

Add feature to Open-Source React Chrome extension

We are using the open-source UI.Vision RPA extension (“web macro recorder”). The goal of this project is to add array support for the executeScript_ **Sandbox** command.

The goal of this project is to add a feature to an Open-Source (React-based) Chrome extension for test automation.

Extension in Chrome store: <https://chrome.google.com/webstore/detail/ui-vision-rpa/gcbalfbdfieckjlnblleoemohcganoc>

To explain: In Version 7 of this extension the executeScript_ **Sandbox** command can no longer return **arrays**, as described here:

<https://forum.ui.vision/t/ui-vision-rpa-chrome-extension-migration-from-manifest-v2-to-manifest-v3/8957/27>

Screenshot of the issue:

The screenshot shows the UI.Vision RPA extension interface. At the top, there are buttons for 'test', 'Save', 'Record', 'Step', 'Play Macro', and a settings icon. Below this, there are two tabs: 'Table View' and 'Source View (JSON)'. The 'Table View' is active, displaying a table with columns: Command, Target, Value, and Ops. The table has three rows:

Command	Target	Value	Ops
1 open	https://google.com		// +
2 executeScript	return ["this", "works", "ok"];	a1	// +
3 executeScript_Sandbox	return ["does", "not", "work"];	a2	// +

Below the table is an 'Add' button and a form with fields for Command, Target, Value, and Description. A red arrow points from the 'Value' field of the 'executeScript_Sandbox' row to the 'Value' field of the 'Add' form.

At the bottom, there are tabs for 'Logs', 'Variables', 'Screenshots', 'CSV', and 'Visual'. The 'Variables' tab is active, showing a table of internal variables:

Name	Value
IRUNTIME	"3.32s"
ISTATUSOK	true
IURL	"https://www.google.com/"
A1	["this", "works", "ok"]
A2	["getter":(), "setter":(), "properties":{"0": "does", "1": "not", "2": "work"}, "proto": {"getter":(), "setter":(), "properties":{}}, "proto": {"getter":(), "setter":(), "properties":{}}, "proto": null, "class": "Array", "class": "Array"]

A red arrow points from the 'Value' field of the 'Add' form to the 'A2' variable in the 'Variables' table.

^You see in the screenshot how the executeScript commands returns an array, but `executeScript_sandbox` returns an internal structure (that is the bug to be fixed!).

For running the Javascript inside executeScript_**Sandbox** the extension using this open-source

<https://neil.fraser.name/software/JS-Interpreter/>

The challenge is that the js-interpreter also returns internal data structure for object/array (see screenshot below).

Instead, it should work like the executeScript command which returns array properly (but requires a website to run, that is why it is no solution for us).

So, we see 3 approaches to solve this:

- Make some **changes to UI Vision** to convert the returned structure to an array
- And/or make some **changes to the JS-Interpreter** library used
- Or **find another JS library** that works better

Other solutions are also welcome. The issue is solved once the array handling of `executeScript_sandbox` works just like `executeScript`.

Building the extension/Source code

The project uses React Node V12.16.1 and NPM V6.13.4.

Build video: <https://www.screencast.com/t/PN32qj29> (on Windows, but same on Mac/Linux)

The ready-to-compile full source of the project is available before your bid if you want to check. The challenge is that this is a rather powerful extension so we assume it takes some time to understand the source code.

Expected Result

This milestone is completed once the array handling of `executeScript_sandbox` works just like `executeScript`.

Of course, no regression issues/new bugs should be introduced with this fix ;-)