

Имена: Калина Бухлева

фн: 61681

Начална година: 2013

Програма: бакалавър, (СИ) **Курс:** 4

Тема: Създаване на тестови сценарии и тестване на puffin посредством Java/Selenium WebDriver

Дата: 2017-05-24

Предмет: wwwTech2017_18_8ed_SI_winter

имейл: kalinastbuhleva@gmail.com

преподавател: доц. д-р Милен Петров

Предаване: Задачата се предава в архив с попълнен настоящия документ, проекта/проектите с кодовете, README.txt файл, който описва съдържанието на архива; папка с допълнителни компоненти и използвани ресурси, архива да се казва [wwwTech2017_8ed_SI_fn9999_project.zip](#). (Успех!). (Редактирайте **маркираните зони** в жълто с коректната информация)

ТЕМА: Създаване на тестови сценарии и тестване на puffin посредством Java/Selenium WebDriver

1. Условие

Уеб базирано приложение за дефиниране на тестови сценарии с примери за puffin

Приложението трябва създава проекти с тестове и да ги изпълнява. Всеки може да добави проект. Всеки проект има свои функционални изисквания, които трябва лесно да се добавят от потребителя. Потребителят трябва да опише функционалните изисквания стъпка по стъпка и напише необходимата имплементация за тяхното тестване. Щом функционалното изискване е запаметено, то може да бъде променяно и изпълнявано. За всяко изпълнение се връща резултат.

2. Въведение

Selenium Web Driver е инструмент за писане на автоматизирани тестове за уеб страници. То имитира действията на истински потребител. За да улесни тестването приложението ще пропусне досадната част по подготовка на драйвъра като предлага на потребителя готов драйвър, който сам стартира преди теста и се изключва след него. Приложението е интегрирано и с Cucumber-JVM налагайки BDD разработка.

3. Теория

Един типичен бизнес проект би започнал с даването на примерни сценарии на това, което трябва да прави системата от заинтересованите лица. Всички усилия се хвърлят в изпълнението на тези сценарии. Примерите се превръщат в критерия за валидация посредством автоматизирани тестове. Резултатът от тези тестове уверява заинтересованите лица, че желаните изисквания са покрити. В идеалния случай, доклада от тестовете е лесен за разбиране от всеки. BDD ни позволява да "съживим" документацията, като тя може да бъде променяна в процеса на тестване.

На практика, behavior-driven development е подобно на test-driven development, където всички заинтересовани лица имат знания и умения за програмиране. Но това не винаги е ситуацията в една организация, BDD позволява да включим по голяма част от заинтересованите лица и потребители, които имат малък или никакъв опит с програмиране.

4. Използвани технологии

Selenium Web Driver - инструмент за писане на автоматизирани тестове за уеб страници. То имитира действията на истински потребител.

Cucumber - софтуерен инструмент за тестване на софтуер. Изпълнява тестове написани в BDD стил.

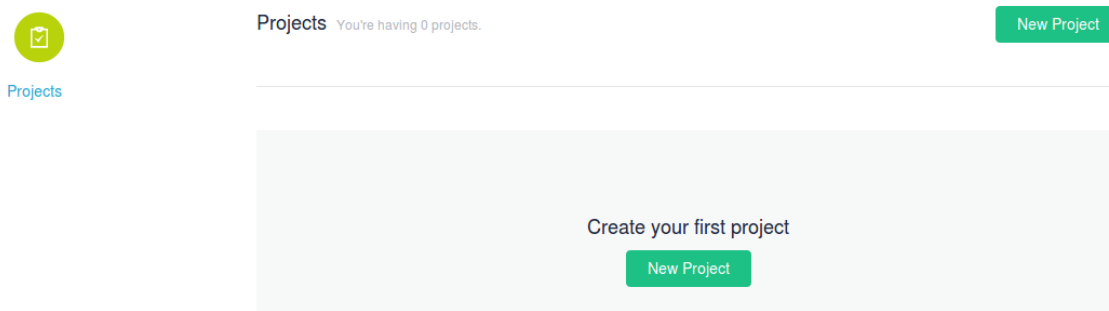
5. Инсталация и настройки

Приложението представлява стандартен **maven** проект, за да стартирате в **develop mode** и да се използва стандартната база от данни. Първо проекта трябва да се компилира и да се стартира плъгина, които ще свали необходимите драйвъри с **mvn install**. В директорията target/test-classes/selenium-standalone-binaries ще се разпаковат необходимите файлове. Файлът application-config.properties трябва да бъде редактиран така, че да сочи към phantomjs файла, свален преди малко. Стартирайте приложението с **jetty:run**.

За **production mode** е нужно да редактирате application-config.properties и да създадете .war файл с **mvn install**.

6. Кратко ръководство на потребителя

Стартирайте със създаването на проект. Избирайки Create Project от главната страница.



При успешно попълване на появилата се форма, ще видиш новия си проект на главната страница.



Projects

Projects You're having 1 projects.

New Project



Puffin

Избирайки името на проекта ще отвориш страница подобна на главната, където можеш да създадеш нов feature.

Всеки feature има заглавие и описание.

New Feature

Save Feature

Title e.g. "Login"

Login

Description e.g. "Any register user can log in"

Any register user can log in

Последван от сценарий за неговото тестване.

Scenario Title e.g. "Login with valid credentials"

Login with valid credentials

Step

When

e.g. "I am a registered user"

I open login page

```
driver.get("http://loremipsum2.fmi.uni-sofia.bg/WEBTECH/www_8ed_referats/");
```

Add Step

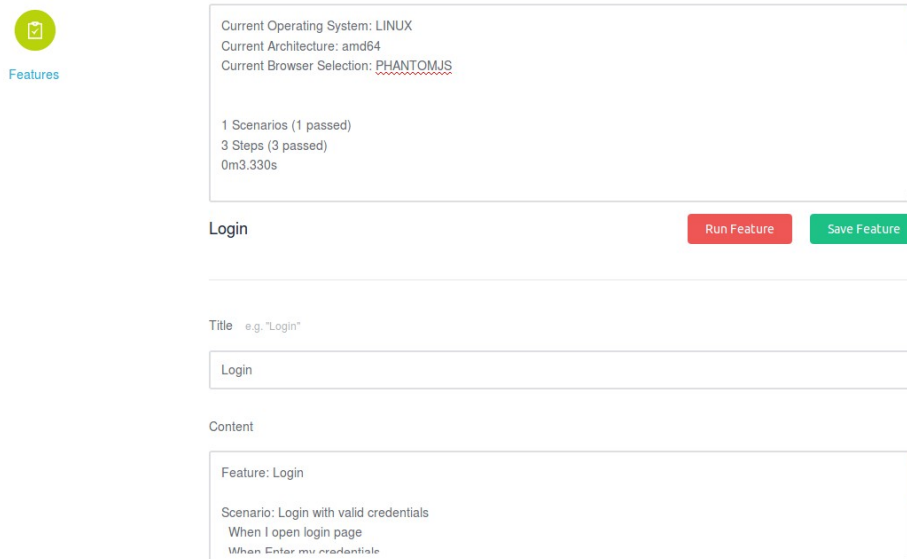
Remove Step

Step

When

e.g. "I am a registered user"

Всеки сценарии може да бъде изпълнен изпирайки името му от старницата на проекта и натискайки Run Feature. Същата страница позволява и модификация на теста.



The screenshot shows the Cucumber Features page. On the left, there is a green circular icon with a white checkmark and the word "Features" below it. The main content area is a scrollable list of features. The first feature is "Login". It shows the following details:

- Current Operating System: LINUX
- Current Architecture: amd64
- Current Browser Selection: PHANTOMJS
- 1 Scenarios (1 passed)
- 3 Steps (3 passed)
- 0m3.330s

Below the list, there are two buttons: "Run Feature" (red) and "Save Feature" (green). Below the buttons, there is a form for editing the feature. The "Title" field contains "e.g. 'Login'". The "Content" field contains the following text:

```
Feature: Login
Scenario: Login with valid credentials
  When I open login page
  When Enter my credential
```

7. Примерни данни

```
package web.technology.selenium.framework.tests
```

```
import cucumber.api.java.Before
import cucumber.api.java.en.Then
import cucumber.api.java.en.When
import org.openqa.selenium.By
import org.openqa.selenium.WebDriver
import org.testng.annotations.Test
import web.technology.selenium.framework.DriverBase
```

```
import java.util.concurrent.TimeUnit
```

```
@Test(groups = "cucumber")
public class TestSteps extends DriverBase {
```

```
    private WebDriver driver;
```

```
    @Before
    public void setup() {
```

```

        driver = getDriver();
    }

    @When("^I open login page\\$")
    public void i_open_login_page() throws Throwable {
        driver.get("http://loremipsum2.fmi.uni-
sofia.bg/WEBTECH/www_8ed_referats/");
    }

    @When("^I enter my credentials\\$")
    public void enter_my_credentials() throws Throwable {
        driver.findElement(By.name("name")).sendKeys("user");
        driver.findElement(By.name("pass")).sendKeys("test");
        driver.findElement(By.className("green")).click();
    }

    @Then("^I can see main page\\$")
    public void can_see_main_page() throws Throwable {
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
        driver.findElement(By.xpath("//h2")).isDisplayed();
    }
}

```

8. Описание на програмния код

Приложението има MVC структура. Моделите са : Project, UFTFeature, UFTScenario, UFTStep. Има две таблици за проектите и за feature-ите. Един контролер и множество view-та, които видяхме по-горе. Два service-a изпълняват бизнес логиката съответно за проектите и тестовете.

9. Приноси на студента, ограничения и възможности за бъдещо разширение

Използва се само един ClassLoader при изпълнение на тестовете, което пречи на паралелното им изпълнение.

10. Какво научих

Как да заредя програмно динамично създаден клас през URLClassLoader.

11. Използвани източници

[1] Selenium Web Driver - <http://www.seleniumhq.org/projects/webdriver/>

[2] Cucumber - <https://cucumber.io/>

Предал:

/фн, имена, спец., група/

Приел:

/доц. Милен Петров/