



Documentation

Getting Started

What is Segment?
[How Segment Works](#)
Getting Started Guide
A Basic Segment Installation
Planning a Full Installation
A Full Segment Installation
Sending Data to Destinations
Testing and Debugging
What's Next
Use Cases

Guides

Connections

Unify

Engage

Privacy

Protocols

Segment App

API







Partners

Glossary

Config API

Help

Connection Modes

Device-mode	Cloud-mode
 Web	 Web
 Mobile	 Mobile
 Server	 Server

Partner Owned

• This integration is partner owned. Please reach out to the partner's support for any issues.

Our Vero destination code is all open-source on GitHub if you want to check it out: [JavaScript](#), [Server](#).

Getting Started

Vero helps you send targeted emails to customers based on their behavior.

When you enable Vero in the Segment web app, your changes appear in the Segment CDN in about 45 minutes,

and then Analytics.js starts asynchronously loading Vero's `m.js` onto your page. This means you should remove Vero's snippet from your page.

Vero only records custom events and custom user data so no events or users will appear in Vero until you start using the API outlined below.

Identify

Client Side

When you call [Identify](#) on analytics.js, Segment augments `traits` to have `traits.id` set to the `userId`, and then call Vero's user with the augmented traits object. You should provide both a `traits.email` and a `userId` for Vero to work best.

If no `email` is attached, the user is created in Vero but cannot be emailed. If you send omit the `userId`, Vero will use the email as the ID, which is is not recommended. Check out the [Vero docs](#) for more information.

A `userId` is a required value for all types of calls. Be sure you call Identify with a `userId` for subsequent track calls to populate into Vero correctly. For server side calls, you will have to manually pass in the `userId` at the top level.

Server Side

When you call [Identify](#) from one of Segment's server-side languages, Segment calls Vero's REST API and update the traits for the customer with that `userId`. If your `userId` is an email, Segment sets the trait `email` as your `userId` in the update call.

Track

When you call [Track](#), Segment sends the event to Vero with the event `name` and `properties` you provide. Events will be matched to the current user.

You can also unsubscribe users by sending a Track event, passing in the user's ID as a property, like so:

```
analytics.track('Unsubscribe', {
  id: '12345'
});
```

Note: If you'd like to explicitly specify a user's email with track events that is not an event metadata, you can send that under `context.traits.email`.

Sending Data from Vero

Vero supports sending [email events](#) to other tools on the Segment platform. These events will be sent as Track calls to the other destinations you've turned on.

To enable this feature,

- 1 Log into Vero and go to Settings
- 2 Then go to [Integrations](#)
- 3 Hit 'view' next to the Segment integration
- 4 Enter in your Segment write key at the bottom.

3

In order for Vero to send Segment events about email bounced, delivered, opened, clicked and marked as spam, you need to enter your **Segment writeKey** below. You can find it on the Segment dashboard > Project Settings > API Keys.

Segment writeKey

Required. The write key is used by Vero to send data to your Segment project

Save

Need further help? Please contact our team: support@getvero.com

Group

When you call [Group](#), the `traits` included in the call will be set to the current user's **Group** property in Vero.

Alias

Segment's [Alias](#) method can be used from your server to "re-identify" an existing user identity to a new one.

Most of the time this happens when you identify a visitor by their email address after they opt in, then later re-identify with a database ID when they become registered users.

To connect the two identities you'll need to [Alias](#) their current identity to their new one.

Here's a python example of using [Alias](#) to update the identity from an email address to a database ID:

```
analytics.alias('example@example.com', '8765309')
```

Features

Tags

The destination is capable of both adding and removing tags in Vero for a given user. Because `tags` is not a common property of events, this functionality is invoked using an [destination specific option](#).

To start using this feature, pass an object called `tags` with the following properties:

FIELD		TYPE	DESCRIPTION
<code>id</code>	<i>optional</i>	String	The user Id to associate tags with. If this is not specified, the destination will simply use the <code>userId</code> from the event itself.
<code>action</code>	<i>required</i>	String	Must be either 'add' or 'remove'. Indicates whether you would like to add or remove the tags for the given user.
<code>values</code>	<i>required</i>	Array	An array of strings representing the tags to either add or remove.

Here is an example using Segment's Node.js library:

```
analytics.identify('324LKJF', {
  email: 'example@example.com'
}, {
  integrations: {
    Vero: {
      tags: {
        id: '235FAG', // This is optional! If not defined, we default to the event's userId (ie: 324LKJF)
        action: 'add',
        values: ['warriors', 'giants', 'niners']
      }
    }
  }
})
```

Engage

You can send computed traits and audiences generated using [Engage](#) to this destination as a **user property**. To learn more about Engage, schedule a [demo](#).

For user-property destinations, an [identify](#) call is sent to the destination for each user being added and removed. The property name is the snake_cased version of the audience name, with a true/false value to indicate membership. For example, when a user first completes an order in the last 30 days, Engage sends an Identify call with the property `order_completed_last_30days: true`. When the user no longer satisfies this condition (for example, it's been more than 30 days since their last order), Engage sets that value to `false`.

When you first create an audience, Engage sends an Identify call for every user in that audience. Later audience syncs only send updates for users whose membership has changed since the last sync.



Real-time to batch destination sync frequency

Real-time audience syncs to Vero may take six or more hours for the initial sync to complete. Upon completion, a sync frequency of two to three hours is expected.

Settings

Segment lets you change these destination settings from the Segment app without having to touch any code.

SETTING	DESCRIPTION
API Key (required)	<code>string</code> . You can find your API key in your Vero Settings .
Auth Token (required)	<code>string</code> . Your auth token allows you to send data to Vero through our server-side libraries. You can find your <code>auth_token</code> on Vero's account page .

This page was last modified: 27 Oct 2023

Need support?

Questions? Problems? Need more info? Contact Segment Support for assistance!

[Visit our Support page](#)

Help improve these docs!

[Edit this page](#)

[Request docs change](#)

Was this page helpful?

 Yes

 No

Get started with Segment

Segment is the easiest way to integrate your websites & mobile apps data to over 300 analytics and growth tools.

Your work e-mail

[Request Demo](#)

or

[Create free account](#)

© 2025 Segment.io, Inc.

[Privacy](#)

[Terms](#)

[Website Data Collection Preferences](#)

