Soft Quotas

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

Allow respondents to complete their response once past questions with quotas are presented, even if the quotas are later met by another respondent. In essence, preventing a situation where a respondent is many pages into their response but is disqualified because another respondent completes the survey and fills a quota from a question early in the survey.

Effort: ✓ ✓ ✓

Solution

The script below uses a Webhook Action to check if the quotas have been filled. If one has, the name of the quota is saved to a Textbox and the response is disqualified. If not, the respondent is allowed to proceed because the quota is set to continue collecting responses when full.

Your admin **must** provide you API access for this solution.

Limitations

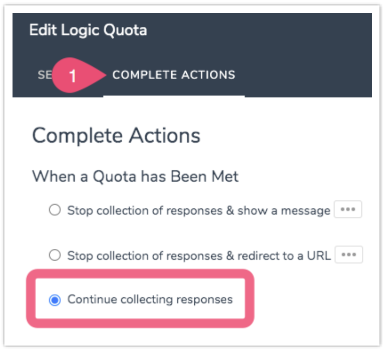
- The quota questions must be Radio Buttons or Checkboxes. Let us know if you need another question type.
- Works with Logic Quotas, not the Overall Quota.

Step 1: Add Quota

1. **Create logic quota**, for example a quota for **Gender=Male** named **Male - soft quota**.

2. Set the **quota limit**.

3. Set the quota to **Continue collecting responses** when filled:



Complete Actions

When a Quota has Been Met

- ○ Stop collection of responses & show a message
- ○ Stop collection of responses & redirect to a URL
- ● Continue collecting responses

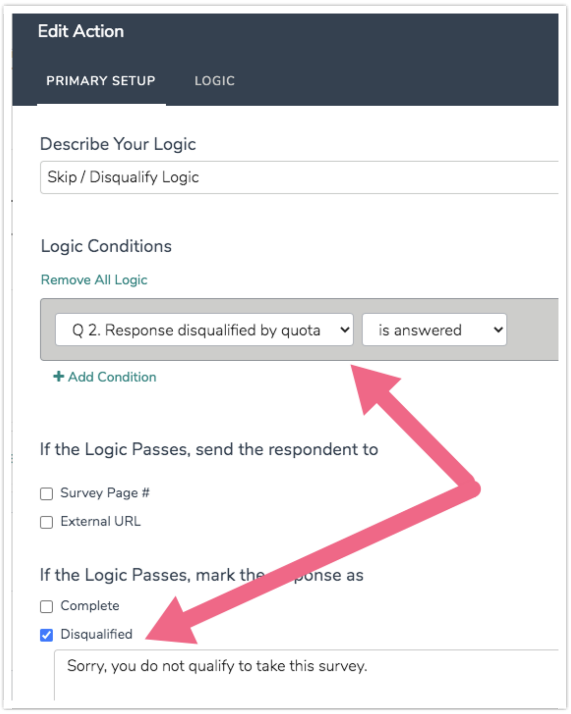
Step 2: Add new page

1. Create a **new page** after the quota question has been asked.

2. Set the page's **Layout** > **CSS Class Name** to **sg-hide.**

3. Add a **Textbox question** to the page titled **Response disqualified by quota**. If the respondent meets a quota that is filled the script will enter the quota name in this Textbox.

4. Add **Skip/Disqualify** logic to the page to disqualify the response if the textbox above is-answered



Describe Your Logic: Skip / Disqualify Logic

Logic Conditions

Q 2. Response disqualified by quota | is answered

If the Logic Passes, send the respondent to

- ☐ Survey Page #
- ☐ External URL

If the Logic Passes, mark the response as

- ☐ Complete
- ☑ Disqualified

Sorry, you do not qualify to take this survey.

Step 3: Add Webhook Action to get all quotas

Add a new Webhook Action to the TOP of the page above with the settings below. This will get counts for all quotas.

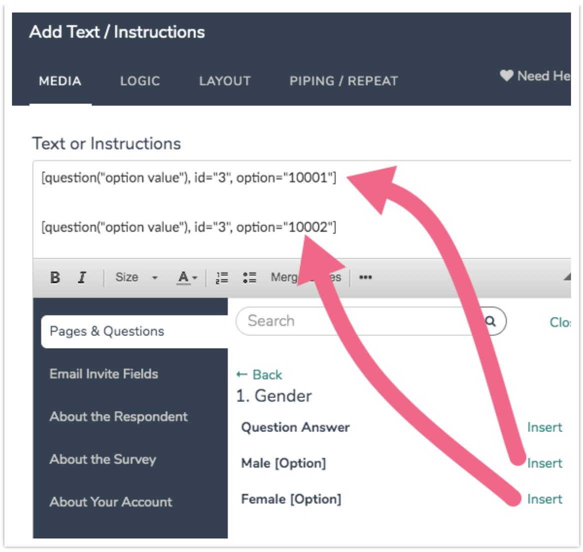
Method	Get
URL	https://api.alchemer.com/v5/survey/[survey("id")]/quotas?api_token=***&api_token_secret=*** Enter your API Key and Token Secret above. • For **EU** data center change the URL to: api.alchemer.eu • For **CA** data center change the URL to: api.alchemer-ca.com
Fields to Pass	Post Custom Fields, but no other selections in this section
Asynchronous Connect	No
What do you want to do with the data/content returned from the URL?	Display it

Step 4: Setup Javascript Action

Add the Javascript Action below.

Setup the array **DATA_TO_TEST** to associate the quotas with the quota questions. The **quotaName** must exactly match the name you gave the soft quota and the **answer** must be a merge code for the specific option being tested.

To find these merge codes go to any text element and choose to add a merge code, select the option that applies to the soft quota, and copy the resulting merge code to the Javascript.



Text or Instructions

```
[question("option value"), id="3", option="10001"]

[question("option value"), id="3", option="10002"]
```

1. Gender

Question Answer

Male [Option]

Female [Option]

```
/* Alchemer v1

  Apply soft quotas to disqualify respondents.

  See documentation: https://help.alchemer.com/help/soft-quota
*/

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

 const DATA_TO_TEST = [

  // Male
  {
   quotaName: 'Male - soft quota',
   answer: `[question("option value"), id="3", option="10001"]`
  },

  // Female
  {
   quotaName: 'Female - soft quota',
   answer: `[question("option value"), id="3", option="10002"]`
  },

 ]


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


 const LOG = true


 /*
   QUOTA OBJECT, returned from getLiveQuotas()
       {
        "id": "2600",
        "name": "Male - soft quota",
        "description": "",
        "responses": "0",
        "limit": "3",
        "distributed": "false"
```

```
      distributed : false
     }
 */

/***
 * Helper to display error dialog and throw new Error
 */
const assert = (bool, msg) => {
 if (!bool) {
   alert(msg)
   throw new Error(msg)
 }
}

/***
 * Helper: Get a SurveyGizmo element on the page.
 *        Ex: In survey 1234567 on page ID 12 the call getSgId(123, "element")
 *            returns HTML element for "sgE-1234567-12-123-element"
 */
function getSgElemByQID(qid, oid = "element") {
 let surveyInfo = SGAPI.surveyData[Object.keys(SGAPI.surveyData)[0]]
 let id = "sgE-" + surveyInfo.id + "-" + surveyInfo.currentpage + "-" + qid + "-" + oid
 let elem = document.getElementById(id)
 assert(elem, "Javascript error: can't find element with id = " + id)
 return elem
}

/***
 * getLiveQuotas()
 *
 * Parse data from webhook call.
 *
 * {return} Array of quota objects or null for error or no quotas
 */
const getLiveQuotas = () => {

 // check that .sg-http-content has data
 const quotasJSON = document.querySelector('.sg-http-content').innerText
 if (!quotasJSON) {
   console.error("ERROR getQuotas(), no .sg-http-content")
   return null
 }

 // check that webhook call was successful
 const parsed = JSON.parse(quotasJSON)
 if (!parsed.result_ok === "ok") {
   console.error("ERROR getQuotas(), result not ok: ", parsed)
   return null
 }

 if (LOG) console.log("parsed.quotas = ", parsed.quotas)

 // check that there are quotas for this survey
 return parsed.quotas || null
}

/***
 * applyQuotas()
 *
 * {dataToTest} array of objects to test against quotas
 * {liveQuotas} live quotas
 * {return} First quota object to apply or null if no quotas apply
 */
const applyQuotas = (dataToTest, liveQuotas) => {

 for (let i = 0; i < dataToTest.length; i++) {
    for (let i = 0; i < dataToTest.length; i++) {

      // if the data applies to the quota
      if (dataToTest[i].answer) {

        if (LOG) console.log("testing quota ", dataToTest[i].quotaName)

        const quota = liveQuotas.find((q) => q.name === dataToTest[i].quotaName)

        if (!quota) {
          const err = `ERROR, Javsacript error - quota not found: ${dataToTest[i].quotaName}`
          console.error(err)
          alert(err)
          return null
        }

        if (LOG) console.log(`-- responses = ${quota.responses} -- limit = ${quota.limit}`)

        // quota has been met!
        if (quota.limit <= quota.responses) {
          if (LOG) console.log("-- QUOTA HAS BEEN MET")
          return quota
        }
      }
    }
    return null
  }

  /***
   * saveDqReason()
   *
   * Save info on the quota that is full and causing the DQ
   *
   * {dqQuota} - the quota the is DQ'ing the respondent
   */
  const saveDqReason = (dqQuota) => {
    if (LOG) console.log(`DQ quota being applied - ${dqQuota.name}`)

    // assumes that the reason is the ONLY Textbox on the page
    const dqReasonElem = document.querySelector('input.sg-input')
    if (LOG) console.log("dqReasonElem = ", dqReasonElem)

    dqReasonElem.value = `${dqQuota.name} -- ${dqQuota.responses} \/ ${dqQuota.limit}`
  }

  /***
   * main()
   */

  const liveQuotas = getLiveQuotas()

  if (liveQuotas) {

    const dqQuota = applyQuotas(DATA_TO_TEST, liveQuotas)

    if (dqQuota)
      saveDqReason(dqQuota)

  } else {
    alert("Javascript error - no live quotas to process")
    console.error("ERROR - no live quotas to process")
  }

  // submit and move to next page
  (document.querySelector("#sg_NextButton") || document.querySelector("#sg_SubmitButton")).click()
})
```

```
})
```

Related Articles