

Автоматизиране на тестването

Писане на тест кейси за cucumber

Автоматизирано тестване

Най-често се автоматизират тестовете,
КОИТО:

- ♦ Често се повтарят – Regression tests, Smoke tests, Sanity Check.
- ♦ Трудни за ръчно изпълнение – Performance tests, Load tests , Stress tests.



Митове за автоматизирането

- ♦ Автоматизираните тестове могат да заместят хората.
- ♦ Веднъж направена, от автоматизацията само се печели, без ре-инвестиция.
- ♦ Всеки тест може да бъде автоматизиран.
- ♦ Един инструмент за автоматизация е подходящ за всички видове тестове.



Цели на автоматизирането

Тестовете трябва да:

- ♦ Помогнат да подобрим качеството.
- ♦ Намалят риска.
- ♦ Лесни за изпълняване.
- ♦ Лесни за писане.



“Quality is free, but only to those
who are willing to pay heavily for it.”

- Lister, Demarco: “Peopleware”

Добри практики

- ♦ Пишете кратки тестове – колкото по-дълъг е теста толкова повече усилия ще изисква.
- ♦ Пишете тестове, които са независими един от друг. Не трябва да има тестове зависими от други тестове, или от техни данни.
- ♦ Тестовите трябва да могат да бъдат многократно изпълнявани.

Инструменти за автоматизация

- ♦ Cucumber.
- ♦ Selenium IDE.
- ♦ Selenium WebDriver.
- ♦ Soap UI – за тестване на SOAP базирани веб сървиси.
- ♦ JMeter – за performance, load и stress tests.
- ♦ NUnit, JUnit, PyUnit и др – за unit tests.

Automated Acceptance tests

- ♦ Идеята за такива тестове се появява при Agile методологията, и по точно Test-Driven Development.
- ♦ Тези тестове изразяват, какво трябва да прави софтуера за да бъде “acceptable”.
- ♦ Тестовите се пишат преди кода, и фейлват докато не се напише правилно кода.
- ♦ Acceptance тестовите са различни от Unit тестовите.

Behaviour Driven Development

- ♦ Behaviour Driven Development, се базира на Test Driven Development, като разширява тестовете добавяйки бизнес стойност към тях. Тоест тестваме бизнес приложението на кода, докато го пишем.
- ♦ BDD практиките изискват тестовете да бъдат написани, така че всеки в екипа да може да ги прочете и разбере.

Жива документация

- ♦ Всеки, включително и бизнес хората в екипа, могат да разберат и да редактират теста.
- ♦ Софтуер може, да прочете и да изпълни теста веднага.

Feature: Sign up

Sign up should be quick and friendly.

Scenario: Successful sign up

New users should get a confirmation email and be greeted personally by the site once signed in.

Given I have chosen to sign up

When I sign up with valid details

Then I should receive a confirmation email

And I should see a personalized greeting message

Scenario: Duplicate email

Where someone tries to create an account for an email address that already exists.

Given I have chosen to sign up

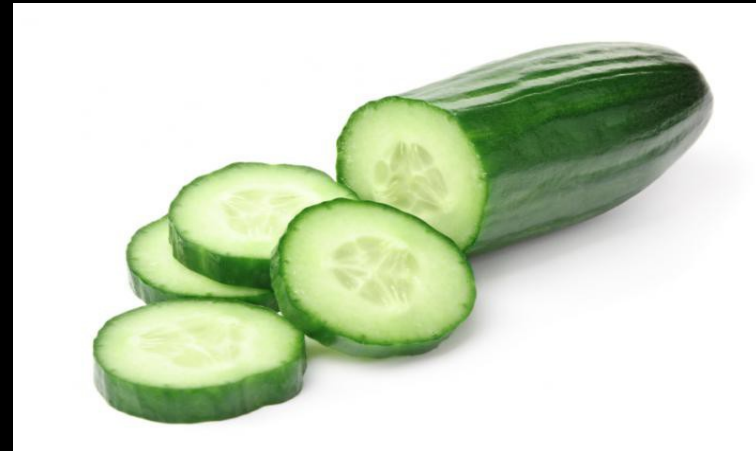
But I enter an email address that has already registered

Then I should be told that the email is already registered

And I should be offered the option to recover my password

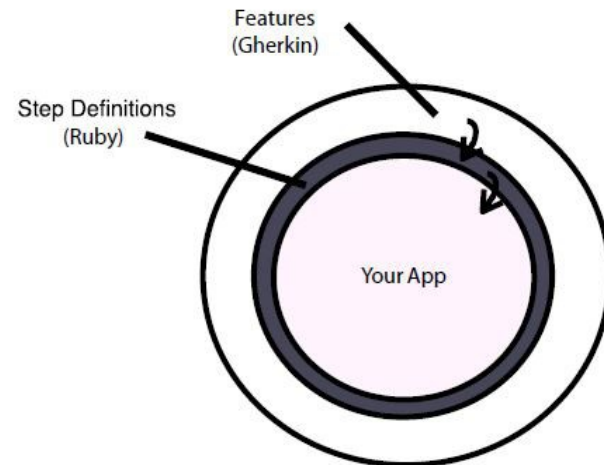
Cucumber

- ♦ Cucumber е софтуерен инструмент, който се използва за тестване на софтуерни продукти.
- ♦ Изпълнява acceptance tests, написани в специален (Gherkin) синтаксис.
- ♦ Cucumber е написан на езика Ruby, но може да се използва със много други езици, като Java.



Как работи Cucumber?

- ♦ Cucumber е софтуерен продукт, който трябва да се инсталира на компютър.
- ♦ Използва Gherkin синтаксиса за да разбира тест кейсите.
- ♦ Използва програмен език, за да дефинира стъпките и да изпълни тестовте.



Какво е Gherking синтаксис?

- ♦ Gherkin е набор от ключови думи, за писане разбираеми acceptance тестове.
- ♦ Gherkin прави тестовете лесни за разбиране от програмисти, QA и бизнес представители (stakeholders).
- ♦ Пример:
<https://github.com/yahmedov/gherkin/blob/master/sample.feature>

Формат и Синтаксис на Gherking

- ♦ Gherkin файловете използват .feature разширението. И се записват като обикновен текстов файл.
- ♦ Ключови думи – Feature, Background, Scenario, Given, When, Then, And, But, *, Scenario Outline, Examples.

Ключови думи в Gherkin

- ♦ Feature – всеки gherkin файл започва с тази ключова дума, тя не променя поведението на теста само обозначава къде може да се запише заглавие и описание на фийчър.
- ♦ Във валиден gherkin файл следващата ключова дума след Feature, трябва да бъде Scenario, Background или Scenario Outline.

Ключови думи в Gherkin

- ♦ Scenario – всеки фийчър има по няколко, сценария които дават кратък пример (тест) за това как трябва да работи системата.
- ♦ Всеки сценарий е един конкретен тест.
- ♦ Ако системата покрие изискванията на теста този сценарий ще е “pass”, ако не ги покрие сценарият ще е “fail”.

Ключови думи в Gherkin

- ♦ Given, When и Then – използваме “Given” за да напомним контекста на средата в която се тества, “When” за да извършим някаква интеракция със системата, и “Then” за да проверим изходния резултат.
- ♦ Пример:

```
Scenario: Successful withdrawal from an account in credit  
  Given I have $100 in my account # the context  
  When I request $20               # the event(s)  
  Then $20 should be dispensed    # the outcome(s)
```

Ключови думи в Gherkin

- ♦ And и But – може да добавим допълнителни стъпки, към всяка стъпка от сценария.
- ♦ За Cucumber няма разлика, коя точно ключова дума сте използвали.
- ♦ Пример:
https://github.com/yahmedov/gherkin/blob/master/and_but.feature

Ключови думи в Gherkin

- ♦ Можем да заместим Given/When/Then със астериск “*”.
- ♦ За Cucumber смисъла на теста е все още същия, но е по труден за разбиране от хора.
- ♦ Пример:

```
Scenario: Attempt withdrawal using stolen card
* I have $100 in my account
* my card is invalid
* I request $50
* my card should not be returned
* I should be told to contact the bank
```

Ключови думи в Gherkin

- ♦ Background – във фийчър в който сценариите имат общи стъпки, общите стъпки могат да се запишат отделно, за да няма повтаряне на стъпки.
- ♦ Пример:
<https://github.com/yahmedov/gherkin/blob/master/background.feature>

Ключови думи в Gherkin

- ♦ Scenario Outline – когато искаме да изпълним един тест многократно, но с различни тестови данни.
- ♦ Записваме тестовите стъпки в Scenario Outline и задаваме стойности в Examples.
- ♦ Пример:
https://github.com/yahmedov/gherkin/blob/master/scenario_outline.feature

Въпроси



Задача за домашна работа

Разпишете Cucumber тестове, използвайки Gherkin синтаксиса, за Regression тестване на основната функционалност на QA test site: <http://qa.soft-intellect.com/test/>
!Разделете тестовете във файлове по фийчъри, и ги качете в github.