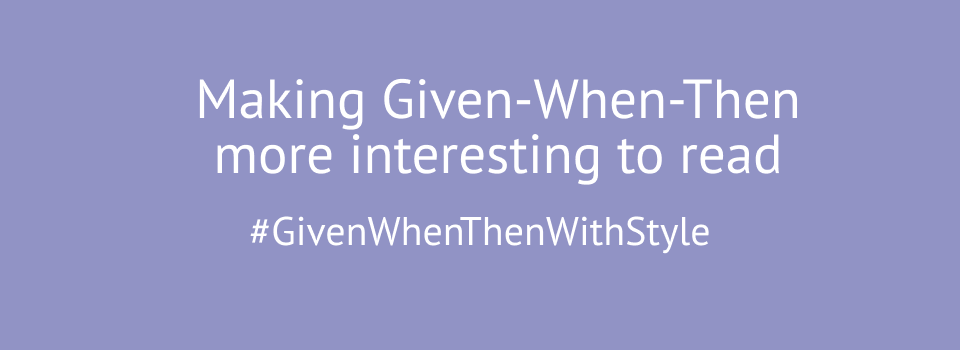
* Сценарий не должен ссылаться на UI так как в тесте должно быть описан юс кейс юзера и то, какую проблему он пытается решить, так же конкретные действия юзера не должны находиться в Given степе, их можно перенести в When степ: When I try to optimize the peak shape by applying the force peak settings to the chromatogram integration settings

**Given When Then Best Practices: Make Scenarios Interesting**

**2021-09-20**

**Gherkin**



**Share!**

Given-When-Then structure can become repetitive, especially for complex scenarios, and this can cause readers to skip or ignore parts. Instead of a new challenge, this week I’ll give you some nice tricks to make feature files less monotonous, to keep reader interest and help them pay attention.

**Use bullet points for groups of steps**

For scenarios that require several steps to set up, or that check several outcomes, repeating the basic keywords can become quite boring. Using And/But conjunctions can make the list slightly less repetitive, but this usually isn’t enough to make long scenarios easy to read.

Long lists of steps usually contain groups that relate to the same subject. For example, there are two groups in the following scenario, one for the user’s payment method and the other about the order:

Scenario: Card purchase

Given a user is signed in

And the user has a VISA card registered

And the card billing address country is DE

And the card has active strong customer authentication

And the requested order delivery is 15 November

And the order weight is 0.5 kg

And the order delivery address is in the US

And the order value is 100 EUR

When the user attempts to purchase

...

You can use bullet points instead of Gherkin keywords, and this trick can be quite useful to make a nice visual distinction and let readers see logical groups clearly. Start the group with a keyword (Given/When/Then), but use an asterisk (\*) instead of the keyword for the rest of the steps in the group. Here’s an example:

Scenario: Card purchase

Given a user is signed in

And the user has a registered payment method

\* card type is VISA from DE

\* strong customer authentication is active

And the user makes an order

\* requested delivery of 15 November

\* package weight 0.5 kg and declared value 100 EUR

When the user attempts to purchase

Then ...

This way of visualising groups makes related steps immediately obvious. You can use asterisks in any section, not just set-ups. In theory, you can completely replace all Given/When/Then keywords with bullet points, but that will bring us back to the problem of all steps looking the same. For best results, mix asterisks and keywords to break up the monotony.

**Use inline tables instead of structured groups of steps**

When the steps in a group have a common structure, an even better trick is to replace the steps with inline tables. For example, we can rewrite the set-up of the previous scenario with two tables:

Scenario: Card purchase

Given a user is signed in

And the user has a registered payment method

| card type | billing country | SCA |

| VISA | DE | Active |

And the user makes an order with

| requested delivery | weight | value |

| 15 November | 0.5 kg | 100 EUR |

When the user attempts to purchase

Then ...

This trick works well for relatively short structures, where all the properties can fit nicely into a single line. For longer structures, you can flip the table and make it vertical. Here’s an example:

Scenario: Card purchase

Given a user is signed in

And the user has a registered payment method

| card type | VISA |

| billing country | DE |

| SCA | Active |

And the user makes an order with

| requested delivery | 15 November |

| weight | 0.5 kg |

| value | 100 EUR |

When the user attempts to purchase

Then ...

Tables are a great way to visually show structure, especially if you work with only one level of hierarchy. If you have multiple levels of hierarchy in step groups, you can mix asterisks and tables.

**Split long lists of examples into blocks**

Using tables for grouping is quite a common trick, and in Given-When-Then tools tables are mostly used to show examples under a scenario outline.

An outline can become boring to read if it is followed by a table with many examples. Readers will not be able to notice subtle differences in long tables. One potential workaround is to create several outlines with the same scenario, and break up examples into smaller groups. This is not ideal because the scenario is replicated, and whoever updates the file will need to remember to change it in multiple places. Instead of copying the scenario, you can just list multiple groups of examples, each with its own header. Each group can optionally have a title and a header, explaining what’s specific and important about that group. Here’s an example:

Scenario outline: Christmas delivery period calculation

During the Christmas order period shipping and

customs can get a bit busy

Given an order of 0.5 kg

And the delivery time is between 15 November and 15 January

And the delivery address country is <destination country>

When the user previews delivery

Then the expected period should be <delivery period>

Examples: European union

There are no customs checks and we can ship overnight

to all EU countries

| Destination country | Delivery Period |

| DE | 1 day |

| FR | 1 day |

| LT | 1 day |

| GR | 1 day |

| IR | 1 day |

Examples: EEC/Non EU

Same rules apply for EEC countries that are not in EU

| Destination country | Delivery Period |

| CH | 1 day |

| NO | 1 day |

Examples: Brexit

Nobody knows what's going to happen, leave a lot of time

for eventual border problems

| Destination country | Delivery Period |

| GB | 1 month |

**Group scenarios into rule blocks**

When a single feature has many scenarios, often it’s useful to break a long file apart and replace it with a directory containing several smaller files. However, if you want to use the same background context in all the scenarios, this solution is not good, since the background will have to be repeated across all the files. In such cases, you can use the *Rule* keyword for an intermediate grouping between a feature and scenarios. The Rule keyword does not affect automation at all, but it allows you to introduce a section title and description to a group of scenarios.

In the following example, we can use a common shipping rate table across a range of scenarios, grouping them by order type:

Feature: Order shipping estimator

Background:

Given the following shipping rates

| country | weight | price |

| DE | 100 gr | 0.1 EUR |

| DE | 1 kg | 0.5 EUR |

| DE | 10 kg | 2 EUR |

| AT | 100 gr | 0.1 EUR |

| AT | 1 kg | 0.5 EUR |

| AT | 10 kg | 2 EUR |

| US | 100 gr | 0.5 EUR |

| US | 1 kg | 2.5 EUR |

| US | 10 kg | 5 EUR |

Rule: Choosing shipping method

Depending on whether we are shipping a perishable or non-perishable item,

we must select the appropriate container and delivery mechanism

Scenario outline: Food shipping to EU

...

Scenario outline: Food shipping outside EU

...

Rule: Cost estimation

Scenario outline: EU without tax

...

Scenario outline: Non-EU without tax

...

Scenario outline: Non-EU with tax

...

Rule: Delivery time estimation

...

**Use markdown in descriptions**

The previous tricks work with most Given-When-Then tools, but SpecFlow also has one particular trick that other tools do not support and it can make contextual descriptions below the title easier to read. You can format contextual descriptions using markdown, for example use the two asterisks (\*\*) to emphasise a part of the sentence. SpecFlow+ will apply markdown formatting in the results, showing the enclosed words in a bold font. Similarly, instead of pasting a lot of text from an external source into a feature file, you can use markdown to create a web link. Here’s an example:

Rule: Choosing shipping method

Depending on whether we are shipping a \*\*perishable\*\* or \*\*non-perishable\*\* item,

we must select the appropriate container and delivery mechanism. The detailed

explanation of this process is on the [product shipping wiki](https://wiki/shipping-rules)

This is a great way to avoid duplicating common description text across many feature files. For example, if you want to reference the same shipping rules from several feature contexts, set them up on a wiki and link to the appropriate page