

Producing Living Documentation Reports with Cucumber and Serenity BDD



John Ferguson Smart

AUTHOR OF “BDD IN ACTION”

@wakaleo www.johnfergusonsmart.com



Overview



Writing good living documentation

Feature and scenario names

Feature, scenario, and table descriptions

Generating living documentation reports



Writing Good Living Documentation



Good Living Documentation



Are written as
business
specifications, not test
scripts



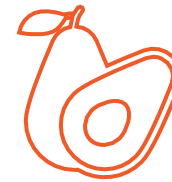
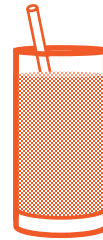
Use business
language, not
technical language



Are self-explanatory



Diet-friendly Smoothies





Diet-friendly Smoothies

Smoothies are organized into categories based on their number of calories

Dieting members are shown which smoothies fit their diet requirements

Dieting members should be proposed healthy alternatives

Apple and kale	160 kcal	🔥
Orange and yogurt	240 kcal	🔥🔥
Triple berry blend	380 kcal	🔥🔥🔥

The one where Diana is shown the flame count for the smoothies she views

The one where Doug selects a high calorie smoothie



Feature and Scenario Names



Feature File Names and Directory Structures

```
▼ features
  ▼ healthy_smoothies
    diet_friendly_options.feature
  ▼ loyalty_cards
    super_smoothie_program.feature
  ▼ managing_orders
    canceling_an_order.feature
    placing_an_order.feature
    viewing_order_history.feature
```

- ✓ **Readable business language**
- ✓ **Use a meaningful directory structure**
- ✓ **Avoid codes or identifiers**
- ✓ **(Although numerical ordering can be OK)**



Feature File Names and Directory Structures

```
▼ features
  ▼ healthy_smoothies
    diet_friendly_options.feature
  ▼ loyalty_cards
    super_smoothie_program.feature
  ▼ managing_orders
    1_placing_an_order.feature
    2_viewing_order_history.feature
    3_canceling_an_order.feature
```

- ✓ **Readable business language**
- ✓ **Use a meaningful directory structure**
- ✓ **Avoid codes or identifiers**
- ✓ **(Although numerical ordering can be OK)**



Scenario Names

Scenario Outline: Earning points
Given **Michael** is a Morning Fre
When **Michael** purchases **<Quantity>**
Then he should earn **<Points>** p

Examples:

Drink	Quant
Banana	2
Triple Berry Blend	1
Earl Grey	3

- ✓ A "table of contents" for your feature
- ✓ List business rules
- ✓ Or list key examples
- ✓ Go from general to more specific



Scenario Names



Diet-friendly Smoothies

Smoothies are organized into categories based on their number of calories

Dieting members are shown which smoothies fit their diet requirements

Dieting members should be proposed healthy alternatives

Apple and kale	160 kcal	🔥
Orange and yogurt	240 kcal	🔥🔥
Triple berry blend	380 kcal	🔥🔥🔥

The one where Diana is shown the flame count for the smoothies she views

The one where Doug selects a high calorie smoothie



Scenario Names

Feature: Diet-friendly Smoothies

Scenario Outline: Smoothies are organized into categories based on their number of calories

Scenario: Dieting members are shown which smoothies fit their diet requirements

Scenario: Dieting members should be proposed healthy alternatives



Feature and Scenario Descriptions



Feature Descriptions

Feature: Diet-friendly Smoothies

Scenario Outline: Smoothies are organized into categories based on their number of calories

Scenario: Dieting members are shown which smoothies fit their diet requirements

Scenario: Dieting members should be proposed healthy alternatives



Feature Descriptions

Feature: Diet-friendly Smoothies

Free-text description

Morning Freshness wants to attract health-conscious customers by proposing options to help smoothie buyers know if a smoothie is in line with their diet plans.

With the 'diet-friendly' mode, Morning Freshness members can buy smoothies knowing that they are not threatening their diet plans.

We hope this will encourage people on diets, or health-conscious customers in general, prefer Morning Freshness as a smoothie provider.

Scenario Outline: Smoothies are organized into categories based on their number of calories



Scenario Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

When a <Smoothie> smoothie contains <Calories> calories

Then it should be classed as a <Flame Count> flame smoothie

Examples:

Category	Smoothie	Calories	Flame Count
Under 200	Apple and Kale	160	1
Under 300	Orange and Yogurt	240	2
Under 400	Triple Berry Blend	380	3
400 or more	Strawberry Special	490	4



Scenario Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

Each category is represented by a number of "flames" - the more "flames" the smoothie has, the more calories it contains.

When a <Smoothie> smoothie contains <Calories> calories
Then it should be classed as a <Flame Count> flame smoothie

Examples:

Category	Smoothie	Calories	Flame Count
Under 200	Apple and Kale	160	1
Under 300	Orange and Yogurt	240	2
Under 400	Triple Berry Blend	380	3
400 or more	Strawberry Special	490	4



Example Table Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

Each category is represented by a number of "flames" - the more "flames" the smoothie has, the more calories it contains.

When a <Smoothie> smoothie contains <Calories> calories

Then it should be classed as a <Flames> flame smoothie in the <Range> range

Examples:

Category	Smoothie	Calories	Flames	Range
Under 200	Apple and Kale	160	1	Healthy
Under 300	Orange and Yogurt	240	2	Healthy
Under 400	Triple Berry Blend	380	3	Indulgent
400 or more	Strawberry Special	490	4	Indulgent



Example Table Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

Each category is represented by a number of "flames" - the more "flames" the smoothie has, the more calories it contains.

When a <Smoothie> smoothie contains <Calories> calories
Then it should be classed as a <Flames> flame smoothie in the <Range> range
Examples:

Category	Smoothie	Calories	Flames	Range
Under 200	Apple and Kale	160	1	Healthy
Under 300	Orange and Yogurt	240	2	Healthy

Examples:

Category	Smoothie	Calories	Flames	Range
Under 400	Triple Berry Blend	380	3	Indulgent
400 or more	Strawberry Special	490	4	Indulgent



Example Table Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

Each category is represented by a number of "flames" - the more "flames" the smoothie has, the more calories it contains.

When a <Smoothie> smoothie contains <Calories> calories
Then it should be classed as a <Flames> flame smoothie in the <Range> range

Examples: Healthy Smoothies

Category	Smoothie	Calories	Flames	Range
Under 200	Apple and Kale	160	1	Healthy
Under 300	Orange and Yogurt	240	2	Healthy

Examples: Indulgent Smoothies

Category	Smoothie	Calories	Flames	Range
Under 400	Triple Berry Blend	380	3	Indulgent
400 or more	Strawberry Special	490	4	Indulgent



Example Table Descriptions

Scenario Outline: Smoothies are organized into categories based on their number of calories

Each category is represented by a number of "flames" - the more "flames" the smoothie has, the more calories it contains.

When a <Smoothie> smoothie contains <Calories> calories
Then it should be classed as a <Flames> flame smoothie in the <Range> range

Examples: Healthy Smoothies

Smoothies under 300 calories are classed as 'Healthy'

Category	Smoothie	Calories	Flames	Range
Under 200	Apple and Kale	160	1	Healthy
Under 300	Orange and Yogurt	240	2	Healthy

Examples: Indulgent Smoothies

Smoothies with 300 or more calories are classed as 'Indulgent'

Category	Smoothie	Calories	Flames	Range
Under 400	Triple Berry Blend	380	3	Indulgent
400 or more	Strawberry Special	490	4	Indulgent



Demo



Living Documentation with Serenity BDD



What Is Serenity BDD?



Open source



Higher quality
automated
acceptance tests



Test reports and
living
documentation



Learn more at

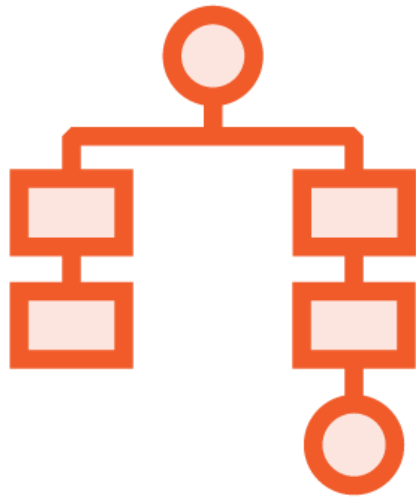
[*serenity-bdd.github.io*](https://serenity-bdd.github.io)

or at

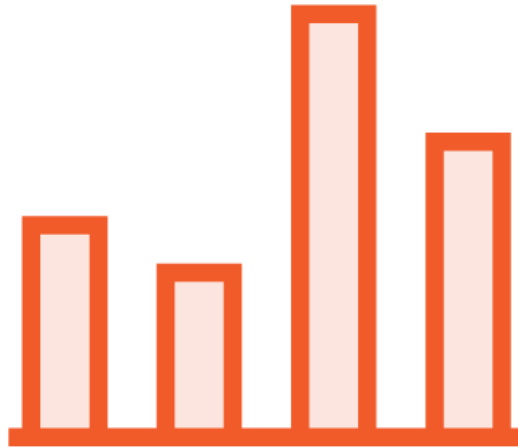
[*serenity-bdd.info*](https://serenity-bdd.info)



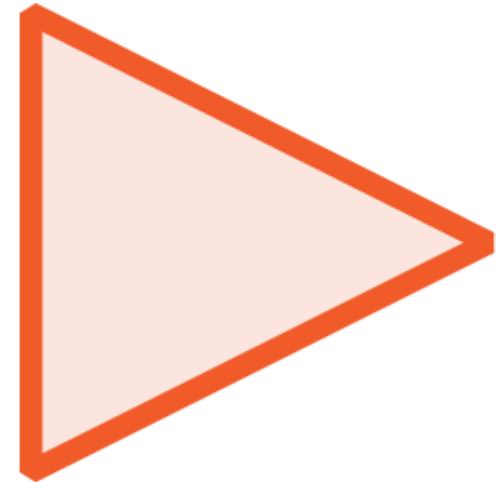
Adding Serenity BDD Reports



Serenity BDD
dependencies

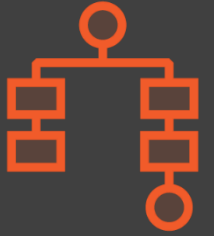


Serenity BDD reporter



Serenity BDD test
runner

Serenity BDD Dependencies in Maven



```
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>4.2.0</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-junit</artifactId>
  <version>4.2.0</version>
  <scope>test</scope>
</dependency>
```

Cucumber dependencies



Serenity BDD Dependencies in Maven

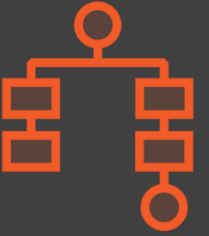


```
<dependency>
  <groupId>io.cucumber</groupId>
  <artifactId>cucumber-java</artifactId>
  <version>4.2.0</version>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-core</artifactId>
  <version>2.0.48</version>
  <scope>test</scope>
</dependency>
```

Serenity BDD core
dependency



Serenity BDD Dependencies in Maven

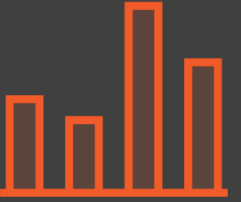


```
<dependency>
  <groupId>net.serenity-bdd</groupId>
  <artifactId>serenity-cucumber4</artifactId>
  <version>1.0.12</version>
  <exclusions>
    <exclusion>
      <groupId>io.cucumber</groupId>
      <artifactId>cucumber-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

Serenity BDD
Cucumber 4
dependencies



Serenity BDD Reporter in Maven

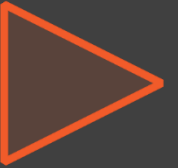


```
<plugin>
  <groupId>net.serenity-bdd.maven.plugins</groupId>
  <artifactId>serenity-maven-plugin</artifactId>
  <version>2.0.4</version>
  <executions>
    <execution>
      <id>serenity-reports</id>
      <phase>post-integration-test</phase>
      <goals>
        <goal>aggregate</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Serenity BDD
Maven plugin



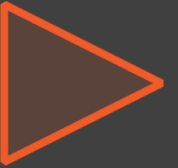
Serenity BDD Cucumber Runner



```
import cucumber.api.CucumberOptions;  
import cucumber.api.junit.Cucumber;  
import org.junit.runner.RunWith;  
  
@RunWith(Cucumber.class)  
@CucumberOptions(features = "src/test/resources/features")  
public class AcceptanceTestSuite {}
```



Serenity BDD Cucumber Runner



```
import cucumber.api.CucumberOptions;  
import net.serenitybdd.cucumber.CucumberWithSerenity;  
import org.junit.runner.RunWith;  
  
@RunWith(CucumberWithSerenity.class)  
@CucumberOptions(features = "src/test/resources/features")  
public class AcceptanceTestSuite {}
```



Summary



What have you learned?

- Structuring your requirements hierarchy
- Naming features and scenarios
- Feature, scenario, and example table descriptions
- Serenity BDD living documentation

Conclusions



Course Summary



BDD is a collaborative requirements discovery practice

Example and Feature Mapping

Writing executable specifications in Gherkin

Automating executable specifications with Cucumber

Living Documentation with Serenity BDD

