SurveyCTO

Help - Exporting and publishing data - Publishing data to the cloud

Below, please find topics relating to how you can publish your incoming form data straight to the cloud (for visualization, analysis, and more).

Introduction to cloud publishing

You might want to publish a subset of your incoming data to the cloud so that you can easily visualize that data, share it with others, or even trigger outside processes (like sending SMS messages). SurveyCTO allows you to easily configure this kind of publishing, to Google Sheets, Google Fusion Tables, Zapier, or a wide variety of other systems via our generic webhooks interface.

See the following help topics for details on publishing your data to particular outside systems.

Limitations

When publishing SurveyCTO data to the cloud, there are two key limitations to keep in mind:

1. **Your SurveyCTO subscription may not support cloud publishing**. Some lower-priced SurveyCTO subscriptions don't support cloud publishing. With your current subscription, cloud publishing is: *Supported*. For more about your SurveyCTO subscription, see *Your subscription: details and management*.
2. **You can't publish encrypted data**. If you have encrypted your form data with your own encryption keys, you can publish only those form fields that were explicitly marked as *publishable* (i.e., fields for which you indicated "yes" in the *publishable* column of your *survey* worksheet). This is because SurveyCTO can't read (or share) encrypted data.

   You should take care not to mark sensitive, highly-confidential fields as *publishable*, as *publishable* fields will not be as strongly protected as other fields in encrypted forms. (They will still be encrypted in transit, but they will be readable by SurveyCTO and by any outside service to which you publish them.)

Publishing to Google Sheets

Google Sheets (a.k.a. "Google Spreadsheets") is a powerful, flexible, free online spreadsheet program, available as part of Google Drive and the Google Docs suite of cloud-based office software. There are innumerable ways that you might manage, monitor, share, process, or analyze your data in Google Sheets, so we try to make it easy to publish your data directly from SurveyCTO into Sheets.

You can get started by going to your server console's Export tab, scrolling down to the *Advanced: publishing form and dataset data to the cloud* section, and clicking the ON/OFF toggle to *ON* if you haven't already enabled cloud publishing.

To configure any one of your forms to publish to Google Sheets, click on the *Configure* option for that form, and then click *Add connection* in the *Google Speadsheets* row of the panel that appears.

You will then be asked to give the SurveyCTO server permission to access your Google Sheets and Google Drive accounts. This permission is necessary so that SurveyCTO can list your existing Sheets and publish to the one you select, and it only takes one or two clicks (plus logging in to your Google account, if you haven't already). SurveyCTO will only read from or write to Google Sheets as you instruct in configuring publishing options.

Next, you will need to decide exactly where you want to publish your form data: to an existing Sheet (in which case you just select it from the list) or to a new Sheet (in which case you use the + button, enter the name of the new Sheet to create, and then click the checkmark button to create that Sheet).

After that, you will need to decide exactly which form fields to publish, to exactly which columns on the Google side (we call this process "mapping"). There is an *Add all* button to simply add all form fields at once (or, in the case of encrypted forms, all form fields that have been marked as *publishable*), plus an *Add* button to add fields one-by-one. For each added field, SurveyCTO will default to publishing to a column name on the Google side that matches the field name on the SurveyCTO side – but you can override this default in cases where you want the column name on the Google side to be different.

Note that fields that are inside repeat groups will be listed with an * at the end of their names, and they must be mapped to destination columns that also end in *. When repeated data is published, the * will be automatically replaced by *_1* for the first instance, *_2* for the second, and so on.

Other options

Finally, you have a few other options available:

1. You can indicate one of your form fields to use in uniquely identifying records on the Google side. If you do, then new form submissions will either update an existing row of the Google Sheet – if a row with a matching value in the corresponding column already exists – or insert a new row. If you don't specify a unique identifier, then new submissions will simply publish as new rows. You would likely want to use a unique identifier if you were using a Google Sheet to merge data from multiple sources. (This unique identifier has to be one of the form fields that you listed in the field mapping. I.e., you also have to publish whatever field is being used to uniquely identify rows.)
2. You can indicate one of your form fields to use as a kind of filter. If you do, only submissions for which the specified field contains the value *1* will be published. For example, if you wanted to only publish data for children, you might select an "ischild" field as the filter; in your form, that could be a *calculate* field with a calculation expression like "if(${age} < 18, 1, 0)".
3. You can check *Publish existing data* if you want to publish existing form submissions. If you don't check this option, then only new submissions that come in (after you configure publishing) will publish to Google.

Publishing

As submissions come in to the server, they will be automatically published to your Google Sheet according to the field mappings and other options that you have configured – but there will be a brief delay of up to ten minutes.

By default, SurveyCTO publishes into the first (left-most) worksheet in your target document. However, you might have some kind of dashboard set up as your first worksheet, or you might otherwise prefer the raw data to publish to another worksheet rather than the first one. In that case, just name the worksheet into which raw data should publish "data": if SurveyCTO sees a worksheet named *data*, it will publish to that worksheet even if it's not the first one.

See the Google Sheets website for details on configuring and using Google Sheets itself.

Publishing to Google Fusion Tables

Google describes Fusion Tables as "an experimental data visualization web application to gather, visualize, and share data tables." If you're collecting data using SurveyCTO, it's a free and simple method of monitoring and visualizing your incoming data.

You can get started by going to your server console's Export tab, scrolling down to the *Advanced: publishing form and dataset data to the cloud* section, and clicking the ON/OFF toggle to *ON* if you haven't already enabled cloud publishing.

To configure any one of your forms to publish to Google Fusion Tables, click on the *Configure* option for that form, and then click *Add connection* in the *Google Fusion Tables* row of the panel that appears.

You will then be asked to give the SurveyCTO server permission to access your Google Fusion Tables and Google Drive accounts. This permission is necessary so that SurveyCTO can list your existing Fusion Tables and publish to the one you select, and it only takes one or two clicks (plus logging in to your Google account, if you haven't already). SurveyCTO will only read from or write to your Fusion Tables as you instruct in configuring publishing options.

Next, you will need to decide exactly where you want to publish your form data: to an existing Fusion Table (in which case you just select it from the list) or to a new Fusion Table (in which case you use the + button, enter the name of the new table to create, and then click the checkmark button to create it).

After that, you will need to decide exactly which form fields to publish, to exactly which columns on the Google side (we call this process "mapping"). There is an *Add all* button to simply add all form fields at once (or, in the case of encrypted forms, all form fields that have been marked as *publishable*), plus an *Add* button to add fields one-by-one. For each added field, SurveyCTO will default to publishing to a column name on the Google side that matches the field name on the SurveyCTO side – but you can override this default in cases where you want the column name on the Google side to be different.

Note that fields that are inside repeat groups will be listed with an * at the end of their names, and they must be mapped to destination columns that also end in *. When repeated data is published, the * will be automatically replaced by *_1* for the first instance, *_2* for the second, and so on.

Other options

Finally, you have a few other options available:

1. You can indicate one of your form fields to use in uniquely identifying records on the Google side. If you do, then new form submissions will either update an existing row of the Fusion Table – if a row with a matching value in the corresponding column already exists – or insert a new row. If you don't specify a unique identifier, then new submissions will simply publish as new rows. You would likely want to use a unique identifier if you were using a Fusion Table to merge data from multiple sources.

(This unique identifier has to be one of the form fields that you listed in the field mapping. I.e., you also have to publish whatever field is being used to uniquely identify rows.)

2. You can indicate one of your form fields to use as a kind of filter. If you do, only submissions for which the specified field contains the value *1* will be published. For example, if you wanted to only publish data for children, you might select an "ischild" field as the filter; in your form, that could be a *calculate* field with a calculation expression like "if(${age} < 18, 1, 0)".
3. You can check *Publish existing data* if you want to publish existing form submissions. If you don't check this option, then only new submissions that come in (after you configure publishing) will publish to Google.

Publishing

As submissions come in to the server, they will be automatically published to your Fusion Table according to the field mappings and other options that you have configured – but there will be a brief delay of up to ten minutes.

See the Google Fusion Tables website for details on configuring and using Google Fusion Tables. If you are planning to publish dates, times, or locations to a Google Fusion Table, see below for how to reformat those field types for easy use within Fusion Tables.

Formatting data for Google Fusion Tables

By default, date, time, and location information published directly from SurveyCTO into a Google Fusion Table is not easy to work with. This is because Google prefers that dates, times, and locations be in a different format. For now, the work-around is to add extra *calculate* fields to your survey forms that store dates, times, and locations in a Google-friendly format (as necessary for any dates, times, and locations you want to work with easily in Google Fusion Table).

If you would like to map a *geopoint* field to Google Fusion Tables, you will need to publish a Google-friendly version of the location that only contains the latitude and longitude (no altitude or accuracy). You can do that easily by including a *calculate* field in your form that has a *calculation* column like this: "short-geopoint(${gpsfieldname})" (without the surrounding quotes and with your *geopoint* field name in the place of "gpsfieldname"). If you publish that shortened location field to Google Fusion Tables, then you will be able to set that column to be a location-type column. You will then be able to use map-style visualizations for that table, with data that streams in from your SurveyCTO form.

Please note, however, that Fusion Tables map visualizations (which rely on Google Maps) may not be enabled for your version of Fusion Tables if you are a Google Apps customer (see below to learn more about Fusion Tables within Google Apps domains).

Finally, if you want *date* and *datetime* fields to be easily recognizable to Google Fusion Tables, you will need to format them specially as well. For a *date* field, simply publish a calculate field with "format-date-time(${fieldname}, '%Y-%b-%e')" as its *calculation* column (without the quotes and with your field name in the place of "fieldname"). For a *datetime* field, publish a *calculate* field with "format-date-time(${fieldname}, '%Y-%b-%e %H:%M:%S')" as its *calculation* column. Google should easily recognize those date and time formats.

Using Google Fusion Tables in Google Apps accounts

If your Google account is a Google Apps account rather than a regular Gmail account (i.e., if your Google-based email address ends in something other than @gmail.com), there are a few things that you need to know in order to use Google Fusion Tables.

First, if Google says that your account does not have access to Google Fusion Tables, your domain administrator needs to enable Google Fusion Tables for your Google Apps domain. To guide him or her, you can share this list of steps:

1. Log in to admin.google.com to administer the Google Apps domain.
2. At the very bottom of the Admin Console, click "More controls".
3. Then click "More Google apps".
4. Then click the funnel icon in the top-right and turn off the "Show top featured services" filter (so that all available services are shown).
5. Then find and click on the hyperlink for "Fusion Tables (experimental)".
6. Finally, click the red "OFF" button on the left to switch it to "ON".

Second, Fusion Tables map visualizations (which rely on Google Maps) may not be enabled for Google Apps customers. See this Google help topic for details.

(Apologies if this help topic is out-of-date. Any changes on Google's side will render it so.)

Publishing to Zapier

Zapier is a web service designed to connect other web services to each other. It acts as a kind of bridge, and it can take data streaming in from your SurveyCTO forms and trigger a wide variety of possible actions: you could send SMS's or emails, route the incoming data into other database or visualization systems, and much more. A single submission coming in from a SurveyCTO form can even trigger a series of different actions via Zapier.

Step one: Setting up the SurveyCTO side

You can get started by going to your server console's Export tab, scrolling down to the *Advanced: publishing form and dataset data to the cloud* section, and clicking the ON/OFF toggle to *ON* if you haven't already enabled cloud publishing.

To configure any one of your forms to publish to Zapier, click on the *Configure* option for that form, and then click *Add connection* in the *Zapier* section of the panel that appears.

You'll first need to choose a good name for your Zapier connection, based on the form and the subset of fields that you intend to publish to Zapier. After that, you'll need to select exactly which form fields to publish (but note that, for encrypted forms, only form fields that have been explicitly marked as *publishable* will be listed). There's a *Select all* button if you simply want to publish all fields.

Then, you have a few other options available:

1. You can choose whether or not to include a hyperlink to the full submission in SurveyCTO, in the data that publishes to Zapier. (If you do include the hyperlink, and you happen to also be publishing a form field named "submission_url", choose a different name for the hyperlink.)
2. You can check *Publish existing data* if you want to publish existing form submissions. If you don't check this option, then only new submissions that come in (after you configure publishing) will publish to Zapier.

Step two: Setting up the Zapier side

Once you save the new connection details on the SurveyCTO side, you'll be given an API key to use in configuring the connection on the Zapier side. Copy that to your clipboard.

To start creating a SurveyCTO-powered "zap" on the Zapier side, click here for instructions on linking to SurveyCTO... (That page will give you updated information on how to get started, depending on SurveyCTO's public listing status.)

When the time comes to configure your SurveyCTO account on the Zapier side, you'll be prompted for an API key. Just paste in the one you copied from your SurveyCTO console. (See this page for more about starting a SurveyCTO zap in Zapier, and see the Zapier website for more about how to configure and use Zapier more generally.)

Tip: A potential point of confusion is that SurveyCTO gives you a separate API key – and you have to link a separate "account" on the Zapier side – for every form that you publish to Zapier. Within a form, you can then configure multiple "connections" to publish different sets of fields (but in most cases you'll just need one connection per form). On the Zapier side, there is no way to name a linked "account" when it's created, but you can go to your dashboard to rename account links after you've created them. We strongly advise that, if you're going to publish multiple forms to Zapier, you rename your accounts to help keep track of which ones are associated with which forms.

Publishing

As submissions come in to the server, your selected fields will be automatically published to Zapier – but there will be a brief delay of up to ten minutes.

Publishing to other systems via webhooks

Webhooks allow web services to trigger actions, push data, or otherwise connect to other web services. Initially the exclusive domain of programmers who sought to use certain technologies and conventions to connect systems, webhooks are increasingly useful to non-programmers as well. You can use webhooks to publish incoming SurveyCTO form data to a wide variety of outside systems, but typically you will need some technical expertise and/or instructions from the receiving system in order to successfully configure everything.

You can get started by going to your server console's Export tab, scrolling down to the *Advanced: publishing form and dataset data to the cloud* section, and clicking the ON/OFF toggle to *ON* if you haven't already enabled cloud publishing.

To configure any one of your forms to publish via webhooks, click on the *Configure* option for that form, and then click *Add Webhook* in the panel that appears.

You'll first need to choose a good name for your webhook connection, based on the system to which you're publishing and what you intend to publish. After that, you'll need to select exactly which form fields to publish (but note that, for encrypted forms, only form fields that have been explicitly marked as *publishable* will be listed). There's a *Select all* button if you simply want to publish all fields.

Other options

Finally, you have a few other options available:

1. You can choose whether or not to include a hyperlink to the full submission in SurveyCTO, in the data that publishes to the webhook. (If you do include the hyperlink, and you happen to also be publishing a form field named "submission_url", choose a different name for the hyperlink.)

2. You can include one extra field, if you wish, as a text summary of the submission (e.g., as its title or alert text, depending on the system to which you're publishing). If you choose to include an extra field, you can choose its name and its contents; in its contents, you can use *${fieldname}* references to include data from the submission being published, just like in a form label. For example, you might include a summary like "Submission received from ${enumerator_name}, for household headed by ${hh_head}". (Note, however, that you can only reference "publishable" fields in encrypted forms.)
3. You can choose to embed the contents of binary fields (files attached to submissions) in the data published to the webhook. If you don't choose to do so, file fields will simply publish as hyperlinks. (Note that you can't embed binary fields for encrypted forms.)
4. You can check *Publish existing data* if you want to publish existing form submissions. If you don't check this option, then only new submissions that come in (after you configure publishing) will publish to the webhook.

Publishing

As submissions come in to the server, your selected fields will be automatically published to your chosen webhook – but there will be a brief delay of up to ten minutes.