Account Webhooks: Receive Survey and Response Notifications

Last Modified on 01/09/2019 4:47 pm EST |

Account Notification Webhooks allow you to receive notifications regarding activity in your account via an HTTP POST to an external URL.

Webhooks are typically used by developers to register endpoints that will be notified when activities of their choosing take place. This allows for a more seamless integration with SurveyGizmo that limits the need for repetitive polling of the API.

If you are looking to use Webhooks to send data to and from surveys visit our Survey Webhooks documentation.
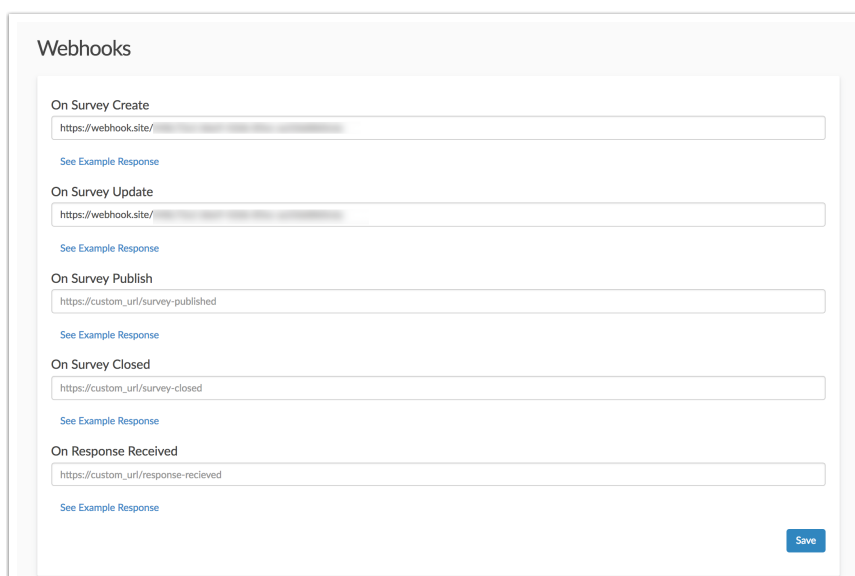
Available account notifications

- Survey-create notification
- Survey-update notification
- Survey-publish notification
- Survey-close notification
- Response-received notification

Setup

We recommend testing these notifications before you start sending data to your production endpoints.

1. To set up an Account Notification Webhook go to **Account > Integrations > Webhooks**.
2. Enter your endpoint URL in the field below the notification/s you wish to receive.
3. Click **Save**.

Review details of the data posted in each webhook type below.

Survey Notifications

Survey Create - The survey-create notification will post to the endpoint you specify when surveys are created. This includes newly created surveys, as well as, copied surveys created via the user interface or the API. See the below table for specifics about the data.

Survey Update - The survey-update notification will post to the endpoint you specify when surveys are updated. This includes the creation of or edits to questions, pages, actions, media elements, or translations made via the user interface or the API. See the below table for specifics about the data.

Survey Publish - The survey-publish notification will post to the endpoint you specify when share methods (link, email campaign, etc.) are created via the user interface or the API. The default survey share link is created when a user navigates to the share tab, which will fire the survey publish notification. See the below table for specifics about the data.

Survey Close - The survey-close notification will post to the endpoint you specify when a survey's status is changed to Closed. See the below table for specifics about the data.

Trigger Fields	Description	Type	Available Webhooks
type	The trigger type: `survey`, `page`, `question`, `translation`, `theme`	string	survey update
id	The ID of the element that was changed (survey ID, question ID, Page ID, etc.)	integer	survey update

User Fields	Description	Type	Available Webhooks
id	SurveyGizmo User ID	integer	survey create, survey update, survey publish, survey close
name	The username	string	survey create, survey update, survey publish, survey close
email	The user's login email address	string	survey create, survey update survey publish, survey close

Survey Fields Trigger Fields	Description	Type	Available Webhooks
survey_id	Survey ID	integer	survey create, survey update, survey publish, survey close
survey_title	Survey title	string	survey create, survey update, survey publish, survey close
survey_status	The survey status: `Launched` or `Closed`	string	survey create, survey update, survey publish, survey close
survey_folder	Array with survey folder id and name	array	survey create, survey update, survey publish, survey close
survey_theme	Array with theme ID and the theme name	array	survey create, survey update, survey publish, survey close
teams	Object with teams that have access to the survey	object	survey create, survey update, survey publish, survey close
Account Fields	**Description**	**Type**	**Available Webhooks**
id	SurveyGizmo Account ID	integer	survey create, survey update, survey publish, survey close
name	Account name as specified in the Contact Info under **Account > Summary > Account Overview**	string	survey create, survey update, survey publish, survey close
			survey create,

Trigger Fields	Description	Type	Available Webhooks
parent_id	Master Account ID	integer	survey update, survey publish, survey close

Survey Link Fields	Description	Type	Available Webhooks
id	Link ID	integer	survey publish
type	Link type: `link` or `email`	string	survey publish
name	Link name	string	survey publish
url	Link URL	string	survey publish

Example Response for Survey Notifications:

```
{
  "webhook_name": "On Survey Update",
  "trigger_context": {
    "type": "survey",
    "id": 1234567
  },
  "data": {
    "user": {
      "id": 12345,
      "name": "John Smith",
      "email": "john.smith@email.com"
    },
    "survey_id": 1234567,
    "survey_title": "My New Survey",
    "survey_status": "Launched",
    "survey_folder": [],
    "survey_theme": {
      "id": "68551",
      "theme": "My Custom Theme"
    },
    "teams": {
      "252692": {
        "name": "Team 1",
        "id": 252692
      }
    },
    "account": {
      "id": 33333,
      "name": "SurveyGizmo",
      "parent_id": null
    }
  }
}
```

Response Received Notification

The response received notification will post to the endpoint you specify when a completed or disqualified response is recorded via either normal survey taking or the API. See the below table for

specifics about the data.

Response Fields	Description	Type
is_test	Flag for test data	boolean
session_id	Unique response session ID	string
account_id	Account ID	integer
survey_id	Survey ID	integer
response_status	Response status: `Complete` or `Disqualified`	string
url_variables	URL Variables passed as part of the response	array
Survey Link Fields	**Description**	**Type**
id	Link ID	integer
type	Link type: `link` or `email`	string
name	Link name	string
url	URL	string
Contact Fields*	**Description**	**Type**
Email	Contact email address	string
First Name	Contact first name	string
Last Name	Contact last name	string
Organization	Contact organization	string
Team	Contact organization	string
Group	Contact group	string
Role	Contact role	string
Phone (Home)	Contact home phone	string
Phone (Fax)	Contact fax phone	string
Phone (Work)	Contact work phone	string
Address	Contact address	string
Suite/Apt	Contact suite/apt	string
City	Contact city	string

Response Fields	Description	Type
State/Region	Contact state/region	string
Country	Contact country	string
Postal Code	Contact postal code	string
Job Title	Contact job title	string
Website	Contact website	string
Region	Region	string
Invite Custom 1-10	Invite Custom Fields 1-10	string

*Present only if share method is an Email Campaign.

Example Response for Response Received Notification:

```
{
  "webhook_name": "On Response Received",
  "data": {
    "is_test": false,
    "session_id": "1591213853_5ed7ff1d3b37d9.13571736",
    "account_id": 33333,
    "survey_id": 1264838,
    "response_status": "Complete",
    "url_variables": {
      "sguid": "bG0PQDHDyDDhEGvl"
    },
    "survey_link": {
      "id": 8004308,
      "type": "link",
      "name": "Default Link",
      "url": "www.surveygizmo.com/s3/4835933/Survey"
    },
    "contact": {
      "Email": "test@email.com",
      "First Name": "John",
      "Last Name": "Smith",
      "Organization": "SurveyGizmo",
      "Division": "",
      "Department": "",
      "Team": "Sales",
      "Group": "",
      "Role": "Account Manager",
      "Phone (Home)": "",
      "Phone (Fax)": "",
      "Phone (Work)": "",
      "Address": "123 Main St",
      "Suite/Apt": "333",
      "City": "Boulder",
      "State/Region": "CO",
      "Country": "United States",
      "Postal Code": "12345",
      "Job Title": "",
      "Website": "",
      "Region": "CO",
      "Invite Custom 1": "1234567",
      "Invite Custom 2": "",
      "Invite Custom 3": "",
      "Invite Custom 4": "",
      "Invite Custom 5": "",
      "Invite Custom 6": "",
      "Invite Custom 7": "",
      "Invite Custom 8": "",
      "Invite Custom 9": "",
      "Invite Custom 10": ""
    }
  }
}
```

Get Real-time Response Data

If you wish to get real-time response data you can access survey data immediately without waiting for response processing with just two pieces of information: 1) the survey link and 2) the session ID.

The response received notification contains these two pieces of info that you can use in combination with the base link so you can construct a URL to get data from a response in real-time. The schematic

for constructing a real-time response URL is like so:

```
{survey_link}?snc={session_id}&__output=json
```

The survey link portion of the constructed link is the share link that was used to collect the response. For email campaigns this will be the base link as provided under the **Advanced Settings** section of the email campaign.

Link URL Format   https:// s-246769-i.sgizmo.com/s3/   Edit
                   Note: This is not a live link, do not copy and use elsewhere

So for the above example response, the following link will return the JSON for that response:

```
www.surveygizmo.com/s3/4835933/Survey?snc=1591213853_5ed7ff1d3b37d9.13571736&__output=json
```

The JSON will return the entire survey runtime object. In the below example JSON we highlight the portions of the JSON that allow you to access survey question data.

```
{
    "summary":{
        "is_test":false,
        "session_id":"1591213853_5ed7ff1d3b37d9.13571736",
        "account_id":160589,
        "survey_id":4835933,
        "response_status":"Complete",
        "url_variables":{
            "snc":"1591213853_5ed7ff1d3b37d9.13571736"
        },
        "survey_link":{
            "id":8004308,
            "type":"link",
            "name":"Default Link",
            "url":"www.surveygizmo.com\/s3\/4835933\/Checkbox"
        },
        "contact":[

        ],
        "is_last_page":true
    },
    "data":{
        "SurveyRuntime":{
            "BuildErrors":[

            ],
            "ResponseID":0,
            "ResponseSession":1,
            "SessionID":"1591213853_5ed7ff1d3b37d9.13571736",
            "PageHistory":[
                {
                    "SKU":1,
                    "TIME":1591213853,
                    "SPAWN_VALUE":null
                },
                {
                    "SKU":2,
                    "TIME":1591213856,
```

```
            "TIME":1591213883,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591213883,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591214484,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591217930,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591218345,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591218446,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591218789,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591218790,
            "SPAWN_VALUE":null
         }
      ],
      "PagePath":[
         {
            "SKU":1,
            "TIME":1591213853,
            "SPAWN_VALUE":null
         },
         {
            "SKU":2,
            "TIME":1591218790,
            "SPAWN_VALUE":null
         }
      ],
      "URLVariables":{
         "__slug":{
            "key":"__slug",
            "value":"www.surveygizmo.com\/s3\/4835933\/Survey",
            "type":"url"
         },
         "__output":{
            "key":"__output",
            "value":"json",
            "type":"url"
         },
         "snc":{
            "key":"snc",
            "value":"1591213853_5ed7ff1d3b37d9_13571736"
```

                  "value":"1591213853_5ed7f11d3b57d9.13571736",
                "type":"url"
            }
        },
        "LastPageShown":null,
        "IPAddress":"50.232.185.226",
        "UserAgent":"Mozilla\/5.0 (Macintosh; Intel Mac OS X 10_12_6) AppleWebKit\/537.36 (KH
ML, like Gecko) Chrome\/83.0.4103.61 Safari\/537.36",
        "AcceptLanguage":"en-US,en;q=0.9,he;q=0.8,mo;q=0.7,ro;q=0.6,rm;q=0.5",
        "Referer":"https:\/\/app.surveygizmo.com\/distribute\/share\/id\/4835933",
        "Language":"English",
        "PagesRandomized":false,
        "JumpTo":false,
        "Interaction":3,
        "InteractionDateTime":1591213856,
        "HasQuotas":null,
        "SurveyStatus":"Complete",
        "SessionStarted":1591213853,
        "queue_name":"Responses_prod_4_133",
        "Country":"United States",
        "Region":"CO",
        "City":"Lafayette",
        "Postal":"80026",
        "Latitude":39.997001647949,
        "Longitude":-105.09739685059,
        "DMACode":751,
        "LinkSource":"8004308",
        "SurveyLink":{
            "iLinkID":"8004308",
            "iSurveyID":"4835933",
            "iInviteID":null,
            "eType":"link",
            "eStatus":"Active",
            "sSubType":"standard",
            "sName":"Default Link",
            "sIdentifier":"www.surveygizmo.com\/s3\/4835933\/Checkbox",
            "iDomainID":null,
            "sDomain":"www.surveygizmo.com",
            "sAutoTokenVariables":null,
            "eSSL":"True",
            "sEmbedDomain":null,
            "iQuotaLimit":null,
            "dScheduledClose":null,
            "dScheduledOpen":null,
            "eVoteProtect":"None",
            "sVoteProtectURLVariable":null,
            "iUserIDCreated":"183830",
            "iUserIDLastModified":"183830",
            "dCreated":"2019-02-13 10:45:16",
            "dModified":"2019-02-13 10:45:16",
            "iCustomerID":"160589",
            "iUserIDDeleted":null,
            "dDeleted":null,
            "eMode":null,
            "sMessage":null,
            "sVar1":null,
            "sVar2":"SendHook",
            "sVar3":null,
            "sCloseMessage":null,
            "sLanguage":"Auto",
            "sTPSData":null,
            "iReceivableID":null,
            "iMobileThemeID":null,
            "bThemeObject":"TY0xCoQwFETv8k9glF1hUm5vmu2XqF82YBLJTyPi3Y1oYTlvHjMG6gX6\/tmzfGL
U0lDaba1WsUaLWlCG+a49q2QKwnGsV1INSPI68+\/+WWf6iWg+jTfS4YkNgYMl9q+j8LsjDfQwbG+\/Wg="

```
VOnBqboAUtyXmbVtlGqr49s2QXgxCaUlINSPI68y\/exHf61Yeyy1FfuHyNyU4I9ajULn1DfOzdzGe\/Hw==",
            "bMetaObject":"S7QysKquBQA=",
            "iThemeID":null,
            "eServerLocation":"us-denver",
            "ePopupCookie":"Off",
            "eDefaultLink":"False"
        },
        "ismobile":false,
        "TestData":false,
        "TestDataCollab":false,
        "queue_number":1,
        "CurrentRuntimeOutput":"json",
        "HTMLEmbed":false,
        "dNumbering":true,
        "PageDirection":-3,
        "FingerprintValid":"sgfbbfcb3a9eccc624df1cd2754ee80e3d",
        "MinProgressStep":100,
        "TopProgress":0,
        "TotalTime":3,
        "DataQuality":{
            "CheckBox":{
                "one":{
                    "2":2
                }
            }
        },
        "Fingerprint":"2395398737",
        "AvgQuestSeconds":0.007,
        "HighContrast":"false",
        "interaction_format_version":3.1
    },
    "SectionSubmitted":null,
    "Sections":{
        "1":{
            "SectionRuntime":{
                "Hidden":false,
                "Disabled":false,
                "Shown":true,
                "Shuffled":false,
                "Submitted":true,
                "XNumberSelected":false,
                "HasErrors":false,
                "NextPage":2,
                "LastPage":1,
                "StartQuestion":1,
                "EndQuestion":1,
                "PipePrototype":false,
                "Piped":false,
                "PipedValue":null,
                "PipeHash":null,
                "Errors":[

                ],
                "Number":null,
                "start_time":1591213853,
                "hide_all_after_until_value":false,
                "PostRenderContent":false,
                "PreRenderContent":false,
                "progress_updated":true,
                "end_time":1591213856,
                "page_time":3,
                "AvgQuestSeconds":0.007
            },
            "Questions":{
```

```
            "2":{
                "QuestionRuntime":{
                    "Hidden":false,
                    "Required":false,
                    "Soft-Required":false,
                    "Disabled":false,
                    "Rendered":true,
                    "HasErrors":false,
                    "IsSingleSelect":false,
                    "Errors":[

                    ],
                    "Answered":false,
                    "Number":"1.",
                    "Populated":false,
                    "PipePrototype":false,
                    "Piped":false,
                    "PipedValue":null,
                    "PipedHash":null,
                    "Data":{
                        "atoms":{
                            "10001":{
                                "data_type":1,
                                "raw_value":"Downtown",
                                "file_name":null,
                                "file_contents_compressed":null,
                                "file_type":null,
                                "file_length":null,
                                "datetime":null,
                                "hour24":null,
                                "minute":null,
                                "second":null,
                                "timezone":"CST",
                                "range_start":null,
                                "range_end":null,
                                "listsource_uri":null,
                                "listsource_id":null,
                                "sentiment":null,
                                "listsource_ident":null
                            }
                        }
                    },
                    "ElapsedSeconds":0.007,
                    "LimitQuotaError":null,
                    "Type":"CHECKBOX"
                }
            }
        }
    },
    "2":{
        "SectionRuntime":{
            "Hidden":false,
            "Disabled":false,
            "Shown":true,
            "Shuffled":false,
            "Submitted":false,
            "XNumberSelected":false,
            "HasErrors":false,
            "NextPage":0,
            "LastPage":1,
            "StartQuestion":2,
            "EndQuestion":1,
            "PipePrototype":false,
            "Piped":false,
```

```
                    "PipedValue":null,
                    "PipeHash":null,
                    "Errors":[

                    ],
                    "Number":null,
                    "hide_all_after_until_value":false,
                    "start_time":1591213856,
                    "PostRenderContent":false,
                    "PreRenderContent":false,
                    "progress_updated":true
                  },
                  "Questions":{
                    "1":{
                      "QuestionRuntime":{
                        "Hidden":false,
                        "Required":false,
                        "Soft-Required":false,
                        "Disabled":false,
                        "Rendered":true,
                        "HasErrors":false,
                        "IsSingleSelect":false,
                        "Errors":[

                        ],
                        "Answered":false,
                        "Number":"",
                        "Populated":false,
                        "PipePrototype":false,
                        "Piped":false,
                        "PipedValue":null,
                        "PipedHash":null,
                        "Data":{
                          "atoms":[

                          ]
                        },
                        "Type":"INSTRUCTIONS"
                      }
                    }
                  }
                }
              }
            }
          }
```
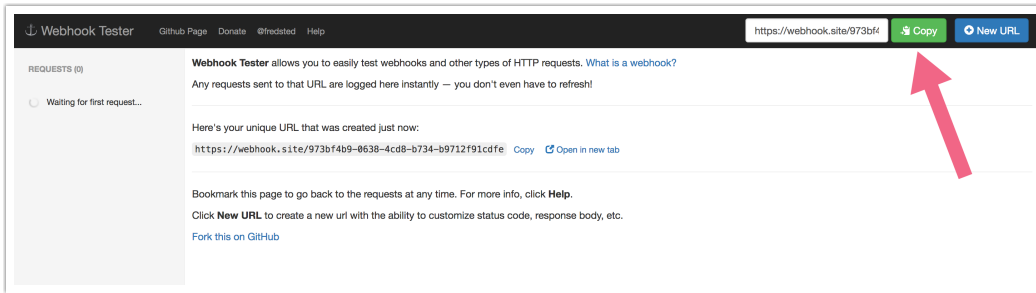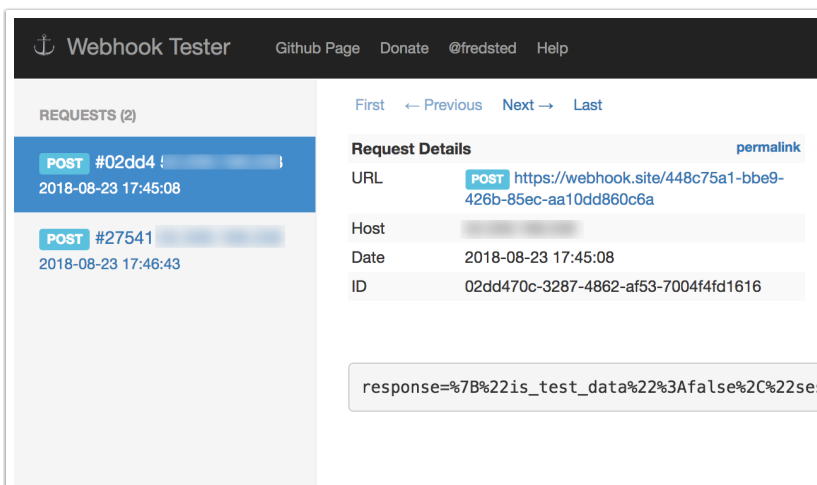
Test Account Webhooks

There are a number of third-party tools available for testing your Webhook, including Webhook Tester and Mockbin. We have highlighted testing with Webhook Tester below.

1. Go to the following link: https://webhook.site/#/.
2. A unique URL will be automatically generated for you when you enter the site. Click the copy button to the upper right to copy your unique testing URL to your clipboard. *Note: Do not use the link in the image, it is a unique link that was created for our particular instance, it will not work for you.*

3. Go to your Account Webhooks via **Account > Integrations > Webhooks**.
4. Paste your unique Webhook Tester URL into the Account Webhook that you want to test.
5. Save your Webhooks and perform one of the associated tasks to trigger the Webhook. For example, to test the **On Survey Create** webhook, create a survey.
6. To view the results that get posted, return to Webhook Tester. Each time a new response is submitted it will appear in your request list in the left menu. Click it to review the request details.

FAQ

How will Account Webhooks work in Master/Sub Accounts?

Account Webhooks are only available at an account level. So, Account Webhooks set up in a master account will only send notifications regarding activity in the master account. If you wish to receive notifications of activity in your sub accounts you can log in to your sub accounts and set up AccountWebhooks there to do so.