SMRT Systems Help Center

SMRT Systems Help Center > Learning Center > Marketing

Search

Articles in this section

Setting Up Birdeye

Johan Baversjo
3 years ago · Updated

Follow

Introduction

The Birdeye integration allows you to send "Check In" events to Birdeye, when an order is returned to the customer in SMRT. Birdeye will then send review requests automatically, either over Email or SMS depending on the customer's preference in SMRT.

Prerequisites

You need a Birdeye account, with an active campaign automation. Please contact your Birdeye Representative for how to set this up.

You will also need access to a Birdeye API key. Please refer to this guide for how to g an API Key: https://support.birdeye.com/s/article/Where-can-I-find-the-API-key-for-m

Help

Lastly, you will also need a Segment account, that you've connected SMRT as a source. Please follow this guide if you haven't done so: https://support.smrtsystems.com/hc/en-us/articles/115010091648-Setting-Up-Segment

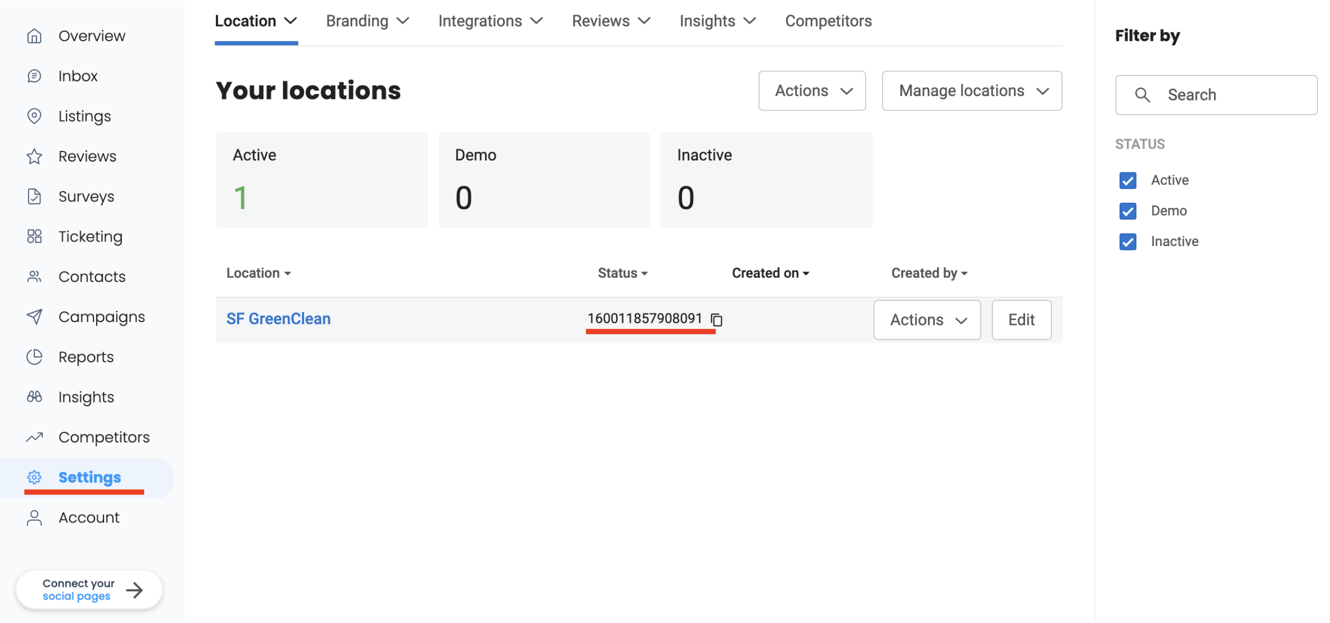
Store Location Setup

For each store you with to capture reviews for, you'll need to copy its Birdeye BID into the SMRT External Store Identifier field. Please follow the steps below for each of your stores.

1. Open up your Birdeye account.

2. Go to settings

3. highlight your location, and copy the long number (BID number) highlighted in red for the store location:

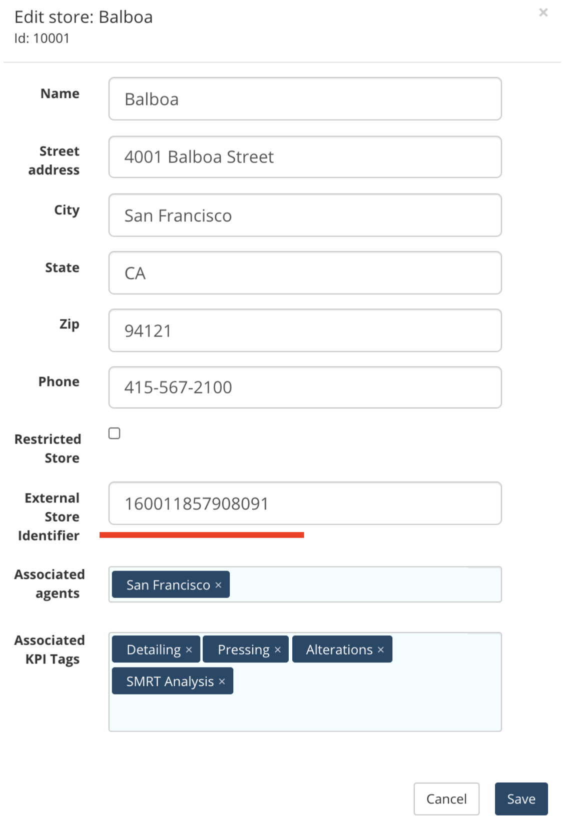


Location	Status	Created on	Created by	
SF GreenClean	160011857908091			Actions Edit

4. Open up the stations configuration in SMRT under Settings => Stations

5. Press the cogwheel next to the name of the store you're configuring

6. The below popup will appear. Insert the BID number from step 3 into the external store identifier as seen below, and press save:



Edit store: Balboa
Id: 10001

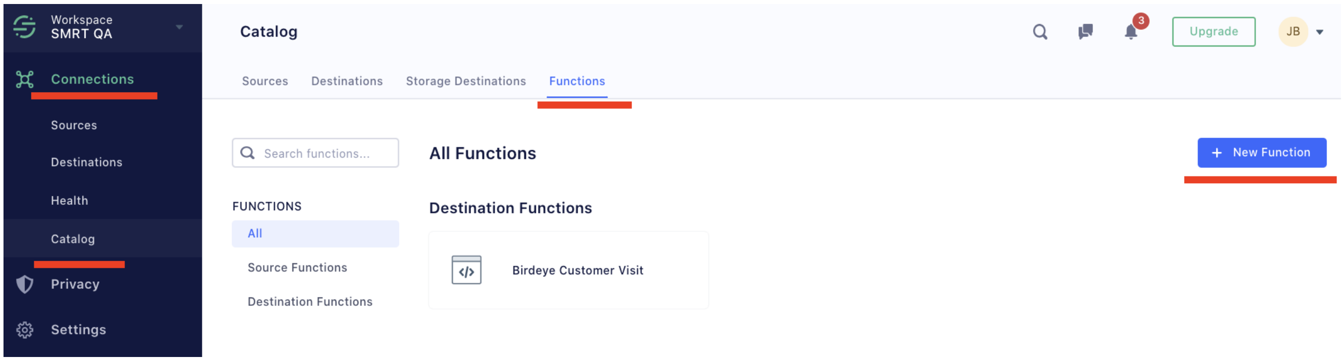
Name	Balboa
Street address	4001 Balboa Street
City	San Francisco
State	CA
Zip	94121
Phone	415-567-2100
Restricted Store	□
External Store Identifier	160011857908091
Associated agents	San Francisco ×
Associated KPI Tags	Detailing × Pressing × Alterations × SMRT Analysis ×

Repeat for each store. You have now associated the stores in SMRT with locations in Birdeye.

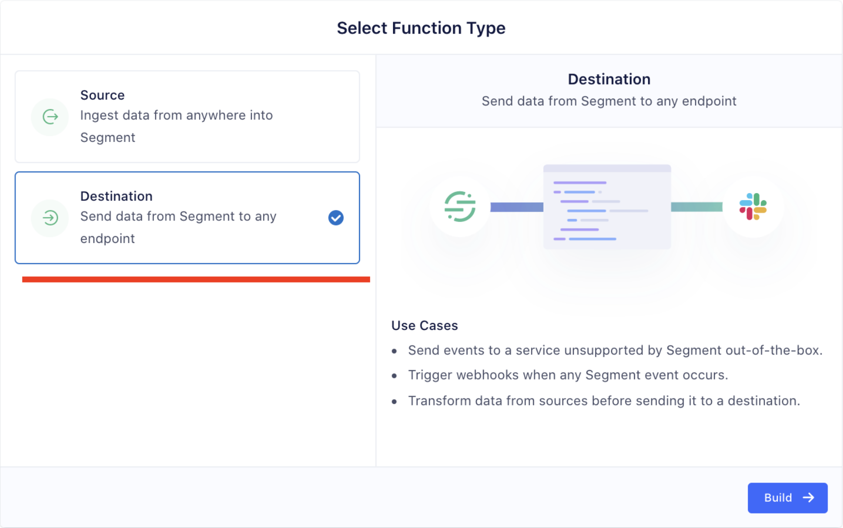
Segment Birdeye Integration

Segment doesn't come with a Birdeye integration out of the box. As such, SMRT has created one. The code for it is listed at the bottom of this page. Follow the steps below, to setup this integration for your segment account.

1. Sign in to your segment workspace at https://app.segment.com/ , and go to connections => catalog => functions => new function as shown in below screenshot:



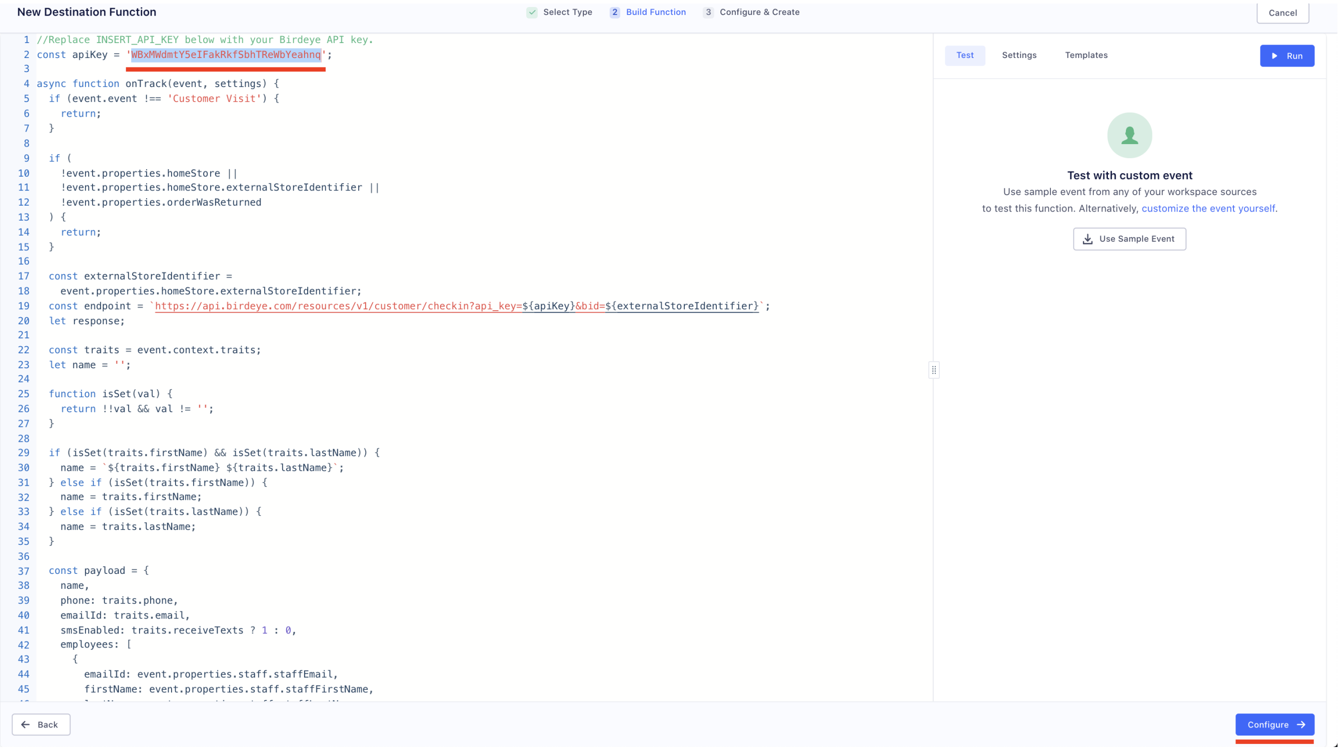
2. Select function type "Destination" and press "Build" in the bottom right:


Select Function Type
Source
Ingest data from anywhere into Segment
Destination
Send data from Segment to any endpoint
Destination
Send data from Segment to any endpoint
Build

Use Cases

- Send events to a service unsupported by Segment out-of-the-box.
- Trigger webhooks when any Segment event occurs.
- Transform data from sources before sending it to a destination.

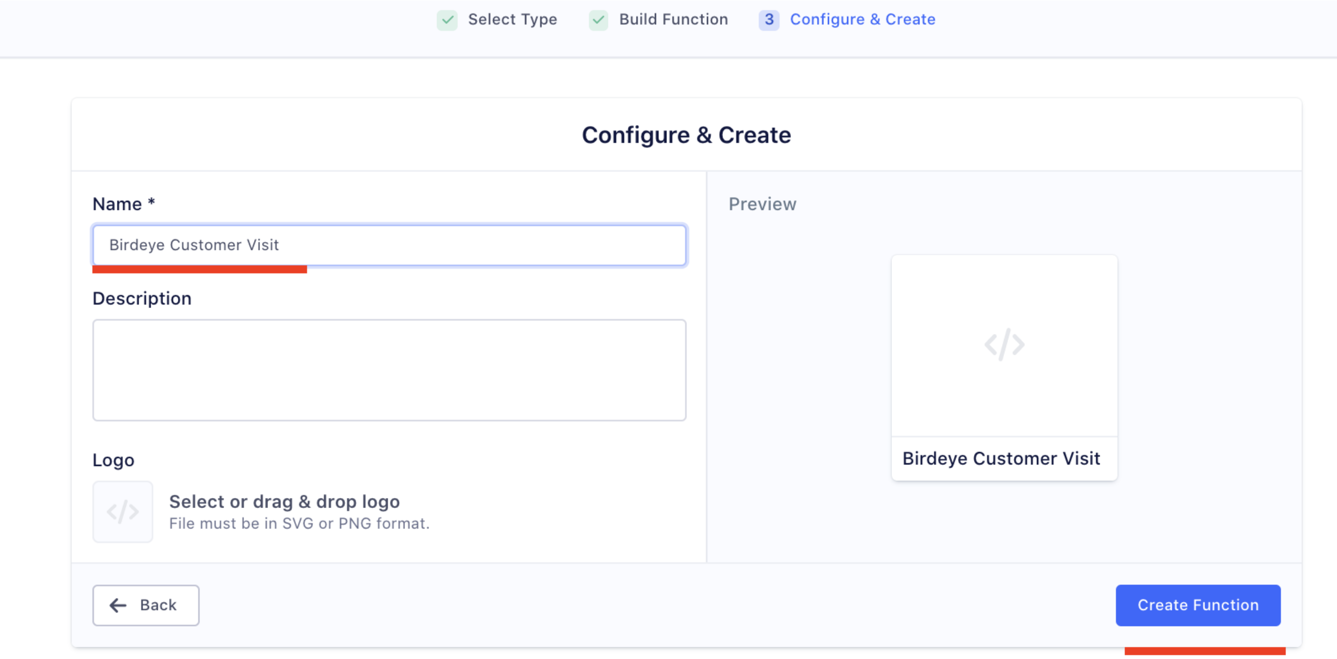
3. Paste the code at the bottom of this guide under **Integration Code**, into the source code editor on the left side of the screen as shown in the screenshot below.



```
//Replace INSERT_API_KEY below with your Birdeye API key.
const apiKey = 'WBxMWdmtY5eIFakRkfSbhTReWbYeahnq';

async function onTrack(event, settings) {
  if (event.event !== 'Customer Visit') {
    return;
  }

  if (
    !event.properties.homeStore ||
    !event.properties.homeStore.externalStoreIdentifier ||
    !event.properties.orderWasReturned
  ) {
    return;
  }

  const externalStoreIdentifier =
    event.properties.homeStore.externalStoreIdentifier;
  const endpoint = `https://api.birdeye.com/resources/v1/customer/checkin?api_key=${apiKey}&bid=${externalStoreIdentifier}`;
  let response;

  const traits = event.context.traits;
  let name = '';

  function isSet(val) {
    return !!val && val != '';
  }

  if (isSet(traits.firstName) && isSet(traits.lastName)) {
    name = `${traits.firstName} ${traits.lastName}`;
  } else if (isSet(traits.firstName)) {
    name = traits.firstName;
  } else if (isSet(traits.lastName)) {
    name = traits.lastName;
  }

  const payload = {
    name,
    phone: traits.phone,
    emailId: traits.email,
    smsEnabled: traits.receiveTexts ? 1 : 0,
    employees: [
      {
        emailId: event.properties.staff.staffEmail,
        firstName: event.properties.staff.staffFirstName,
```

4. Replace the text **INSERT_API_KEY** with your Birdeye API key. Please see the Prerequisites section for how to get the API key from Birdeye. Then press configure.

5. Name the function Birdeye Customer Visit and press Create Function:

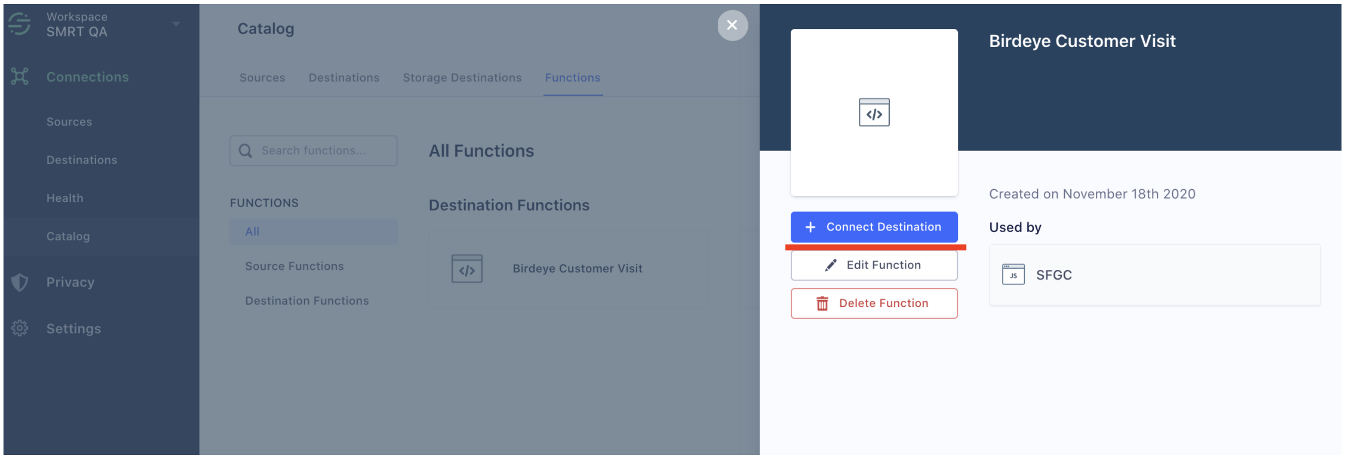


Name *

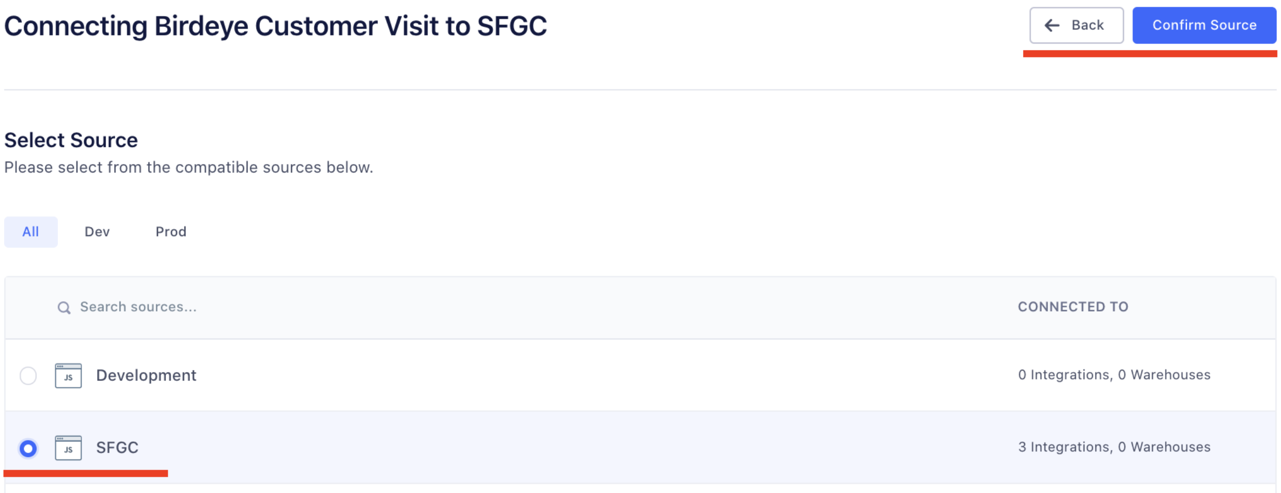
Birdeye Customer Visit

Description

6. After creating the function, a popup should appear on the right side of the screen showing a button **Connect Destination** that you should press. You can get back to this screen at a later time, by going to Connections => Catalog => Functions => Birdeye Customer Visit.



7. Select the SMRT source that you already setup (see the Prerequisites section above), and press **confirm source**.



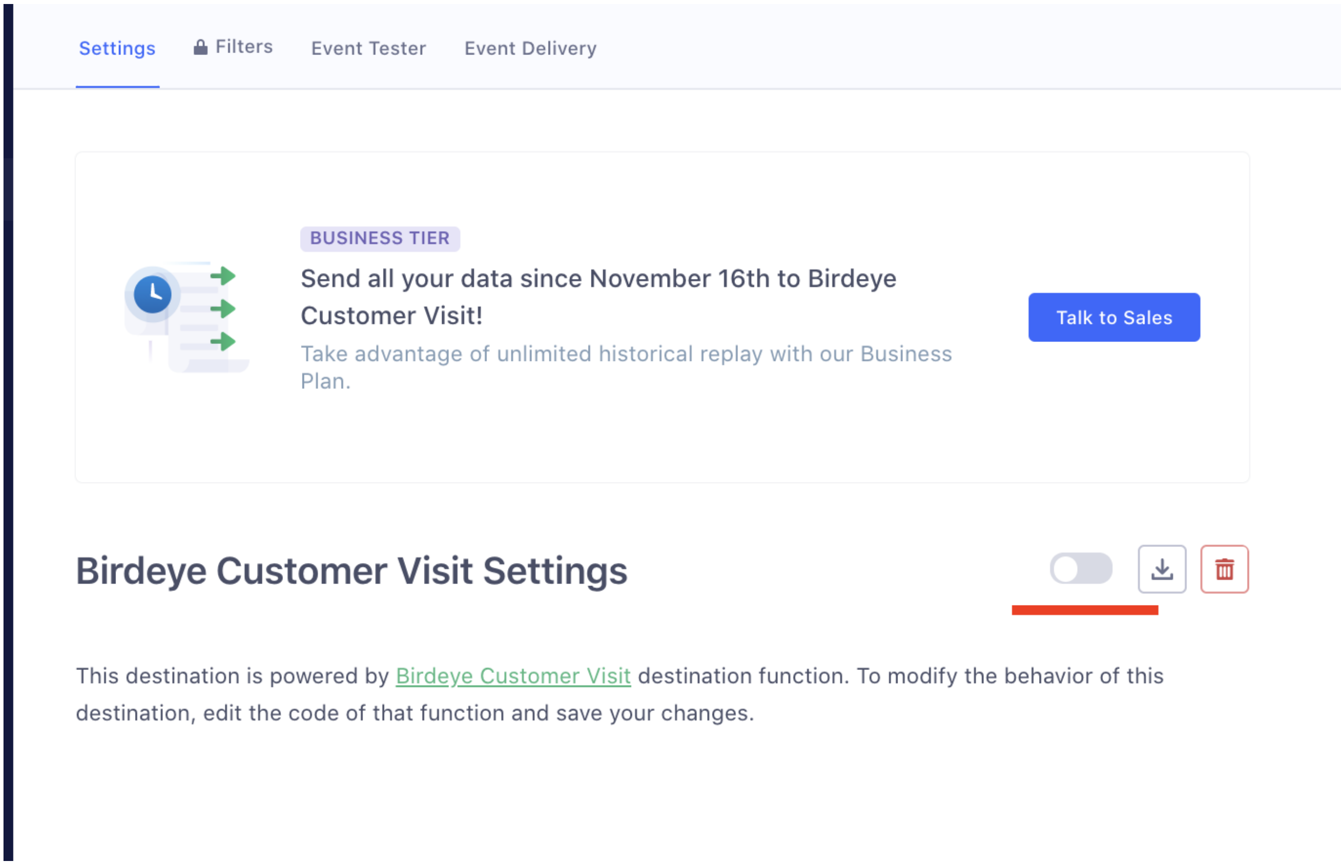
Connecting Birdeye Customer Visit to SFGC

Select Source

Please select from the compatible sources below.

	CONNECTED TO
○ Development	0 Integrations, 0 Warehouses
● SFGC	3 Integrations, 0 Warehouses

8. On the next screen, flip the toggle to enable the integration. This activates the integration



BUSINESS TIER

Send all your data since November 16th to Birdeye Customer Visit!

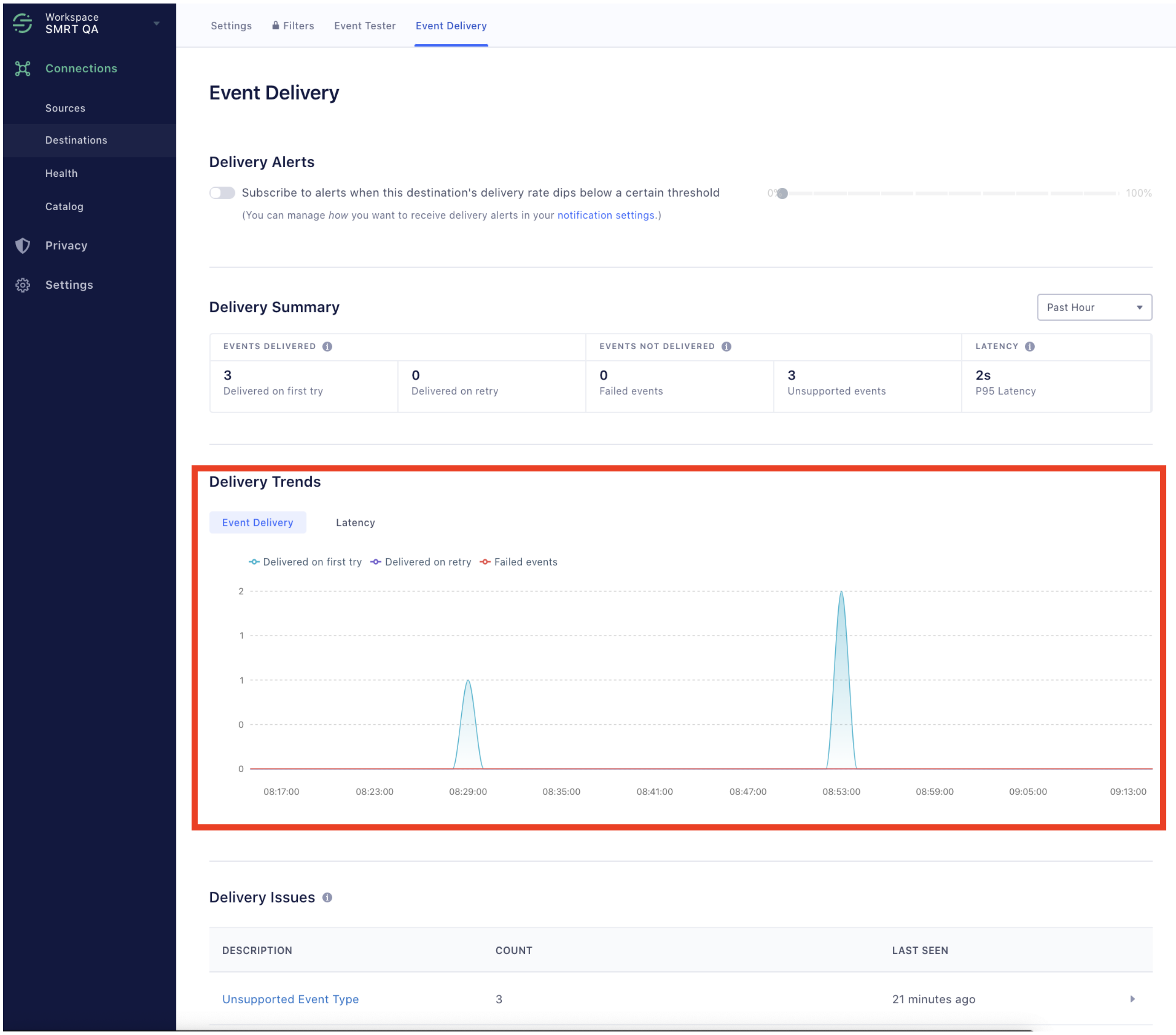
Take advantage of unlimited historical replay with our Business Plan.

Birdeye Customer Visit Settings

This destination is powered by Birdeye Customer Visit destination function. To modify the behavior of this destination, edit the code of that function and save your changes.

Monitoring Integration

In segment, you can go to the delivery tab under the function, to monitor the integration. Please note that event delivery from SMRT is delayed by 10-20 minutes, and only triggers on store pickup or delivery, and not on new orders.



Event Delivery

Delivery Alerts

Subscribe to alerts when this destination's delivery rate dips below a certain threshold 0% 100%

(You can manage *how* you want to receive delivery alerts in your notification settings.)

Delivery Summary

Past Hour

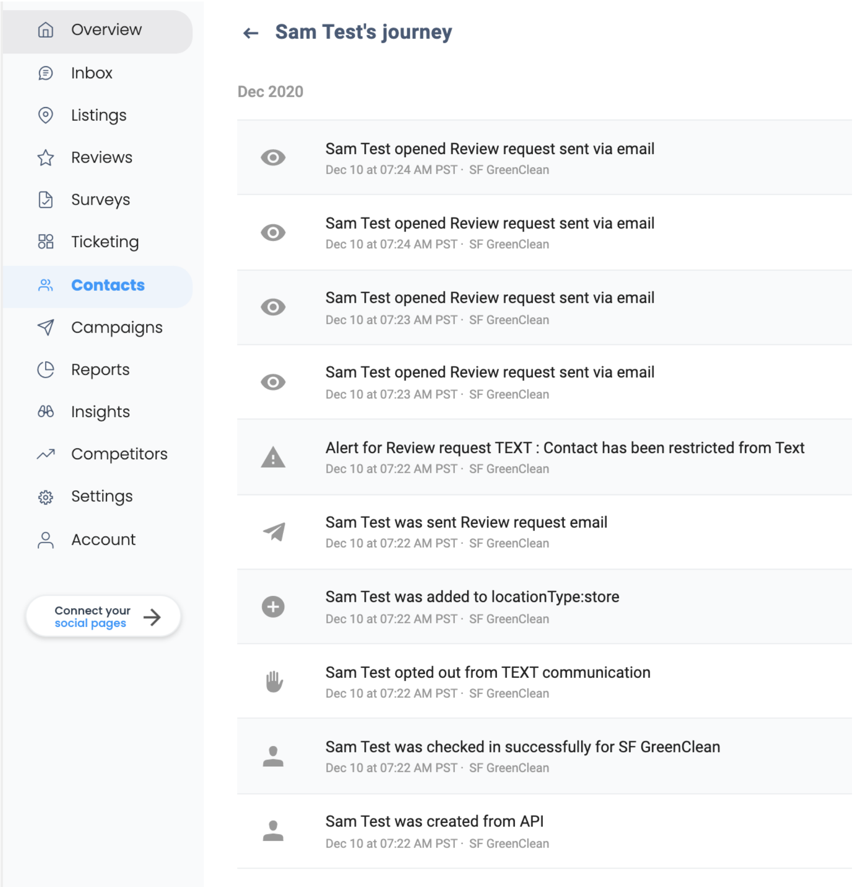
EVENTS DELIVERED		EVENTS NOT DELIVERED		LATENCY
3 Delivered on first try	0 Delivered on retry	0 Failed events	3 Unsupported events	2s P95 Latency

Delivery Trends

Delivery Issues

DESCRIPTION	COUNT	LAST SEEN
Unsupported Event Type	3	21 minutes ago

In Birdeye, you can also monitor the integration by going to the "Contacts" page. Any contacts with recent order returns, should be listed at the top (the page is sorted by last activity). If you hover a contact and press actions => contact journey, you should see the API at work like so:



← Sam Test's journey

Dec 2020

Sam Test opened Review request sent via email
Dec 10 at 07:24 AM PST · SF GreenClean

Sam Test opened Review request sent via email
Dec 10 at 07:24 AM PST · SF GreenClean

Sam Test opened Review request sent via email
Dec 10 at 07:23 AM PST · SF GreenClean

Sam Test opened Review request sent via email
Dec 10 at 07:23 AM PST · SF GreenClean

Alert for Review request TEXT : Contact has been restricted from Text
Dec 10 at 07:22 AM PST · SF GreenClean

Sam Test was sent Review request email
Dec 10 at 07:22 AM PST · SF GreenClean

Sam Test was added to locationType:store
Dec 10 at 07:22 AM PST · SF GreenClean

Sam Test opted out from TEXT communication
Dec 10 at 07:22 AM PST · SF GreenClean

Sam Test was checked in successfully for SF GreenClean
Dec 10 at 07:22 AM PST · SF GreenClean

Sam Test was created from API
Dec 10 at 07:22 AM PST · SF GreenClean

Integration Code

```
//Replace INSERT_API_KEY below with your Birdeye API key.
const apiKey = 'INSERT_API_KEY';

async function onTrack(event, settings) {
        if (event.event !== 'Customer Visit') {
                return;
        }

        if (
                !event.properties.homeStore ||
                !event.properties.homeStore.externalStoreIdentifier ||
                !event.properties.orderWasReturned
        ) {
                return;
        }

        const externalStoreIdentifier =
                event.properties.homeStore.externalStoreIdentifier;
        const endpoint = `https://api.birdeye.com/resources/v1/customer/c
        let response;

        const traits = event.context.traits;
        let name = '';

        function isSet(val) {
                return !!val && val != '';
        }

        if (isSet(traits.firstName) && isSet(traits.lastName)) {
                name = `${traits.firstName} ${traits.lastName}`;
        } else if (isSet(traits.firstName)) {
                name = traits.firstName;
        } else if (isSet(traits.lastName)) {
                name = traits.lastName;
        }

        const payload = {
                name,
                phone: traits.phone,
                emailId: traits.email,
                smsEnabled: traits.receiveTexts ? 1 : 0,
                employees: [
                        {
                                emailId: event.properties.staff.staffEmai
                                firstName: event.properties.staff.staffFi
                                lastName: event.properties.staff.staffLas
                        }
                ],
                additionalParams: {
                        customerAgent: traits.agent,
                        locationType: event.properties.locationType
                }
        };
        try {
                response = await fetch(endpoint, {
                        method: 'POST',
                        headers: {
                                Authorization: `Basic ${btoa(settings.ap
                                'Content-Type': 'application/json'
                        },
                        body: JSON.stringify(payload)
                });
        } catch (error) {
                // Retry on connection error
                throw new RetryError(error.message);
        }

        if (response.status >= 500 || response.status === 429) {
                // Retry on 5xx (server errors) and 429s (rate limits)
                throw new RetryError(`Failed with ${response.status}`);
        }
}
```

Was this article helpful?

Yes No

1 out of 1 found this helpful

Return to top ^

Recently viewed articles

Birdeye Segment Setup

Customer.io Setup

Merging Items With 2 Barcodes

Barcode Placement

Remove & Replace Buttons

Related articles

Email Blasts Through Mailchimp

Delivery Module

Recommended Event Based Emails

Delivering to a Locker

Group Billing Setup

Comments

0 comments

Be the first to write a comment.

SMRT Systems Help Center