Track JS monitors your web applications for JavaScript errors, alerting you with amazing context about how the user, application, and network got into trouble. The `analytics.js` trackJs Destination is open-source. You can browse the code on GitHub.

Getting Started

1. From the Segment web app, click **Catalog**.
2. Search for "Track JS" in the Catalog, select it, and choose which of your sources to connect the destination to.
3. Enter your Token as retrieved from your Track JS set up page.
4. Your changes appear in the Segment CDN in about 45 minutes, and then Analytics.js starts asynchronously loading Track JS onto your page. This means you should remove Track JS' snippet from your page.
5. The Track JS Destination doesn't use any Segment API calls (e.g. identify, track, etc) so, once it's loaded, it will automatically start recording error data.

Non-supported options

We do not currently support Track JS' `onError` and `serialize` options because they pose a XSS vulnerability.

To work around this issue, however, you can directly set these options in the `window._trackJs` object on your page. These options will be merged with the settings you have configured in the interface once the Track JS script is loaded.

We also do not support `version`, or `sessionId`. These can either be set prior to initialization as described above, or can be set after the snippet has been loaded with the `ready` callback

```
analytics.ready(function(){
  window.trackJs.configure({ sessionId: '123456', version: '1.0' });
});
```

Note that the `userId` parameter is automatically determined.

Settings

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

SETTING	DESCRIPTION
Application Key	`string`. Optional. The name of the section of your application that this Segment project encompasses. For more info see here
Callback BindStack	`boolean`, defaults to `FALSE`. Whether TrackJS should record the stacktrace for how a callback came to be bound, allowing the creation of an 'async stacktrace' in the TrackJS UI. This gives you more context of how the application arrived at the function that failed. Enabling this has a nontrivial performance penalty, test before enabling in production
Callback Enabled	`boolean`, defaults to `TRUE`. Whether TrackJS should watch for errors on host-function callbacks, such as addEventListener, and setTimeout. Disabling this greatly decreases the likelihood of recording a stacktrace for an error.
Console Display	`boolean`, defaults to `TRUE`. Whether calls to console.log and other console functions should be displayed into the browser's console.
Console Enabled	`boolean`, defaults to `TRUE`. Whether TrackJS should wrap and watch the console for events. If disabled, window.console is not guaranteed to exist in all browsers.
Console Error	`boolean`, defaults to `TRUE`. Whether TrackJS should transmit errors on calls to console.error.
Enabled	`boolean`, defaults to `TRUE`. Whether to enable TrackJS for the page. If false, the TrackJS watchers will not be set up. The window.trackJs API will still be available and function locally, but no errors will be transmitted from the browser
Network Enabled	`boolean`, defaults to `TRUE`. Whether TrackJS should wrap and watch AJAX calls. If disabled, network events will not appear in your Telemetry Timeline.
Network Error	`boolean`, defaults to `TRUE`. Whether TrackJS should transmit errors when an AJAX call completes with a 'failing' status code, 400 or greater. If you wish to customize the list of status codes you want to capture errors from, set this to true and filter out status codes using the onError callback.
Token *(required)*	`string`. You can find your token under Track JS setup page.

SETTING	DESCRIPTION
Visitor Enabled	`boolean`, defaults to `TRUE`. Whether TrackJS should watch visitor interaction events, such as clicks and field input.
Window Enabled	`boolean`, defaults to `TRUE`. Whether TrackJS should watch window.onerror for global errors

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