Automatic Save and Continue from Share Link

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

Automatically include a Save and Continue function without requiring a respondent to enter an email address. When following the survey **Share Link**, respondents return to the in-progress response *or* the **Thank You page** if their survey response is completed.

Effort: ✔ ✔ ✔

Special Considerations

This is a cookie-based solution with the following characteristics and keys for success:

- Functions when a respondent uses the same computer and browser.
- A response is created as soon as the **Share Link** is clicked.
- If two respondents share the same computer/device they **will not be able** to create multiple responses.
- To allow a respondent to start a new response they can open an incognito window, clear cookies, or use a different browser/computer.
- The survey has a slight delay displaying the first page.

Solution

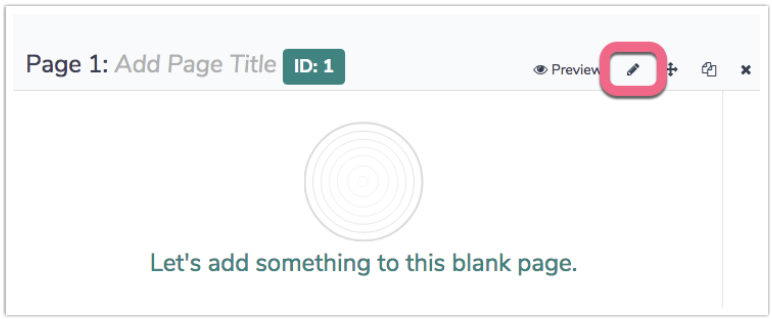
Step 1: Add CSS

1. Add the following CSS to via the **Style tab** in the top toolbar when editing a survey. Click **Style > HTML/CSS Editor (found in lower right side of the page) > Custom CSS.** Paste in the following code on the CSS Tab:

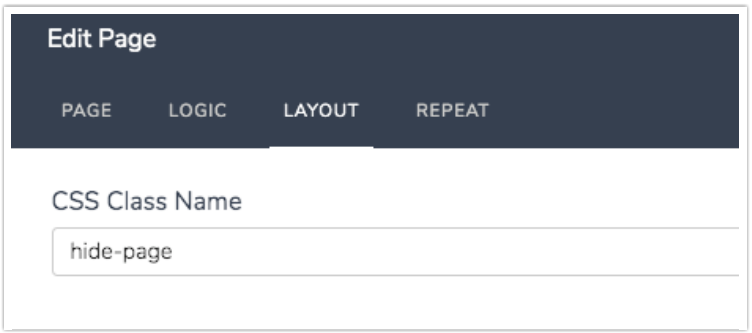
```
.hide-page {
  visibility: hidden;
}
```

Step 2: Setup first page of survey

1. On the first page of the survey, select the ***edit pencil*** found at the top right of the page:



2. On the popup window that displays, select the **Layout tab.** Paste in hide-page in the **CSS Class Name field**



3. Copy the script found below and paste it into a Javascript Action on the first page:

```
/* Alchemer v01

  Automatic Save and Continue, return user to their existing response.

  Documentation and updates: https://help.alchemer.com/help/automatic-save-and-continue-javascript

  How it works:

   -- First time the Share Link is accessed:
      - create a sguid, save as a cookie, start a new response with that sguid
      - See 'sguid' documentation: https://help.alchemer.com/help/sguid-the-url-variable

   -- Second time the Share link is accessed:
      - check if there is a sguid cookie, return to that reponse
      - add URL variable returned-to-session=true
```

```
    - add URL variable returned-to-session=true

    -- Responses use a URL variable is in the form:
                sguid-1234557=dfERgsd1223dfreGER
       where 1234567 is the survey ID and a 25 character random code serves as the SGUID

    -- DEV NOTE: Setting the first page to "sg-hide" causes problems when clicking Back
       from the 2nd page.
  */

// ****************************
// ** No changes needed below **
// ****************************

'use strict';

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

  const LOG = false

  /***
   * Prepare browser to create a new response or return to user's existing response
   *    on next reload() by adding sguid URL variable.  Pushes to window.history.
   *
   * squid {string} the respondent's sguid
   * returningToSession {t/f} are we returning to an existing session
   */
  const updateURL = (sguid, returningToSession) => {
    if (LOG) console.log(`setup url refresh with sguid = ${sguid}, returningToSession = ${returningToSession}`)
    var newurl =
      window.location.protocol +  "//" +
      window.location.host +  window.location.pathname +
      '?sguid=' + sguid +
      ((returningToSession) ? '&returned-to-session=true' : '')
    window.history.pushState({ path: newurl }, '', newurl);
  }

  /***
   * Get a variable from the URL
   *
   * name {string} name of URL variable we're looking for
   * return {string/null} the varaible value or null
   */
  const getUrlParam = (name) => (new URL(window.location.href)).searchParams.get(name)

  /***
   * Create a random code to serve as a unique sguid
   *
   * length {int} length of the sguid to create
   * return {string} random sguid
   */
  const createSguid = (length) => {
    var result = ''
    var chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'
    for (var i = 0; i < length; i++)
      result += chars.charAt(Math.floor(Math.random() * chars.length))
    return result
  }

  /***
   * Get the survey ID
   *
   * return {string} the survey ID
   */
  const getSID = () => SGAPI.surveyData[Object.keys(SGAPI.surveyData)[0]].id
```

```
/***
 * Get the survey's sguid variable name
 *
 * return {string} the cookie name in the form: sguid-1234567
 */
const getSguidCookieName = () => 'sguid-' + getSID()

/***
 * Get a cookie value
 *
 * name {string} case-sensitive name of cookie to get
 * return {string/null} cookie value or null
 */
const getCookie = (name) => {
  let rawArray = document.cookie.split(';')
  for (let i = 0; i < rawArray.length; i++) {
    const pair = rawArray[i].split('=')
    if (pair[0].trim() === name) {
      return decodeURIComponent(pair[1].trim())
    }
  }
  return null
}

/***
 * Set a cookie value
 *
 * name {string} case-sensitive name of cookie to set
 * value {string} value to set
 * daysToLive = 365 {int} how long should cookie live
 * return {string/null} cookie value or null
 */
const setCookie = (name, value, daysToLive = 365) => {
  // Encode value in order to escape semicolons, commas, and whitespace
  var cookie = name + "=" + encodeURIComponent(value) + "; max-age=" + (daysToLive * 24 * 60 * 60)
  if (LOG) console.log("adding cookie = ", cookie)
  document.cookie = cookie
}

/***
 * Show the current page by removing hide-page css classname
 */
const showPage = () => document.querySelector('body').classList.remove('hide-page')

/***
 * main()
 */

const urlSnc = getUrlParam('snc')
// Don't do anything if this is an Edit Link
if (urlSnc) {
  if (LOG) console.log("Returning to existing response via an Edit Link")
  showPage()
  return
}

const urlSguid = getUrlParam('sguid')
let cookieSguid = getCookie(getSguidCookieName())

if (urlSguid && urlSguid === cookieSguid) {
  if (LOG) console.log("We're on the current response!")
  showPage()
} else {
  if (LOG) console.log("Preparing to reload page")
  // if this is a new response create a sguid for it
```

```
    // if this is a new response create a sguid for it
    if (!cookieSguid) {
      if (LOG) console.log("-- creating a new sguid")
      cookieSguid = createSguid(25)
      setCookie(getSguidCookieName(), cookieSguid)
      updateURL(cookieSguid, false)
    } else {
      if (LOG) console.log("-- returning to existing response")
      updateURL(cookieSguid, true)
    }
    location.reload()
  }
})
```

Related Articles