**Real – time Scenarios (OBSTACLE #1)**

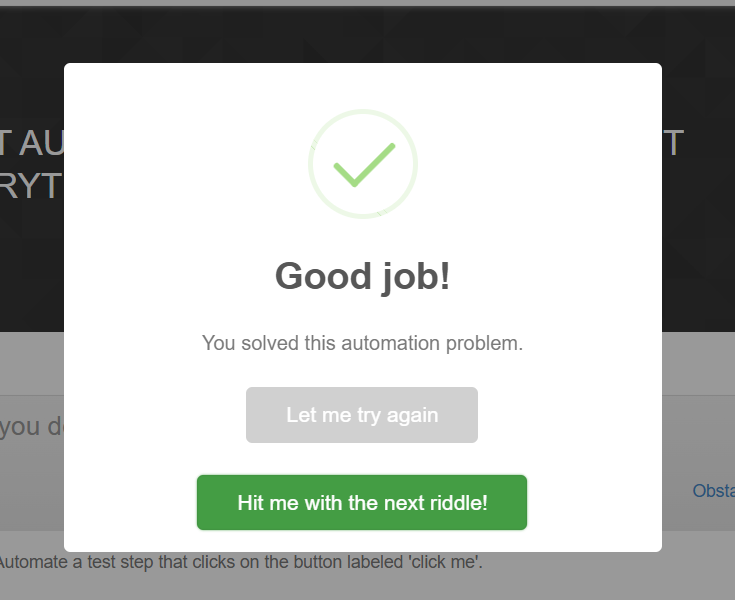
**IDs are not everything – Elements with same IDs**

Obstacle list provided by tosca: <https://obstaclecourse.tricentis.com/Obstacles/List>

**1. Obstacle:**  <https://obstaclecourse.tricentis.com/Obstacles/22505/retry>

Automate a teststep that clicks on the button labelled “Click me”

* Create module
* Add it to the testcase
* Run testcase

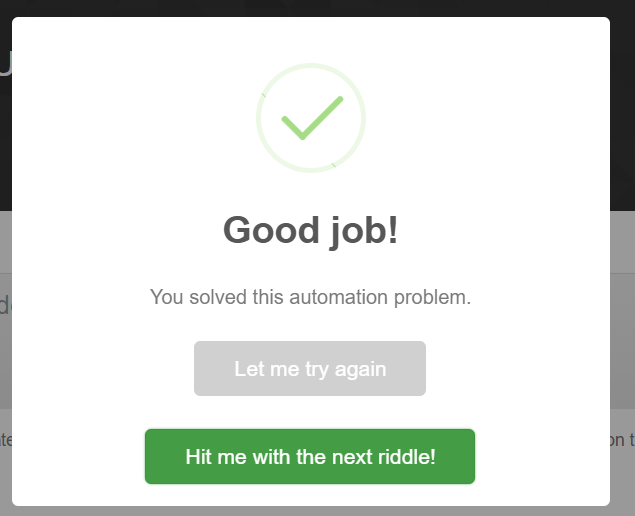


**2. Obstacle:**

Duplicate Elements with same properties

<https://obstaclecourse.tricentis.com/Obstacles/12952/retry>

* Click on the second button
* If there are same elements then select the nearest tag



**3. Obstacle:**

Dynamically Changing table elements – Not a table

<https://obstaclecourse.tricentis.com/Obstacles/64161/retry>

Let's assume your 'application under test' generates an offer id every time you click the according button. Unfortunately, it gets added to the table randomly. Find a way to automate a stable test step that

1) clicks the button,

2) buffers the generated offer id and

3) enters it into the edit box. The test steps need to run every time!

* Use Anchor control to store the Order ID
* To make order ID uniquely identifiable

**4. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/72954/retry>

Dynamically changing ID property

Use one module creating two automated test steps to click the button two times in a row. Sounds easier than it is!

**5. Obstacle:**

**6. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/94441/retry>

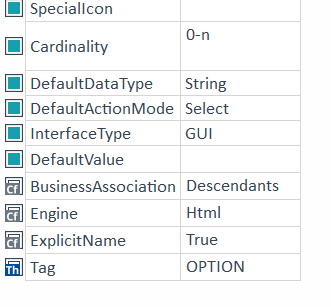
Multiselect List Box | Cardinality | Explicit Name

Multiselect list box: Select all the testing methods, that are supported by Tosca (Functional, End2End, GUI testing and Exploratory Testing)

Cardinality 🡪 using object “n” number of times

* Change the cardinality of object from 0-1 to 0-n. so that we can use it n number of times

Explicit name 🡪 True so that we can use index of the particular object



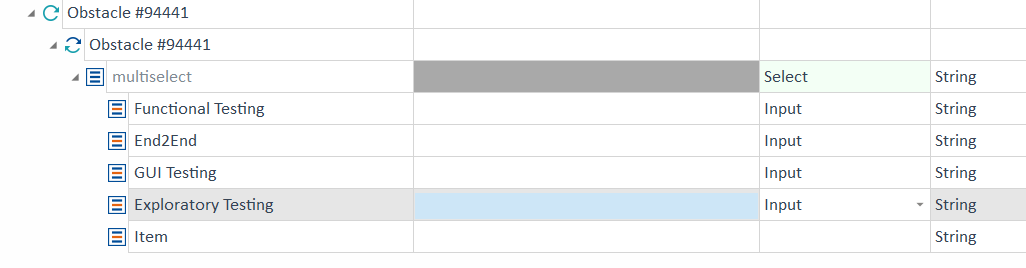
Steps

1. Select complete table

2. then select only 1 object from that table for module

3. change the object name in module to “Item”

4. set properties cardinality and explicit name



Run testcase

**7. Obstacle:**

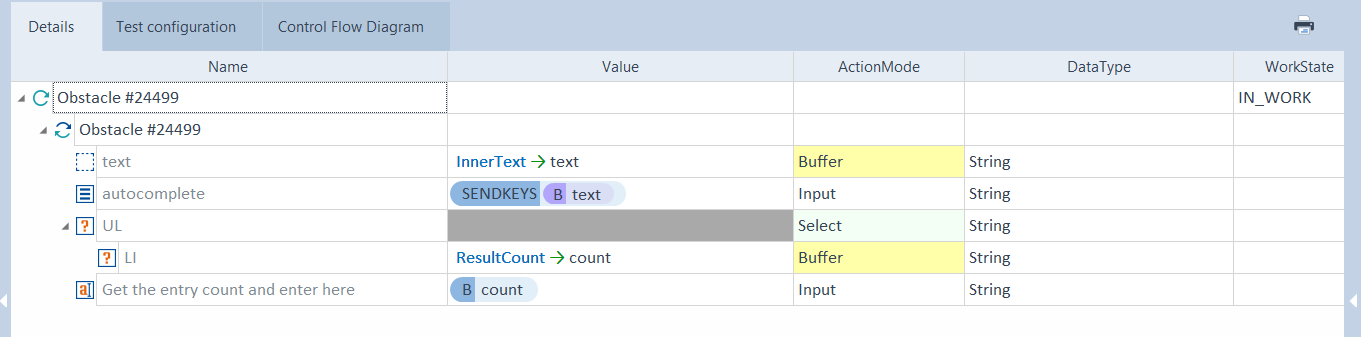
Autocomplete TextBox | Result count | Inner text

<https://obstaclecourse.tricentis.com/Obstacles/24499/retry>

Autocomplete Textbox: Type the given characters in the box, then count the entries and enter the number in the textbox.

Search it in the textbox and see how many results are shown in drop down

Then in next box write the number of results is shown

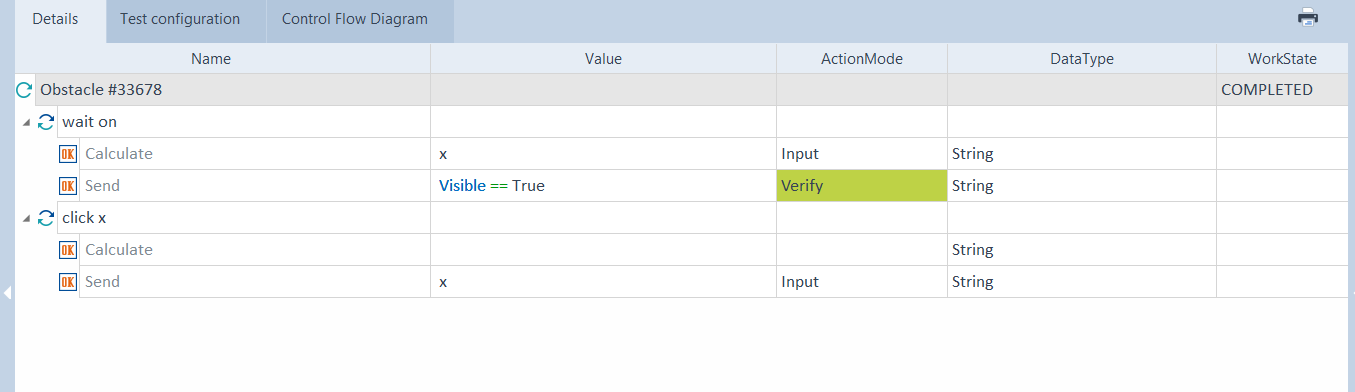


**8. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/33678/retry>

Control Properties and waiton

 Click on the button "CALCULATE" automatically. After some time the button "SEND" gets visible/enabled. Click on it to complete the obstacle.

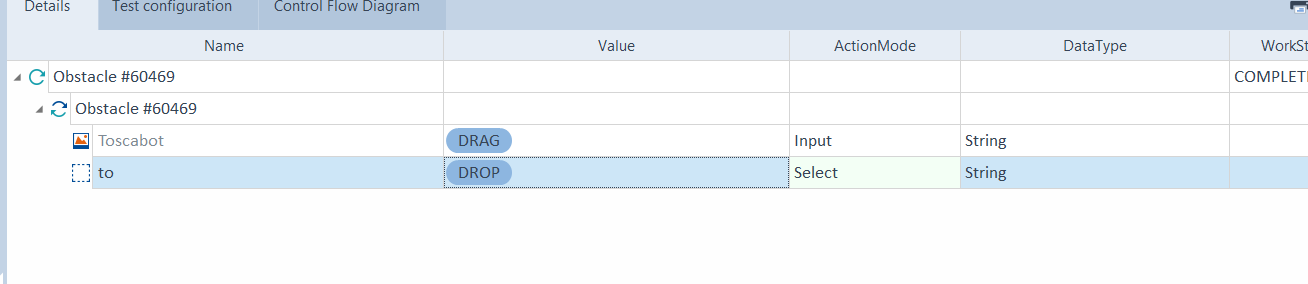


**9. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/60469/retry>

Drag and drop

 Drag the image from the left to the right to complete this obstacle.



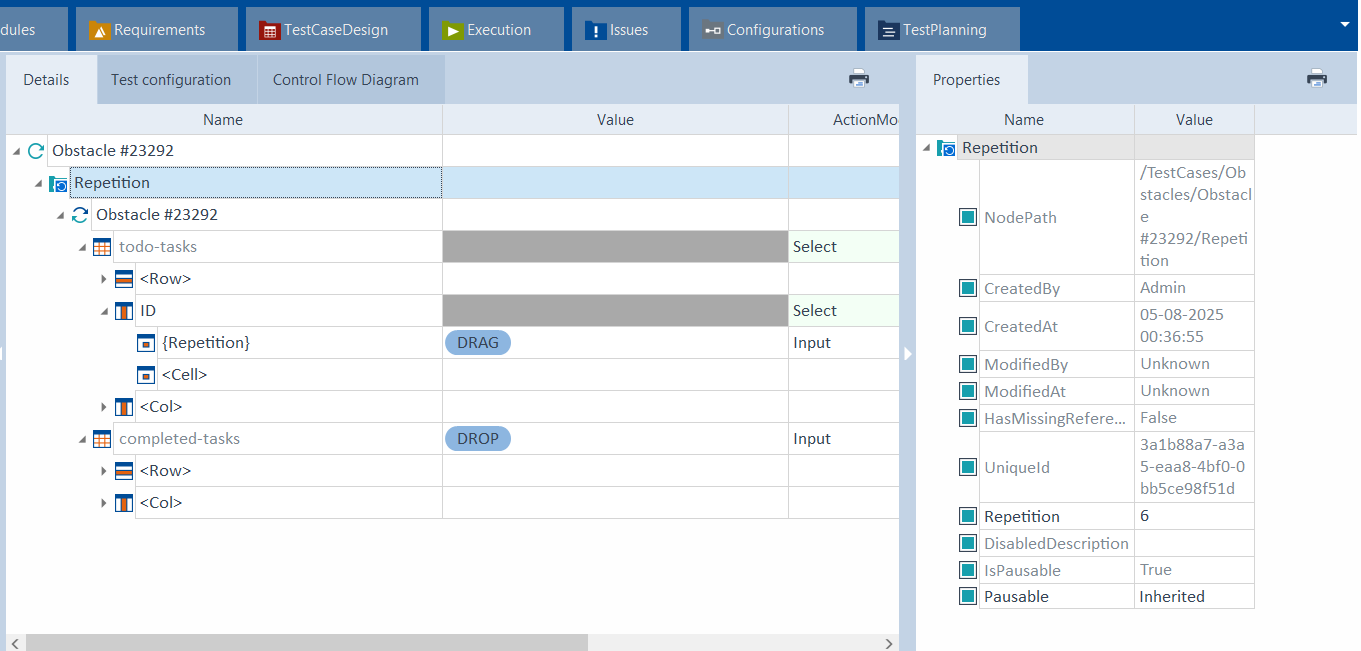
**10. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/23292/retry>

Move all tasks from 'ToDo' to 'Completed' in the order of their given task ID (1, 2, ... 6).

Table steering, Drag, Drop

* Use repetition add the module to the repetition folder
* Change the repetition number to 6 in properties
* Then to the ID in column add the {Repetition}



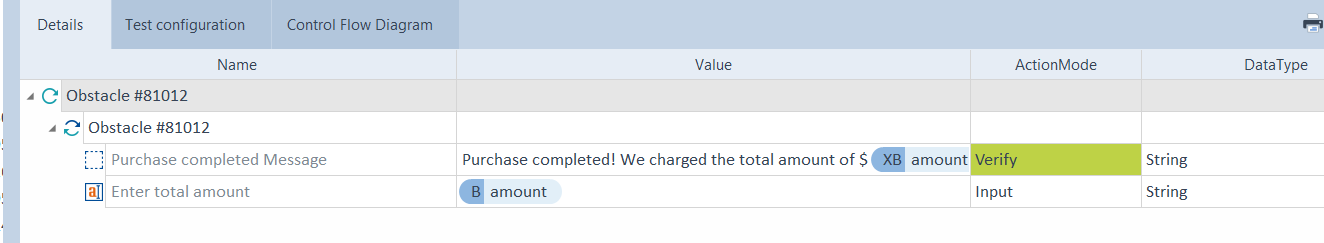
**11. Obstacle:**

Please automate a test step taking the total amount shown in the success message and enter it into the text box.

<https://obstaclecourse.tricentis.com/Obstacles/81012/retry>

Extract text | XBuffer | Dynamic text | Standard module, Automation tool

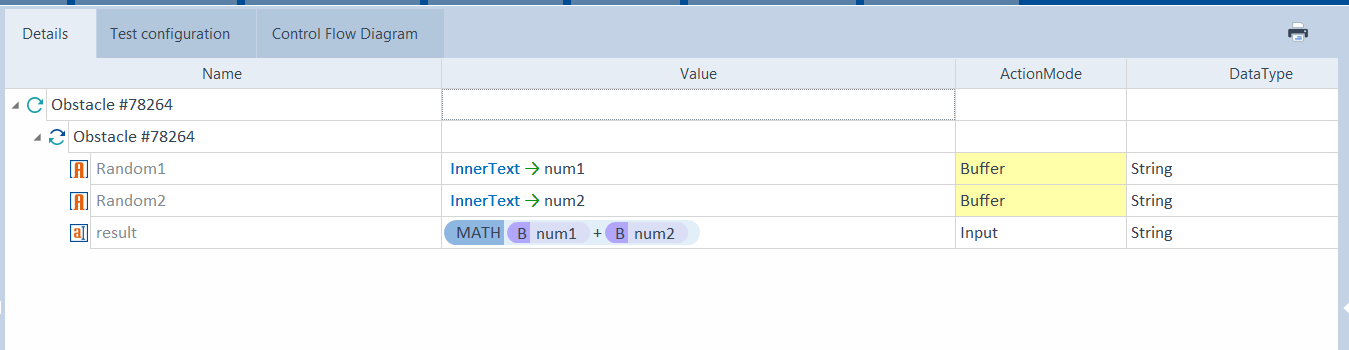
* Take total amount from the text (mixed number and text)



**12. Obstacle: Addition**

<https://obstaclecourse.tricentis.com/Obstacles/78264/retry>

 Input the sum of the two randomly generated numbers into the sum field



**13. Obstacle:**

<https://obstaclecourse.tricentis.com/Obstacles/87912/retry>

Be- Fast automate

XML Engine, Load, create and Scan XML Files

Click the button 'Load Books' to load an XML structure of books to the text area. Search through the XML and find the ISBN of the book titled 'Testing Computer Software'. Enter the ISBN to the text box. Do all of that automatically

* Use TBox Read/Create file Standard module
* This will help you to store the .xml file in local drive
* And then use same module to read the .xml file

**14. Obstacle:**

**15. Obstacle:**

**16. Obstacle:**

**17. Obstacle:**

**18. Obstacle:**