

Record the Order Randomized Pages are Displayed

Scripting Solutions

Additional scripting solutions will be added in the future. Please reach out to Alchemer with comments and suggestions on solutions you'd like to see via the link [here](#).

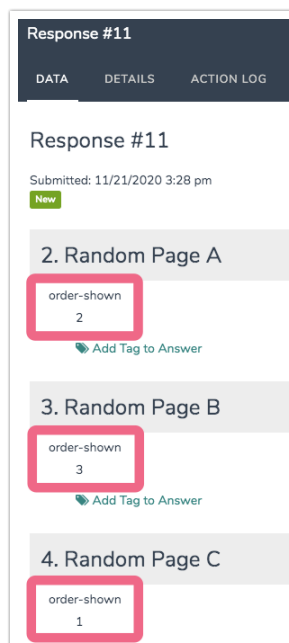
Scripting and Other Custom Solutions

We're always happy to help you debug any documented script that is used as is. That said, we do not have the resources to write scripts on demand or to debug a customized script.

If you have customization ideas that you haven't figured out how to tackle, we're happy to be a sounding board for Alchemer features and functionality ideas that might meet your needs. Beyond this, check out our [Professional Services](#); these folks have the scripting chops to help you to achieve what you are looking for!

Goal

Record the order [Randomized Pages](#) were presented to a respondent. The below example shows the order three randomized pages were presented in response #11:



Effort: ✓ ✓ ✓

Solution

Step 1: Randomize

Randomize pages in your survey. Follow the aforementioned link for more information on setting up a random page display.

Step 2: Track order each page was presented

Add a **Hidden Value Action** titled *order-shown* to each randomized page to track the order it was shown among the pages being tracked. Users may add this to each page being tracked or just the pages they want to know the order they appeared in the group of tracked pages.

Step 3: Add Javascript to survey

1. Add the Javascript code below to the **Style tab > HTML/CSS Editor (lower right of page) > Custom Header**.
2. Add the *Page IDs* for the randomized pages you want to track to the **TRACKED_PIDS** array highlighted in the code (the order they appear in the array doesn't matter).

```
<script>
/* Alchemer v01

Track the order randomized pages are displayed.

Documentation and updates: https://help.alchemer.com/help/record-order-randomized-pages-are-presented
*/

'use strict';

document.addEventListener("DOMContentLoaded", function() {

const TRACKED_PIDS = [ 1, 3, 5 ]

// *****
// * no changes needed below *
// *****

const HVA_TITLE_TO_SAVE_TO = "order-shown"
const SHOWN_PIDS = `[page("pagepath")]`.split(',').map(s => parseInt(s)) // PIDs array with current PID at the end
const LOG = true

/**
 * Test boolean value and alert() and throw Error if it's false
 *
 * bool {t/f} value to test
 * msg {string} message to throw in new Error
 */
const assert = (bool, msg) => {
  if (!bool) {
    alert(msg)
    console.error(msg)
    throw new Error(msg)
  }
}

/**
```

```

/****
 * Get an element based on it's Question ID
 *
 * qid {integer/string} question ID
 * section = "element" {string} the final section of the element id
 * return {element} looks for id's in the form: "sgE-1234567-12-123-element"
 */
const getElemByQid = (qid, section = "element") => {
  const surveyInfo = SGAPI.surveyData[Object.keys(SGAPI.surveyData)[0]]
  const id = "sgE-" + surveyInfo.id + "-" + surveyInfo.currentpage + "-" + qid + "-" + section
  const elem = document.getElementById(id)
  assert(elem, "Javascript error: can't find element with id = " + id)
  return elem
}

/****
 * Get the Question ID based on the question / hidden value title
 *
 * title {string} question title
 * return {int / null} the question ID or null if title not found
 */
const getQidByTitle = (title) => {
  const questionsObj = SGAPI.survey.surveyObject.questions
  const qid = Object.keys(questionsObj).find(key =>
    questionsObj[key].title.toLowerCase() === title.toLowerCase())
  return qid || null
}

/****
 * Get the current Page ID
 *
 * return {int} the current page ID
 */
const getPid = () => parseInt(SGAPI.survey.pageId)

/****
 * Determine if we're on a tracked page
 */
const onTrackedPage = (trackedPids, pid) => trackedPids.includes(pid)

/****
 * Determine if page order was already saved and we're viewing the page again
 *
 * title {string}
 * return {t/f} True if page order was already saved
 */
const wasOrderSaved = (title) => {
  assert(getQidByTitle(title), `Javascript: This page order is being tracked and needs a Hidden Value Action titled: ${
HVA_TITLE_TO_SAVE_TO}`)
  const val = getElemByQid(getQidByTitle(title)).value.trim()
  return val && val !== "not-set"
}

/****
 * Determine if there is an HVA to save the order shown on this page
 *
 * title {string}
 * return {t/f} True if there is an HVA to save the order shown.
 */
const canSaveOrder = (title) => {
  const qid = getQidByTitle(title)
  if (!qid)
    console.warn(`\n\n***\nJavascript: page id '${getPid()}' is part of a tracked group of pages but does not have a Hid
den Value Action named '${HVA_TITLE_TO_SAVE_TO}' to save the order it was shown.\n***\n\n`)
  return qid !== null
}

```

```

}

/**
 * Save the order this page was shown
 *
 * title {string} title of the Hidden Value Action to save to
 * orderShown {int} the 1-based order this page was shown in the page randomization group
 */
const saveOrderShown = (title, orderShown) =>
  getElemByQid(getQidByTitle(title)).value = orderShown

/**
 * Get the 1-based order this page was shown in the randomized page group
 *
 * shownPids {array of int} array of PIDs shown up to the current PID
 * trackedPids {array of int} what randomized PIDs we're tracking
 * currentPid {int}
 */
const getOrderShown = (shownPids, trackedPids, currentPid) => {
  assert(shownPids.slice(-1)[0] === currentPid, `Javascript: expected shownPids to end with currentPid: ` + shownPids + ' -- ' + currentPid)

  let order = 0
  shownPids.forEach(pid => {
    if (trackedPids.includes(pid))
      order++
  })

  assert(order, "Javascript: page order wasn't determined")
  return order
}

/**
 * main()
 */

try {

  if ( onTrackedPage(TRACKED_PIDS, getPid())
    && canSaveOrder(HVA_TITLE_TO_SAVE_TO)
    && !wasOrderSaved(HVA_TITLE_TO_SAVE_TO))
    saveOrderShown(HVA_TITLE_TO_SAVE_TO, getOrderShown(SHOWN_PIDS, TRACKED_PIDS, getPid()))

} catch(err) {
  console.error(err)
}
})
</script>

```

Related Articles