



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

Vidora provides a machine learning platform for marketing, retention, and product teams to quickly and easily transform raw consumer data into valuable business decisions. Examples include: [next-best-action](#), next-best-offer, [dynamic decisioning](#), [predictions](#), and prescriptive modeling.

This destination is maintained by Vidora. For any issues with the destination, [contact the Vidora Support team](#).

Getting Started

- 1 From the Destinations catalog page in the Segment App, click **Add Destination**.
- 2 Search for "Vidora" in the Destinations Catalog, and select the "Vidora" destination.
- 3 Choose which Source should send data to the "Vidora" destination.
- 4 Go to the [Vidora dashboard](#), find and copy the "API key".
- 5 Enter the "API Key" in the "Vidora" destination settings in Segment.

Track

If you aren't familiar with the Segment Spec, take a look at the [Track method documentation](#) to learn about

what it does. An example call would look like:

```
analytics.track("event_name", {
  property1: "example",
});
```

When sending any data to Vidora, a `timestamp` and either a `userId` or `anonymousId` are required.

Segment sends Track calls to Vidora as a `track` event with the `event_name` as the event type and will include any corresponding `properties`.

Page

If you aren't familiar with the Segment Spec, take a look at the [Page method documentation](#) to learn about what it does. An example call would look like:

```
analytics.page({
  name: "page_name",
  properties: {
    property1: "example",
  },
});
```

When sending any data to Vidora, a `timestamp` and either a `userId` or `anonymousId` are required.

Segment sends Page calls to Vidora as a `pageview` event along with the `page name` and corresponding `properties`.

Screen

If you aren't familiar with the Segment Spec, take a look at the [Screen method documentation](#) to learn about what it does. An example call would look like:

```
[[SEGAnalytics sharedAnalytics] screen:@"screen_name"
  properties:@{ @"property1": @"example" }];
```

When sending any data to Vidora, a `timestamp` and either a `userId` or `anonymousId` are required.

Segment sends Screen calls to Vidora as a `screenview` event along with the `screen name` and corresponding `properties`.

Identify

If you aren't familiar with the Segment Spec, take a look at the [Identify method documentation](#) to learn about what it does. An example call would look like:

```
analytics.identify("userId123", {
  trait1: "example",
});
```

When sending any data to Vidora, a `timestamp` and either a `userId` or `anonymousId` are required.

Segment sends Identify calls to Vidora as an `identify` event along with any corresponding user `traits`.

Group

If you aren't familiar with the Segment Spec, take a look at the [Group method documentation](#) to learn about what it does. An example call would look like:

```
analytics.group("groupId", {
  name: "Group Name",
  trait1: "example",
});
```

When sending any data to Vidora, a `timestamp` and either a `userId` or `anonymousId` are required.

Segment sends Group calls to Vidora as an `group` event along with any corresponding user `traits`.

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 Vidora 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 <i>(required)</i>	<code>string</code> . You can find your Vidora API key in the API>Documentation section of Cortex

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

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](#)

