Share Status of Quotas from a Survey

> ## 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.

Goal

Share the status of a survey's quotas with other members of an organization or team.

Quota	Current	Limit	% filled
Boost: Female 18-24	18	35	51%
Boost: Female 25-34	65	65	100%
Boost: Female 35-44	55	55	100%
Boost: Female 45-54	55	55	100%
Boost: Female 55-64	40	40	100%
Boost: Male 18-24	17	35	49%
Boost: Male 25-34	59	65	91%
Boost: Male 35-44	55	55	100%
Boost: Male 45-54	55	55	100%
Boost: Male 55-64	40	40	100%
Targeted: Males, 18-24	100	100	100%
Targeted: Males, 25-34	117	200	59%
Targeted: Males, 35-44	200	200	100%

Effort: ✔ ✔ ✔

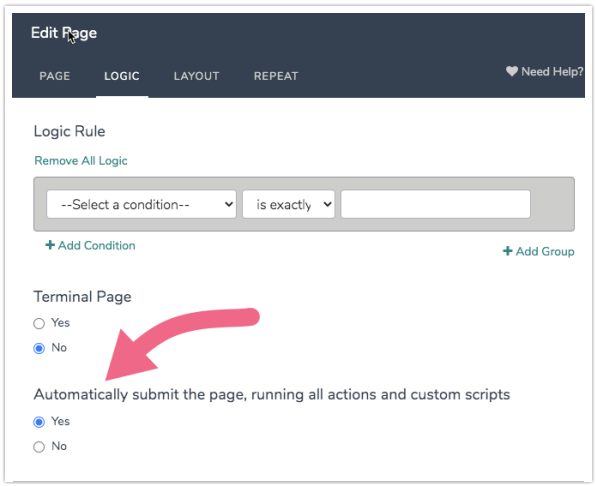
Solution

This solution requires users create a new survey to display the quota counts of the primary survey. Users share a link to this survey to the stakeholders needing to monitor the primary survey quotas. Only Logic and Distributed Logic quotas display, ***not*** the Overall Quota.

Step 1: Create a New Survey

1. Create a survey.

2. Set the first page to **auto-submit.**



3. Add the following code to the survey via **Style tab > HTML/CSS Editor (lower right of page) > Custom CSS to format the quotas:**

```
/* Format quotas */
.sg-http-content {
  display: none;
}

th, td {
  border: 1px solid black;
  padding: 10px;
  text-align: right;
}

th {
  background-color: lightgray;
}

table {
  border-collapse: collapse;
}
```

4. Add a new Page #2

Step 2: Add a **Webhook Action** to the top of the new Page #2 with the following settings:

Setting	Value
Method	Get

URL	api.alchemer.com/v5/survey/1234567/quotas?api_token=***&api_token_secret=*** Replace api.alchemer.com with your datacenter: • US: api.alchemer.com • CA: api.alchemer-ca.com • EU: api.alchemer.eu Replace 1234567 with the survey ID containing the quotas Replace the end with your API token Choose: https 
What do you want to do with the data/content returned from the URL?	Display it

Step 3: Add a Javascript Action with the code below

```
/* Alchemer v01

   Display the quotas returned from a Webhook Action

   Documentation and updates: https://help.alchemer.com/help/share-status-of-quotas-from-a-survey
*/

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

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

  const LOG = true

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

  /***
   * Sort two strings in alpha order
```

```
  * Sort two strings in alpha order
  */
 const alphaSortFn = (a, b) => a.localeCompare(b)

 /***
  * Sort two quotas in alpha order
  */
 const quotaSortFn = (a, b) => alphaSortFn(a.name, b.name)

 /***
  * Parse all quotas from webhook call.
  *
  * return (quota array) array of quota objects with format:
        {
         "id": "2600",
         "name": "fill-Male-A",
         "description": "",
         "responses": "0",
         "limit": "100",
         "distributed": "false"
        }
  */
 const getAllQuotas = () => {

   // get webhook call result elem
   const webhookResultElem = document.querySelector('.sg-http-content')

   // test
   if (!webhookResultElem)
     throw new Error("Can't find Webhook data. Check that Webhook Action is on this page and has 'Display it' select
ed")

   if (LOG) console.log("\n---- Raw return from webhook ----\n" + webhookResultElem.innerText + "\n---- ^^^^^^^^
^ ----\n")

   const quotasJSON = webhookResultElem.innerText

   if (!quotasJSON)
     throw new Error("Webhook call failed with no data returned")

   const parsed = JSON.parse(quotasJSON)

   // check webhook return
   if (!parsed.result_ok) {
     if (parsed.code === 403 && parsed.message.startsWith('POST calls are not allowed'))
       throw new Error("Webhook call failed, check that the Webhook Action's Method is set to 'Get'")
     throw new Error("Webhook call failed, check that you entered the correct datacenter, API token, and Survey ID: "
+ parsed.code + ' - ' + parsed.message)
   }
   if (LOG) console.log("getAllQuotas() = ", parsed.quotas)
   if (!parsed.quotas.length)
     throw new Error("No quotas were found, check that Webhook is referencing the correct survey ID")

   return parsed.quotas
 }

 /***
  * Get an index for a continuous value.
  *
  * Example:
  *     ['green','orange','red'][getIndex(0.2, 1, 3)] // 'green'
  *
  * Example:
  *     getIndex(-25, 100, 3) // 0
  *     getIndex(0,   100, 3) // 0
  *     getIndex(40,  100, 3) // 1
```

```
     getIndex(40,  100, 3) // 1
 *    getIndex(80,  100, 3) // 2
 *    getIndex(100, 100, 3) // 2
 *    getIndex(200, 100, 3) // 2
 *
 * x (int/float) the continuous value
 * max (int/float) index will be calculated based on 0-max, however actual x value can be outside this range
 * numIndexes (int) the return will be in the range 0-(numIndexes - 1)
 */
const getIndex = (x, max , numIndexes) => {
  const index = x / (max / numIndexes)
  // min/max to keep it in the range of 0 to (numIndexes-1)
  return Math.max(
          Math.min(
            Math.floor(index),
            (numIndexes - 1)),
          0)
}

/***
 * Get an HTML formatted display for the % quota filled column
 *
 * responses (int) number of responses received
 * limit (int) the quoto limit
 * return (html text) calculated percent with color formatting for red, orange, green
 */
const formatPercentFilled = (responses, limit) => {
  const percent = responses / limit * 100
  const color = ['red', 'orange', 'green'][getIndex(percent, 100, 3)]
  return `<span style="color:${color};">${(percent).toFixed(0)}%</span>`
}

/***
 * Show quotas
 */
const showQuotas = () => {

  let quotas = getAllQuotas()

  // log
  if (LOG) {
    console.log("quotas = ")
    console.log(
      quotas.map((quota) => `  ${quota.name} - ${quota.responses}` )
      .sort(alphaSortFn)
      .join('\n'))
  }

  // table inside a div that provides css layout of a Text element
  let tableHTML =
     '<div class="sg-question sg-type-instruction">'
    +  '<table style="margin-left: auto;  margin-right: auto;">'
    +    `<tr><th>Quota</th><th>Current</th><th>Limit</th><th>% filled</th></tr>`
    +    quotas
           .sort(quotaSortFn)
           .map(quota => `<tr><td>${quota.name}</td><td>${quota.responses}</td><td>${quota.limit}</td><td>${for
matPercentFilled(quota.responses, quota.limit)}</td></tr>`)
           .join('')
    +  '</table>'
    +  '</br>'
    + '</div>'

  let elemWebhook = document.querySelector('.sg-http-content')
  elemWebhook.insertAdjacentHTML('afterend', tableHTML);
 }
```

```
  /**
   * main()
   */
  try {
    showQuotas()
  }
  catch (err) {
    console.error("ERROR.message ", err.message)
    console.error("ERROR ", err)
    alert("Javascript error: " + err.message)
  }
})
```

Related Articles