DoesQA Actions and Selector Types Testing Report

This document summarizes the findings from executing a test specification described in the following document:

https://waynemroseberry.github.io/assets/DoesQA_selectorcomparison_elementaction_testspec.pdf

The primary goal was to explore DoesQA behavior designing and executing workflows using different types of selector addressing schemes for different types of HTML objects. The approach was end-to-end, focusing on everything from the user experience to problems in the application.

1 Contents

2 Covered 2

3 Approach 4

4 Issues discovered 4

4.1 Data loss and stuck in "Saving…" state if workflow has a loop 4

4.2 Node I was editing vanished after trying to add an element 5

4.3 Errors trying to use elements for actions not suited to them are not self-explanatory and do not offer user any help to fix problem 5

4.4 "Issue still not clickable" using Visible Text selector on SELECT element seems misleading. 6

4.5 "Check" action seems to look at "Value" attribute to do the comparison, while "Select" action seems to use the "Visible Text" attribute to select. This difference makes using the two together confusing. 6

4.6 Check Text for Visible Text on a SELECT element always returns a catenated list of all the Option element inner text, making checking specific item impossible (?). 7

4.7 The Touch action on a SELECT element leaves the drop down displayed even if the Select action is executed immediately after. 8

4.8 Elements can have the same name, but the element picker UI for actions does not give you enough information to tell which is which 8

4.9 Duplicated connected nodes do not duplicate their connections 10

4.10 When nodes branch, only one of the branches is selectable for moving. 11

4.11 Default location of helper comments occlude connection lines, making it difficult to select them. 12

4.12 Tab order in run reports page hits the "open in new window" icon twice 12

4.13 Are the automatically selected names for runs necessary, do they run the risk of creating combinations some customers might object to? 15

4.14 Sometimes CTRL+Click deselects the prior selected item instead of adding the new item to the set of collected items 15

5 Feature Suggestions 16

5.1 Feature Idea: Make element ids assignable to variables that can be branched in workflows to allow fast creation of common flows to different elements. 16

5.2 It would be useful to be able to select just a branch and run only that branch and its descendants rather than the whole workflow 16

2 COVERED

The following DoesQA features were covered

Action types: Touch, Set, Select, Check Value, Check Text, Name Test

Workflow branching

Workflow Run

Workflow Reporting monitoring, viewing

Element creation and editing

Element types: CSS, XPath, Visible Text

The following small set of HTML element types were covered:

SELECT element

Input element

Span

The following addressing types were used:

CSS with ID attribute, without ID attribute

Test data

The following items were created on DoesQA and available for inspection to investigate issues found:

Workflow that demonstrates “Saving…” stuck state and data loss	https://app.does.qa/app/flows/13894/designer

Workflow that was built to represent most of the test coverage in this report	https://app.does.qa/app/flows/13927/designer
Span – addressable by CSS ID	**wayneblog_pickuplable_cssid** `#pickupbldg`
	wayneblog_pickuplable_full_xpath `/html/body/span[2]`
	wayneblog_pickuplable_visibletext bldg2
	wayneblog_pickuplable_xpath `//*[@id="pickupbldg"]`
	wayneblog_ridername_css `body > form:nth-child(5) > table > tbody > tr > td:nth-child(2) > input`
	wayneblog_ridername_visibletext enter rider name
	wayneblog_ridername_xpath `/html/body/form[2]/table/tbody/tr/td[2]/input`
	wayneblog_start_fullxpath `/html/body/form[1]/table/tbody/tr/td[1]/select`
	wayneblog_start_idselector `#start`
	wayneblog_start_visibletext Building 1
	wayneblog_start_xpath `//*[@id="start"]`
	wayneblog_time_css `body > form:nth-child(5) > table > tbody > tr > td:nth-child(1) > select`

	wayneblog_time_xpath `/html/body/form[2]/table/tbody/tr/td[1]/select`

Testing was directed against a web page located here: https://waynemroseberry.github.io/assets/testform.html

Full HTML of the web page is available in the test specification.

3 Approach

Workflow started by opening the page waynemroseberry.github.io/assets/testform.html

Next steps in workflow branched for each combination of action, control, and addressing type did the following:

1. Touch the element
2. Set the element to a value
3. Check the element value to see if it changed as intended

Also had a check for page elements that were dependent on prior actions (a SPAN element that changed based on drop down) and branched in the workflow to sequence prior checks to checking that the span changed – similar combination of addressing type as specified above.

4 Issues Discovered

4.1 Data loss and stuck in "Saving…" state if workflow has a loop

I inadvertently created a workflow that has a loop in it. On open, this workflow goes into a state where the "Saving…" icon appears and never goes away. Once in this state, two things happen:

1. Run will not launch a run
2. Subsequent edits to the workflow will be lost if one clicks away from the Designer view (e.g. click "Workflows")

An example of a workflow in this state is located at https://app.does.qa/app/flows/13894/designer. To reproduce the problem do the following:

1. Open https://app.does.qa/app/flows/13894/designer
2. Note the "Saving…" icon appears
3. Drag an action of any sort onto the design surface
4. Click "Workflows"
5. Open the same workflow again

RESULT: the actions previously placed on the surface are not there, and there was no prompt on step 4 warning that edits will be lost

4.2 NODE I WAS EDITING VANISHED AFTER TRYING TO ADD AN ELEMENT

I do not have a repro on this, and it only happened once.

1. I selected a set of four nodes that were chained together, duplicate nodes
2. Edit first node in chain, click "+" to create new element
3. I added the element (sorry, forget which one and what kind…), OK

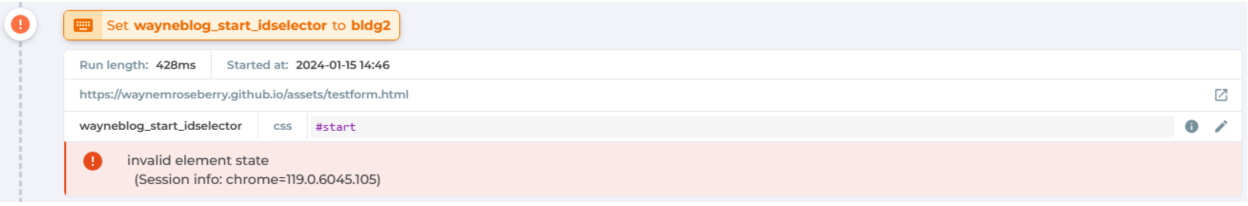
At this point, the entire set of new duplicate nodes disappeared. I am wondering if it was related to the loop in workflows causing an infinite "Saving…" state that also resulted in data loss when switching views.

4.3 ERRORS TRYING TO USE ELEMENTS FOR ACTIONS NOT SUITED TO THEM ARE NOT SELF-EXPLANATORY AND DO NOT OFFER USER ANY HELP TO FIX PROBLEM

I saw this several times when I tried to use an action that was incompatible with the element type. For example, the SET action with a SELECT element, or the SELECT action with an INPUT element. The error you get tends to be low-level, as if thrown by the inner helper libraries. I got this one trying to use the SET action on a SELECT element:

- Run script, step above fails with this error: "invalid element state (Session info: chrome=119.0.6045.105"

Here is the visual:



Set wayneblog_start_idselector to bldg2

Run length: 428ms   Started at: 2024-01-15 14:46

https://waynemroseberry.github.io/assets/testform.html

wayneblog_start_idselector   css   #start

invalid element state
(Session info: chrome=119.0.6045.105)

Issues:

- User does not have any guidance on what is wrong, or what to do.
- It seems DoesQA should realize the operation is incompatible with the type of object, could it offer assistance, or somehow make the error moot?
- There is no link to help documentation from the error to guide the user further, is that a good way to go here?

This kind of error runs counter to the goal that DoesQA seems to have to simplify the authoring of web automation scripts. For someone to understand this error, they need to recognize the difference between control types, be comfortable with the sorts of errors that don't tell them what to do. In other words, they need to be programmers that are more comfortable in code than in a graphic programming environment.

4.4 “Issue still not clickable” using Visible Text selector on SELECT element seems misleading.

1. Page has a FORM with a “SELECT” element where one of the inner text options is “Building 1”
2. Create a TOUCH action where the method is Visible Text, and the value is “Building 1”

Result – I got the following error:

```
element ("//body//*[contains(translate(text(), 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',
'abcdefghijklmnopqrstuvwxyz'), 'building 1')]") still not clickable after 120000ms
```

Is it that the item is not clickable, or is it that DoesQA did not find the element at all? When I try to do a search on the same page in the browser via CTRL+F, it is likewise unable to find the SELECT element, even when “Building 1” is visibly selected.

This seems related to the incompatible action/element type issue mentioned earlier, but in this case the outcome seems like it might be different because it seems the element is just not found at all. That seems very different than being “not clickable.” The similarity to the prior incompatible action issue is that the user is offered no guidance on what to do next, so some of the way of addressing that might be similar, even if there is also another issue to resolve on explaining what is happening in a way that seems less ambiguous?

4.5 “Check” action seems to look at “Value” attribute to do the comparison, while “Select” action seems to use the “Visible Text” attribute to select. This difference makes using the two together confusing.

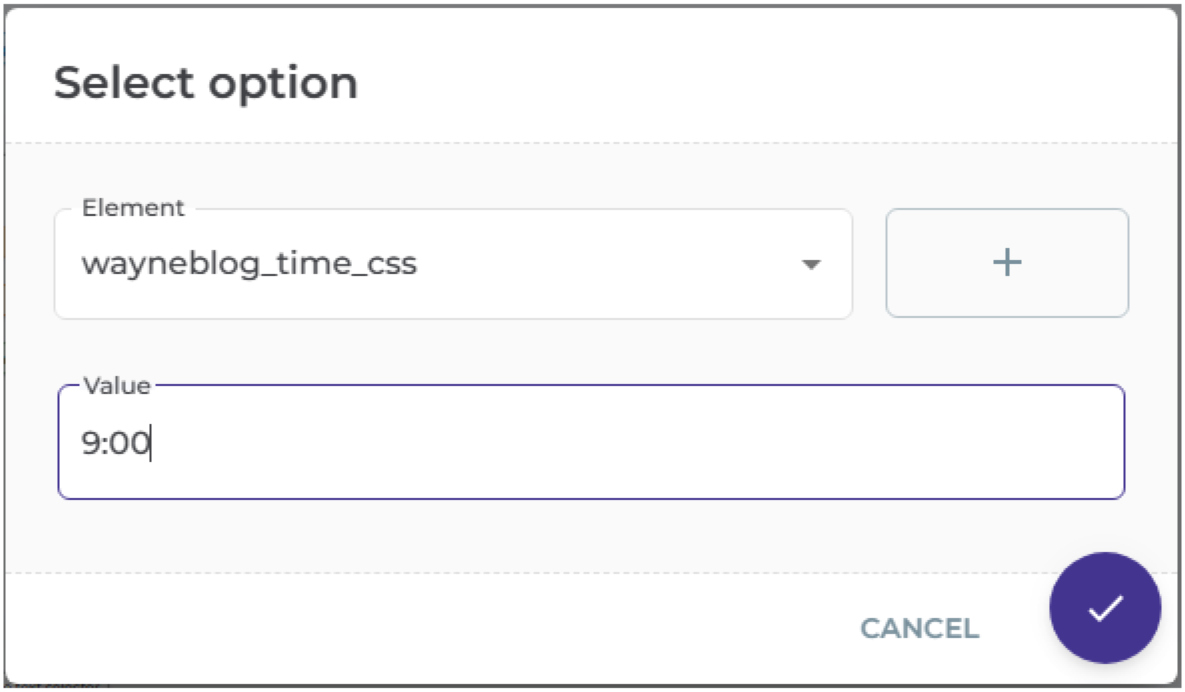
1. On a web page with a SELECT element, the entries are like the following – where “value” and inner text are different (common when programmatic name and display name need to differ).

```
<select name="pickuptime">
  <option value="eight">8:00</option>
  <option value="nine">9:00</option>
  <option value="ten">10:00</option>
  <option value="eleven">11:00</option>
</select>
```

2. Create a SELECT action for the above element, using “9:00” as the value to select
3. Follow that with a CHECK action for the above element, also using “9:00” as the value to check

RESULT: the CHECK action fails even after the SELECT action succeeds. The error message on the Check action is “Expected: “9:00”, Received: “nine””. This incongruity is confusing between the two actions.

It is made even more confusing because in the action editing dialog, both are called “Value”, which is an unfortunate coincidence that DoesQA is using this word in a different sense than the HTML does, but the “value” attribute is a special case, so it leads one to believe that maybe DoesQA might really mean the “value” attribute and not just the generic word Value.



Select option

Element: wayneblog_time_css

Value: 9:00

This does raise a difficult problem. There are good reasons why the value and display text of a SELECT OPTION are different, and there are valid reasons why someone would want to check either one of them. It feels uncertain that DoesQA arbitrarily picked one of those to use for its SELECT action and a different one for its CHECK action.

4.6 Check Text for Visible Text on a SELECT element always returns a catenated list of all the Option element inner text, making checking specific item impossible (?).

Page has this HTML:

```
<select name="pickuptime">

  <option value="eight">8:00</option>

  <option value="nine">9:00</option>

  <option value="ten">10:00</option>

  <option value="eleven">11:00</option>

</select>
```

A SELECT action with Visible Text to match on “9:00” (or any value in the list) yields an error like the following:

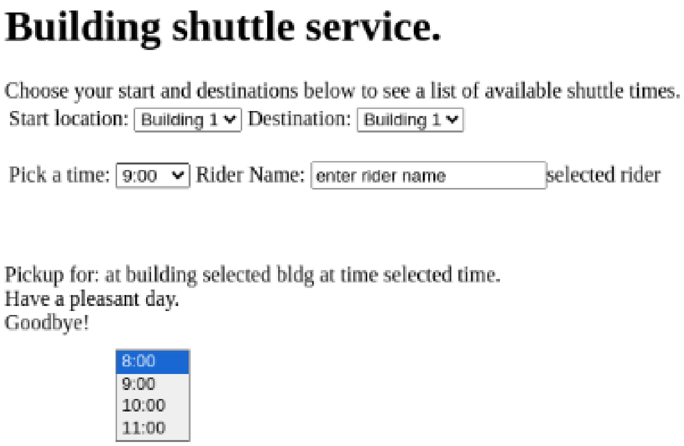
RESULT: an error… “expect(received).toEqual(expected) // deep equality - Expected - 1 + Received + 4 - 9:00 + 8:00 + 9:00 + 10:00 + 11:00”

4.7 The Touch action on a SELECT element leaves the drop down displayed even if the Select action is executed immediately after.

Below is a screen shot of what happens after executing the following:

1. Touch the SELECT element listed
2. SELECT the value “9:00”

As seen below, the drop down menu is still displayed, even though a value was selected.



Building shuttle service.

Choose your start and destinations below to see a list of available shuttle times.
Start location: Building 1 Destination: Building 1

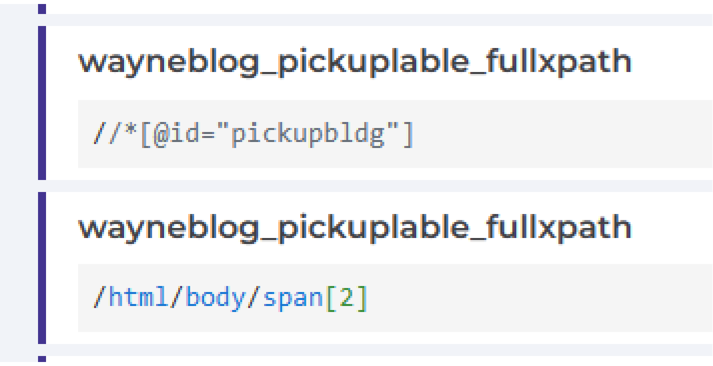
Pick a time: 9:00 Rider Name: enter rider name selected rider

Pickup for: at building selected bldg at time selected time.
Have a pleasant day.
Goodbye!

I am not sure of the expectation. Should the drop-down menu collapse once the value is selected? This is normally what happens when a user clicks with the mouse, but I believe there are modes where it remains open, such as when scrolling through the list.

4.8 Elements can have the same name, but the element picker UI for actions does not give you enough information to tell which is which

1. Create two elements with the same name, but different selection information (e.g. they have different Xpaths)
2. Look in the Elements tab, they will both be there, but their XPath selection data is there, so you can tell them apart



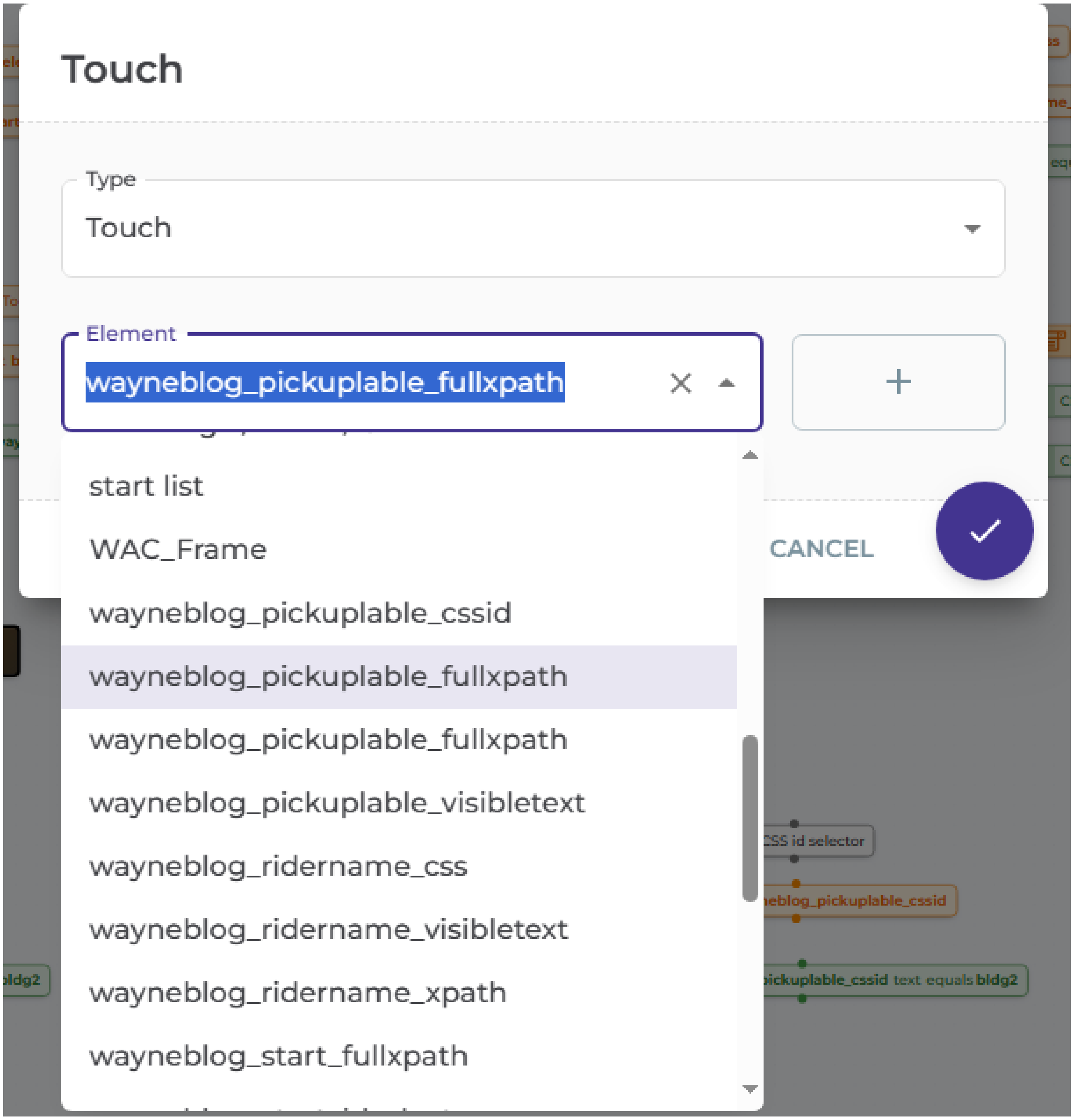
wayneblog_pickuplable_fullxpath

```
//*[@id="pickupbldg"]
```

wayneblog_pickuplable_fullxpath

```
/html/body/span[2]
```

3. Open a workflow in designer, edit an action, click the element selector
4. Both elements will be listed, but only by name, so you cannot tell them apart



Touch

Type

Touch

Element

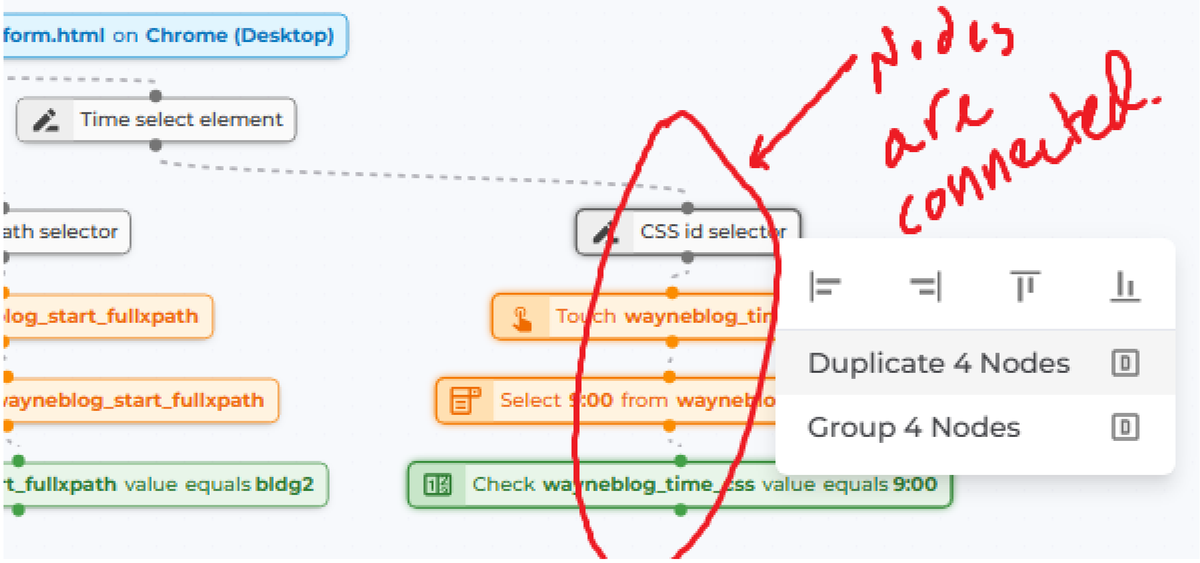
wayneblog_pickuplable_fullxpath

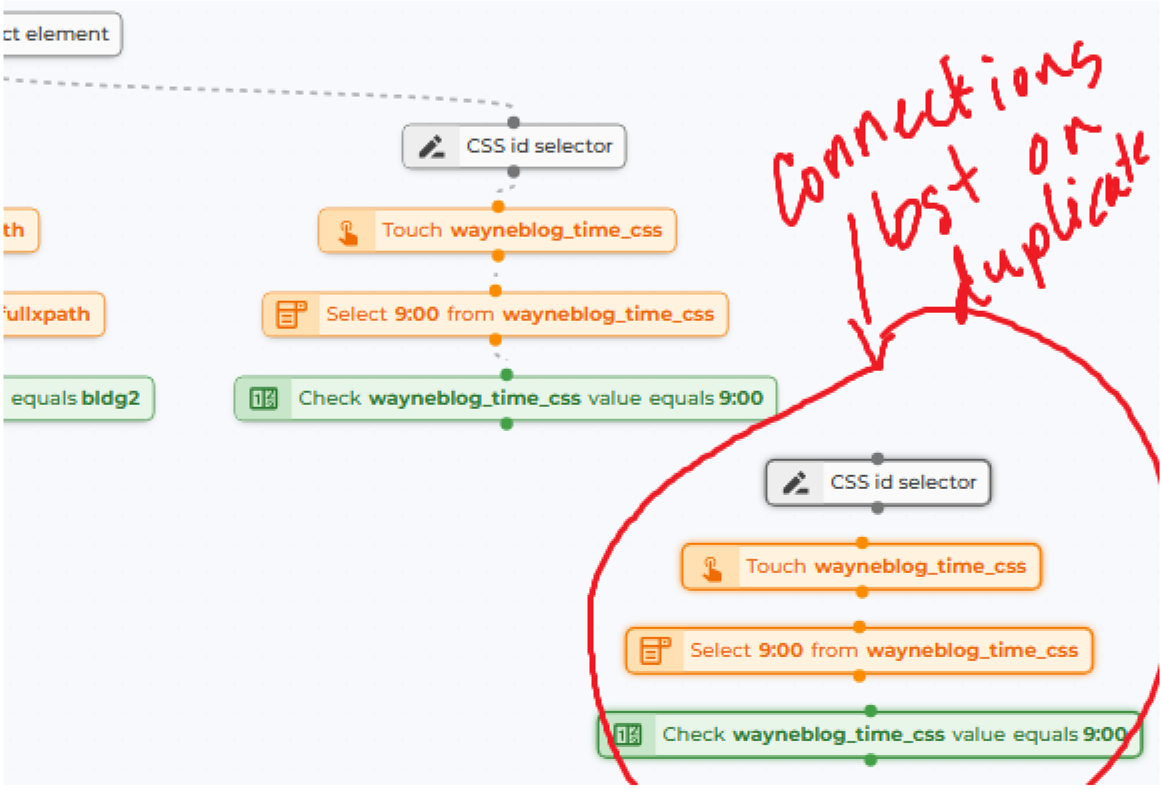
- start list
- WAC_Frame
- wayneblog_pickuplable_cssid
- wayneblog_pickuplable_fullxpath
- wayneblog_pickuplable_fullxpath
- wayneblog_pickuplable_visibletext
- wayneblog_ridername_css
- wayneblog_ridername_visibletext
- wayneblog_ridername_xpath
- wayneblog_start_fullxpath

4.9 DUPLICATED CONNECTED NODES DO NOT DUPLICATE THEIR CONNECTIONS

1. Create a series of actions on the designer workflow page and connect them
2. Use C TRL+click to select them all, Right+Click->Duplicate Nodes

Result: the duplicated nodes are not connected together. The user has to edit them to chain them together like the nodes they were duplicated from. It seems to me that preserving the connections between the duplicated connected nodes is the desired behavior. I can understand how connections to existing nodes might be a design problem, but it seems one should try to reduce the amount of "fixup" work necessary to get the desired state.

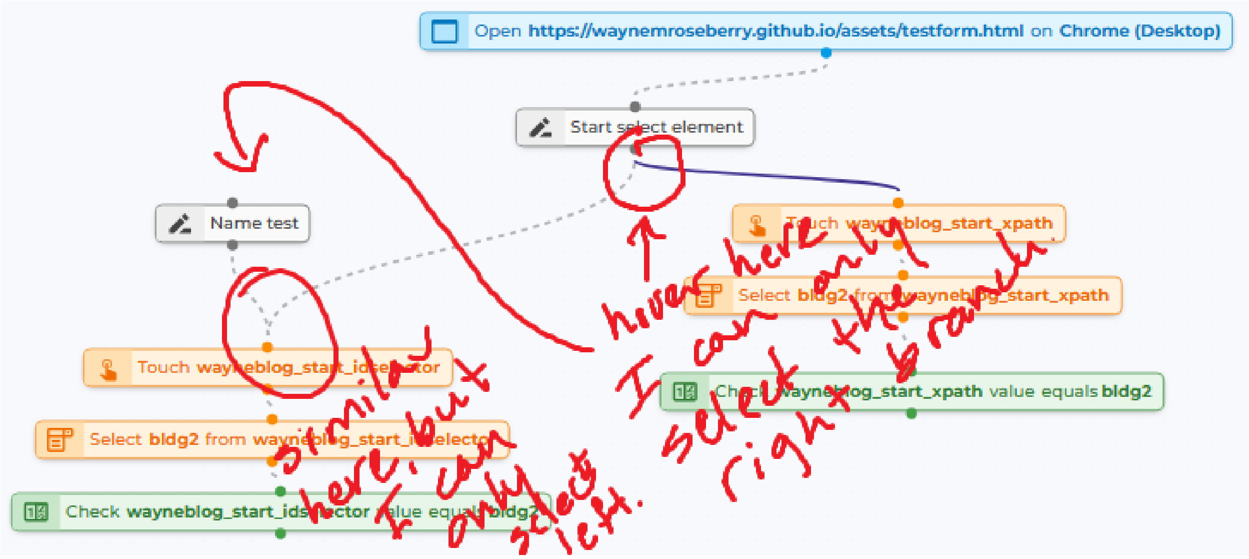



ct element
CSS id selector
th
Touch wayneblog_time_css
ullxpath
Select 9:00 from wayneblog_time_css
equals bldg2
Check wayneblog_time_css value equals 9:00
Connections lost on duplicate
CSS id selector
Touch wayneblog_time_css
Select 9:00 from wayneblog_time_css
Check wayneblog_time_css value equals 9:00

4.10 When nodes branch, only one of the branches is selectable for moving.

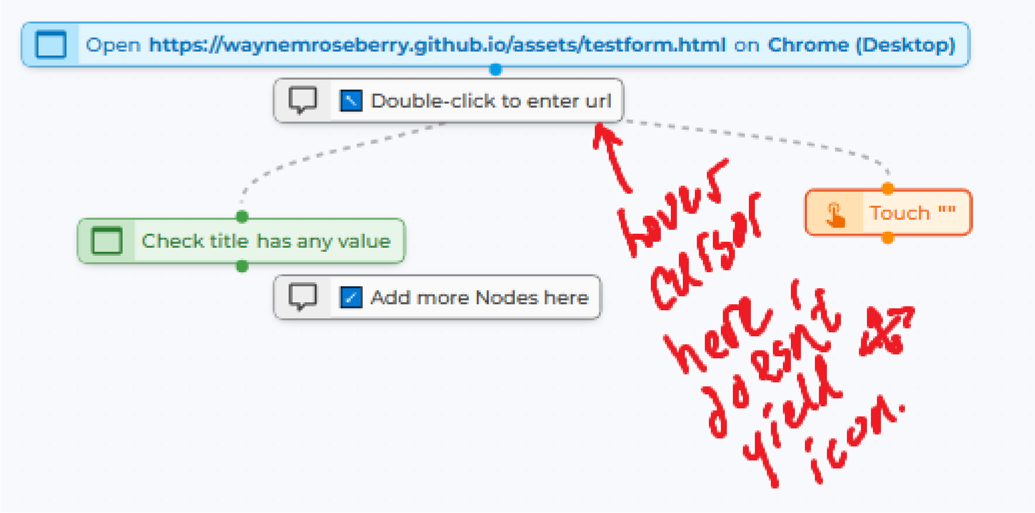
1. Drag a node onto the designer surface
2. Drag two other nodes, and connect both of them to the prior node from the same connection point
3. Hover your mouse over each of the connection lines until you get the four way arrow – click

RESULT: only one of the lines will be selected. The hot zone for both seem to overlap so tightly that you can only select one of the connection lines. In some cases, branches can exist such that some lines are entirely unselected – see the screen capture below for such a case.


Open https://waynemroseberry.github.io/assets/testform.html on Chrome (Desktop)
Start select element
Name test
Touch wayneblog_start_xpath
Select bldg2 from wayneblog_start_xpath
Check wayneblog_start_xpath value equals bldg2
Touch wayneblog_start_idselector
Select bldg2 from wayneblog_start_idselector
Check wayneblog_start_idselector value equals bldg2
hover here
I can only select the right branch
similar here, but I can only select left.

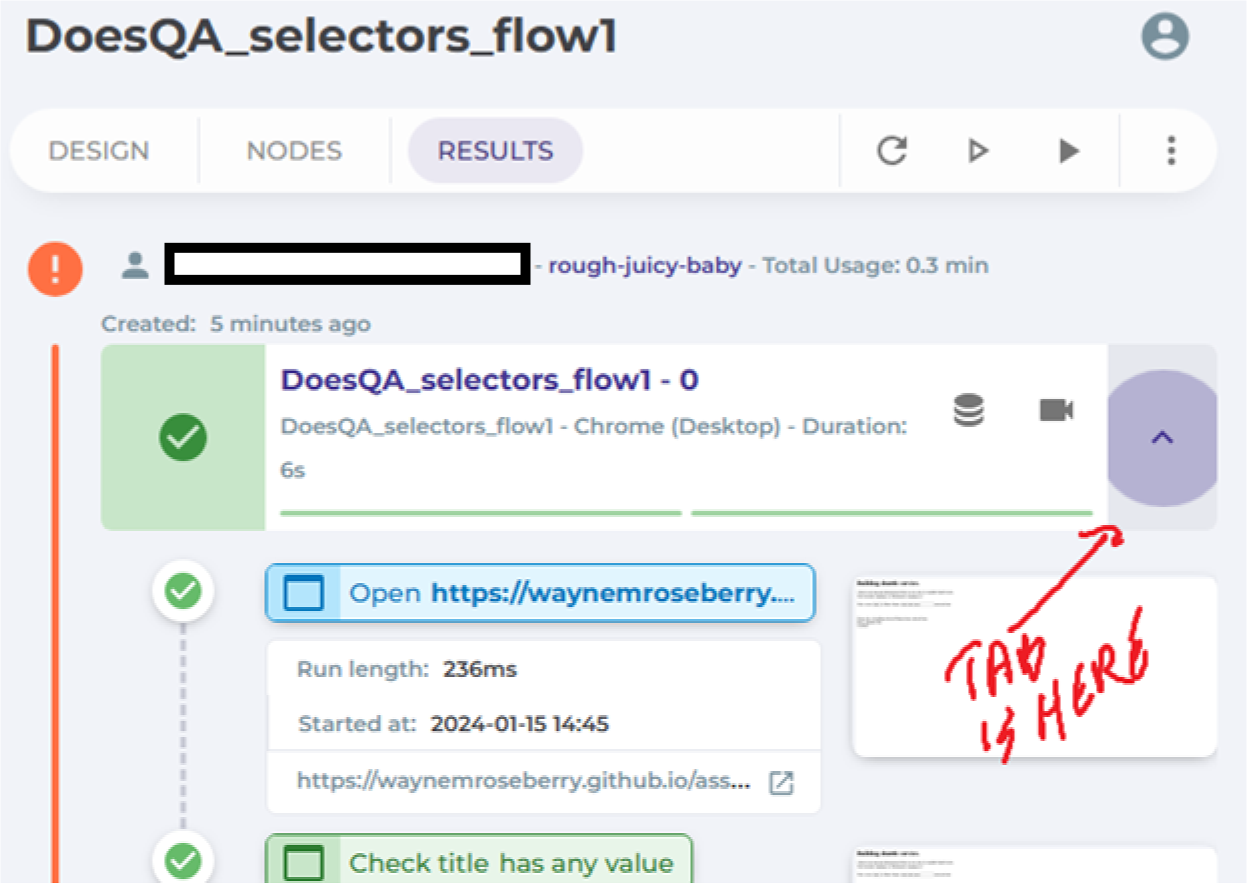
4.11 DEFAULT LOCATION OF HELPER COMMENTS OCCLUDE CONNECTION LINES, MAKING IT DIFFICULT TO SELECT THEM.

The default workflow configuration has the “Double-click to enter url” comment sitting above and occluding the dashed connection line, making it difficult to get the four way arrow icon necessary to move the connector line. This is easily addressed by moving the comment, but it was confusing the first time I tried.


Open https://waynemroseberry.github.io/assets/testform.html on Chrome (Desktop)
Double-click to enter url
Check title has any value
Touch ""
Add more Nodes here
hover cursor here doesn't yield ✥ icon.

4.12 TAB ORDER IN RUN REPORTS PAGE HITS THE “OPEN IN NEW WINDOW” ICON TWICE

- Run a workflow, or open the “Results” tab for an existing workflow
- For a result row, click the down arrow icon to expand the results
- Hit TAB until you have the expand/collapse button selected
- Hit TAB again – now the “open in new window” icon is selected with a semi-rectangular outline
- Hit TAB again – the “open in window” icon is selected again, this time with a gray background outline



DoesQA_selectors_flow1

[redacted] - rough-juicy-baby - Total Usage: 0.3 min

Created: 5 minutes ago

DoesQA_selectors_flow1 - 0

DoesQA_selectors_flow1 - Chrome (Desktop) - Duration: 6s

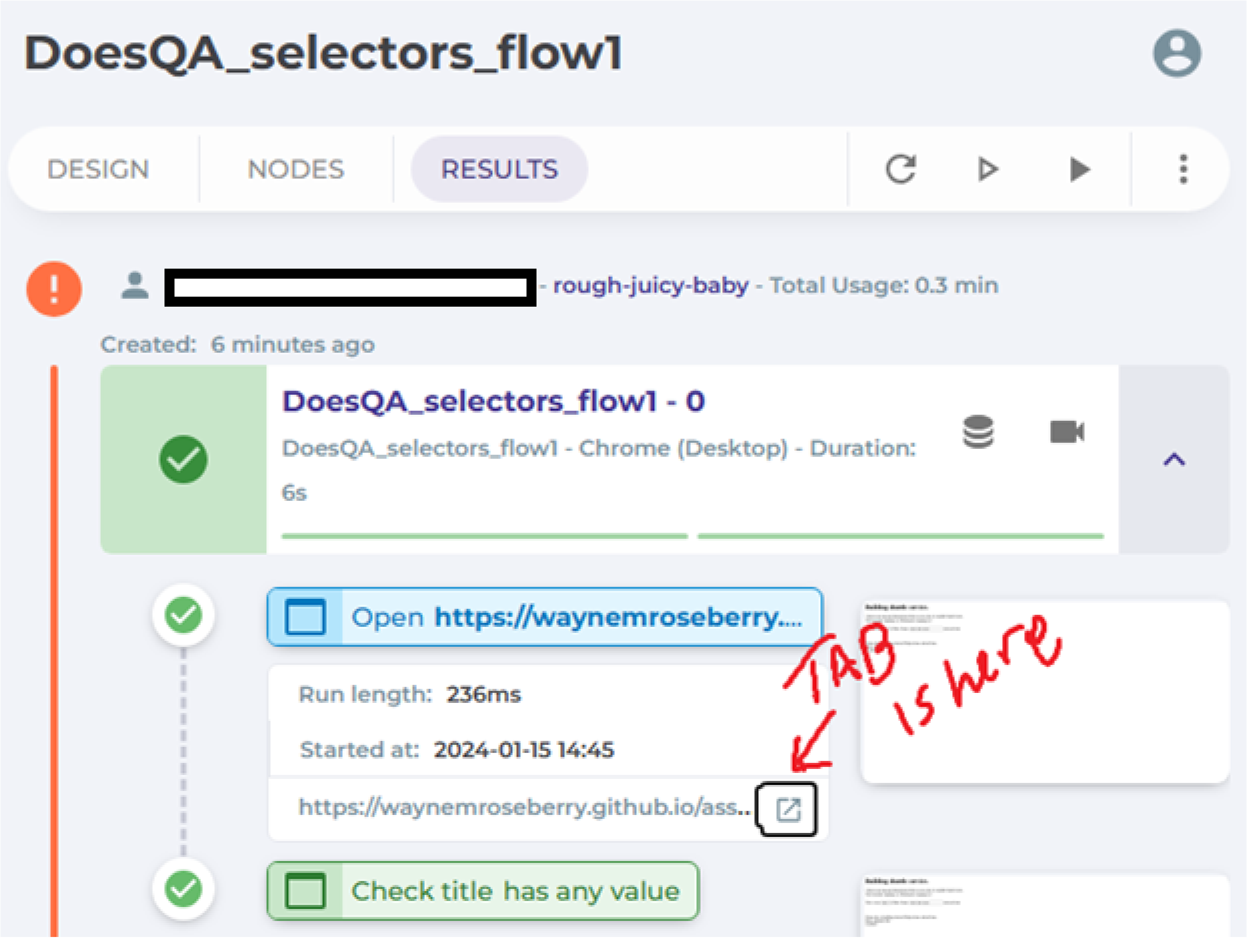
Open https://waynemroseberry....

Run length: 236ms

Started at: 2024-01-15 14:45

https://waynemroseberry.github.io/ass...

Check title has any value



DoesQA_selectors_flow1

[redacted] - rough-juicy-baby - Total Usage: 0.3 min

Created: 6 minutes ago

DoesQA_selectors_flow1 - 0

DoesQA_selectors_flow1 - Chrome (Desktop) - Duration: 6s

Open https://waynemroseberry....

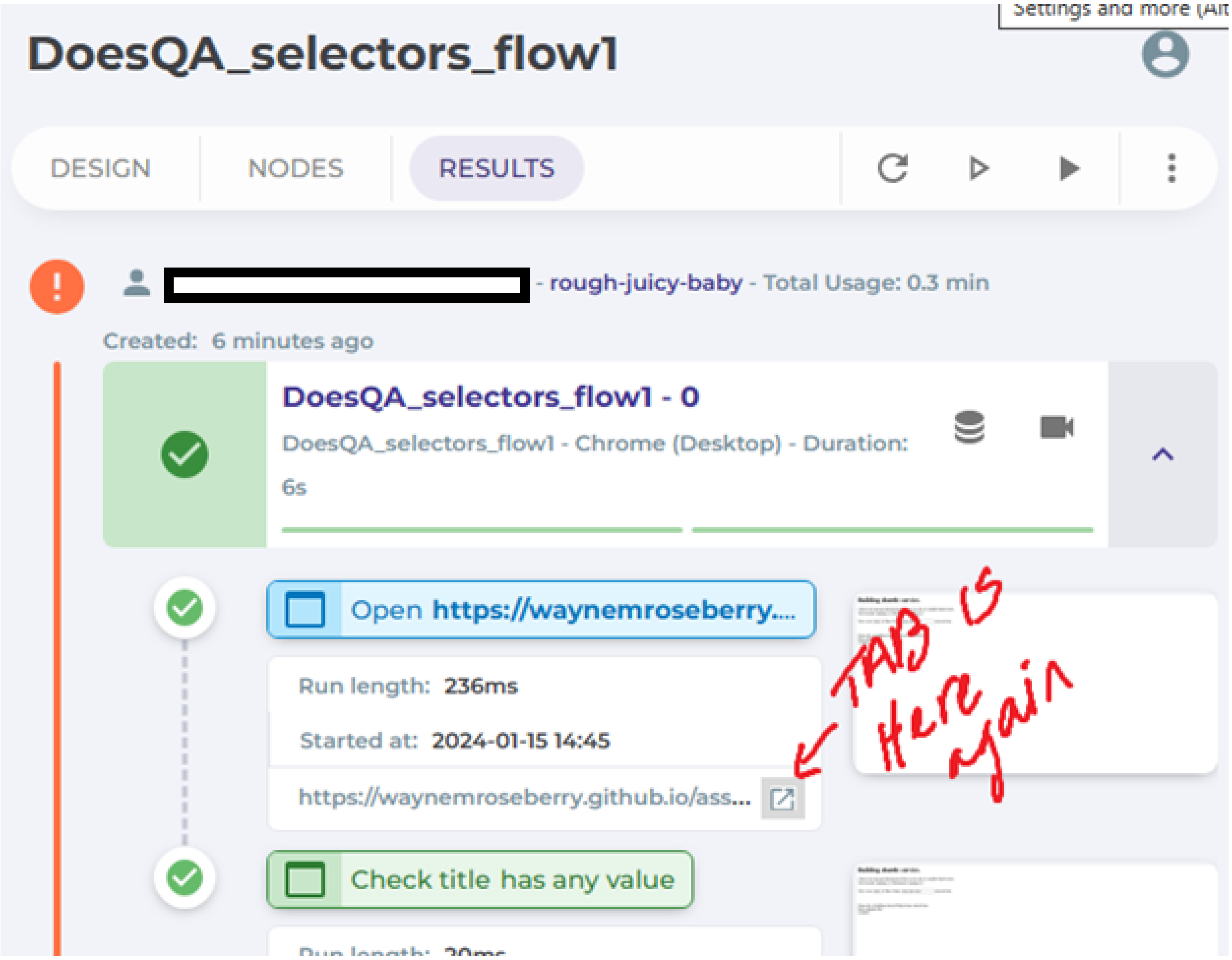
Run length: 236ms

Started at: 2024-01-15 14:45

https://waynemroseberry.github.io/ass...

TAB is here

Check title has any value



Settings and more (Alt

DoesQA_selectors_flow1

[redacted] - rough-juicy-baby - Total Usage: 0.3 min

Created: 6 minutes ago

DoesQA_selectors_flow1 - 0

DoesQA_selectors_flow1 - Chrome (Desktop) - Duration: 6s

Open https://waynemroseberry....

Run length: 236ms

Started at: 2024-01-15 14:45

https://waynemroseberry.github.io/ass...

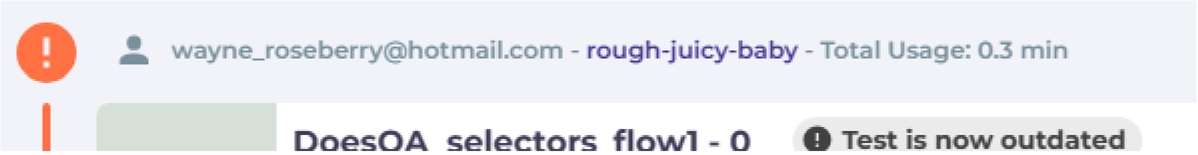
TAB is here again

Check title has any value

Run length: 20ms

4.13 Are the automatically selected names for runs necessary, do they run the risk of creating combinations some customers might object to?

One of my runs had the name "rough-juicy-baby" which is perhaps mildly amusing, but also kind of creepy sounding in a pickup line at a sleazy bar sort of way. My experience in other companies is that when there is some kind of non-controlled content out there, some customers take offense when the combinations accidentally stumble on some meaning, and will even go so far as to go to press with conspiracy theories the company is targeting them or some group they represent.



wayne_roseberry@hotmail.com - rough-juicy-baby - Total Usage: 0.3 min

DoesQA_selectors_flow1 - 0

Test is now outdated

4.14 Sometimes CTRL+Click deselects the prior selected item instead of adding the new item to the set of collected items

I do not have a repro for this. It happened more than once, but when I try to make it happen it does not.

1. Select an action node by click on it in the designer
2. Hold down CTRL, click a node connected to the first one

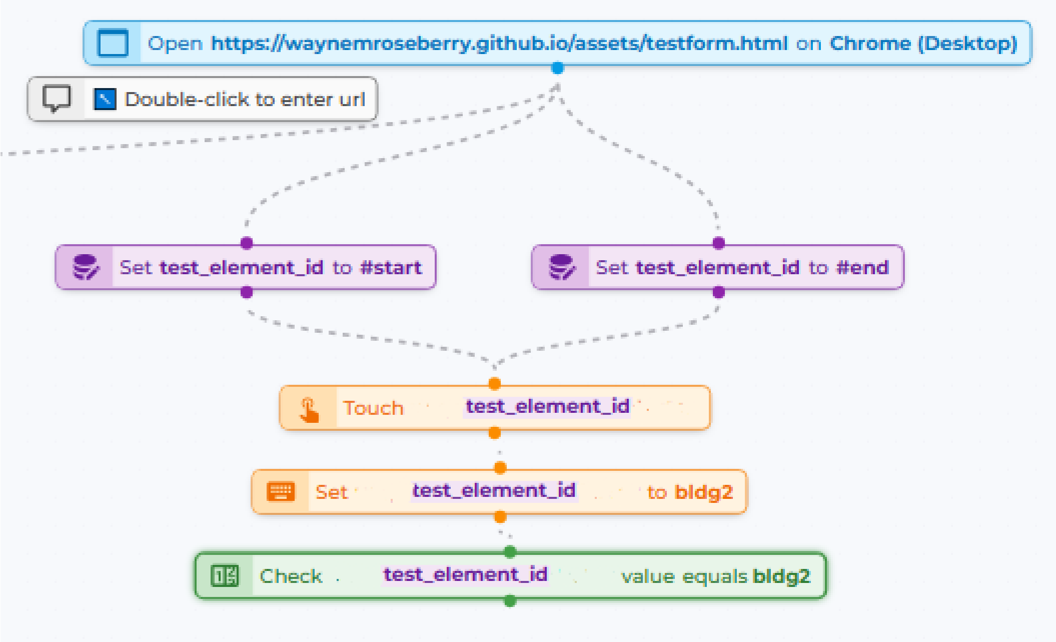
RESULT: the first node is deselected… sometimes…

5 Feature Suggestions

5.1 Feature Idea: Make element ids assignable to variables that can be branched in workflows to allow fast creation of common flows to different elements.

1. As a developer of a workflow, I have similar steps I want to use with different elements.
2. I would like to be able to express those similar steps via branching, or some similar mechanism where I create one set of sets and have alternative paths that pass a different element.

A visual of what that might look like is below. Maybe there is a way to do this sort of thing, but right now as far as I could tell if I have the same set of steps for multiple elements, I wind up doing a lot of duplication and a lot of clicking and editing to make that happen. The concept below allows me to save lots of duplicate sequences, lots of typing.



Open https://waynemroseberry.github.io/assets/testform.html on Chrome (Desktop)

Double-click to enter url

Set test_element_id to #start

Set test_element_id to #end

Touch test_element_id

Set test_element_id to bldg2

Check test_element_id value equals bldg2

5.2 It would be useful to be able to select just a branch and run only that branch and its descendants rather than the whole workflow

If one is working on a workflow with a lot of branches, sometimes it is useful for debugging reasons to run only branch and its sub branches rather than the whole workflow.

6 SUPPORTING MATERIALS

Raw testing notes can be found in the following location: WayneMRoseberry.github.io/assets/DoesQA selector testing notes january 15 2024.pdf at main · WayneMRoseberry/WayneMRoseberry.github.io