

Cucumber testing – Cum un castravete te ajută cu testarea

Referat Individual

Bălescu Alexandru

Grupa 344

În primul rând, ce implică Cucumber testing?

Cucumber este o unealtă software care ajută cu Behavior Driven Development.

Ce problemă rezolvă testarea cu Cucumber?

În majoritatea cazurilor, testele sunt greu de citit de către persoanele care nu sunt prea experimentate în domeniu. Tocmai de aceea, folosind Cucumber și scriind cod folosind Gherkin, testele pot fi scrise în așa mod încât să fie citite într-un limbaj cât mai natural. Astfel, testele scrise folosind Cucumber devin o dovadă concretă a progresului și o garanție a calității pe care o are produsul software. Aceste teste pot fi prezentate facil, eficient și citeț unui public care nu are un fundament tehnic.

Cum funcționează Cucumber?

Cucumber se bazează pe scenarii. Un scenariu este o listă de pași pe care Cucumber îi verifică. Ca Cucumber să înțeleagă scenariile, acestea trebuie scrise folosind Gherkin. Gherkin se bazează pe cuvinte cheie. Acestea se află la fiecare început de linie: Scenario, Example, Given, When, And, Then etc.. Fiecare pas este reprezentat de un statement, acest statement corespunde unei funcții. Această asociere se realizează de obicei prin anotări.

Trade-offs:

Acest tip de testare, chiar dacă facilitează înțelegerea de teste scrise pentru cei care nu au background tehnic, face scrierea de teste mai anevoioasă și mai time-consuming pentru developeri. Totuși, un dezvoltator software bun ar trebui să se concentreze să scrie cod de calitate, citeț și scalabil.

Opinie personală:

Consider că acest tip de testare are sens în cazul testelor mai ample, care imită comportamentul unui user. Testarea unitară este foarte eficientă pentru a te asigura că o bucată de cod funcționează as intended. Fiecare tip de testare își are locul și doar pentru că poți să folosești un singur tip de testare, nu înseamnă că e și recomandat.

Exemplu:

Feature: Is it Friday yet?

Everybody wants to know when it's Friday

Scenario: Sunday isn't Friday

Given today is Sunday

When I ask whether it's Friday yet

Then I should be told "Nope"

Code:

```
package hellocucumber;

import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
import static org.junit.jupiter.api.Assertions.*;

class IsItFriday {
    static String isItFriday(String today) {
        return null;
    }
}

public class Stepdefs {
    private String today;
    private String actualAnswer;

    @Given("today is Sunday")
    public void today_is_Sunday() {
        today = "Sunday";
    }

    @When("I ask whether it's Friday yet")
    public void i_ask_whether_it_s_Friday_yet() {
        actualAnswer = IsItFriday.isItFriday(today);
    }

    @Then("I should be told {string}")
    public void i_should_be_told(String expectedAnswer) {
        assertEquals(expectedAnswer, actualAnswer);
    }
}
```

Referinte:

<https://cucumber.io/docs/cucumber/>