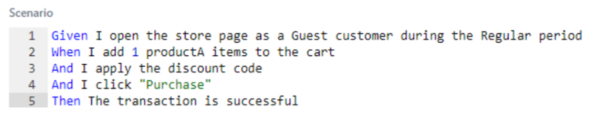
Maximize the efficiency of using Cucumber Scenarios in Xray

What you'll learn

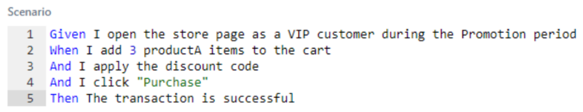
- The shortcomings of "typical" Cucumber Scenarios
- How BDD Steps Library helps with reusability
- How data-driven Cucumber testing optimizes scaling & efficiency

Overview

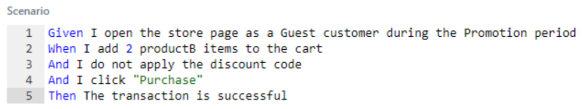
Let's say we have five Cucumber tests that validate the same flow but with different data permutations. Furthermore, since several users created these tests manually, we can see slight discrepancies in the step wording.


Scenario

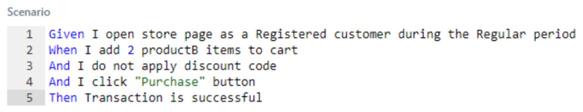
```
Given I open the store page as a Guest customer during the Regular period
When I add 1 productA items to the cart
And I apply the discount code
And I click "Purchase"
Then The transaction is successful
```


Scenario

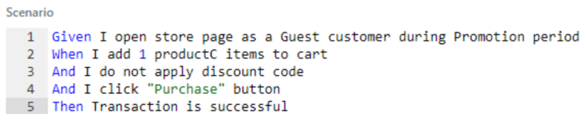
```
Given I open the store page as a VIP customer during the Promotion period
When I add 3 productA items to the cart
And I apply the discount code
And I click "Purchase"
Then The transaction is successful
```


Scenario

```
Given I open the store page as a Guest customer during the Promotion period
When I add 2 productB items to the cart
And I do not apply the discount code
And I click "Purchase"
Then The transaction is successful
```


Scenario

```
Given I open store page as a Registered customer during the Regular period
When I add 2 productB items to cart
And I do not apply discount code
And I click "Purchase" button
Then Transaction is successful
```


Scenario

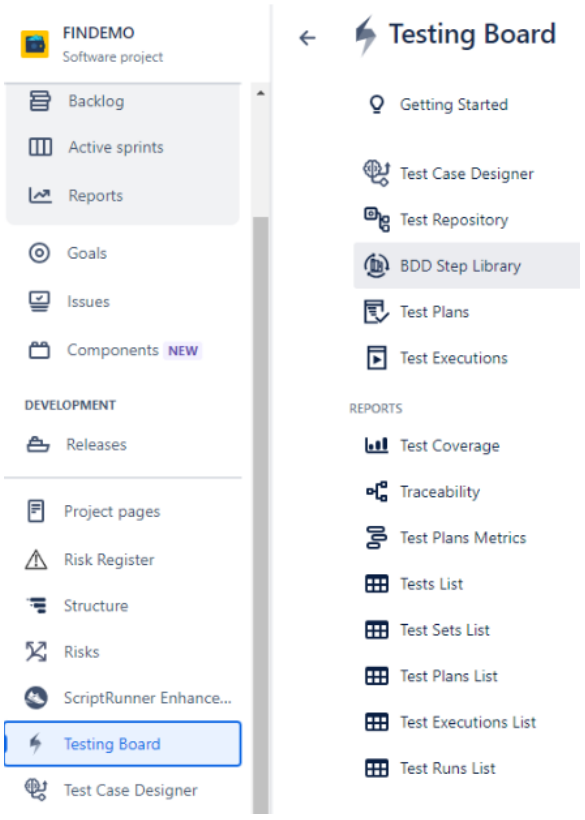
```
Given I open store page as a Guest customer during Promotion period
When I add 1 productC items to cart
And I do not apply discount code
And I click "Purchase" button
Then Transaction is successful
```

Why is this problematic? Having 5 test issues means 5x maintenance whenever the specification needs to change. Further, manual multi-user creation or adjustment leads to messier step definitions for automation engineers (as we can have slightly different steps, with different wording, that perform exactly the same actions).

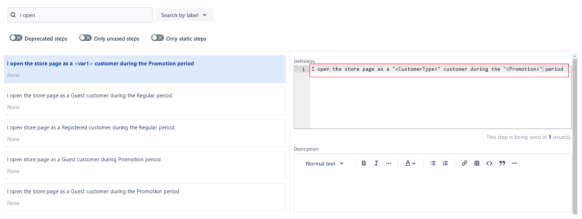
In this tutorial, we will optimize the efficiency of this mini regression suite via data-driven automation with the help of BDD Steps Library and Parameterized Tests features of Xray.

BDD Steps Library

First, we will consolidate the steps and improve future reusability. Xray indexes Gherkin steps automatically in the library whenever users create/edit Cucumber-type Tests/Preconditions. We will open the project library from the Testing Board.



Now, we will choose any redundant steps and edit it to have the desired syntax version. We will parameterize the dynamic parts with descriptive variable names. We will also surround those parameters in quotes as an optional good practice.



I open the store page as a <var1> customer during the Promotion period
None

I open the store page as a Guest customer during the Regular period
None

I open store page as a Registered customer during the Regular period
None

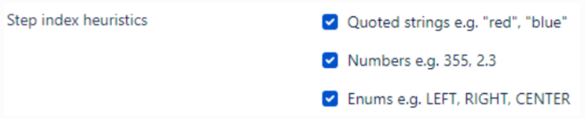
I open store page as a Guest customer during Promotion period
None

I open the store page as a Guest customer during the Promotion period
None

```
I open the store page as a "<CustomerType>" customer during the "<Promotion>" period
```

This step is being used in 1 issue(s).

In the list of the steps on the left, we can see that Xray already extracted the notion of a parameter for one dynamic occurrence (<var1>). The rules for the automatic identification of data-driven elements are defined in the settings we visited earlier.

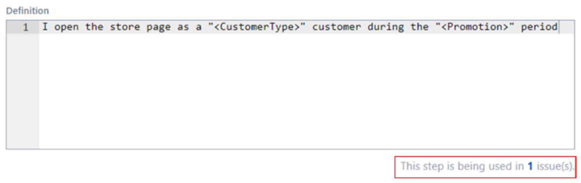


Step index heuristics

- ✓ Quoted strings e.g. "red", "blue"
- ✓ Numbers e.g. 355, 2.3
- ✓ Enums e.g. LEFT, RIGHT, CENTER

Enums include uppercase strings (like VIP in our example). Also, docstrings and elements delimited by pipes are supported.

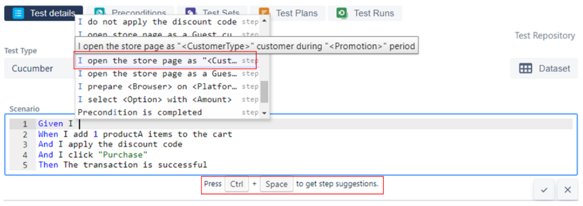
Once the library is updated for all step variations, we can go to one of the Cucumber tests for which we’ ve done the edits (using the link under the definition).



```
I open the store page as a "<CustomerType>" customer during the "<Promotion>" period
```

This step is being used in 1 issue(s).

We adjust its whole Scenario definition using the auto-complete functionality (for the steps that have not been updated automatically from the library). Remember to prompt the suggestions using the shortcut specified in the tooltip.

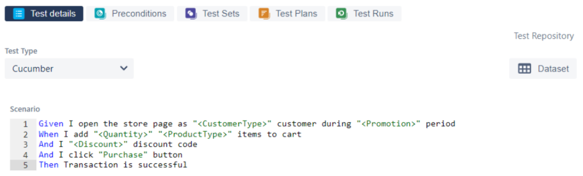


```
I do not apply the discount code          step
I open store page as a Guest cu…          step
I open the store page as "<CustomerType>" customer during "<Promotion>" period
I open the store page as "<Cust…          step
I open the store page as a Gues…          step
I prepare <Browser> on <Platfor…          step
I select <Option> with <Amount>           step
Precondition is completed                 step
```

```
Given I 
When I add 1 productA items to the cart
And I apply the discount code
And I click "Purchase"
Then The transaction is successful
```

Press Ctrl + Space to get step suggestions.

The final form of our data-driven Scenario looks like this:



```
Given I open the store page as "<CustomerType>" customer during "<Promotion>" period
When I add "<Quantity>" "<ProductType>" items to cart
And I "<Discount>" discount code
And I click "Purchase" button
Then Transaction is successful
```

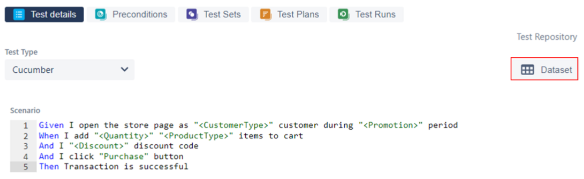
Once the step optimization is completed in this updated “master” test issue, we can delete or archive the other four Cucumber tests and remove the outdated iterations of steps from the library (keep in mind, steps not marked as “static” are automatically deleted when they are no longer being used by any issue).

If tests require the same actions, you can select the option from the suggested list instead of typing. For new actions, consider creating them in the BDD Steps Library first, rather than in the test issues - that way you can avoid redundancy and improve consistency upfront.

Next, we will complete the upgrade to data-driven automation using parameterized Cucumber tests.

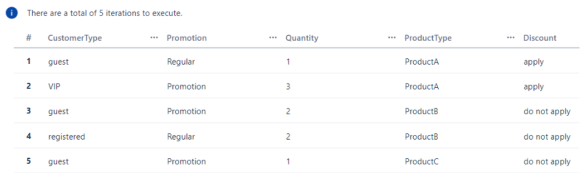
Parameterized Cucumber Tests

With the variables syntax (<>) already in place, we only need the dataset to provide the permutations for the execution. We can add it from the same screen by specifying the data manually or importing it from CSV.



```
Given I open the store page as "<CustomerType>" customer during "<Promotion>" period
When I add "<Quantity>" "<ProductType>" items to cart
And I "<Discount>" discount code
And I click "Purchase" button
Then Transaction is successful
```

In this case, we just recreated the same permutations that were present in the five original tests. Just keep in mind that one of the benefits of this data-driven approach is that you can easily scale the dataset up or down without worrying about adjusting multiple test issues one by one.

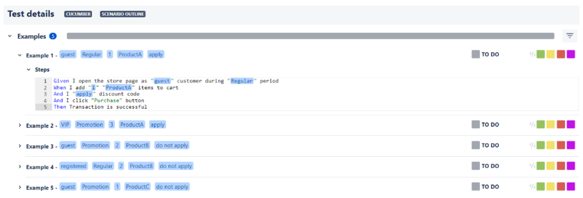


There are a total of 5 iterations to execute.

#	CustomerType	Promotion	Quantity	ProductType	Discount
1	guest	Regular	1	ProductA	apply
2	VIP	Promotion	3	ProductA	apply
3	guest	Promotion	2	ProductB	do not apply
4	registered	Regular	2	ProductB	do not apply
5	guest	Promotion	1	ProductC	do not apply

Additionally, you consider leveraging the Test Case Designer feature of Xray Enterprise to not only generate more complex datasets faster but also to access data coverage analysis visualizations.

When we get to the execution, here is what you see in the Test Run details - several iterations of the same Scenario, with the data values replacing parameter syntax and the execution status per data permutation.

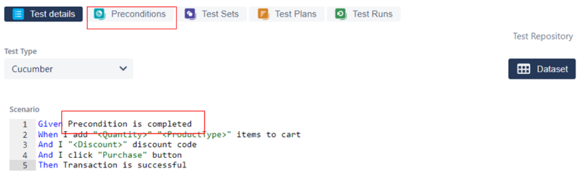


Example 1 -	guest	Regular	1	ProductA	apply	TO DO
Example 2 -	VIP	Promotion	3	ProductA	apply	TO DO
Example 3 -	guest	Promotion	2	ProductB	do not apply	TO DO
Example 4 -	registered	Regular	2	ProductB	do not apply	TO DO
Example 5 -	guest	Promotion	1	ProductC	do not apply	TO DO

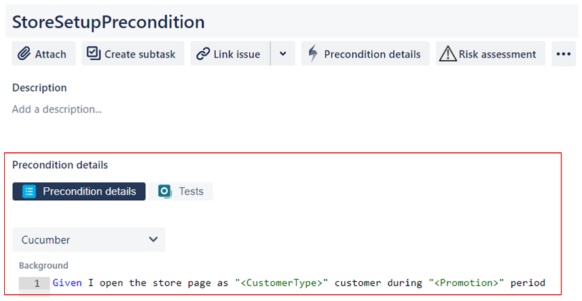
```
Given I open the store page as "guest" customer during "Regular" period
When I add "1" "ProductA" items to cart
And I "apply" discount code
And I click "Purchase" button
Then Transaction is successful
```

Data-driven precondition

As an optional exercise, let’s say we notice that many different tests will use the Given step as the precondition. We then decide to separate it out.



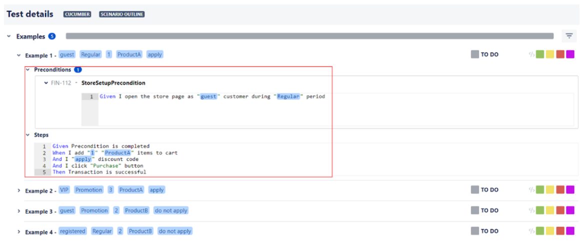
```
Given Precondition is completed
When I add "<Quantity>" "<ProductType>" items to cart
And I "<Discount>" discount code
And I click "Purchase" button
Then Transaction is successful
```



```
Given I open the store page as "<CustomerType>" customer during "<Promotion>" period
```

Here is the tricky part - the dataset, including parameters from both the test and the precondition, is always in the **test issue**. Whenever you have a data-driven precondition, you must ensure the dataset for **each** associated test has the parameters and values specified in that precondition.

So, in our example, we can execute the updated test issue without any changes to the dataset to see this breakdown in the Test Run details. Notice that the value replacement highlighted in blue is exactly the same as before we had a precondition.



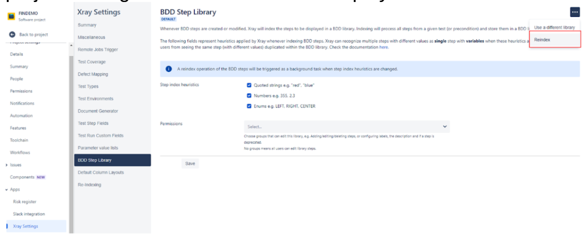
Example 1 -	guest	Regular	1	ProductA	apply	TO DO
Example 2 -	VIP	Promotion	3	ProductA	apply	TO DO
Example 3 -	guest	Promotion	2	ProductB	do not apply	TO DO
Example 4 -	registered	Regular	2	ProductB	do not apply	TO DO

```
Given I open the store page as "guest" customer during "Regular" period
```

```
Given Precondition is completed
When I add "1" "ProductA" items to cart
And I "apply" discount code
And I click "Purchase" button
Then Transaction is successful
```

Tips

- If the BDD Steps Library feature is out of sync for any reason, you can launch Reindex from the project settings to make sure all the steps you need are in the library.



BDD Step Library
DEFAULT

Whenever BDD steps are created or modified, Xray will index the steps to be displayed in a BDD library. Indexing will process all steps from a given test (or precondition) and store them in a BDD l

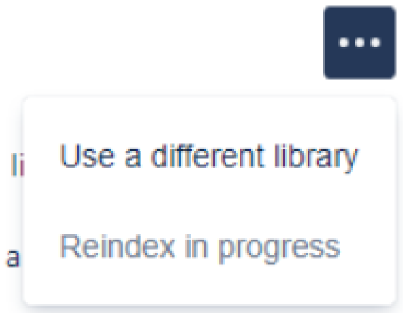
The following fields represent heuristics applied by Xray whenever indexing BDD steps. Xray can recognize multiple steps with different values as **single** step with **variables** when these heuristics a users from seeing the same step (with different values) duplicated within the BDD library. Check the documentation here.

A reindex operation of the BDD steps will be triggered as a background task when step index heuristics are changed.

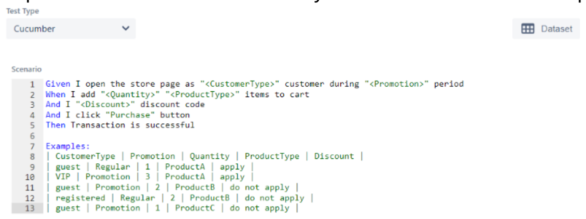
Step index heuristics: ✓ Quoted strings e.g. "red", "blue"; ✓ Numbers e.g. 355, 2.3; ✓ Enums e.g. LEFT, RIGHT, CENTER

Permissions: Select...
Choose groups that can edit this library, e.g. Adding/editing/deleting steps, or configuring labels, the description and if a step is deprecated.
No groups means all users can edit library steps.

  You can track the status from the same triple-dot menu.



- Ensure the parameter name syntax is the same between the script and the dataset.
- Consider adding the project-level lists in Xray settings for parameter values that you frequently reuse.
- If necessary, you can override the test-level dataset with the one in the test plan or the test execution.
- You can achieve the same results using the Scenario Outline format with the Examples table in the definition (like having embedded data) and no dataset defined at the Test level; Xray supports this implicit data in case you want to embed it directly in the Gherkin specification. It depends on the dataset size and if you wish to mix it with the script for review clarity purposes.



```
Given I open the store page as "<CustomerType>" customer during "<Promotion>" period
When I add "<Quantity>" "<ProductType>" items to cart
And I "<Discount>" discount code
And I click "Purchase" button
Then Transaction is successful

Examples:
| CustomerType | Promotion | Quantity | ProductType | Discount |
| guest | Regular | 1 | ProductA | apply |
| VIP | Promotion | 3 | ProductA | apply |
| guest | Promotion | 2 | ProductB | do not apply |
| registered | Regular | 2 | ProductB | do not apply |
| guest | Promotion | 1 | ProductC | do not apply |
```

References

- BDD Steps Library
- Parameterized Tests