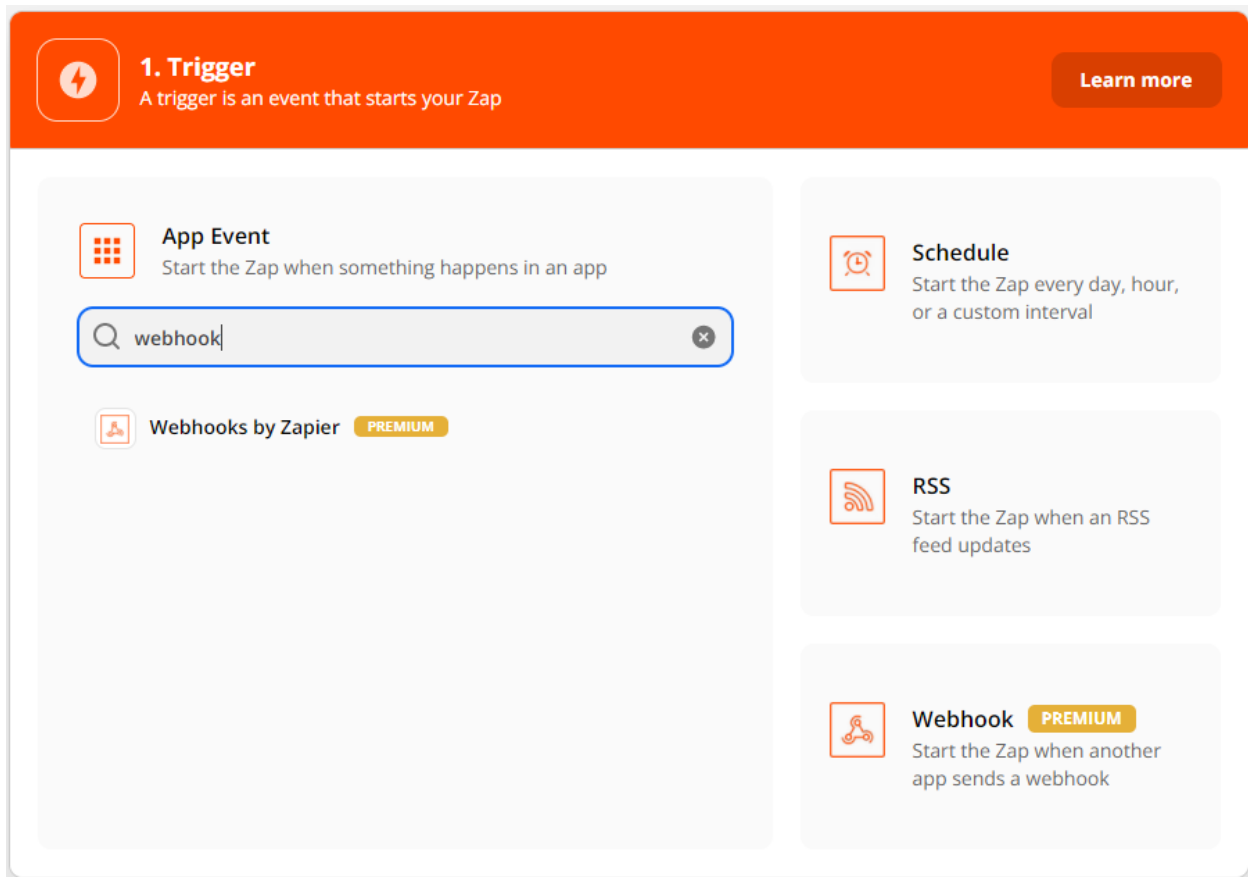


Zapier Integration - Webhook

Note: As with many SaaS platforms, including our own, improvements are constantly made to the user interface (UI). Over time, the UI may look somewhat different from the screenshots we show here but fundamentally the process is the same.

A. Creating a CodeREADr Trigger

1. Go to Zapier and select “Create a Zap”. For the Trigger, select “**Webhooks by Zapier**”.



2. For the Trigger Event choose “**Catch Hook**”.

Trigger **1. Webhooks**

Choose app & event

Webhooks by Zapier PREMIUM Change

Trigger Event (required)

Choose an event

Retrieve Poll
Poll a URL and look for new entries.


Catch Hook INSTANT
Wait for a new POST, PUT, or GET to a Zapier URL.

Catch Raw Hook INSTANT
Wait for a new POST, PUT, or GET to a Zapier URL. Gives the request body unparsed (max 2 MB) and also includes headers.

2. Action
An action is an event a Zap performs after it starts

+

3. Copy the **“Custom Webhook URL”**.

 Trigger
1. Catch Hook

▼ Choose app & event ✓

Set up trigger

Custom Webhook URL

Copy

We've generated a custom webhook URL for you to send requests to. You can append `silent/` if your application prefers getting an empty response. [Learn more about using webhooks.](#)

Silent Mode
 Enable to respond with an empty body.

Pick off a Child Key

By default, Zapier gives you the entire payload of the webhook. If this is specified, Zapier will only grab the child key from the object(s) sent to Zapier. For example, given `{"contact": {"name": "Mike"}}`, add "contact" here to only return `{"name": "Mike"}`. Traverse more deeply nested children by using dot-separated syntax. Line item payloads will run your Zap for each instance of the child key.

Refresh fields

Continue

4. Paste the URL on the Advanced step Postback field when creating/editing your CodeREADr service. It's important to also check the option "**Receiver Only**". You should also create an [Export Template](#) so you can choose what data to post, column headers, column order, and format the data as needed (see Google Sheet example below).

For example, you can format timestamps, choose GPS locations to include the address and/or a Google Maps URL, format clickable photo URLs, and optionally use regex or formulas for customizing the posted data.

On-device:

- Duplicate scans on-device are INVALID.
(This only applies when scanning to the device and only checks against the scans currently on the device.)
- Manual Sync (auto check for new database after manual scan upload).
- Auto Sync (auto upload saved scans and download new databases).
Please visit [HERE](#) before using this feature.
 - (optional) Sync scans directly to my server.
- Email me when on-device scans are uploaded.
- Postback each on-device scan uploaded to a specified URL.
 - Receiver Only Full Postback

Enter your Postback URL:

For advanced users - requires IT integration. ([Learn more here.](#))

Choose a [template](#) for your POST variables:



Template name: **Zapier Collected Data Offline**


created on: Jan 24, 2022 1:42 PM

 Edit

The variables that will be posted to your Postback URL.

Barcode Scanned
Validity
Timestamp
Photo
Comments
GPS Location
Task
Scanning Device



5. You should then test your trigger. It's important to **first make a scan with the CodeREADr mobile app** so the test process has data to conduct the test.

 Trigger
1. Catch Hook ? ...

▼ Choose app & event ✓

▼ Set up trigger ✓

Test trigger

 → 

Test your trigger


Send a request to this webhook URL:
[https://hooks.zapier.com/hooks/catch/\[redacted\]a8he](https://hooks.zapier.com/hooks/catch/[redacted]a8he) then test your trigger.


[Test trigger](#)

6. If the test is successful, you will see the fields you created in the Export Template along with sample scan data.


Set up trigger ✓

Test trigger

 **We found a request!**
This request was sent to <https://hooks.zapier.com/hooks/catch/...a8he>. [Learn more about test data.](#)

 Request A ⌵

Search...

Barcode Scanned: a_0000000000 

Validity: Valid - Barcode in Database

Timestamp: 2022-01-24 13:53:34

Photo: <https://dl.dropboxusercontent.com/s/jfvyyxv6z1lg96w/s...t442930804>

Comments: Match

GPS Location: geo://28.519041, ... 4.841970,78,iOS,address{16186 Johns ... , FL 34787 United States}

Task: xxxxx5

Scanning Device: 363632

querystring:

◀ [Progress bar] ▶

[Continue](#)

7. Next you will be prompted to create the Action. In this case, we chose to connect with the **Google Sheets** app. It's recommended to test with an app like Google Sheets before connecting other applications. It's also recommended to select the "**Create Spreadsheet Row**" action for this test.

The screenshot shows the Zapier configuration screen for the action "2. Create Spreadsheet Row in Google Sheets". At the top, there is a header with a Google Sheets icon, the word "Action", and the title "2. Create Spreadsheet Row in Google Sheets". Below this, the section "Choose app & event" contains a dropdown menu with "Google Sheets" selected, indicated by a red arrow. To the right of this dropdown is a "Change" link. Below the app selection is the "Action Event" dropdown, which has "Create Spreadsheet Row" selected, also indicated by a red arrow. To the right of this dropdown is a "(required)" label. Below the event selection, there is a note: "This is performed when the Zap runs." At the bottom of the configuration area is a large blue "Continue" button.

8. Now you will need to set up the action.

First, you should set up your Google Drive account as the recipient. **You need to create a spreadsheet with the matching columns you created in your Export Template.** Once you've connected your Google Drive account, you will need to also select the actual spreadsheet and worksheet to post scanned data (see Google Sheet example below).

Zapier will detect the columns of the spreadsheet. You need to match the test results with each column. When you click the fields (e.g. "Barcode Scanned" in this example), you need to select the test fields to lock them into the appropriate field.

