need to export the timestamp. **If no timestamp is specified when importing, the data will show a timestamp from the time the data was received**.

2. Decide which destinations need to receive the data.
By default, data coming into Segment will be forwarded to all destinations connected to a given source. To limit data to specific destinations, the `integrations` object must be modified. With historical data, you often only want to send the data to a specific destination or into your data warehouse. For example, in Node.js set the `integrations` object as follows.

```
analytics.track({
    event: 'Upgraded Membership',
    userId: '97234974',
    integrations: { 'All': false, 'Vero': true, 'Google Analytics': false }
 })
```

3. Once you've done that, you'll need to write an application or worker to send the data to Segment.
You will need to cycle through each set of data and map it to a Segment server-side library method or build an array matching the HTTP Import API format.
Tip: Segment recommends using a Segment library for this process, as they set contextual message fields like `message_id` (used for deduping) and `sent_at` (used for correctly client clock skew) that Segment's API uses

to correct behavior upon ingestion.

Tip: The server-side libraries will automatically batch requests to optimize for performance and prevent linear request volume. This batching behavior is modifiable, and some of the underlying libraries implement a configurable max queue size that may discard messages if you enqueue requests much faster than the client can flush them. We recommend overriding the max queue size parameter for the library to a high value you're comfortable you can remain under in your batch job.

Demo projects

The following projects are open-source and do not have official Segment support. If you encounter issues, the best way to get help is by opening an issue on the project's GitHub page. Feel free to clone the repository and adjust the code to suit your unique needs.

One of Segment's Success Engineers wrote an alpha prototype Node.js app for importing data utilizing the HTTP API, which we've included below:

Example Node.js import application

Additionally, one of Segment's Software Engineers developed a React App with more out of the box functionality for importing events. The features include a modern UI, transformations, and event format checking prior to import:

Desktop React CSV uploader

MarketLytics has documented their experience using the alpha prototype importer and offer some helpful visuals and tips.

Alternative solution

If a server-side library doesn't meet your needs, you can use the Segment bulk import HTTP API directly.

Note: When you use the HTTP API to export historical data to upload to Segment, remove all the original `sent_at`, `message_id`, and `project_id` fields from the archived message before forwarding them back to Segment.

Method 2: Using Reverse ETL

Please refer to the Reverse ETL guide for more details.

This page was last modified: 14 May 2024

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

Privacy

Terms

Website Data Collection Preferences