Gherkin Format (Goal To Be Achieved)

*The document is designed for reference from various blogs to give an idea of how we can start taking steps towards BDD in Mendeley Data. It is not everything but a basic start which we want to implement and Gherkin is more of English Language and can be customised to our needs.*

Feature – When, Where & How to Use

|  |
| --- |
| *A single feature of the system, an aspect of a feature or a high-level description of a feature.* |
| Feature: Keywords or a Name  Description (Optional) |
| Feature: Refund item  Sales assistants should be able to refund customers' purchases.  This is required by the law, and is also essential to keep customers happy.  Rules:  Customer must present proof of purchase  Purchase must be less than 30 days ago |
| Feature: Serve coffee  As a Business owner  I want customers should be able to buy coffee  So that I can earn money |

Scenario – When, Where & How to Use

|  |
| --- |
| *A scenario illustrates a business rule and consists of a list of steps, which must start with one of the keywords Given, When, Then, But or And. It is also a test which execute specification of system.* |
| Scenario: Keywords or a Name  Given - Describe an initial context  When - Describe an event  Then - Describe an expected outcome  And - Use And for several Given / When / Then to make it more readable  But - Use But for several Given / When / Then to make it more readable |
| Scenario: feeding a small suckler cow  Given the cow weighs 450 kg  When we calculate the feeding requirements  Then the energy should be 26500 MJ |
| Scenario: feeding a small suckler cow  Given the cow weighs 450 kg  When we calculate the feeding requirements  Then the energy should be 26500 MJ  And the protein should be 215 kg  But the fat should be 10 kg |

Given – When, Where & How to Use

|  |
| --- |
| *The purpose of Given steps is to put the system in a known state before the user (or external system) starts interacting with the system (in the When steps). Avoid talking about user interaction in givens. If you have worked with use cases, givens are your preconditions.* |
| Given there are no users on site  Given the database is clean  Given I am logged in as "Mahesh" |

When – When, Where & How to Use

|  |
| --- |
| *The purpose of When steps is to describe the key action the user performs* |
| When I am on "/some/page"  When I fill "username" with "Mahesh"  When I fill "password" with "123456"  When I press "login"  When I call “ls –la” |

Then - When, Where & How to Use

|  |
| --- |
| *The purpose of Then steps is to observe outcomes. The observations should be related to the business value/benefit in your feature description. The observations should inspect the output of the system (a report, user interface, message, command output) and not something deeply buried inside it (that has no business value and is instead part of the implementation).*   * *Verify that something related to the Given+When is (or is not) in the output* * *Check that some external system has received the expected message (was an email with specific content successfully sent?)* |
| Given I logged in as admin in Terminal  When I call "echo hello"  Then the output should be "hello" |

And / But - When, Where & How to Use

|  |
| --- |
| *If you have several Given, When or Then steps you can write:*  Scenario: Multiple Givens  Given one thing  Given another thing  Given yet another thing  When I open my eyes  Then I see something  Then I don't see something else |
| *Or you can use And or but steps, allowing your Scenario to read more fluently:*  Scenario: Multiple Givens  Given one thing  And an other thing  And yet an other thing  When I open my eyes  Then I see something  But I don't see something else |
| *If you prefer, you can indent scenario steps in a more programmatic way, much in the same way your actual code is indented to provide visual context:*  Scenario: Multiple Givens  Given one thing  And an other thing  And yet an other thing  When I open my eyes  Then I see something  But I don't see something else |

Scenario Outlines- When, Where & How to Use

|  |
| --- |
| *Copying and pasting scenarios to use different values quickly becomes tedious and repetitive* |
| Scenario: eat 5 out of 12  Given there are 12 cucumbers  When I eat 5 cucumbers  Then I should have 7 cucumbers  Scenario: eat 5 out of 20  Given there are 20 cucumbers  When I eat 5 cucumbers  Then I should have 15 cucumbers |
| *Scenario outlines allow us to more concisely express these examples using a template with placeholders, using Scenario Outline, Examples with tables and < > delimited parameters*  Scenario Outline: eating  Given there are <start> cucumbers  When I eat <eat> cucumbers  Then I should have <left> cucumbers  Examples:  | start | eat | left |  | 12 | 5 | 7 |  | 20 | 5 | 15 |  *Note: The placeholders indicate that when the Examples row is run they should be substituted with real values from the Examples table. If a placeholder name is the same as a column title in the Examples table, then this is the value that will replace it.* |

Background - When, Where & How to Use

|  |
| --- |
| *Background allows you to add some context to the scenarios in a single feature. A Background is much like a scenario containing several steps.* |
| Feature: Multiple site support  As a Mephisto site owner  I want to host blogs for different people  In order to make gigantic piles of money  Background:  Given a global administrator named "Greg"  And a blog named "Greg's anti-tax rants"  And a customer named "Dr. Bill"  And a blog named "Expensive Therapy" owned by "Dr. Bill"  Scenario: Dr. Bill posts to his own blog  Given I am logged in as Dr. Bill  When I try to post to "Expensive Therapy"  Then I should see "Your article was published."  Scenario: Dr. Bill tries to post to somebody else's blog, and fails  Given I am logged in as Dr. Bill  When I try to post to "Greg's anti-tax rants"  Then I should see "Hey! That's not your blog!"  Scenario: Greg posts to a client's blog  Given I am logged in as Greg  When I try to post to "Expensive Therapy"  Then I should see "Your article was published." |