



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
<input checked="" type="checkbox"/> Web	<input type="checkbox"/> Web
<input type="checkbox"/> Mobile	<input type="checkbox"/> Mobile
<input type="checkbox"/> Server	<input type="checkbox"/> Server

Getting Started

When you enable Ramen in the Segment web app, your changes appear in the Segment CDN in about 45 minutes, and then Analytics.js starts asynchronously loading Ramen's JavaScript library onto your page. This means you should remove Ramen's snippet from your page if you already have it there.

- As of March 2016, Ramen supports asking questions of anonymous website visitors in addition to logged-in users. Calling `page`, `track`, and `group` will work without calling `identify`. However, when you call `identify` you

must pass in an `id` or else the Ramen destination will ignore the call.

Identify

When you make an `Identify` call on Analytics.js, it identifies the logged-in user to Ramen. Here is a full example, which we will annotate below:

```
analytics.identify('314159', {  
  email: 'ryan@charturl.com',  
  name: 'Ryan Angilly',  
  created_at: 1234567890,  
  is_profitable: true,  
  plan: 'Startup',  
  mrr: 149.99  
});
```

Here is a list of the attributes we process:

- `id` becomes the `id` of the user in Ramen.
- `email` becomes the `email` of the user in Ramen.
- `name` becomes the `name` of the user in Ramen.
- `created_at` becomes the `customer_created_at` timestamp for the user in Ramen. It can be passed in as a `Date` or seconds since the epoch.
- If `company` is an `Object` with an `id` attribute, we'll treat that information as if it was passed to `group` and create a company in Ramen.
- Attributes with names ending in `_at` will be parsed as times.
- Attributes with names beginning with `is_` will be parsed as Booleans. **Note:** Ramen will ignore any calls to `identify` that do not contain `userId`.

Group

Segment supports Ramen Companies in Analytics.js through the `group` method. Users can be put into multiple groups, which will associate them to multiple companies in Ramen. Here is a full example, which we will annotate below:

```
analytics.group('314159', {  
  name: 'Ryan Angilly',  
  url: 'https://charturl.com',  
  created_at: 1234567890,  
  is_awesome: true,  
  plan: 'Startup',  
  mrr: 149.99  
});
```

Here is a list of the attributes we process:

- `id` becomes the `id` of the company in Ramen.
- `url` becomes the `url` of the company in Ramen.
- `name` becomes the `name` of the company in Ramen.
- `created_at` becomes the `company_created_at` timestamp for the company in Ramen. It can be passed in as a `Date` or seconds since the epoch.
- Other attributes can be passed in as well, and will be parsed according to the same rules as denoted in the `identify` section.

Page

When you call `page` on Analytics.js, it registers a new pageview in Ramen and checks to see if any questions should be asked. Ramen does not support passing in any attributes to `page`. You can call `page({ title: "Our page" })` but the attributes (ie. `title` in this example) will be ignored.

Track

When you call `track` on Analytics.js, it registers a new event in Ramen and checks to see if any questions should be asked. Ramen does not support passing in any attributes to `track` beyond the event name. You can call `track('Subscribed', { plan: "Startup" })` but the attributes (ie. `plan` in this example) will be ignored.

Features

Secure Mode

If you want to enable Ramen [secure mode](#) for analytics.js, you can pass in the `timestamp` and `auth_hash` variables by rendering it in your server-side templates. The `timestamp` should be a Unix timestamp (epoch seconds). The `auth_hash` is a SHA256 hash of several attributes. The hash is not based on the email, it is based on:

- `user.email` - If you do not store emails for your customers, use an empty string
- `user.id`
- `user.name` - If you do not have a name for your customers, use their email address or an empty string
- `timestamp`

Ramen API Secret Here's an example of a Ruby web server rendering an identify call with secure mode:

```
analytics.identify('<%= current_user.id %>', {
  email : '<%= current_user.email %>',
  createdAt : <%= current_user.created_at.to_i %>
}, {
  integrations: {
    Ramen : {
      timestamp: <%= @ts ||= Time.now.to_i %>,
      auth_hash : '<%= (Digest::SHA256.new << "#{current_user.email}:#{current_user.id}:#{current_user.name}:#
{@ts}:RAMEN_ORGANIZATION_SECRET_KEY").to_s %>'
    }
  }
});
```

`RAMEN_ORGANIZATION_SECRET_KEY` is found in Ramen's RamenJS documentation.

Troubleshooting

I'm seeing a 404 error

Check to make sure your Ramen `ORGANIZATION_ID` is correct.

I'm seeing a 401 error

The request you are sending is not valid. Several things can cause this:

- An invalid `user.email`. It can be blank, but it cannot be an invalid email address.
- A missing `user.id`.
- Not including `timestamp` and `auth_hash` once Secure Mode has been enabled. You can see if Secure Mode is enabled by visiting your RamenJS settings page in Ramen. If the above all look correct, and you are sending `timestamp` and `auth_hash`, check the following:
- `timestamp` is being dynamically generated (ie. it is not accidentally hard-coded).
- `auth_hash` is being calculated correctly If you are still having trouble, you can email [Ramen support](#).

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 Ramen 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	<code>string</code> . Your Ramen API key for the server-side integration
Organization Id	<code>string</code> . Your Ramen organization id

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](#)

