Self-Healing Questions for Scenario Editor

1. Could we have a list of what actions are included in self-healing, please?
2. I’m a little unsure why Assert Contains text can be included within self-healing but Assert Equal / Contains can’t be, could this be explained?
3. If it is the way that it’s coded could we look at alternatives, for example, could we have an Assert contains an option for attributes i.e. Assert Contains href.
4. Is it possible to include more complex actions that would self-heal? i.e. “Assert text and store value”

Responses:

Below list of assessors that can be SelfHealed:

- findElsByJqStr
- focus
- blur
- expand
- doSearch
- collapse
- formSubmit
- pointerOut
- pointerOver
- pointerEnter
- pointerLeave
- pointerDown
- pointerUp
- click
- click2
- clickIfFound
- rightClick
- mouseOver
- mouseMove
- mouseDown
- mouseUp
- drop
- drag
- nativeDrag
- nativeDrop

- clickXY
- mouseOverXY
- mouseMoveXY
- mouseDownXY
- mouseUpXY
- mouseDrag
- mouseDrop
- doubleCkick
- doubleClickXY
- toggleStyle
- setCaretPosition
- sentEnter
- sentText
- selectText
- clearText
- sendKeys
- setValue
- setText
- setValuetoEditor
- setSelected
- setChecked
- reOrderListForVue
- break
- paste
- **assertContrainsText**
- **assertDoesNotContainText**
- waitUntilVideoCompleted
- keyboardCommand
- setHTML

The two **asserts** that can be self-healed are bolded above, assertContainsText and assertDoesNotContainText.

Regarding #2: Self-healing can only update assessors. Assert equal uses a value and does not use an assessor. You would normally store “validatethisnumber” then you would assert equal “validatethisnumber” with a value of “5” or whatever the number is. Here is a screenshot example:

```
2  Store                |  submitdis  |  getAttribute(button("cancellation-submit"),"disabled")
3  Set FindOnlyEnabled  |  true       |
4  Assert Equal         |  submitdis  |  "disabled"
```

Where you store a “value” in this case the name is submitdis. Then you assert equal to the value that you stored.

The same logic applies to assert contains, but slightly different. Assert contains works similar to assert text, however only looks for something inside a stored value.

No		Action	Accessor	Value
1		Assert Contains	"This is a test to see if blue is in this text"	"blue"

Note that the usual use case for this action follows the above screenshot where you store a value with text. Then you can assert contains text inside what you stored. It's similar but different to assertContainsText and the reason why it can't self-heal is that assert contains uses a value as an assessor (or stored element) whereas assertContainsText uses an assessor.

Regarding how you can make more complex actions that could self-heal. I have an idea that will allow you to have more dynamic scripts and allow for more change, but won't be self-healed. The solution to the problem is creating SmartTags. Essentially telling AIQ that the element on the page is recognized as a value. Instead of having 1 set assessor, you could create a SmartTag that would pick up all assessors that look similar to “SmartTag 1” and allow your scripts to become more dynamic as you test without having to manually update all your assessors for your assert statements. It will just assist in picking up elements on the page without updating. Again, this is **not Self-Healing** but would improve the functionality of your scripts by making them more dynamic allowing for less maintenance (purpose of Self-Healing).

I had a meeting with our lead developer about your use case asking for more complex actions that would self-heal. He said at this moment we don’t currently support it, and it would be a major project to add self-healing to more assert’s as we would need AI deep learning to automatically pull assessors from the page that doesn’t currently exist. In the future, it's definitely a possibility, but at this moment and in the near future we don’t see this as a feature we will be able to support. We agree 100% that it would be awesome to include, just a massive project to start and finish.

Last updated: August 25, 2023

If you have any feedback on this topic, please contact us at documentation@appvance.ai

---

Revision #8
Created 2026-07-31 15:40:15 UTC by Admin
Updated 2026-11-17 21:59:23 UTC by AI_Mananger