

Тест-кейсы в коде

или

где хранить тест-кейсы автоматизатору

Ольга Ким

QIWI

Как хранить тест-кейсы

- Excel или Google Docs
- Test Case Management Tool
- BDD
- Другие варианты?

Excel или Google Docs

Плюсы:

- Просто записывать

Минусы:

- Тяжело поддерживать
- Неудобно использовать



Test Case Management Tool

Invoicing (1)

<input type="checkbox"/> ID	Title
<input type="checkbox"/> T2954...	Footer формы

Создание счёта (6)

<input type="checkbox"/> ID	Title
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметром email
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметром lifetime
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметром comment
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметром billId/bill_id
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметром extra/customField[]/extras
<input type="checkbox"/> T2954...	Выставление счёта на оплату с параметрами success_url и fail_url (пока не готов)

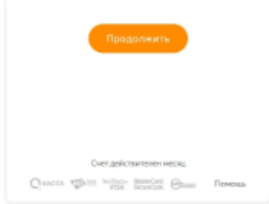
Test Case Management Tool

T2954670 Footer формы ↓

Invoicing: полное регрессионное тестирование > Web version > Invoicing

Type Functionality	Priority 2 - Test If Time	Estimate None	Milestone None
References None			

Steps

Step	Expected Result
1. Открыть любой счёт с формой и проверить содержимое футера	<ol style="list-style-type: none">1. Выводится срок действия счёта2. Логотипы партнёров3. Ссылка на страницу помощи 
2. Нажать на <i>Помощь</i>	Осуществляется переход на https://qiwi.com/support.action

Results & Comments ?

History & Context ?

Defects ?

Add a comment ..

Test Case Management Tool

Плюсы:

- Легко поддерживать
- Удобно отмечать результаты прогона
- Интеграция с баг-трекером

Но ведь на проекте только автоматизаторы...



BDD Scenarios. Gherkin

Feature: Calculator

Scenario Outline: Sum of the two numbers

Given two numbers `<a>` and ``

When we try to find sum of numbers

Then result should be `<result>`

Examples:

<code>a</code>	<code>b</code>	<code>result</code>
<code>3</code>	<code>2</code>	<code>5</code>

language: no

Funksjonalitet: Gjett et ord

Eksempel: Ordmaker starter et spill

Når Ordmaker starter et spill

Så må Ordmaker vente på at Gjetter blir med

Eksempel: Gjetter blir med

Gitt at Ordmaker har startet et spill med ordet "bløtt"

Når Gjetter blir med på Ordmakers spill

Så må Gjetter gjette et ord på 5 bokstaver

BDD. Cucumber steps

```
@Given("^two numbers (\\d*) and (\\d*)")
public void given(double a, double b) {
    this.a = a;
    this.b = b;
    this.calc = new Calculator();
}

@When("^we try to find sum of numbers")
public void whenGetSum() {
    result = calc.sum(a, b);
}

@When("^we try to find product of numbers")
public void whenGetProduct() {
    result = calc.product(a, b);
}

@Then("^result should be (\\d*)")
public void then(double res) {
    Assert.assertEquals(res, result, delta: 0.0001);
}
```

```
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>3.0.2</version>
</dependency>
```


BDD. Test class

```
@CucumberOptions (  
    features="src/test/resources/cucumber",  
    glue="cucumber.steps",  
    format = {  
        "pretty",  
        "html:target/cucumber-reports/cucumber-pretty",  
        "json:target/cucumber-reports/CucumberTestReport.json",  
        "rerun:target/cucumber-reports/rerun.txt"  
    })  
  
@Test  
public class CalculatorTest extends AbstractTestNGCucumberTests {  
  
}
```

```
<dependency>  
    <groupId>io.cucumber</groupId>  
    <artifactId>cucumber-testng</artifactId>  
    <version>3.0.2</version>  
</dependency>
```

BDD

Плюсы:

- Легко переиспользовать
- Сценарии хранятся там же, где и код автотестов

Минусы:

- Громоздко



Сценарий в классе с тестом

Плюсы:

- Наглядно
- Удобно поддерживать



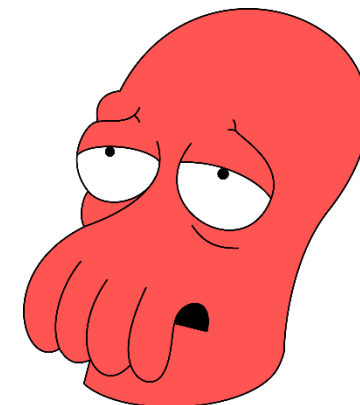
JavaDoc

```
/**
 * Открываем главную страницу
 * Переходим на страницу Solutions
 * Проверяем наличие breadcrumbs
 */
public class BreadCrumbsTest extends BaseTest {

    @Test(dataProvider = "getMenuItemData", dataProviderClass = BreadCrumbsData.class)
    public void breadCrumbsTest(String menuItem, String breadCrumbs) {

        MainPage mainPage = new MainPage().open();
        SolutionPage solutionPage = mainPage.openSolutionPage(menuItem);

        assertEquals(solutionPage.getBreadCrumbsText(), breadCrumbs);
    }
}
```



Плюсы Groovy

- Скорость освоения
- Синтаксический сахар
- Компактность кода
- Динамическая типизация ?

```
def javaString = 'java'  
def groovyString = "${javaString}"  
def yetAnotherGroovyString = ""  
    |   ${groovyString}  
    |   ${2 + 2}  
    ""
```

```
def list = [1, 3, 5]  
def map = [1: "blue", 2: "green"]
```

Плюсы Spock

- Скорость освоения
- Gherkin-style описание теста
- Тест-кейсы в коде!



Getting Started

```
compile 'org.codehaus.groovy:groovy-all:2.5.2'
```

```
compile 'org.spockframework:spock-core:1.1-groovy-2.4'
```

```
<dependency>  
  <groupId>org.codehaus.groovy</groupId>  
  <artifactId>groovy-all</artifactId>  
  <version>2.5.2</version>  
  <type>pom</type>  
</dependency>
```

```
<dependency>  
  <groupId>org.spockframework</groupId>  
  <artifactId>spock-core</artifactId>  
  <version>1.1-groovy-2.4</version>  
  <scope>test</scope>  
</dependency>
```

Spock



```
def "Pay with card"() {  
  given: "Prepare token"  
  |  
    def token = QaApi.getToken(userInfo)  
  |  
  
  when: "Pay"  
  |  
    def payResponse = BillCoreCardRequest.payCard(token, mwAcId, payCard)  
  then: "Check status"  
  |  
    payResponse.errorCode == ErrorCode.ERR_OK  
  
  when: "Find invoice status"  
  |  
    def invoiceStatus = BillCoreRequest.findStatus(mwAcId)  
  then: "Check status"  
  |  
    invoiceStatus == TBillCoreStatus.PAYABLE  
}
```


В заключение

Контакты:



Всякое полезное:

