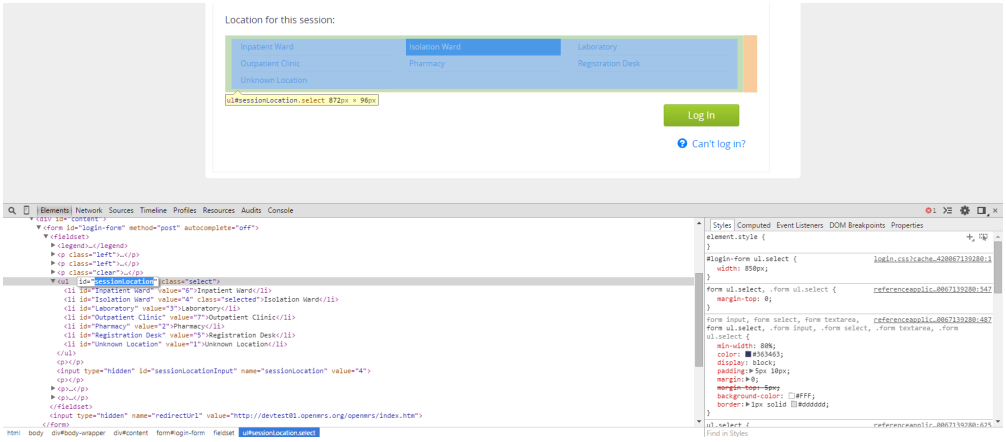
- one test class which is a subclass of org.openmrs.uitestframework.test.TestBase. See VisitTest.
- several "Page" classes which are subclasses of org.openmrs.uitestframework.page.AbstractBasePage which implements the Page interface. See PatientDashboardPage

Note that Page classes are used in several tests. They can be updated and get more power
Tests usually have an **@Before** method that creates an instance of each Page class the test needs, as well as creating test data, such as a Patient. This method is called once before each test.

You can also create an **@After** method that is called after each test.

Recommended repetitive algorithms (login, logout) at the beginning and at the end of all the tests transfer to methods **@Before/@After** to avoid repetition code
It is also important not to use selenium in the Test classes, but rather in Page classes.

How to take an element from the example html page



We can take an element from page by his id. Example code for this table:

```
private static final By LOCATION_LIST = By.id("sessionLocation");
```

How to take the List of "li" elements from table

Example code for this table:

```
 WebElement locationList = driver.findElement(LOCATION_LIST);

 List<WebElement> locationCollection = locationList.findElements(By.xpath("id('sessionLocation')/li"));
```

locationCollection is List of WebElements - "Inpatient Ward" button, "Isolation Ward" button, etc.

How to get text from WebElement (locationCollection)

Example code:

```
return locationCollection.get(index).getText();
```

Where "index" is index of table element

How to click on WebElement (locationCollection)

Example code:

```
locationCollection.get(index).click();
```

Where "index" is index of table element

This code is taken from https://github.com/Izaron/openmrs-distro-referenceapplication/blob/gci-4-3/ui-tests/src/main/java/org/openmrs/reference/page/LocationPage.java

And much more!

We can fill the text in the input forms, press the buttons, move to different pages, launch JS code. Use Selenium and example tests!

How do get an idea for a test

You just have to test things that no one has still tested. For example, you can test the registration of the patient, changing the visit options, etc. Of course, you can use several classes of Pages, update these classes, or even create new Page class
It is desirable that your tests covered all the functionality at Reference Application